19th ICCRTS
"C2 Agility: Lessons Learned from Research and Operations."

Opportunities for Next Generation BML: Semantic C-BML

Topics
Experimentation, Metrics, and Analysis
Modeling and Simulation
Data, Information and Knowledge

Names of Authors
Samuel Singapogu
George Mason University C4I Center

Point of Contact
Samuel Singapogu
George Mason University
4400 University Drive, Fairfax, VA 22030
703-993-3082
ssingapo@c4i.gmu.edu

Report Documentation Page		Form Approved OMB No. 0704-0188

1. REPORT DATE **JUN 2014**	2. REPORT TYPE	3. DATES COVERED **00-00-2014 to 00-00-2014**
4. TITLE AND SUBTITLE **Opportunities for Next Generation BML: Semantic C-BML**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **George Mason University,C4I Center,4400 University Drive,Fairfax,VA,22030**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES **Presented at the 18th International Command & Control Research & Technology Symposium (ICCRTS) held 16-19 June, 2014 in Alexandria, VA. U.S. Government or Federal Rights License**

14. ABSTRACT

Battle Management Language is an unambiguous language to facilitate the command and control of forces and equipment in a military environment and to provide information for situational awareness. Coalition Battle Management language (C-BML) is a standardization effort to define BML in a Coalition environment to support information exchange among a range of simulation systems and command and control (c2) Systems. A SISO Product Development Group (PDG) was formed in the spring of 2006 to implement the standardization process. The group has concluded the first phase of development; that phase is focused on the formalization of syntax. Recent coalition experiments have shown that complex XML schemas can impede development and testing speed . This paper demonstrates that an ontology-based model can provide better readability and allow users to generate necessary XML components. A semantically enriched C-BML can support processing C-BML expressions based on semantic constraints. This paper presents an analysis of the applicability of the current standards of semantic representation languages including Resource Description Framework (RDF) and Web Ontology Language (OWL) to a C-BML Ontology. We explore ontological alignment of C-BML Ontology with upper level ontologies for time. Based on recent experiments of the NATO Modeling and Simulation Group technical activity 085, we present use-cases and benefits of having ontological reasoning as a means of processing C-BML documents.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**38**	

Abstract

Battle Management Language is an unambiguous language to facilitate the command and control of forces and equipment in a military environment and to provide information for situational awareness. Coalition Battle Management language (C-BML) is a standardization effort to define BML in a Coalition environment to support information exchange among a range of simulation systems and command and control (c2) Systems. A SISO Product Development Group (PDG) was formed in the spring of 2006 to implement the standardization process. The group has concluded the first phase of development; that phase is focused on the formalization of syntax.

Recent coalition experiments have shown that complex XML schemas can impede development and testing speed . This paper demonstrates that an ontology-based model can provide better readability and allow users to generate necessary XML components. A semantically enriched C-BML can support processing C-BML expressions based on semantic constraints. This paper presents an analysis of the applicability of the current standards of semantic representation languages including Resource Description Framework (RDF) and Web Ontology Language (OWL) to a C-BML Ontology. We explore ontological alignment of C-BML Ontology with upper level ontologies for time. Based on recent experiments of the NATO Modeling and Simulation Group technical activity 085, we present use-cases and benefits of having ontological reasoning as a means of processing C-BML documents.

1. Introduction

Battle Management Language was developed to represent and exchange digitized Command and Control (C2) information among C2 and simulation systems. The language makes use of a grammar called the C2 Lexical Grammar (C2LG)[1]. Since its development, BML has been in a number of applications and scenarios ranging from proof of concept for military applications, testing interoperability of C2 and simulation Systems for use in real-time C2 operational environment. In the fall of 2006 Simulation Interoperability Standards Organization formed a Product Development Group to oversee the development of a standard Language that can be used in a coalition environment [2].

In a world where alliances and coalitions are increasingly significant to military operations, there is need for a language that can work not only across multiple environments but also harness the power of semantic BML for meaningful interoperability [3]. C-BML is based on a significant body of research [4].

Recent coalition experiments [23] have shown that working with a complex XML schema limits the speed of developing, integrating and testing C2 systems and simulation systems. Full compliance with a complex schema requires a system to parse and process for every element in the schema regardless of whether or not most elements in the schema are used. In the recent MSG-085 experiments, most elements in a complex schema (Phase 1 draft schema of C-BML) did not prove necessary. A formal ontology-based model, combined with procedures to create and extend XML syntax, enhances readability and helps with an efficient prototyping and testing process. Such an ontology captures entities, relationships from a data model and allows users to focus on and even reason with necessary components. In addition, there has been research to show a process to create XML schemas from an ontology [24]. This allows users to creating a workable, non-complex schema that's appropriate for their use and yet compliant with the standard.

This paper addresses possibilities for an ontology-based approach to C-BML. Section 2 describes the current state of the standardization process of C-BML. Section 3 explores the benefits of having semantics formalized as part of the CBML standardization process. Sections 4 and 5 review the current state-of-the-art and standards in the Semantic Web and ontologies. Section 6 demonstrates how a C-BML ontology can use existing upper level ontologies to map entities. Section 7 explores how this ontology can help C-BML applications. Section 8 shows how reasoning can be used in a C-BML ontology to identify logical inconsistencies and derive inferences. The final sections make concluding remarks and address future work.

2. Status of SISO Coalition Battle Management Language

Phase 1 of the SISO C-BML Product Drafting Group (PDG) has finalized a formal schema for C-BML composites and guidelines on using the composites to create C-BML expressions (such as Plans, Requests, Orders and Reports). The underlying data model

definition is in the form of a XML schema. Its vocabulary is based on the Joint Consultation Command and Control Information Exchange Data Model (JC3IEDM). The C-BML Phase 1 effort also has specified the interdependence of C-BML with the SISO Military Scenario Definition Language.[6]

The next Phase of the C-BML standard is focused on developing the semantic framework for C-BML. The semantic framework should be able to expand on the data model and XML schemas from Phase 1 to create a semantic layer for C-BML that can capture the entities, their properties and relationships.

3. Semantic Requirements for Next-Generation BML

The main goal of the Semantic Web is to be able to structure and link data in a machine-readable knowledge representation. A knowledge representation system can also make use of a classifier to dynamically infer new classes/concepts and expand an ontology. Reference [15] provided an early discussion on building an ontology for C-BML. It noted that a C-BML ontology is needed for the following reasons

i)   An ontology formalizes the definition and meaning of common terms
ii) It formalizes the doctrinal rules for Orders and Reports
iii) It eases interoperability because of shared vocabulary and meaning
iv) It allows performing powerful reasoning on operational semantics

[16] describes the integration of C-BML Phase 1 and MSDL. The experiment observes that integration between a simulation system (JSAF) and a C2 system had irregularities that could have been solved by automated rules. These rules can be very easily incorporated into an ontology, while a purely syntactic C-BML would need an overlay system that raises a new set of integration issues. All entities in C-BML are structured around the "five Ws"- *Who*, *What*, *When*, *Where*, and *Why*. *Who* is conceptually any object (OBJECT_ITEM in JC3IEDM schema space). This maps to a class or concept in an ontology. The rest of the Ws can be mapped to attributes of this concept (or other concepts).

The current standard for knowledge representation is Resource Description Framework (RDF). All resources are identified with a URL, essentially all "things" can be represented in RDF. This makes it a feasible basis for a C-BML ontology. The current standard for Ontology and rule specification is Web Ontology Language (OWL). It is based on Description Logic; [17] demonstrates that Description Logic (and therefore OWL) is capable of reasoning through semantics that can be reduced to set-theory operations. Based on C-BML experimentation efforts to define the Phase 1 XML Schema, no relationships or rules have been established that are beyond set theory relationships. Therefore, OWL should be sufficient as a language for C-BML. Reference [19] supports this by outlining reasons why OWL is suitable for C-BML.

C-BML Phase 1 has identified the vocabulary and the construct for defining valid BML expressions in a coalition environment. Adding a semantic layer to it has the following advantages:

a) Common vocabulary and understanding: Although the schema for C-BML has been formalized, it is still quite likely that elements in the XML document might have different interpretations across systems. Having an ontology will formalize what each element means and how they are related to each other.

b) Allowing extensions: Formalizing a C-BML ontology will make it possible for other languages to use and even extend C-BML without affecting the semantic consistency of the elements in C-BML. This will be useful for efforts such as GeoBML[18] that apply BML to specialized contexts. This result will enrich the applications that are created in the C2 environment.

c) Within coalition C2 environment, there are a number of domain assumptions that are assumed but are not captured in an XML schema- for example, relationships between units and areas. A formalized C-BML ontology will make explicit these domain assumptions.

4. The Semantic Web

The Semantic Web is a collaborative effort based on W3C standards to capture semantics in a machine understandable way. The goal of the Semantic Web is to have standards and mechanisms that make is possible for systems to have not only data but also the meaning and relationships of data (knowledge representation, structured data and linked data). The semantic data can be represented in a number of ways, the most popular of which are the Resource Description Framework (RDF) and the Web Ontology Language (OWL) standards. RDF is based on representing resources using a Uniform Resource Identifier (URI) and linking them through a triplet of a Subject, Predicate and Object. The OWL standard can be used to create and capture a rich, complex representation of Knowledge that can be used for reasoning, checking for consistencies and making inferences and implicit assertions.

Knowledge Representation and Semantic web standards have evolved to make it possible to capture and represent semantic data. The current prevalent standards for knowledge representation are RDF, RDF Schema (RDFS) and OWL. The most basic element in the semantic web is a URI. The URI can be used to identify any piece of information irrespective of its complexity. RDF defines a framework to define a triplet consisting of <subject> <predicate> <object>. The subject and the object are URIs while the object can be either a literal or a URI. Defining a triplet allows a machine to not only understand the type of information but also the relationships within the information. For example:

<http://…Unit:UnitA><http://…UnitRelationship:hasAsCommander>   <http://…Unit: UnitB>

The above example tells the semantic process that there are two Units of type Unit that have a relationship called "hasAsCommander". The three main components of RDF are: Resources, Properties and Classes. Resources are anything that can be identified by a URI or a literal. Properties are relationships that may exist among resources. Classes are groupings of similar resources. RDF has a schema (RDFS) that allows for content created using RDF to be serialized to XML. RDF is a flexible, scalable way to define data and their relationships.

5. Ontologies

Ontologies are a way of knowledge representation using concepts and relationships between them. OWL is based on RDF but adds richness to definitions and relationships. For example, OWL allows for definitions of relationships between classes (complement) and property inferences (symmetric, transitive). OWL representation makes it possible to make inferences using Description Logic. This helps in the enrichment of knowledge by making implicit knowledge explicit. The high level abstract Ontology representation in OWL is through annotations, axioms and facts. Facts are simple assertions about entities. Axioms are assumed knowledge in the Ontology. Annotations are machine-readable meta-data of an ontology. An ontology is like a highly enriched Data Dictionary that makes it possible to have a common vocabulary in a domain.

6. Higher Level Ontologies for C-BML

Ontologies are designed to be reusable [20]. C-BML can benefit by reusing existing, applicable ontologies. In this section we demonstrate a process of mapping C-BML concepts to an existing ontology for time called OWL-Time. OWL-Time was design to be an upper level Ontology to represent time in different forms. The following example maps a C-BML *When* (one of the '5Ws' of C-BML) to entities in OWL-Time [7]. This process can be repeated for other applicable C-BML entities and upper-level Ontologies.

OWL-Time has at its core the class ":TemporalEntity". This class has two subclasses:
a) :Instant – This can be used to represent a point in time without any interior points
b) :Interval - This can be used to represent a period of time

C-BML defines a *When* element as a description of a timeframe in which an action is to occur (Order or Request) or when an action or event has occurred [5].

OWL-Time can express facts about time instants and intervals and perform temporal associations, assertions and inferences. It is designed to work across time zones and as demonstrated next is capable of capturing time in C-BML   In C-BML, *When* is defined in terms of the following composites: TaskWhenLight, TaskWhen, EventStart, EventEnd and ReportedWhen. These composites use a number of lower level elements. Some of them use JC3IEDM codes and cannot be mapped to an element in OWL-Time. Such elements   are: jc3iedm:ActionTaskTimingDayCode, jc3iedm:ActionTaskTimingHourCode, jc3iedm:DatetimeTypeFix18 and eight others. These are codes that qualify the time in a C-BML environment. There are two elements in the C-BML specification that capture time applicable to OWL Time. They are jc3iedm:DurationType19 and jc3iedm:DatetimeTypeFix18. jc3iedm:DurationType19 is of type integer. This captures the duration of time and can be mapped to the class :DurationDescription in OWL-Time. C-BML requires that Duration be of type integer whereas DurationDescription can capture integers along with qualifiers as to whether the duration corresponds to seconds, minutes all the way until years. DatetimeTypeFix18 is a string literal specified as a chronological point measured using Coordinated Universal

Time (UTC). The ISO notation used is "YYYYMMDDHHMMSS.SSS" to represent time in years, months, days, hours, minutes, and seconds/milliseconds. DatetimeTypeFix18 can be mapped to the DateTimeDescription class which is a super set of the ISO notation in that it can additionally capture dayOfWeek and dayOfYear. An example mapping from C-BML TaskWhenLight to OWL-TIME follows:
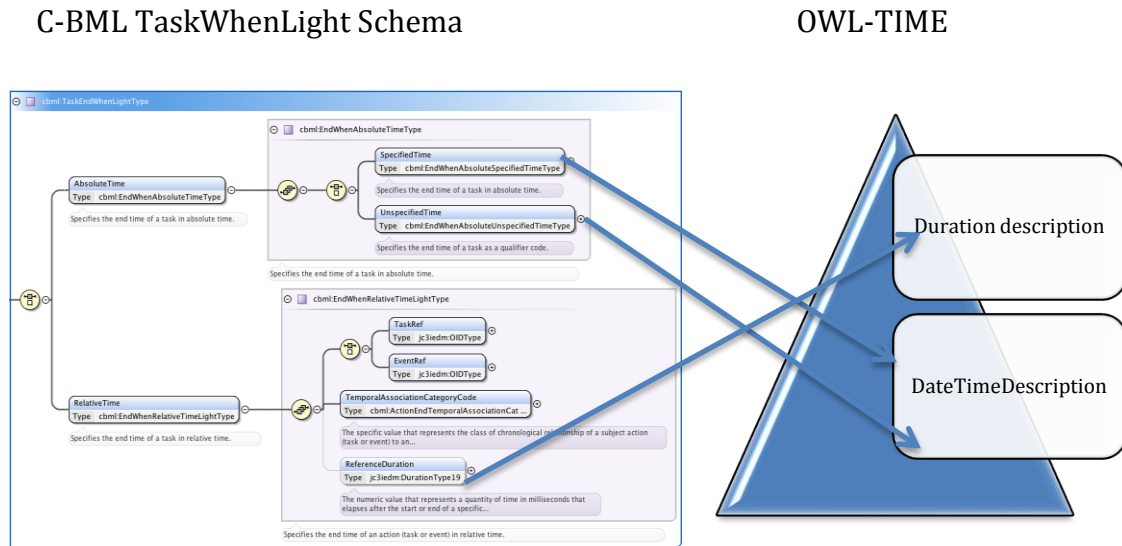
C-BML TaskWhenLight Schema   OWL-TIME

Figure 1: Mapping from TaskWhenLight to OWL-TIME

```
:DateTimeDescription
    a     owl:Class ;
    rdfs:subClassOf
        [ a     owl:Restriction ;
          owl:cardinality 1 ;
          owl:onProperty :unitType
        ] ;
    rdfs:subClassOf
        [ a     owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :second
        ] ;
    rdfs:subClassOf
        [ a     owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :minute
        ] ;
    rdfs:subClassOf
        [ a     owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :hour
        ] ;
    rdfs:subClassOf
        [ a     owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :day
        ] ;
    rdfs:subClassOf
        [ a     owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :dayOfWeek
        ] ;

    rdfs:subClassOf
        [ a     owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :dayOfYear
        ] ;
    rdfs:subClassOf
        [ a     owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :week
        ] ;

    rdfs:subClassOf
        [ a     owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :month
        ] ;
    rdfs:subClassOf
        [ a     owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :year
        ] ;
    rdfs:subClassOf
        [ a     owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :timeZone
        ] .

:hasDateTimeDescription
    a     owl:ObjectProperty ;
    rdfs:domain :DateTimeInterval ;
    rdfs:range
:DateTimeDescription
```

Table 1: The OWL definition of DateTimeDescription

It should be noted that the OWL-Time has many other elements that may not be of interest to the C-BML standard and could be considered as overhead. But, using an existing w3c standard ontology like OWL-Time has the advantage of rich expressiveness and the power of reasoning through the ontology in addition to the possibility of richer Time expressions if C-BML needs it in the future.

Upper Level Ontology for "Where": In C-BML a *Where* is a Geographic feature to represent points, lines, areas and features. There are a few Geographic Ontologies that provide knowledge representation for geographic information. The most applicable

standard for C-BML appears to be the W3C Geographic Vocabulary GEO OWL[8] based on Geo RSS [www.georss.org]. The top class of GEO OWL is "geometry". A ":geometry" can be of type Point, LineString, Polygon or envelope (They are modeled after the elements in Geographic Markup Language).
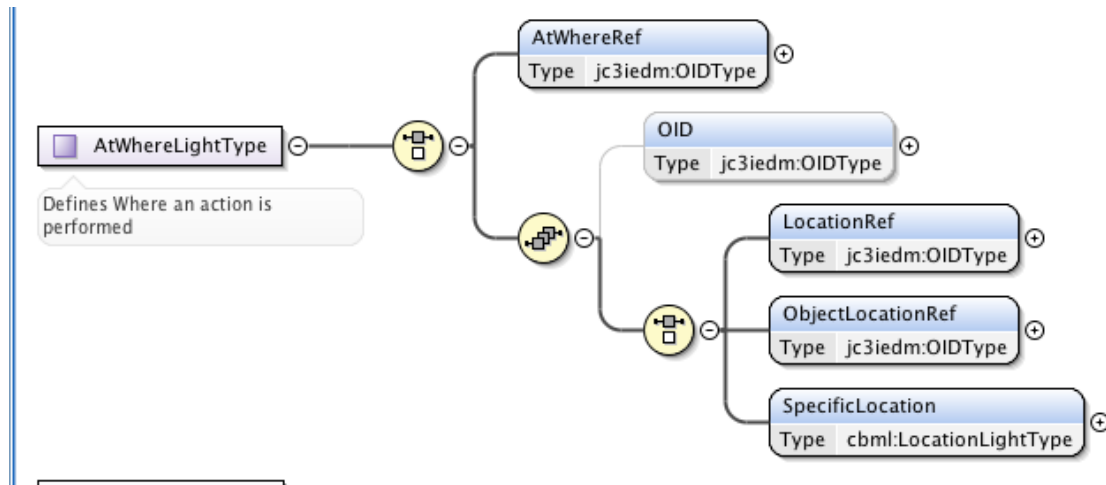

Figure 2 : The definition of "AtWhereLightType" in C-BML Phase1 specification

Fig 3 shows a definition of "AtWhereLight" in the C-BML Phase 1 specification. The most frequently used Location type is a "SpecificLocation" which in turn is a Point, Line, Surface or "CorridorArea". Ontologically, Point and Line map to the corresponding items in Geo OWL and Surface can be mapped to Area and "CorridorArea" can be mapped to a Polygon in Geo OWL. When the C-BML Ontology is formalized, like the *When*, there will be elements in *Where* that cannot be mapped to elements in Geo OWL. But elements such as Point, Line, Surface and "CorridorArea" can be mapped to elements in Geo OWL

7 How can Ontologies help C-BML?

[9] notes that operational BML lacks clearly delineated rules governing its use concerning syntax and semantics. This fact is amplified in a coalition environment. It has already been noted that in the increasingly coalition and interoperability-oriented C2 operations, BML would need to have a formalized, common vocabulary and semantics [10]. Having a C2 ontology would allow for seamless, meaningful exchange of digitized C2 information across C-BML compliant systems. Also, the process of semantic formalizations could raise important discussions where agreement in doctrine interpretation may be lacking. [11] notes that there is no trivial mapping from Syntax to semantics. Therefore, it cannot be assumed that Grammar and XML Schema (Syntax) implicitly define necessary semantics.

8. Reasoning on C-BML

Ontologies capture explicit knowledge in the form axioms and facts. Ontology reasoning uses knowledge reasoning and first order logic reasoning to derive implicit knowledge from the Explicit knowledge and the properties of Entities and relationships. RDFS Schemas define relationship inferences through properties such as rdfs:subclassOf, rdfs:subProperty that allow a reasoner to make inferred relationships. Knowledge can be inferred through equality, reflexivity, and transitivity. Most reasoners use First-Order Predicate logic to derive inferences and expand the Knowledge base, although there has been recent work that suggest probabilistic reasoners such as ELOG [12] or Pronto [13] can also be used. Additionally, abductive reasoning is another form of reasoning that arrives at possible hypothesis based on observations. This is particularly interesting in the context of C-BML because frequently used C-BML expressions are Reports. Reports are observations, typically made on units, objects or areas. With a formalized ontology, these Reports (observations) can be fed to an abductive reasoner to suggest possible hypotheses. These can be helpful to a C2 operator to understand why a particular observation might be important. An example of using reasoning in semantic C-BML to detect semantic errors is illustrated below:

Consider a General Status Report in C-BML. This provides status information on a perceived Executer (Unit/Organisation) at a particular time and place. There can be multiple General Status Reports on the Executer pertaining to the same time and place. It is quite possible (although semantically unreasonable) that these reports provide different locations for the same Executer at the same time. A C-BML implementation would not be able to detect this inconsistency without an additional layer of "unformalized" programming. An ontology with the following rule can detect this inconsistency.

A visualization of a OWL ontology focused on the Executer and its relationship to "Reported Location" and "ReportedTime" can be found below:
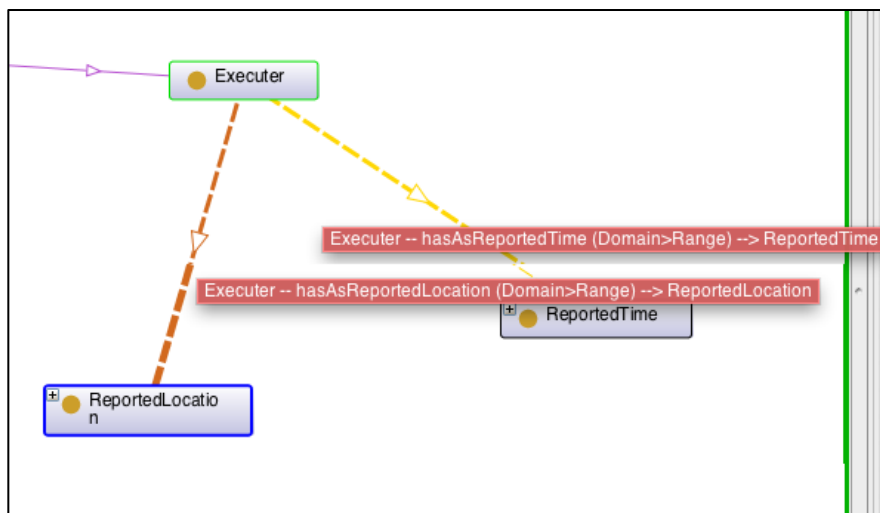

Figure 3: A visualization of a OWL Ontology definition focused on the Executer and its relationship to ReportedTime and ReportedLocation

A human readable Semantic Web Rule Language (SWRL) representation of the rule is shown below:

$$\big(Executer(x), hasAsReportedTime(x, t1)\big)$$
$$\wedge \big(Executer(x), hasAsReportedTime(x, t1)\big)$$
$$\Longrightarrow (hasAsLocation(x, l1) = hasAsLocation(x, l2))$$

This rule is applicable to an Ontology that has a entity called Executer with two object properties- "hasAsReportedTime" and "hasAsLocation". The rule states that for any object "x" of an Executer if that Object has a time "T1" and a new instance of a Report also has the same object "x" with the same reported time "T1", then the location for that Executer should also be the same.

Another example of the use of reasoning in a C-BML Ontology is the use of inferences to derive "new knowledge". This is illustrated in the simple examples below:

Example 1:

Consider the axioms:
ObjectProperty (a:isAUnit domain(a:Tasker) range(a:Unit))
ObjectProperty (a:isAsubordinateOf domain(a:Taskee) (a:Tasker))

The axioms represented by these Object properties are:
- *A Tasker should be a Unit (as opposed to a Equipment)*
- *A Taskee is subordinate to a Tasker*

Now consider an Order that has a Tasker as: "1060: 1st Battalion Commander" and a Taskee as: "1062: Company A". Using the two axioms, we can assert the knowledge:

"'1060: 1st Battalion Commander' is a Unit who is a commanding officer to '1062: Company A'"

Example 2:

Axiom: *ObjectProperty(a:isAfterTask domain(a:Task) range(a:Task)*

allows us to use the transitivity of the "isAfterTask" property so that with the assertions:

Assertion1: *Task1 isAfterTask Task2,*
Assertion2: *Task2 isAfterTask Task3*
The inference is derived:
Inference: *Task1 isafterTask Task3*
Note: This inference can be derived in the Ontology even if the two tasks are in separate C-BML Orders

9. Future Work

The evolution of BML to date has been incremental. A number of NATO Modeling and Simulation Group (MSG) 085 experiments [21] have provided needed feedback. An evolving standard should be able to work through new changes and the MIP Change Proposal (CP) Framework in the MIP Information Model (MIM) is being explored as a framework to preserve MIP compliance[22]. A semantic C-BML should be able to align with the MIM CP. Also, work has been done to extend OWL to work with axioms and assertions based on probability like PR-OWL[14]. Future work can explore the applicability of PR OWL to C-BML based on use cases.

10. Conclusions

C-BML continues to evolve, to better support C2-simulation interoperation in the coalition environment. Developing a semantically enriched C-BML addresses avenues of interest both in doctrinal formalization and operational efficiency. An ontology based C-BML standard can be used to capture the full expressivity of a data model while allowing users to create and implement a required subset of the ontology as a XML schema. This will help in faster development time and testing time of C-BML implementations. A semantically enriched C-BML system can be used to pre-process C-BML documents to make sure that data is not only syntactically valid but also maintain semantic integrity. In addition, a semantic C-BML can have Task/Plan specific rules that can be used to generate abductive hypotheses based on Reports. Prevalent Semantic Web standards such as RDF, RDFS and OWL are suitable to create the domain Ontology for C-BML.

References:

[1] Schade, Ulrich, Bastian Haarmann, and Michael R. Hieb. "A Grammar for Battle Management Language." In Proceedings of the 2011 IEEE/ACM 15th International Symposium on Distributed Simulation and Real Time Applications, pp. 155-159. IEEE Computer Society, 2011.

[2] Sudnikovich, W., Ritchie, A., de Champs, P., Hieb, M. and Pullen, J., "NATO Exploratory Team – 016 Integration Lessons Learned for C2IEDM and C-BML", *IEEE Spring Simulation Interoperability Workshop,* San Diego CA, USA, 2006.

[3] Carey, S., Kleiner, M., Hieb, M. and Brown, R., "Standardizing Battle Management Language – A Vital Move Towards the Army Transformation", *IEEE Fall Simulation Interoperability Workshop*, Orlando, FL, USA, 2001.

[4] Blais, C., Galvin, K. and Hieb, M., "Coalition Battle Management Language (C-BML) Study Group Report", *IEEE Fall Simulation Interoperability Workshop*, Orlando FL, USA, 2005.

[5] Blais, Curtis, Dick Brown, Sidney Chartrand, Saikou Diallo, Kevin Heffner, Stan Levine, Samuel Singapogu, Marc St-Onge, and Dan Scolaro. "Coalition Battle Management Language (C-BML) Phase 1 Information Exchange Content and Structure Specification." (2010).

[6] Heffner, Kevin., Gupton,Kevin.," Implementing a Standards Development Framework for the Coalition Battle Management Language", 18th ICCRTS 2013

[7]Pan, Feng. *Representing complex temporal phenomena for the semantic web and natural language*. ProQuest, 2007.

[8] http://www.w3.org/2005/Incubator/geo/XGR-geo-20071023/

[9] Sudnikovich, William P., J. Mark Pullen, Martin S. Kleiner, and Scott A. Carey. "Extensible battle management language as a transformation enabler."*Simulation* 80, no. 12 (2004): 669-680.

[10] Tolk, Andreas, and Curt Blais. "Taxonomies, ontologies, and battle management languages–recommendations for the coalition BML study group." In *Spring Simulation Interoperability Workshop, 05S-SIW-007, San Diego, CA*. 2005.

[11] Schade, Ulrich, Bastian Haarmann, and Michael R. Hieb. "A Grammar for Battle Management Language." In *Proceedings of the 2011 IEEE/ACM 15th International Symposium on Distributed Simulation and Real Time Applications*, pp. 155-159. IEEE Computer Society, 2011.

[12] Noessner, Jan, and Mathias Niepert. "ELOG: a probabilistic reasoner for OWL EL." In *Web Reasoning and Rule Systems*, pp. 281-286. Springer Berlin Heidelberg, 2011.

[13] Klinov, Pavel, and Bijan Parsia. "Pronto: A practical probabilistic description logic reasoner." In *Uncertainty Reasoning for the Semantic Web II*, pp. 59-79. Springer Berlin Heidelberg, 2013.

[14] Costa, Paulo C.G. (2005) Bayesian Semantics for the Semantic Web. Doctoral Dissertation. Department of Systems Engineering and Operations Research, George Mason University: Fairfax, VA, USA. p. 312.

[15] Blais, C., C. Turnitsa, and P. Gustavsson. 2006. "A Strategy for Ontology Research for the Coalition Battle Management Language (C-BML) Product Development Group." Paper 06F-SIW-003. In *Proceed-ings of the Fall Simulation Interoperability* Workshop. *Simulation Interoperability Standards Organization. Orlando FL.*

[16] Pullen, J., Douglas Corner, R. Wittman, A. Brook, O. Mevassvik, and Anders Alstad. "MSDL and C-BML Working Together for NATO MSG-085." In *Proceedings of the 2012 Spring Simulation Interoperability Workshop*. 2012.

[17] BAADER, F., CALVANESE, D., MCGUINNESS, D.L., NARDI, D., PATEL-SCHNEIDER, P.F., eds.: The Description Logic Handbook. Cambridge Univ. Press (2003)

[18] Hieb, M. R., M. Powers, J. M. Pullen, and M. Kleiner. "A geospatial battle management language (geoBML) for terrain reasoning." *11th ICCRTS* (2006).

[19] Gupton, Kevin., "On the Use of the Web Ontology Language (OWL) for C-BML Standards Development" BML Symposium 2011

[20] Bontas, E. Paslaru, Malgorzata Mochol, and Robert Tolksdorf. "Case studies on ontology reuse." *Proceedings of the IKNOW05 International Conference on Knowledge Management*. Vol. 74. 2005.

[21] Pullen, J., et al. "Technical and Operational Issues in Combining MSDL and C-BML Standards for C2-Simulation Interoperation in MSG-085." *NATO Modelling and Simulation Symposium, Stockholm, Sweden*. 2012.

[22] Heffner, Kevin, Nico Bau, and Michael Gerz. *"An Approach Using MIP Products for the Development of the Coalition Battle Management Language Standard"*. 11th ICCRTS (2013).

[23] Pullen, J., et al. "Linked Heterogeneous BML Servers in NATO MSG-085." *IEEE Fall 2013 Simulation Interoperability Workshop, Orlando, FL*. 2013.

[24] Klein, Michael, et al. "The relation between ontologies and schema-languages." *Proceedings of the Workshop on Applications of Ontologies and Problem-solving Methods*. 2000.

Opportunities for Next Generation BML: Semantic C-BML

Samuel Singapogu

C4I Center, GMU

Outline

- Introduction
- Current state of C-BML
- Need for a semantic C-BML
- Enhancements provided by a semantic layer
- Elements in a Semantic C-BML
- Reasoning through a Knowledge representation
- Future Work and Conclusions

Introduction- BML

- Battle Management Language is an unambiguous language to facilitate the command and control of forces and equipment in a military environment and to provide information for situational awareness.

- BML has a accompanying grammar- Command and Control Lexical grammar (C2LG)

- *One of the goals of BML is to provide "**Shared Semantics** between C2 and M&S via a Common Tasking Description"*

- BML is based on work initiated by the C4I Center outlined in [*Carey, S., M. Kleiner, M. Hieb, and R. Brown, "Standardizing Battle Management Language – A Vital Move Towards the Army Transformation," IEEE Fall Simulation Interoperability Workshop, September 2001*]

Introduction: C-BML

- C-BML is applying BML to a coalition context
- A standard has been approved following a SISO balloting process
  - Based on a specification provided by the C-BML product development group (*Blais, Curtis, et al; SISO Fall 2011 SIW)*
  - My work provides insight into the use of OWL in phase 2 standardization
- Phase 1 focused on formalizing syntax in terms of a XML schema
  - Vocabulary based on the JC3IEDM data model
 - Sought to provide full expressivity of the data model

Limitations in the current standard

- An XML based system built on the full expressivity of the JC3IEDM limits the speed of
  - Development/extensions
  - Integration
  - Testing
- Interoperability in Phase 1 is on the syntax level
- The need for a semantic C-BML has been suggested by Blais, Turnitsa, and Gustavsson (SISO Fall SIW 2006); my work provides:
  - A path forward in using upper level ontologies
  - A context for the use of reasoning in semantic C-BML

Introduction: Semantic Web

- The semantic web is a framework of linked web data in a shared machine-readable knowledge representation
  - Shared semantics
  - Linked data
  - Machine readable
- Based on W3C standards
- Semantic representation enables:
  - Formalization of shared semantics
  - The ability to infer knowledge
  - The use of a reasoner to check for semantic inconsistencies

The Semantic Stack
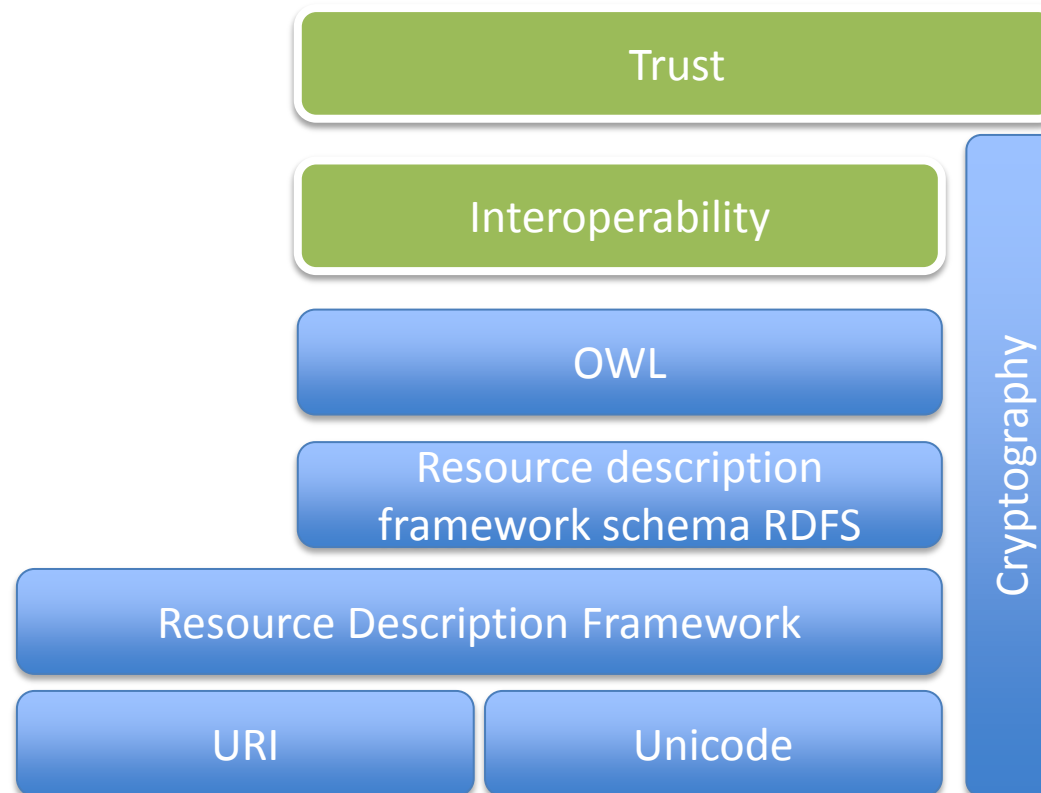
Current standards in ontology creation

- Resource Description Framework
  - Based on Uniform Resource Identifier (URI)
  - Any element can be defined (and disambiguated) using a URI
  - Knowledge is represented using a <subject> <predicate> <object> triplet

An example of a semantic representation(in C-BML context)

- Representation of a Unit using URI: http://urlNamespaceOfUnit:UnitA

- Representation of a relationship:
  http://urlNamespaceOfUnitRelationship:UnitHasAsCommander

- An RDF axiom (asserted knowledge in the ontology):

<http://urlNamespaceOfUnit:UnitA>
<http://urlNamespaceOfUnitRelationship:UnitHasAsCommander>
<http://urlNamespaceOfUnit:UnitA >

- RDFS is:
  - Flexible
  - Easily scalable

Introduction: OWL

- Web Ontology Language is a current standard to model and represent knowledge in the form of an ontology
  - The goal is to model and represent knowledge in a machine readable fashion
  - Based on RDFS, can be serialized to XML
  - Compliant to description logic, which makes it computationally decidable and has adequate logical expressivity
  - Available reasoners can be used to derive inferred knowledge

- OWL is a W3C standard (*http://www.w3.org/TR/owl-features/*);

Why does C-BML need an ontology?

- It formalizes the definition and **meaning** of common terms
- It formalizes the **doctrinal** rules for Orders and Reports
- It eases **interoperability** because of a shared vocabulary and defined meaning
- It allows performing powerful **reasoning** on operational semantics
- A model driven(ontology-driven) framework facilitates easy extensions
  - Gupton, Blais and Heffner (International Journal for Intelligent Decision Support Systems, October 2011) suggest model based data engineering as a development approach; my work lays a foundation for OWL as a central data model.

A path to creating semantic C-BML

- Evaluate relevant upper level ontologies
- Extract semantic pieces from existing Phase 1 work
  - Entities recognition (XSD elements, XSD types)
  - Taxonomy classification (subsumption relation, IS-A relationships)
 - C-BML specific relationships
  - Doctrine based axioms

Upper Level Ontologies

- General purpose ontologies that define entities in a particular domain (time, geography, ...)
- Why use upper level ontologies?
  - Reusability
  - Easier extension management
  - Easier mapping between systems that use the same upper level ontology
- Gupton, Blais and Heffner (IJIDSS, October 2011) have proposed the use of upper level ontologies; my work:
  - Identifies applicable upper level ontologies and their alignment with C-BML
  - Provides a context for reasoning on semantic C-BML

Upper level ontologies relevant to C-BML

- C-BML vocabulary is based on the 5Ws('Who', 'What', 'When', 'Where' and 'Why')
- OWL-Time *(relevant to 'When')*
  - Upper level ontology that represents time in different forms, temporal constraints and axioms
  - The core class "TemporalEntity" has two sub classes: Instant and Interval
- Geo-OWL *(relevant to 'Where')*
  - Upper level ontology to represent geometric shapes
  - The top class geometry can be of type Point, LineString, Polygon…

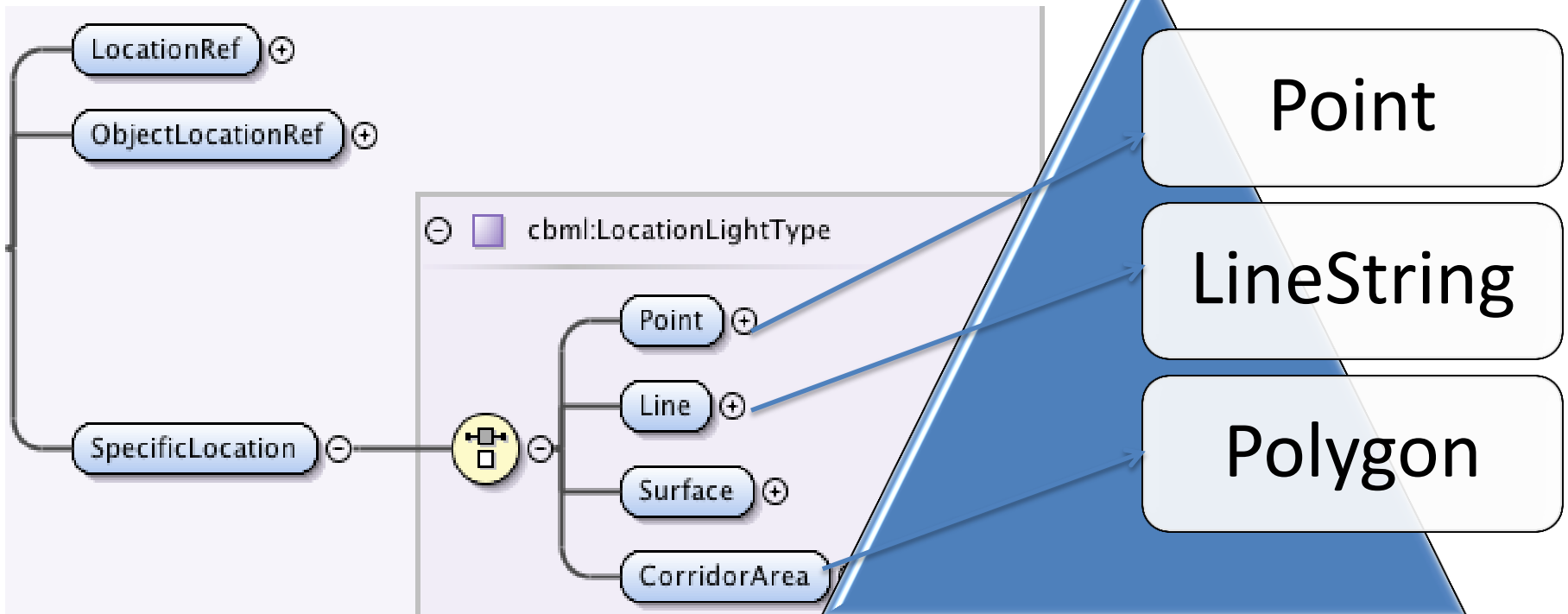# Mapping between C-BML and OWL Time
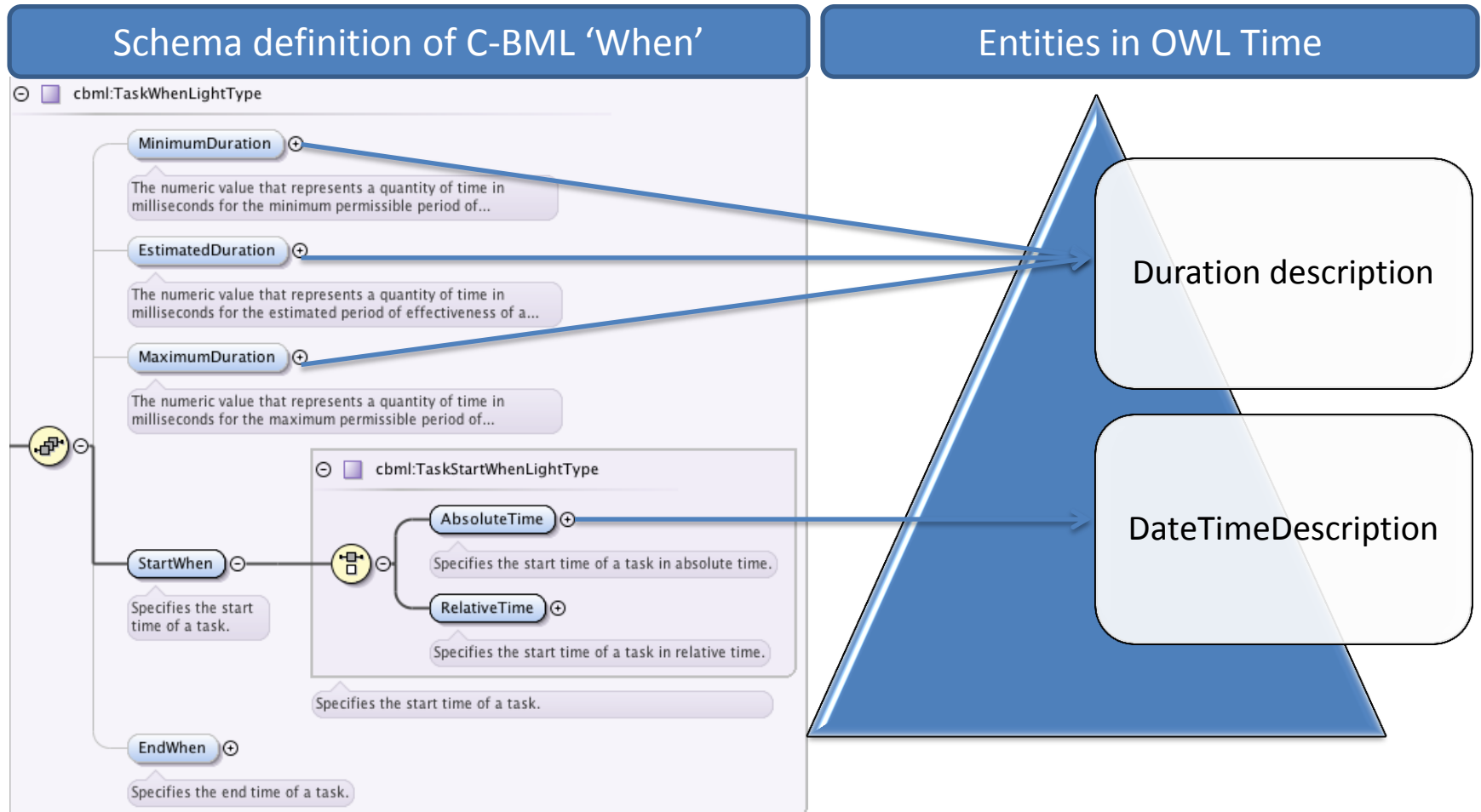
Advantages of using OWL-Time

- Rich expressiveness of time entities
  - Models both time instants and time intervals
  - Temporal constraints can be established
- Powerful reasoning over temporal concepts
  - Task1 <u>after</u> Task2 can be modeled
  - New temporal relationships can be inferred using a reasoner

Mapping between C-BML and Geo-OWL

Reasoning in semantic C-BML

- Reasoning has two main goals
  - Checking for semantic inconsistencies
  - Gaining inferred knowledge
- OWL captures knowledge in a way that existing reasoners (HermIT, Jena..etc) can automatically derive new knowledge
- OWL reasoners are based on First order predicate logic

Semantic C-BML reasoning example 1

- Checking for semantic inconsistencies
  - An example in the "reports" context: "*Executer(x), hasAsReportedTime(x,t1)∧Executer(x), hasAsReportedTime(x,t1)⟹(hasAsLocation(x,l1)=hasAsLocation(x,l2))*
  (SWRL syntax)*

  Checks to make sure that multiple reports provide consistent reported data locations of a "Executer"

- A ontology provides a formal, machine understandable way to check for semantic inconsistencies

Semantic C-BML reasoning example 2

Deriving inferred knowledge:

Example asserted axioms:

Axiom 1: *ObjectProperty (a:isAUnit domain(a:Tasker) range(a:Unit))*

Axiom 2*: ObjectProperty (a:isAsubordinateOf domain(a:Taskee) (a:Tasker))*

The knowledge represented by these Object properties are:

1: A Tasker should be a Unit (as opposed to a Equipment)

2: A Taskee is subordinate to a Tasker

Semantic C-BML reasoning example 2
continued

Example Task:

Tasker:"1060: 1st Battalion Commander"

Taskee: "1062: Company A"

Using the two axioms, we can infer the knowledge:

"'1060: 1st Battalion Commander' is a Unit who is a commanding officer to '1062: Company A'"

Semantic C-BML reasoning example 3

Axiom: ObjectProperty(a:isAfterTask domain(a:Task) range(a:Task)

Note: Object Properties are transitive

Assertion1: Task1 isAfterTask Task2,

Assertion2: Task2 isAfterTask Task3

The inferred knowledge is:

Task1 isafterTask Task3

Note: *This inference can be derived in the Ontology even if the two tasks are in separate C-BML Orders/Tasks*

Conclusions

- Semantic C-BML can help in:
  - Model driven development and ease of scalability and extension management
  - Better interoperability with shared semantics and common, formal conceptualization
  - Formalization of doctrinal rules/axioms and semantic restrictions
  - Checking for semantic inconsistencies
- OWL is an adequate language to model C-BML

Future Work

- Development of C-BML in a model driven framework
  - OWL could be used as the central semantic model
  - Alignment with the MIP information Model and Change Proposals(CP)
- Abductive reasoning as a way to hypothesize knowledge based on reported data
- Explore ways to extract semantic elements from Phase 1 specification
  - How do XML schema schema elements relate to entities?
  - What relationships can be extracted from XML schemas?