Joint Battle Management Language (JBML) - US Contribution to the C-BML PDG and NATO MSG-048 TA

Dr. J. Mark Pullen
Dr. Michael R. Hieb
Dr. Stan Levine
C4I Center
George Mason University
Fairfax, VA
{mpullen,mhieb,slevine}@netlab.gmu.edu

Dr. Andreas Tolk
Virginia Modeling, Analysis and Simulation Center
Old Dominion University
Norfolk, VA 23529
atolk@odu.edu

Curtis Blais
MOVES Institute
Naval Postgraduate School
Monterey, CA 93943
clblais@nps.edu

Keywords:

Coalition Battle Management Language (C-BML), C2-simulation interoperability, Web service information exchange

ABSTRACT: *The Joint Battle Management Language (JBML) is being developed as an unambiguous language for tasking and reporting. This paper summarizes significant US national contributions to the current SISO Coalition Battle Management Language (C-BML) Product Development Group activities. It focuses on application of the well-known principles of BML in the joint warfighting context to enable command and control of simulated Joint and Coalition forces. The JBML design is characterized by three layers that enable configurable solutions, not only from the information system perspective, but also from a domain-specific information exchange view. The main ideas are to assemble meaningful sentences of domain-specific information elements in an unambiguous structure that captures the commander's intent (domain services), defined in terms of meaningful objects that compose data into information elements of general application (composite services), and represented using standardized data elements that are entities of the JC3IEDM (atomic services), which provides a standard vocabulary for all three layers. The services are implemented as Web services supporting C-BML Phase 1. The domain configuration uses a schema motivated by initial work on formal grammar, intended to support C-BML Phase 2. The Web service is configured using this domain-specific knowledge, in the form of an XML Schema Definition. The data encodings are tightly connected with the Joint Command, Control and Consultation Information Exchange Data Model (JC3IEDM), although the higher levels of JBML introduce abstractions that encapsulate the complexity of the underlying data model intended to make the consistent application of JBML as an interface language straightforward. The paper focuses on the JBML layered approach and how these elements contribute to the C-BML standardization activity and its application in the NATO Modeling & Simulation Group Technical Activity 048.*

1. Introduction

The Joint Battle Management Language (JBML) activity is contributing significant efforts from the US in support of the Simulation Interoperability Standards Organization (SISO) Coalition Battle Management Language (C-BML) Product Development Group (PDG). JBML is not just "yet another BML" but is intended to become the first contribution to a growing family of standards.

While JBML is being developed to solve real requirements of the warfighter in support of exercises and experimentation, a significant aspect of the project targets the conceptual challenge to find answers that will support the C-BML standard development process. The project also is intended to support international collaboration within SISO as well as within the NATO Modeling & Simulation Group (MSG) Technical Activity on C-BML (MSG-048 TA). To meet all of these goals, JBML has been laid out as a multiphase project which is now reaching the end of its Phase 1. JBML is on the leading edge of developments in this domain; decisions made in the early phases of the effort can be expected to improve over time. This is consistent with the approach of the C-BML PDG, which is developing the C-BML standard in incremental phases also.

We begin in the next section with a short overview of the JBML project, focusing on the history and related BML activities to give a picture of the constraints under which the project works. Section 3 describes the JBML architecture, which was developed collaboratively by the participating organizations. The layered approach was designed to allow the evaluation of complementary or alternative prototype implementations and is expected to be a key factor to success of the JBML project. Section 4 describes anticipated results of initial JBML development. Section 5 shows the relevance for SISO: how JBML can contribute to the C-BML PDG and where additional research is needed.

This paper represents an updated version of the contribution "Joint Battle Management Language (JBML) - US Contribution to the C-BML PDG," presented as paper 07S-SIW-022 during the Spring Simulation Interoperability Workshop 2007.

2. The JBML Project

The goal of this project is to develop a standard Battle Management Language applicable to US Service and Joint Users as an input to the SISO C-BML process.

The need to interface Command and Control (C2) systems with Modeling and Simulation (M&S) systems has long been established. However, in the absence of DoD-wide standards for C2-to-M&S interoperability, almost every simulation has a unique C2 interface. The BML effort addresses this need by basing its semantics on the international Multinational Interoperability Program (MIP) data standards [1]. Of particular interest for JBML is that the MIP provides a common, system-independent C2 vocabulary for data interchange.

2.1 History

From its beginnings, BML was not envisioned as an exclusively technical solution but as an approach to supporting the operational needs and requirements of the warfighter. Using a vocabulary already defined in doctrinal language as well as in command and control reports therefore was essential. Carey and colleagues describe the overall process used to show the feasibility of defining an unambiguous language based on manuals capturing the doctrine of the US Army in [2]. Sponsored by the US Army's Simulation-to-C4I Interoperability Overarching Integrated Product Team (SIMCI OIPT), the first BML project started by analyzing more than 70 doctrinal manuals related to tasking and reporting, beginning with general manuals, such as the Field Manual 3-0 on Operations [3] and the Universal Joint Task List as published by the Joint Staff [4], and including the field manuals of branches of the Army, such as Field Artillery, Air Defense Artillery, Engineers, Military Police, and many more manuals down to the platoon level. This work was focused on defining an unambiguous Operational Order which led to using the *5W Structure for BML* to describe military tasks: Who is ordered to do What; Where, When, and Why. This work laid the foundation for all follow-on activities and was featured in [5].

The US Army BML effort developed a prototype for battalion operations orders that demonstrated the principles of BML in 2003. Under sponsorship of the US Defense Modeling and Simulation Office (DMSO) and the US Joint Forces Command (JFCOM), the Extensible BML (XBML) project was started as a follow-on, with two main objectives: (1) using web technology for the information exchange between the systems' interfaces to create a net-capable prototype; (2) using the Command and Control Information Exchange Data Model (C2IEDM) as a basis to represent the information to be exchanged between the systems. Both of these goals were achieved and the related work was published in [6], in addition to multiple SIW contributions. The C2IEDM is an earlier version of the MIP's current JC3IEDM data model.

The XBML prototype was used for an international experiment, driven by interest of an exploratory team of NATO's Modeling & Simulation Group (MSG-ET016). The experiment and results are described in detail in [7].

JFCOM was particularly interested in the XBML project's potential to increase interoperability between C2 systems and simulations of the US military Services. The Air Operations BML (AOBML) effort was supported by JFCOM J7 to evaluate whether the concepts of BML are applicable to air forces as well as ground forces. To this end, the XBML prototype was enriched by an interface to US Air Force command and control system Theater Battle Management Control System (TBMCS) and Air Warfare Simulation (AWSIM) systems, with the result that BML was shown to be feasible and applicable to air operations. A corollary result was recognition that the object/entity focus of C2IEDM is different from the activity focus of air warfare (e.g. the action of a sortie, which is the main point of interest for air operations). While the first phase of AOBML focused on integrating the systems using Web technology, a second phase was conducted focusing more on the identification of information exchange objects making up the AOBML The work is described in [8].

2.2 Related BML Research Activities

As stated above, JBML was envisioned as the first in a family of BML efforts that share a common core, but function individually in their own domains. The work on Ground Forces BML and Air Operations BML is described above. Recently added activities are the work of the Naval Postgraduate School (NPS) on a Navy BML under the JBML project [9] and work supported by Army

Topographic Engineering Center (TEC) on geoBML, in which BML concepts are leveraged for terrain reasoning [10, 11].

Table 1 summarizes recent and current activities focus on BML, including NATO activities related to the MSG Exploratory Team ET-016 [7] and the NATO PATHFINDER Integration Environment MSG-027 experiments [14], which is explained further in section 2.3. It is noteworthy that C-BML, as a standardization activity, is not resourced to produce an implementation, whereas the JBML project will deliver both a specification and a reference implementation that can be used to evaluate that specification.

2.3 Other Related Research Activities

The main objective of BML is to define an unambiguous language for tasking and reporting. The infrastructure is web-based. These ideas are common to a series of related activities that either directly use BML results or that have the potential to contribute to BML efforts.

In Sweden, work is being conducted to adapt military command and control results for crisis management within coalition forces. To this end, the feasibility of a Coalition Crisis Management Language (C-CML) is being evaluated. Ideas and initial results have been published in [12] and have been discussed in Europe as well as in the Asian-Pacific region.

As mentioned above, NATO MSG is interested in these and related activities, such as using C2IEDM to enable Command and Control coupling with M&S is recognized. In addition to the ET-016 activity that led to the establishment of the technical activity MSG-048, other groups are looking into the use of C-BML ideas as well. The NATO PATHFINDER project used the atomic layer of enabling Web services as described in [13] to conduct a successful trans-Atlantic experiment with command and control systems and simulation systems from several nations exchanging information based on C-BML ideas. The experiment is described in [14].

The work on BML has several views and contributing techniques. The need to align these various contributions has been documented in [1]. Current work focuses on the definition of a formal syntax and grammar to build the foundation of a BML language and grammatical approaches that are used to build higher order information exchange constructs out of the low-level information exchange objects.

Tolk's work in defining information exchange objects has focused on the definition of atomic information objects used as minimal building blocks for BML and their formal composition and aggregation into higher BML elements [15]. These ideas were applied in the second spiral of the Joint Rapid Scenario Generation (JRSG) capability (formerly known as JRD3C) within the Joint Event Data Initialization Services (JEDIS), which is described in [16]. It should be pointed out that the approach of JEDIS is engineering method driven in that the information to be exchanged is identified and mapped by engineers. The result is used to configure the system-specific Web service access layers. The JEDIS services themselves can be standardized and are rooted in the JC3IEDM data philosophy. Both MSDL and JEDIS target the domain of scenario generation.

Schade and Hieb have developed a lexical grammar to formalize Command and Control communications (the C2 Lexical Grammar) that can be applied to BML. Their initial results [17, 18, 25] have been utilized to structure the Phase I JBML schema (see below). Their focus is to utilize an existing computational linguistics grammar, the Lexical Functional Grammar [19] to construct a formal BML. Additional research by Tolk and Diallo on a grammar-based approach to information representation constraints is described in [20, 21]. Their focus is to define the information exchange capability of potentially participating systems.

Table 1. BML Activities

	Specification	Ground	Air	Naval	Implementation	Software Services	International
C-BML	X	X	X	X		X	X
ET-016		X			X		X
MSG-027		X			X	X	X
JBML	X	X	X	X	X	X	
geoBML	X	X			X		
XBML		X			X	X	X
Army BML		X			X		
AOBML			X		X		
MIP/JC3IEDM	X	X	X	X			X

Finally, the companion PDG Military Scenario Definition Language (MSDL) provides a complementary standard for initializing simulations. While C-BML focuses on the information exchange for tasking and reporting during execution, MSDL focuses on the initialization of systems on a broader basis. The close relationship between MSDL and C-BML is documented in [22]. Any BML implementation requires initialization of all participants to equivalent states, including in particular a description of types for the information elements to be exchanged. While MSDL aims at rigorous definition of minimal structures and enumeration of attributes, later phases of C-BML are planned to focus on composability and the configuration of information exchange elements based on grammar and ontology formalisms.

2.4 The Layered Services of JBML

This section provides a description of the Web services implemented as open source Java software in the JBML project. The intention is to provide a reference implementation that can serve as basic infrastructure for the project, and to offer this to the C-BML standards effort. The implementation is based on Web service networking standards [6, 23].

Figure 1 provides an overview of the JBML Web service Architecture. The layers will be described in detail in the following subsections.

- The BML Domain Configured Service (DCS) represents the domain-specific language in form of a grammar-based schema that is utilized by implementing Web services.
- The schema defines the DCS in terms of the BML Base Services (BBS) which represent the information element groups that specify information objects of interest such as the 5Ws (who, what where, when, why) and other constructs of interest.
- The lowest layer represents the information exchange of information elements. This layer is normally hidden for the user. In JBML, these are BML Common Data Access Services (CDAS).

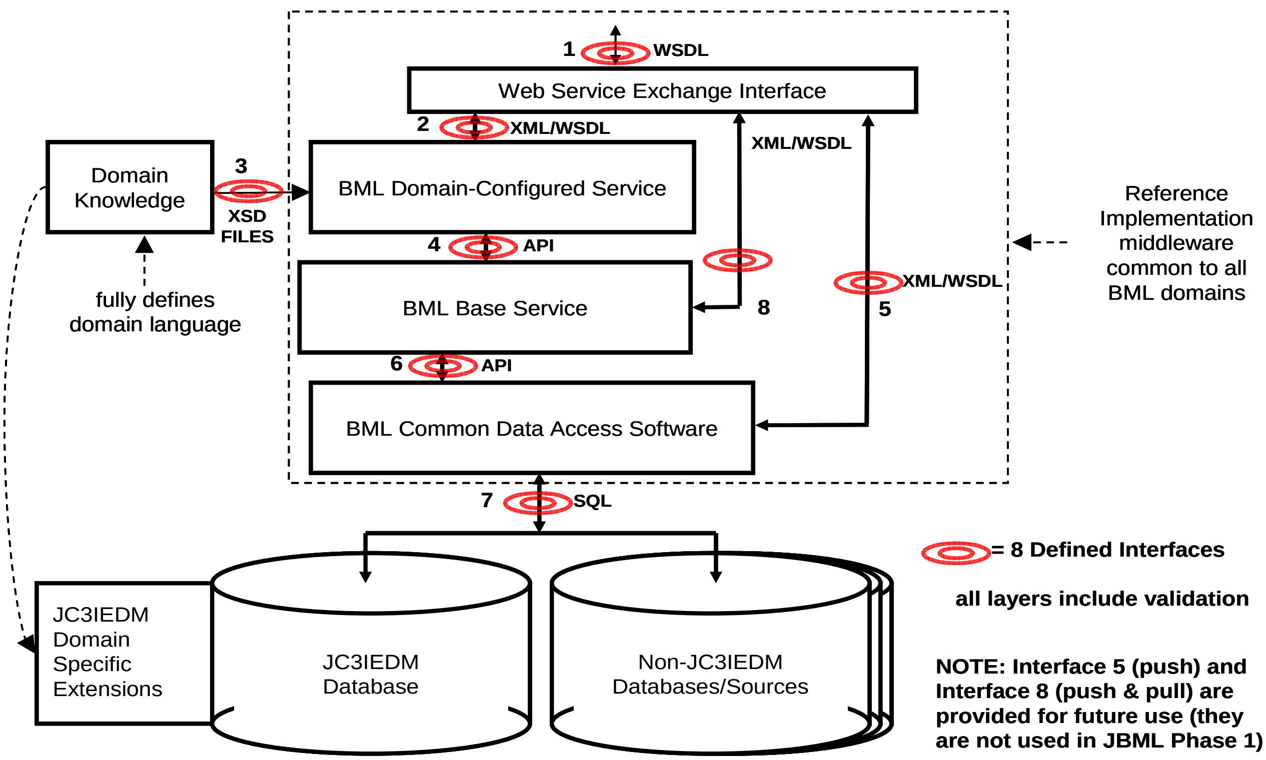

Figure 1: JBML Architecture Overview

BML Domain Configured Service

The DCS layer implements BML in a domain context. In the case of an operations order, the transaction at this layer specifies all information about a given task (e.g., who, what, when, where, and why). For a position report, the transaction at this layer will include all information about the updated location (e.g., who, where, when-valid, precision, etc.).

The DCS is implemented in the Document-Literal mode by a generic Web service that is configured by an XML schema. Schade and Hieb describe a formal grammar that can be used to represent all orders that are possible in XBML [17, 18]. We have defined an XML format that can be used by every BML order, based on the tags given in Table 2. The grammar is based upon the task (the "what") and a syntax is given that describes the task in terms of the military organization that gives the task (the "tasker-who"), the military organization that received the task (the "taskee-who"), a friendly or hostile military organization that is affected by the task (the "affected-who"), specific temporal terms denoting the start and end of the task, as well as geospatial terms, the reason why the task is performed, a label for a specific task and additional terms that may be added to completely characterize a task ("modifier").

Table 2: BML Primitives proposed in [17, 18]

`<command> (verb)`	`<tasker-who>`
`<taskee-who>`	`<affected-who>`
`<what> (action)`	`<where>`
`<start-when>`	`<end-when>`
`<why>`	`<label>`
`<modifier>`	

The Domain Knowledge Schema (DKS) for the DCS represents, for each distinct BML order, the grammar tags to be used, the BML Base Service transactions that will take place when that order is received, and the validation conditions to be applied. The DCS has a configuration file interface (3) for the DKS. Our standard proposal at this layer will define and explain all possible options for the DCS schema.

The DCS higher level interface (2) is defined using a Web service Description Language (WSDL) and is XML/SOAP based. The lower level interface (4) uses the API of BBS described below.

BML Base Service (BBS)

The BBS provides composite BML elements such as Who, What, When, Where, and Why. These are composite in the sense that they implement a composition of multiple JC3IEDM tables. Other BBS elements may be introduced for new and existing BML domains as required. The BBS accesses all of the database tables relating to the composite element through the software that implements the Common Data Access Services (CDAS) described below. Our JBML specification at this layer will identify the information objects exposed by the database tables to be updated for each BML information element (who, what, etc.) and the validation conditions to be applied. The BBS lower level interface (6) exercises the CDAS API.

The close relationship of BBS and the primitives in Table 2 is intentional; JBML uses these primitives as composites. One school of thought, associated with the ideas proposed in [15, 20], would carry this approach further; in [21] it is proposed that a new capability is required to establish independence of the primitives and unambiguous relations between them, a topic of current research which remains to be evaluated by implementation.

The BBS services are not accessed by the user of JBML, who instead uses the DCS. However, in order to support continued research in expanded BML, the JBML software has an option to expose the BBS as a Web service.

BML Common Data Access Service

The main objective of the CDAS is to provide a mechanism for the BBS to both read and update the database tables directly. For testing and debugging purposes, the CDAS supports inspection of every database table used in any domain of BML in order to support understanding of system behavior during development. Changes to the database do not overwrite the previous values but instead archive them and provide new valid values.

Within the current implementation of JBML, there are two higher level interfaces to the CDAS. One is an internal interface (6), defined as a software API. This interface is active in both directions (write and read). The second (5) is defined using a WSDL and XML/SOAP based. For JBML use, this interface is be configured for one-way (pull only) access, to be used for inspecting (reading) database tables. However, the CDAS software also offers the option of exposing a two-way interface so that the JC3IEDM representation of the data can be exchanged with systems capable of using this interface. This interface will be included in JBML's proposed C-BML specification, which will define the JC3IEDM entities used and a standard XML format to access them.

The CDAS lower level interface (7) provides an SQL based capability to access database tables representing the JC3IEDM entities.

The role of the JC3IEDM in the C-BML specification is a matter of some debate at present. It is clear that using the JC3IEDM adds significant value as the basis for the vocabulary associated with the grammar implemented in the DCS, since the MIP has invested a very great effort in identifying the terminology of command and control. Beyond this, one school of thought is to define a standard JC3IEDM interface into C-BML-based systems, so as to enable interoperation with other systems that implement the JC3IEDM. Another point of view is for future phases of the C-BML standard to omit the JC3IEDM and focus only on an unambiguous, grammar-based information exchange at higher layers. In order to facilitate exploring the alternatives, we have concluded that the best idea is to create specifications for all three layers in such a way that they can work together to provide a functioning BML: a grammar-based upper layer, a transaction-based middle layer, and a lower layer indicating the specific mapping from the higher-level representation to JC3IEDM entities and attributes. However, we believe the standard should not mandate use of all three layers, but rather allow the system designer to choose the layer(s) at which to comply.

The cascading definition of DCS, BBS, and CDAS does not imply that the implementation is done using cascading Web services as well. As described in [15], the recommended best practice is to implement standardized services as efficiently as possible. That means that once a DCS is identified the access to the composite layer, whether driven by persistent data requirements as in the current JBML or transient data driven as the C-BML enabling Web service approach described in [14], can be configured directly utilizing the APIs shown in Figure 1. The use of objects representing the agreed standard on lower levels, for example using Java objects representing tables as described in [8], ensure consistency in case lower layer standards are modified. As long as the interface standard is used at the Domain level for access and the information object exchange is satisfied on the lower level, the implementation details are irrelevant to the standard.

2.5 Differences from Earlier Work

There are two significant differences between the JBML implementations and the concepts described in [13] and [15].

First, the JBML approach is a combination of top-down and bottom-up driven, using high-level, user-driven concepts to define the elements derived from a rigorous grammar based on the ideas captured in [17, 18] as well as the layered Web services architecture. Earlier implementations, including JEDIS, are designed bottom-up and compose information objects describing the minimal information exchange request entities (the philosophically inclined might think of these as the greatest common divisors of entropic elements). As such, JBML has the potential to be more flexible than earlier solutions, as long as the supported grammar solutions are aligned with compatible views of Doctrine, Representation, and Protocols as captured in the BML Triangle [20]. Research into merging the approaches featured in [17, 18, 20, 21] holds the possibility to support improvements in future phases of C-BML.

Second, in the current JBML implementation the BML CDAS has been streamlined to support highly efficient data access. JEDIS and related C-BML work conducted for PATHFINDER use database-independent information exchange object definitions, which also access a database in the implementation. Direct use of the same interface for immediate service-to-service communication without the use of persistent data storage is described in [22]. While the general implementation is very flexible, the JBML implementation is more efficient with the chosen JC3IEDM database, which was specified for use in C-BML Phase 1 [1]. In other words, the current version implements a persistent data object exchange as the mode of choice; data object transfer directly between C2 and simulation system are not supported in this version; to support them would require replacing the CDAS layer.

The persistent data objects supported by JBML can be seen as state of the art, for multiple reasons.

- User can inspect how the information is passed and which information is passed when and by whom.
- Databases ensure consistent information exchange based on its business rules avoiding inconsistencies resulting from bad data.
- Databases allow easier debugging.
- Databases allow easy visualization of the contained information, using standard tools.
- Systems using this database (here the JC3IEDM) can directly utilize alternative information exchange means to utilize the data, such as replication mechanisms, message generators, etc.
- Middleware using a database allows the overall system to function even though the C2 and simulation systems are not all connected continuously. This is extremely useful for development, where some systems may not be in continuous operation. It also provides a more robust system in operation, since it avoids the situation

where any C2, simulation, or network outage can disable the entire system.

However, in the future both modes of exchange (persistent and transient objects) may be needed. This issue will need to be dealt with in the C-BML standard.

2.6 Example

This subsection gives some abridged examples of services to help the interested reader work through the concepts. As the implementation is based on open source ideas, more examples can be requested from the authors of this paper. It can be seen that, in addition to creating a BML namespace, we are using the namespace developed by the MSDL product development group for location and task organization information, as a first step toward developing compatible C-BML and MSDL standards.

BML-DCS (Domain Layer)

An extract of the DCS order input transaction schema is shown below. The example is truncated at end of GroundOrder, which would be followed by AirOrder, etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  xmlns="http://netlab.gmu.edu/JBML/BML"
  xmlns:msdl="http://netlab.gmu.edu/JBML/MSDL"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://netlab.gmu.edu/JBML/BML"/>
  <xsd:import
    namespace="http://netlab.gmu.edu/JBML/MSDL"
    schemaLocation="msdlTypesV0.6.xsd"/>
    <xsd:include
      schemaLocation="AirTypesV1.1.xsd"/>
    <xsd:include
      schemaLocation="GroundTypesV1.1.xsd"/>
    <xsd:include
      schemaLocation="MaritimeTypesV1.1.xsd"/>
    <xsd:annotation>
      <xsd:documentation xml:lang="en">
        JBML Schema v1.1
      </xsd:documentation>
  </xsd:annotation>
  <xsd:complexType name="OrderIdentificationType">
    <xsd:sequence>
      <xsd:element name="TaskeeWho" type="WhoType"/>
      <xsd:element name="TaskerWho" type="WhoType"
                   minOccurs="0"/>
     <xsd:choice>
       <xsd:element name="OrderIssuedWhen"
                    type="WhenType" minOccurs="0"/>
       <xsd:element name="OrderID" type="xsd:string"
                    minOccurs="0"/>
     </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="OrderType">
    <xsd:sequence>
     <xsd:element name="CommandersIntent"
                  type="FreeTextType" minOccurs="0"/>
     <xsd:element name="Command" type="CommandType"
                  maxOccurs="unbounded"/>
     <xsd:element name="OrderIssuedWhen"
                  type="WhenType"/>
     <xsd:element name="OrderID"
                  type="OrderIDType"/>
     <xsd:element name="TaskerWho" type="WhoType"/>
     <xsd:element name="TaskOrganization"
                  type="msdl:TaskOrgType"
                  minOccurs="0"/>
     <xsd:element name="ControlMeasures"
                  type="MultipleControlMeasuresType"
                  minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
 <xsd:complexType name="CommandType">
   <xsd:choice>
     <xsd:element name="GroundCommand"
        type="GroundCommandType" minOccurs="0"
        maxOccurs="unbounded"/>
     <xsd:element name="AirCommand"
        type="AirCommandType" minOccurs="0"
        maxOccurs="unbounded"/>
     <xsd:element name="MaritimeCommand"
        type="MaritimeCommandType" minOccurs="0"
        maxOccurs="unbounded"/>
   </xsd:choice>
 </xsd:complexType>
 <xsd:simpleType name="OrderIDType">
   <xsd:restriction base="xsd:string">
     <xsd:pattern value="[a-zA-Z0-9_\-]*"/>
   </xsd:restriction>
 </xsd:simpleType>
 <xsd:complexType
    name="MultipleControlMeasuresType">
    <xsd:sequence>
      <xsd:element name="ControlMeasure"
                   type="WhereType"
                   maxOccurs="unbounded"/>
    </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="GroundCommandType">
   <xsd:sequence>
     <xsd:element name="TaskeeWho" type="WhoType"/>
     <xsd:element name="What"
                  type="GroundBMLWhatType"/>
     <xsd:element name="Where"
                  type="WhereType"/>
     <xsd:element name="StartWhen"
                  type="WhenType"/>
     <xsd:element name="EndWhen"
                  type="WhenType" minOccurs="0"/>
     <xsd:element name="AffectedWho"
                  type="WhoType" minOccurs="0"/>
     <xsd:element name="Why" type="GroundWhyType"
                  minOccurs="0"/>
     <xsd:element name="Label" type="LabelType"/>
   </xsd:sequence>
 </xsd:complexType>
...
```

BBS (Composite Layer)

An example of a BBS GroundCommand push input method invocation associated with the above is:

```
   BBSGroundCommand_push(
// ID of TaskerWho 104 TR           “1029”,
// OrderIssuedWhen                  “00200ZJUL2006”,
// ID of TaskeeWho UIE9 FA          “1017”,
// What                              “ADVANCE”,
// Where                              ComplexWhere,
// StartWhen                        “020700ZJUL2006”,
// EndWhen                          “020900ZJUL2006”,
// Why-WhyAffected                  “BREAKUP”,
// CommandLabel                     “INIT_ADVANCE”,
// AffectedWhoID (null in this case)“”
// Order Domain                     “GROUND”);

Note: ComplexWhere is of WhereType (It is a class
describing the latitude, longitude, ElevationAGL etc…
of an object)
```

The resulting composite input transactions update a total up to of 25 tables (the exact number depends on whether all optional data is present), based on the mapping from the BML objects to the corresponding JC3IEDM tables, as indicated in Figure 2 (not all tables are shown).

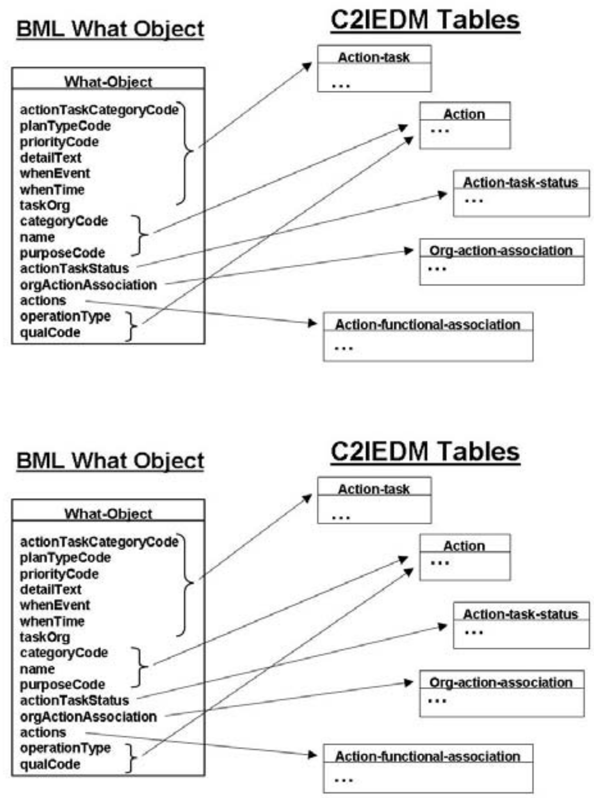

Figure 2: <what> partial object C2IEDM/JC3IEDM mapping

CDAS Web service example

An example of the BML CDAS transaction API invocation is shown below.

```
String tableName = “act”;
String columnName = “act_id”;
String keyValue = “2265”;
String result =
  getTable(tableName,columnName,keyValue);
```

This example retrieves the action table row associated with the previous example. There also exists a Java method *updateTable* that is used by the BBS to update the database.

3. Anticipated Results

The JBML project has a great legacy from previous work in C2-Simulation interfaces. While earlier work has been very ad-hoc in nature, resulting in “stovepipe” interfaces, recently the US DoD has embraced the concept of *standards-based interoperability*. This will allow systems of systems (C2 systems, simulation systems, and many other C4I systems) to be assembled without requiring custom engineering of each system-to-system interface. In the JBML project we are moving toward that goal, by developing a language standard that rests on commercial standards as described above (XML, SOAP, Web services, etc.). Moreover, we recognize that it is incumbent on us to demonstrate effective interoperation in a Joint Operational context in order to validate our results. We envision that the body of specification detail for BML will grow rapidly, given a rational structure and the expectation that any investment in compliance with the SISO standard will enable future reuse.

One of the next steps will be to develop JBML so that it will represent reports as well as orders for common military messages. Shade and Hieb have extended their C2 Lexical Grammar to address reports [25].

We are focusing on the US Joint Forces Command (JFCOM) as the initial user of JBML. JFCOM possesses both a high expertise in the various military operations supported by JBML along with demanding missions in joint training and experimentation that require specific capabilities. In particular, they anticipate that their new Political, Military, Economic, Social, Information and Infrastructure (PMESII) simulation [24] will contain multiple interfaces that use BML. Between PMESII and continued ongoing training exercises, there will be many opportunities to stress (and thereby improve) BML interfaces in the next few years. We anticipate continuing to broaden the scope of BML-based system interoperability, while also increasing the richness of the BML interface.

Another anticipated result is alignment of C-BML with MSDL, as already mandated by the Standardization Activity Committee (SAC) of SISO, and JEDIS, as requested by JFCOM users. The main advantage of MSDL is that a standardization effort is already going on in alignment with C-BML activities. However, MSDL is not yet completely aligned with JC3IEDM structures. The main advantage of JEDIS is that it is based on the JC3IEDM, but it is neither a standard nor exposed so far to the SISO community. First reports, such as [16], have begun to address this shortcoming.

4. Contributions to C-BML and Summary

The immediate contribution of the JBML project to C-BML is the service architecture described above, which will provide a regular and extensible framework upon which a powerful, flexible and growing family of standards can be created. This mature approach is the antithesis of the ad-hoc, cobbled-together interfacing that has been the norm in "stove-piped" systems of systems.

In addition, SISO has provided an ideal opportunity to make BML more effective by pairing the C-BML and MSDL standards efforts, with the mandate that the resulting standards are fully compatible and thus form a cornerstone of a growing system-level standards base. As a result, we believe the information sharing capabilities of the US and its coalition partners will continue to become more effective leading to greater cooperation and collaboration.

Acknowledgments

The authors of this paper are only a fraction of the JBML team standing behind this effort. All team members deserve recognition. The following individuals deserve particular recignition: Dr. Harry Keeling, HU, Testbed Lead; Mr. David Perme, Gestalt, Air Lead; Mr. Michael Powers, TEC, Program Manager; Mr. John Roberts, ACS, Ground Lead; Ms. Shea Smith, JATTL, JFCOM Coordinator; and Mr. John Kearley, DRC, Scenario Lead, and Chaitanya Krishna Makineni, GMU, Web service developer.

5. References

[1] Simulation Interoperability Standards Organization, SISO-REF-016-2006: *Coalition-Battle Management Language (C-BML)* Study Group Final Report, 2006

[2] 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

[3] US Department of Defense, Headquarters Department of the Army, *Field Manual N. FM 3-0: Operations*, Washington, DC, 2001

[4] US Department of Defense, Chairman of the Joint Chiefs of Staff *CJCSM 3500.04B Universal Joint Task List (UJTL)*, Washington, DC, 1999

[5] Sudnikovich, W., J. Pullen, M. Kleiner, and S. Carey, "Extensible Battle Management Language as a Transformation Enabler," *SIMULATION, Vol. 80*, pp. 669-680, 2004

[6] Tolk, A. and J. Pullen, "Using Web services and Data Mediation/Storage Services to Enable Command and Control to Simulation Interoperability," *Proceedings of the 9th IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2005)*, pp. 27-34, Montreal, Canada, October 2005

[7] Galvin, K., W. Sudnikovich, P. deChamps, M. Hieb, J. Pullen, and L. Khimeche, "Delivering C2 to M&S Interoperability for NATO - Demonstrating Coalition Battle Management Language (C-BML) and the Way Ahead," IEEE Fall Simulation Interoperability Workshop, September 2006

[8] Perme, D., M. Hieb, J. Pullen, W. Sudnikovich, and A. Tolk, "Integrating Air and Ground Operations Within a Common Battle Management Language," IEEE Spring Simulation Interoperability Workshop, April 2005

[9] Blais, C. and J. Jensen, "A Maritime Component for the Joint Battle Management Language," IEEE Spring Simulation Interoperability Workshop, March 2007

[10] Powers M., M. Hieb, J. Pullen, and M. Kleiner, "A Geospatial Battle Management Language (GeoBML) for Terrain Reasoning," *Proceedings 11th Command and Control Research and technology Symposium*, CCRP Press, 2006

[11] Hieb, M., S. Mackay, M. Powers, H. Yu, M. Kleiner, and J. Pullen, "Geospatial Challenges in a Net Centric Environment: Actionable Information Technology, Design and Implementation," Paper 6578-43, SPIE Defense and Security Symposium, Defense Transformation and Net Centric Systems Conference, Orlando, 2007

[12] Gustavsson, P., J. Wemmergård, J. Garcia, and M. Norstedt Larsson, "Expanding the Management Language Smorgordsbord towards Standardization of Coalition-Crisis Management Language C-CML," IEEE Spring Simulation Interoperability Workshop, 2006

[13] Tolk, A., S. Diallo, K. Dupigny, N. Sun, and C. Turnitsa, "A Layered Web services Architecture to Adapt Legacy Systems to the Command & Control Information Exchange Data Model (C2IEDM)," IEEE European Simulation Interoperability Workshop, 2005

[14] Tolk A. and J. Boulet, "Lessons Learned on NATO Experiments on C2/M&S Interoperability," IEEE Spring Simulation Interoperability Workshop, March 2007

[15] Tolk, A., S. Diallo, C. Turnitsa, and L. Winters, "Composable M&S Web services for Net-centric Applications," *Journal of Defense Modeling and Simulation 3* (1) 27-44, 2006

[16] Perme D, H. Tran, B. Tedesco, C. Pandolfo, S. Diallo, and A. Tolk, "Joint Event Data Initialization Services (JEDIS) – Implementing a Service Oriented Architecture for Initialization," IEEE Spring Simulation Interoperability Workshop, March 2007

[17] Schade, U. and Hieb, M.R., "Formalizing Battle Management Language: A Grammar for Specifying Orders," IEEE Spring Simulation Interoperability Workshop, April 2006

[18] Schade, U. and M. Hieb, "Development of Formal Grammars to Support Coalition Command and Control: A Battle Management Language for Orders, Requests, and Reports,"s *Proceedings of the 11th International Command and Control Research and Technology Symposium,* CCRP Press, 2006

[19] Bresnan, J., *Lexical-Functional Syntax*, Blackwell, Malden, MA, 2001

[20] Tolk, A., S. Diallo, and C. Turnitsa, Merging Protocols, Grammar, Representation, and Ontological Approaches in Support of C-BML," IEEE Fall Simulation Interoperability Workshop, September 2006

[21] Diallo, S. and A. Tolk, "Adaptive Generative Grammar for JC3IEDM Web Services," IEEE Spring Simulation Interoperability Workshop, March 2007

[22] Diallo, S., W. Civinskas, and A. Tolk, A., "An Application Extension for the Military Scenario Description Language," IEEE Fall Simulation Interoperability Workshop, September 2006

[23] Morse, K., R. Brunton, J. Pullen, P. McAndrews, A. Tolk, and J. Muguira: "An Architecture for Web Services Based Interest Management in Real Time Distributed Simulation," *Proceedings of the 8th IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2004)*, pp. 108-05, Budapest, Hungary, October 2004

[24] Snyder, D. and A. Tolk, "Exchanging PMESII Data to Support the Effects-Based Approach," *Proceedings of the Command and Control Research and Technology Symposium (CCRTS)*, San Diego, CCRP Press, 2006

[25] Schade, U. and M. Hieb, "Battle Management Language: A Grammar for Specifying Reports," IEEE Spring Simulation Interoperability Workshop, March 2007

Author Biographies

CURTIS BLAIS is a Research Associate in the Naval Postgraduate School Modeling, Virtual Environments, and Simulation (MOVES) Institute. Mr. Blais is a Ph.D. candidate in MOVES conducting research into the application of current and emerging Semantic Web technologies to obtain valued information for military decision-makers in net-centric architectures and to improve interoperability across C2 and M&S systems.

MICHAEL R. HIEB is Research Associate Professor with the Center of Excellence in C4I at George Mason University. He was the Co-Chair of the SISO CBML Study Group and also was on the team that developed the initial BML concept for the US Army while with Alion Science and Technology. He was the Technical Director of the MRCI interface project while at SAIC. He received his PhD in Information Technology at George Mason University in 1996, developing an instructable Modular Semi-Automated Forces agent. He has published over 70 papers in the areas of BML, Simulation Interoperability with Command and Control Systems, and Multistrategy Learning.

STAN LEVINE is Research Professor with the Center of Excellence in C4I at George Mason University. He also serves as a senior consultant to several Army organizations in the areas of information system technologies, architectures, System of Systems acquisition, and interoperability. He has over 33 years experience in systems acquisition. He holds a BSEE and a MS degree in Physics from Monmouth University, and a PhD in Engineering Management from Madison University. He is a recipient of the Army's three highest Civilian Service Awards. He is also a member of the Federal 100 top executives who had the greatest impact on the government information systems community.

J. MARK PULLEN is Professor of Computer Science at George Mason University, where he serves as Director of the C4I Center and also heads the Center's Networking and Simulation Laboratory. He holds BSEE and MSEE degrees from West Virginia University, and the Doctor of Science in Computer Science from the George Washington University. He is a Fellow of the IEEE, Fellow of the ACM, and licensed Professional Engineer. Dr. Pullen teaches courses in computer networking and has active research in networking for distributed virtual simulation and networked multimedia tools for distance education. He has served as Principal Investigator of the XBML and JBML projects.

ANDREAS TOLK is Associate Professor in the Faculty for Modeling, Simulation, and Visualization at the Engineering Management Department of the College of Engineering and Technology at Old Dominion University (ODU) of Norfolk, Virginia. He has over 16 years of international experience in the field of Applied Military Operations Research and Modeling and Simulation of and for Command and Control Systems. He is affiliated with the Virginia Modeling Analysis & Simulation Center (VMASC). His domain of expertise is the integration of M&S functionality into real world applications based on open standards. He received a Ph.D. and an M.S. in Computer Science from the University of the Federal Armed Forces in Munich, Germany.