13th ICCRTS: C2 for Complex Endeavors

Using Web Service-Based Command and Control to Support Coalition Collaboration in C2 and Simulation

Topic 9: Collaborative Technologies for Network-Centric Operations
Topic 7: Network-Centric Experimentation and Analysis
Topic 11: Multinational Endeavors

J. Mark Pullen, Stanley Levine and Michael Hieb

Point of Contact: J. Mark Pullen

George Mason University C^4I Center

MS4B5, 4400 University Drive, Fairfax VA 22030

+1.703.993.1538

mpullen@gmu.edu

Using Web Service-Based Command and Control to Support Coalition Collaboration

Abstract

The NATO Modeling and Simulation Group Technical Activity 48 (MSG-048) was chartered to evaluate a Command and Control (C2) Language, Coalition Battle Management Language, for Multinational and NATO C2 collaboration supported by modeling and simulation tools. To achieve this, MSG-048 is using an emerging open technical standard based on the US Joint Battle Management Language (JBML) prototype Web services, which were enhanced to meet coalition requirements. An initial demonstration in December 2007 consisted of three different operational national C2 systems interoperating with three different national simulations, supported by the JBML Web services and a C2 Grammar Graphical User Interface. In all, eight software systems from five nations successfully interoperated, showing a high likelihood that the approach can be expanded to support a wide range of coalition collaboration. This capability was achieved in only six months, helped by availability of an Internet Reference Implementation that all parties could use to test from their home laboratories, and a high level of cooperation among technical personnel and military subject matter experts from all participating nations. This paper provides a description of the Web service-based language used, the architecture and components of the overall architecture (with focus on the XML-based language schema), and the use of Web services to support agile and flexible operations.

Keywords: Coalition Operations, Command and Control, Simulation, Web Services, C2 Grammar

1. Introduction

Military simulations are used to model a wide variety of operations. A current trend is to design and build monolithic simulation systems. These systems then often are federated with other simulations and various ancillary systems. The monolithic nature of these systems makes it challenging to model coalition operations and other operations that involve high levels of human collaboration. It is possible to use such simulations in a Net-Centric environment by encapsulating them as services, but more fundamental change is needed to enable the current class of military simulations for more agile operations.

Service Oriented Architectures (SOAs) offer a very advantageous transition point for future simulations. Rather than being deployed as separate systems (and programs) in a future networked force, simulations should be built as services that work organically within C2 services in both planning and execution phases of operations. Building simulation services in an SOA also would permit these services, if developed to implement general functionality, to be integrated effectively to other service systems used outside the military, such as Geographic Information Systems (GISs).

In moving toward an SOA, semantic interoperability is one of the most difficult challenges. Data standards need to be designed that are not restrictive, but that also enable the high level of understanding needed to achieve common awareness. The work described in this paper is aimed at producing such a semantic interoperability standard necessary for evolving simulations to

work within an SOA. We view this as an essential step in enabling simulations to be used in complex endeavors.

The scope of BML today is narrow in that it is concerned with traditional military operations. This paper addresses the very difficult issue of coalition interoperability as well as the technical issues of interfacing legacy C2 and simulation systems. Our approach is to build semantic services that will accommodate both a traditional planning process as well as future collaborative processes. While we view this as leading to a more agile capability, the current C2 and Simulations with which we have experience primarily deal with kinetic effects and well-structured domains.

This paper reports on the first, successful phase of a multinational project that is demonstrating and evaluating a capability for interoperation of Command and Control (C2) systems with Modeling and Simulation (M&S) systems for coalition operations. The resulting system is quite complex (~N^2 possible interactions among N subsystems), but the use of a common language makes it relatively simple to implement and allows full automation of the interface, resulting in timely impact on coalition operations. The result provides an important step toward true agility, since it provides rapid, effective information sharing among coalition organizations. The key enablers of this capability are an emerging standard language for military operations, the Battle Management Language (BML) [1] and an SOA using a Web service repository [2] based on the Joint Command, Control and Consultation Information Exchange Data Model (JC3IEDM). The Web service schema and Reference Implementation software which provided the basis for interoperation was developed in the Joint Battle Management Language (JBML) project [3, 4].

The need to interface C2 systems with simulation systems has long been established. The Department of Defense has worked at great length to establish general simulation-to-simulation standards (*e.g.*, High Level Architecture - HLA). However, work to establish standards for C2-simulation interoperability has been limited. Accordingly, almost every simulation has a unique C2 interface. A notable exception is the BML initiative and its use of the international Multilateral Interoperability Program (MIP) data standard JC3IEDM as a system-independent community vocabulary for passing plans orders, and reports between C2 systems and simulations. BML seeks to manage the potentially high level of complexity in Service, Joint and Coalition C2-simulation interoperation by providing a common means of exchanging information that all C2 and simulation systems can implement.

BML had its beginnings in work sponsored by the US Army's Simulation-to-C4I Interoperability Overarching Integrated Product Team (SIMCI OIPT). Carey *et al.* [5] describe the overall process used to show the feasibility of defining an unambiguous language, based on manuals capturing the doctrine of the US Army. This 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 [6] and the US Joint Staff's Universal Joint Task List [7], and including the field manuals Army elements such as Field Artillery, Air Defense Artillery, Engineers, and Military Police, down to the platoon level. This work resulted in definition of an unambiguous Operational Order (OPORD) using the traditional "5 Ws" (who-what-when-where-why) to describe military tasks [1]. This first 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 chartered to build on that initial work, with two main objectives: (1) using Web technology for the information exchange between the systems' interfaces for the network-centric environment, and (2) using the MIP's Command and Control Information Exchange Data Model (C2IEDM, an earlier version of the JC3IEDM) as a basis to represent the information to be exchanged between the systems. 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, using the Theater Battle Management Control System (TBMCS) and the Air Warfare Simulation (AWSIM) system with positive results [8]. XBML also became the basis for an international experiment, driven by interest of the Exploratory Team which was formulating the proposal that led to MSG-048. That experiment and its results are described in detail in [9].

The latest progression in US work on BML was the JBML, which expanded BML into the Joint arena including ground, air and maritime domains and urban warfare, as demonstrated in May 2007. JBML also achieved considerable technical progress by creating a revised Web service schema, based on lexical grammar and designed to facilitate expansion into other military realms, which was implemented in the open source JBML Web Services as described below [3, 4]. In parallel with JBML, the US Army Topographic Engineering Center (TEC) has been developing a geospatial BML (geoBML) which will bring a wealth of geospatial data to the C2-M&S environment [10].

The need for C2-simulation interoperability in coalition operations is even greater than that of national Service and Joint operations. Coalitions must function despite greater complexity due to significant differences among doctrine and human language barriers; thus the agility to train and rehearse rapidly before the actual operation is highly important [11]. The NATO Modeling and Simulation Group (MSG), in recognition of this need, chartered Technical Activity MSG-048 to explore the promise of BML in coalitions, combined with SOA technologies [12]. BML activities to date are summarized in Table 1.

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			X	X	
AOBML			X		X		
MIP/JC3IEDM	X	X	X	X			X

The remainder of this paper describes the technologies and development approach used in MSG-048's successful initial demonstration, held at the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2007 in Orlando, Florida. The following major sections describe the BML representation, the Web services that were used, and the resulting demonstration. The paper concludes with observations regarding the enablers for MSG-048's

rapid development and a look at its future plans, including reflections as to how this work relates to the concepts of collaboration and agility.

2. Battle Management Language Representation

The JBML schema and Web services have a structure based on the Command and Control Lexical Grammar (C2LG). This section describes the C2LG and then shows how it was used to structure development of the JBML schema.

2.1 Using a Lexical Functional Grammar to Formalize BML

While the BML concept has been described in a number of papers, in particular [1,5], the formal language required additional research. Here we describe how the C2LG provides such a formalism for BML, as developed in the computational grammar work of Hieb and Schade [13], following linguistic principles. These grammars also provide the basis for representing more complex concepts used for Command Intent as shown in [14]. Here we describe only the tasking grammar that was used to design the Joint Battle Management Language schema. The vocabulary used is taken from the JC3IEDM, and also is used in the C2LG.

The format of orders is defined by the NATO Standard STANAG 2014 "Format for Orders and Designation of Timings, Locations and Boundaries" [15]. In this format, an Operational Order is divided into five Sections: (1) Situation, (2) Mission, (3) Execution, (4) Administration and Logistics, (5) Command and Signal, with associated annexes. Section 3 is used to "summarize the overall course of action," "assign specific tasks to each element of the task organization," and "give details of coordination." The tasking grammar [13] scope covers Section 3, "Execution," which consists of the Commander's Intent and the assignment of single specific tasks to specific units, as well as giving details of coordination. We define the basic rule of the tasking grammar as:

(1) Tasking → CI OB* C_Sp* C_T*

This rule means that a tasking expression consists of the command intent (indicated by CI), basic order expressions to assign tasks to units (OB), spatial coordination (C_Sp), and temporal coordination (C_T). The asterisk indicates that arbitrarily many of the respective expressions can be concatenated together.

According to the linguistic principles given above, we define basic order expressions as composed of a verb, denoting a task, and its frame. For the tasking grammar, tasking verbs are taken from JC3IEDM's table "action-task-activity-code." Thus, the rules to expand OB have the general form given in (2a). (2b) and (2c) give instance examples for the tasks "advance" and "defend," respectively.

(2a) OB → Verb Tasker Taskee (Affected|Action) Where
Start-When (End-When) Why Label (Mod)*

(2b) OB → ***advance*** Tasker Taskee Route-Where
Start-When (End-When) Why Label (Mod)*

(2c) OB → ***defend*** Tasker Taskee Affected At-Where
Start-When (End-When) Why Label (Mod)*

Tasker is the name of the one who gives the order. Taskee is the name of the unit that is ordered to execute the task. Start-When and End-When are temporal phrases expressing when the execution of the task has to start and when it has to be finished. End-When is not needed in all cases, as indicated by the parentheses. Tasker, Taskee, Start-When, and End-When appear in each basic order rule.

Affected in (2a) must be a term in the expression if someone (*e.g.*, the enemy) will be directly affected by the task. Whether Affected is part of a rule depends on the tasking verb. For example, it is there in the case of ***attack*** or ***defend*** because the executing unit is tasked to attack the enemy or to defend against the enemy, but it is not there in the case of ***advance***. The tasking verbs come with frames that express which types of constituents are required, *e.g.*, a constituent of type Affected. This enforces the principles of completeness and coherence. Action is similar to Affected. It only appears if the task affects an action, as a task of type ***assist*** does – the unit is tasked to assist the execution of another task by another unit. In addition, the type of the Where also is determined by the verb. It is currently an At-Where or a Route-Where. An At-Where denotes a location, and a Route-Where a path to a location. A Route-Where can be expanded to more complex concatenations of constituents as in "**from** LocationA **to** LocationD **via** LocationB **and** LocationC."

Each basic rule ends with Why, Label and the optional Mod. Why represents a reason why the task specified by the rule is ordered – the mission's purpose. Label is a unique identifier for its task. By this identifier the task can referred to in other expressions, especially in temporal coordination. The optional Mod (for modifier) is a wild-card that represents additional information that can be used to describe a particular task, for example "formation" to specify a particular formation for an advance or "manner" to express whether the task in question has to be completed as fast as possible or more slowly, without taking any risks. Modifiers are particularly important for decision support.

2.2 The JBML Schema

The system plan for JBML Phase 1 sought to take advantage of the above-described work of Hieb and Schade to create a schema that would allow orderly expansion as BML enters new domains. Knowing that the C2LG can represent any Ground Domain order that has been expressed in BML to date, we structured a schema based on the C2LG that has similar expressiveness. It is a measure of the effectiveness of this approach that the JBML project was able to add the Air and Maritime Domains without deviating from the pattern set for the Ground Domain. However, it is important to note that, in the work to date, the Web services implement the syntax of C2LG, not its semantics. For example, the JBML Web Service will accept an order that directs a Ground Domain unit to cross a deep body of water even though the unit has within its organic transport capability neither aircraft nor maritime vessels. The full power of a grammar, as envisioned for future versions of BML, will be able to identify such anomalies and reject them as incorrect.

Figure 1 shows part of the JBML schema. A review of the contents of this figure will show that it is directly drawn from the C2LG as described above.

```
<xsd:complexType name="TaskType">
   <xsd:choice>
      <xsd:element name="GroundTask"
                   type="GroundTaskType" minOccurs="0"
                   maxOccurs="unbounded"/>
      <xsd:element name="AirTask"
                   type="AirTaskType" minOccurs="0"
                   maxOccurs="unbounded"/>
      <xsd:element name="MaritimeTask"
                   type="MaritimeTaskType" minOccurs="0"
                   maxOccurs="unbounded"/>
   </xsd:choice>
</xsd:complexType>

<xsd:complexType name="GroundTaskType">
   <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>
...
```

Figure 1. JBML mid-level schema showing role of C2LG

3. Web Service Architecture

This section describes how the JBML Web services implement the grammar-based schema described in section 2. Key aspects are the layered architecture and the middleware function provided by the service which enables asynchronous communication among the C2 systems and simulation systems using BML.

3.1 Layered Services

The Web services were implemented as open source Java software by the JBML project. The intention was to provide a reference implementation serving both as basic infrastructure for the project, in support of a standards draft for the Simulation Interoperability Standards Organization's Coalition Battle Management (C-BML) standards effort [16]. The implementation is based on Web service networking standards [17] and was originally reported in [18]. Figure 2 provides an overview of the JBML Web service Architecture. The layers will be described in detail in the following subsections, with references to the numbered interfaces given in parentheses, *e.g.* "(3)". The layers are:

- 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 BML Base Services (BBS) which the terms in which the DCS are represented. These represent the information element groups that specify information objects of interest such as the 5Ws of military orders (who, what where, when, why) and other constructs of interest. In Web services terminology, these implement "business rules."
- The lowest layer represents the information exchange of information elements. This layer is normally hidden from the user. In JBML, this is called the Common Data Access Service (CDAS). It provides for access to the database.

It would be possible to implement these three layers as cascading Web services, where the Web service at one layer invokes a Web service at a lower layer. While the layers are in fact configurable to be exposed as Web services, the design in Figure 2 avoids that because it would compound the already low performance of Web services. Since the three layers are present in the same computer, we access the lower layers through a software API rather than the Web service wrapper.

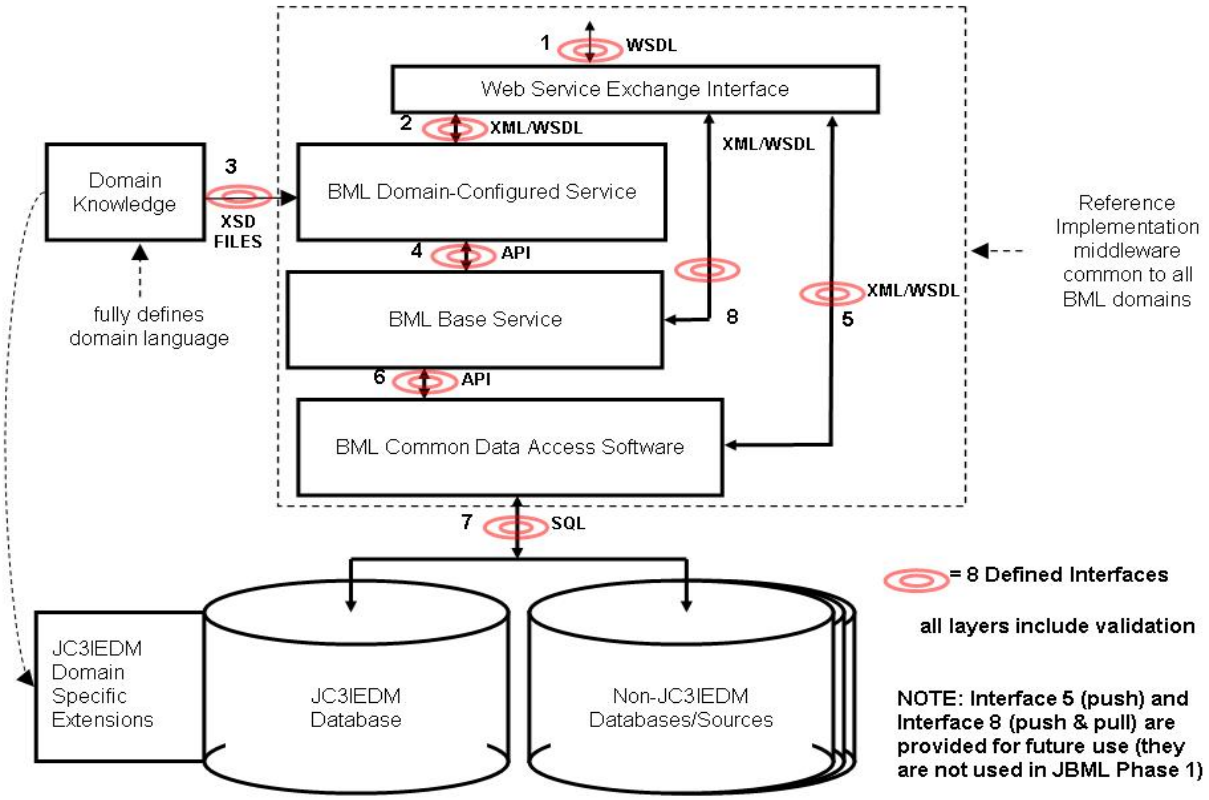


Figure 2. Web service layered architecture overview

3.2 BML Domain Configured Service (DCS)

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). The primary input to the DCS is an order, composed in terms of tasks such as the *GroundTaskType* shown in Figure 1. Another example would be a position report (not yet implemented), where the transaction at this layer will include all information about the updated location (*e.g.*, who, where, when-valid, precision, confidence in the data, etc.).

The DCS is implemented in the Document-Literal mode by a generic Web service that is configured by an XML schema, the Domain Knowledge Schema (DKS). It 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. 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 next.

3.3 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 identifies 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 Figure 1 is intentional; JBML uses these primitives as composites.

The BBS services are not accessed by the user of JBML, who 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 (8). Figure 3 shows a segment of the BBS schema pertaining to the WhenType. Note that values of the "WhenModifier" are taken from the JC3IEDM. In either form of access, the BBS provides a way to deal with the fact that the various *Who/What/When/Where/Why* transactions may require multiple database table updates under the JC3IEDM (in the case of *What*, up to 25 tables). As a result, it is important that any such transaction be treated atomically so that two of them do not have interleaved access when updating the database, as that could leave the database in an inconsistent state.

The JBML schema uses the MIP JC3IEDM definitions as the lexicon of its vocabulary. The JBML Web Service stores all elements of its BML order in a MIP-compliant JC3IEDM database. It therefore is essential to have well-defined mappings between the BBS schema and the JC3IEDM entities that define the database tables. This mapping forms the business logic of JBML; compliance with the mapping ensures that the database always can be used to pull the same orders that were pushed. An example of such a mapping, displayed in the IDEF1x [19], is given in Figure 4.

```
<xsd:complexType name="WhenType">
    <xsd:choice>
        <xsd:element name="DTG" type="DtgType"/>
        <xsd:element name="RelativeToTask" type="LabelType"/>
    </xsd:choice>
    <xsd:attribute name="modifier" type="WhenModifier" use="optional"
            default="AT"/>
</xsd:complexType>
<xsd:simpleType name="WhenModifier">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="AFT"/>
        <xsd:enumeration value="ASAP"/>
        <xsd:enumeration value="ASAPAF"/>
        <xsd:enumeration value="ASAPNL"/>
        <xsd:enumeration value="AT"/>
        <xsd:enumeration value="BEF"/>
        <xsd:enumeration value="NLT"/>
        <xsd:enumeration value="NOB"/>
    </xsd:restriction>
</xsd:simpleType>
```

Figure 3. Segment of the BBS schema defining the type for *When*

3.4 BML Common Data Access Service (CDAS)

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 also exposes a Web service that allows inspection of every database table used in any domain of BML, 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, thus sustaining an audit trail. The CDAS was developed based on a set of Web services provided by the developer of the NATO *Pathfinder* experiment [20], which did all of its data exchange at the level of the JC3IEDM.

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 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. 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. (If this is done, the participating systems will need to deal with the database consistency issue raised in section 3.3 above.) 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.

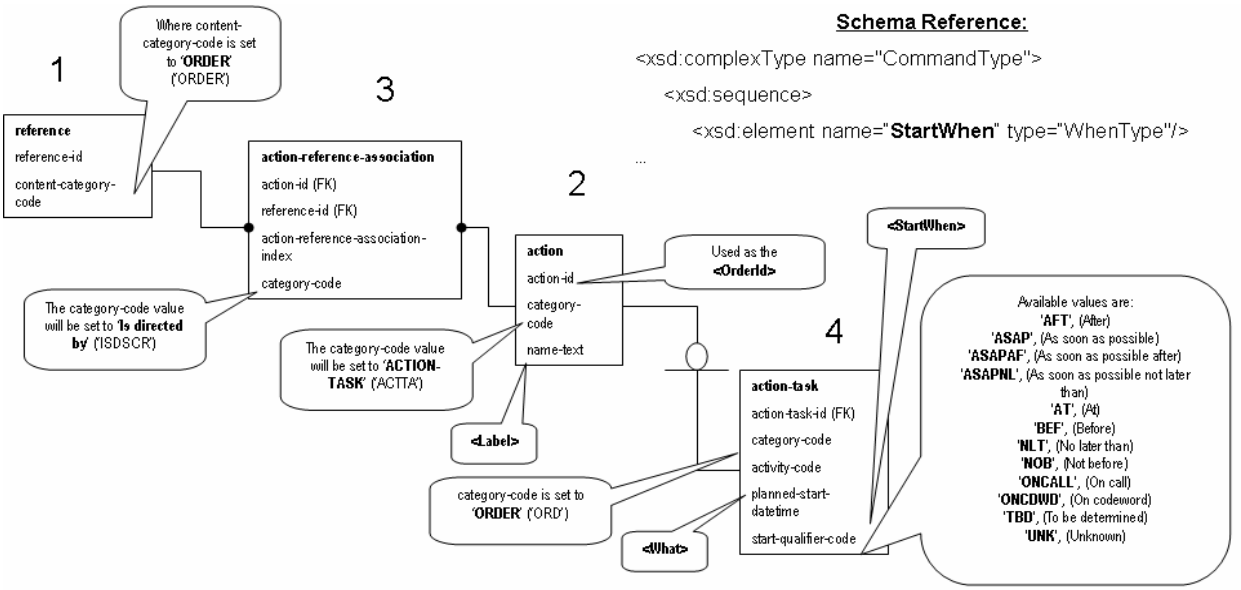


Figure 4. JBML mapping to JC3IEDM: schema field <StartWhen>

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. We believe the real power of BML lies in the ability to present abstract, semantically rich orders and reports, so we favor use of the high-level schema. However, this need not preclude use of software objects that compose transactions on the JC3IEDM interface according to well-established business rules and thus allow interoperation via the JC3IEDM.

3.5 Middleware Role

An extremely powerful capability gained from the Web service repository is asynchronous operation of system elements. Figure 5 shows how the various elements in the JBML Phase 1 demonstration were interconnected. In principle, it would have been possible to bypass the Web service and allow the BML output of each C2 system to become the BML input of each simulation system. However, that would require all subsystems to be online and fully operable simultaneously. In a distributed system, especially one that is under pressure from adversaries and might be using battlefield communications, this is not a reasonable assumption. Using the Web service allowed any subsystem to push or pull BML orders at any time. The whole system therefore remained functional despite individual subsystem outages. This style of operation proved particularly useful in the MSG-048 development process, which spanned six time zones.

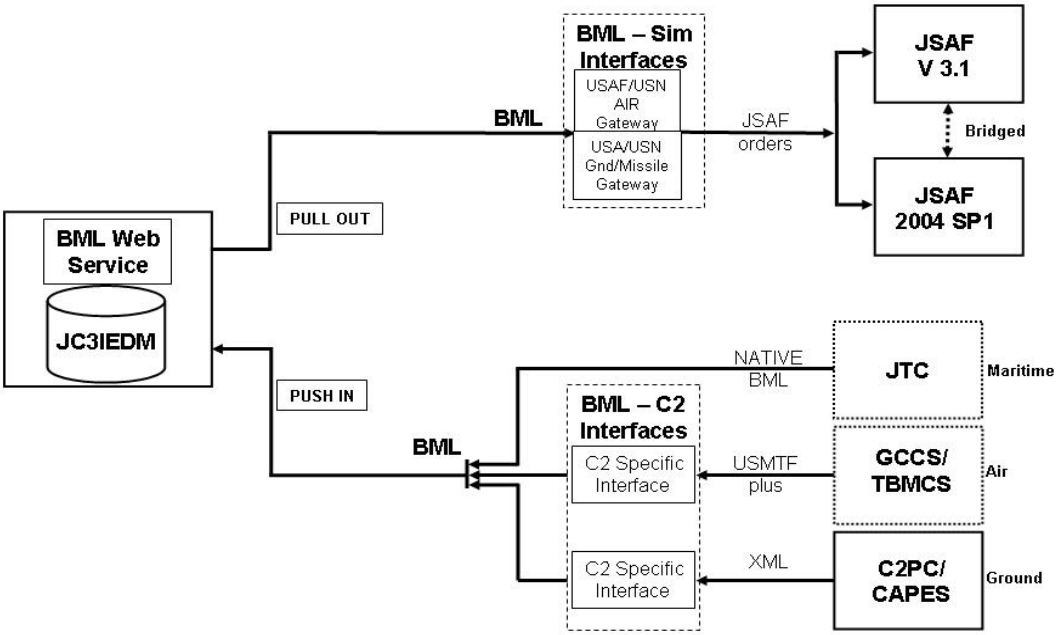


Figure 5. System configuration for Phase 1 JBML demonstration

4. MSG-048 Demonstration

The proof of the utility of the JBML design came in the MSG-048 demonstration. The JBML Phase 1 demonstration was completed in May, 2007 just before the MSG-048 meeting and part of the demonstration was repeated for the meeting. This stimulated participants to undertake an ambitious project: multinational C2-simulation interoperation, with a commitment to completion in six months. Figure 6 shows the architecture of the MSG-048 November, 2007 demonstration.

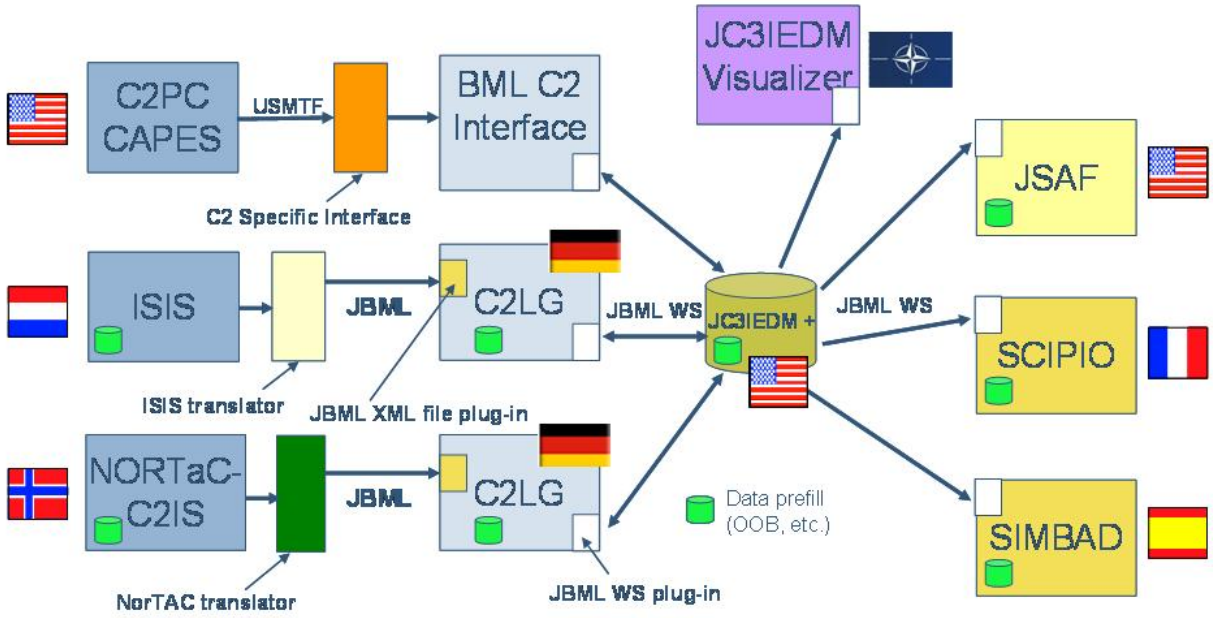
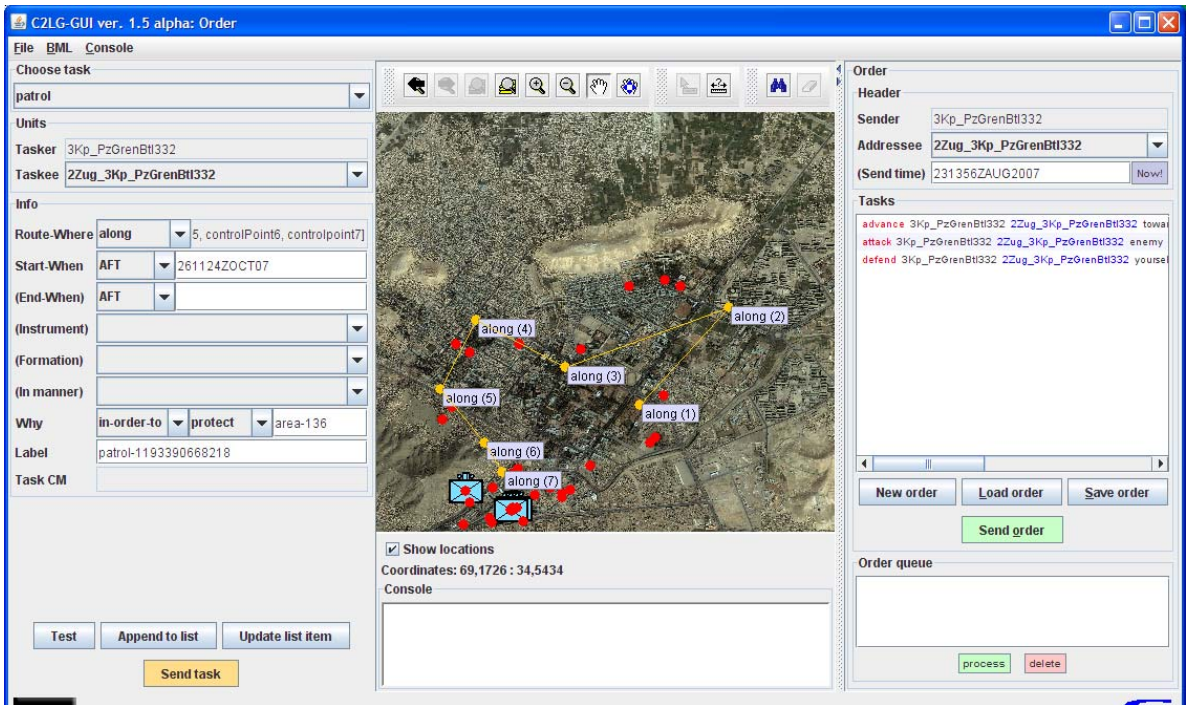


Figure 6. MSG-048 Demonstration Architecture



4.1 C2 Lexical Graphic User Interface

In order to explore the grammar specified in Section 2.1, a C2LG Graphical User Interface (GUI) was constructed by the German research center FGAN. This GUI is shown in Figure 7. The C2LG GUI was created to generate "pure" BML statements that were valid grammar statements. Thus the C2LG implemented functionality to create BML sentences that corresponded to both task and report sentences.

During the initial design phase of MSG-048, it was realized that the C2 systems being used would not generate a complete set of information as needed for the JBML web services. The C2LG GUI was used as an "integration hub" to take the input from C2 systems and construct a "valid" JBML Order that could be sent and ingested by different nation's simulations.

In order to implement this functionality, a number of modules were added to the C2LG GUI in order to read in incomplete order information, allow the order to be completed with the GUI, and then format the order as a JBML web service. Since the JBML web services did not perform validity checking at the "domain" level, the C2LG GUI also served as a useful check that the order contained tasks that were syntactically valid.

4.2 C2 Systems

The three national C2 systems, shown in Figure 6, that were used in the MSG-048 demonstration were developed for distinctly different purposes. Each of them is an operation ground C2 system in the nation that provided it.

C2PC/CAPES

This is a workstation-based C2 system, developed by the US Marine Corps and also used by the US Army. It features and architecture supporting *injector* modules that can manipulate and display external data. The US Army has adapted its CAPES system to serve as a planning module injector for C2PC. Following the injector architecture, a BML module also could be added such that C2PC would generate BML as a native language; at present it is interfaced to the BML Web service by a software module. C2PC also has the advantage that it is available in an unclassified form for US participation in experiments.

NORTaC

The NORTaC C2I system supports the Norwegian army at brigade level and below, in planning operations and to maintain situational awareness. It also coordinates use of indirect fire weapons. NORTaC is integrated with tactical communications networks, military messaging systems and GPS. It can exchange information with allied forces through a MIP gateway. Other interfaces are with the Battlefield Management System, Link 16 and artillery Common Technical Interface Design Plan. The system has been used in large national and multinational exercises. NORTaC was used with the C2LG GUI in the MSG-048 demonstration.

ISIS

ISIS is the Integrated Staff Information System of the Royal Netherlands Army. It is one of the first applications that resulted from their C2 Support Centre program to produce a generic, configurable and distributed Command and Control information system in an evolutionary process. The overall system, known as C2 Framework (C2FW), is the baseline for a suite of C2 applications that will provide staff sections, vehicles and individual combatants with a common operational picture (COP) to provide situational awareness. The C2FW is the foundation for a family of C2 Information Systems, including ISIS, which is the member of the C2FW family that is aimed at the static domain (compound, command post). ISIS was used with the C2LG GUI in the MSG-048 demonstration.

4.3 Simulation Systems

As with C2 systems, the three national simulation systems used in the MSG-048 demonstration have very purposes and different models of operation, although they do have the commonality that all are intended for use in training.

JSAF

This is a constructive simulation that represents objects down to individual platforms and combatants. It was developed and is maintained for training and experimentation by the US Joint Forces Command (JFCOM), which makes it available to US forces and allies. It can represent a very wide range of land, air, and maritime elements. JSAF can function in HLA federations and also can be linked to other simulations (including other instances of JSAF) by the DIS protocol. The JBML project has developed an interface between JSAF and the XML schema associated with the JBML Web Services.

SCIPIO

SCIPIO is a full command post training simulation system built for the French Army and equipping their Army Headquarters Training Center in support of staff training for brigade and division headquarters. SCIPIO provides the training center with 3 major key capabilities: (1) an Automated simulation control based on command agents; (2) Command, Control, Communications, Computers and Intelligence (C4I) integration, providing formatted reports to compatible C4I systems as well as orders transmitted to the command agents and monitored by exercise controllers; (3) warfare modeling with high level automation where parent entities are able to command and control their assigned subordinates. The SCIPIO training environment includes exercise preparation tools, facilities for operating 35 battalion-level response cells, and after-action tools for visual reply and statistical analysis.

SIMBAD

This is a training simulation of the Spanish forces. It is designed around predefined engagement scripts that are executed under control of a Tactical Events Manager. SIMBAD has a geographic information system (GIS) based GUI with geographical and tactical overlays. It is able to function in an HLA federation, using a proprietary Federation Object Model inspired by the MIP C2IEDM. However, SIMBAD does not employ command agents. In order for it to accept BML orders, a set of scripts was generated that represented all the scenario activities that might be required by the orders. These scripts were invoked by a BML interface to SIMBAD,

4.4 Web Services

The Web services used in the MSG-048 demonstration were the JBML Web Services described in section 3 above. Only two small changes were implemented for MSG-048: the ability to have a Task-level Control Measure, and the addition of the Route-Where described in section 2.1. Yet two other C2 systems and two other simulations, each with different design goals, were able to achieve a fair degree of interoperation without any significant workarounds. This underscores the value of using the C2LG as a basis for the Web service schema. The Web services were maintained available full-time on an Internet site. Because each C2 and simulation system can push and pull orders independently and asynchronously, this allowed parallel development by all national groups involved, and was credited by project participants as empowering the rapid development of the demonstration capability.

5. Conclusions

The MSG-048 November 2007 demonstration provides very strong evidence in favor of the techniques employed as a basis for an SOA approach to using simulations with C2 systems. First and foremost, the approach based in formal linguistics (using the C2LG-inspired schema) and associated Web Services, with strong semantics (using the JC3IEDM-derived vocabulary), provided an extremely effective medium of expression for communication among the various systems. Perhaps equally important, the network-centric development methodology proved highly effective, especially when employed by national development teams with a highly cooperative spirit, including technical developers and military subject matter experts.

The project described in this paper is only the beginning of MSG-048's development of a BML/SOA-based environment for evaluation of the potential of these techniques for coalition C2/simulation interoperation. The plan for 2008 calls for evaluation of multiple additional experimental coalition configurations. After that, if the technology and associated development methodology proves as successful as indicated by this first step, MSG-048 will seek its application in a NATO exercise or experiment.

We expect the science of BML also will continue to develop. One dimension of this development is advancing the technical application of BML with new simulation capabilities, as described in a companion paper at this conference by Borgers *et al.* [21], showing how the C2LG can be used to communicate command intent to agents. Muguira *et. al.* [22] show how the C2LG can be used for embedded simulation in a C2 application and Gustavsson *et. al.* [23] show how BML can facilitate military training. Another dimension is application of BML techniques to new domains, for example Crisis Management as described in [22]. Underlying all of this is continued development of lexical grammar techniques, as in the companion paper by Schade and Hieb [25], which investigates new, more powerful languages, derived from the C2LG work, that are designed to be used in a Crisis Management or other multi-agency environment where military C2 is at most part of the ongoing communication and consultation.

References

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

[2] Tolk, A. and J. Pullen, "Using Web services and Data Mediation/Storage Services to Enable Command and Control to Simulation Interoperability," Paper presented at 9th IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2005), Montreal, Canada, 2005.

[3] Pullen, J., M. Hieb, S. Levine, A. Tolk, and C. Blais, "Joint Battle Management Language (JBML) - US Contribution to the C-BML PDG and NATO MSG-048 TA," Paper presented at IEEE European Simulation Interoperability Workshop, June 2007.

[4] Levine, S., M. Pullen, M. Hieb, C. Pandolfo, C. Blais, J. Roberts and J. Kearly, "Joint Battle Management Language (JBML) Phase 1 Development and Demonstration Results," Paper presented at IEEE Fall Simulation Interoperability Workshop, Orlando, FL, 2007.

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

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

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

[8] Perme, D., M. Hieb, J. Pullen, W. Sudnikovich, and A. Tolk, "Integrating Air and Ground Operations within a Common Battle Management Language," Paper presented at IEEE Fall Simulation Interoperability Workshop, Orlando FL, 2005.

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

[10] Hieb, M., S. Mackay, M. Powers, M. Kleiner, and J. Pullen, "The Environment in Network Centric Operations: A Framework for Command and Control," Paper presented at 12th International Command and Control Research and Technology Symposium, Newport, RI, 2007.

[11] Tolk, A, M. Hieb, K. Galvin, L. Khimeche, and J. Pullen, "Developing a Coalition Battle Management Language to facilitate Interoperability between Operation CIS and Simulations in support of Training and Mission Rehearsal", Paper presented at 10th Command and Control Research and Technology Symposium, McLean, VA, 2005.

[12] 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," Paper presented at IEEE Fall Simulation Interoperability Workshop, September 2006.

[13] Schade, U. and M. Hieb, "Development of Formal Grammars to Support Coalition Command and Control: A Battle Management Language for Orders, Requests, and Reports, Paper presented at 11th International Command and Control Research and Technology Symposium, Cambridge, UK, 2006.

[14] Hieb, M. and U. Schade, "Formalizing Command Intent Through Development of a Command and Control Grammar," Paper presented at 12th International Command and Control Research and Technology Symposium, Newport, RI, 2007.

[15] NATO Military Agency for Standardization, *STANAG 2014: Formats for Orders and Designation of Timings, Locations, and Boundaries*, Brussels, BE, Edition 9 (2000).

[16] Blais, C., K. Galvin and M. Hieb, "Coalition Battle Management Language (C-BML) Study Group Report," Paper presented at the IEEE Fall Simulation Interoperability Workshop, Orlando FL, 2005.

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

[18] Pullen, J., K. Makineni, and P. McAndrews. "A Grammar-Based Web Service Enabling Multi-domain Distributed Interoperation of Command/Control and Simulation Systems, Paper presented at 11th IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2007), Chania, Greece, October 2007.

[19] US National Institute of Standards and technology, Federal Information Processing Standards Publication 184 (IDEF1X), 1993.

[20] Tolk A. and J. Boulet, "Lessons Learned on NATO Experiments on C2/M&S Interoperability," Paper presented at IEEE Spring Simulation Interoperability Workshop, Norfolk VA, 2007.

[21] Borgers, E., M. Spaans, J. Voogd, M. Hieb, and R. Bonse, "Using a Command and Control Language to Simulate Operations in a Multi-Agent Environment," Paper presented at 13th International Command and Control Research and Technology Symposium, Bellevue, WA, 2008.

[22] Gustavsson, P., J. Wemmergård, J. Garcia, and M. Norstedt Larsson, "Expanding the Management Language Smorgasbord towards Standardization of Coalition-Crisis Management Language C-CML," Paper presented at IEEE Spring Simulation Interoperability Workshop, San Diego CA, 2006.

[23] Muguira, J. A., V. Kamath, and M.R. Hieb, "Applying Battle Management Language (BML) to a Command and Control Application: Lessons Learned", Paper presented at the IEEE Spring Simulation Interoperability Workshop, Providence, RI, 2008.

[24] Gustavsson, P., M.R. Hieb, M. Grönkvist, V. Kamath, Jakob Blomberg, and Joakim Wemmergård. "BLACK-CACTUS – Towards an Agile Joint/Coalition Embedded C2 Training Environment," Paper Presented at the IEEE Spring Simulation Interoperability Workshop, Providence, RI, 2008.

[25] Schade, U. and M. Hieb, "A Linguistics Basis for Multi-Agency Coordination," Paper presented at 12th International Command and Control Research and Technology Symposium, Newport, RI, 2007.