

Simulation Integration for Healthcare Education, Training and Assessment

Rachel Ellaway
Northern Ontario School of Medicine
rachel.ellaway@normed.ca

Jeremy Cooperstock
McGill University
jer@cim.mcgill.ca

Bruce Spencer
NRC-IIT
bruce.spencer@nrc-cnrc.gc.ca

Abstract

Funded as part of the 'Network Enabled Platforms' program the Health Services Virtual Organization (HSVO) Project has developed a network-enabled platform (NEP) consisting of a messaging specification, a collection of devices, a bus interface architecture and a middleware hub (SAVOIR) that can support the authoring execution and tracking of multiple independent and integrated simulation activities. The HSVO NEP allows web devices to be coordinated, to talk to each other, to control each other, to use services from elsewhere, to use each other as services. This means that instructors and learners can have access to any services and devices at any time and location thereby supporting both scheduled and on-demand practice or instruction.

1. Simulation Silos

Simulation has come to form an essential part of healthcare education [1]. Despite the many modalities currently in use (such as mannequins, box trainers, virtual patients etc) there is little or no integration between these tools and systems. This limits their use to somewhat siloed and disconnected ways of working which in turn limits their availability and their return on investment.

There is therefore an unmet challenge regarding how simulation devices can be integrated into 'simulation continua' [2]. The Healthcare Services Virtual Organization (HSVO) project was set up to address these issues through the creation of a network enabled platform to support simulation training integration [3]. This paper describes the design and development of the HSVO network enabled platform.

The challenge of multiple independent and isolated simulation devices is not a new one. The development of the High Level Architecture (HLA - now the IEEE standard 1516) by the US military was intended to address simulation integration by federating existing systems [4]. However, the HLA has a number of limitations; it cannot integrate hardware components and it is based on creating

larger simulations from smaller simulation components based on sharing data rather than integrating control.

In designing a platform for integrating medical education simulation devices the HSVO educational developers identified that a central hub-like control model (rather than a peer-peer architecture such as HLA) was required to allow tutors to design and run scenarios and to monitor the actions of the learners and the devices with which they interacted.

2. Developing an Architecture

The authoring and execution of scenarios potentially consisting of multiple interacting devices and users across multiple locations was the primary rationale for the Project with particular applications using mannequin, virtual patient and visualization devices of various kinds.

2.1. Use Cases

In order that the Project could articulate the complex and relatively unfamiliar educational and clinical aspects of the platform to the developers and engineers building the platform the HSVO Project developed two main use cases:

2.1.1. Use Case 1: Active Participation. Groups of learners at multiple locations work through clinical scenarios that start with an on-screen virtual patient activity. At predefined points the path taken by different learners or the values of counters embedded in the virtual patient trigger the platform to move data (such as the simulated patient's vital signs) and then the locus of activity to a high fidelity human mannequin. Some helper devices (such as clinical guidelines) are provided to the learners while others underpin the patient model (such as physiological algorithms).

2.1.2. Use Case 2: Active Observation. A group of learners remotely observe an autopsy or an operation. Using a camera array they dynamically select and share different points of view. Stereoscopic and volumetric data sets are rendered in

response to steps in the procedure or to requests (push) from the tutor or from the students (pull).

2.2. Design Specifications

Given the high level functional requirements for the HSVO platform there were three major challenges to be addressed in the Project:

1. Simulation devices are highly heterogeneous with no standard way of controlling, communicating or otherwise expressing their data and functionality. There was therefore a requirement to standardize the way that different simulation devices could communicate and be controlled. This was addressed within the HSVO Project by the creation of:
 - a. A messaging specification that standardized the expression and transport of commands, reports and other data across the platform.
 - b. A 'bus interface' middleware layer for translating between any given device's native functionality and the HSVO standard messaging schema to allow it to communicate with the middleware core. The bus interface should also accommodate any HSVO functionality not supported by its device. For instance, the bus interface would itself disable access to a device if a native 'pause' function were unavailable.
2. Simulation activities are also profoundly heterogeneous. While some devices allow their scenarios to be pre-defined, these scripts are highly device specific. There was therefore a need to create a common activity definition format to allow for multi device activities to be authored and executed using the HSVO platform. This was addressed by the creation of an integrating and control tool called SAVOIR (Service-oriented Architecture for a Virtual Organization's Infrastructure and Resources) that could not only simultaneously control multiple devices, sessions and activities, but also allow for the authoring of activities and the presentation of data and communication services between participants.
3. There is currently no standardization of the technologies or techniques used to express the tools or services they would need to use. The platform therefore needed to be technology and context independent allowing for any programming language and network technology to connect and integrate with the platform. The open source MULE Enterprise Service Bus was used to

provide a technologically agnostic framework for the input and output of HSVO messages and different teams deliberately used different programming tools such as Java, Python, C# to test its technical ubiquity.

3. Devices as Services

The specific services and resources integrated within HSVO include:

- Simulations based around narrative and gaming components (virtual patients) using OpenLabyrinth, a free and open source virtual patient toolset. Any kind of virtual patient design can be rendered and integrated using this web-based toolset. As such it represents other services including learning management and assessment systems
- Human patient simulators in the form of mannequins were represented by using Laerdal's SimMan 3G, an untethered physical simulator consisting of a robotic mannequin and a number of computers and adjunct equipment such as RFID tagged 'drugs' and other interventions.
- Camera arrays consisting of user-selected real and virtualised (image-based rendering) camera views. These devices were developed specifically for this project.
- 3D visualization using the Remote Stereo Viewer (RSV) and VolSeg tools (www.digitalanatomy.org) working with stereoscopic and volumetric data sets.
- Physiologic algorithms represented by a mathematical model of hypovolemic shock. This involves setting starting conditions after which the patient will bleed out in a certain time. The model changes based on different kinds of user actions.
- Adjunct data sources such as clinical guidelines database and bibliographic resources such as Medline.

These devices are connected through a 'bus interface' that translates between what the device can do and what the platform can do. Each bus interface is keyed to its device to create a service in tandem with its device (see figure 1).

Multiple services can be connected to the HSVO network hub called SAVOIR over a common message bus implemented using Enterprise Service Bus (ESB) technologies (see figure 2).

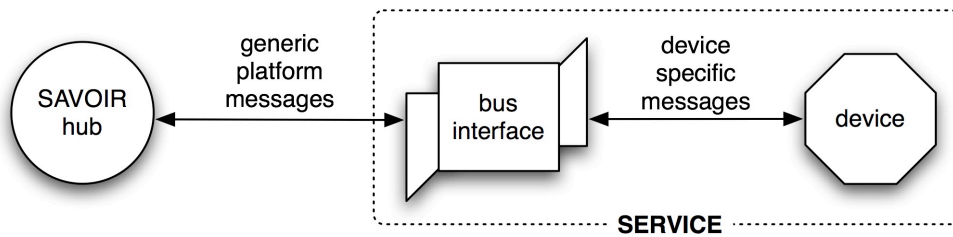


Figure 1: a device is connected to the HSVO platform through a bus interface that translates between the standard HSVO messaging protocol and the device’s native methods. Together a device and its bus interface constitute a ‘service’.

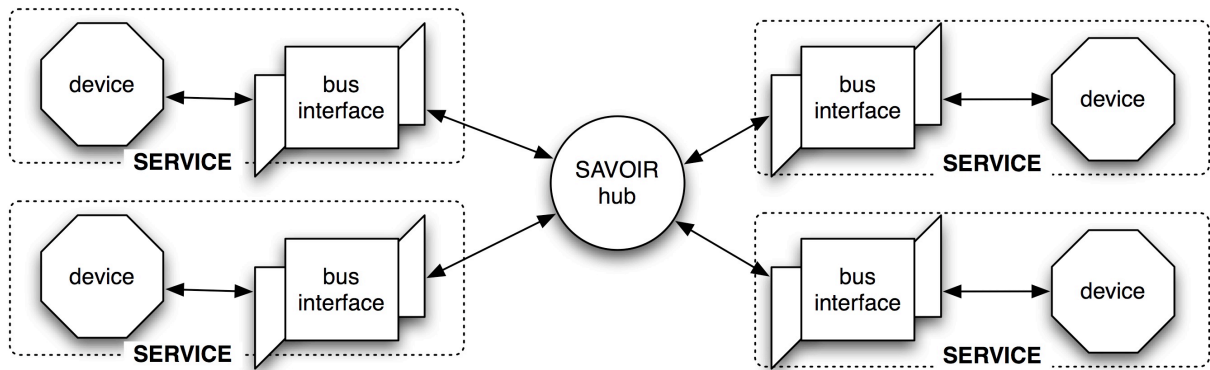


Figure 2: the HSVO platform is comprised of multiple device/bus interface ‘services’ interacting through a common middleware hub.

<i>Method</i>	<i>Description</i>
getProfile	A message to a service requesting a report on its current capabilities
reportProfile	A message from a service reporting on its current capabilities
getStatus	A message to a service requesting its state in a specified session context
reportStatus	A message from a service reporting on its state in a specified session context
load	A message to a service telling it to load a local activity and await a ‘start’ message in a specified session context
start	A message to a service telling it to start playing a loaded activity in a specified session context
pause	A message to a service telling it to suspend the loaded activity in a specified session context
resume	A message to a service telling it to resume playing the loaded activity following a previous ‘pause’ message in a specified session context
stop	A message to a service telling it to stop the loaded activity in a specified session context
endSession	A message to all services in a specified session context to stop and end logging

Table 1: methods in the HSVO messaging protocol

Although the project integrated a number of specific devices into the HSVO platform these were intended to be examples rather than the definitive (and closed) range of functionality available from the platform. Other devices can be added as new services

to extend the platform as long as they have some controllable features. A software development kit was created to support the creation of new bus interfaces consisting of code, worked examples and a best practice guide.

4. Scenarios and Rules

The HSVO network enabled platform can be used in ad-hoc mode where services are launched directly from the user interface. However, for training purposes the key capability is that of authoring and running predefined scenarios and sessions where a scenario is a predefined sequence of activities presented through one or more edge services.

The HSVO platform provides a web-based authoring tool for creating scenarios that are made up of one or more services, the configuration for each service and the rules that define the sequence of how each service is presented to the users. For instance, a simple hypothetical scenario could involve learners starting off working through a screen-based simulation and depending on whether they make a critical decision the activity switches to another device such as a mannequin for resuscitation (for a bad choice) or further instruction (for a good choice).

Scenarios can include with multiple instances of the one service (such as multiple instances of OpenLabyrinth loaded with different labyrinths in different sites). Once created a scenario definition can be saved for later use and reuse. The format used is based on Drools (<http://labs.jboss.com/drools>).

A session is created by taking a predefined scenario and adding start and end times along with the participants and the network locations or 'endpoints' where the different parts of the scenario will be rendered. One scenario can be used multiple times to create different sessions.

During runtime, a Drools workflow engine runs the authored session, sampling the state of the different services involved, evaluating rules and when rules are tripped issuing commands to different services to change their state. Other helper services look after users, edge services and session logging.

5. Messaging Protocol

HSVO has developed an XML-based messaging specification to standardize the connection and control of services and the exchange of data between them. More specifically the messaging takes place between the SAVOIR hub and the bus interface as a means of:

- Sampling the service's capabilities outside any given session context (getProfile, sendProfile).
- Sampling the service's state within a session context (getStatus, sendStatus).
- Controlling the service remotely within a session context (load, start, stop, pause, resume).
- There are also methods for taking a variable or parameter value in one service and sending it to another service.

The messaging takes place between the SAVOIR hub and a service's bus interface. The bus interface then translates or otherwise interprets the SAVOIR commands and then communicates with the device and/or controls it accordingly. See table 1 for a selection of core methods in more detail.

Messages are sent to the SAVOIR bus via a common HTTP gateway and messages going back to the bus interface can be sent using different methods such as a device/service-specific JMS queue – see figure 3. While physical devices are by definition location specific (such as the mannequin or camera array) online devices can be launched in many locations. When a user logs into the SAVOIR platform and starts a scenario or launches a service in ad-hoc mode a Java start widget is downloaded and run on that machine. This allows SAVOIR to start and stop web and system level applications on that machine. Clearly this introduces security issues for users of the HSVO platform and work is ongoing to address the security challenges while retaining the ability to field and control multiple devices at multiple locations.

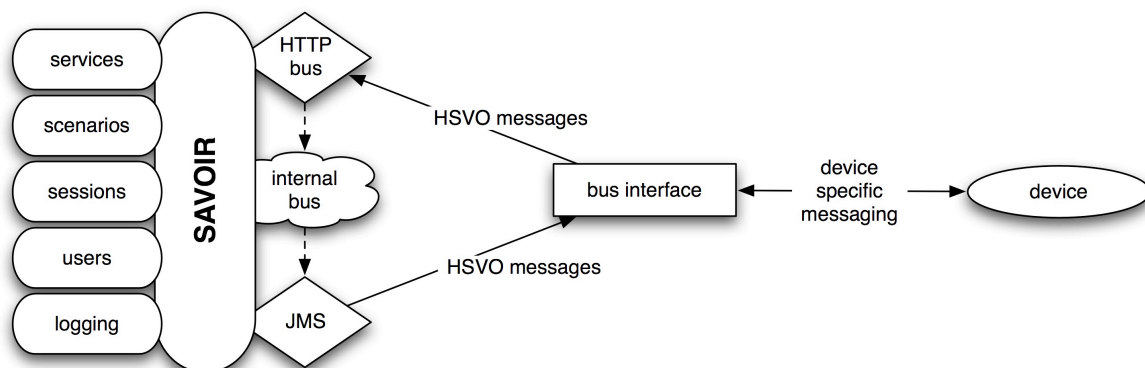


Figure 3: communication between bus interface and SAVOIR bus (HTTP (in) and JMS (out))

6. Network

Because some of its services need very high bandwidth (such as the visualizers and camera arrays) the HSVO platform makes use of an articulated private network between the different partner sites that varies between 2 and 10 Gb/s. In addition HSVO makes use of user-definable lightpaths to connect nodes on its network by taking remote control of the network hardware involved. This involves the ARGIA and CHRONOS toolsets that can take remote control of chains of switches across a layer 1 network [5]. While some resources such as the visualization tools and the camera array require significant bandwidth, others such as OpenLabyrinth and SimMan require very little.

The HSVO platform is designed to work across multiple network nodes and endpoints allowing it to connect participants and services from multiple locations. Bus interfaces can be controlled remotely allowing them to deliver different services to different endpoints (typically a user's computer).

Although lightpath capacity is available, in practice the platform uses a hybrid mix of layer 3 and lightpath connections that reflect the location and forms of access afforded by different devices.

7. The HSVO NEP in Use

The Northern Ontario School of Medicine and McGill University's medical school have healthcare education and training responsibilities for the greater part of Ontario and Quebec respectively with a combined landmass of over 2 million square kilometers. Although distributed healthcare education is clearly an ongoing requirement for all professions and at all levels in remote and rural communities, there are many challenges in bringing high quality activity-based education to them. The ability to locate some devices in communities and provide access to others remotely through an integrated learning environment such as the HSVO platform is clear. Not only can learners remain in their communities allowing clinical services to be maintained but these teams can learn and develop together and with other teams, and a distributed community of professional learning and development can be built and sustained.

The HSVO platform has been used to allow groups of geographically distributed learners to come together around different activities and simulation tools. For instance, one activity involved teams of learners from the Universities of Ottawa and Cork to compare management decisions and their outcomes in a simulated patient case to explore clinical reasoning and the many differences in the healthcare systems in Canada and Eire. Another activity was more competitive between learners from NOSM and

McGill around their management of an emergency case involving the disclosure of some but not all the information available to each team. In both cases the learners used simulation tools along with communication tools (such as the Isabel web conferencing platform - <http://isabel.dit.upm.es>). See figure 4 for a selection of images taken from different HSVO simulation and training sessions.

The full capabilities and forms of use are under investigation at the time of preparing this paper so a definitive description of the breadth and depth of the ways in which the platform can be used has yet to be developed. However, there are a number of key applications that the platform is able to support:

1. Distributed participants – learners, tutors or technicians are located in different locations and interact through multiple services.
2. Distributed services – services and resources used by the services (such as datasets and configuration files) are located at different nodes across the network. As an example the RSV service application can sit on a server in Ottawa while pulling data from a server in Sunnyvale, California and presenting it to a user in Thunder Bay.
3. Integrated services – services can share parameter values. For instance, the same virtual patient may be realized on multiple services by exchanging the same vital signs between services even if the services present them or use them in different ways.

8. Discussion

The challenge of simulation integration, at least at the technical level is addressed through the development and use of the HSVO network enabled platform. Middleware, logic, rules, messages and an extensible connector framework architecture make the platform highly adaptable and extensible for adding services and creating new and innovative scenarios and sessions.

Furthermore, the use of a simple hub-based model addresses the needs of healthcare educators and those of developers building services connected to the platform. The XML messaging model has been designed to be very simple and adaptable

HSVO allows web devices to be coordinated, to talk to each other, to control each other, to use services from elsewhere, to use each other as services. This means that instructors and learners can have access to any services and devices at any time and location thereby supporting both scheduled and on demand practice or instruction.

In use the platform is proving to be both engaging and useful with many new kinds of activities and forms of working in simulation education and training arising from the platform's affordances.



Figure 4: the Hsvo platform in use with different simulation and training scenarios

9. References

- [1] R. Riley, Ed. "Manual of Simulation in Healthcare", Oxford University Press, Oxford, UK, 2008.
- [2] R. Ellaway, R. Kneebone, K. Lachapelle, D. Topps "Connecting and combining simulation modalities for integrated teaching, learning and assessment", *Medical Teacher* 31(8), Informa, UK, 2009, pp725-731.
- [3] R. Ellaway, D. Topps, K. Lachapelle, J. Cooperstock. "Integrating Simulation Devices and Systems" in "Proceedings of MMVR 2009". J. Westwood, S. Westwood, R. Haluck et al. IOS Press, Amsterdam, NL, 2009. pp88-90.
- [4] F. Kuhl, R. Weatherly, J. Dahmann. "Creating Computer Simulation Systems: An Introduction to the High Level Architecture", Prentice Hall, Upper Saddle River, NJ, 1999.
- [5] J Wu, M Savoie, S Campbell, H Zhang "A network management tool for resource-partition based layer 1

virtual private networks." *Int. J. Network Mgmt* 19, Wiley InterScience, Hoboken, NY, 2009, pp139-152.

10. Acknowledgements

Many thanks to CANARIE for funding this activity and to the project team that made it happen: Jeff Blum, Martin Brooks, Scott Campbell, Bryan Copeland, Cristina Dalroti, Parvati Dev, Bruno Emond, LeRoy Heinrichs, Justin Hickey, Bobby Ho, Michael Kirlaw, Kevin Lachapelle, Sandy Liu, Jordan MacDonald, Kapildev Misra, Aaron Moss, Adriana Olmos, Swaroop Patnaik, Alison Peek, Rene Richard, Mark Richards, Roger Sanche, Michel Savoie, Steve Senger, George Shorten, Kevin Smith, John Spence, Haijian Sun, Craig Symington, David Topps, Yonghua You, and Hanxi Zhang.

See the Hsvo Project website for more information: www.hsvo.ca