SISO-REF-051-2014

WebLVC Study Group

Final Report

18 September 2013

Submitted to:
SISO Standards Activities Committee

SAC Approved: 12/17/13 **EXCOM Approved: 02/10/14**

TABLE OF CONTENTS

1 INTRODUCTION / OVERVIEW

In recent years, several new technologies and standards have been developed in the broad Web community that enable highly-interactive, low-latency, real-time web-based applications that run in a browser.

Based on these technologies (WebSocket, WebGL, HTML5, etc.), it is now possible to develop web-based M&S applications like Plan View Displays, Stealth Viewers, Instructor Operator Stations, and even web-based flight simulators and first-person gaming applications. In fact, many companies and programs are doing exactly these things.

However, there is currently no standard interoperability protocol for linking these new web-applications with each other, and with traditional M&S federations in a way that is:

- High-performance enough for the needs of these applications
- Natural to use in a JavaScript environment.
- Flexible enough to support interoperability regardless of the protocol being used in the target federation (e.g. DIS, HLA 1.3, HLA 1516, HLA Evolved, TENA, etc.)

It is important that SISO support a standard mechanism to achieve interoperability in this realm, so that multiple, gratuitously non-interoperable solutions don't quickly develop.

The goals of this SG were:

- Develop a common understanding of the problem of interoperability between web applications and traditional LVC federations
- Evaluate the proposed WebLVC concept and specifics, to determine whether it looks like a reasonable starting point for a SISO standards development effort, or whether an alternative solution should be pursued
- Help refine and develop the WebLVC idea (if there is consensus that it is the right starting point), with the goal of building its maturity before kicking off a formal SISO Product Development Group (PDG).

- Achieved common understanding of the problem.
- Achieved consensus that the WebLVC Concept proposed is the right direction to pursue:
  - -- A JSON-based protocol for representing Modeling and Simulation semantics…
  - -- …that is generic enough to support arbitrary object models,
  - -- …that allows auto-generation of WebLVC messages from an HLA FOM were deemed critical,
  - -- …and that also includes a "default" object model based on DIS/RPR semantics.
- Achieved consensus that the group would like to see the WebLVC concept further developed into an eventual SISO Standard through a WebLVC PDG.

2 REFERENCES

n/a

3 DEFINITIONS

Term	Definition
WebSocket	A standard mechanism for high-performance, bi-directional communication between web client applications and web servers.
WebGL	A standard API for accessing a graphics processing unit directly from a web client application running in a browser.
HTML5	The latest version of the HTML standard, which is used to define the content of web pages.

Term	Definition
JavaScript	The standard language supported by all major browser developers, for developing and executing client-side web applications.
JSON	JavaScript Object Notation - a standard text-based format for representing structured data in a way that is both human-readable and machine-readable (similar to XML). Because JSON syntax is the same syntax used to represent objects natively in JavaScript, JSON is the most natural way to commuicate structured data to and from web applications.
Web Applications	Client-side JavaScript applications that run natively in a web browser without requiring browser plug-ins.
WebLVC	A specification for representing Modeling and Simulation semantics in JSON messages, which may be exchanged between web applications and web servers via WebSockets.

4 ACRONYMS AND ABBREVIATIONS

Acronym	Definition
JSON	JavaScript Object Notation
HLA	High-level Architecture
DIS	Distributed Interactive Simulation
TENA	Test and Training Enabling Architecture
DDS	Data Distribution Service
LVC	Live, Virtual, and Constructive
RPR FOM	Real-time Platform Reference Federation Object Model

5 SG REPORT

5.1 SG OFFICERS

Officer Role	Name	Organization	Country
Chair	Len Granowetter	VT MAK	USA
Vice Chair	Don MacGregor	Naval Postgraduate School	USA
Secretary	Brad Dillman	Real-time Eng. & Simulation	Canada
SAC TAD	Kevin Gupton	ARL University of Texas	USA

5.2 SG MEMBER LIST

Name	Organization	Country
Aaron Dubois	VT MAK	USA
Shagoto Nandi	VT MAK	USA
Steve Sheasby	Raytheon	USA
Dan Gregory	Thales	UK
Jean-Louis Igarza	Antycip	France
David Drake	JHU APL	USA
Stephan Sandberg	Pitch	Sweden
Frederik Antelius	Pitch	Sweden
Patrice LeLeydour	Thales	France
Felix Rodriguez	VT MAK	USA
Colin Timmons	DND / CAF SECO	Canada
Jared Quintana	Lockheed Martin / DMOC	USA
Jose Ruiz	DGA (French MoD)	France
Yannick Guillemer	French MoD	France
Bastien Olivo	French Army	France
David Graham	CAE	Canada

Name	Organization	Country
Laurent Prignac	MBDA	France
Kurt Lessman	Trideum	USA
Andy Bowers	MITRE	USA
Curt Schroeder	Antycip Simulation	UK

5.3 TASK DESCRIPTIONS FROM THE TOR

1. Develop a common understanding of the problem of interoperability between web applications and traditional LVC federations

2. Evaluate the proposed WebLVC concept and specifics, to determine whether it looks like a reasonable starting point for a SISO standards development effort, or whether an alternative solution should be pursued

3. Help refine and develop the WebLVC idea (if there is consensus that it is the right starting point), with the goal of building its maturity

4. Establish recommendations for next steps for the topic within SISO, e.g., a recommendation to the SAC for whether to establish a WebLVC PDG.

5.4 PRODUCT DESCRIPTIONS FROM THE TOR

1. The main product of this SG will be a final report to the SAC, describing findings and recommendations for next steps. In particular, the report should contain a recommendation on whether the SAC should establish a WebLVC PDG. If the SG does recommend establishing a PDG, a TOR shall define the specific scope for the PDG. A PN shall define the product(s) the PDG will produce and will clarify requirements and use cases that the product(s) shall try to satisfy. The SISO Products produced shall be in accordance with all relevant SISO-ADM products (e.g., SISO-ADM-002, SISO-ADM-003, and SISO-ADM-005).

2. As a result of its task to refine and mature the WebLVC concept, the SG output also includes an initial draft of a partial WebLVC specification, to be used as a starting point by a WebLVC PDG.

5.5 SIGNIFICANT RESULTS AND/OR ACHIEVEMENTS

- The group achieved a common understanding of the problem: How do we link web-based applications with each other, and with traditional M&S federations in a way that is:
– High-performance enough for the needs of M&S applications
– Natural to use in a browser-based (JavaScript) environment
– Flexible enough to support interoperability regardless of protocol used in the target federation

- The group achieved quick consensus that the WebLVC Concept proposed is the right direction to pursue. In fact several members of the group had independently come to the same top-level conclusion: That a JSON-based protocol that encodes the semantics of modeling and simulation data would enable easy integration of web-based applications with traditional M&S federations.

- There was quick consensus that the WebLVC protocol needed to be flexible enough to support arbitrary object models. However, there was much discussion on two key questions:
- Should there be a default object model based on the semantics of DIS/RPR FOM?
- Is there a need to allow manual generation of object-model-specific WebLVC messages, or should the WebLVC Standard consist solely of a set of rules for automatically generating WebLVC messages from an HLA FOM?

- There was also significant discussion about whether the WebLVC specification should dictate that the messages be sent over WebSockets, or leave the choice of transport mechanism to the user.

See below for the results of these discussions.

In the end, there was a strong consensus to further develop the WebLVC concept and protocol, with the goal of developing it into a SISO Standard. The group's consensus recommendation was to form a WebLVC PDG for this purpose immediately.

5.5.1 DISCUSSION FOR SIGNIFICANT RESULT AREA 1

Should there be a default object model based on the semantics of DIS/RPR FOM?

There was quick consensus that the WebLVC protocol must be flexible enough to support arbitrary object models. But there was much discussion about whether a default object model, based on the semantics of DIS/RPR FOM should be provided. In the end, consensus was that a default object model is a good idea - to promote a-priori interoperability among web applications developed by different people on different projects. The semantics, data representations, and conventions of DIS and RPR FOM have been developed by the SISO community over two decades, and there is a clear, present, and common need to represent precisely those semantics in the WebLVC environment. It would be counter--productive to interoperability if each project made conflicting decision about how to represent the same semantics in JSON/WebLVC messages.

We also agreed that the WebLVC default object model does not need to include a message for every DIS PDU or RPR FOM object or interaction class. A version 1.0 of a WebLVC Standard should include specifiations only for those messages that the PDG feels are likely to be required by WebLVC users. If no one is interested in a particular message type, leave it out. It can always be added to a future version later, or implemented as an extension by users.

5.5.2 DISCUSSION FOR SIGNIFICANT RESULT AREA 2

- Is there a need to allow manual generation of object-model-specific WebLVC messages, or should the WebLVC Standard consist solely of a set of rules for automatically generating WebLVC messages from an HLA FOM?

There was consensus that the ability to automatically generate WebLVC messages from arbitrary HLA FOMs is a critical design requirement for a WebLVC Standard. The most important benefit of this isn't just to save work in writing WebLVC messages. It's to allow implementation of truly object-model-independent WebLVC Servers (that can translate HLA data to WebLVC and vice versa without writing FOM-specific code). There was agreement we need to specify a standard set of mappings from HLA representation to JSON representation for each data type, to allow auto-generation of WebLVC messages in a standardized way.

There was a suggestion that once we have dictated a set of standard mapping for generating WebLVC messages from an HLA FOM, it would be no longer necessary to explicitly provide a default object model in order to achieve a-priori interoperability among users who required the semantics of DIS/RPR FOM. Different users who applied the standard mapping rules to the RPR FOM would achieve a common result.

However, in the end, there was consensus around the idea of explicitly providing message specifications for the default DIS/RPR-based obejct model, and using a manual process for designing at least the most common messages types. Although auto-generation is important, WebLVC messages that blindly mirror FOM representations would be harder to use than WebLVC messages that are heavily inspired by RPR, but tailored to the expectations/needs of web applications.

It was also pointed out that allowing manually-generated messages (in addition to auto-generated) is necessary to support cases where WebLVC Messages are not being generated from an HLA FOM. WebLVC cannot consist solely of a set of rules of auto-generating JSON messages from an HLA FOM, when HLA is not only possible source for describing semantics.

5.5.3 DISCUSSION FOR SIGNIFICANT RESULT AREA 3

Should the WebLVC specification dictate that messages must be sent over WebSockets, or leave the choice of transport mechanism to the user?

There was consensus that there are many use cases where users might want to send WebLVC messages over transport mechanisms other than WebSockets, just like some people send DIS PDUs over TCP, shared memory, multicast, etc. rather than the “standard” UDP broadcast. But there was also consensus that we should somehow include a WebSocket transport recommendation to allow a-priori interoperability (much as DIS describes its standard practice of using UDP broadcast).

We will leave it up to an eventual PDG to determine how to best describe the idea that the messages are meant to be sent over WebSockets to support browser-based applications, but may also be sent using alternate transport methods to support alternate use cases.

5.6 SUMMARY OF TECHNICAL FINDINGS

Finding No.	Finding Description
1	Representing simulation semantics in JSON messages is a good technical approach.
2	Auto-generating WebLVC messags from an HLA FOM is a good techical approach.
3	Sending WebLVC messages over WebSockets is a good technical approach.
4	Performance of WebLVC is sufficient for the needs of many M&S applications.
5	It is possible to use WebLVC in native (non-browser-based) applications as well.
6	The WebLVC concept lends itself to client-specific interest-management techniques.

5.7 RECOMMENDATIONS FOR FUTURE SISO ACTIONS - THE WAY AHEAD

Recommendation No.	Recommendation
1	The group's strong consensus was to recommend the immediate creation of a PDG to develop a WebLVC Standard based on the concept proposed in the Study Group. The Study Group considered whether we should remain a Study Group until we have a more mature draft of a WebLVC specification, and transition to a PDG later in the process. But there was consensus that it would be better to move to a PDG earlier in the process (i.e. now), so that the development and maturing of the WebLVC specification could happen under the auspices of a PDG. In conjunction with this Final Report, the Study Group will be submitting a Product Nomination to the SAC, requesting the creation of a WebLVC PDG.