

Coalition Battle Management Services (CBMS)

Dr. Saikou Y. Diallo
VMASC
Suffolk, VA
sdiallo@odu.edu

W. D. Scott Wood, Warren W. Bizub
JCS DD J7 JCW
Suffolk, VA
winston.wood@js.mil, warren.bizub@hr.js.mil

ABSTRACT

The Command and Control (C2) Community has a set of systems, standards and protocols that it uses to solve its main interoperability challenge which is to communicate a consistent common operational picture (COP) of the real battle space. Similarly, the modeling and simulation (M&S) community has evolved its own standards and protocols to deal with its main interoperability challenge which is to accurately model and simulate the synthetic battle space using one or more systems. However, as we move towards a system of system environment where live, virtual and constructive (LVC) systems are connected and the line between real and simulated entities is purposefully obscured, we are facing a new set of system of system interoperability challenges that cut across communities. In this new environment where information sharing is key, standardized metadata takes prevalence over standardized interfaces, which means that we need to look at standards as a set of interoperability enablers that must be coherently organized in a “system of standards.”

In this paper, we present the coalition battle management services (CBMS) as a system of system approach to dealing with system of system interoperability in general and C2 to simulation interoperability in particular. We distinguish between the *CBMS approach*, which is based on the observation that an interoperability system must be developed following interoperability requirements; the *CBMS architecture* based on the design pattern of the World Wide Web (WWW); and the *CBMS implementation* based on standards and open source tools. We describe how systems can use CBMS in support of interoperability and present a CBMS use case of C2 to simulation interoperability in a coalition environment using the Coalition Battle Management Language (C-BML) and the Military Scenario Definition Language (MSDL). Finally, we discuss the implications of CBMS on M&S standards and system of system interoperability.

ABOUT THE AUTHORS

Dr. Saikou Y. Diallo is a Research Assistant Professor at the Virginia Modeling, Analysis and Simulation Center at Old Dominion University. He received his M.S and Ph.D. in Modeling and Simulation from Old Dominion University. He leads the M&S Interoperability lab at VMASC and serves as the technical lead for the CBMS project. He is also the co-lead of the C-BML drafting group.

Mr. W. D. Scott Wood is the Live Systems Integration Lead for the Joint and Coalition Warfighting (JCW) facility – Environment Development – Futures Branch. As a DoD Civil Servant and a former Naval Reserve CAPTAIN, Scott has been affiliated with the JCW since February 1995 supporting Training Transformation; Joint Force Training; and C4I Technology Assessment. Significant assignments include TM4 LVC Technical Manager with JWFC Joint National Training Capability (JNTC); Deputy Strategy at JWFC; Deputy Assessments Joint C4ISR Battle Center (JBC); and Deputy Reserve Training Director J7 Directorate. He has a BS degree in Systems Engineering from the US Naval Academy.

Mr. Warren Bizub is the Program Director for Environment Development Division Futures Branch. He also served as the Program Director Joint Advanced Concepts, Technical Management Division Chief of the Joint Training Development Solutions Group and Director, Joint Advanced Training Technologies with USJFCOM. He received his BS degree in Ocean Engineering from Florida Institute of Technology.

Coalition Battle Management Services (CBMS)

Dr. Saikou Y. Diallo
VMASC
Suffolk, VA
sdiallo@odu.edu

W. D. Scott Wood, Warren W. Bizub
JCS DD J7 JCW
Suffolk, VA
Winston.wood@js.mil, warren.bizub@hr.js.mil

INTRODUCTION

As we move towards more coalition based engagements and we pivot towards increased collaboration with the Pacific region, the demand for coalition training and mission rehearsal at multiple levels of command (from strategic to tactical) will likely increase. Consequently, the ability to support federations where C2 devices, people and sensors, unmanned aerial vehicles (UAVs), virtual and constructive simulation systems are interoperating in a realistic and operationally relevant way will become more and more important. In order to support LVC federations in a coherent way, several challenges at the technical and managerial levels have to be addressed. At the technical level, the amount of information shared amongst participants will grow beyond simulated information to include not only messages such as tasks, orders, reports and requests but also images, voice and video. Potentially, the numbers of entities and messages that can be exchanged in such environments to effectively command, control and communicate for non-trivial scenarios can reach into the hundreds of thousands if not millions which will create an issue with scalability among others. In addition, there is a proliferation of models, formats, content and interfaces that could lead to a dramatic increase in gateways and translators depending on the type and number of systems involved. Furthermore, the problem of initializing and coordinating such a heterogeneous collection of systems needs to be addressed in a comprehensive way through standards at multiple levels. At the managerial level, the technical choices will directly impact the time and cost of developing, maintaining and reusing federations in general and LVC federations in particular.

In this paper, we propose a system of system approach to deal with these challenges. The approach is named Coalition Battle Management Services (CBMS) because it is based on the ideal of C-BML which is to seamlessly connect C2, simulations and robotic forces. We show how this approach can be mapped to a set of services that are used to support the interoperability of information systems and system of systems. We describe a CBMS implementation and discuss the

advantages of using web-based standards and the open source community to support interoperability in DoD environments.

PROBLEM STATEMENT

While there are many common definitions (IEEE, 1990), the idea of interoperability in M&S is to produce and share information by connecting existing models and simulations. In order to be useful, most standards simultaneously deal with the *technical* challenge of connecting the simulations, the *syntactic* challenge of consuming the information and the *semantic* challenge of using the information. This tight coupling of technical, syntactic and semantic interoperability is the root cause of many of today's interoperability problems because current standards prescribe how systems produce and consume data through a given interface. This interface-centric approach makes it difficult to transition from one technical solution to another because it often requires wholesale changes to systems that are technically interoperable through existing interfaces thus significantly disrupting federations at a high cost. It also accounts for the difficulty in adopting new standards and policies even when older standards are outdated. Simply stated, since each standard presents a rigid set of requirements and guidelines that adherents must follow in order to be interoperable with compliant systems, integrating new systems or moving to new standards is very difficult because of the amount of changes that have to be made at each level of interoperability.

In addition within the last ten years, there has been an attempt to develop and standardize a C-BML that will make C2 systems interoperable with simulation systems as well as robotic forces (Blais, Galvin & Hieb, 2010). C-BML focuses on capturing and describing orders, reports and requests using the eXtensible Markup Language (XML). The community has already begun to address the technical aspects of getting information from C2 system to simulation system in a coordinated way (Pullen, Corner & Singapogou, 2009). Furthermore, the use of XML to capture information is more and more prevalent in

M&S and M&S standards as more and more systems produce and consume XML. Therefore, the interoperability of XML-based systems needs to be addressed with a comprehensive approach that fits within a larger effort to support system of system interoperability.

By focusing on C2 to simulation interoperability, C-BML presents a new set of challenges to the simulation community. One of the challenges is related to the fact that C2 systems deal with real people and systems exchanging orders, requests and reports in near real time and typically focus on producing a common operational picture (COP). Constructive simulation systems on the other hand deal with the synthetic environment and use models/simplifications of the real world to simulate the real battlefield. C2 systems produce, exchange and consume messages. While standardized structured message formats (tactical messages) are used to support interoperability, the use of free text documents (Word, PowerPoint), chat and voice to carry information is very common. In contrast, constructive simulation systems deal exclusively with structured data where the combination of data and data structure is used to convey information and drive the execution of computer programs. Simulation interoperability standards provide the structure and interface descriptions to allow simulation systems to connect and interoperate amongst each other and live systems. However, we lack a comprehensive approach to deal with structured and unstructured information simultaneously without large scale changes to existing gateways, translators, object models and programs.

Another challenge is related to the semantics and pragmatics (interpretation) of information. This topic has been discussed and a layered approach (Wang, Tolk & Wang, 2009) as well as a grammatical approach have been proposed (Shade and Hieb, 2006). In C2 interoperability, semantics and pragmatics are expressed and aligned through shared doctrine, techniques, tactics and procedures or simply through dialogue when there are misunderstandings. In simulation interoperability this is a larger issue that is harder to solve since computers do not understand semantics inherently, which means that semantics have to be imposed. Conceptual modeling to help make the semantic of models more transparent is a recommended practice that is rarely followed. The current state of the art is to enforce a common data model with the agreed upon shared semantics of the systems but the data model falls short of expressing the pragmatics or how data should be processed and used. The pragmatics are represented in the form of algorithms (input, output, state transitions) and equations that are embedded within each system and the federation is managed

through a recommended system engineering process that guides when and how information is exchanged.

Because of these challenges, we need an interoperability approach, architecture and implementation that are specifically designed to support system of system interoperability. In the next section, we present the CBMS approach to dealing with the problem space described in this section.

CBMS SYSTEM AND ARCHITECTURE

Diallo, et.al. (2012) used Model Theory to mathematically derive a set of requirements for system of system interoperability. The authors assert that in order to support interoperability, a system must be able to:

- *Support multiple referents:* A referent is the real or imagined object that is being modeled. An interoperability system must be able to support any referent whether it is live, virtual or constructive and not create interfaces that only support specific referents.
- *Support multiple models:* The system must allow multiple models of the same referent or multiple models of multiple referents to be interoperable. This is a requirement for the system to support multiple syntaxes and data models and not to have specific interfaces that support only certain referents modeled in a specific context to interact in a specific way.
- *Support multiple valid models:* Without entering into a discussion about validity, this requirement is to ensure that the interoperability system does not internally interpret and/or translate information in order to conform to some internal representation. According to this requirement, the interoperability system should neither have its own version of what is true nor translate information to fit a data model for internal purposes. As discussed earlier, this is a requirement that is not followed by any of the existing interoperability standards.
- *Support multiple simulations:* The interoperability system must be reusable to support multiple federations without a specific configuration to accommodate federates. This requirement is not for federates which have to be configured depending on the federation; it is for the interoperability system to be independent of the federation it is supporting i.e., adding, removing or changing federates

does not affect the interoperability system. This is not the case today in most federations where the interoperability system must be adapted to the federation it is supporting. Adding or removing systems impacts federates but also the interoperability system.

- *Support multiple simulators:* The interoperability system must be able to support current and future systems regardless of how they are implemented. This requirement seems to call for a service-oriented approach but it is calling for a clear separation between the interoperability system's implementation and that of any potential system it will support. This is not the case today as most interoperability systems' implementation language determines what federate it can support.

Based on those requirements, *the CBMS approach for designing a standalone interoperability system (CBMS system) whose purpose is to enable other systems to be interoperable is to look at the World Wide Web (WWW) and how it is designed.* The WWW represents successful interoperability at work. Every day millions of users browse, exchange, create and share information from disparate sources with each other. The industry proven benefits include reduced cost, opening the aperture of developers contributing to solutions, higher quality, and accelerated technology innovation. The WWW is easy to use, adaptive, scalable, reliable and resilient. New ideas and technologies can be easily integrated. Legacy applications disappear when they become irrelevant. The WWW represents the interoperability vision sought within the military domain. DoD closed architectures and proprietary solutions limits the ability to provide the warfighter WWW application capabilities employed by industry and readily available in the open source community. In the next section, we discuss how we use the design of the WWW as a blueprint for architecting CBMS in order to fulfill the requirements discussed in this section

CBMS Architecture

In the case of C2 to Simulation interoperability, the principles and design of the WWW present very striking parallels. Simply stated, the WWW is designed as a system of system that share information. Everything around the WWW is designed to facilitate this transfer of information regardless of the medium (voice, text, video, etc.). *The WWW is independent of the systems that produce and/or consume the information and it is independent of the content of the*

information. If we look at the use case where a C2 system generates information (order, report or request) and needs to send it to a simulation system or the use case where a simulation system needs to have access to information when it is available, exchanging C-BML information is similar to exchanging any form of information across a network. Furthermore, if we consider the C-BML message as being part of a document made of one or more messages, then *C2 to simulation interoperability is equivalent to exchanging documents using some transport mechanism over a network.* How this information is produced, interpreted and used is independent of how it gets from one system to another. Furthermore whether information is described as a protocol data unit, as part of a federation object model or as a tactical message, *information is information.* It is the most important component and everything else is there to support better description of information, better transfer of information or better interpretation of information.

Another key observation is that while the WWW supports the exchange of structured information, there is a lot of information that is unstructured (free text, voice) or semi structured. The WWW works despite the fact that several heterogeneous systems are producing and or consuming information near real time. If we make a parallel to C2 to simulation interoperability where unstructured and structured information coexist, we should be able to reproduce a WWW of systems (C2, simulations or any system) that produce and exchange real and/or simulated information in real or faster than real time. This idea is not new and has been proposed as the global information grid (GIG) or net-centricity (Alberts, 2002). This is only possible if, similarly to the WWW, we decouple the transport of information from its content, the content of information from its meaning and its meaning from how it is used. In addition, since C-BML is under development and we recognize that many standards will emerge in the future, we need to minimize the impact of changes to the interoperability system that we are proposing.

Using the ideas of the WWW (exchange of documents using a transport mechanism), the CBMS system is architected following a service oriented approach. The CBMS system is partitioned into technical, syntactic, semantic, pragmatic and conceptual interoperability components that can be combined to deliver an overall interoperability solution. In Figure 1 we show the CBMS components which are grouped into two functional areas. The information exchange components support all aspects of information exchange including its transport, storage and validation. The second functional area has components

that focus on the management and support of system of system interoperability in terms of search, control (start, stop, save) and synchronization as well as visualization (COP).



Figure 1: CBMS components

Each component is a service that plays a role in support of interoperability. Whenever relevant, we identify and use standards in conjunction with a service. While standards are needed and necessary, it is important to recognize that they must be focused to address a need and must not over-specify in areas where successful standards already exist. It is important to note that we distinguish between standards and their implementations. Standards are agreed by a community of interest in support of interoperability, implementations of standards can vary in conformance to the standard specification as well as in efficiency and extensibility among other criteria. In general, we agree that adherence to standards is a good practice but we would like to be able to select an implementation of a standard that best suits our needs.

This systems approach allows us to combine and utilize standards while minimizing potential impacts on systems that will use CBMS. For instance, we can use current simulation interoperability standards such as the Distributed Interactive Simulation (DIS) (IEEE, 1998) and the High Level Architecture (HLA) (IEEE, 2000) that support time management and simulation control (start, stop, save) to develop services that federates can use if and when necessary. These services can be used standalone or combined with other services in support of interoperability. As a result of attempting to deal with interoperability and using a service oriented approach, the CBMS system is similar to the WWW infrastructure in that it treats every system connected to it as an information system. Systems that use CBMS consume services and as such are responsible for selecting the service they want, producing what is required for the service to be executed and consuming what is produced by the service. As per the requirements, CBMS is agnostic of format, content, model, referent and implementation. This forces systems that use CBMS to be in the

driver's seat of interoperability as they have to decide what information they need, what information they can produce, how to interpret the information once they receive it and how to react (pragmatics) to the information once it is interpreted.

The systems approach also allows the addition of components as services. For instance, a translation service can be added to CBMS as an additional component to take in one standard format and translate it into another. At the managerial level, it is easier to migrate to newer standards and implementations because of the minimal effect of change on the overall system which translate into reductions in migration cost. As implementations of existing standards improve and evolve, the systems approach of CBMS allows a phased transition that minimally affects existing federations. This results in improvements in performance with a reduction in cost (assuming the newer implementation is better than the old one). In addition, because of the shift to information-centric interoperability, the issue of version control and compatibility disappears as long as the CBMS component provides the same functionality.

In the next section, we present the CBMS implementation in detail and discuss how systems can use CBMS in support of interoperability.

CBMS IMPLEMENTATION

CBMS is implemented as a collection of web services that make use of existing standards and design patterns. The use of web services in support of M&S interoperability has been promoted by Tolk and Diallo (2005) among others. In the case of CBMS, the selection of which standards and technologies to use is driven from the requirements of an interoperability system and the functionality that we need to support. The CBMS implementation has the following features which match the CBMS components:

- Support the transport of structured documents between C2 and simulation systems. The Hyper Text Transport Protocol (HTTP) used by the WWW is the standard of choice because it decouples the transport of information from its format. We use the Representational State Transfer (REST) software architecture design for distributed systems as a best practice because it allows us to treat C-BML documents as resources that are exchanged over HTTP.
- Support syntactic, semantic and pragmatic validation of information contained in C-BML documents. The use of the XML schema definition (XSD) standard, the standard web

ontology language (OWL) and the Java Community Process standard 94 (JSR-94) respectively allows us to support each level of validation.

- Support a subscription service that makes information available to users on demand and as soon as it is available (similar to subscribing to a newsletter). We use Server-Sent Event (SSE) which is in the process of being standardized as part of the HTML 5 specification by the world wide consortium (W3C).

The use of these standards and best practices allows CBMS to:

- Support filtering of information based on user defined criteria (similar to defining search criteria on the WWW). In this case, we rely on the “Not only Structured Query Language” approach which provides us with the flexibility of supporting multiple formats and relieves us from using a traditional relational database management system. This approach allows us to support the integration of legacy and future messages and message formats (similar to the creation and destruction of web sites).
- Support the visualization of data on mobile devices as well as traditional user interfaces from multiple geographically distributed sources. This is possible because we use HTTP as the underlying transport mechanism.

The CBMS implementation itself follows the modular principles of the WWW. Table 1 lists the CBMS components and the software used to implement it.

Table 1: Overview of CBMS Software Components

Software	Capability
Jersey	Java API for implementing RESTful web services.
Guice	Generic framework providing support for the dependency injection design pattern.
Jena	A framework for building semantic web applications with an ontology API and reasoning engine.
Drools	Business Rule Management System (BRMS) which checks data against a rule set.
BaseX	API to the XML database management system.

Each service is implemented independently as a module using open source products found on the WWW. The services are connected via an injection engine that separates what the service does from how it does it. This gives us the ability to easily migrate to better solutions as they emerge without significantly impacting other components or the CBMS architecture. Using CBMS, systems exchange C-BML documents containing a combination of orders, reports and requests. Documents can be combined to form a resource. Each resource can be accessed and manipulated separately. A system can create, delete, modify and search a resource. Similarly, a system can subscribe to information tailored to their needs before or during runtime and CBMS will provide that information as soon as it is available.

Figure 2 shows a step by step flow of how to use CBMS to exchange information which can be broken down into to following steps:

- CBMS connectivity: The first thing systems need to do is become CBMS compatible at the technical level. This means that systems must become HTTP compatible and support SSE for the subscription service. The CBMS implementation offers an interface that systems can use. This interface also supports the production, parsing and consumption of documents.
- The standard service: The standard service consists of a combination of the transport, validation and storage services. A system can send, delete, retrieve or modify a document or sets of documents. When a system sends a document, CBMS automatically validates its syntax against a schema that can be located anywhere on a network. Along with the document, a system can specify a semantic description of the entities and relationships mentioned in the document and a set of rules that govern what they can and cannot do. For instance, a system can send a C_BML order that tasks a tank in another system to move from one location to another. Along with this document, the system can specify a tank ontology that describes tanks and their attributes (tanks are mounted with four soldiers including at least one driver and one gunner, etc.) and a tank rule set that specify conditions under which a tank can follow a move order (the order has to come from a hierarchical superior, the tank has to be mounted and cannot be in execution mode etc.). The system representing the tank ensures that the order is consistent with the specified constraints or notifies the tasking system that

it cannot perform the task due to rules or ontological violations. If the document is syntactically valid it is stored the way it is produced. CBMS in accordance with the requirements for an interoperability system does not have its own internal data model such as a relational data model and therefore does

not change a document for storage purposes. The standard service can be reduced to the transport function if required. Validation at all levels and storage can be turned on or off depending on the needs of the federation being supported.

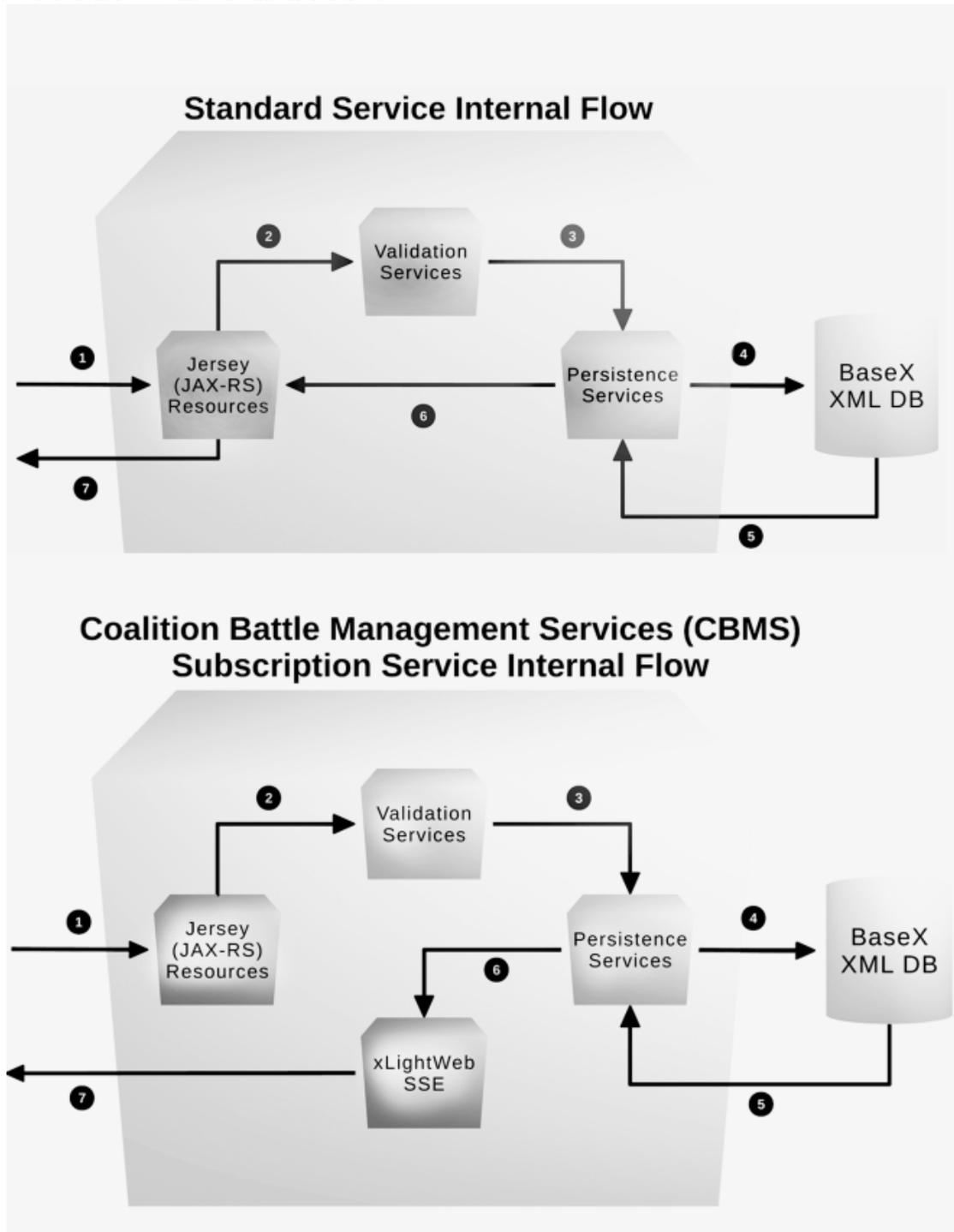


Figure 2: CBMS Information Flow

- The subscription service: The standard service requires each system to post documents when necessary regardless of what system ends up using it. Conversely, it requires each system to get documents based on their needs. As a simple example, the standard service allows a system to post a report and any system that is interested in the report can have access to it and can potentially use it. The subscription service gives system the ability to describe content that they are interested in receiving in the future through SSE. Through CBMS, a system can for instance declare that it is interested in documents that contain information about explosive devices within an area of interest in a given timeframe. This information is sent to the system as soon as it becomes available regardless of who produced it. CBMS manages the notification process, thus freeing the system to continue executing without having to constantly poll. The subscription service allows a conversation to take place between two or more systems where each system only receives information that it relevant to it and only shares information that is needed.
- Using information: Once a system has connected to CBMS and is able to send and receive information it must be able to use this information. Based on the CBMS approach, each system is responsible for implementing the means by which it handles information. The main advantage of this approach is that systems can become compliant with multiple standards simultaneously without the need for translators and gateways. A system can be C-BML compliant by simply being able to produce and consume information in accordance with the C-BML schema which provides both the syntax and semantics of C-BML data and metadata. Similarly, the same system can become MSDL compliant or comply to however many standards it needs to adhere to. If the system needs to internally interpret information it receives through CBMS, it has to decide internally how to implement its interpretation. It is worth noting that misalignments and mismatches can still occur but this is not due to CBMS rearranging information or imposing its interpretation of what it is transporting. The mismatches are

due to variance in the systems and their respective interpretation.

At the managerial level, the adoption of open source tools and standards provides us access to free resources without the cost of having to manage, maintain and improve a proprietary solution. The CBMS implementation is currently hosted on a server at the Virginia Modeling Analysis and Simulation Center (VMASC). CBMS is also packaged as self-contained software that can be deployed as a server on a computer to support internal testing and integration. In the next section, we present how CBMS is used in a coalition environment.

CBMS Users and Use Case

CBMS is used in the NATO Modelling and Simulation group 085 (MSG-085) which deals with C2 to simulation interoperability. During IITSEC 2011, CBMS was demonstrated in a ground maneuver scenario involving the U.S. and Spain systems. In this scenario depicted in Figure 3, a U.S. and Spanish Company coordinate maneuver to approach and clear a building in which insurgents are retrenched. The federation involves automatically initializing a C2 surrogate and two simulation systems (OneSAF for the U.S. and VR forces for Spain) using MSDL (SISO, 2008). Once initialized, the C2 system uses C-BML to task entities in the simulations to move to their specific areas. The entities use C-BML to report their position and status at a given interval. In addition, entities provide a SPOT report whenever applicable. Once in their respective areas, the U.S. company provides support by fire. While the Spanish company clears the house, the U.S. company moves to a phase line beyond the house to ensure that the enemy will not attack from behind.

CBMS was used to exchange C-BML and MSDL messages between the surrogate C2 system and the simulations as well as the exchange of reports between the simulations which were federated. The technical integration took less than two weeks and involved one developer from each nation. As described earlier, each simulation was made CBMS compatible and C-BML and MSDL compliant which means that each simulation is able to produce and consume C-BML and MSDL information. The scenario was developed independently at the U.S military academy by French cadets who also generated the C-BML and MSDL messages.

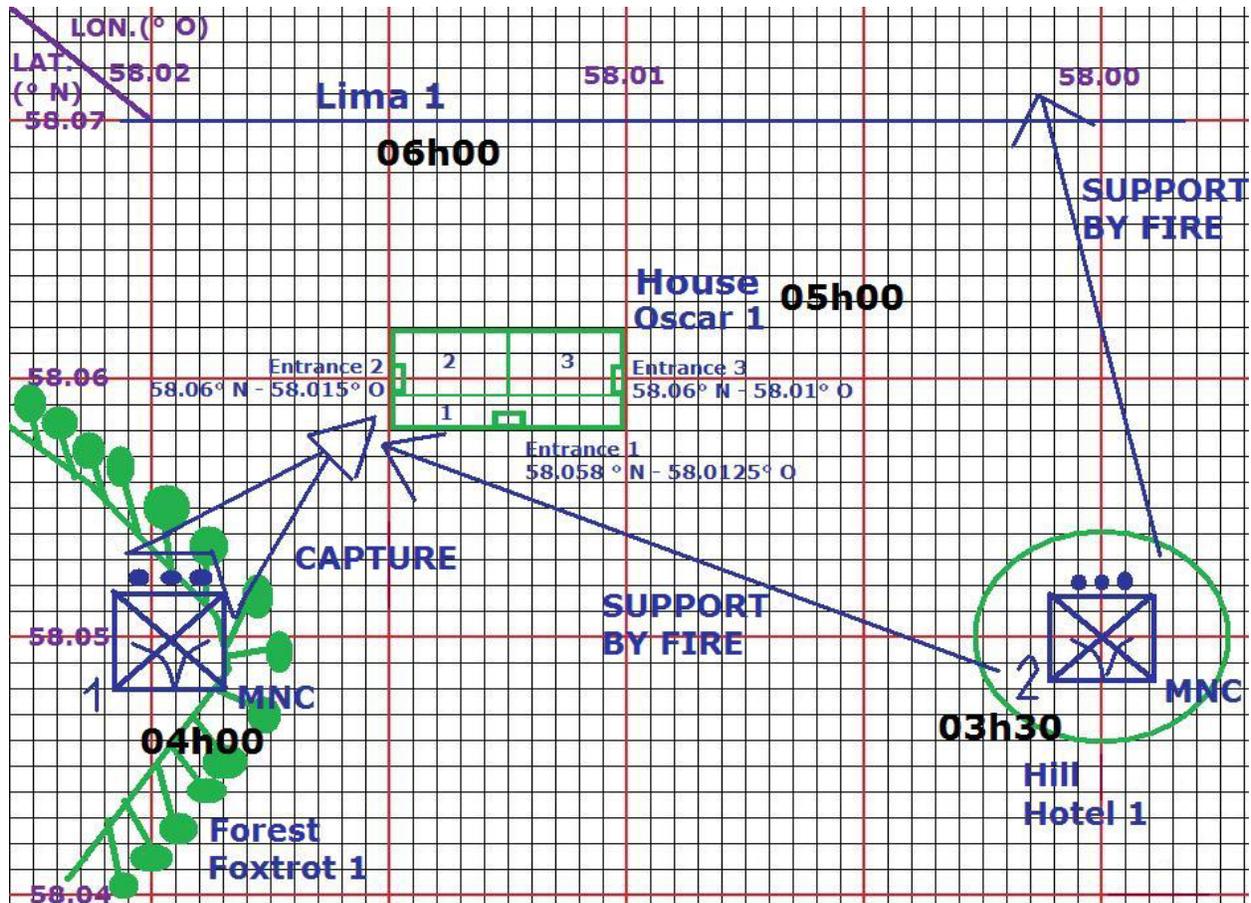


Figure 3: Ground Maneuver Scenario

France used CBMS at the EUROSATORY event to demonstrate C2 to simulation interoperability across echelons (brigade to platoon). CBMS has been offered as an infrastructure in the MSG-106 which deals with enhanced computer-aided exercise architecture, design and methodology.

CONCLUSION AND FUTURE WORK

Because CBMS uses open standards and freeware, the cost of integration is negligible because there is no upfront purchase cost due to the use of freeware, and integration might be less costly than alternatives given the basis in documented standards. In addition, the tools used are maintained and updated by the open source community which reduces maintenance and improvement cost. We want to emphasize that all of the standards used are industry standards that already exist and are extensively used. As we have shown, these standards are applicable to the problem set we are dealing with and we could not find any compelling reason why they cannot be used as extensively to make military systems interoperable. CBMS has been used to exchange messages from C2 to simulation, to initialize

multiple simulations using MSDL, to task entities within simulations using C-BML, to command and control simulations using MSDL and C-BML simultaneously and to interoperate multiple simulations using C-BML. CBMS has also been used to exchange binary files such as images and video between live unmanned vehicles developed at the Virginia Modeling Analysis and Simulation Center (VMASC) and simulation systems.

In every case, the CBMS infrastructure was not physically modified. Developers reuse the same interface and focus on producing information following the syntax and semantics of the language and using that information as required. The subject matter experts focus on the operational picture to specify what information is needed and how to interpret it. Domain knowledge can be formally captured using OWL while the interpretation of knowledge can be specified in the form of rules and procedures that can be captured by a standardized rules engine. CBMS provides services to validate domain knowledge and rules. Another approach is to use domain specific languages (DSL) to capture expert knowledge as has been discussed by

Tapp et.al. (2011). The technical staff can focus on how to produce and consume that information without worrying about details of how that information will be transferred and interpreted.

In terms of interoperability, the emergence of language-based standards such as C-BML and MSDL has generated a change in focus from an interface-centric approach to a system of system information-centric approach that parallels how the WWW is built.

It could be argued that language-based standards are becoming more prevalent because of the need to couple and reconfigure systems for different purposes rapidly and cheaply. Reuse of well-established WWW transport mechanisms is useful and removes semantics from the protocol level, but software still must exist to provide semantic interpretation for true interoperability. Abstracting semantic interoperability away from the transport level is a good first step.

The main contribution of the CBMS approach is to recognize that interoperability is a general problem that should be approached as a system of system problem. In other words, CBMS provides an approach applicable to different situations by focusing on information exchange instead of *ad hoc* solutions that change with respect to federates. We use existing open standards and freeware to develop CBMS, a set of services that support interoperability at the technical, syntactic and semantic level. This service approach supports the future addition/modifications of standards and complements current efforts such as HLA.

Currently CBMS is mainly used to support C2 to simulation interoperability where the requirements for information exchange frequency range in minutes or hours. The main question that needs to be answered is whether interoperability systems such as CBMS should support real-time or faster than real-time information exchange frequency. While the initial feasibility in terms of frequency has been demonstrated, future work include scaling CBMS to very large federations across multiple levels of security exchanging information near real time or faster than real time. In order for CBMS and other web-based approaches to survive and be adopted within the military community, they must exceed the performance of current approaches in terms of scaling and efficiency in addition to their already demonstrated capability.

Within the coming months, we will perform formal performance testing on every component as well as an overall performance test using operationally relevant uses cases. We intend to report those results to the community in the near future.

ACKNOWLEDGEMENTS

The authors acknowledge the Coalition Warfare Program (CWP) for their funding support in developing CBMS. The authors also thank the coalition partners in the MSG-085 group and notably Canada and Spain for providing systems and support for test and integration. Finally, the authors thank the U.S. Military Academy for their feedback and technical support.

REFERENCES

- IEEE(1990). "A Compilation of IEEE Standard Computer Glossaries". New York: IEEE Press.
- Blais C., K. Galvin & M. Hieb. (2005) Coalition Battle Management Language (C-BML) study group report. *Proceedings of the IEEE Fall Simulation Interoperability Workshop*. IEEE CS Press.
- Pullen, J., D. Corner & S. Singapogu (2009). Scripted Battle Management Language Web Service Version 1.0 Operation and Mapping Description Language. *Proceedings of the 2009 Spring Simulation Interoperability Workshop*, San Diego
- Wang, W., Tolk, A., & Wang, W. (2009). The levels of conceptual interoperability model: Applying systems engineering principles to M&S. *Proceedings of the 2009 Spring Simulation Multiconference*. San Diego.
- Schade, U. & M. Hieb (2006). Development of Formal Grammars to Support Coalition Command and Control: A Battle Management Language for Orders, Requests, and Reports. *11th International Command and Control Research and Technology Symposium*, Cambridge, UK.
- Diallo, S.Y., Tolk, A., Graff, J., & Barraco, A. (2011) Using the levels of conceptual interoperability model and model-based data engineering to develop a modular interoperability framework. *Proceedings of the 2011 Winter Simulation Conference*. Phoenix, AZ.
- Alberts, D.S., (2002). *Information Age Transformation: Getting to a 21st Century Military*. Washington, DC, CCRP Publications. First published 1996.
- IEEE.(1998). Standard for Distributed Interactive Simulation. IEEE Std 1278-1998.
- IEEE.(2000). "Standard for modeling and simulation (M&S) high level architecture (HLA)-framework and rules." IEEE Std 1516-2000.
- Simulation Interoperability Standards Organization (2008). Standard for Military Scenario Definition Language (MSDL) SISO-STD-007-2008.
- Tapp, M., Chartrand, S., & Campeau, J. (2011). Experiences in Leveraging M&S Expertise by Hiding Software Complexity. *Proceedings of Interservice/Industry Training, Simulation, and Education Conference*. Orlando, FL.