Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. D. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

WEBLVC - AN EMERGING STANDARD AND NEW TECHNOLOGY FOR LIVE, VIRTUAL AND CONSTRUCTIVE SIMULATION ON THE WEB

Peter Swan

VT MAK
150 Cambridge Park Drive, 3rd Floor
Cambridge, MA 02140, USA

ABSTRACT

With the power and capability of new web technologies such as HTML5, WebGL and WebSockets, customers are starting to realize the benefits of migrating simulation and training systems to thin client web-based environments. However, what is missing is a standard interoperability protocol for linking these new web-applications with each other, and with traditional M&S federations.

A protocol called WebLVC has been proposed to fill that gap, and is the basis for a standards product development activity within the Simulation Interoperability Standards Organization (SISO). WebLVC is an interoperability protocol that enables web-based applications (typically JavaScript applications running in a web browser) to interoperate in Modeling and Simulation (M&S) federations. WebLVC client applications communicate with the rest of the federation through a WebLVC server, which participates in the federation on behalf of one or more clients. The WebLVC protocol defines a standard way of passing simulation data between a web-based client application and a WebLVC server - independent of the protocol used in the federation. Thus, a WebLVC client can participate in a DIS exercise, an HLA federation, a TENA execution, or other distributed simulation environment.

The WebLVC protocol specifies how to encode object update messages, interaction messages, and administrative messages as JSON (JavaScript Object Notation) objects, which are passed between client and server using WebSockets. WebLVC is flexible enough to support representation of arbitrary types of objects and interactions (i.e. arbitrary Object Models). However, WebLVC does include a "Standard Object Model" definition based on the semantics of the DIS protocol and HLA's RPR FOM. Users can extend the Standard Object Model by adding new types of objects, attributes, interactions, and parameters; or can choose to represent the semantics of entirely different Object Models (e.g. other HLA FOMs, ANDEM models, etc.)

Now is the right time to migrate to the Web. HTML5 provides native support for the latest multimedia such as canvas, video and audio. WebGL offers 3D graphics in a browser without plug-ins. WebSockets offers flexible, real-time, bi-directional networking. JSON (JavaScript Object Notation) is a lightweight data-interchange format that is easy for humans to read and write and is easy for machines to parse and generate. And there is an explosion of JavaScript libraries, many of them open source.

MÄK has taken the initiative to propose and promote an open standard through SISO, and to build a suite of products including a WebLVC Server and suite of WebLVC JavaScript applications. We have also integrated our scenario generation, visualization and terrain server products into the WebLVC architecture to create a complete web-based simulation environment.

- **WebLVC Server**. The WebLVC Server participates in the native federation on behalf of one or more web applications – using whatever M&S interoperability protocol is required by the federation. For example, if the native federation uses HLA, the WebLVC Server must connect to the federation through the RTI as any other federate would, and must translate between the WebLVC protocol and the federation's chosen FOM. WebLVC clients can participate in any federation whose language the gateway can speak. In fact, the WebLVC Server can simultaneously translate between multiple protocols.

- **WebLVC.js**. The WebLVC JavaScript library implements the client side of the WebLVC Protocol, and can be used directly by web-based applications to communicate with the WebLVC Server.

- **WebLVC Suite**. The WebLVC Suite comprises the Server plus a set of complete applications including:

    - 2D-3D Viewer - 2D tactical map display and 3D display, side by side.
    - Close Air Support (Uses VR-Forces) - This is an example of a web-based role player station built using WebLVC. It allows you to call for close air support using a standard 9-line interface.
    - VR-Forces Control - This web app provides simple VCR-style controls for playing, pausing, and rewinding the current VR-Forces scenario.
    - Commander Station App (Uses VR-Forces)
    - Remote Logger Control App (Uses MÄK Data Logger)
    - Remote Camera Control App (Uses VR-Vantage)
    - MÄK WebLVC Components - Building blocks for custom app development

- **VR-Forces**. VR-Forces is MÄK's complete simulation solution – a powerful and flexible Computer Generated Forces (CGF) platform to fill your synthetic environments with urban, battlefield, maritime, and airspace activity. It has an API for remote control connected to WebLVC applications for simulation management, environment control, entity creation and control, and issuance of complex parameterized tasks. Run VR-Forces sessions in the cloud and access them via WebLVC clients.

- **VR-Vantage**. VR-Vantage IG is MÄK's configurable desktop Image Generator (IG) for out-the-window (OTW) scenes, camera views, and sensor channels. Render your scene on a central server, control it from a WebLVC application, and stream video into WebLVC clients.

- **VR-TheWorld Server**. VR-TheWorld Server is a powerful open standards-based streaming terrain server that lets you stream in elevation, features, and imagery. It is delivered with a global base map, but you can also easily populate it with your own custom source data through a web-based interface. The server can be deployed on private or public networks to provide streaming terrain data to WebLVC clients and VR-Forces and VR-Vantage applications.

With MÄK's WebLVC Products:

- Developers can create compelling, customized, multi-user, web-based simulations.
- Instructors can bring their IOS with them on a tablet as they walk around the room, allowing them to change the weather, time of day, or inject faults into the scenario while standing over the trainee's shoulder.
- Presenters can use their smartphones to remotely control a Stealth or Common Operating Picture display at the front of the room – easily flipping between saved viewpoints, attaching to various entities, and even switching between visual and sensor modes with a few simple taps.
- Role players can use intuitive browser- or tablet-based apps to task and command entities that are simulated by a SAF or simulation engine running in the server room.
- Trainees can participate in exercises remotely – either on private networks or over the internet, without requiring thick clients or powerful desktop machines.
- IT staff can provide automatic and central updates of software, terrain, and other content – without having to install software and licenses on each client machine.
- Program managers can save money and reduce risk, by taking advantage of virtualization, cloud-based simulation, and the thin-client paradigm.