Test-Beds Using the High-Level Architecture and Other Distributed-Simulation Frameworks

Kiyoul Kim, Tae Woong Park, Rana Riad, Sayli Bhide, Oloruntomi Joledo, Luis Rabelo, Gene Lee, John Pastrana, Mario Marin
Simulation Interoperability Laboratory
Department of Industrial Engineering and Management Systems
University of Central Florida, Orlando, FL 32816, USA
kiyoulkim2010@knights.ucf.edu

Keywords: Space exploration, High Level Architecture, distributed-based simulation, warfighter

Abstract

This paper discusses an environment being developed to model space missions and LVC for warfighter scenarios. Several models representing different phases of missions and engineered systems could be used to abstract complex systems. These models can be built using different simulation paradigms. New tools such as VR-Forces can be utilized to support these environments. A very important feature is the utilization and support of the High Level Architecture (HLA) that provide capabilities to build virtual test beds. This paper presents our on-going work.

1. INTRODUCTION

Our completed initial Virtual Test Bed (VTB) development efforts for modeling space shuttle missions and operations at NASA Kennedy Space Center (KSC) are based on the High Level Architecture (HLA) and the Run-Time Infrastructure (RTI) [7,9]. HLA is a distributed simulation architecture for interoperation and reuse of simulations. It facilitates interoperability among different types of models and simulation applications and promotes reuse of simulation software modules [1]. HLA is intended to provide a general purpose distributed simulation architecture suitable for any type of model and broad range of application including training, logistics planning, analysis, and simulation-based acquisition [8,10]. HLA can support virtual, constructive, and live models and has inherent capabilities for both real-time and logical-time execution.

It is very important to state that HLA was developed in the mid 1990's. In 1995, Defense Advanced Research Projects Agency (DARPA) funded three industry teams to develop concepts for the definition of a high level architecture. The results of these efforts would eventually lead to the development of the DoD HLA 1.3 standards, and subsequently to the IEEE 1516 family of standards which continues to grow. Then commercial developers have been expressly involved in the active development of the IEEE HLA standards from the beginning (1998) [1,4]. The Runtime Infrastructure (RTI), a software implementation of the HLA Interface Specification, defines the common interfaces for distributed simulation systems during the federation execution of the HLA simulation. It is the architectural foundation that promotes portability and interoperability. All shared information exchanged during a federation execution must be passed through the RTI.

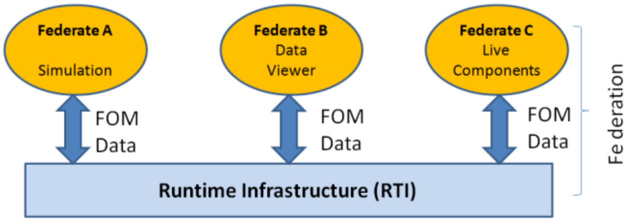

Figure 1. Conceptual diagram of a distributed simulation system using HLA/RTI

The above figure shows a logical view of an HLA federation. Like the figure, multiple federates exchange data with each other during simulation execution. The simulation data exchange follows a Federation Object Model (FOM). The RTI provides a general set of services that support the simulations in carrying out these federate-to-federate interactions and federation management support functions. All interactions among federates go through the RTI.

The objective of the VTB developments is to provide a collaborative computing environment that supports the creation, execution, and reuse of simulations that are capable of integrating multidisciplinary models representing the elements of complex systems (e.g., space exploration, military systems) [2]. Our new developments with the VTB are based on a layered approach. The enhanced VTB architecture design approach adopts the benefits of layered architectures and more flexible middleware solutions to achieve a desirable interoperability and scalability distributed simulation platform.

This paper expands on lessons learned from our initial developments carried out in order to model complex systems.

2. VIRTUAL TEST BED

We are building an enhanced VTB using a distributed hierarchical simulation platform based on HLA and cloud computing and LVC. These are very unique developments. These demos will be utilized to measure the flexibility of an approach for mission design, validation of strategies, and advancements in tackling complex problems where advanced engineered systems are used.

A deficiency of the HLA is that it is not well suited for large-scale distributed simulation systems. Hence, a cloud based simulation system can enhance the capability of the HLA. Cloud computing provides computing services remotely to users through the internet, thereby minimizing the burden related with managing computing resources and facilities [5]. The benefits that can be realized from cloud computing include but are not limited to on-demand simulation resources, shared and reuse of simulation resources, load balancing capacity improvement. Other advantages of cloud computing are cost reduction, resource sharing and time saved for new service deployment.

HLA provides very few security features when used as a distributed simulation framework. It cannot guarantee integrity and confidentiality of the data exchanged between different federates connected through the web. There are possibilities of intrusion as illegal users can access network through web enabled HLA/RTI and any federate may connect and get access to data exchanged between federates [13]. It is also possible for intruders to tamper with the data in transmission networks. To deal with security problems involved in web enabled HLA/RTI, cloud security features such as Hypertext Transfer Protocol Secure (HTTPS), Identity-based cryptography (IBC) and Public Key Infrastructure (PKI) can be adopted. The communication between federates and RTI needs security checks and also requests for data requires authentication. Users can be authenticated to prevent unauthorized users joining the federation and sensitive data can be encrypted to maintain the confidentiality.

Tablets provide ease of operation over traditional desktop computers. Tablets can even provide simplicity over laptops. Apple, Samsung, Amazon, Google, Microsoft are some of the leading companies involved in the production of tablets. At present, the most widely used operating systems on tablets are iOS by Apple and Android by Google. Tablets are light in weight which makes them more portable. However, they provide less storage space as compared to desktops or laptops. To overcome local storage space and processing power drawbacks, tablets can work in conjunction with the cloud.

The application of tablet computing in the cloud can provide flexibility of operation in spacecraft and military systems. Tablets can be used by astronauts as mobile devices for monitoring and visualization of space systems. The tablet can work as a display interface, while all computing and processing is done via the cloud. Data processed on the tablet can also be saved into cloud. Astronauts can query the system, input their observations and perform online data mining to spot trends through the use of tablets. With voice and gesture recognition, astronauts can connect with components to form "network ontology". Using the computing hierarchical/distributed infrastructure, astronauts can also study correlations and run simple simulation models of the current observed situations.

The addition of Web Sciences [9,12] to this distributed simulation and cloud computing scheme will allow for other types of user interfaces and applications in space exploration. This is very unique and has the capability to include in the future mixed-reality approaches.

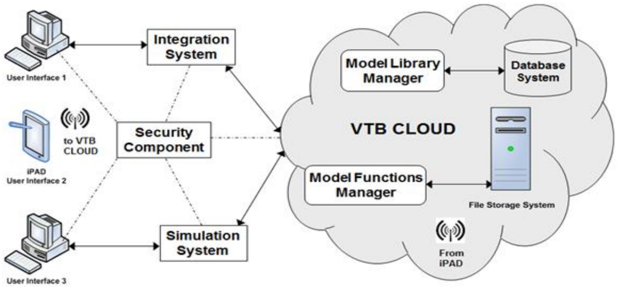

Figure 2. VTB with HLA, Cloud Computing, and Web Sciences

3. AN EXAMPLE OF THE NASA SHUTTLE GROUND OPERATION

As an example of the current developments in particular in visualization, we explain the NASA Shuttle Ground operations. We used Google SketchUp (http://www.sketchup.com/) in order to build the different 3D components.

Google SketchUp is a free 3D design and modeling software released by Google. It was designed for users who feel difficulty using high-level 3D modeling software such as AutoCAD. The software provides easy-to-use Graphic User Interface (GUI) and enough feature richness, so users can design any architectural model from conceptual and detailed designs. Figure 3 shows the Vehicle Assembly Building (VAB) in the Kennedy Space Center (KSC).

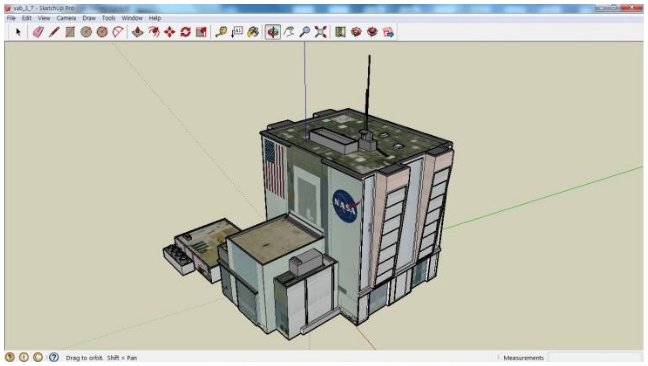

Figure 3. Vehicle Assembly Building (VAB) in Google SketchUp

SketchUp has many advantages as a 3D modeling software for this project. SketchUp supports most of the model formats supported by Computer Generated Forces (CGF) simulation platforms such as VR-Forces (http://www.mak.com/products/simulate/computer-generated-forces.html). These platforms have the resources to create synthetic environments with urban, battlefield, maritime, and airspace activity. VR-Forces supports several industry standards, objects, and formats such as 3DS, COLLADA, OBJ and OpenFlight. Since SketchUp has a rich web based 3D model repository named the 3D Warehouse, users can download high fidelity models from the repository. Users also obtain support from the community. After downloading Space Operation models from the repository, we imported the models to VR-Forces scenario through the Entity Editor which enables editing 3D models and parameters. We found that COLLADA format showed the best quality in the VR-Forces. Figure 4 shows the Space Shuttle model edited in the Entity Editor.

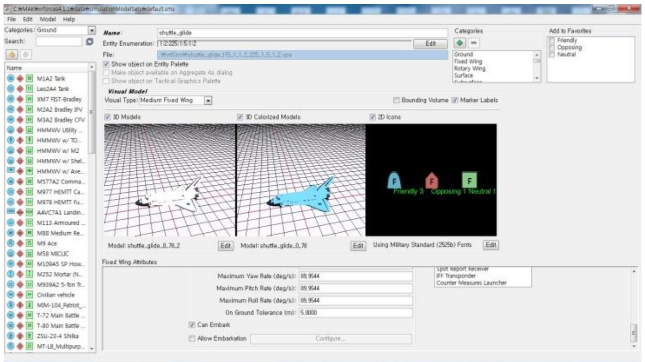

Figure 4. Space Shuttle Model in the Entity Editor

VR-Forces is an all-encompassing mission generator complete with simulation of hundreds of models and terrains as well as scenario editing capabilities. VR-Forces has reactive simulation using artificial intelligence. The models used in the scenario may react to other properties of objects placed in the scenario as well. Scenarios are easy to build using two-dimensional or three-dimensional views for positioning each object into the terrain. Models may also be edited and configured to meet the needs of the scenario. Vehicle dynamics, sensor capabilities, and reactivity to the environment are all capabilities that are offered by the simulator. VR-Forces in the NASA space mission simulation was used to demonstrate the path of action for the space shuttle. The image of the Kennedy Space Center (KSC) on the Atlantic coast terrain was obtained and inserted into VR Forces as the platform of the mission. Models used in the mission were the space shuttle imported from Google SketchUp, the Vehicle Assembly Building (VAB), and other props including the runway, vehicles, and pedestrians. The mission was designed in three stages. Stage one showed the path of the space shuttle from the Orbiter Processing Facility (OPF) to the VAB (Figure 5).

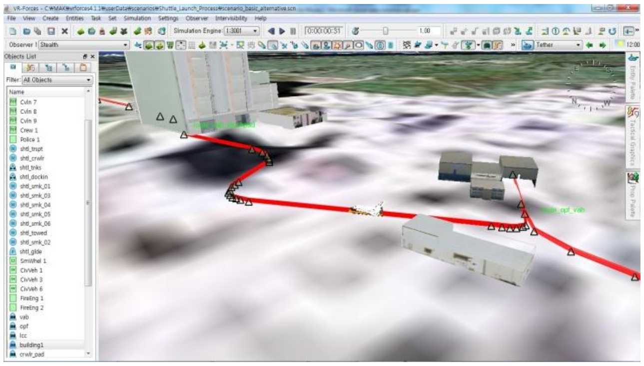

Figure 5. Space Shuttle moving from OPF to VAB

The shuttle then proceeds to the launch pad for lift off to the International Space Station (ISS) as shown in Figure 6.

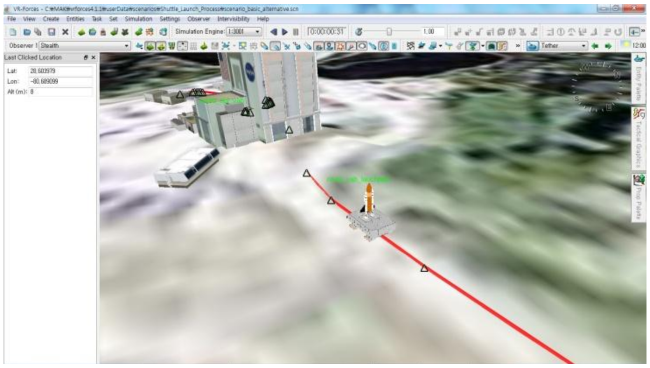

Figure 6. Space Shuttle on the Crawler (from VAB to the Launch Pad)

The control logic of the entire mission is modeled using a discrete-event simulator (www.anylogic.com). The Orbiter is processed in the Orbiter Processing Facility (OPF). When the hardware problems are resolved, system or component repairs/ replacements are completed and

Orbiter modifications are achieved, the vehicle is prepared for roll over to Vehicle Assembly Building (VAB) for vehicle operations. Then the vehicle is transferred from VAB to the Launch Pad (LP). Next step is to launch the Shuttle in to the Space to reach International Space Station (ISS). The final step is the entry and landing of the Shuttle at a particular location (e.g., KSC or Vanderberg, California). The flowchart is as shown in the Figure 7.

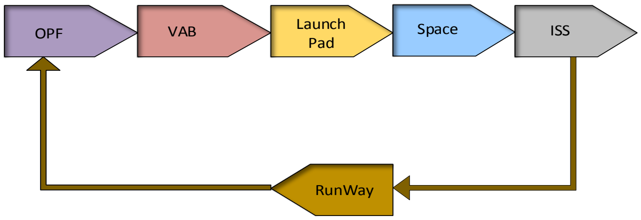

Figure 7. Flowchart of Space Shuttle Operation

This model in AnyLogic is a probabilistic simulation model of the operational life cycle of the Space Shuttle through ground facilities at Kennedy Space Center (KSC). Flight operations such as ascent, mission duration and landing are also modeled. This discrete event simulation model is built by consulting NASA experts and using the processing times/features of the NASA Shuttle as a baseline.

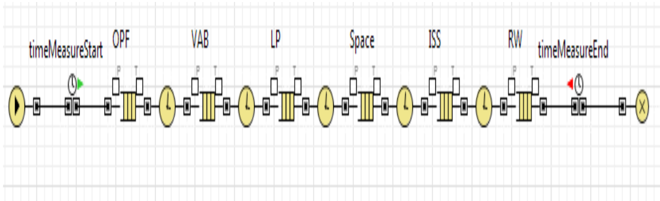

Figure 8. Discrete Event simulation model of Shuttle Operations using AnyLogic

With the utilization of the RTI and the different federates, it is possible to drive VR-Forces from the AnyLogic Model and then, a sophisticated visualization, animation, simulation is performed.

4. LESSONS LEARNED: SEVERAL RTI PLATFORMS ARE AVAILABLE AND YOU HAVE TO SELECT AN APPROPRIATE ONE

One of the lessons learned during this initial effort was the selection of the RTI. The performance of the RTI is crucial to the optimization of the federation. For this reason, the evaluation and choice of an RTI was considered during the design phase. The implementation language of the RTI can have an impact on performance. For example, Java implementations may require more system resources while the cross-platform nature of Java enables it to run without modification on any Java-enabled platform. Other independent variables that affect performance include: number of federates, distribution of federates, Data Distribution Management, network transport mode, objects per federate, attributes per object, interactions per federate, parameters per interaction, attribute buffer size, interaction buffer size, and data bundling. The effects of these independent variables on measures of comparison such as latency and throughput should be evaluated before a choice of an open source or commercial RTI is made [6].

Commercial RTIs are more robust in operation than open source RTIs. Commonly used commercial HLA-compliant RTI implementations are the MÄ K Real-time RTI, Pitch portable RTI (pRTI) and RTI Next Generation. For this experiment we utilized the MÄ K Real-time RTI.

Designed for superior performance, MÄ K RTI implements the HLA specifications in C++. The MÄ K RTI supports a wide variety of network topologies and architectures. It has been verified by the Defense Modeling and Simulation Office (DMSO) to be compliant with the HLA Interface Specification and the IEEE 1516. The MÄ K RTI has exceptionally low latencies as compared with other RTI implementations [6] and it is compatible with Windows (7/Vista/XP) and Red Hat Enterprise Linux.

5. LESSONS LEARNED: ADVANCE VISUALIZATION IS IMPORTANT

Another important lesson learned was related to visualization. Visualization is an important feature of modern simulation modeling environments. As our research of different visualization paradigms continues, we find that two types of visualizations are required in the context of the VTB distributed simulation [6]. First, a visualization of data and/or the specialized functions is an essential part of Commercial Off-The-Shelf (COTS) tools. In order to integrate the visualization tool into the VTB, a federate has to be created. This federate will interact with both the RTI and the visualization's external interface. A second type of visualization will have a simulation engine which includes a set of integrated animation facilities to display the state of the system being simulated, which may allow user-model interaction.

Our research has found that there are many visualization tools available. In addition, another system with distributed capabilities and one of the most popular and complete simulation and visualization COTS available is SIMbox from SIMGON (http://www.simigon.com/overview.html) a Modeling, Simulation & Training solutions provider. It is a platform which provides the ability to create, modify, manage and deploy any simulation-based content.

6. LESSONS LEARNED: HIGH-PERFORMANCE COMPUTING PLATFORMS CAN BE USED TO SPEED UP AND SUPPORT MULTIPLE-RESOLUTION MODES

Our current work with warfighter scenarios and LVC has required the addition of "accelerators" in order to improve the real-time capabilities and data integrity. We have used high-performance computing platforms such as the Synchronous Parallel Environment for Emulation and Discrete Event Simulation (SPEEDES) [3,11] with success. Our developments are benchmarking WarpIV (the next generation of SPEEDES - http://www.warpiv.com/) in order to be integrated to our VTB. Performance-based training required the generation of sophisticated environments (e.g., realistic) and required the solution to higher-order levels of mathematical equations.

7. CONCLUSIONS

Distributed simulation is very important to tame complexity. It is essential to emphasize the hybrid nature of distributed simulation models where discrete-event and continuous models are required due to the nature of the engineered systems [8,10]. There are many sources of expertise required to build and model these engineered systems. Then, there is a need for different type of models to have the analysis capability to encompass their sub-systems, processes, and life cycles.

This approach of Hierarchical/Distribution simulation modeling can be used for planning at different levels (i.e., strategic, operational, and tactical). It is very important to appreciate the level of integration to be achieved with other information systems and the real-time issues involved in particular for advanced concepts. Scripted visualization and simulation visualization are very different concepts. Simulation visualization is the one requested by the analysts while scripted visualization is just a replica of a movie.

This paper outlined some of our preliminary work that will evolve toward a more sophisticated and responsive simulation environment. We will report our progress in future papers.

References

[1] Dahmann, Judith S., Richard M. Fujimoto and Richard M. Weatherly. 1998 "The DoD High Level Architecture: An Update." Paper presented at the Winter Simulation Conference, Washington, DC, December 13-16.

[2] Dawson, Jeffrey W. 2006. "A Holistic Usability Framework for Distributed Simulation Systems." PhD diss., University of Central Florida.

[3] Fujimoto, Richard M. 2003. "Parallel Simulation: Distributed Simulation Systems." Paper presented at the 35th Winter simulation Conference, New Orleans, Louisiana, December 07 - 10.

[4] Fujimoto, Richard M., Asad W. Malik and Alfred J. Park. 2010. "Parallel and Distributed Simulation in the Cloud." *SCS M&S Magazine*, July.

[5] He, Heng, Ruixuan Li, Xinhua Dong, Zhi Zhang and Hongmu Han. 2012. "An Efficient and Secure Cloud-Based Distributed Simulation System." *Appl. Math* 6: 729-36.

[6] Malinga L., and Willem H. le Roux. 2009. "HLA RTI Performance Evaluation." Paper presented at the 2009 SISO European Simulation Interoperability Workshop, Istanbul, Turkey, July 13-16.

[7] Marin, Mario, Luis Rabelo and Jose Sepulveda. 2006. "Spaceport Simulation Models Integration." *Journal of Aerospace* 114: 1264-71.

[8] Rabelo, Luis, Hamidreza Eskandari, Tarek Shaalan and Magdy Helal. 2007. "Value Chain Analysis Using Hybrid Simulation and AHP.*" International Journal of Production Economics* 105: 536-47.

[9] Rabelo, Luis, Paul Fishwick, Zach Ezzell, L. Lacy and Nabeel Yousef. 2012. "Ontology-Centred Integration for Space Operations." *Journal of Simulation* 6: 112–24.

[10] Rabelo, Luis, Magdy Helal and Albert Jones. 2005. "Hybrid Simulation for Enterprise Simulations." *International Journal of Computer Integrated Manufacturing* 18: 498-508.

[11] Steinman, Jeff, et al. 1999. "The SPEEDES-Based Run-Time Infrastructure for the High-level Architecture on High-Performance Computers." Paper presented at the Advanced Simulation Technologies Conference, San Diego, California, April 11-14.

[12] Wainer, Gabriel A., Rami Madhoun, Khaldoon Al-Zoubi. 2008. "Distributed simulation of DEVS and Cell-DEVS models in CD++ using Web-Services." *Simulation Modeling Practice and Theory* 16: 1266-92.

[13] Zhang, Zhihui, X. D. Chai and B. C. Hou. 2011. "System security approach for web-enabled HLA/RTI in the cloud simulation environment." Paper presented at the In Industrial Electronics and Applications (ICIEA) Conference, Beijing, June 21-23.