Client Perspective: Implementing C2SIM in a Client

Dr. Robert Wittman
7515 Colshire Drive
McLean, VA 22102
UNITED STATES

rwittman@mitre.org

ABSTRACT

This paper is one of a coordinated set prepared for a NATO Modelling and Simulation Group Lecture Series in Command and Control – Simulation Interoperability (C2SIM). This paper provides a client side application developers perspective when creating a C2SIM application.

1.0 INTRODUCTION

This paper introduces the implementation of C2SIM technologies from a client perspective. Experience from the NATO MSG-085 C2 and Simulation standards-based interoperation effort are used to highlight aspects of the process, tools, data models, and the need and opportunity for applying server-based technologies for additional automation and cost-savings.

The specific topics covered within this section include the following:

- The rationale for using the C2SIM standards;
- A crawl, walk, run approach to implementing C2SIM standards-based processes and technologies;
- A look at a specific instance demonstrated during the NATO MSG-085 effort;
- Lessons learned and associated core capabilities that facilitate a wide range of use-cases associated with C2SIM related applications; and
- Some additional thoughts on the future C2SIM standards efforts.

2.0 BACKGROUND

As a basis for the client-side development perspective that follows I would like to share some of the projects and their contribution to the C2SIM knowledge base. From 2000-2014 I was involved in concept exploration and development of the US Army OneSAF simulation. OneSAF provided an excellent incubator for early MSDL and C-BML concept development.

During this period all of the presenters were active participants and/or leads within the C-BML and MSDL Study and Product Development Groups.

While my fellow presenters were involved with early Exploratory Teams and MSG-048, my participation began at the start of the NATO MSG-085 activity in support of the MSDL related activities; as appointed by the US National Lead Mr. Jeff Gavlinski of the Army Modeling and Simulation Office (AMSO). During this period I participated and led a number of multi-nation initialization focused demonstrations and exploratory efforts.

3.0 BENEFITS OF COMBINED MSDL AND C-BML USE

In addition to the efficiencies and cost savings identified in an earlier presentation for a common transmittal format for military scenario simulation and C2 initialization information, there are a number of advantages of using both the MSDL and C-BML standards. The advantages focus on the economies of scale and cost saving associated with using international standards rather than creating “localized” federate, federation, or coalition data-models and agreements for each new event.

First, by using an open international standard data model and transmittal format, applications, tools, and datasets can be developed for a wide range of users without cost, schedule, and performance impact to each application developer. This is contrasted to non-standardized approaches where, in the extreme, each party in a federation must have intimate knowledge about the other systems to create new capabilities to support point-to-point or one-off localized common data model exchanges.

Secondly, by using a well-defined data model, transmittal format, and a standard way to access the datasets, users can change their internal data models and capabilities as necessary to support end-users without impact to the greater community so long as they respect the standard-based interfaces.

Thirdly, new systems can plan and support the standards without modification. This allows the new systems access to existing datasets and other tools and services that support the standard formats and exchange mechanisms.

4.0 MSDL AND C-BML CONTENTS

As mentioned in the earlier MSDL Section and provided here again for clarity, the MSDL data model holds nine primary elements. These elements describe a military scenario according to the MSDL PDG’s military scenario definition:

The nine primary data elements include [2]:

- Scenario ID– Describes meta data regarding the scenario;
- Options – Describes the parameters to be applied across the scenario;
- Environment – Describes scenario time, extents of the geographic area, and the weather, meteorological and oceanographic conditions;
- ForceSides – Describes Sides and Forces relationships for a scenario;
- Organization – Describes the organizations within a scenario;
- Overlays – Describes collections of tactical graphics and associates them with a particular unit or entity owners;
- Installation – Describes the installations as they stand at scenario start time for the forces, sides, or units;
- Tactical Graphics – Describes the tactical action-based information; and
- MOOTW Graphics – Describes the Military Operations Other Than War action-based information for a scenario.

The OneSAF scenario editing application in the Figure 1 provides a graphical view of the contents of a valid MSDL file import. The pane on the right shows the task organization an equipment elements included within the file as well.

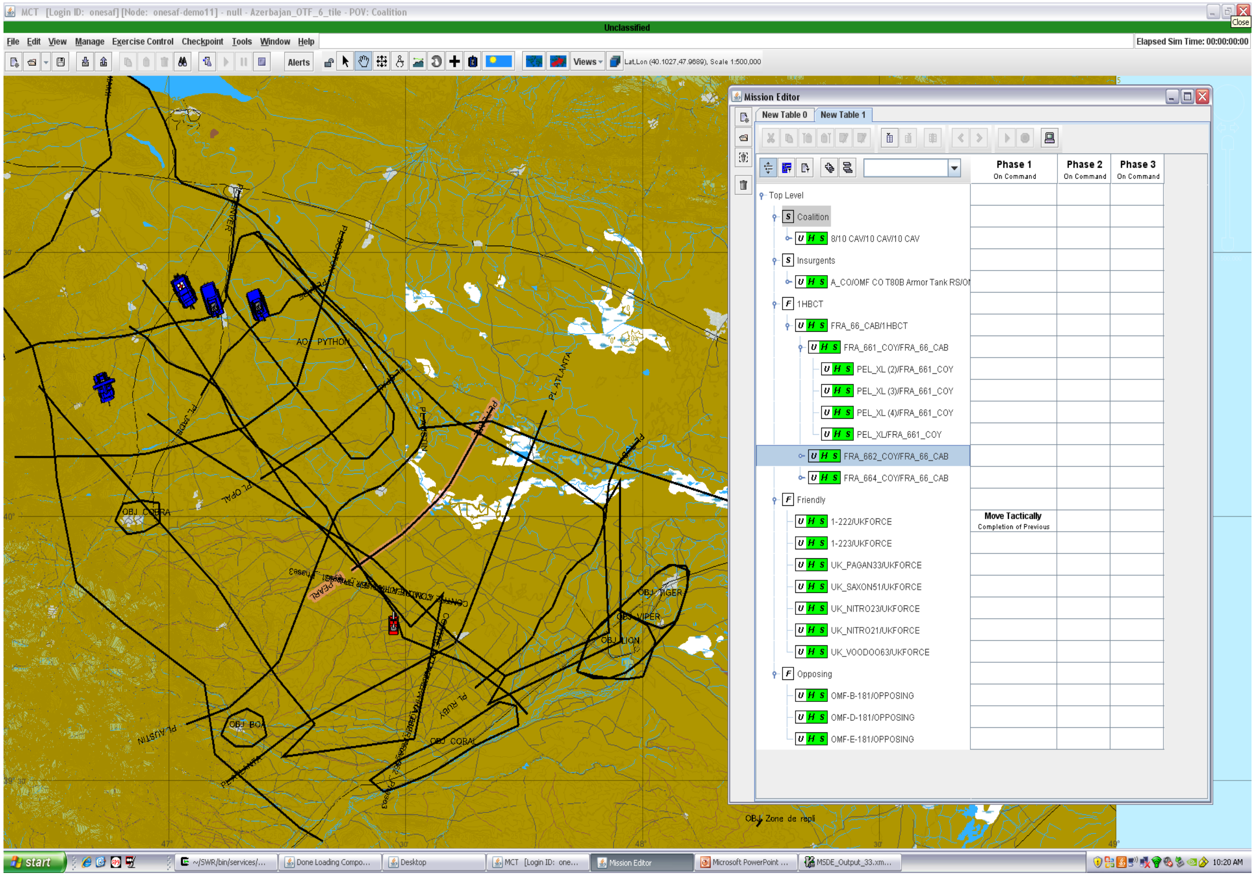


Figure 1: OneSAF Scenario Editing Capability.

As mentioned earlier C-BML provides a formalized data model to hold the who, what, why, where, and when of an activity such that it can be unambiguously exchanged, parsed, and used in support of computer assisted exercises and other simulation-based activities.

This remains consistent with early work by Scott Carey, Martin Kleiner, Michael Hieb, and Richard Brown as described in their Simulation Interoperability Workshop Paper titled, “Standardizing Battle Management Language – A Vital Move Towards the Army Transformation” 01F-SIW-067 where they define BML 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. [1]

Along with this definition, they provided four principles that continue to guide the standardization efforts:

1. BML must be unambiguous;
2. BML must not constrain the full expression of a commander’s intent;

3. BML must use the existing C4I data representations when possible; and
4. 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.

5.0 CRAWL, WALK, RUN DEVELOPMENT

We now move into a discussion of implementing MSDL and C-BML technologies using a progressive: crawl, walk, run implementation approach.

5.1 The Crawl Phase

In alignment with the objective of MSG-085, a subset of the participating nations kicked off with a piloting effort in late 2010 early 2011 timeframe to use MSDL for coordinated scenario initialization across a number of simulations and developers that had no prior MSDL experience.

5.1.1 The Participants

As depicted in Figure 2 the effort was led by the USA with Germany, Spain, Great Britain, France, and Sweden as active participants. The objective was to collaborate on the development of a common MSDL file that would contain common task organization information, initial locations, some simple tactical graphics, and the geographic extents of the playbox.

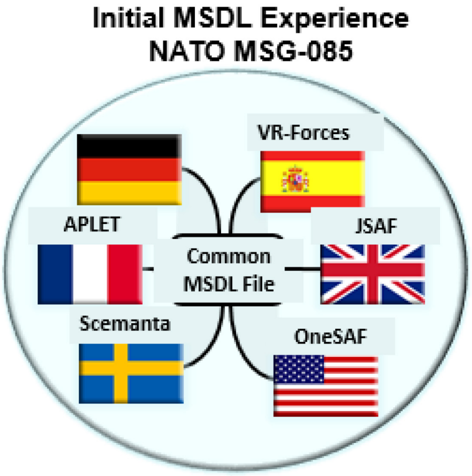


Figure 2: Initial MSDL Experience in NATO MSG-085.

This MSDL file would then be used to provide the scenario initialization data to a number of simulations and C2 devices. A "visual-basic scripted" Microsoft Excel spreadsheet was used as part of the early data collection, integration, and production process.

5.1.2 The Process

The group agreed to a five step process, as listed below on and shown in Figure 3, to coordinate the scenario initialization data. The process relied on manual orchestration and combination of the products, as we will see many of these manual steps were later identified for automation, and thus were incorporated into server-side-services. The first order steps of the process are as follows:

1. Federate X produces a partial MSDL File containing their contributed ORBAT information;
2. The human integrated creates a master "integrated" scenario file;
3. Federates receive the master MSDL file and import into their end systems;
4. Federates modify start-positions and/or other MSDL data (tactical graphics, side force relationships, etc.) and save an updated file;
5. The Federates provide the integrator their updates and the integration process repeats until the MSDL scenario is tuned appropriately.

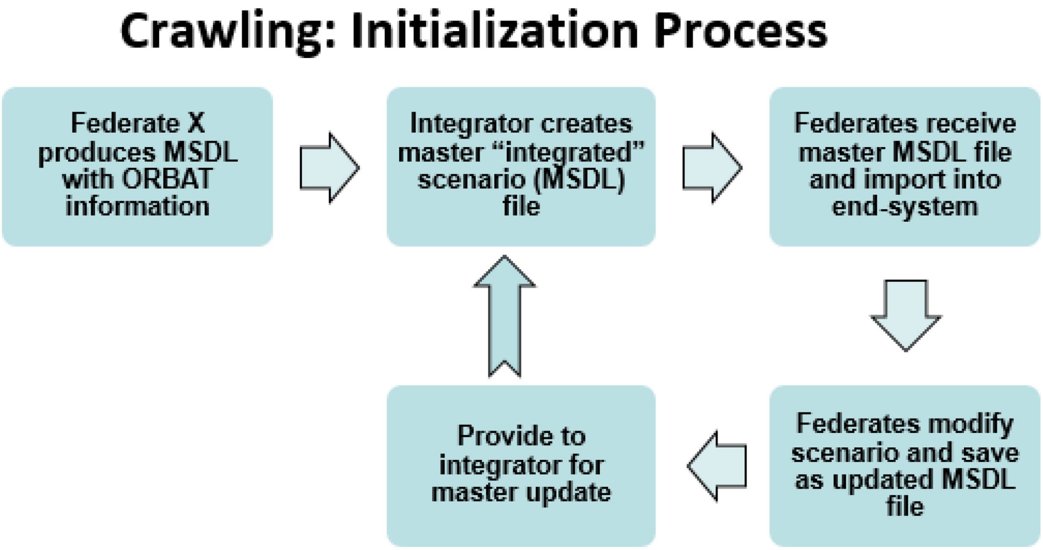

Crawling: Initialization Process
Federate X produces MSDL with ORBAT information
Integrator creates master "integrated" scenario (MSDL) file
Federates receive master MSDL file and import into end-system
Federates modify scenario and save as updated MSDL file
Provide to integrator for master update

Figure 3: The Crawl Phase Initialization Process.

5.1.3 The Integration Tools and Decisions

The integration tools used as part of this initial process were three commonly available commercial or open source tools:

1. Microsoft Excel was use to list, identify, and manage all contributed data. Excel uses a native XML schema format and an MSDL importer and exporter was created using the Excel scripting capability;
2. Notepad++ an open source and freely available tool was used for simple edit, cut-and-paste processes; and
3. Visual Basic Scripting was used to automate multinational Order of Battle (ORBAT) integrations.

Several coordinating decisions became necessary to ensure understanding and appropriate integration of the individual MSDL products: The primary coordinating decision included:

1. XML namespaces to allow extensions to the MSDL file;
2. Common Scenario Identification information and version numbers;
3. Common coordinate system to identify initial placement and geographic extents;
4. Common and inclusive playbox;
5. Sides and force relationships; and
6. Which nations contributed which units and threat forces.

5.2 The Walk Phase

After achieving the initial goals, the group identified a new set of objectives for the "walking" phase. The focus of this activity was threefold:

- Test out a concept to link C-BML data with an MSDL data;
- Post and access MSDL and C-BML from a common repository;
- Import, run and age simulation data in MSDL & C-BML data files; and
- Repost this data for import by other federates.

During this process, the MSDL was linked using the Reference element within the Scenario Identification element to a planned orders and tasks set that referenced the units and equipment within the MSDL file.

As shown in Figure a production federate then posted the MSDL and C-BML files to a repository that could be accessed by the other members of the coalition. A second federate was then responsible to:

1. Import the MSDL and referenced C-BML file;
2. Run the simulation;
3. Export the "aged" data as new MSDL and C-BML files; and
4. Post the information to the common repository.

Other federates were then able to import and execute the new "aged" MSDL and C-BML files.

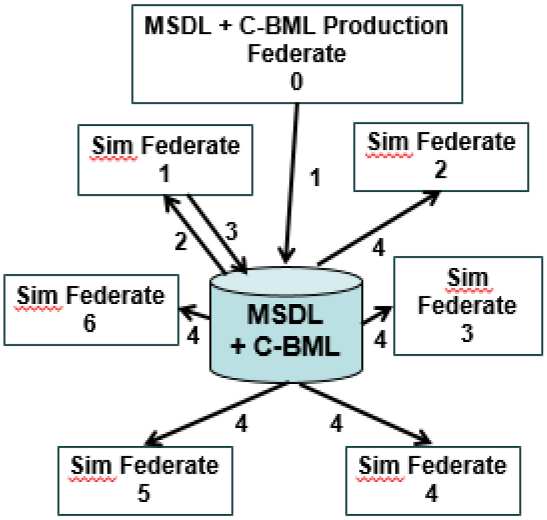


Figure 4: Walk Phase Initialization Process.

5.2.1 Walk Phase Federation Agreements

The "walking" phase also continued the practice of documenting federation agreements. During this phase, the group introduced the following federation agreements:

- Unique identifier range assignments were given to each participant contributing to the scenario initialization information ;
- Standards-based unit and platform naming conventions were established based on the Distributed Interactive Simulation (DIS) standard enumerations;
- Discussed mechanisms to house local client enumeration information. A reference file-based scheme like that use for the C-BML files was prototyped and tried, as was a scheme based on injecting additional elements within the MSDL schema;
- Discussed mechanism to include MSDL and C-BML data within a FOM to support run-time HLA compliance;
- Introduced a mechanism to support non-contiguous terrain extents;
- Agreed to a sequence for interpreting tactical graphic information; and
- Continue to develop integrator tools to support early feedback to scenario initialization data developers.

5.2.2 The Walk Phase Conclusion

The "walking" phase was concluded by:

1. Including MSDL-based initialization of the Joint Advanced Deep Operations Coordination System (JADOCS) with MSDL data;
2. Supporting a web-based interface to the GMU web-services to allow auto import and integration of MSDL information. The GMU server would then publish the MSDL data to all interested subscribers;

3. Extending the scenario information battle book tool to support more automated integration;
4. Providing lessons learned from the MSDL and C-BML integrations to the C-BML and MSDL Product Development Groups.

5.3 The Run Phase

The "running" phase, where we are looking to be during the next C2SIM efforts, are highlighted with an accessible C2 and scalable simulation initialization process as shown in Figure 5. The process will allow for fully integrated and harmonized MSDL and C-BML data; will support a variety of use-cases including training, testing, experimentation, and mission planning and mission rehearsal; and will be able to scale without regard to use case and number of participants.

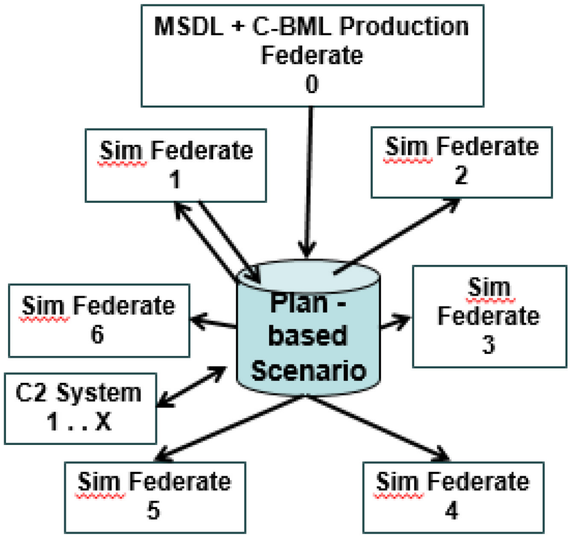


Figure 5: The Run Phase Initialization Process.

At the conclusion of the run phase the fully integrated data set will be supported for demonstration within an operationally useful event. Lessons learned from this activity will be provided to SISO for use within the C2SIM standardization effort.

6.0 A COALITION EXAMPLE

The following subsection provides a coalition "walking" Phase example.

At the 2013 International Training and Education Conference (ITEC) the C2SIM technology set was demonstrated as part of a Sweden, Great Britain, and a USA Air and Land Vignette demonstration. The objective of the demonstration was to show the utility of the C2SIM standards in supporting Coalition Command and Staff mission planning and mission rehearsal training.

The vignette as graphically depicted in Figure 6 starts with a ground force reconnaissance unit cueing Unmanned Air Vehicles (UAV) and a Close Air Support CAS mission. The UAV mission then provides additional reconnaissance of threat forces which are sent to drive the Commander and Staff decision-making process. They respond with specific tasking for the simulated subordinates using C-BML where possible.

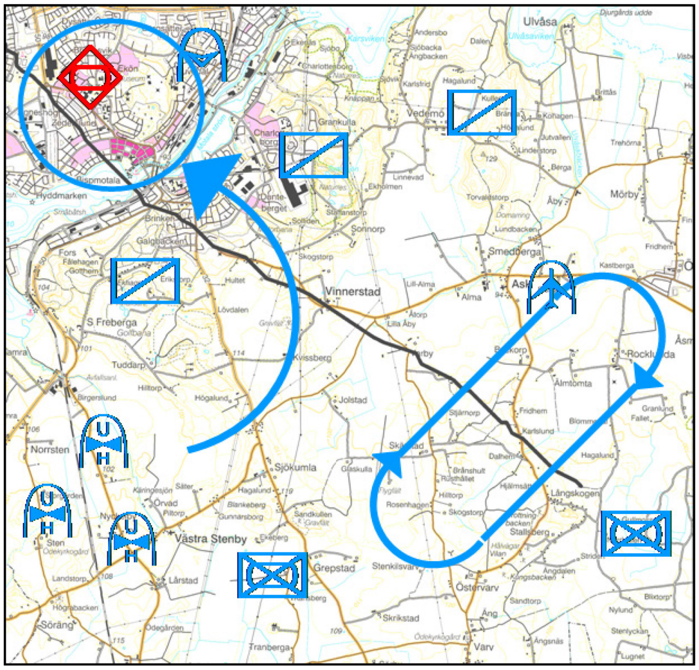


Figure 6: Air-Land Vignette.

For this event, both MSDL and C-BML data files were used to initialize the simulations and C2 devices. The simulations used to drive the event included OneSAF (USA) and JSAF (UK). The C2 devices in use included a Swedish ground maneuver system called 9LandBMS (providing orders to OneSAF and displaying SA from reports sent by all simulations), ICC, and JADOCS (provided air tasking to JSAF and OneSAF). The MSDL and C-BML files were integrated and distributed using the WISE (Saab)/SBML server over a wide area network. The WISE/SBML server also provided a capability to translate between different and evolving variants of the pre-balloting C-BML schema.

6.1 The Technical Architecture

This architecture used in support of this widely distributed demonstration is shown in Figure 7 with the GMU/Saab server at the center of the slide with simulation and C2 clients accessing the MSDL and C-BML publish and subscribe services. All of the services were made available by the WISE/SMBL server through a Virtual Private Network tunnel over the internet.

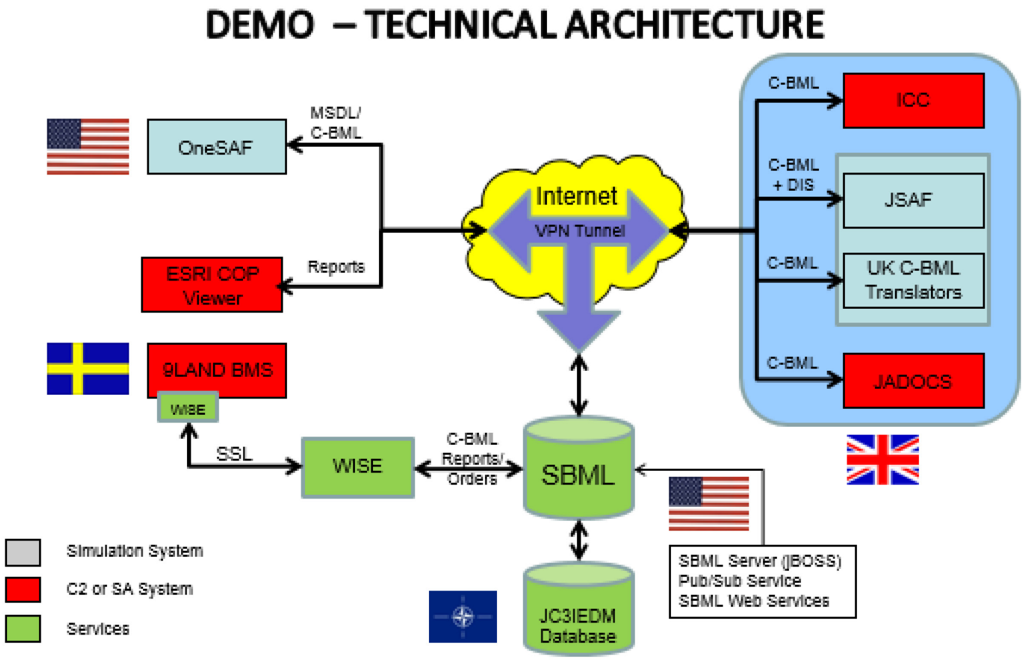


Figure 7: Technical Architecture.

For the demonstration, the items on the left side of the SBML server were executing locally on-site and the demonstration event. These applications included:

- OneSAF, which initialized with a common MSDL and linked C-BML data set and received ground and air forces orders and reports from the server. OneSAF simulated all friendly and threat forces with the exception of CAS aircraft simulated with JSAF.
- ESRIs Common Operational Picture which also received and graphically displayed C-BML reports; and
- Saab's 9LandBML provided a ground maneuver display using C-BML reports and also provided ground orders to OneSAF units using C-BML formatted orders.

Those elements in the blue box were run in the UK and included ICC for an integrated command and control view based on C-BML reports. JSAF which was initialized using MSDL and simulated both ground and air forces. JSAF also use C-BML translators for sending and receiving orders and reports. Finally JADOCS was used to create and provide air tasking in C-BML format.

6.2 The OneSAF Client Side

Figure 8 depicts of a number of resulting client-side (OneSAF) capabilities and tools that were prototyped during within the crawl, walk, run phases.

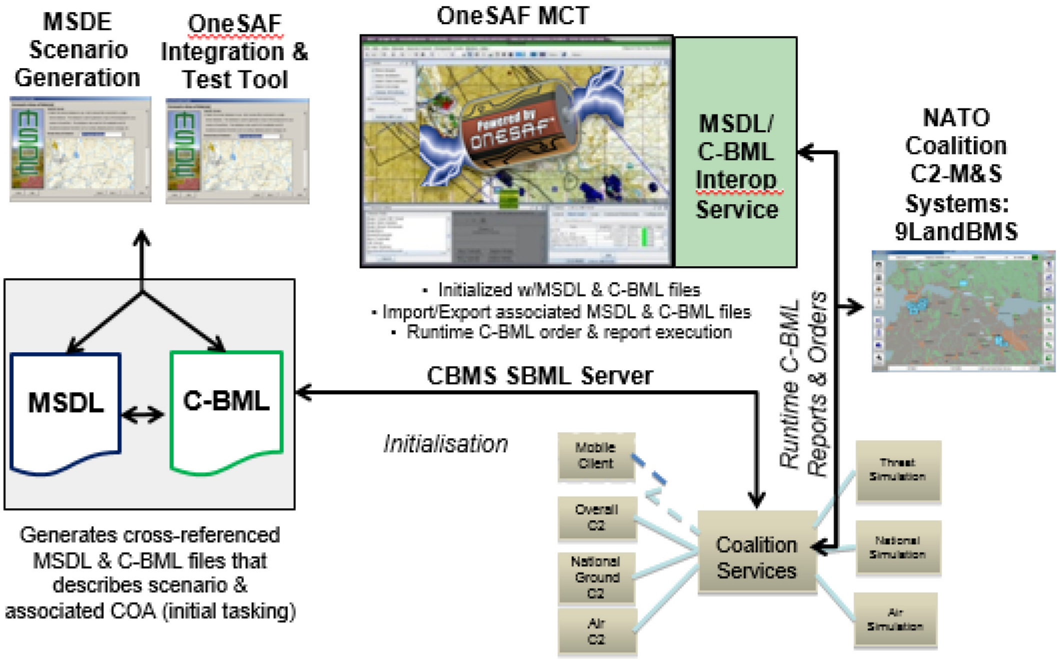


Figure 8: The OneSAF Client Side.

The tools include:

- Scenario Generation Tools: The OneSAF Military Scenario Development Environment (MSDE), a Microsoft Powerpoint-based tool, allowed the creation, import, modification, and export of OneSAF and standards-based MSDL files. It was used early on as a test harness to ingest and merge MSDL data files.
- The OneSAF Management and Control Tool (MCL) was enhanced to create valid MSDL files and referenced valid C-BML and BML variant type files as well as a host of other OneSAF specific referenced files for electronic order of battle, supply information, and internal OneSAF behavior tasking files.
- The OneSAF simulation was enhanced to support MSDL and C-BML publish and subscribe mechanisms necessary to communicate with the GMU SBML Coalition services.
- The OneSAF simulation was used to import ground, air, and naval orders published via C-BML from surrogate C2 systems as well as the 9LandBML system.

7.0 CLIENT SIDE LESSONS LEARNED

This section discusses the lessons learned and impact to requirements and client side design.

7.1 Simulation Support Requirements for Operational Use

From the client-side perspective OneSAF's evolution is in direct response to lessons received from active participation within the DARPA Deep Green program; a number of US Simulation and Mission Command

Interoperability (SIMCI) projects; and the NATO MSG-048 and MSG-085 projects. From participation in these projects Government and MITRE engineers on the OneSAF team identified a number of critical requirements for simulations to support embedded Mission Command applications across a range of Mission Planning/Mission Rehearsal; training; testing; and experimentation. [3]

The required simulation services derived from active participation in these projects included:

- fully automated behaviors that are initiated based on command level orders,
- faster than real-time execution to support the timelines associated with mission planning cycles,
- setup, execution, and control transparent to the operational user,
- ability to base the simulation start point on Mission Command data,
- user selectable branch points and optimization criteria,
- command selectable reporting and running estimates,
- comparison of plans with actual execution,
- easy separation and identification of simulation and real-world data,
- opposing force initialization and behavior representations, and
- a broad range of warfighting functional area representations.

The team then began implementing capabilities supporting these requirements. While not all of these services are completely functional within OneSAF, their initial limited versions allowed for prototyping integrated end-to-end mission planning and mission rehearsal operational capabilities. As such, OneSAF proved very useful in supporting the MSG-085 activity. Through participation in MSG-085 a number of additional insights regarding multi-national C2 coalition simulation were identified and include:

- Applying SISO standards for simulation initialization (MSDL) and for order, report, and request transmission (C-BML) reduce development and integration costs when operating in a single nation or coalition mission planning/rehearsal execution.
- Common supporting web-enabled infrastructure for publish and subscribe, persistence, multi-server bridging, and merging of initialization and order and report service eases cross coalition-based integration.
- Defining specific coalition initialization and order, report, and request agreements reduce development, integration, and rework costs when developing and executing mission planning activities within a federation environment.
- Loose coupling of coalition resources is necessary to allow multi-nation simulation with Mission Command (MC)/C2 applications.
- Web-enabled access to simulation and MC and C2 systems is necessary to allow distributed access to mission planning/mission rehearsal assets.

Several US projects currently are underway, extending OneSAF's foundational building blocks to extend and more fully support integrated simulation services within the Mission Command computing environment.

7.2 MSDL Extension Architecture

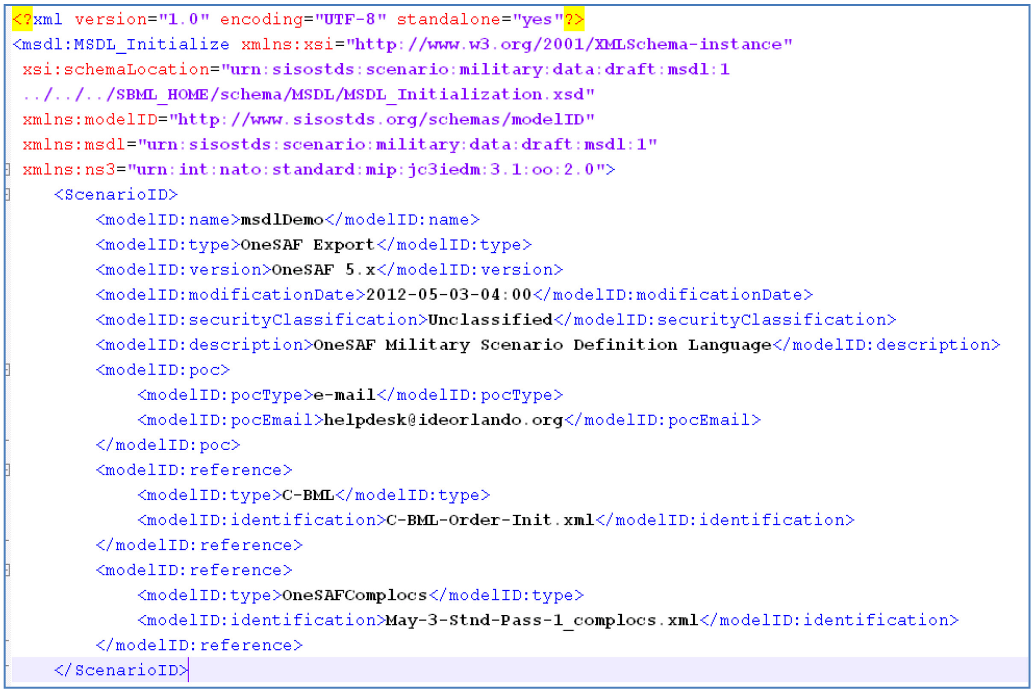
A formalized and standard MSDL extension architecture is critical to the long-term viability of MSDL. Several interesting and potentially viable design alternatives have been proposed and debated at SISO via papers and presentations. Other designs have been implemented within existing simulation programs.

For example, early versions of OneSAF used a modified MSDL schema to allow local data extensions while still allowing schema validation. The OneSAF MSDL implementers inserted an xs:any element after all complex elements to allow new elements to be appended and validated at the tail of existing complex elements. This allowed them to insert local extensions conforming to non-MSDL schemas that could then be shared when necessary across federations for data import and validation. While this worked for OneSAF-only federations, there were some important drawbacks to this approach when employed in a larger more complex federated environment:

- First, it necessitates a change to the MSDL standard schema to include the xs:any element after all complex elements.
- Secondly, the xs:any attribute precludes the use of the xs:all compositor, used liberally within the MSDL schema, and instead requires the xs:sequence compositor. This change potentially invalidates existing MSDL instance documents (elements within the xs:all compositor can be in any order and those within a sequence compositor must be in specific order) and requires evolving those data sets to a new standard.

The extension design proposed and used within MSG-085 with the previous example is significantly different as it leverages and appropriately extends existing elements within the Version 1.0 standard MSDL schema. It also is consistent with the coordinated and general consensus agreement by the MSDL Product Development Group (PDG) and the Coalition-Battle Management Language (C-BML) Group made during the Spring 2011 PDG meetings.

Figure 9 below shows a valid MSDL standard “ScenarioID” element, that holds the fields within the http://www.sisostds.org/schemas/modelID namespace, populated with scenario description data and a reference element identifying a C-BML reference and a locally defined OneSAF composition locations mapping file reference.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<msdl:MSDL_Initialize xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:sisostds:scenario:military:data:draft:msdl:1
 ../../../SBML_HOME/schema/MSDL/MSDL_Initialization.xsd"
 xmlns:modelID="http://www.sisostds.org/schemas/modelID"
 xmlns:msdl="urn:sisostds:scenario:military:data:draft:msdl:1"
 xmlns:ns3="urn:int:nato:standard:mip:jc3iedm:3.1:oo:2.0">
    <ScenarioID>
        <modelID:name>msdlDemo</modelID:name>
        <modelID:type>OneSAF Export</modelID:type>
        <modelID:version>OneSAF 5.x</modelID:version>
        <modelID:modificationDate>2012-05-03-04:00</modelID:modificationDate>
        <modelID:securityClassification>Unclassified</modelID:securityClassification>
        <modelID:description>OneSAF Military Scenario Definition Language</modelID:description>
        <modelID:poc>
            <modelID:pocType>e-mail</modelID:pocType>
            <modelID:pocEmail>helpdesk@ideorlando.org</modelID:pocEmail>
        </modelID:poc>
        <modelID:reference>
            <modelID:type>C-BML</modelID:type>
            <modelID:identification>C-BML-Order-Init.xml</modelID:identification>
        </modelID:reference>
        <modelID:reference>
            <modelID:type>OneSAFComplocs</modelID:type>
            <modelID:identification>May-3-Stnd-Pass-1_complocs.xml</modelID:identification>
        </modelID:reference>
    </ScenarioID>
```

Figure 9: An MSDL Schema Snippet.

The OneSAF Management and Control Tool (MCL) was enhanced to create valid MSDL files and referenced valid C-BML and BML variant type files as well as a host of other OneSAF specific referenced files for electronic order of battle, supply information, and internal OneSAF behavior tasking files.

The ability to export MSDL standards-based files that reference localized files containing orders, report-based perception information, and supply data supports a powerful scenario construction concept. It allows the same task organization and playbox to be used and combined with pre-existing set of orders, perception-based information, supply information, etc. that are stored and accessible in these referenced files.

To support the multiple file generation/export capability, a rigorous set of reporting functions should be available to allow the user to understand the contents and use the exported data.

7.3 Client Side Merge Capabilities

In progressing through the crawl, walk, run Phases, it became clear that a robust MSDL and C-BML merge capability is required to allow the different scenario contributions, made by different participants, to mature and integrate at the appropriate pace. It is also critical for the merge capability to have undo capability as well as an ability to report and track all merges as is commonplace with a wide variety of software configuration and management control tools.

During this prototyping process some of the fundamental merging requirements were identified and implemented but many of the specific collaboration-based rules that guide the merging process still need to be identified, modeled, and implemented.

7.4 Server Side Automation

As mentioned earlier a common supporting web-enabled infrastructure as was provided by the GMU SBML server for publish and subscribe, persistence, bridging, and merging of initialization data sets eases cross coalition-based integration. This is essential for rapid coalition construction and execution. It also provides an integrated platform for addition coalition-wide management and control tools as will be discussed within the infrastructure section that follows.

8.0 FUTURE C2SIM STANDARDS DEVELOPMENT

This section provides some thoughts on future C2SIM architecture standards and best practices.

The next generation C2SIM environment is evolving rapidly and benefits from contributions across the M&S, C2, and wider web-enabled community. As such the new C2SIM standard must take advantage of emerging concepts across a variety of domains to include:

- Data-modeling standard and best practices to ensure a flexible and extensible design that allows new areas to be standardized as they are ready;
- Web-enabled infrastructure standards such as HTML5 and the support pf service-based capabilities;
- Open standards such as the Open Services Gateway initiative allowing services bundles to be deployed to support capability-focused service mixing;
- Possible support for easy and extensible runtime format creation to instantiate the logical data model in an XML, JSON, FOM, etc. format; and
- An ability to transform existing MSDL and C-BML data sets in to the new formats.

It is also critically important that the standardization work is supported with operational real-world experience and feedback. SISO’s relationship with NATO provides an important link to provide operational feedback to the standards development process as supports a feed into the C2SIM STANAG creation and publication process.

9.0 REFERENCES

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

[2] Simulation Interoperability Standards Organization, Standard for: Military Scenario Definition Language (MSDL), 2008

[3] Wittman, R., “OneSAF as an In-Stride Mission Command Asset,” International Command and Control Research and Technology Symposium 2014, Alexandria, VA