Simulation Scenario Encoding for Reuse

Captain Michael J. Eady, USMC
Marine Corps Training and Education Command
Quantico, Virginia
michael.eady@usmc.mil

Lieutenant Colonel David W. Parkes, USA
Joint Staff J7
Suffolk, Virginia
david.w.parkes.mil@mail.mil

ABSTRACT

The United States Army and United States Marine Corps employ the Virtual Battlespace 3 (VBS3) commercial game for first-person small-unit training and have invested significantly in training scenarios constructed using proprietary tools and data formats. Open standards data structures need to be utilized in order to move toward improved interoperability, address the statutory intent for open competition and affordability, and protect investments made in models, terrain, and other elements of scenarios that are separate and distinct from the game engine source coding. Expanding capabilities for open scenario interchange will improve scenario reuse while creating greater opportunities for simulation data interchange and open competition for future virtual training capabilities. This paper describes and demonstrates initial application of Extensible Markup Language (XML) technologies to represent and interchange simulation scenario data. Design of XML data structures to capture a subset of a VBS2 scenario's data content is successfully demonstrated, and the capability to transform content from the XML model back to the VBS2 scenario data formats utilizing an Extensible Stylesheet Language Transformation (XSLT) document is discussed. Proposed extensions to existing and developing simulation standards are made in order to accommodate the set of data used in VBS2 scenarios. The research provides a foundation for future efforts to determine the feasibility of creating an open XML schema that addresses all critical aspects of a simulation scenario, which will enable open competition for the first-person "games for training" requirement while preserving investments in proprietary data structures.

ABOUT THE AUTHORS

Captain Michael J. Eady is an active duty United States Marine with the primary military occupational specialty (MOS) of Infantry Officer. Following graduation from the Naval Postgraduate School's Modeling, Virtual Environments, and Simulation Institute in 2014 with a Master's of Science, he received the additional MOS of Modeling and Simulation Officer. He is currently assigned to the Marine Corps Training and Education Command's training simulations branch, where he works to ensure simulation systems and applications are fielded, accredited, and sustained to support training of the Marine air-ground task force (MAGTF), and is helping to shape the Marine Corps Live, Virtual, Constructive Training Environment (LVC-TE).

Lieutenant Colonel David W. Parkes is an active duty United States Army Officer. He served 10 years in the Infantry through various levels of command, and another 8+ years as a Simulation Operations Officer (FA57) supporting all echelons of Army training (Squad through Corps). He is a graduate of the Naval Postgraduate School's Modeling, Virtual Environments, and Simulation Institute with a Master's of Science. He is currently assigned to the Joint Staff, J7 in Suffolk, VA where he has worked with Combatant Commands synchronizing simulation supported exercises and is now leading the Environment Branch which involves future development within Joint Live, Virtual and Constructive (JLVC), as well as the Architecture to support.

Simulation Scenario Encoding for Reuse

Captain Michael J. Eady, USMC
Marine Corps Training and Education Command
Quantico, Virginia
michael.eady@usmc.mil

Lieutenant Colonel David W. Parkes, USA
Joint Staff J7
Suffolk, Virginia
david.w.parkes.mil@mail.mil

INTRODUCTION

The United States Army and United States Marine Corps employ the Virtual Battlespace 3 (VBS3) commercial game for first-person small-unit training and have invested significantly in training scenarios constructed using proprietary tools and data formats. Open standards data structures need to be utilized in order to move toward improved interoperability, address the statutory intent for open competition and affordability, and protect investments made in models, terrain, and other elements of scenarios that are separate and distinct from the game engine source coding. Expanding capabilities for open scenario interchange will improve scenario reuse while creating greater opportunities for simulation data interchange and open competition for future virtual training capabilities. This paper describes and demonstrates initial application of Extensible Markup Language (XML) technologies to represent and interchange simulation scenario data. Design of XML data structures to capture a subset of a VBS2 scenario's data content is successfully demonstrated, and the capability to transform content from the XML model back to the VBS2 scenario data formats utilizing an Extensible Stylesheet Language Transformation (XSLT) document is discussed. Proposed extensions to existing and developing simulation standards are made in order to accommodate the set of data used in VBS2 scenarios. The research provides a foundation for future efforts to determine the feasibility of creating an open XML schema that addresses all critical aspects of a simulation scenario, which will enable open competition for the first-person "games for training" requirement while preserving investments in proprietary data structures.

Department of Defense Reuse Policy and Cost Savings

Reuse is simply "the practice of using again, in whole or part, existing modeling and simulation (M&S) tools, data, or services" (Under Secretary of Defense (AT&L), 2007). The benefits of scenario and data reuse in the M&S community are many. The reuse and modification of scenarios over time improves the quality of scenarios and can further validate the original. Also, units utilizing reused scenarios are able to more quickly set up training events, spending less time on development of scenarios, and more time actually training individuals. Other benefits include improvements in cost effectiveness and commonality in training when reused over a large population.

Official Department of Defense (DOD) policy indicates that reuse of M&S assets is encouraged (Under Secretary of Defense (AT&L), 2007). Supporting this policy is the implementation of best practices to strengthen the Defense Department's buying power, improve industry productivity, and mitigate risk to national technological superiority, wrapped into an initiative called Better Buying Power (BBP), established by the Under Secretary of Defense for Acquisition, Technology, and Logistics (AT&L) in 2010. The BBP, and its successors BBP 2.0 and now 3.0, calls for new M&S acquisitions programs to enforce open architectures and promote effective competition in order to maximize the government's investment with the intent of facilitating reuse across the joint force (Under Secretary of Defense (AT&L), 2015). Further, the DOD's Acquisition M&S Master Plan specifically notes that reuse is highly desirable at any point where the asset being reused can meet a requirement more cost-effectively (Under Secretary of Defense (AT&L), 2006).

REUSE TECHNICAL INITIATIVES AFFECTING DOD M&S

Reflecting the concern for savings in cost and time of development of new systems in DOD M&S, many efforts conducted over the past few years have been performed. Some of these efforts are described in the following subsections.

Military Scenario Definition Language

The Military Scenario Definition Language (MSDL), developed and approved by the Simulation Interoperability Standards Organization (SISO) in late 2008, serves as a common point of reference for verifying and initializing military scenarios across a range of military simulations, C4I, and mission command (MC) systems. The MSDL is an international XML standard for describing military scenarios, but has been shown to require model extensions for broader application (Beris & Whittington, 2008). Defined through an XML Schema, the MSDL functions to develop and reuse military scenarios across simulations and scenario generation tools, and is intended to evolve and grow over time (Military Scenario Definition Language Product Development Group, 2008). As user knowledge and experience grows in implementing the MSDL for capturing snapshots of a military scenario, improvements and additions will likely increase functionality and ease of use, resulting in a robust standard for scenario interoperability and reuse.

Perhaps one reason explaining the relatively slow adoption of MSDL by the military simulation community is the lack of scenario generation user interfaces available to a wide user base. Without a significant user base with access to tools that can effectively generate desired scenarios and output validated MSDL-compliant scenarios for experimentation, the standard faces issues in succeeding (Ullner & Lundgren, 2008). Alternatively, should the DOD begin requiring MSDL compliance as part of its contracts as it does in the case of Distributed Interactive Simulation (DIS) and High Level Architecture (HLA) specifications, the user base would grow by necessity, forcing faster adoption and quicker improvements to the MSDL specification.

Through collaborative research with Saab Training Systems, Ullner and Lundgren (2008) set out to accomplish three tasks in relation to MSDL. First, they aimed to create a common reference model for serious games that identified key elements of interactive entities within several serious games. Second, their work resulted in the creation of a proof of concept application interface for generating scenarios that validate with the MSDL schema. Finally, Ullner and Lundgren exported scenarios from their scenario editor in MSDL form for use in initializing scenarios in MSDL-compliant simulations. This research demonstrated that a common, standard reference model, or list of attributes necessary and common to all simulation scenarios, paired with an MSDL-compliant user interface that could export scenarios, could easily serve to produce reusable and interoperable scenarios across many simulations.

There are nine foundational elements to the MSDL specification. They range in the data they hold from scenario meta-data to specific order of battle and tactical graphic representations to be interpreted and used during the scenario's execution (Tolk, 2012a). These nine elements define those characteristics and the associated data that must be transformed into a "common" framework to facilitate accurate representation across each simulation, but more importantly, to promote interoperability and stimulation of mission command systems. These nine elements are: Options, Scenario ID, Environment, Force Sides, Organizations, Overlays, Installations, Tactical Graphics, and Military Operations Other Than War (MOOTW) Graphics.

Command and Control Systems – Simulation Systems Interoperation

The Command and Control Systems – Simulation Systems Interoperation (C2SIM) PDG began work at the 2014 Fall Simulation Interoperability Workshop and combines two previous SISO efforts. It combines MSDL (now called "C2SIM – Initialize") and the Coalition Battle Management Language (now called C2SIM – TaskingReporting). While it is still early in the development cycle of C2SIM ,the effort aims to standardize simulation and C2 initialization data transport, expand the range of tasking and situational awareness information, and provide a logical data model containing a core set of data elements common to most C2 and simulation systems.

Regardless of the progression of versions in SISO standards, DOD is currently on the right path to endorse, embrace, and help create standards in order to increase scenario reusability and interoperability across simulation applications. The Defense Modeling and Simulation Coordination Office (MSCO) efforts with regards to established Technical Working Groups (TWG) and Architecture Working Groups (AWG) will enhance information sharing opportunities. We believe that further development of the C2SIM standard can be informed by our research exploring XML representations of VBS2 scenario data.

Joint Training Data Services

The Joint Training Data Services (JTDS) is an initiative to provide simulation initialization databases based on authoritative data sources data preparation techniques to support storage, access, and utilization of information in C2 and simulation systems (Chambers, 2015). The initiative is an online web-enabled system, available worldwide 24/7 on the JS J7 Cloud via the Non-classified Internet Protocol Router (NIPR) and Secret Internet Protocol Router (SIPR) networks. The services provided by JTDS include the Order of Battle Service (OBS), Terrain Generation Service (TGS), and a potential third to support faster than real time simulation capability and staff planning. OBS is a fast, single repository for data holding more than 200 million records that possesses intuitive drag-and-drop scenario building tools. It provides an integrated rule set for Joint Live, Virtual, and Constructive (JLVC) data model error checking, and also produces output as a standardized documented XML file, providing a common data set for JLVC initiation. The data can also be accessed via a machine to machine application programming interface. The data produced for scenario events is archived for reuse.

The JTDS initiative created an XML representation of order of battle information, providing content through the Order of Battle Service (OBS). Via the online OBS Repository provided by JTDS, units and trainers can download scenario files (.obs format) containing validated force structure data for an exercise, which can be shared over multiple simulation applications to ensure a common, correlated data set is utilized (https://milgaming.army.mil/). The information commonly contained in an OBS file includes command structure, strength, and equipment of units involved, as well as their descriptions and locations where they are initialized in the simulation. Much of the information coming from the OBS file is also critical in populating necessary data utilized by runtime interoperability frameworks.

Cloud Computing Services

Cloud computing is a method that the DOD intends to leverage through the Joint Information Environment (JIE), not only for common business applications, but for supporting management of M&S data as well. Characterized by on-demand self-service, broad network access, resource pooling, rapid elasticity, and measurable service (Mell & Grance, 2011); cloud computing is essentially a seamless link to multiple data centers. Cloud-enabled reuse of computing resources is critical to maximizing data efficiency and ultimately reducing overall cost if widely implemented – a larger cloud equals higher cost savings. Ensuring these resources are operated at higher capacity with decreased infrastructure needed is extremely helpful. In addition, a multiple data center construct enables the consolidation of "like" databases, decreased redundant security layers, as well as implementation of Virtual Machine (VM) technology, affording the user the opportunity for ease of access and data sharing. Each Service is crafting JIE directed policy for mandated identification and consolidation of information technology assets in order to eliminate unnecessary redundancy, cost, and risk. Collaboration on these efforts, across the services, will enhance the integration and consolidation efforts.

SCENARIO DATA

Overview

VBS2 is a first-person shooter style capability, small unit tactics simulation utilized by the U.S. Army and Marine Corps for a wide variety of training scenarios and applications in a three dimensional (3D) virtual environment. Since the Marine Corps' enterprise license purchase in 2006, the capability improved significantly across a range of functions, including improvements to the game engine, terrain import and adjustment capabilities, addition of call for fire and casualty evacuation scenarios via scripting commands, and the development of a robust HLA and DIS gateway for interoperability compliance (Bohemia Interactive, 2012a). The simulation has proved a valuable and utilized training tool, as evidenced by the Marine Corps' investment of $10.5 million in a contract to continue improving VBS2 functionality awarded in late 2011 (Marine Corps Systems Command/ Program Manager–Training Systems Orlando, 2011). Similarly, the U.S. Army announced in July of 2013 that the successor product to VBS2, VBS3, would serve as the Army's flagship game for training for at least the next five years upon delivery, with an assurance from the product developer that VBS3 will focus on reuse and open standards (Bohemia Interactive, 2013).

Designed to run on a single desktop or laptop computer, VBS2 employs a proprietary game engine called Real Virtuality 3 (RV3) developed over a thirteen year period by the company Bohemia Interactive. RV3 real-time renders large, high-fidelity terrain areas, loading 3D models and the terrain during runtime to provide a realistic, heavily populated virtual environment (Bohemia Interactive, 2012a). As an entity-level simulation, human participants in the execution of a simulation run control a character entity, inclusive of a wide range of realistic military occupational specialties. Because of the entity-level nature of the simulation, along with realistic rendering of terrain and objects in the virtual environment, it offers small unit level users an environment to conduct mission rehearsal exercises and procedural training events in a cost-effective and repeatable manner.

The networking architecture of the product allows for up to 100 users to execute a highly customizable training scenario. The game utilizes a standard client-server model for networking, meaning that one computer is designated as the server, through which all other machines connected in the network will communicate during runtime. Only one server is permitted during a networked scenario execution, which houses a master copy of all of the objects, terrains, and entities within the virtual environment (Bohemia Interactive, 2012a). The servers currently do not interoperate with each other, however the servers can be linked through an HLA/DIS gateway developed by Calytrix Technologies called "LVC Game," which has been developed in parallel with the overall capability since 2007 (Bohemia Interactive, 2012a).

VBS2 Scenario Data Structures

Scenario Objects. Scenario data are stored in several files that are combined for loading into the simulation prior to execution. There are three files required for scenarios to be run locally (not in a networked training mission utilizing a dedicated server). These files are as follows (Bohemia Interactive, 2012b):

- mission.sqm: describes which units in a scenario are playable by the user (as opposed to AI–controlled), and defines groups of players or units. This file is essential in starting a networked scenario. This file is parsed first, meaning that units and groups are the first elements of the scenario to be created during initialization.
- mission.sqf: a script file which executes after all of the units and groups have been initialized. This file will create vehicles, waypoints, markers, and any other control measures or editor objects built into the scenario.
- mission.biedi: describes ALL editor objects in the scenario. This file is essentially the combination of the mission.sqm and mission.sqf files, and when edited directly outside of the program, will initialize additional editor objects not saved through the VBS2 offline mission editor (OME), as long as the content is correctly formatted.

All of the above listed files are text files, readable with any text editor. While application help manuals indicate that directly editing these files should never be attempted due to the possibility of corrupting the mission, this research indicates that when done correctly, there is no adverse effect in editing the files directly, particularly the .biedi file, which is of the most interest to this research, as it consolidates all of the editor objects in the scenario. For running networked training missions, the program wraps the three mission files into a compressed format ending in .pbo which when unwrapped by the game engine during initialization, expands back into a file tree structure. The .pbo files used for networked and training scenarios are generally encoded and cannot be directly edited using a text editor in the same manner as the mission files can. Files of this nature are also expressly mentioned in the user license agreements for both the Army and Marine Corps as forbidden for unpacking or manipulation unless a case is specifically addressed and allowed by the license agreements. While publishing proprietary code structures in a paper would violate user license agreements, the structures in the aforementioned files which define entities in the virtual world are straightforward enough to allow for a comparison to the characteristics of how scenarios are defined by standards such as MSDL.

3D Models. There are several open 3D modeling languages in widespread use today for interchange of 3D models across software systems, including the Web3D Graphics Language (X3D; http://www.web3d.org) and Collada (https://collada.org) (much like Open Scene Graph). Both X3D and Collada are XML representations of 3D objects. These provide the graphical geometries as well as scripted behaviors that for example, animate articulated components such as tank turrets and weapons.

VBS2 uses a proprietary format for 3D models in the game, with a ".p3d" file extension. Oxygen 2 (O2) is the proprietary modeling software utilized to create P3D files for use in the simulation. O2 provides the capability for transfer of P3D files to another proprietary file format (.fbx), which can be manipulated in Autodesk's 3DSMax software (Bohemia Interactive, 2012c). The end-user license agreements for the USMC and USA are clear as to the use of the models provided by Bohemia Interactive. Models developed by BISim are not to be exported for use in any software application outside of VBS2. This clause does not preclude modification of models to be utilized within the game, nor does it disallow Army and Marine developers to add their own models.

Numerous tools and 3D production products exist that can convert across 3D formats, but seldom without some loss of information. Open standards, such as X3D, provide promise for greater interchange across formats, but the problem of proprietary formats remains. While it would be desirable for 3D models of troops, vehicles, weapon systems, buildings, environmental features, and other aspects of the battlespace (to include both the visual representation and associated behaviors) to be easily interchanged across systems, this remains a challenging area for future research.

PARSER AND XML SCHEMA DESIGN

To demonstrate a method of preserving scenarios in a reusable and interchangeable format, we developed a software parser implementation using only a representative subset of a full VBS2 scenario's data format. The parser reads data describing scenario objects from a VBS2 scenario file, stores the extracted data in internal software objects, writes out that information in XML structures conforming to an XML schema we created to define various VBS2 objects in XML. We selected Java as the language of choice for developing this software.

The software design is based on the Interpreter design pattern: "Given a language, define a representation for its grammar along with an interpreter that uses the representation to interpret sentences in the language" (Gamma, Helm, Johnson, & Vlissides, 1995, p. 243). Grand (1998) defines a pattern for Java programs named "Little Language" using the general Interpreter pattern.

Key components of the design are:

- *Client*: The program or subprogram (class instance) that controls the initiation and completion of the processing. The *Client* creates an instance of the *Parser* class to parse information from an *InputStream* object. The *Client* calls the *Parser* object's *parse* method to initiate the action.
- *Lexical Analyzer*: The *Lexical Analyzer* object instance is created by the *Parser* object's *parse* method. The *Lexical Analyzer* reads characters from the *InputStream* object that was passed to it. The *Lexical Analyzer* applies grammar rules to extract tokens from the *InputStream* object. Tokens are defined as "the constructs that make up a grammar," and are classified as either terminal (not made up of subordinate tokens) or nonterminal (higher level constructs defined in terms of terminal tokens) (Grand, 1998, pp. 292-293). As stated in (Grand, 1998, p. 293): "The rules that determine how to recognize sequences of characters as terminal tokens are called *lexical analysis rules*. The rules that determine how to recognize nonterminal tokens as sequences of terminal and nonterminal tokens are called *productions*." Java has a built-in class named *StreamTokenizer* that separates a text stream into tokens based on simple syntactic rules (e.g., using separators such as spaces, special characters such as "=", etc.).
- *Parser*: Created by the *Client* object, the *Parser* object's *parse* method matches tokens from the *InputStream* object (extracted by the *Lexical Analyzer*) against the productions of the grammar (Grand, 1998, p. 303).

In constructing the sample application, we used the NetBeans Integrated Development Environment (IDE) (https://netbeans.org/) for Java software development. NetBeans enables a developer to create a software project, automatically generating initial Java software files in accordance with the developer's project objectives. In our case, we used NetBeans to auto-generate files for a basic Java application. NetBeans also provides a selection for auto-generating Java class files corresponding to a specified XML schema using the Java Architecture for XML Binding (JAXB). In addition to providing mechanisms for generating Java objects from the XML schema file, JAXB provides method calls for reading data from an XML structure into Java objects (unmarshalling data) and for writing data from the Java objects to an XML structure (marshalling data). In our case, the source data are contained in a VBS2 file

using a specialized text format that is interpreted by our parser software to transfer the information to the internal Java objects, and the data are then marshalled out to the XML representation.

TRANSFORMING XML CONTENT TO VBS2 AND MSDL DATA STRUCTURES

Based on our XML representation of VBS2 data, we developed an Extensible Stylesheet Language Transformations (XSLT) (http://www.w3.org/TR/xslt) document that reads from the XML document and reproduces an excerpt of the file verified to be correct by showing that the XSLT-created scenario data still loads correctly into the VBS2 software. XSLT is an XML language that provides instructions for reading data from an XML file and writing out a text file. The output file can be another XML format, Hyper-Text Markup Language (HTML), or other text-based format. In our case, we want to read data from the XML representation of VBS2 scenario data and write out a result file containing that data but in the VBS2 format. This process is shown in Figure 1. Once the VBS2 data content is available in XML, transformations can be readily developed to interchange information with other XML-based data models, such as MSDL and IWARS, as well as with real-world data such as the OBS and National Information Exchange Model (NIEM). Follow-on work can also consider how metadata standards, such as the DOD Discovery Metadata Specification (DDMS) and the Intelligence Community Information Security Markup (IC-ISM), can be integrated into the XML structure to enable the scenario data to become discoverable across the DOD data enterprise.

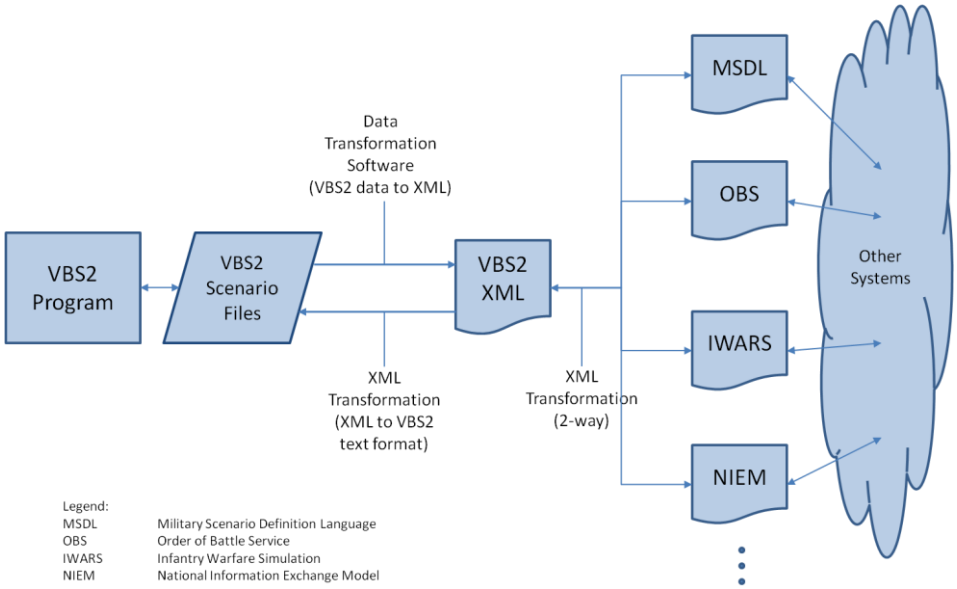

Figure 1. Data transformations between VBS2 scenario data and XML-based data models through an XML representation of the VBS2 data.

Proof of Concept Design and Execution

To demonstrate the functionality of our software we utilized a very simple test case. A VBS2 scenario was created in the program's offline mission editor, comprised of a desert terrain containing two entities: a rifleman and a tank. The scenario file was then saved and the instance of the game closed. Next, we navigated to the location of the "mission.biedi" file and opened it using a standard text editor. The sections of the proprietary data structure defining the rifleman and the tank were removed from the .biedi file and pasted into a new text file. Our software parser was then used to read in this new text file and transform the two data objects (rifleman and tank) into java objects, and then write out an XML file validated with the XML schema we created.

We then desired to show that from our generated XML file format, we could automate a process for transforming the XML file back into the format needed for scenario initialization. We utilized our XSLT file to accomplish this. The output generated by the XSLT application was pasted directly into the original .biedi file that we had removed sections from previously. That file was then saved, and an instance of the game opened. We used the VBS2 user interface to navigate to the scenario we manipulated, initialized it, and visualized the results in the virtual environment. Success was evident in correct data transformation in that both the rifleman and tank were present in the environment. We

were able to assign ourselves the role of controlling both entities, and successfully manipulated them both in the environment according to expectations. The rifleman was able to move about the scene and fire his weapon, and could interact with the tank by driving it and firing its weapon systems. This test case was repeated three times, using different entities (a person and a vehicle) for each test case.

Challenges Associated with Transformation of VBS2 Schema to MSDL Data Structures

While the proof of concept cases outlined above only utilized data structures representing a person and a vehicle, our parser was developed for eleven different object types commonly implemented in VBS2(including: prefix, emptyVehicle, group, IED, intel, marker, object, trigger, unit, vehicle, and waypoint). The eventual goal of implementing an approach similar to ours would be in transforming our XML schema into an industry standard open source scenario initialization format such as MSDL. For each object type we examined, the MSDL was analyzed for possible matching elements or attributes that could serve as a location for storing data during format interchange. Direct attribute and element overlaps were observed for the following elements necessary for a unit initialization and description in VBS2: naming and unique designation of a unit, force side, combat effectiveness, direction of movement (azimuth), position, and scenario time. Of importance is the interchange of positional data when transforming between any simulation systems and languages. Both VBS2 and MSDL utilize a variety of different coordinate systems to describe the position of entities in the simulation. This issue of varying coordinate systems would need to be addressed in the development of any transformational software application moving scenario data between the XML Schema formats. VBS2 scenario initialization files utilize "in-game" coordinates that must be translated into one of the standard coordinate representations through function calls to the game engine. It is recommended that transformational software be coded to always populate the MSDL element of msdl:CoordinatePointType/GCC field, with appropriate functionality incorporated to translate "in-game" coordinates to geocentric coordinates utilizing the correct datum.

In addition to the coordinate system challenge associated with scenario data interchange is the type of equipment being described. The MSDL handles equipment type through the utilization of NATO stock number (NSN) codes (Ruiz et al., 2013). VBS2 defines equipment type by simply referring to the proprietary file name of the 3D model associated with the desired equipment type, which the game engine utilizes to display the correct visualization after pulling the model from a database. These very different methods of specifying equipment types pose an issue to simple transfer of data from one format to another. We propose that a solution is to utilize DIS enumeration codes (SISO-REF-010-2011.1) as the standard method of describing equipment type in the MSDL. The standardized coding is already in wide use and is well-established in the military simulation user community. The brunt of achieving this solution would be in accurately mapping the simulation's model database to the SISO enumerations. Once this is achieved, a simple lookup table could be utilized in a parser to translate the equipment type description into an enumeration capable of being decoded into the appropriate MSDL elements.

WAY AHEAD

The work outlined in this paper successfully demonstrated design of XML data structures to capture a portion of the VBS2 scenario data content and showed the capability to transform content from the XML model back to the VBS2 scenario data formats. We assess that it is feasible to develop an open XML schema that addresses the majority of information contained in a VBS2 scenario initialization file. However, it must be noted that proprietary 3D models are the intellectual property of BISim. If the Army and/or the Marine Corps move away from BISim as the sole-source provider for first-person shooter entity-level gaming requirements, any preserved scenario files would need to be altered or updated to refer to an open or government-owned models database rather than the sample models that populate the current versions of the game.

While not directly addressed in this research, VBS2 allows users to develop custom entity behaviors through the use of a proprietary scripting language similar in form to the C++ programming language. This scripting language makes function-calls directly to the game engine to render desired events and behaviors based on event triggers in the simulation, or at a pre-established time. Based on the work performed during this research and the understanding of the game's functionality, it is assessed that incorporating scripted commands into the XML data structures proposed in this document would be extremely difficult. The primary issue is that a standardized scripting language does not exist among DOD simulations, so any scripting commands taken from scenario files and placed in an XML structure would be arbitrary strings of text to any other simulation or scenario initialization data structure. One possible

approach to addressing this issue would be to utilize data structures similar to the hierarchical task networks (HTN) used by the COMBATXXI simulation application to represent entity-level behaviors (Fitzpatrick, Reeves, & Balogh, 2012).

The research included an initial investigation into storage of data from VBS2 scenario files into MSDL data structures. There is potential for the usual interchange of unit/entity name, location, and orientation/disposition, together with use of the general extensibility capability in MSDL as demonstrated by the RSG project to store portions of the data that do not associate readily with elements in the MSDL data structure. More research is needed to determine how best to employ MSDL, whether directly (transferring scenario data content directly into MSDL structures), as part of the XML representation of VBS2 scenario data (employing MSDL XML data structures in the XML representation through reference to the MSDL namespace and schema), or through transformation, as implied in the data flow shown in Figure 1.

The ideal way ahead for the DOD investing in "games for training" technologies should be close coordination between government requirements managers and standards development organizations, namely SISO. Interoperability and initialization standards must be developed with heavy input from the DOD and then the DOD must establish policy to require adherence to these standards when executing acquisition strategies. As these policy issues are crafted, research and development is needed to identify methods, such as the proof of concept presented in this paper, that will capture past investments in scenarios and preserve them for utilization in future simulations.

ACKNOWLEDGEMENTS AND NOTES

The research supporting this paper was performed at the Naval Postgraduate School's Modeling, Virtual Environments, and Simulation (MOVES) Institute under the guidance and support of Mr. Curtis Blais and Dr. Imre Balogh. The authors wish to thank Mr. Blais for his significant time investment in teaching, mentoring, software development and helping to develop the concepts and scope of the effort, and Dr. Balogh for his revisions and guidance.

The views expressed in this paper are those of the listed authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

REFERENCES

Beris, J. V., & Whittington, E. S. (2008). *Defining stability, security, transition, and reconstruction (SSTR) operations requirements for future Department of the Navy training and analytical models and simulations*. (Master's thesis, Naval Postgraduate School). Retrieved from http://hdl.handle.net/10945/3880

Bohemia Interactive. (2012a). *White paper: VBS2* (Version 2). Retrieved from http://distribution.bisimulations.com/docs/VBS2_Whitepaper.pdf

Bohemia Interactive. (2012b). VBS2 2.0 User Manual Version 2.12.08 [Wiki]. Retrieved from http://manuals.bisimulations.com/vbs2/2-00/manuals/

Bohemia Interactive. (2012c). P3D to FBX Converter [Wiki]. Retrieved from https://resources.bisimulations.com/w/index.php?title=P3D_to_FBX_converter#Known_issues

Bohemia Interactive. (2013). BISim to deliver VBS3 to the U.S. Army [Press release]. Retrieved October 25, 2013, from http://products.bisimulations.com/press-releases/tuesday-july-16-2013-2152/bisim-deliver-vbs3-us-army

Chambers, S. (2015). *Joint and Coalition Warfighting (JCW) / Joint Training Data Services (JTDS)*. Unpublished presentation, Joint Coalition Warfighting (JCW) Environment Development Division, Joint Staff, J7, Suffolk, VA.

Command and Control Systems – Simulation Systems Interoperation Product Development Group. (2014). [data file]. Simulation Interoperability Standards Organization. Retrieved from http://www.sisostds.org/StandardsActivities/DevelopmentGroups/C2SIMPDGPSGCommandandControlSystems.aspx

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Grand, M. (1998). *Patterns in Java, volume 1: A catalog of reusable design patterns illustrated with UML*. New York, NY: John Wiley & Sons, Inc.

Fitzpatrick, C., Reeves, D.E., & Balogh, I. (2012). Use of hierarchical task networks to model complex operational tasks in COMBATXXI. 80th Military Operations Research Society Symposium. Colorado Springs, CO: United States Air Force Academy.

Joint Staff J7. (2014). Joint Live Virtual and Constructive 2020 (JLVC 2020) system operational concept (OpsCon) ISO/IEC/IEEE 29148 (Version 0.998). Suffolk, VA: Joint Staff J7 Environment Development Division (EDD).

Marine Corps Systems Command/ Program Manager–Training Systems Orlando. (2011, October 17). Special notice: Marine Corps intends to award a five year indefinite quantities contract (IQC) on a sole source basis to Bohemia Interactive Simulations Inc. Retrieved from Navy Electronic Commerce Online website: https://www.neco.navy.mil/synopsis/detail.aspx?id=336496

McLaughlin, B. (2001). Java and XML, 2nd edition: Solutions to real-world problems. Sebastopol, CA: O'Reilly & Associates, Inc.

Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing (Special Publication 800-145). Gaithersburg, MD. National Institute of Standards and Technology.

Military Scenario Definition Language Product Development Group. (2008). Standard for: Military Scenario Definition Language (MSDL) (SISO-STD-007-2008). Orlando, FL: Simulation Interoperability Standards Organization. Retrieved from http://www.sisostds.org/ProductsPublications/Standards/SISOStandards.aspx

Ruiz, J., Desert, D., Hubervic, A., Guillou, P., Jansen,R.E.J., Reus, N., Henderson, H.C., Fauske, K.M., & Olsson, L. (2013). In *Proceedings of the Fall Simulation Interoperability Workshop*. Orlando, FL: Simulation Interoperability Standards Organization.

Tolk, A. (Ed.). (2012a). *Engineering principles of combat modeling and distributed simulation*. Hoboken, NJ: John Wiley & Sons, Inc.

Ullner, F., & Lundgren, A. (2008). *Lessons learned from implementing a MSDL scenario editor: A tool for the serious gaming community* (Bachelor's thesis, LTH School of Engineering: Lund University, Helsingborg, Sweden). Retrieved from http://portal.ch.lu.se/Campus.NET/Services/Publication/Export.aspx?id=984&type=doc

Under Secretary of Defense (AT&L). (2006, April 17). Department of Defense acquisition modeling and simulation master plan. Washington, DC: Author. Retrieved from http://www.acq.osd.mil/se/docs/AMSMP_041706_FINAL2.pdf

Under Secretary of Defense (AT&L). (2007, August 8). Modeling and simulation (M&S) management (DOD Directive 5000.59). Washington, DC: Author. Retrieved from www.dtic.mil/whs/directives/corres/pdf/500059p.pdf

Under Secretary of Defense (AT&L). (2013, April 24). Implementation directive for Better Buying Power 2.0 - achieving greater efficiency and productivity in defense spending. Washingtion, DC: Author. Retrieved from http://www.acq.osd.mil/docs/USD(AT&L) BBP 2.0 Implementation Directive (24 April 2013).pdf

Under Secretary of Defense (AT&L). (2015, April 9). Implementation directive for Better Buying Power 3.0 – achieving dominant capabilities through technical excellence and innovation. Washingtion, DC: Author. Retrieved from http://bbp.dau.mil/docs/BBP3.0ImplementationGuidanceMemorandumforRelease.pdf (15 May 2015).pdf