A Grammar-Based Web Service Enabling Multi-domain Distributed Interoperation of Command/Control and Simulation Systems

J. Mark Pullen
C4I Center
George Mason University
mpullen@gmu.edu

Krishna Makineni
C4I Center
George Mason University
cmakinen@gmu.edu

Priscilla McAndrews
C4I Center
George Mason University
pmcandre@gmu.edu

Abstract

The Joint Battle Management Language (JBML) project is at the forefront of the current interest in supporting development of distributed systems with simulation elements by means of Web services. This paper reports on work to develop an extensible capability to serve as a basis to expand JBML coherently, in multiple domains of warfare simulation (land, air, and maritime). The project has created a schema for a Web service that serves as the external interface for orders being pushed into the Web service by the different military components' domain-unique Command and Control (C2) systems, and pulled from the service by different simulations that function as a cooperating system. The Web service consists of multiple, re-usable layers in an open source implementation that is intended to serve as a basis for an expanding open standards development effort for the Coalition Battle Management Language. We report on the approach used to design and develop the schema and supporting Web service based on a lexical grammar, and the results when three C2 systems and two simulation systems were combined, using the Web service as its core.

.

1. Introduction

The military uses networked information technology known as Command and Control (C2) systems to provide direction across distributed forces and status feedback from those forces. The military distributed simulation technical community has sought for many years to interface these C2 systems with distributed simulations in order to provide more valid and realistic results and to reduce costs associated with manual interfaces between C2 and simulation systems. Initial attempts along these lines created custom-built linkages that bound a given C2 system to a related simulation system. Creating such interfaces proved very expensive because of the large effort to understand the two systems and ensure the information exchanged across the interface had the same meaning to both systems.

The US Army undertook a project to show the feasibility of defining a common language for the C2-simulation interface, known as Battle Management Language (BML). The BML project achieved considerable success by methodically analyzing the doctrine of the US Army, contained in [1], and defining an interface based on the standard US Army five-paragraph operations order [2]. This, and any BML, is intended to serve as input to a fully automated process and therefore must be unambiguous.

In the aftermath of the Army success, there was significant interest in creating a standard, open definition of BML. This was envisioned as a "Joint" (all military components, including ground, air, and naval forces [3]) and also as a "Coalition" (spanning multiple allied nations) version of BML. It was to be based on open technologies, in particular Web services and the Command and Control Information Exchange Data Model (C2IEDM) [4]. The C2IEDM is a NATO standard for structuring data, as in relational databases, that is developed by the Multilateral Interoperability Programme (MIP), an open standards body [5]. The resulting project [6] was an important factor in formation of an open standards effort known as the Coalition Battle Management Language (C-BML) of the Simulation Interoperability Standards Organization (SISO) [7]. Subsequently, the NATO Modeling and Simulation Group (MSG) investigated the possibility of using C-BML to address NATO needs. Their one-year study was capped by demonstration of C2 and simulation systems of the USA and France interoperating over Web services, using the C2IEDM for data exchange [8].

Along the way to development of a C-BML capability, the approach to the use of Web services has become more sophisticated. Initial interest based on a mediation approach [9] has grown to a view of functional layers supporting re-use [10] and composition of multiple C2IEDM transactions into a single building-block service [11]. All of this set the stage for the Joint Battle Management Language Project (JBML) [12, 13]. JBML set out to create a design and reference implementation of a proposed standard approach to a family of related BML dialects, each particular to a particular military domain, as depicted in Figure 1. It also adopted the Joint Command, Control, and Consultation Information Exchange Data Model (JC3IEDM), the latest evolution of the C2IEDM. Interest in this model may ultimately expand to domains of structured, cooperative activity beyond the military, for example dealing with civil crises such as fires, floods, and storms [14]. This paper extends [13], providing details on the design and implementation of the Web service.

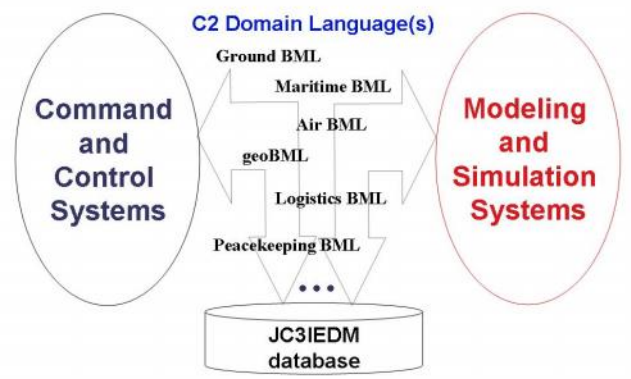

Figure 1. System concept of JBML

2. Grammar as a Basis for BML

To provide a basis for JBML, it was necessary to step back from the ad-hoc interfaces employed by its predecessors and define a conceptual basis that can be used across a wide range of domains. The necessary theoretical basis for this was found in the work of Hieb and Schade [15, 16], who have analyzed the information content of BML orders and status reports. They found that every order needed in the BML system can be represented by a lexical grammar [17] using a production rule of the general form:

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

The symbols of this production rule for the grammatical phrase B are:

Verb is an action, normally a task;

Tasker is a *Who*, the unit which commands the task;

Taskee is a *Who*, the unit which executes the task;

Affected is a *Who*, the unit affected by the task;

Action is another action/task affected by the task;

Where is a *location phrase*;

When is a *time phrase*;

Why is a "terminal symbol" giving the purpose of the action;

Label is given to the task in order allow it to be referred in other basic expressions;

Mod is a modifier, specific to the nature of the task;

* refers to zero or more occurrences of (Mod)

( ) indicates options

Given these grammar symbols, the JBML Web service was designed based around the primitives shown in Figure 2 below. The grammar provides a means of expressing the semantics of BML, while the JC3IEDM provides its vocabulary.

<task>	<tasker-who>
<taskee-who>	<affected-who>
<what>	<where>
<start-when>	<end-when>
<why>	<label>
<modifier>	

Figure 2. Web service design primitives

The plan for C-BML development called for use of an XML representation to be used from the beginning of the standardization effort, supplemented by a grammar-based approach after the first year. However, some structure for the XML had to be adopted in order for the JBML project to go forward. The grammar of Hieb and Schade was adopted as a basis for the XML schema because it provides a structure that has been shown to be capable of expressing the full range of BML orders and also a clear way to express those

orders. An example of the resulting XML schema section for the Ground domain is shown in Figure 3 below.

```
<xsd:complexType name="GroundTaskType">
        <xsd:sequence>
                <xsd:element name="TaskeeWho"
                                type="WhoType"/>
                <xsd:element name="What"
                                type="WhatType"/>
                <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="WhyType"
                                minOccurs="0"/>
                <xsd:element name="Label"
                                type="LabelType"/>
        </xsd:sequence>
</xsd:complexType>
```

Figure 3. Ground domain task schema fragment derived from Hieb-Schade Grammar

3. The Multilayered Web Services

This section describes 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 submit this as a standards draft to the C-BML standards effort. The implementation is based on Web service networking standards [18]. Figure 4 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 the 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 of military orders (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, this is the Common Data Access Service (CDAS) which provides for access to the JC3IEDM database.

It would be possible to implement these three layers as cascading Web services. While the layers are in fact configurable to be exposed as Web services, the design in Figure 4 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.

3.1 BMLDomain Configured Service (DCS)

In JBML, all input/output for the middleware Web service occurs at this level. 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). Figure 5 shows a segment of the XML that defines an order, in terms of tasks such as the GroundTaskType shown in Figure 3. For a position report which is not yet implemented, the transaction at this layer will include all information about the updated location (e.g., who, where, when-valid).

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. 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 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.

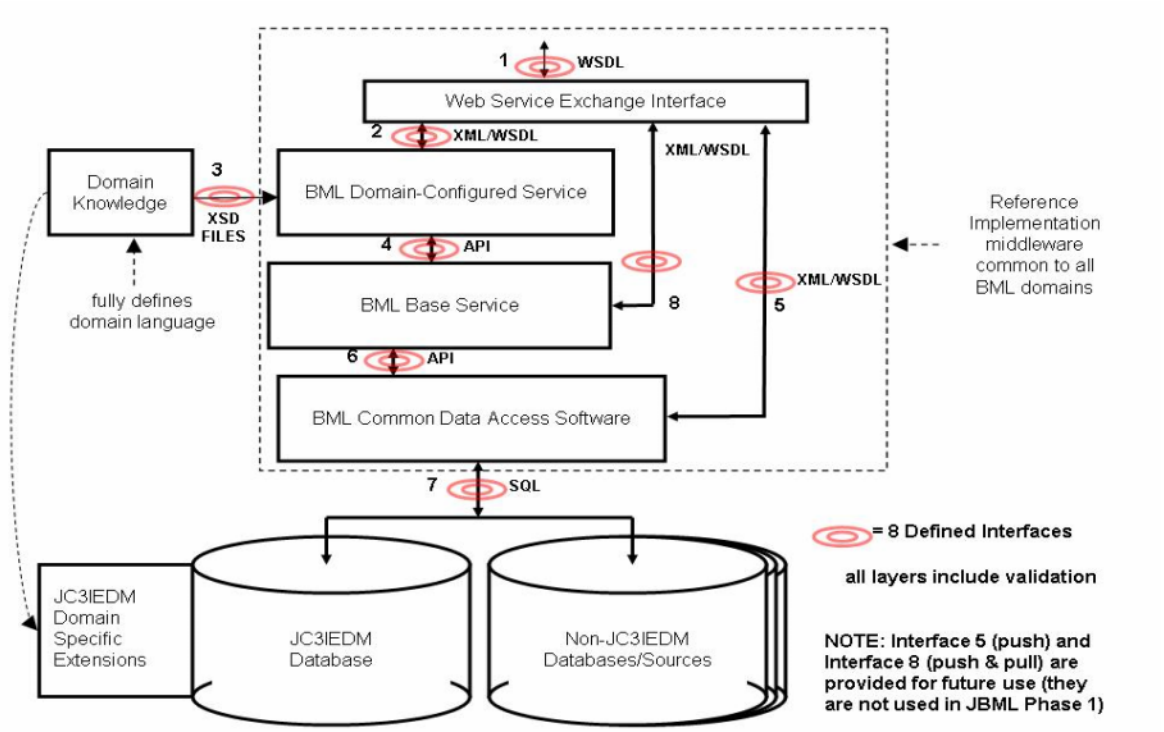

Figure 4. Web Service Layered Architecture Overview

```
<xsd:complexType name="OrderType">
   <xsd:sequence>
      <xsd:element name="OrderMode" type="OrderModeType"
            minOccurs="0" default="SINGLE"/>
      <xsd:element name="TaskersIntent" type="FreeTextType"
                minOccurs="0"/>
      <xsd:element name="Task" type="TaskType" 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="EnemyTaskOrg" type="msdl:TaskOrgType"
            minOccurs="0"/>
      <xsd:element name="ControlMeasures"
                type="MultipleControlMeasuresType" minOccurs="0"/>
      <xsd:element name="TargetList" type="TargetListType"
            minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
<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>
```

Figure 5. XML Schema segment illustrating DKS

3.2 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 JC3IEDM information entities to be queried for each BML information element (who, what, etc.) and the validation conditions to be applied. The BBS lower level interface (6) invokes the CDAS API. The close relationship between BBS and the primitives in Figure 2 is intentional; JBML uses these primitives as composites.

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 (8). Figure 6 shows a segment of the BBS schema pertaining to the WhenType. 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 *where*, up to 14 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.

3.3 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 (makes accessible) 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 invalidate 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 [19], which did all of its data exchange at the level of the JC3IEDM.

```
<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 6. Segment of the BBS schema defining the type for *When*

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. JBML's proposed C-BML specification defines the JC3IEDM entities used and a standard XML format to access them. The CDAS lower level interface (7) provides an SQL 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 above.) Another point of view is that future phases of the C-BML standard should 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 implementing the JC3IEDM. 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.

4. Demonstration of Distributed Operation

The first phase of JBML was demonstrated in May, 2007 in the configuration shown in Figure 7. The Web service is shown at the left; the Maritime, Air and Ground C2 systems are shown at the lower right; and the two Joint-Automated Forces simulations (JSAF) are shown at the upper right. The style of command interface to the recently developed JSAFv3.1 was found more suitable for simulated operations in the Air domain, whereas we were able to save a significant software refit by using JSAF2004 to support the Ground and Maritime domains. The various C2 systems were able to push their orders asynchronously in preparation for simulation, following which the simulations pulled Joint orders and executed them. The entire system was demonstrated to be fully functional [20].

The demonstration was performed around a scenario that was a Joint Task Force located in the Caspian Sea Area. The Joint Task Force was tasked with a Joint Urban Operations Order that drove an Air Battle Plan, a Maritime Operations Order and a Ground Operations Order. Each of these processed into a common, extensible and vetted JBML Web service based on an XML schema. This information was then converted and used to drive two linked Joint Semi-Automated Forces simulations.

A Graphical User Interface was used to produce files containing Maritime orders for Tomahawk cruise missile strikes. The output files were produced in native BML and used as input to the BML Web Service during the exercise. The original intent was to integrate this with the JC3IEDM-based Tactical Collaboration system, a surrogate/prototype C2 system that has operated as a client to GCCS in Navy exercises/experiments.

A Theater Battle Management Core System (TBMCS) was used to produce files containing an Air Battle Plan and an Airspace Control Order for both Air Force and Navy Air missions. These files were produced before the demonstration and translated into the Command and Control Data Interchange Format and used for input to the BML Web Service.

The Ground Operations Order was created by the Combined Arms Planning and Execution System (CAPES) injector component of the Command and Control Personal Computer (C2PC) developed by the US Marine Corps. This is a Windows client/server network application that displays positional tactical track data and provides a complete geographically based situational awareness capability, including the capability to display Global Command and Control System (GCCS) data.

Air Tasking Order and Airspace Control Order information were converted to a BML compliant format. In the future, this function could be performed as part of the TBMCS or any other system that had the capability of producing Air Tasking Orders.

The Ground Operational Orders were exported from the CAPES/C2PC application as an XML mission file. The mission file was then read into an XML Parser application that converted the native CAPES format into BML and used the JBML Push command to store the ground forces order information within the JBML database.

The Air, Maritime, and Ground BML information was pushed into the BML Web Service which was hosted as part of a JC3IEDM compliant server. The Air Force and Maritime Air Tasking Order information was pulled out of the BML Web Service and converted to a set of JSAF scenario files and pushed to the JSAF version 3.1. Cruise Missile Maritime Order and the Ground Operational Order information was pulled out of the BML Web Service and communicated to JSAF version 2004 SP1 by way of a customized UDP interface as Distributed Interactive Simulation (DIS) (IEEE 1278) packets.

The two JSAF systems (version 3.1 and version 2004 SP1) were linked via DIS gateway applications supplied with each JSAF version. Both JSAFs displayed consistent simulation information throughout the demonstration.

5. Conclusions

This paper describes a major step forward in interoperation of military command and control systems with distributed simulations. The Joint Battle Management design, based on a lexical grammar, provides a general, extensible definition of orders and tasks using the Hieb and Schade grammar for semantics and the JC3IEDM standard for vocabulary. An open source reference implementation of this design has been developed and demonstrated supporting C2 systems for Ground, Air and Maritime domains along with two coupled instances of the JSAF simulation. We expect to continue this line of development to create a general and growing capability to exchange military orders and situation reports through a generic, unambiguous Battle Management Language interface.

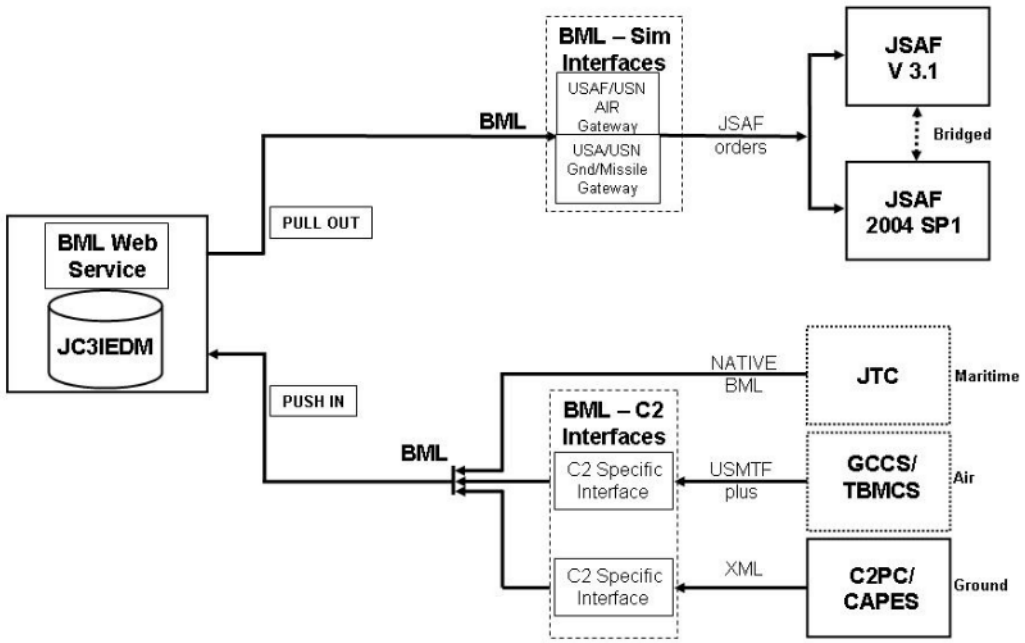

Figure 7. System configuration for Phase 1 JBML demonstration

6. References

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

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

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

[5] Multilateral Interoperability Programme website, http://www/mip-site.org

[6] 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

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

[8] 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

[9] 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

[10] 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

[11] 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

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

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

[14] 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

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

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

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

[18] 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

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

[20] Levine, S. et al., "Joint Battle Management Language (JBML) – Phase 1 Development and Demonstration Results," IEEE Fall Simulation Interoperability Workshop, September 2007