Command and Control System to Simulation System Interoperation: Development of the C2SIM Standard

Dr. J. Mark Pullen, Douglas Corner, and Dr. Samuel Singapogu
George Mason University C4I Center
Fairfax, VA 22030
703-993-3682, 703-993-3682, 703-993-6082
mpullen@c4i.gmu.edu, dcorner@c4i.gmu.edu, ssingapo@c4i.gmu.edu

Dr. Curtis Blais
MOVES Institute
Naval Postgraduate School
Monterey, CA 93943
831-656-3215
clblais@nps.edu

Dr. Douglas Reece
VT MAK
150 Cambridge Park Dr., 3rd Floor
Cambridge, MA 02140
dreece@mak.com

Jim Ruth
Trideum Corporation
1000 4th Street, Suite C
Leavenworth, KS 66048
jruth@trideum.com

Keywords:
Interoperability, command and control, modeling and simulation, reference architecture, C-BML, MSDL, C2SIM

ABSTRACT: *A Simulation Interoperability Standards Organization (SISO) Product Development Group (PDG) is developing a standard for Command and Control System to Simulation System Interoperation (C2SIM). The C2SIM standard will improve upon and replace the current SISO standards Military Scenario Definition Language (MSDL; SISO-STD-007-2008) and Coalition Battle Management Language (C-BML; SISO-STD-011-2014). Current work in the PDG is specifying a core logical data model ontology addressing key areas of C2-to-Simulation interoperation such as initialization and tasking/reporting as well as transformation of the ontology data model to an Extensible Markup Language (XML) schema for implementation. In addition, the PDG has defined initial extensions to the core ontology to represent general military concepts and concepts particular to land operations suitable for implementation, demonstration, and evaluation of the standard in a distributed environment. NATO has a Modeling and Simulation Group Technical Activity that is supporting these activities. This paper describes the technical approach and status of development of the C2SIM standard, describing pre-ballot products, a reference implementation, and planned evaluation activities.*

1. Introduction

The military command and control (C2) community and the modeling and simulation (M&S) community have worked together for many years with the goal of achieving effective information exchange across C2 systems and M&S systems. There is a long-standing requirement for effective interactions to support live/virtual/constructive training, exercise replay, mission planning, mission rehearsal, and mission re-creation. During those years, information technology and knowledge representation have progressed to provide new and improved techniques for enabling computer systems to better understand and process information.

For nearly two decades, the Simulation Interoperability Standards Organization (SISO) has been active in this area, researching and developing standards for interoperability among C2 systems and M&S systems. A SISO Product Development Group (PDG) currently is developing a new standard for Command and Control System to Simulation System Interoperation (C2SIM), improving upon and replacing the existing SISO standards Military Scenario Definition Language (MSDL; SISO-STD-007-2008) [1] and Coalition Battle Management Language (C-BML; SISO-STD-011-2014) [2]. Current work in the PDG is specifying a core logical data model ontology addressing key areas of C2-to-Simulation interoperation such as initialization and tasking/reporting as well as transformation of the ontology data model to an Extensible Markup Language (XML) schema for implementation [3]. In addition, the PDG has defined an initial extension to the core ontology to represent general military concepts and an extension to represent concepts particular to land operations. The core and extensions are suitable for implementation, demonstration, and evaluation of the standard in a distributed environment. NATO Modeling and Simulation Group Technical Activity 145 (MSG-145) is actively supporting these activities.

This paper describes the technical approach and status of development of the C2SIM standard, describing pre-ballot products, a reference implementation, and planned evaluation activities.

2. Existing SISO Standards for C2 System to Simulation System Interoperation

As introduced above, the C2SIM standard will replace two existing SISO standards, the Military Scenario Definition Language (MSDL) and the Coalition Battle Management Language (C-BML). The following subparagraphs provide brief overviews of each standard. Refer to the standard documents [1] [2], C-BML Guide [4], and associated XML schema files for full information about these standards.

2.1 Military Scenario Definition Language

The Military Scenario Definition Language (MSDL) provides a starting point for interactions between C2 systems and M&S systems by providing a common representation of essential elements of the battlespace or, more broadly, the operational situation. The standard specifies an XML schema structure for MSDL documents that conform to the standard. Principal concepts in the structure include:

- ScenarioID (mandatory) – provides identification of the scenario and its purpose.
- Options (mandatory) – provides global parameters about the scenario and its content.
- Environment (optional) – describes the simulated physical environment in which the execution is to occur (e.g., area of interest, weather, time).
- ForceSides (mandatory) – describes the structure of the forces and sides involved in the execution.
- Organizations (optional) – describes the structure of the units and equipment involved in the execution.
- Overlays (optional) – describes the logical overlays used to group the intelligence elements/instances in the scenario. Ownership of a specific overlay is determined through the intelligence elements/instances contained in that overlay.
- Installations (optional) – describes the detected installations as determined by the intelligence gathering process of each force, side, or unit individually.
- TacticalGraphics (optional) – describes the tactical information as known by a particular force, side, or unit individually.

- MOOTWGraphics (optional) – describes the detected MOOTWGraphics (Military Operations Other Than War) instances as determined by the intelligence gathering process by each force, side, or unit individually.

2.2 Coalition Battle Management Language

C-BML provides a structured format for tasking military forces and for reporting on results of that tasking for data interchange across C2 systems, simulation systems, and robotic and autonomous systems (RAS). The Joint Consultation Command Control Information Exchange Data Model (JC3IEDM), now known as the Multilateral Interoperability Programme (MIP) Information Model (MIM) [5], formed a major part of the conceptual basis for the C-BML data model. Also, the standard used common who, what, when, where, and why (5W's) constructs as the basis for its expressions. Both of these conceptual foundations differed significantly from the MSDL data model. However, like MSDL, the C-BML standard specifies an XML schema structure for expressing orders and reports in conformance with the standard. Figure 2.1 shows the high-level XML structure of a C-BML order/task (note the XML elements relating to the 5W's).

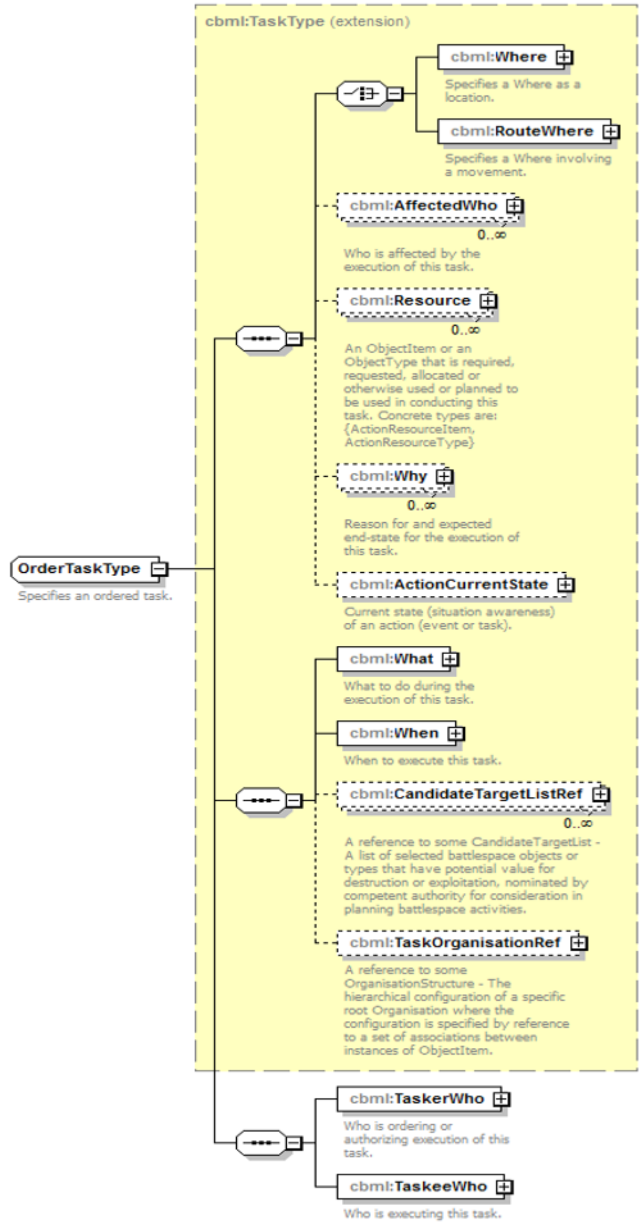

Figure 2.1: Top-level Schema Structure for the C-BML OrderTaskType.

As indicated above, one shortcoming of C-BML is that it is not well integrated with MSDL when considering the conceptualizations found in each data model. It also does not provide standardized message constructs, only components to be used in defining tasking and reporting messages. Like JC3IEDM, C-BML was focused primarily on maneuver warfare and land operations. Users would need to extend the model significantly to address other warfare domains. For these reasons, the SISO community determined that a more flexible and extensible second-generation standard is needed to overcome these limitations.

3. Command and Control System to Simulation System Interoperation Product Development

SISO commissioned the C2SIM Product Development Group (PDG) September 2014 to work towards the goals outlined in the C2SIM Product Nomination [6]. The PDG quickly organized into three groups, each led by a vice-chair, to work on the logical data model, initialization, and tasking-reporting. The PDG also collected a number of use cases across the C2 and simulation communities for setting the scope of the C2SIM and the core data model. The core data model developed progressively to cover common classes needed by C2SIM applications. The initialization and tasking-reporting sub-groups worked on assimilating work from MSDL and C-BML and aligning it with the goals, progress, and vision of C2SIM. This resulted in a Unified Modeling Language (UML) diagram for the core data model and three standard documents as the artifacts produced by the PDG. The larger PDG group was kept informed every month of progress and direction in the three sub-groups. In addition, the PDG and NATO MSG-145 provided regular and valuable review of the work evaluating the core data model and the standard documents.

In the spring of 2017, after much evaluation and deliberation, the core data model team found limitations in modeling C2SIM data elements using UML. The group identified a number of reasons to move to defining data elements in a stronger semantic format (ontology) to formalize C2SIM semantics, leverage hierarchical representations of data, and move in the direction already set by earlier research and recommendations [4][5][6]. The PDG reviewed this proposal and validated the move from a UML representation of the core data model to an ontology representation of the data elements. In addition, the PDG also validated the need to merge the three sub-groups in order to have a unified data model for C2SIM and unified standard document that outlines the C2SIM standard. Transforming the core data model into a semantic format created the first draft of the core ontology defining a set of classes, datatype properties, and object properties that are necessary and useful to C2SIM applications. An ontology sub-group created within the PDG reviewed the ontology for completeness and correctness (to be clear, however, the ontology has not yet been fully specified; refer to [10] for a more complete discussion). The PDG and NMSG-145 has reviewed the ontology and offered important review recommendations and changes. As part of the review, the PDG has validated the need for the core ontology to contain classes necessary across C2SIM applications and for a set of classes that cover generic military applications. The classes that cover generic military applications is packaged as a Standard Military Extension (SMX) that imports the core ontology. Section 4 of this paper presents an overview of the core ontology. Section 5 discusses the SMX ontology. The PDG has outlined rules and methods to create other C2SIM extensions and has prepared the Land Operations Extension (LOX) as an example and artifact of C2SIM extensions useful to create orders, tasks and reports for C2SIM messaging in that domain. Section 6 describes the LOX extension ontology. Section 7 describes a reference implementation based on the C2SIM core ontology, extensions, and messaging.

4. C2SIM Core Ontology

The core data model is intended to provide the means to express the content of messages sent between any and all C2 and simulation systems. These messages include both simulation initialization and operational C2 information, so the data model must include elements that define battlefield objects, tasks, and observations. The data model also is intended to define the structure of messages sent for initialization and C2. To support these goals, the data model has a high-level organization with three parts:

- MessageConcept—defines the structure of all C2SIM messages and the sub-structure of messages in the C2 domain (e.g., orders and reports).
- InitializationConcept—defines complex information elements contained in initialization messages.
- C2SIMContent—defines information elements in the C2SIM domain including objects, actions, and physical attributes.

The PDG defined the core data model as an ontology using the W3C Web Ontology Language (OWL) [11]. The use of a formal ontology provides a hierarchical class structure for the data model and will support a capability for knowledge interchange and inference in the future [10]. The PDG constructed th ontology using the Protégé ontology editing tool [12], which supports graphical editing and multiple output formats, including the Resource Description Framework (RDF) [13] XML (rdf/xml) format (see [3] for discussion of the use of this format in C2SIM). The figures in this paper showing the ontology elements are taken from the Protégé user interface display.

Figure 4.1 shows the top levels of the core data model ontology. Each name in the figure is an OWL class, and the indentations show the subclass hierarchy. Several

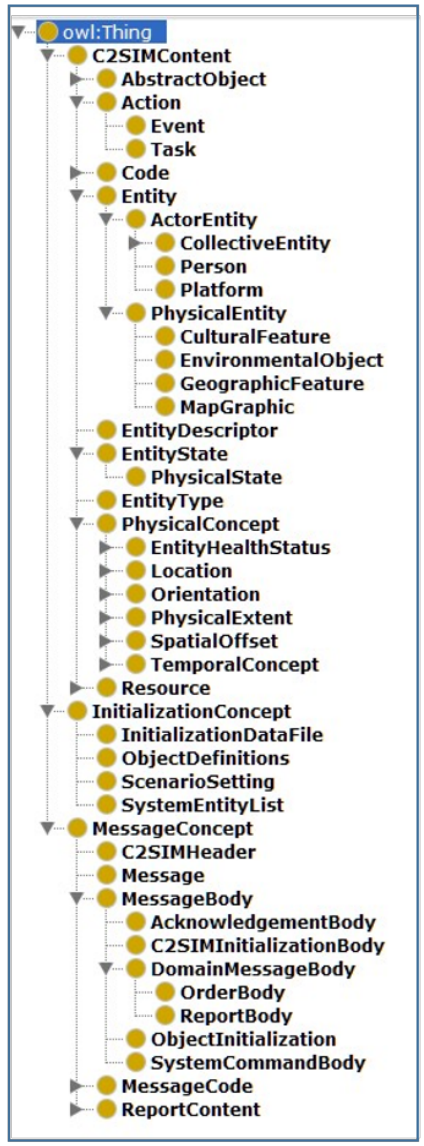

Figure 4.1: Class Hierarchy in the C2SIM Core Data Model Ontology.

of the most important subclasses have been expanded in the figure (e.g., Entity class).

The C2SIMContent class is a collection that includes Actions, Entities, and PhysicalConcepts. Actions include Tasks, which are the key part of orders. Actions also include Events, which are named moments in time that may be referred to by other elements. Entities include Actors, which can perform tasks, issue orders, and send reports, and non-actor PhysicalEntities. PhysicalEntities have a location, an extent, and possibly other physical properties. Actor Entities also may have a location and extent, but in some cases they may not have these properties. For example, a higher-level command unit may not have a physical presence in the bounds of the battlespace.

The key subclass in the MessageConcept collection is the Message class. All members of the Message class have a C2SIMHeader that includes a subset of the elements identified in the Agent Communication Language (ACL) Message Structure Specification [14] produced by the Federation for Intelligent Physical Agents (FIPA) [15]. Messages also include content, which is defined in the MessageBody class. MessageBody has subclasses for several types of messages that are exchanged between systems during initialization (see Section 7), and a subclass DomainMessageBody that has subclasses for the C2 "domain" messages. In principal, domain messages are exchanged between Actors, not just systems, so all of these messages include sender and receiver elements identifying these Actors.

The InitializationConcept class has several subclasses that describe the information exchanged in initialization messages. The ObjectDefinitions class includes an unbounded number of Action, Entity, or AbstractObject individuals; this is the container that holds the definitions of all of the units, graphics, and other information needed to initialize a scenario (similar to the use of MSDL). The ScenarioSetting class is defined by properties providing information like start time, scenario version, and geographic extent for the scenario. SystemEntityList is a mapping of Actors to their system locations—in other words, the name of the simulation system that owns them, in most cases. Finally, the InitializationDataFile class has a file name and a system name, and supports the use of system-specific initialization files (e.g., terrain files) as part of the scenario definition.

5. Standard Military Extension

The PDG designed the core data model to be generally applicable to a variety of domains that might use C2 processes or systems, such as military, law enforcement, or emergency management. However, the PDG recognizes that initial users are likely to be from the military domain. In order to support different military users without necessitating the re-invention of data elements that are common to all of these users, the PDG developed a Standard Military Extension (SMX) to the core data model that includes military-oriented information elements. Military extensions, such as the Land Operations Extension (see Section 6 below), will start from the SMX ontology. Using the Protégé tool, these layered ontologies are integrated into a single consolidated ontology, providing a very practical way to achieve the extensibility that is required in C2SIM. The process of transforming the consolidated ontology to an XML schema that describes the complete set of C2SIM messages is described in [3].

Figure 5.1 shows the SMX ontology as displayed in Protégé. The bold-face type shows where the SMX changes or adds to the core class hierarchy. For example the SMX adds the concept of ForceSide to the AbstractObject class. In the EntityType class, which is a simple string in the core, the SMX adds subclasses to support representation of entity types

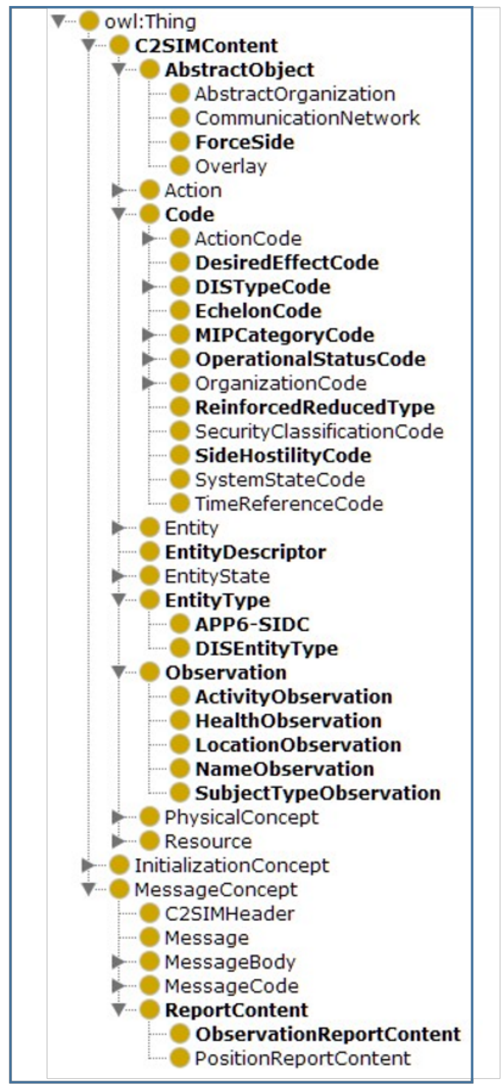

Figure 5.1: Class Hierarchy for the C2SIM Standard Military Extension Ontology (bold-face type indicates changes or additions to the core ontology).

using Distributed Interactive Simulation (DIS) types [16] or NATO APP6 codes [17]. The Code class (enumerated values) includes several enumerations to support these types. The SMX adds an ObservationReportContent class, as well as an Observation class, for sending more complex force information in reports.

6. C2SIM Domain Extensions

As shown with the standard military extension, users extend the core model ontology by creating a new ontology that *imports* the core and then adds (1) new classes or new subclasses to classes defined in the core; (2) new datatypes; or (3) new data or object properties. Protégé can perform such an import operation and displays the additions within the imported ontology. The core is not changed in the process of creating such an extension; therfore, two independent extensions can be developed at the same time and not interfere with one another.

The extension process can be carried out more than once, such that a new extension ontology can import an existing extension ontology and thereby make use of the classes in both the core and the existing extension. The intent of building these extensions is to allow C2SIM users to have the smallest possible implementation using only the portion of the ontology that is directly pertinent to their use.

The PDG anticipates that some scenarios may require the use of multiple extensions that were developed independently; e.g., an Air Operations Extension along with a Land Operations Extension (LOX). Users create a combined extension by importing first one and then the other extension, in either order. Conflicts in either names or differing implementations of the same capability would have to be resolved manually, although future use of an automated reasoner in a tool like Protégé will help identify such conflicts. The PDG anticipates that C2SIM users will wish to develop and standardize extensions, and that the C2SIM PSG will review each proposed extension to make sure that it does not conflict with the core nor with existing extensions.

As an example, the C2SIM PDG has built a LOX on top of the SMX. Figure 6.1 shows the key parts of the LOX ontology. The main addition to the SMX is a set of classes to support plans. The PlanPhase class defines one phase of a plan for one or more units. PlanPhases recursively may contain other PlanPhases so that plans may be described for different echelon levels at the same time. PlanPhases are started by PlanPhaseTriggers which may be Events, the completion of another PlanPhase (thereby defining a sequence), or an order.

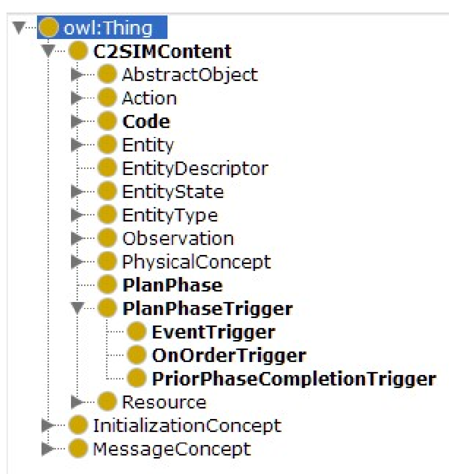

Figure 6.1: Class Hierarchy for the C2SIM Land Operations Extension Ontology.

7. C2SIM Reference Implementation

The C2SIM standard is intended to support a coalition of systems that is assembled as needed for a particular use. Each of these systems may bring part of a scenario definition to the coalition and, in addition, there will likely be global information provided by scenario designers. As a practical matter, it is expected that there will be some master controller element to synchronize the initialization of the scenario; or, as part of the master controller, it will be useful to have a server system. The server can implement initialization functions so that each C2SIM system (that is, a C2SIM interface attached to a C2 or simulation system) does not have to implement all of these functions. The server can also be the source of global scenario initialization data such as terrain database configuration information or a map of simulation objects and their host systems. This section describes the initialization process in more detail on a reference implementation that assumes a control console that can issue commands and a server that assembles and shares the initialization data. The section then describes additional server functions that have proven useful in C2-simulation interoperation experiments.

7.1 C2SIM Scenario Initialization Process

A C2SIM scenario is initialized using the messages described above and whose message body classes were shown in the core data model ontology class hierarchy (Figure 4.1). The SystemCommandBody class has four individuals defined in the core ontology, representing four system commands: SubmitInitialization, ShareScenario, InitializationComplete, and StartScenario. The approach here assumes that there is a master controller that can inject messages into the C2SIM network, and a server to marshal initialization data.

The C2SIM systems, including the server, start in an Uninitialized state. Initialization is accomplished through the use of initialization messages in the following sequence:

SubmitInitialization
A controller entity sends this message to C2SIM systems. When those systems receive this message, they should submit their database of object definitions. In the reference implementation, the marshalling system is the C2SIM server but that won't necessarily be the case. The system name will be in the message, rather than assuming that the sender of the message is the recipient.

ObjectInitialization
This message is sent by a C2SIM system in response to a SubmitInitialization message. It is sent to the system indicated in that message. Any system receiving this message (i.e., the server) will add the definitions to an internal list of submitted definitions. This message will contain object definitions with elements in the quantity indicated by the cardinality in the ontology (e.g., optional, 1, 0 or more, or 1 or more). Objects will include entities—units, graphics, etc.; tasks; scenario definition data such as playbox extent; maps of object identifiers (IDs) to system names; and, for military extensions, plans. It is expected that the C2 systems will provide most of the entity definitions. A master controller entity may provide the scenario definition data. The marshalling system in the reference implementation is the scripted server; in a coalition without a server, a simulation could be the marshalling system.

Share Scenario
This message is sent by a controller entity. When a C2SIM system with a populated list of definitions (i.e., it is the marshalling system identified in the SubmitInitialization message) receives this message, it will send a C2SIMInitializationBody message to all C2SIM systems.

C2SIMInitializationBody
This message contains the scenario data, the combined list of definitions of all objects in the scenario, and a map of object IDs to system names. When simulations receive this message, they should start a new, empty scenario with terrain or other data defined by the scenario data and create objects that are mapped to this simulation in the object-system map. Simulations and other instantiated objects such as robotic systems will also retain IDs and information about units superior and subordinate to their object instances, and about graphics and other non-actor objects.

Initialization Complete
This message is sent by each C2SIM system after receipt of a C2SIMInitializationBody message to indicate that the system has completed its required scenario initialization and is ready to start the scenario. It is broadcast to the coalition.

Start Scenario
This command indicates that the scenario has started. If no other mechanism is in place to start a scenario running on a system, then this message can cause a synchronized start. However, other simulation or scenario management mechanisms (such as DIS Siman PDUs) may be in use to start the scenario run. This message in any case indicates to all C2SIM systems that no further initialization messages will be sent, and any received later can be ignored.

7.2 Other Reference Server Functionality

Previous battle management protocol projects have implemented one or more central servers to perform the following functions:

- Distribution of messages to exercise participants, based on characteristics defined in a subscription process.
- Validation of messages.
- Logging of messages enabling after-exercise replay.
- Simulation state control.
- Marshalling and distribution of simulation initialization information.
- Multiple server support.
- Translation between different battle management protocols.
- Tracking of unit position enabling late joiners to obtain current status.

The George Mason University C4I and Cyber Center (http://c4i.gmu.edu) has built a number of reference implementations of battle management languages over the years. As the C2SIM standard develops, developers continue to update the reference implementation server to implement fully the evolving standard. A description of the current implementation follows [18].

GMU Server Configuration: The GMU C2SIM server(s) usually runs as a virtual machine (VM) using the following components:

- Linux Centos 7
- Java Version 8
- Java Document Object Model (JDOM) 2.0.6
- Apache Tomcat 8.0.30 Web Services (Representational State Transfer (REST)-ful WS)
- Apache Apollo 1.7.1 Messaging (Streaming Text-Oriented Messaging Protocol (STOMP))

Figure 7.1 illustrates the typical configuration.

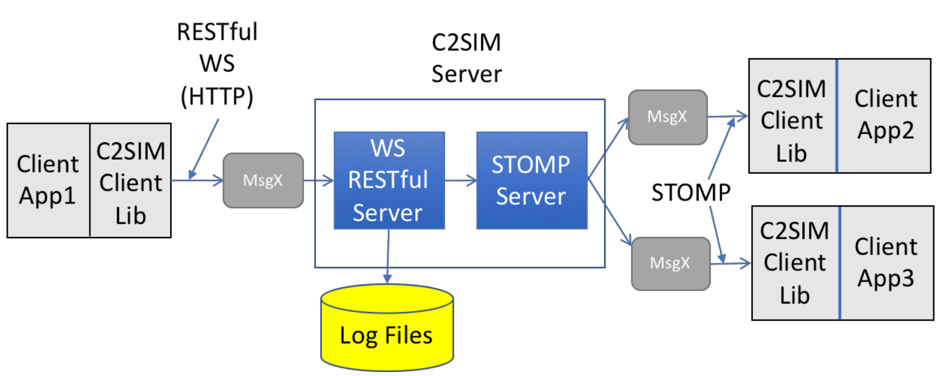

Figure 7.1: C2SIM Reference Implementation.

The current server runs under Linux; however, since all components are Java-based, the server could easily be hosted on Windows or MacOSX. The Linux host currently is implemented under VMWare Fusion, which allows it to run in a Linux virtual machine on other platforms. Clients may be C2 systems, simulators, or command line interfaces on other hosts.

Message Distribution: XML messages are prepared and passed to a C2SIM client library which forwards them using RESTful Web Services [19] over a transmission control protocol (TCP) socket to the C2SIM server. The Web Services application is written in Java 8 and implemented on Tomcat. Various functions are performed on the input message which is then forwarded to a STOMP server for distribution [20]. Clients receive messages via persistent STOMP connections. Filtering of messages can be performed through the use of subscription parameters when the client STOMP connection is first made.

Message Validation: Messages are not subjected to full XML validation, which would limit throughput, but are checked for "well formedness" and for required elements as they are processed.

Message Logging: All messages are logged exactly as entered along with the arrival time and an identification of the submitter. These logs can be resubmitted using a ReplayClient enabling a full replay of message traffic from the exercise. This can be done from the beginning of the execution or from a specified time and can be replayed at original clock speed or at full speed ignoring the time between messages.

Simulation State Control: The C2SIM server coordinates the operation of the simulation exercise by establishing and enforcing a set of states. The states are: Unitialized; Initializing; Initialized; Running; and Paused. Commands issued to the server to change states are: SubmitInitialization; InitializationComplete; ShareScenario; ReadyToRun; and StartScenario. Figure 7.2 shows a state diagram for the server.

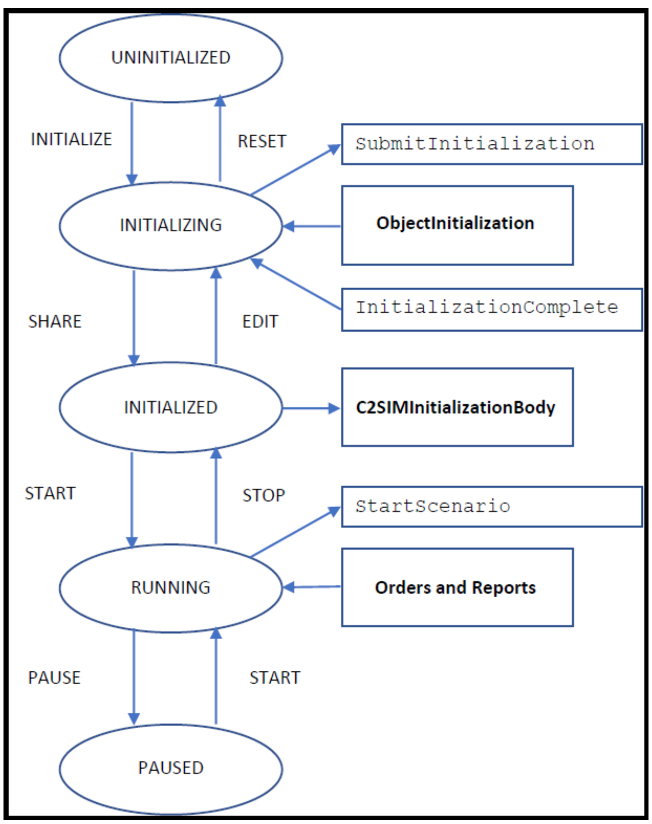

Figure 7.2: Server State Diagram showing Commands Causing State Transitions.

Marshalling and Distribution of Simulation Initialization Information: Initialization information, particularly information about units, may be submitted by several participants during the Initializing phase. This information is marshalled into a single set of data and distributed in a consolidated message when the SHARE command is received prior to the start of the scenario execution.

Multiple Server Support: It is possible to have more than one server in an exercise, each one supporting a set of clients. A single "back to back" client can be connected to two servers and will forward information between them. Each server "marks" the document indicating that it handled that particular document. [21]

Translation: Over the years, a number of different implementations of languages for scenario initialization and operation have been developed, including:

- Joint Battle Management Language (JBML)
- Integrated Battle Management Language (IBML)
- Coalition Battle Management Language (C-BML)
- Military Scenario Definition Language (MSDL)
- Command and Control Systems - Simulation Systems Interoperation (the emerging C2SIM standard)

The C2SIM server has the ability to perform translation across these protocols to support those participants still using older standards, thus providing backwards compatibility to earlier approaches. [22]

Tracking Unit Positions: As position reports are received from simulators, the server keeps track of the latest position of each simulated unit. Query capabilities are provided to enable "late joiners" to obtain current unit position rather than using the initial position from initialization.

Scenario Initialization: The approach described below assumes that there is a master controller entity that can inject these messages into the Coalition network. While this approach is part of the current reference implementation, it is subject to change as the C2SIM standard evolves toward eventual balloting.

8. Community Evaluations

The North Atlantic Treaty Organization (NATO) Modeling and Simulation Group (MSG) has chartered Technical Activity (TA) programs to advance C2SIM capabilities. MSG-048 and its follow-on MSG-085 successfully demonstrated the value of C2SIM and its technical feasibility. Building technical interoperability standards is a complex and time-consuming process. The C2SIM standardization efforts have been developed with SISO oversight, which relies on voluntary support from across government, industry, and academia to carry out its work. Under the NATO Science and Technology Organization (STO) umbrella, several efforts were formed in parallel and often in concert with SISO to assist in the validation and development of proposed C2SIM interoperability standards. Both MSG-048 and MSG-085 have identified the need for greater clarity and maturity in C2SIM operational requirements [23].

MSG-048 conducted a TA from 2006 to 2009 that involved an assessment of the concept of C-BML [24]. MSG-048 performed preliminary analyses and a series of experiments that supported the development of an initial set of requirements and recommendations for subsequent BML standardization efforts. The result confirmed the workability, usefulness, and applicability of using a standardized, digitized form for the exchange of military orders and reports among C2 and simulation systems. This established the potential to increase the efficiency and effectiveness of coalition forces during training exercises, planning activities, and coalition operations.

The follow-on activity to MSG-048 was MSG-085. This group had a mission to assess the operational relevance of C-BML while contributing to C2SIM standardization and assisting in increasing the Technical Readiness Level (TRL) of C-BML technology to a level consistent with operational employment by stakeholders. This effort also included the use of MSDL for scenario initialization [23].

The follow-on activity to MSG-085 is MSG-145, which seeks to solidify the operational foundation of C2SIM beyond its beginnings as a technical study. The capabilities demonstrated have been matured to address extended use cases demonstrating how individual nations utilize their national C2 and simulation systems within a coalition military force. Specific objectives of MSG-145 include:

- Exploit C2SIM through an operational, conceptual, and executable scenario development process.
- Develop extensions to the unified C2SIM core data model.
- Inform the standards development process.
- Educate the community of practice on C2SIM technology.
- Make recommendations for "covering" the C2SIM standard with a NATO Standardization Agreement (STANAG). [25]

MSG-145 begins with the MSG-085 provided C2SIM system architecture. Participating nations have developed wide-ranging, standalone use cases that will leverage C2SIM in the existing comprehensive military environment. Each standalone use case has a scenario that supports its implementation. MSG-145 does not have the luxury of testing multiple scenarios and must develop an integrated scenario that will allow each use case to be executed in parallel with all other use cases [26]. MSG-145 has planned a phased validation of C2SIM in 2019, with the following major objectives/milestones:

- In May 2019, the national teams participating in MSG-145 will conduct individual and pairwise experiments to validate that C2SIM is able to meet their projected needs for C2-simulation interoperability.
- In June 2019, MSG-145 will leverage the NATO Coalition Warrior Interoperability eXploration, eXperimentation, eXamination eXercise (CWIX; [27]) events from 2017 to 2019 to use C2SIM in a representative coalition operational environment, within the overall context of Modeling and Simulation as a Service (MSaaS).
- In July 2019, the national teams will participate in a distributed mission planning exercise, modelled after the successful (but minimally distributed) final demonstration of MSG-085 [28] which will stress the ability of multiple C2 systems and simulations to interoperate in an operational context.
- The results of the validation process will inform MSG-145's final report and are expected to validate the SISO C2SIM PDG's work.

9. Conclusion and Next Steps

The long-held goal of standardized coalition C2-Simulation interoperation, complete with message structures, system-wide initialization, and extensibility, has nearly been achieved. This paper has described how C2SIM builds on the roots of first-generation MSDL and C-BML to achieve these goals. With the support of NATO MSG-145, the C2SIM PDG has assembled a practical, unified standard to replace (and be backward-compatible with) MSDL and C-BML. The standard draft and supporting ontologies are nearing maturity and a validation process is underway. The authors have high hopes that the resulting C2SIM standard will enable far greater standardized interoperability of C2 and simulation across coalitions. The eagerly anticipated result will be that coalition C2 can employ the power of integrated simulation where each national force uses its own C2 system and is represented by a simulation tailored to its personnel, equipment, and doctrine.

10. References

[1] Simulation Interoperability Standards Organization (SISO): Standard for Military Scenario Definition Language (MSDL), SISO-STD-007-2008, May 11, 2015.

[2] Simulation Interoperability Standards Organization (SISO): Standard for Coalition Battle Management Language (C-BML) Phase 1, SISO-STD-011-2014, April 14, 2014.

[3] Blais, C., Schade, U., Sikorski, L., Wolski, M., Singapogu, S., and Gautreau, B.: "A Transformation Process for Generating an Extensible Markup Language (XML) Schema from a Formal Ontology for Practical Application in C2SIM Implementations," Paper 2019-SIW-018, Simulation Innovation Workshop, Orlando, FL, February 2019.

[4] Simulation Interoperability Standards Organization (SISO): Guide for Coalition Battle Management Language (C-BML)

Phase 1, SISO-STD-011-2014, April 14, 2014.

[5] Gerz, M., and Bau, N.: "A Platform-Independent Reference Data Model for a Future Interoperability Solution," Proceedings of the 17th International Command and Control Research and Technology Symposium, Fairfax, Virginia, 2012.

[6] Simulation Interoperability Standards Organization (SISO): Product Nomination for Command and Control Systems - Simulation Systems Interoperation, SISO-PN-010-2014 revision 1.1, August 2017.

[7] Blais, C., Turnitsa, C., and Gustavsson, P.: "A Strategy for Ontology Research for the Coalition Battle Management Language (C-BML) Product Development Group," Paper 06F-SIW-003, Fall Simulation Interoperability Workshop, Orlando, FL, 2006.

[8] Singapogu, S. "Opportunities for Next Generation BML: Semantic C-BML," Proceedings of the 19th International Command and Control Research and Technology Symposium, Alexandria, VA, June 2014.

[9] Singapogu, S., Gupton, K., and Schade, U.: "The Role of Ontology in C2SIM," Proceedings of the 21st International Command and Control Research and Technology Symposium, London, UK, September, 2016.

[10] Blais, C., Reece, D., and Singapogu, S.: "From Information Description to Information Understanding: The Role of Ontology in Emerging SISO Standards," Paper 2019-SIW-013, Simulation Innovation Workshop, Orlando, FL, February, 2019.

[11] World Wide Web Consortium (W3C): OWL 2 Web Ontology Language Document Overview (Second Edition), W3C Recommendation 11 December 2012, https://www.w3.org/TR/owl2-overview/.

[12] Musen, M.A.: "The Protégé Project: A Look Back and a Look Forward, AI Matters. Association of Computing Machinery Specific Interest Group in Artificial Intelligence, 1(4), June 2015.

[13] World Wide Web Consortium (W3C): RDF 1.1 XML Syntax, W3C Recommendation 25 February 2014, http://www.w3.org/TR/rdf-syntax-grammar/.

[14] Foundation for Intelligent Physical Agents (FIPA): FIPA ACL Message Structure Specification, December, 2002, http://www.fipa.org/specs/fipa00061/SC00061G.pdf.

[15] Posad, S.: "Specifying Protocols for Multi-Agent Systems Interaction," ACM Transactions on Autonomous and Adaptive Systems, 2(4), November, 2007.

[16] Institute of Electrical and Electronics Engineers (IEEE): IEEE Standard for Distributed Interactive Simulation (DIS) – Application Protocols, IEEE 1278.1-2012, December, 2012.

[17] North Atlantic Treaty Organization (NATO): NATO Joint Military Symbology, NATO-APP-6, October, 2017.

[18] George Mason University (GMU): C2SIM Server Implementation 8.1: User Instructions, GMU C4I-Cyber Center, https://netlab.gmu.edu/trac/OpenBML/raw-attachment/wiki/C2SIMClientandServer/C2SIM%20Server%20Reference%20Implementation%209.0.pdf.

[19]World Wide Web Consortium (W3C): Web Services Architecture, W3C Working Group Note, 11 February 2004, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211.

[20] STOMP Protocol Specification, Version 1.2, https://stomp.github.io/stomp-specification-1.2.html.

[21] Pullen, J. M., Corner, D., Remmersmann, T., and Trautwein, I.: "Linked Heterogeneous BML Servers in NATO MSG-

085," Paper 13F-SIW-024, Fall Simulation Interoperability Workshop, Orlando, FL, September, 2013.

[22] Pullen, J. M., Corner, D., Gustavsson, P., and Wittman, R.: "Order and Report Schema Translation in WISE-SBML Server," Paper 13F-SIW-023, Fall Simulation Interoperability Workshop, Orlando, FL, September, 2013.

[23] North Atlantic Treaty Organization (NATO): NATO Modeling and Simulation Group 85 (NMSG-085) Final Report: Standardisation for C2-Simulation Interoperation, AC/323(MSG-085)TP/640, November, 2015.

[24] North Atlantic Treaty Organization (NATO): NATO Modeling and Simulation Group 48 (NMSG-048) Final Report: Coalition Battle Management Language (C-BML), AC/323(MSG-048)TP/415, February, 2012.

[25] North Atlantic Treaty Organization (NATO): MSG-145 Operationalization of Standardized C2-Sim Interoperability - 1st Meeting Minutes, NATO Collaboration Support Office, March, 2015.

[26] Galvin, K., and Ruth, J.: "Architecture Driven Scenarios in Support of C2-Simulation (C2SIM)," Proceedings of the 22nd International Command and Control Research and Technology Symposium, Playa Vista, CA, November, 2017.

[27] Pullen, J. M., Khimeche, L., and Galvin, K.: C2SIM in CWIX: Distributed Development and Testing for Multinational Interoperability, MP-MSG-159-13P, NATO MSG Symposium, Ottawa, Canada, October, 2018.

[28] Pullen, J. M., and Khimeche, L.: "Advances in Systems and Technologies toward Interoperating Operational Military C2 and Simulation Systems," Proceedings of the 19th International Command and Control Research and Technology Symposium 2014, Alexandria, VA, June, 2014.

Author Biographies

DR. J. MARK PULLEN is Professor of Computer Science and Director of the Center of Excellence in Command, Control, Communications, Computing, Intelligence (C4I) and Cyber. His doctorate in Computer Science is from George Washington University; he also holds degrees in Electrical Engineering from West Virginia University. Previously he was a Program Manager at DARPA and an Associate Professor of Electrical Engineering at the US Military Academy West Point, NY. Dr. Pullen's research interests include networked multimedia applications, emphasizing command and control, networked education, and training, distributed virtual simulation, and interoperation of command and control with simulations. He serves as Co-Chair of the SISO Product Development group for C2SIM. Dr. Pullen is a Fellow of the IEEE, Fellow of the ACM, and licensed Professional Engineer. He received the IEEE Harry Diamond Memorial Award and the Defense Superior Service Award, and was a leader in the team that received the NATO Scientific Achievement Award for 2013.

DOUGLAS CORNER is a member of the staff of the George Mason University C4I Center. He holds a MS in Physical Chemistry from the Ohio State University and an MS in Computer Science from George Mason University He is the lead software developer C2SIM reference implementation server.

DR. SAMUEL SINGAPOGU is a post-doctoral associate at the GMU C4I and Cyber Center having completed his dissertation in a novel method for knowledge discovery of XML data applied to C2 systems. He has been involved in the drafting of C-BML, applications involving BML and IBML, and currently serves as the lead editor for development of the C2SIM standard.

DR. CURTIS BLAIS is a member of the research faculty in the Naval Postgraduate School Modeling, Virtual Environments, and Simulation (MOVES) Institute. He has 45 years of experience in modeling and simulation, including specification, design, development, and application of simulations for training and analysis, software engineering project management, and modeling and simulation education. He chairs the C2SIM Product Support Group in SISO. Dr. Blais earned a BS and MS in Mathematics at the University of Notre Dame and a Ph.D. in MOVES at the Naval Postgraduate School.

DR. DOUGLAS REECE is a Principal Engineer at VT MAK. He has been developing physical and behavioral models for simulations for over 20 years, most recently as part of the VR-Forces team at VT MÄK. His simulation experience also includes OneSAF, JSAF, DISAF, ModSAF, CCH, and the IST SAF. This work has covered many areas from ship and aircraft movement control to individual combatant behavior to traffic behavior to aggregate-level warfare. He has contributed to numerous research programs aimed at developing intelligent computer controlled entities.and to simulation interoperability standards for representing humans and C2 messages. He received his Ph.D. in Computer Science from Carnegie Mellon University in 1992.

JIM RUTH is a Senior Military Analyst at Trideum Corporation and serves as the Lead Simulation to Mission Command Interoperability (SIMCI) Architect. Mr. Ruth has been a contributor to the C2SIM standard development for the past five years. He is a member of the US delegation to NATO MSG-145 and supported its precursor, MSG-085.