2010

Integrating the Coalition Battle Management Language (C-BML) into the Military Scenario Definition Language (MSDL)
Integrating the Coalition Battle Management Language (C-BML) into the Military Scenario Definition Language (MSDL)

Curtis Blais  
MOVES Institute  
Naval Postgraduate School  
clblais@nps.edu  

Jeff Abbott  
Systems Architect  
CAE USA Professional Services  
jeff.abbott@caemilusa.com  

Keywords:  
Simulation, combat modeling, scenario definition, agent-based simulation, MSDL, plans, orders, C-BML

ABSTRACT: The Coalition Battle Management Language (C-BML) is a common language for expressing and exchanging plans, orders, requests, and reports across command and control systems, modeling and simulation systems, and robotic systems. A Phase 1 C-BML Specification is being drafted in preparation for balloting later this year. The C-BML information exchange structure and content portion of the specification provides a means to express plans, orders, reports, and requests using the Extensible Markup Language (XML).

The Military Scenario Definition Language (MSDL) is an approved SISO standard (SISO-STD-007-2008, October 2008) that specifies an XML-based language designed to support military scenario development. The standard provides the modeling and simulation community a common mechanism for verifying and loading military scenarios, the ability to create a military scenario that can be shared between simulations and C4I devices, a way to improve scenario consistency between federated simulations, and the ability to reuse military scenarios as scenario descriptions are standardized throughout the Army, Joint, and international communities and across simulation domains; e.g., training exercise, analysis, etc. MSDL files are used to initialize data in simulations; however, the current version of the MSDL standard does not include expression of tasks for simulated forces to perform that can be preloaded for execution or scheduling when the simulation starts. Expression of tasks in MSDL requires the ability to express plans and orders, which is provided by the Phase 1 C-BML Specification.

This paper provides a brief introduction to C-BML and MSDL and then describes how the currently proposed C-BML information exchange structure and content specification can be used to provide a tasking language for plans and orders in MSDL scenario representations.

1. Introduction

DoD uses a variety of modeling and simulation systems for analysis, training, experimentation, acquisition, and mission planning and rehearsal. Often there is a requirement to represent the same operational situation in multiple systems. For example, a scenario used in a training exercise may be needed to conduct analysis of future force structures, or vice versa. Or, a scenario used for conducting certain analyses may be employed in an operational experiment evaluating new Command and Control (C2) systems or new tactics, techniques, and procedures (TTPs). Many events now use a federation of M&S systems to represent complex, multi-domain battlespace entities and dynamics. Because of differences in design of the individual federates, common aspects of the scenario have to be expressed in different ways to be understandable to the individual software. The individual M&S system (or federate) representations are not easily interchangeable, even though they often represent very similar aspects of the situation, such as force structures, initial plans and orders, weather conditions, and terrain. It is not currently possible to use a single description of a scenario to initialize common aspects of the battlespace across all federates.

Version 1.0 of the Military Scenario Definition Language (MSDL) [1] was approved by the Simulation Interoperability Standards Organization (SISO) as an international standard in October 2008. The language specifies force structures, environment, and other information for initialization of simulation systems. The standard specifies an Extensible Markup Language (XML) schema to provide a common mechanism for validating and loading military scenarios, to promote sharing of scenario files across simulation and C2 systems, and to improve scenario consistency among federated simulations.

The Coalition Battle Management Language (C-BML) is an emerging standard for expressing and exchanging
plans, orders, requests, and reports across: (1) command and control (C2) systems; (2) live, virtual and constructive modeling and simulation (M&S) systems; and (3) robotic systems participating in Coalition operations. As recommended by the C-BML Study Group final report [2], each phase of the C-BML specification development will describe:

- A data model (specifically, the C-BML SG recommended JC3IEDM as a starting point for all phases of the effort);
- An information exchange content and structure specification defining valid form and content of C-BML expressions;
- An information exchange mechanism specification enabling a common approach to implementation of applications that can process C-BML information;
- Guidelines for adoption and application of the standard that explain C-BML use and provide practical examples.

This paper provides a brief introduction to the current MSDL standard and the emerging C-BML standard, and then describes how the currently proposed C-BML information exchange content and structure specification can be used to provide a tasking language for plans and orders in MSDL scenario representations.

2. Overview of MSDL

The top-level structure of the MSDL XML schema is shown in Figure 1. Top-Level MSDL Schema Structures. The following subparagraphs provide brief overviews of the MSDL data model. It is not possible to provide a complete description of MSDL in this paper. The reader is referred to the current MSDL specification and XML schemas for a full description of the language.

2.1 Primary Scenario Constructs in MSDL

MSDL describes locale, forces, intelligence, situation, and course of action for re-use across multiple C2 and M&S systems. The MSDL Specification [1] defines a military scenario as “a specific description of the situation and course of action at a moment in time for each element in the scenario.” The scenario description largely reflects common Mission, Enemy, Terrain and weather, Troops and support available, Time available and Civil considerations (METT-TC) elements of a military situation. The purpose is to provide the M&S community with:

- A common mechanism for validating and loading military scenarios.
- The ability to create a military scenario that can be shared between simulations and C4I devices.
- A way to improve scenario consistency between federated simulations.
- The ability to reuse military scenarios as scenario descriptions are standardized throughout the Army, Joint, and international communities and across simulation domains; e.g. training exercise, analysis, etc.

Scenario elements can be individual items of equipment, such as a tank or aircraft, or aggregates of troops and equipments, such as an infantry company. The reality of the situation reflects known or established content in the scenario, such as a certain force structure being employed to conduct an operation in the simulation or terrain and weather conditions set for the execution. These descriptions are exact and not the result of interpretation by the scenario elements. Intelligence information reflects knowledge of the battlespace that an entity or force may possess at the outset of the execution, such as knowledge of enemy force positions and activities. This information may be incorrect and incomplete, but represents what is known when the execution begins (and on which simulated entities may begin making decisions and taking action). Some simulations do not start with such information, but establish battlespace awareness through simulated detections as the entities and forces begin to interact in the simulation.

The MSDL description of the scenario is expressed as an XML file conforming to an XML schema described and provided in the SISO specification. The MSDL XML schema defines one global element, the MilitaryScenario root element. All other constructs in the language are defined as global types, either complex or simple types, to maximize reuse of the definitions in creation of other XML languages. MSDL also has extensibility provisions through the use of the XML Schema any construct. This permits an MSDL XML document to contain arbitrary XML structures that may be defined by other schemas.

---

1 The mstd: prefix in element and type names in the MSDL XML schema refers to the MSDL namespace “urn:sisostds:scenario:military:data:draft:mstd:1.” Solid boxes in the figure denote required elements; dashed boxes indicate optional elements.

2 C4I: Command, Control, Communications, Computers and Intelligence
We can examine the content of an MSDL description by examining the structure of the language defined in the XML schema. The root element of the XML file is called `MilitaryScenario` and contains the following child elements (the descriptions here are illustrative, not exhaustive):

- **ScenarioID** (mandatory) – provides identification of the scenario and its purpose.
- **Options** (mandatory) – provides global parameters about the scenario and its content.
- **Environment** (optional) – describes the simulated physical environment in which the execution is to occur (e.g., area of interest, weather, time).
- **ForceSides** (mandatory) – describes the structure of the forces and sides involved in the execution.
- **Organizations** (optional) – describes the structure of the units and equipment involved in the execution.
- **Overlays** (optional) – describes the logical overlays used to group the intelligence elements/instances in the scenario. Ownership of a specific overlay is determined through the intelligence elements/instances contained in that overlay.
- **Installations** (optional) – describes the detected installations as determined by the intelligence gathering process of each force, side, or unit individually.

![Figure 1. Top-Level MSDL Schema Structures](image)
• *TacticalGraphics* (optional) – describes the tactical information as known by a particular force, side, or unit individually.

• *MOOTWGraphics* (optional) – describes the detected MOOTWGraphics\(^3\) instances as determined by the intelligence gathering process by each force, side, or unit individually.

The ScenarioID element contains metadata about the scenario, including the following information: (1) name assigned to the scenario; (2) type of object model; (3) version of the scenario file; (4) date of last modification; (5) classification level; (6) release restrictions; (7) purpose of the scenario; (8) type or class of application to which the scenario applies; (9) description; (10) any limitations on use of the scenario; (11) history of use; (12) keyword (and identification of taxonomy) characterizing the scenario; (13) identification of the organization or person who has a particular role with respect to the scenario; (14) type and identity of any reference; (15) identification of a glyph for visually representing the scenario; and (15) other data deemed relevant by the scenario author. The ScenarioID element, defined through the ModelID schema, includes the *any* compositor, which allows any XML structure from other languages to be inserted and retain validity against the MSDL schema.

The XML design of MSDL employs certain vocabulary from other XML schemas; namely: (1) ScenarioID metadata defined in the ModelID_v2006.xsd schema from the Base Object Model Specification (SISO-STD-003-2006) [3]; and (2) meteorological and battlespace domain values defined in the Joint Command, Control, and Consultation Information Exchange Data Model (JC3IEDM\(^4\)) schema JC3IEDM-3.1-Codes-20061208.xsd. The MSDL XML schema declares namespaces assigned to these external schemas and imports these schemas in support of the definition of MSDL-specific elements and attributes.

The use of namespaces is important in dealing with XML vocabularies – the namespace enables a particular term to be uniquely identified within an XML document while permitting multiple vocabularies to be combined to create more complex languages, as in the case of MSDL’s use of the ModelID and JC3IEDM vocabularies.

Particular required capabilities for MSDL have been postponed to later versions of the specification. These include: additional organizational structures; electronic order of battle information; targeting information; and data structures to hold the planned activities of the organizations and entities defined within the scenario. The latter requirement is intended to be addressed by integration with the emerging C-BML standard.

### 3. Overview of C-BML

The data model portion of the proposed standard has been specified as the JC3IEDM logical model. The information exchange content and structure portion of the proposed standard is being addressed through description of the primary concepts that can be used in C-BML expressions in XML. The following subsections provide a brief overview of the XML structures for C-BML expressions. Refer to [4] for more information about the C-BML standardization effort. Full description of the proposed XML schema is not possible within the length constraints for this paper, but will be presented in full to the community at the Spring 2010 SIW.

The principal information components of C-BML are the 5Ws: Who, What, When, Where, and Why. In the abstract, these information components are fundamental to the expression of plans, orders, requests, and reports for any doctrine of any service, nation, or organization. The following constitute a definition of the 5Ws for purposes of the C-BML standard:

- **Who**: C-BML information component identifying the battlespace object that: is directed to perform an action (plan or order); has been observed or is reporting an action (report); is requested to perform an action; provides the authority or authorization for a plan, order, request or report; is the object of an action.

- **What**: C-BML information component identifying an action to be performed (plan, order, or request) or that has been performed (report).

- **When**: C-BML information component describing the timeframe in which an action is to occur (plan, order, or request) or when an action or event has occurred (report).

- **Where**: C-BML information component providing the location of an object in the battlespace (*C-BML Who*), the location where an action is to occur (plan, order, or request), or the location where an action or event has occurred (report). The location may be a

---


4 JC3IEDM is a well-established data model maintained by the Multilateral Interoperability Programme (MIP). See [http://mip-site.org](http://mip-site.org).
complex object, such as an area or a sequence of locations.

- Why: C-BML information component describing the rationale or purpose of an action to be performed (plan, order, or request), or the desired end state of a planned action.

The 5Ws constitute a portion of the C-BML “doctrine view”: expressions of plans, orders, requests, and reports using terminology particular to a specific nation, service, or organization. This abstraction of fundamental information components in the content of doctrinal expressions of plans, orders, requests, and reports facilitates future employment of the standard by any service, nation, or organization.

Each “W” information component takes on a certain word sense in each expression of a plan, order, request, or report.\(^5\) For example, in the context of an order, one sense for “Who” is the identity of the authority giving an order (tasker), while another sense for “Who” is the identity of organization that will carry out the order (taskee). These distinctions in meaning of a “W” in a specific C-BML expression result in different semantic mappings to the underlying data model. Refer to [4] for a listing of the various usages of the basic 5W terms, resulting in a broader set of basic vocabulary terms that can be used in construction of C-BML expressions.

As with the MSDL standard, the selected formalism for specifying the C-BML information content and structure is the XML Schema language. This language provides a precise description of the information structure and content that can be used to validate XML documents containing C-BML expressions encoded in XML (i.e., to ensure the format and content of an XML document containing C-BML expressions conform to the language specification described by the XML schema). Furthermore, the use of XML facilitates widespread adoption and deployment of the C-BML and MSDL standards.

The C-BML XML representation of the 5Ws provides information elements for use in expressing portions of plans, orders, requests, and reports that can be exchanged across systems through a variety of mechanisms (to be specified as the information exchange mechanism in the C-BML specification). Implementation (by any service, nation, or organization) of C-BML applications conformant to the Phase 1 specification will require transformation of respective information elements in current expressions (e.g., textual or binary message formats), some of which may already use defined XML tag sets, into the C-BML XML structures. Legacy systems will generally require adapters to produce and consume C-BML expressions. Over time, however, as C-BML becomes widely adopted, systems will emerge that natively “speak” C-BML, directly producing and processing C-BML expressions in place of older formats. Either way, systems will obtain the benefits of a shared, common structure and content for the expression of certain information elements in plans, orders, requests, and reports.

The draft Phase 1 C-BML XML schema is partitioned into the following files (note: the numbering in the file names relates to DG versioning of the files and will be changed appropriately when the files are produced and made available for the C-BML specification comment round):

- CBML-1.01-Expressions.xsd
- CBML-1.01-Composites.xsd
- CBML-1.01-Codes.xsd

A principal challenge in the design approach is to ensure strict conformance of C-BML information components to the underlying JC3IEDM logical data model. For conformance to the JC3IEDM, the following JC3IEDM version 3.0.2 schemas are referenced from the C-BML schemas:

- JC3IEDM-3.0.2-SimpleTypes-20090514.xsd
- JC3IEDM-3.0.2-Codes-20090514.xsd

The following schema is based on standard version 3.0.2 JC3IEDM schemas but changes the namespace to the C-BML namespace for added flexibility in evolving the C-BML standard:

- CBML-JC3IEDM-3.0.2-EntityTypes-2009-0514.xsd

The C-BML schemas declare an XML namespace for C-BML-specific terms. The namespace is currently designated by the Uniform Resource Name (URN) “urn:sisostds:bml:coalition:draft:cbml:1”. Schema files declaring XML constructs that are part of the C-BML vocabulary are assigned to the C-BML namespace. References are made to entities, simple types, and codes from the JC3IEDM vocabulary (version 3.0.2) through the use of the XML Schema import statements, allowing a schema having one target namespace (i.e., C-BML in our case) to reference vocabulary from another schema having a different namespace (i.e., the JC3IEDM namespace, in our case). A schema dependency tree is shown in Figure 2.

---

\(^5\) C-BML expressions: A C-BML plan, order, request, or report.
C-BML word sense: The meaning of a C-BML information component (one of the 5 W’s) in a specific C-BML expression.
The top-level C-BML expressions structure is shown in Figure 3.

Very simply, a C-BML expression is a choice of either a C-BML order, C-BML request, or C-BML report. As a quick introduction, we drill down into the structures of these expressions and provide a brief discussion below. A full description of the C-BML schemas will be presented at the Spring 2010 SIW through the paper session and in the C-BML PDG meeting.

Additional detail for the structure of the C-BML order expression is provided in Figure 4.

The structure of each of the C-BML expressions is defined in a global type to permit other schema authors to reuse the type, but with the freedom to give their own name to the element declared to be of that type. For example, the structure shown in Figure 4 defines the OrderType complex type for reuse in other XML schemas.

The Task structure in the OrderType complex type bears strong resemblance to XML structures that have been developed in earlier and ongoing BML research, development, and experimentation. The other structures relate to Context (in the JC3IEDM sense), TaskOrganization (optional), SupportingReport (optional), and SupportingType (optional).

Additional detail for the structure of the C-BML Request expression is provided in Figure 5.

This structure is nearly identical to the structure for a C-BML Order expression, but distinguishes the usage of the C-BML Who as identifying “who” is requesting the task to be performed (RequesterWho) and “who” is requested to perform the task (RequestedWho).

Additional detail for the structure of the C-BML Report expression is provided in Figure . The type permits the selection of one or more specific report structures. A number of common report structures are provided in the Phase 1 draft XML schema; we expect other specialized report structures and content to be designed as the language becomes adopted by various users and for various purposes.

---

6 In the proposed approach, a C-BML Plan is considered to be a collection of C-BML Tasks. Note that the schema allows one or more C-BML Tasks to be specified in a single Order.
4. Alternative Approaches to Integrating C-BML Expressions into MSDL

Ultimately, the MSDL structure will be fundamentally modified in a future version to provide for introduction of C-BML tasks. As an interim measure, to support prototyping and experimentation prior to formalism in the next version of the specification, there are provisions in the current MSDL schema that allow for introduction of other XML structures and content, while retaining the ability to validate the file against the MSDL schema and allowing existing MSDL processing software to parse the files for content conforming to the current MSDL schema. In the following paragraphs, we examine these alternative approaches to encourage early experimentation with the structures.

4.1 Preserving Existing Content and File Validation

In the XML schema structure for MSDL, there is one area where literally anything can be added to the language and still obtain a valid XML document conforming to the MSDL schema. As introduced earlier, the first child element, `ScenarioID`, of the `MilitaryScenario` root element has complex structure defined from the BOM specification. Of particular interest is the final child element in the `ScenarioID` structure, defined in the XML schema as follows:

```xml
<xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
```

The `<xs:any>` declaration is called an element wildcard. This construct allows the entry of one or more elements from any namespace into this part of the structure of an XML document conforming to the MSDL schema. The “##other” value for the namespace attribute allows elements from namespaces other than the defined target namespace (in this case, the BOM namespace http://www.sisostds.org/schemas/modelID) to be included as part of the wildcard. The processContents="lax" attribute instructs the processor to attempt to validate the wildcard elements if it has access to a global XML Schema definition for them (more on this later).

Consider the following notional (and minimal) MSDL file:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2008 (http://www.altova.com)-->
<msdl:MilitaryScenario xsi:schemaLocation="urn:sisostds:scenario:military:data:draft:msdl:1"
```

The `msdl:MilitaryScenario` element represents a scenario of operations within a military domain. The `xsi:schemaLocation` attribute specifies the location of the schema definition for the MSDL specification.

---

7 The OneSAF project is evaluating the use of this construct in every complex type in MSDL.
However, the processContents attribute is actually absent in the ModelID_v2006_FINAL.xsd schema used by MSDL. This means that a processor will attempt to validate the wildcard elements and will raise a validity error if a global XML Schema definition for the wildcard elements cannot be found (same behavior as when processContents="strict"). We changed the entry to processContents="lax" in the MSDL schema to enable the example to validate. A better way, but more complex, is to create a composite schema that imports the MSDL and C-BML namespaces and schemas for the various components we want to include in the structure. In our case, the composite schema would import the MSDL MilitaryScenario schema (which, in turn, imports the other MSDL schemas) and the C-BML Expressions schema, and would then have a single element declared as being of type msdl:MilitaryScenarioType.

### 4.2 Incorporating C-BML into a Future Version of MSDL

During development of the current MSDL standard, earlier working versions of the MSDL XML schemas had a child element of the MilitaryScenario element that was intended to hold course of action information for the forces defined in the scenario. One approach to integration of C-BML into the next version of MSDL is to declare a new child element under MilitaryScenario as (assuming the default namespace is the MSDL namespace and “C_BML” is the defined prefix for the C-BML namespace):

```xml
<xs:element name="Orders" minOccurs="0">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="Order" type="C_BML:OrderType" minOccurs="1" maxOccurs="unbounded"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

The minOccurs value of “0” makes this structure optional in an MSDL scenario file, so MSDL files that conform to the current schema would still be valid under the new schema. The structure permits the inclusion of one or more C-BML OrderType structures in an MSDL Orders element to enable specification of tasks to be performed by the various units defined in the scenario.
5. Other Approaches and Issues

In lieu of balloting and approval of the C-BML standard, some projects are working with other approaches to defining tasks in MSDL. For example, the OneSAF (One Semi-Automated Forces) and Deep Green work with MSDL includes a framework for orders in an event hierarchy consisting of:

Course of Action->Phase->Event->Unit Tasking
(where the tasker owns the course of action).

The C-BML PDG and MSDL PDG and Product Support Group (PSG) should consider a number of questions and issues in upcoming meetings:

- Should MSDL specify a course of action framework that integrates C-BML? Or is that the role of C-BML?
- Reports need to integrate measures that can be used to determine the success or failure of orders to achieve a desired effect. For example, if an tasker/order specifies an arrival time, a taskee/report needs to give status on expected/anticipated arrival time. How will MSDL integrate that concept through C-BML? Should C-BML integrate those specifically in the specification (i.e., through the SupportingReport structure shown in Figures 4 and 5)?
- Unlike the C-BML design approach, MSDL is not directly mappable to JC3IEDM. We need to decide how to define the Who’s in order to integrate C-BML. Other researchers have been working on Unit and Entity Type Compositions that would accomplish that. Those compositions would enable MSDL to retain "aggregate" entity types for planning (functional needs of the mission), but map them to constituent “piece parts” for execution (the means of the mission).

We invite SISO members to raise other issues to aid the groups in refining and integrating these important standards.

6. Conclusion

Preliminary investigations into the application of MSDL are revealing broad applicability for scenario description and interchange across numerous systems (e.g., see [7,8]). MSDL has a well-defined scope of coverage that maintains coherence and ease of use to help achieve the greatest level of acceptance. The MSDL PDG is working closely with the C-BML PDG to ensure compatibility across the two standards. Early implementations will provide valuable sources of information to help examine and refine integration of the representations provided by the two specifications.

7. References

Disclaimer

The opinions expressed in this paper are those of the authors and not necessarily those of the authors’ organizations, SISO, or any of SISO’s working groups.

Author Biography

CURTIS BLAIS is a Research Assistant in the Modeling, Virtual Environments, and Simulation (MOVES) Institute at the Naval Postgraduate School, Monterey, California. Mr. Blais conducts research in a variety of areas, including design of agent-based simulation of non-traditional warfare and application of Semantic Web technologies to address effective interchange of information across M&S and C2 systems. Mr. Blais is also a Ph.D. candidate in MOVES investigating ontological frameworks for obtaining valued information at the right time in semantically-rich network-centric architectures.

JEFF B ABBOTT holds a masters degree in engineering from the University of Central Florida. Mr. Abbott has worked in the area of modeling and simulation for 22 years. Much of that time has been spent working issues of interoperability related to simulation based training. Mr. Abbott has been working standards development under SISO in support of interoperability between battle command and modeling and simulation technologies.