Extensible Battle Management Language as a Transformation Enabler

William P. Sudnikovich
Atlantic Consulting Services, Inc.
167 Avenue at the Common
Shrewsbury, NJ 07702
(732) 460-9416, x27
wsudnikovich@acsinc-nj.com

J. Mark Pullen
George Mason University
Computer Science/C3I MS4A5
Fairfax, VA 22030
(703) 993-1538
mpullen@gmu.edu

Martin S. Kleiner
Northrop Grumman Information Technology
12000 Research Pkwy
Orlando, FL 32826
(321) 235-7920
Martin.Kleiner@ngc.com

Scott A. Carey
Northrop Grumman Information Technology
12000 Research Pkwy
Orlando, FL 32826
(321) 235-7920
Scott.Carey@ngc.com

Keywords: Battle Management Language (BML), Extensible Modeling and Simulation Framework (XMSF), Command and Control Information Exchange Data Model (C2IEDM), Command and Control (C2), Doctrine

Abstract

This paper addresses a new approach to the longstanding problem of interoperating military command and control with combat simulations. There is great potential benefit in such a capability, both for more effective training and as a "what if" capability during military operations. We describe the Battle Management Language, an unambiguous means of describing military operations to people, simulations, and robotic forces. The process that developed the BML and its extension into the realm of Web services are described. The new capability is projected to be an enabler for the ongoing Transformation activities in the US Department of Defense. Integrated with the Command and Control Information Exchange Data Model (C2IEDM), BML as extended through Web services provides a path to improved interoperability within and among US military Services and their allied/coalition partners.

INTRODUCTION

Command and Control Interoperability with Simulations

Command and Control (C2) of military operations has always been a key target for technology advancement. The development of automated digitized military information systems has both enabled and demanded the specialization and differentiation of specific functional area support systems. The focus of communications has shifted away from human to human towards these automated systems. Along with this change has come a proliferation of operating systems, data representation schemata and other aspects that now confront us with a "confusion of languages". [1]

From the 1980's to the present, the use of simulations to support training expanded exponentially. Simulations such as Corps Battle Simulation (CBS), Close Combat Tactical Trainer (CCTT), and the OneSAF Testbed Baseline (OTB) have significantly improved both the quantity and quality of training opportunities. This is particularly true at the brigade, division and corps level, where the primary focus of training is on the command and staff processes. In the past, maneuver space, number of units, and logistic resources made it impractical and unaffordable to conduct effective and realistic command and staff training at the division and corps level. This led to the requirement for the Battle Command Training Program (BCTP). The BCTP spawned the development of the Corps Battle Simulation (CBS) to drive exercises at that echelon. Now, using these supporting simulations, the higher echelons can conduct realistic training in a more frequent and cost effective manner.

One major drawback of using computer-simulated training is the need for large contingents of support personnel, generally contractors, to act as workstation controllers and provide the interface between the training unit and the simulation. To make the training event as realistic as possible, a buffer element consisting of these workstation controllers is used to separate the training audience from the computers. The training audience operates the way they would fight and communicates their orders to the workstation controllers, who then interact with the simulation. This interaction is a two-way process with the workstation controllers acting as role players directing the operations in the simulation and providing feedback to the training audience in a standard military form. The group of workstation controllers is often as large, or larger than, the training audience. A situation that often occurs is for a unit undergoing training to have the need to use enlisted personnel to fill the workstation controller role. This results in soldiers not training in their occupation specialty with their unit but rather devoting time to running a simulation workstation that would never be deployed. While the examples cited are US Army centric, based on the community interest in BML we contend that the issues are similar across all services.

Related to this issue of large contingents of workstation controllers is the lack of an effective means to share information and directives between the simulation and Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) systems. Enabling the C4ISR systems to not only exchange information but to also to interact directly with the simulation will significantly reduce both the workstation controller requirements and the number of system unique "black box" interpreters. Good progress has been made in the area of sharing information; however, essentially no progress has been made in the area of controlling the simulation directly from the C4ISR systems. This is due to the reliance on unstructured, ambiguous "free text" within the operational C2 messages that are passed within the C4ISR systems, which is difficult or impossible to use as input to a computer program.

The Free Text Problem

Methods of communicating command and control information on the battlefield have evolved from simple face to face spoken language to the highly computer driven, automated systems of today. The evolution will continue as the Army transforms to the Future Force. Likewise, modeling and simulation (M&S) have advanced from board games to the very sophisticated computer driven simulations of today. Throughout the evolution, there has been a growing effort to make M&S and C4ISR systems interoperable. As we have indicated, several challenges must be overcome to make this happen. While significant progress has been made in meeting these challenges, the barrier of accounting for free text within our current message formats has not been resolved effectively.

Free text existing in United States Military Text Format (USMTF), Joint Variable Message Format (JVMF), and other message formats exists for the benefit of the human. The highly trained, professional soldier has little problem dealing with this free text. Current automated systems that deal with free text handle it as a single data field and pass the <character string> on. Understanding the content and context of the <character string> does not exist within the system. The introduction of "intelligent agents", "command entities," or other Command Decision Modeling (CDM) types of software invalidates the assumption that a human must process free text. CDM software is replacing humans within simulation systems and there is a need to communicate command and control information to the CDM applications. C4ISR systems also are evolving. The future systems are incorporating automated decision aids, such as course of action development and analysis tools, and mission rehearsal simulations. While some extant and evolving systems, such as the Force XXI Battle Command Brigade and Below (FBCB2), provide AutoFill of certain fields in their orders messages, this is primarily situational awareness information (e.g. time, location, etc.); command information is still carried in free text form.

If free text in our message systems is the problem, then replacing free text with structured text is a way to solve the problem. This was the approach that the EAGLE [2] Battle Management Language (BML) and the Command and Control Simulation Interface Language (CCSIL) [3] took. Both of these languages replaced semi-structured official message formats containing free text fields with highly structured messages containing clearly defined data fields. Communication between a C4I device and an "intelligent agent" within EAGLE or the Command Forces (CFOR) application still requires a human to translate the free text portion of the C4I message into the highly structured CCSIL or EAGLE format.

The Need for a BML

Note that the efforts to solve the free text challenge mentioned so far have all been initiated from the simulation side of the interoperability problem. Based on experiences dealing with CFOR, CCSIL, and the Modular, Reconfigurable C4I Interface (MRCI), it was recognized that this would not solve the problem. The doctrine, training, simulation and C4I development communities need to embark on a joint effort to resolve the problem. [4] This effort now is beginning.

With the increased complexity and numbers of interoperating Battle Command systems it becomes evermore difficult to provide an environment where commanders and staffs can follow the maxim "Train as you fight, because you will fight as you train" with regard to their Battle Command systems. In order to accomplish this it is increasingly necessary to provide high fidelity stimulation to these systems through the use of simulations responding directly to orders produced and disseminated using operational Battle Command systems. Without resolving the free text problem this cannot be accomplished; this situation gives rise to the need for an operational Battle Management Language.

BATTLE MANAGEMENT LANGUAGE

Defining BML
Taking the widest possible interpretation, BML [5] is defined as:

BML is the unambiguous language used to command and control forces and equipment conducting military operations and to provide for situational awareness and a shared, common operational picture.

Four principles guide the discussion of BML though the paper:

- BML must be unambiguous.
- BML must allow full expression of a commander's intent.
- BML must use the existing C4I data representations when possible.
- BML must allow all elements to communicate information pertaining to themselves, their mission and their environment in order to create situational awareness and a shared, common operational picture.

Clearly, the first two principles are difficult to reconcile. Principle three is one that is often ignored or slighted. However, if a "new" representation is to be developed, then it will still have to be "translated" into the organic C4I infrastructure. Thus, many advanced and flexible planning representations, while very well suited to BML, are not appropriate due to integration difficulties.

Previous efforts to support simulations have raised the question "Is there an operational BML?" The answer is yes. It is the language used on a daily basis by military professionals to command and control live forces. Doctrinal manuals such as Field Manual (FM) 101-5-1 [6] (future FM 1-02) define the vocabulary. The associated grammar is defined by other doctrinal manuals and from years of use. It is tailored to interpersonal communications and doctrine provides the baseline of common understanding among all users. However, operational BML lacks clearly delineated rules governing its use (semantics and syntax) and is riddled with ambiguity. It works because the military professionals who use it grow up with it from the moment they enter the service. They learn its idiosyncrasies as they learn the idiosyncrasies of the individuals who use it. When a term is used, it has context based on the operation, unit type and echelon, and individual characteristics of the sender. Likewise, when a sender selects a term to use he does so with an understanding of these same characteristics of the intended audience. Any confusion is resolved through question and answer between sender and receiver. Mentoring and coaching is a part of the process of learning the "informal" BML. While ease of use is this operational language's main strength, it is directly related to its main weakness in relation to automated systems, specifically, lack of structure. As such, it is incapable of supporting the full range of automation that the Army is implementing. It demands further development and modification.

As emerging and future simulations are developed, there are three options for meeting the requirement of BML. First is to maintain the status quo and create BMLs that are specific to each simulation. Second is to develop a BML that is standard within the simulation community and create interpreters between it and the C4I systems. Third is to develop a BML that is standard within both the simulation and C4I domains. To support the "Train as you fight, because you will fight as you train" principles, it is recommended to develop a BML that is standard within both domains.

The BML Concept
The BML as has been defined blends structure that allows automation of the language and ease of use for the military professional. It is not a radical change from the language the commander and staff currently use, but instead an evolution that provides a means to increase structure while remaining transparent to the user. It is based on doctrine and linked to the doctrinal sources, both to ensure standard use/understanding, and to foster concise and precise use of the language. The formal BML must support the "train as you fight" concept and therefore must exist in a single format, at least as far as the military professional user is concerned. The output of the automated system is allowed to fluctuate based on whether the intended audience is a human, an "intelligent agent" or a robotic element of a Future Combat System (FCS). The scope of BML is depicted in Figure 1.

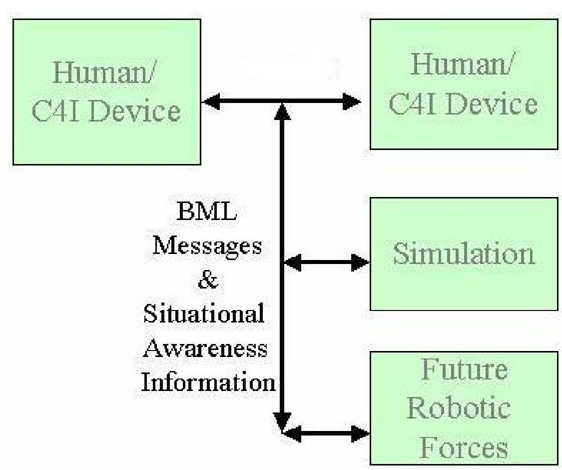

Figure 1: Scope of BML

The initial development of BML for the Army focused on adding the required structure for BML into the Joint Common Data Base (JCDB). [7] The JCDB focuses on two macro areas: first are the physical elements of the battlefield, which are referred to as objects; and second, the data for employment (actions) of the elements (objects) of the battlefield.

The physical elements of the battlefield fall into five general categories: Person, Organization, Materiel, Feature, and Facility. Each has subcategories and descriptive values that are capable of defining unique, individual entities. Using these descriptive values plus the series of standard (or unique) relationships defined among the elements we can describe a thorough "picture" of the battlefield. The picture can be shared among Battle Command systems and, with some work, among the supporting simulations as well. As mentioned earlier, this is significant progress in terms of sharing data and provides to a great degree the Situational Awareness function of a BML. (Note that BML is referred to as facilitating Situational Awareness because it provides for the passing of information. The term Situational Understanding is not used as defined in FM 3-0, Operations [8], since understanding implies cognitive analysis which is a step beyond just the passing of the information.)

The employment aspects within the JCDB are categorized as Situation, Plan, Action, Location and Capability. Each of these "items" also has sub-items and possesses descriptive values. Currently, however, these categories and their relationships do not provide sufficient structure to solve the free text issue, but they do provide a point of departure for further development. A significant portion of the BML project is to extend these areas and their relationships and the JCDB's categorization of the physical elements of the battlefield, to solve the free text problem in consonance with the four principles specified above. To address this disparity the approach is to:

- Build in the vocabulary as contained in FM 101-5-1, Operational Terms and Graphics, as BML data tables.
- Incorporate the doctrinal base into the JCDB.
- Build in the syntax and semantics defined by FM 7-15, The Army Universal Task List (AUTL) [9], the Army Training and Evaluation Program (ARTEP) Mission Training Plans (MTP) and other related field manuals. Doing this allows specific associations between unit types and echelons to be captured as relationships in the data tables.
- Create data oriented messages that eliminate or reduce the free text currently in use.

The doctrinal base provides the philosophical underpinning for the BML, as well as a significant amount of the detail necessary to achieve the stated objectives of the language. As it stands, however, the base needs to tie directly to the AUTL and the ARTEP-MTP to be clear, concise and explicit. The AUTL is developed in conjunction with the Universal Joint Task List (UJTL) [10] and attaches an identifier to various functions, missions, activities, and tasks that the Army, as a military Service, can perform. The tasks are then supported by the ARTEP-MTPs, which describe the tasks in detail along with the conditions and standards for a given type unit at a given echelon. These in turn decompose all the way town through the platoon, squad and team level to individual soldiers, identified by Military Occupational Specialty (MOS) and skill level. When these tasks are associated with the specific vocabulary and unit data contained in the (objective) JCDB it provides the context required to give meaning to the particular use of the terms. BML now has a vocabulary as well as the required syntax and semantics.

BML Technical Approach

A standard format for capturing an Operations Order in BML format can be as simple as a matrix assigning the Who, What, When, Where and Why (the "Five W's") for each subordinate element that is receiving a mission as well as the information needed to coordinate activities. Within the JCDB, tables were identified that were best aligned to represent the Five W's. The ORGANIZATION table provides the "Who." Its relationship to the ORGANIZATION-TYPE table associates the ORGANIZATION-TYPE function and echelon codes to specific organizations. The "What" is provided through the TASK table. TASKS, a directed activity, and EVENTS, a significant occurrence, are categories of ACTIONS, an activity. The ORGANIZATION-TASK table provides the association of tasks to specific organizations based on the organization's function and echelon. Attributes of the TASK table provide the "When" and the "Why." The ACTION-LOCATION table provides the "Where." Numerous other tables exist within the JCDB that contain enumerations that portray information required to coordinate activities such as the WEAPONS-CONTROL-CODE table. This subset of JCDB tables reflects a capability within the JCDB that was used to begin to establish the data and relationships required for BML implementation.

Using set-theoretic notation, BML can take the attributes (Tasks, Units, etc.) for all of the forces within a tactical organization as they would reside within a modified and expanded JCDB and can create a superset and appropriate subsets of the Five W's that are required to provide for the representation of operations orders. [1]

The What_Cd (What Code) is defined as the set of all possible tasks that can be assigned to military forces as defined by doctrinal manuals. A task, T, may be an operation as defined by the UJTL (attack, defend, etc.), a tactical task as defined by FM 101-5-1 (secure, clear, seize, etc.), or an ARTEP-MTP task (conduct tactical movement, conduct tactical road march, occupy an assembly area, etc.).

$$What\_Cd = \{T_1, T_2, \ldots T_i\}.$$

$What\_Cd_{Unit\text{-}type\_echelon}$ is the set of all possible tasks that can be assigned to a Unit of a specific Unit-Type and Echelon as defined by unit-type and echelon specific doctrinal manuals. $What\_Cd_{Unit\text{-}type\_echelon} = \{T_{u1}, T_{u2}, \ldots T_{uj}\}$, where j is the total number of applicable tasks for unit-type and echelon u and $\{T_{u1}, T_{u2}, \ldots T_{uj}\}$ is a subset of $\{T_1, T_2, \ldots T_i\}$ (where T_{u1} is identical to one of the tasks in $\{T_1, T_2, \ldots T_i\}$).

As shown below a task (T_1) might be common to multiple Unit-Types and Echelons. For example the task attack might be common to a Tank/Mech brigade and battalion as well as to a field artillery battalion and an Air Force F-15 Squadron. Other tasks will be unique to a specific unit-type and echelon.

$$What\_Cd_{Tank/Mech\_BDE} = \{T_1, T_5, T_8, T_{19}, T_{24}, T_{30}\}$$
$$What\_Cd_{Tank/Mech\_BN} = \{T_1, T_5, T_8, T_{32}, T_{40}, T_{41}\}$$
$$What\_Cd_{FA\_BN} = \{T_1, T_5, T_{50}, T_{51}, T_{52}, T_{53}\}$$
$$What\_Cd_{F15\_SQ} = \{T_1, T_5, T_{100}, T_{101}, T_{102}, T_{103}\}$$

Each unit has the properties of Unit-Type and Echelon: $Unit_{Unit\text{-}Type\_Echelon}$. $Unit_{Unit\text{-}Type\_Echelon}$ is associated with $What\_Cd_{Unit\text{-}type\_echelon}$ establishing a Who-What relationship.

Why_Cd (Why Code) is the set of all possible purposes for conducting the tasks that are elements of What_Cd. The purposes, P, have been identified in the same doctrinal manuals used to identify the tasks, T. The terms selected convey a reason for conducting a task and in many cases the definition of the term defines an endstate condition. Examples of Why codes for a What_Cd=Attack are “Seize”, “Destroy”, and “Defeat”.

$$Why\_Cd = \{P_1, P_2, \ldots P_n\}.$$

For any given What_Cd element, T_x, where 1<=x<= i; there is a corresponding subset of Why_Cd elements associated with it. $Why\_Cd(T_x) = \{P_{x1}, P_{x2}, \ldots P_{xm}\}$, where m is the total number of applicable purposes for Task T_x and $\{P_{x1}, P_{x2}, \ldots P_{xm}\}$ is a subset of $\{P_1, P_2, \ldots P_n\}$ and P_{x1} is identical to one of the purposes in $\{P_1, P_2, \ldots P_n\}$.

Therefore, a mission or tasking statement that is defined by the Five W’s (Who, What, When, Where and Why) is structured as a finite set of possibilities by these set associations. Given a Who = $Unit_{Unit\text{-}Type\_Echelon}$ (1 BN 40 AR is the name of a unit where Unit-type = Tank/Mech and Echelon = Battalion.), and an association to a $What\_Cd_{Unit\text{-}type\_echelon}$; then selecting a task, T_x, establishes the relationship to $Why\_Cd(T_x)$ allowing us to now select a purpose, P, associated with why we want to do a task, T_x, for this specific instance.

This allows encoding into the database the relationships and associations to enable the unambiguous identification of a task to a unit based on that unit’s type and echelon of employment in the Five W’s format. Specification of tasks in this format allows the automated construction and implementation of orders and taskings that can be communicated to a software

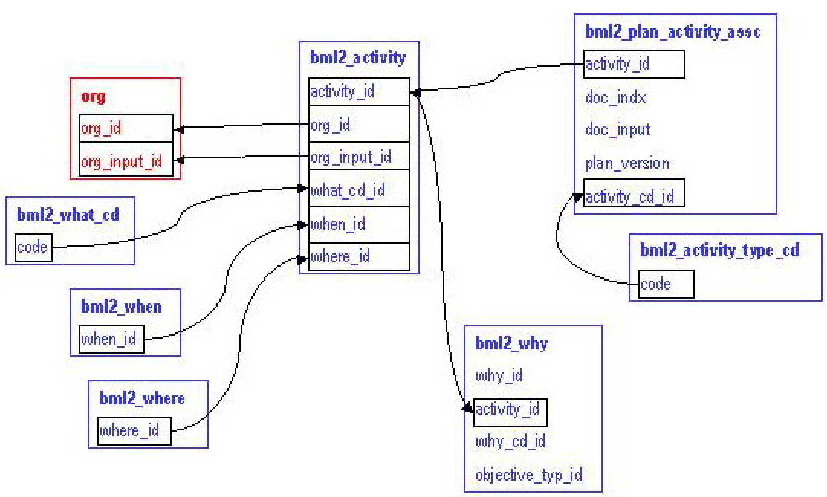

Figure 2: Five W’s Data Association

application in a simulation application. Figure 2 shows the high-level database representation of the Five W's based on a plan (the mission) and an activity (a specific task). [11]

The Army BML Proof of Principle

The preceding sections have defined BML and the concepts necessary to communicate clear and unambiguous mission orders from a human, supported by an automated C2 system, to a subordinate. To demonstrate the concepts described above, a Proof of Principle (PoP) prototype demonstration was developed. To build the prototype the problem had to be bounded. To do this, several factors were applied.

1. Select a scenario that was grounded in reality as much as possible, in that it was developed for realistic training and not specifically to demonstrate the proof of principle.
2. Use BML to link a C4ISR application to a simulation in a manner consistent with doctrine.
3. The scenario needed to be usable by both the simulation and the C4ISR application, which implied the need for a common terrain database and common units and tasks.
4. The scope of units and tasks associated with those units had to be manageable and documented in doctrine.
5. Demonstrate the capability to eliminate free text using the Operations Order/Plan since this is the most challenging of any message and where most free text currently is being employed with the current message formats, the USMTF and JVMF.

To satisfy these factors, an operations order was created based on a real world scenario supporting a Battle Command Training Program Warfighter exercise. This scenario provided a heavy brigade level organization, with its associated supporting units, to base our work and in turn build the respective ARTEP information into the database. The scenario was set in the National Training Center (NTC) to ensure a common terrain database between the Combined Arms Planning and Execution System (CAPES) [12], the chosen C4ISR application, and the OneSAF Testbed Baseline (OTB) [13] simulation. Both of these systems also supported the heavy force organizations and tasks. The entire heavy brigade definition of units and task associations were built into the database even though the goal of the PoP was to show the ability to task and execute the tank and mechanized companies and the scout platoon between CAPES and OTB. The brigade unit and its associated slice elements all have well documented doctrinal manuals to include ARTEP-MTPs.

The configuration of the complete BML PoP demonstration environment is comprised of the following components:

- The **Combined Arms Planning and Execution System** (CAPES). CAPES is a prototype US Army Planning System and is used to develop a Course of Action and generate a plan. This C4ISR component creates operational orders (Opords) that are exchanged using a proprietary tagged Extensible Markup Language (XML) document.

- A **Multi-Source Data Base** (MSDB). The MSDB is based on the U.S. Army Standard data model of the JCDB, which has been extended by the BML development team by introducing over 100 new tables and relations. It is accessed via standardized Structured Query Language (SQL) database manipulation statements based on Open Data Base Connectivity (ODBC) or Java Data Base Connectivity (JDBC). It is implemented in open source software of the Linux environment.

- A BML Demonstrator specific XML-BML Parser. The XML-BML parser reads the information from the CAPES XML document, maps the information to data elements of the MSDB, and inserts the information contained in the document into the MSDB.

- A BML Graphical User Interface (BML GUI). The BML GUI allows viewing and editing of the content of the MSDB under consideration of the semantic and syntactic constraints of the BML. The input of CAPES can be used as a basis to create more detailed operational orders for the subordinate units (which are simulated using the OTB simulation system). The BML GUI interfaces to the MSDB via JCDB SQL calls.

- The C4I Simulation Interface (C4ISI). The MSDB information is based on the U.S. Army doctrinal language. In order to execute such orders, this information has to be mapped from doctrinal terms to OTB interpretable terms. This is done by the C4ISI, which reads the MSDB and generates task commands for OTB.

- The OneSAF Testbed Baseline (OTB). The M&S component of the PoP, OTB is used to simulate the effect of the generated orders. It reads the orders generated by C4ISI and executes them respectively.

Figure 3 shows a block diagram configuration of all of the components of the Army BML PoP demonstrator. PoP demonstrations were conducted for senior Army leadership and generated significant enthusiasm and endorsement for further development of the BML concepts.

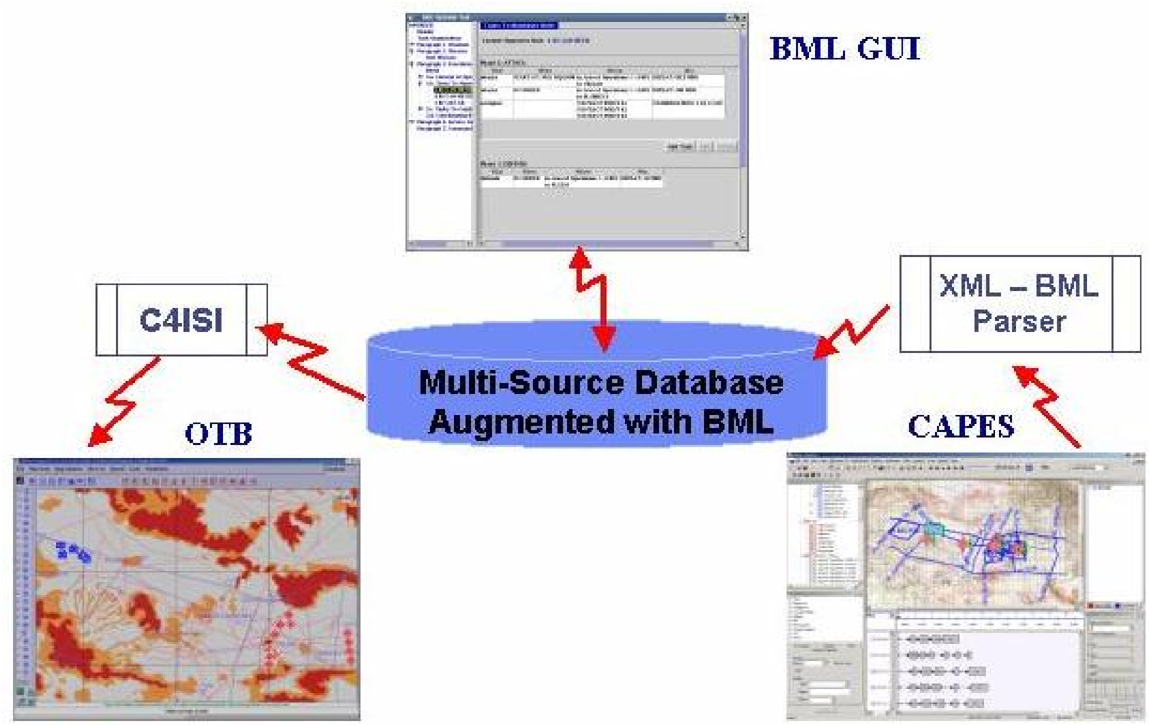

Figure 3: BML PoP Configuration

While building the scenario into BML for the PoP it was quickly discovered that the JCDB, while capturing the physical attributes of the battlefield extremely well, was not sufficient for fully representing the required relationships, associations and enumerations necessary to implement BML. Additional tables were added to support the implementation of BML and the interface to the CAPES application. This extended JCDB was named the MSDB for the PoP to avoid confusion and prevent configuration issues with the tactical JCDB. Additionally, the BML GUI application was developed to allow an operator to view and edit the CAPES generated information within the scope of an operations order.

Figure 4 shows an editing screen from the BML GUI that highlights the Five W representation and the linkage to doctrinal definitions of the "What" code.

Since neither CAPES nor OTB has an understanding of the BML structure embedded in their application, mechanisms for interfacing with the BML representation in the MSDB had to be developed. CAPES generates an Extensible Markup Language (XML) file containing all of the elements of the plan data that are developed in a Course of Action Planning session using that application. The XML parser was developed to parse the CAPES XML file into the appropriate data tables and fields in the MSDB. The ODBC Application Programmer Interface (API) was used to provide the mechanism for interfacing SQL calls to the MSDB. For the MSDB to OTB interface, a C4I Simulation Interface (C4ISI) application was developed. The C4ISI also used ODBC and SQL to access the MSDB. The C4ISI provided the middleware to map the structured BML to task frame primitives that are executed in OTB. For example, a doctrinal task in the BML for a Scout Platoon unit was a "Screen" to achieve some result at a location. OTB does not understand a Screen command so the C4ISI mapped that to a 'Move' with the proper attributes to accomplish the mission identified in BML.

TRANSFORMATION

It is widely recognized that U.S. military forces today face a mandate for change. News media frequently call attention to changes in the international situation that have resulted in a much greater variety of threats, demanding a response capability that is flexible and multi-faceted. In particular, the challenges of rogue nations and terrorism have given military force projection major new dimensions while also increasing the need for operations other than war (OOTW) by the military. The Department of Defense today is engaged in transforming itself to meet these challenges. The Army, Navy, Marine Corps and Air Force of the future will be staffed, trained, and equipped very differently than they have been in the past. They will habitually work together as a joint force

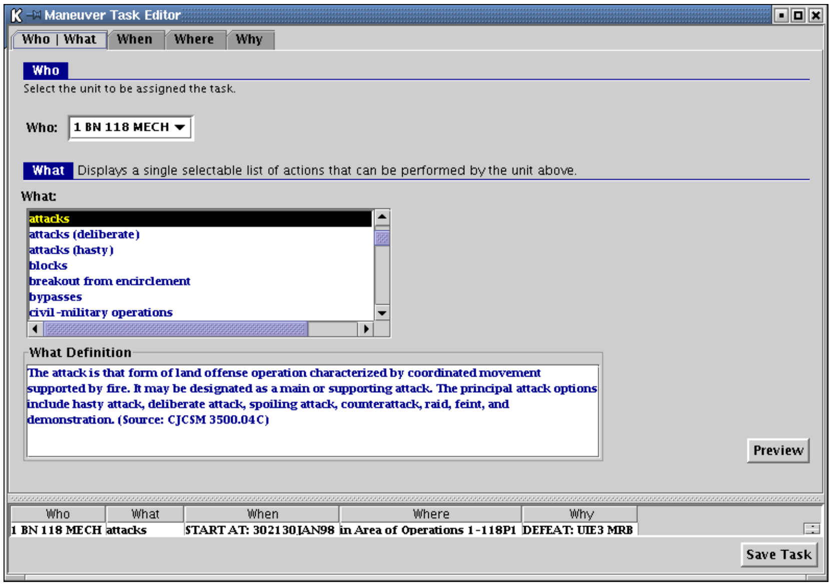

Figure 4: 5 W Task Editor in the BML GUI

and will be challenged to sustain an expeditionary mindset associated with frequent deployment under constantly changing conditions.

Technology plays a major role in this transformation. The ability to call on autonomous, robotic platforms to avoid sending humans in harm's way is but one major facet of the new face of warfare. The new focus on network-centric warfare means the forces will be able to function in a highly coordinated way while maintaining physical distribution that economizes effort and avoids presenting lucrative targets. This amounts to a major new thrust in C4ISR and, combined with the use of robotic weapon systems, increases the premium on effective coupling of all aspects of the force with computers and communications.

The C4ISR support establishment is gearing up to create the Global Information Grid (GIG) and populate it with network-centric enterprise services. The technical approach being followed is one that originated in the commercial sector, where the Internet and the World-Wide-Web have far surpassed their original roots in defense. Today the Web is being expanded from a service that delivers multimedia over the Internet to a worldwide infrastructure for networked interoperation of software systems, using techniques that recently have acquired the collective name "Web services." These consist of a logically complete suite of information transfer protocols that support communication among geographically distributed software processes. The techniques involved support composition of the services and development of solutions far more rapidly than previous generations of software technology. Moreover, they are based in open standards and therefore offer a way to overcome both the prevalence of "stovepipe" C4ISR systems that do not interoperate, and the vendor lock-in associated with dependence on such systems.

A technical partnership was formed in 2002 to find and encourage Web-based methods that can bring this sort of transformation to the modeling and simulation area in defense. The result was a report describing the Extensible Modeling and Simulation Framework (XMSF). [14] XMSF has attracted considerable attention because it offers not only an approach to reduce time and cost of developing interoperable distributed simulations but also a path for achieving the long-desired linkage between simulation and C4ISR systems, using Web services. The Defense Modeling and Simulation Office (DMSO) is supporting extension of the BML project as an early exemplar of the benefits of the XMSF approach.

EXTENSIBLE BATTLE MANAGEMENT LANGUAGE

XMSF C4I Testbed

The primary goal of the Extensible Battle Management Language (XBML) Phase I project was to use the existing Army PoP demonstrator and apply the Web based standards identified in the concepts set forth in the XMSF report. Leveraging the existing Army BML PoP demonstrator facilitated the creation of a C4I testbed under the auspices of the XMSF. It was straightforward to determine which XMSF standards would be applied first. In this section we describe the Phase I XBML effort.

Each of the interfaces between the components comprising the original Army BML PoP demonstration environment were investigated to determine approaches to incorporate open, Web-based methods to implement the interfaces to conform with the principles of XMSF. Two Web-based standards that were chosen to implement the interfaces to construct the XBML prototype were XML [15] and the Simple Object Access Protocol (SOAP) [16]. XML was used to provide a standard way to structure the data passed between the components. SOAP was used to package the ODBC and JDBC calls to the MSDB database. Figure 5 shows a detailed view of the components configured for the Phase I XBML.

responses, including the use of HTTP as the underlying transport mechanism.

The plan output from the CAPES application was already constructed in an XML format. The CAPES XML plan however, was in a format developed earlier in the CAPES program and did not conform to the BML schema. A parser was written for the Army PoP demonstration that mapped the CAPES XML plan format into the BML schema and used ODBC functions to store the data in the MSDB. The XBML project used this parser to accomplish the mapping between attribute tags for the construction of the BML but implemented a SOAP transport mechanism between CAPES and the MSDB.

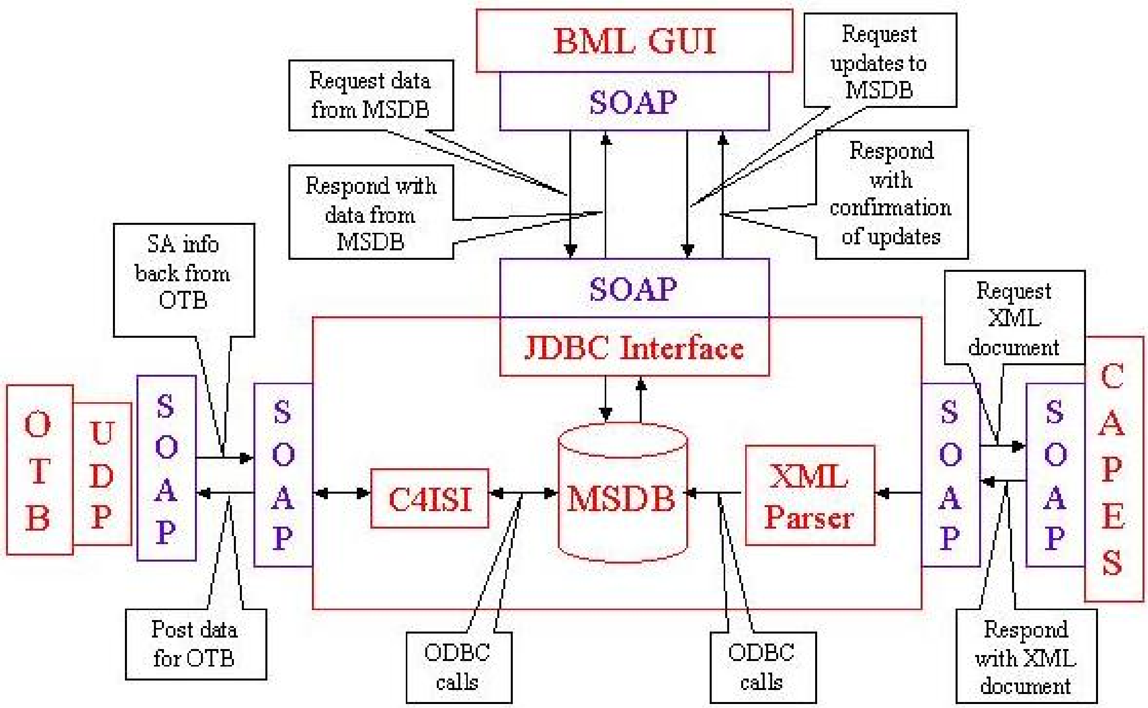

Figure 5: XBML Phase I Configuration

SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. A SOAP message is constructed from a number of elements. The first of these is called the envelope and is used to describe the basics of the SOAP message such as the content and how to process the message. The second part consists of the rules about custom data types. The ability to create custom data types, a feature also of XML, is an important characteristic of SOAP lending to its ability to support extensibility. The last part of the SOAP message describes the application of the envelope and the data encoding rules for representing the Remote Procedure Call (RPC) calls and

For the BML PoP, the interface between the BML GUI and the MSDB was implemented using the standard PostgreSQL JDBC drivers to communicate with the MSDB. To bring this interface into conformance with the XMSF concepts, the JCDB interface was embedded in a SOAP message. The SOAP-based driver works in much the same way as a standard JDBC driver. It implements the standard Java interfaces that any JDBC driver would implement. The main difference is how the driver works. A standard JDBC driver would accept the API calls then use its own transport protocol and API to access the database directly. The SOAP-based driver that was developed is split into two pieces. On the client side is the JDBC driver, this is the class called by the BML GUI and looks like any other JDBC driver. The second part is the

JDBC web service that resides on the same machine as the database being accessed and is never directly called by the client. The web service uses the standard PostgreSQL driver to communicate with the database. It takes the information returned by the PostgreSQL driver, turns it into an XML document and sends it back to the client via SOAP. The client code turns the XML document back into objects that are then accessible via the normal JDBC API.

This approach was chosen for two reasons. First, it allowed us to make the BML GUI SOAP-based without actually having to modify any of the original GUI code, only new code to implement the SOAP elements had to be constructed. Second, this same driver could be re-used by any Java application to access a database accessible via the Internet. In fact, by using the JDBC-ODBC bridge driver provided by Sun Microsystems in the web service, we could now make any database accessible via an ODBC driver accessible over the Internet via our SOAP-based JDBC driver.

Figure 6 shows the SOAP envelope sent as a return message for a data request from the BML GUI to the MSDB. The SOAP-ENV:Envelope tag contains header information about the SOAP protocol. The SOAP-ENV:Body tag indicates that this interaction is the response to a Java Database Connectivity query. The return tag defines the number of database columns and the column headings that will be returned as the query result. This is followed by the actual data. In this example, identification of the higher headquarters Operations Orders that are referenced for the BML GUI Header Page are being returned for insertion into the new document. This data is placed into Java objects that are then used in coding the GUI screen design.

The C4ISI reads BML data from the MSDB and generates task command frames for OTB. In the PoP demonstration, ODBC was used to access the MSDB to get the BML information necessary for generating the tasking orders to send to OTB. The C4ISI creates messages to send to OTB to create friendly and enemy units, place those units on a battlefield, create battlefield features, and create friendly unit activities. The C4ISI sends these messages using the standard User Datagram Protocol (UDP) network interface that is commonly used with OTB. Since OTB could not be modified to use SOAP in place of UDP, a Java and a C++ module were created to run on the same machine as OTB, implementing an interface to SOAP while maintaining the UDP network interface. The C4ISI Core was modified with two Java modules that implemented an interface to SOAP, a C++ module that gets the data from the plan object and gives it to a Java module, and a C++ module that gets the data from a Java module and sends it to OTB via UDP. The Java Native Interface (JNI) was used to invoke the interactions between the C++ and Java code modules.

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<ns1:executeQueryResponse xmlns:ns1="urn:JdbcWebService" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<return xsi:type="ns1:Result"><result numcolumns="3" columnnames="doc_indx doc_descr_txt
doc_dttm"><rows><row>
<colval class="java.lang.Integer">23</colval>
<colval class="java.lang.String">HQ 1ID(M) OPORD 9803</colval>
<colval class="java.sql.Timestamp">2004-01-27 00:00:00.0</colval>
</row><row>
<colval class="java.lang.Integer">25</colval>
<colval class="java.lang.String">HQ X Corps OPORD 9803</colval>
<colval class="java.sql.Timestamp">2004-01-27 00:00:00.0</colval>
</row></rows></result></return>
</ns1:executeQueryResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 6: SOAP Envelope Example

XBML Phase I Result

The results of Phase 1 XBML effort were demonstrated at the 2003 Interservice/Industry Training, Simulation and Education Conference (I/ITSEC). Figure 7 shows the demonstration configuration, which had three nodes, two on the I/ITSEC show floor at the Orlando Conference Center and one node at George Mason University in Fairfax, Virginia. The nodes at I/ITSEC were physically on the I/ITSEC network and communicated to the GMU node through the commercial Internet. The implementation of XML and SOAP in accordance with XMSF principles among the components of the XBML demonstration environment allowed seamless, distributed and remote execution of each of the components. The demonstrations conducted during the I/ITSEC laid the groundwork for extending the XMSF concepts to meet the challenges of future tactical Command and Control requirements.

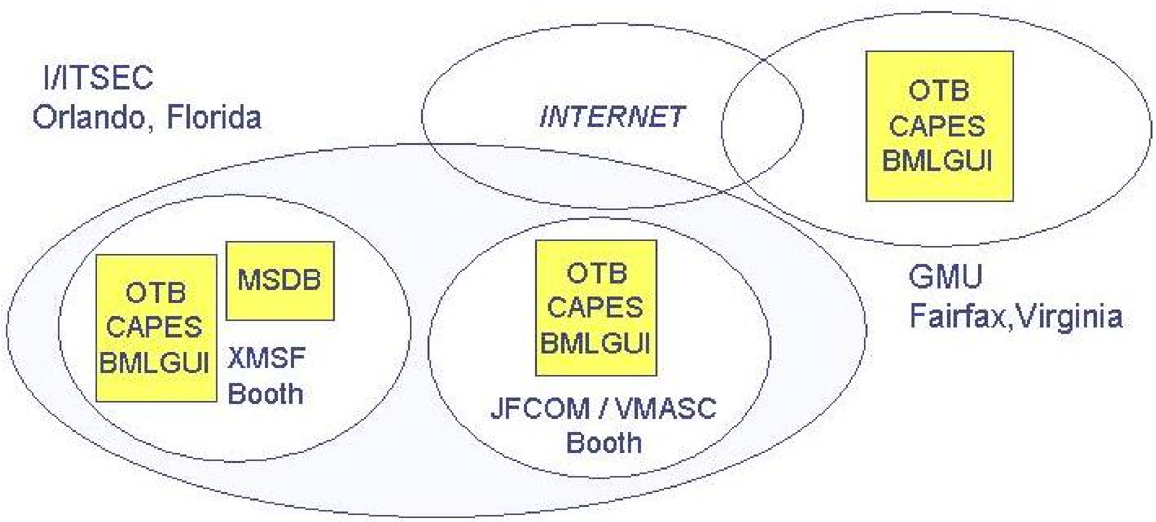

Figure 7: I/ITSEC Demonstration Architecture

Future Efforts

When examining interoperability issues concerning the different military Services, or between nations in a coalition, it is common to discuss issues affecting their working together. It is not common to discuss their C4ISR systems being interoperable with simulations. Since the early 1990's the U.S. Services have undertaken an extensive revision of Service and Joint doctrine to first document and then to standardize and align their doctrine with a goal of improving interoperability in training and execution. In September 2004, the Simulation Interoperability Standards Organization (SISO) initiated a Study Group to evaluate necessary and applicable standards for a Coalition Battle Management Language. A related process is occurring within the formal alliance of NATO which has the NATO Standardization Organization (NSO) whose role is to "enhance interoperability of Alliance forces to train, exercise and operate effectively together, and when appropriate, with forces of Partner and other nations, in the execution of their assigned tasks." [17] As a start toward such a capability, the NATO Research and Technology (RTO) Modeling & Simulation Group (MSG) agreed at its meeting in October, 2004 to establish the Exploratory Team ET-016 to evaluate the applicability of the ideas of a common Battle Management Language.

Motivation for these efforts comes from the rapidly changing nature of warfare, as described in section 3 above. It is almost impossible to imagine a situation in the future when a single U. S. Service will be unilaterally employed. Because future military operations, and a significant amount of training, will be Joint in nature, it is critical that a Joint Service approach be taken to the BML development effort. The same issues that have driven the Army to embark on this program also confront the other Services as they develop both their C4ISR and simulation systems. While the other Services may have related efforts under consideration, it would be valuable to harmonize and standardize these endeavors. Forgoing this, we will face the same set of disparate solutions that will mirror our current intra-Service interoperability problems. A parallel situation exists in NATO and other military coalitions.

The first phase of XBML was relatively straightforward, and set the stage for the following phase, which increases focus on the semantic aspects of developing an extensible BML. Work was performed in Phase I to map the MSDB BML tables to the NATO Standard C2 Data Model – the Command and Control Information Exchange Data Model (C2IEDM) [18]. The next phase of XBML is transforming the MSDB to be based on the C2IEDM. This will take the PoP from an US Army basis to a Joint/Coalition basis in keeping with the XBML project goals. Future effort will also create XBML extensions to a standard XML tagset based on the C2IEDM [19].

The C2IEDM is a data model that provides a common specification and structuring of information in order to achieve automated information exchange. The scope of the analysis carried out in the development of the C2IEDM has been principally directed at producing a corporate view of the data that reflects the multinational information exchange requirements for multiple echelons in land-based wartime operations and crisis response operations (CRO) to include joint interfaces that support land operations. The data model is focused primarily on the information requirements that support the operations planning and execution activities of a military headquarters or a command post. Just as the Five W's were represented in the JCDB, we anticipate that the same concept for developing BML will also work for the other Services, Joint and Combined/Coalition operations using the C2IEDM as a basis.

The C2IEDM has great potential for bridging the longstanding gap among C4ISR systems. Moreover, it provides a natural common language not only among those systems but also among military simulations and between simulation and C4ISR systems. The XBML project therefore has enthusiastically embraced the opportunity to demonstrate the potential of C2IEDM as a basis for information exchange over Web services.

CONCLUSIONS

Interoperability between C4ISR systems and simulations is critical to the Transformation efforts of the U.S. Department of Defense. Operational forces must use their C4ISR systems to interact with supporting simulations to conduct realistic, rigorous training, support mission rehearsals and, in the future, support an expedited military decision making process. Achieving interoperability depends on several factors including alignment of the architectures, common standards, common data/object models, reusable component interfaces, and shared solutions. While extremely important, none of these solve the free text challenge. A well thought out and implemented BML as demonstrated through this proof of principle prototype allows communication of command and control information in a format that can be read, parsed, interpreted and acted upon by software whether in a simulation or a future concept autonomous platform or robotic system.

Extending these concepts to open Web-based interfaces embraces a new architectural vision. BML is an enabling component to support the emerging possibilities created by the Global Information Grid (GIG) under the tenets of Network Centric Operations and Warfare. Expanded to encompass the C2IEDM, BML can serve as the basis for data interoperation among C4ISR systems and simulations. This in turn will allow greatly improved effectiveness in use of simulations for military operations and training.

REFERENCES

[1] Northrop Grumman Information Technology, "Battle Management Language Proof of Principle Prototype Demonstration Final Technical Report", Contract Number N61339-00-D-0708/0005, CDRL A003, February 2003.

[2] Ogren, J., and Fraka, M., "EAGLE Combat Model Battle Management Language (BML)", Powerpoint presentation, BML Symposium at Fort Leavenworth, KS, 25 April 2001. (http://www.simci.org/html/librsry.html)

[3] Salisbury, M., "Command and Control Simulation Interface Language (CCSIL): Status Update" MITRE Informal Report, Twelfth Workshop on Standards for the Interoperability of Defense Simulations, March 1995.

[4] Kleiner, M.S., Carey, S.A., and Beach, J., "Communicating Mission-Type Orders to Virtual Commanders", Paper, Proceedings of the 1998 Winter Simulation Conference, December 1998.

[5] Carey, S., Kleiner, M., Hieb, M. and Brown, R., "Standardizing Battle Management Language – A Vital Move Towards the Army Transformation", Paper 01F-SIW-067, Fall Simulation Interoperability Workshop, Orlando, FL, 2001.

[6] Field Manual 101-5-1, Operational Terms and Graphics, Headquarters Department of the Army, 31 May 1997.

[7] Carnevale, R., "Joint Common Database Overview," January 2001, Program Executive Office (PEO) Command, Control and Communications – Tactical (C3T) Knowledge Center.

[8] Field Manual 3-0, Operations, Headquarters Department of the Army, 14 June 2001.

[9] Field Manual 7-15, The Army Universal Task List, Headquarters Department of the Army, 31 August 2003.

[10] CJCSM 3500.04C, Universal Joint Task List, July 2002.

[11] Sudnikovich, W., Hieb, M.R., Kleiner, M. and Brown, R., "Developing the Army's Battle Management Language Prototype Environment", Paper 04S-SIW-115, 2004 Spring Simulation Interoperability Workshop, Crystal City, VA, 2004.

[12] Agile Commander Advanced Technology Demonstration, June 2001, http://www.c2d.c3sys.army.mil:443/agile.htm

[13] OneSAF Testbed Baseline web site, http://www.onesaf.org/onesafotb.html

[14] Brutzman, D., K. Morse, J. M. Pullen and M. Zyda, Extensible Modeling and Simulation Framework (XMSF): Challenges for Web-Based Modeling and Simulation, Naval Postgraduate School, Monterey, CA, 2002.

[15] Extensible Markup Language (XML), http://www.w3c.org/xml

[16] Simple Object Access Protocol (SOAP), http://www.w3c.org/soap

[17] Hieb, M.R., Sudnikovich, W.P., Tolk, A., and Pullen, J.M., "Developing Battle Management Language into a Web Service", Paper 04S-SIW-113, 2004 Spring Simulation Interoperability Workshop, Crystal City, VA, 2004.

[18] Multilateral Interoperability Programme (MIP): The C2 Information Exchange Data Model (C2IEDM). 20 November 2003.

[19] DoD XML Gallery, http://diides.ncr.disa.mil/xmlreg/user/index.cfm