C2SIM SANDBOX USER GUIDE

1. Distributed Integration and Testing in NATO MSG-145

Integration and testing has been a significant challenge in developing C2SIM by multiple NATO national teams since its beginning as Battle Management Language (BML). In MSG-048, an XML schema developed by one team was exchanged among the participating national teams. This formed the basis for quarterly integration and testing sessions where all teams assembled at a common location. Given the style differences among national teams involved, the integration process was both technical and social in nature and clearly was necessary in order to arrive at an integrated whole. However, it proved very expensive in terms of developer travel and the cost seemed likely to increase as the schemata evolved to become more complex during the standardization process. MSG-085 arrived at a style of Internet-based development and testing, consistent with the fact that the intended product was designed for distributed operation in a networked environment. The new style started as a VPN enclave, where any of the national teams could work with the same server to test and/or demonstrate C2SIM functionality.

2. C2SIM Sandbox Concept

In MSG-145, the integration and testing environment has expanded to become a full C2SIM capability, available over the VPN by remote desktop technology, allowing national teams to test and demonstrate any combination of C2 systems, simulation systems, and servers in the "C2SIM Sandbox." The capability includes the open source Internet-based audio/video/whiteboard/chat conferencing system Jitsi [16], which greatly facilitates group communication.

The C2SIM Sandbox provides a continually available environment, available by VPN to national teams to test and demonstrate C2SIM. It supports the initial trial SISO C2SIM core developed for CWIX, including capabilities for orders, reports, and initializations. In addition to these functions, the C2SIM Sandbox is building experience toward a future "C2SIM as a Service" capability. This has begun by using the C2SIM Sandbox as the nucleus of a distributed testbed operated for MSG-145. Development and testing for CWIX 2018 deployment took good advantage of these capabilities. GMU C4I & Cyber Center plans to implement the C2SIM schema derived from the C2SIM Core and Land Operations Extension ontologies in the Sandbox, as soon as those are stable.

3. Current Component Systems

Figure 1 shows the architecture of the C2SIM Sandbox, including the C2, simulation, server, scheduling, and collaboration/conferencing components. All of these are intended to be accessed remotely, via a Web browser. In addition to interacting with the sandbox components (individually or at multiple remote VPN sites), the systems in the Sandbox are able to interact over the VPN via C2SIM standards with other C2 systems, simulation systems, or server systems.

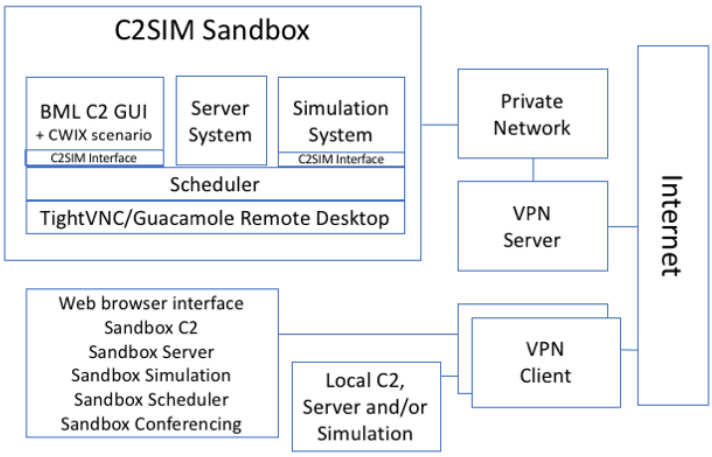

Figure 1. C2SIM Sandbox Architecture

The C2SIM Sandbox currently employs the BMLC2GUI system as a surrogate for C2, the VT-MÄK commercial combat simulation VR-Forces, and an open source server, operating in a virtual computing environment. The server is the C2SIM Reference Implementation server developed for the SISO C2SIM standards effort; it features interoperation with MSDL, C-BML and IBML09 through schema translation. The Sandbox has been tested with external C2IS NORCCIS/SWAP and simulations JSAF and KORA in CWIX2018. Documentation for all of these systems is either posted on or linked on the webpage http://c4i.gmu.edu/OpenBML.

An important capability employed in the Sandbox is virtual/remote desktop via Web browser, which recently has become available commercially in Apache Guacamole. This enables remote participants to work with any application incorporated in the Sandbox, using only the readily available open source OpenVPN client and any HTML5 compliant web browser. The Sandbox environment uses open source remote desktop gateway software and virtual network computing (VNC) to enable remote interaction with virtual machines hosted within a VMWare vSphere hypervisor as well as actual physical machines. The virtualized environment allows for flexibility in the applications and services made available for testing.

C2SIM Sandbox remote application interfaces are intended to be limited to the user GUI capability of each software system. In this mode the Sandbox is able to block inspection of the C2, simulation or server code so that privacy can be provided for software providers. For Windows systems, this is accomplished using a combination of group policy controls to operate the system in a kiosk mode. Instead of physical interaction, users work remotely through the remote desktop gateway accessed within the Sandbox environment's virtual private network. A similar capability is achieved on Linux systems using a window manager optimized for restricting application access. However, at present there are no software systems in the Sandbox requiring this privacy, so a Windows 10 desktop is displayed, limited to files made accessible to user “c2sim”.

A pre-packaged scenario from CWIX 2018 with instructions for operation is included with the Sandbox, to enable the C2SIM configuration to be exercised with only a minimal understanding of the software. The scenario is user-modifiable within a limited scope via the C2 system GUI, to allow users to run alternatives and observe results.

4. Scheduling C2SIM Sandbox Use

The Sandbox has a scheduling calendar. In keeping with the collaborative nature of the Sandbox, the scheduling mechanism does not have an enforcement mechanism; it is simply way for users to show when they want to reserve its use. Users are expected to cooperate by working in the Sandbox only at times they have reserved.

To reserve you will need to be connected to the Sandbox VPN (experiment-access) and then direct your Web browser to: http://10.2.10.30/reserve

You can reserve one-hour slots by entering first/last name, email, phone number, and purpose/note.

You can make a reservation only from available timeslots. You can also view (read only) the calendar of reservations. The link is on the bottom of the reservation page.

When you make a reservation, you should receive email with a link with the info and a link to cancel your reservation or to delete all of your personal information from the system. Remember, the scheduling links only work when connected to the experiment VPN.

5. C2SIM Reference Implementation Server

The C2SIM Sandbox server is a new open-source Java-based server, developed by the GMU C4I and Cyber Center as a reference implementation for the C2SIM Logical Data Model Core and its Land Operations extension. This is a further development based the WISE-SBML translating server. It is capable of translating among XML documents based on different schemata if they are semantically-equivalent and designed to be easy to reconfigure for new schemata. This capability provides a means of backward compatibility from C2SIM to first-generation BML standards MSDL and C-BML. Use of Java is intended to make the server more portable and understandable, at the cost of lower performance than WISE-SBML; it nevertheless has shown transaction rates in excess of 200 reports per second. Like its predecessors, the server supports logging and replay; it also includes support for late joiners and checkpoint/restart.

6. Modes of Use

The C2SIM Sandbox is able to facilitate testing and demonstration in a variety of modes:

- C2SIM demonstrations
  - Schemata: IBML09, CBML Light, C2SIM Core
  - C2SIM Maneuver Warfare when available

    - CWIX 2018 scenario provided (others will be added if contributed)
- C2SIM testing
    - Test C2 with Sandbox Server and Simulation
    - Test Server with Sandbox C2 and Simulation
    - Test Simulation with Sandbox C2 and Server
    - Test C2-Simulation Coalitions with the Server
    - Distributed configurations of all sorts
- C2SIM validation with SISO
- C2SIM-based exercises (scope limited by server performance)
- In the future: C2SIM as a Service, based on the NATO C3 Taxonomy and Modeling and Simulation as a Service (MSaaS) Reference Architecture. The NATO MSCOE intends to make the open source C2SIM Reference Implementation Server available in an MSaaS container for CWIX2019.

GETTING STARTED WITH THE C2SIM SANDBOX

1. You will need credentials for the Experiment-Access VPN where the Sandbox resides. Contact mpullen@c4i.gmu.edu for this. If you need an assigned IP address in the VPN address space (10.2.10/24), for example for a C2 or simulation system, be sure to state that in your request.

2. Much of the software and documentation for the Sandbox will be found on http://c4i.gmu.edu/OpenBML.

3. Minimum software to access the Sandbox is the OpenVPN client and Google Chrome or other HTML5 compliant browser. These are available at no cost on the Internet. Install both and use the information that comes with your VPN credentials to configure OpenVPN.

4. Open a VPN connection to Experiment-Access and use Google Chrome to access URL http://10.2.10.32:8080/guacamole/#/ and login to Guacamole with account:c2sim and password:sandbox.

5. You can communicate with other teams working in the Sandbox, via Jitsi. Use Google Chrome to access https://10.2.10.28/sandbox with an HTML5 compliant browser on your OpenVPN-connected system. Or you can use any other mutually agreeable Internet conferencing system, such as Zoom or WebEx.

6. Start VR-Forces on the Sandbox using the desktop link (a red ball). Chose “launch” on the first GUI and on the second “load recent scenario” or “load scenario from disk,” choosing CWIX. When the main VR-Forces GUI comes up, use the File Menu to load terrain “MAK Earth base (online)” and navigate to Bogaland (AKA west-central Sweden). If you are using a different scenario, delete all units and tactical graphics objects in the Objects List panel at left of the VR-Forces GUI.

   NOTE: If VR-Forces is already running, you can usually avoid all of the above by simply using the File dropdown to “close scenario” and “load recent scenario”.

7. Start the c2simVRF interface using the desktop link. Information and open source code for this software is available at OpenBML/C2SIM-VRForces-v2.2.pdf. It will connect to VR-Forces, subscribe to the C2SIM Server in the Sandbox, and listen for C2SIM Orders.

8. Start the BMLC2GUI using the desktop link. This will let you push C2SIM Order to the Server and observe reports being made by VR-Forces. See OpenBML/BMLC2GUI_User_Guide_v2.7.1.pdf.

9. You will need to follow this procedure on the BMLC2GUI to start system operation:

a. Use the File dropdown on BMLC2GUI to send the server STOP, RESET and INITIALIZE.

b. Use the File dropdown to Open and Push Initialization (ObjectInitialization message) into the server. Red and Blue test initializations are available to use.

c. Use the File dropdown to send the server SHARE and START. The objects you initialize will appear on VR-Forces main GUI; if you don't see them try zooming out. They also appear on the BMLC2GUI.

d. Now you are ready to Open and Push C2SIM orders. Red and Blue test orders are available, or you can create your own. The BMLC2GUI will let you edit an existing order or create a new one, under the File menu.

e. You can also enter orders and receive reports on your own C2IS system, if it has a C2SIM Interface. The server is at 10.2.10.30 in the VPN. See the open source for VR-Forces interface on OpenBML for examples of C++ connection, and the open source for BMLC2GUI on OpenBML for examples for Java connection, using the ClientLib software from OpenBML in both cases.

f. You can also receive orders and generate reports on your own simulation, if it has a C2SIM Interface. The same ClientLib applies there.

g. The BMLC2GUI will record C2SIM XML messages from the server's STOMP output and replay those messages as input to the server so as to repeat a period of Sandbox operation. Use top-row buttons "Record STOMP" and "Play Recording".

h. You can visually monitor the STOMP traffic sent by the server by clicking on the runSTOMP link on the Sandbox desktop. This shows all order and reports distributed by the server. It does not monitor REST inputs; however successful XML inputs generally are reflected in the STOMP output.

i. You can check the operational status of the server by accessing http://10.2.10.30:8080/BMLServer/status. (The current message provides somewhat limited status; we welcome suggestions for improving it.)

j. Please send suggestions for improvement and trouble reports to mpullen@c4i.gmu.edu.

10. You can save files on the Sandbox Windows system; however, they will be deleted nightly.

11. When finished with your Sandbox session, simply shutdown the browser that was connected to http://10.2.10.32:8080/guacamole/#/.

NONE OF THE ABOVE REQUIRES SANDBOX LOGIN/LOGOUT OR SHUTDOWN. DO NOT USE THE WINDOWS SHUTDOWN OR LOGOUT CONTROLS. THESE WILL DISABLE THE SANDBOX UNTIL AN OPERATOR CAN RESET IT!! JUST CLOSE THE BROWSER THAT IS CONNECTED TO GUACOMOLE.