UR2015: Technical Integration Lessons Learned

Richard Williams
Alion Science and Technology
Norfolk, VA
rlwilliams@alionscience.com

Shane J. Smith
Alion Science and Technology
Norfolk, VA
ssmith@alionscience.com

ABSTRACT

In 2006, the United States Joint Forces Command (US JFCOM) Joint Innovation and Experimentation J9 Directorate conducted the Urban Resolve 2015 (UR2015) Experiment. UR 2015 was designed to examine specific solutions to the challenges that will likely confront U.S. military forces in the future urban environment. This "human in the loop" experiment provided training for senior military personnel in decision-making processes by stimulating real-world Command, Control, Communication, Computer, and Intelligence (C4I) systems using an array of simulation technologies. The experiment involved more than 1,000 people at 19 different sites across the United States. It featured extensive use of modeling and simulation (approximately 30 individual simulations including Joint Semi-Automated Forces (JSAF) and OneSAF Testbed (OTBSAF)) running on over 450 computers to create a robust virtual environment that replicated what the urban environment may be like in the future after a major crisis has occurred.

This paper will begin by providing background information on the numerous sites and applications that had to come together to create the UR 2015 federation. Additionally, it will examine the tasks required to integrate these sites and analyze not only the successes, but just as importantly the problem areas encountered. This paper will conclude with guidelines and recommendations for streamlining complex integration efforts when incorporating numerous, diverse simulations distributed over a large number of participating sites.

ABOUT THE AUTHORS

Richard Williams is the Deputy Technical Director for federation and Joint Semi-Automated Forces (JSAF) development at J9. He has participated in federation integrations for Attack Operations 00, United Vision 01, Millennium Challenge 02, and the Urban Resolve experiments. Richard is an Advisor for Alion Science and Technology with a B.S. in Computer Science from the University of Central Florida.

Shane J. Smith is a Simulation and C4I systems integrator for the core JSAF M&S team at JFCOM J9. He has participated in Multi-National Event 4 and the last two phases of the Urban Resolve experiments. Shane is a Software Engineer for Alion Science & Technology with a B.S. in Computer Science from the College of Charleston in South Carolina.

UR2015: Technical Integration Lessons Learned

Richard Williams
Alion Science and Technology
Norfolk, VA
rlwilliams@alionscience.com

Shane J. Smith
Alion Science and Technology
Norfolk, VA
ssmith@alionscience.com

Introduction

In the world of software development, integration is often considered a formalized and strictly defined process. In the agile software development practices integration is a continuous process that takes place through unit tests and daily software builds. While well defined integration practices are easily implemented for stand-alone software projects, what about the integration of multiple applications at diverse locations that have been developed by engineers using different processes and ideas? How does the director of a federation of simulations deal with the challenges posed by integrating a diverse group of participants?

In 2006, the United States Joint Forces Command (US JFCOM) Joint Innovation and Experimentation (JI&E) J9 Directorate completed the final phase of the Urban Resolve Experiment. Like the initial phases, this final phase involved the simulation of urban military operations with rapidly evolving conditions, changing sensor coverage, and injecting the experiment with a large number of simulated civilian entities. This phase, however, also introduced a number of new simulations, Command, Control, Communications, Computer, and Intelligence (C4I) systems, and sites that had not participated before.

The introduction of new and disparate components and technologies within a compressed preparation schedule provided the JFCOM J9 Modeling and Simulation (M&S) team with its most challenging integration to date. This paper shall briefly describe the Urban Resolve 2015 (UR2015) simulations and their locations. It shall then discuss lessons learned in integrating the applications and locations from both a software and network perspective and examine both the successes and failures.

Background

The integration schedule for the third (and final) phase of Urban Resolve consisted of three one-week integration periods separated by month-long development cycles. These integration periods were then followed by three spiral events where operators were able to test and verify the status of the simulation, and then three practice trials. Following the final practice trial, the official UR2015 events took place.

Each site participant in the federation was assigned a lead to keep track of the progress towards meeting goals, and to provide updates to the technical director.

Table 1. Participants in Urban Resolve Phase 3

Participant	INTER-FACE
JSAF	RTI-S
Culture	RTI-S
SOAR	RTI-S
DIS/HLA Gateway	RTI-S
Track Database	RTI-S
SAODB	RTI-S
SEAS	RTI-S
SLAMEM	RTI-S
DTSIM	RTI-S
MODSTEALTH	RTI-S
SSP	RTI-S
TACSIM	RTI-S
Data Loggers	RTI-S
SNN	RTI-S
WALTS	RTI-S
OASIS	RTI-S
CBSIM	RTI-S
CAD	RTI-S
WebTAS	TACSIM
OTB	DIS
FIRESIM	DIS
EADSIM	DIS
ADS	DIS
ATL	DIS
TRMS	DIS
HEL/B2	DIS
CPOF	GCCS
GCCS	C4IGW
CCD16	C4IGW
C2PC	GCCS

Goals for the federation-level integration efforts were separated into three groups by priority. Priority one goals consisted of terrain correlation, road and traffic correlation, support for distributed sensor protocols, building/structure correlation, and dynamic terrain support. Priority two items consisted of object-naming conventions, distributed logging, data analysis, and enumeration control. Priority three items focused primarily on monitoring, pause, resume, save, and restore capabilities.

Each of the integration goals were broken down into finite testable elements which could be documented and distributed to all the applicable participating simulations for verification purposes.

Since the UR2015 experiment comprised a large number of teams, we attempted to maximize developer efficiency by conducting as much testing as possible in parallel. A test plan was created for each event that provided the purpose, schedule, systems, and primary objectives. Daily morning and afternoon meetings were scheduled to plan any testing requiring coordinated action.

At the end of each event, each participant's status was documented and published on a shared website for all participants to view. By constantly monitoring goal progress we were able to focus resources on the areas that needed the most assistance.

UR2015 Simulation Integration

During UR2015 the need to provide integration for a large number of diverse simulation systems was evident from the initial architectural designs. There was an obvious requirement for both Distributed Interactive Simulation (DIS) simulations and High Level Architecture (HLA) simulations to interact for the UR2015 experiment to be successful. These simulation systems included the U.S.Army's OneSAF Testbed (OTBSAF), several U.S. Air Force simulation systems, as well as a number of other HLA simulation systems which will be discussed briefly. Figure 1 provides a high-level view of the initial simulation architecture envisioned at the beginning of the UR2015 experiment integration effort.

HLA Component Integration

The HLA integration for UR2015 involved approximately eighteen primary applications. These applications are outlined in Table 1 where it lists "RTI-S" as its interface. In addition to the more common problems associated with HLA integrations, the use of the new JSAF C2 (Command and Control) (Helfinstine, et al 2005) architecture brought interesting problems to the table.

A common problem encountered during HLA integration included coordinating upgrades to the common elements of the federation. These common elements include the Run-Time Infrastructure (RTI), the Run-Time Initialization Document (RID) file, and the Federation Object Model (FOM). Throughout the early stages of the experiment there were constant changes to the FOM as well as updated releases of the RTI and RID files. Incorporating a large number of HLA simulations that were not directly built and controlled by the core J9 M&S team complicated this task and required a closely coordinated effort. It was critical to advise all participants of upcoming modifications prior to any changes being made. Next, a period of time was scheduled to bring down all of the HLA simulation systems for any upgrade and subsequent restart. The developers working on each of the simulation systems were provided time to get the newest changes incorporated and have their systems back up and running in the UR2015 HLA federation.

The biggest problem faced while integrating the HLA federates came with the new Command and Control feature of JSAF. This functionality (also called the JSAF Control Protocol) was designed to replace the long-standing Persistent Object (PO) protocol. This JSAF Control Protocol functionality provided a novel way to control and view objects on both remote and local machines. A JSAF Control Protocol feature was the automatic migration of the ownership of graphical objects to a local JSAF federate in the event of a network slowdown or outage. However, a problem arose when network connectivity was restored and network connections were reestablished. The reconnected applications would attempt to "renegotiate" ownership of the objects that had automatically migrated to other systems. These attempted "renegotiations" would flood the network with data packets which would result in network slowdowns, which would then initiate another round of automatic migration. The federation would become overloaded from this repeating process, crippling the entire federation and forcing a restart. Several solutions were investigated including implementing Data Distribution Management on the JSAF Control Protocol traffic which was considered a high-risk change. Ultimately, the solution used during UR2015 was to turn off ownership migration during federation failures. This option was chosen due to the higher risk level involved with implementing DDM for the C2 traffic.

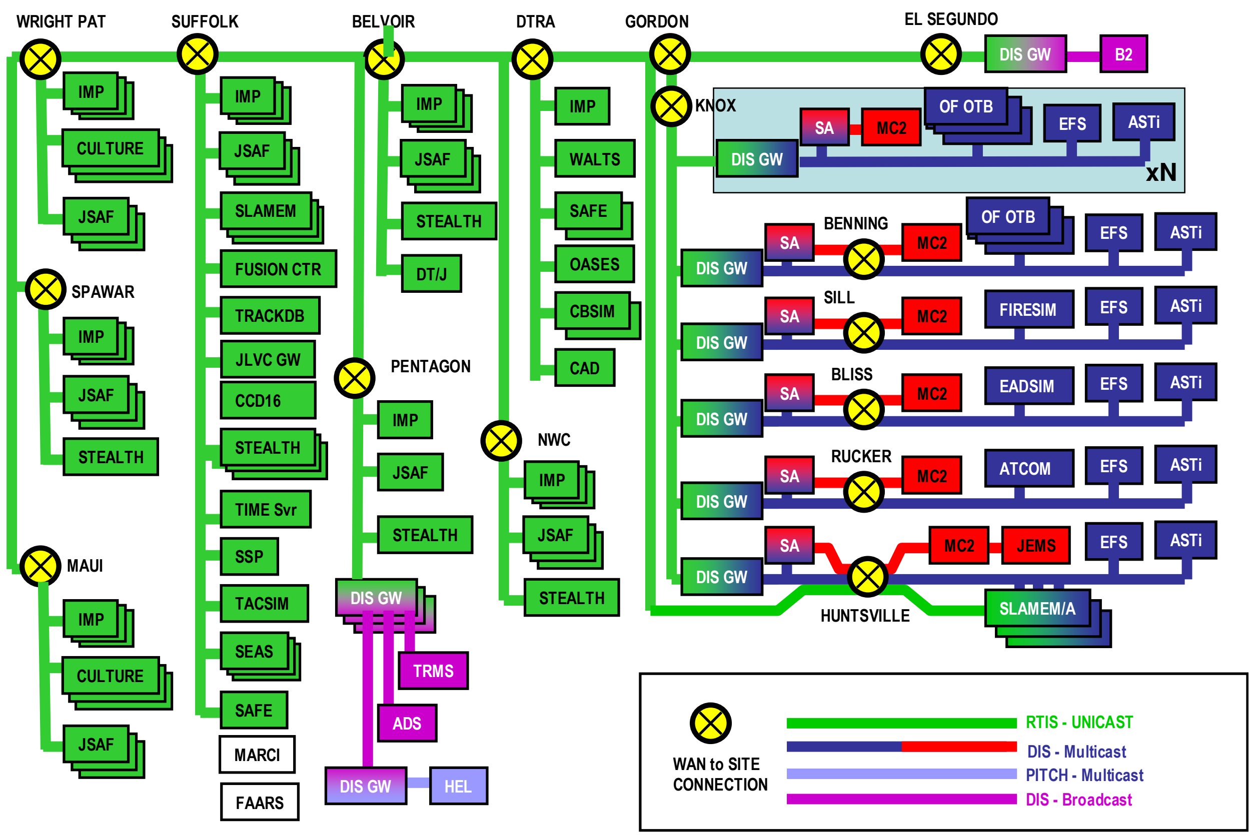

Figure 1: UR2015 Simulation Architecture

DIS Simulation Integration Dilemma

The integration of DIS simulations as a major component to the UR2015 experiment presented some expected, yet significant, issues. One problem was the incorporation of a communications architecture that did not utilize Data Distribution Management (DDM) in the same manner as the UR experimentation environment. Another issue was the differences between how DIS and HLA handled dead reckoning. A third problem was handling sensor footprints across the two architectures. One last problem encountered was merging two different movement models across DIS and HLA. For all but the movement models, the solution was accomplished within one application - the HLA/DIS Gateway.

HLA/DIS Gateways – "Not just a translator"

The HLA/DIS Gateway application within JSAF was one of the most integral pieces of the exercise. In past experiments this gateway allowed DIS and HLA federations to interact with one another by simply "translating" objects and interactions on HLA to Protocol Data Units (PDU) on the DIS network (and vice versa). During UR2015 the gateways not only handled DDM for the DIS federation, they also provided a means to process sensor detections across the two simulation networks, as well enable JSAF's Road-Based Dead Reckoning to be seamlessly used without additional development on the DIS federates.

Data Distribution Management through Gateways

At the time of the UR2015 experiment, the simulation used by the Army and Air Force were limited by the number of entities they could handle. Army systems began to lose performance when approaching the 10,000 entity mark, while Air Force simulation systems experienced the same level of performance degradation at only a few hundred entities. However, on the HLA side of the experiment, CultureSim (a capability in JSAF to simulate the civilian population) could produce up to 200,000 entities. Obviously a solution had to be found to prevent this large number of entities from saturating the DIS networks.

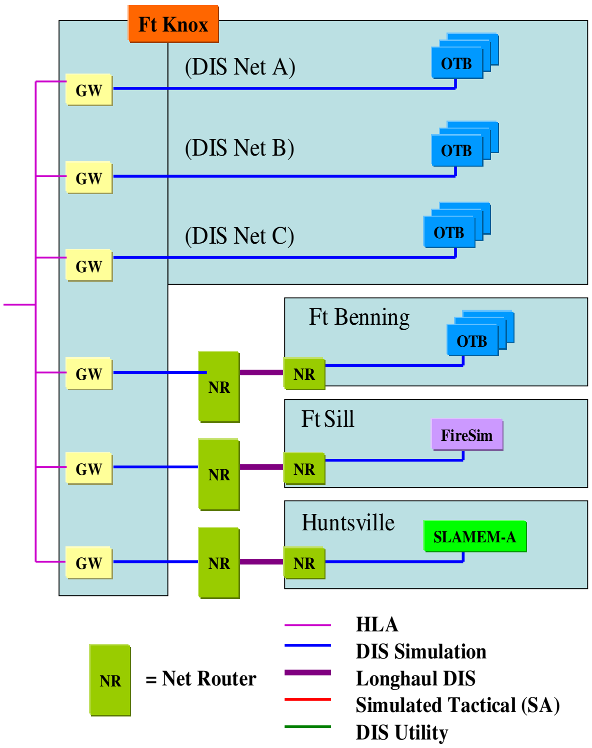

Figure 2: Army Simulation Connectivity

The HLA federation used for UR2015 experiments was able to function with these large numbers of entities by using DDM to minimize the amount of remote entities subscribed to by any simulation at a given time. With the combination of Interest Management Processors (IMPs) and DDM, the amount of HLA traffic on the network can be minimized so only the smallest possible subset of federates will receive a given object (or interaction) in the federation. Currently, no true mapping of this form of DDM exists in the DIS architecture. On DIS, DDM is limited by the number of multicast/broadcast addresses available on the network. To provide some level of DDM, multiple HLA/DIS Gateways were used to transmit DIS packets on separate multicast/broadcast addresses. For the Army simulations, there were originally eight separate HLA/DIS Gateways for eight separate sites/networks. The Air Force systems used a similar method to limit simulation traffic for each of their applications. When entity state PDUs hit their respective gateway, that gateway would create that entity on the HLA side and set up subscriptions using a method discussed in the next paragraph. This allowed different subnets to have their entities dispersed throughout different geographic regions while only receiving entities within their respective operating areas with very little overlap.

The next method used to provide DDM was vehicle-based subscriptions. When the HLA/DIS Gateway receives an entity state PDU it creates a local entity on the HLA side and acts as the simulator for that entity. It then subscribes to objects and interactions based on the entity's associated DDM subscriptions defined in a reader file. For example, if an entity-state PDU for a blue (friendly) ground entity arrives it will look at the gateway's DDM reader file to determine the subscription ranges on various objects and interactions. These subscriptions can be defined on a generic level (i.e. for all blue ground entities) or for specific entity models (vehicle_US_M1A1).

In addition to vehicle-based subscriptions, the Air Force simulations needed to subscribe "on demand" to regions outside the vehicles DDM subscription ranges. To task Air Force simulated air entities, three experimental PDUs were added to the Gateways. The first two PDUs (Attack Order and Mission Status Report) allowed the JSAF operator to use the simulation's Target Pairing Tool (TPT) to task Air Force simulated entities. These Air Force simulated assets would then send a Mission Status Report back to JSAF. Through this method, the Air Force entity would appear in the TPT as a "taskable" unit. Once tasked, the Attack Order PDU provided target attack information to the tasked asset. In response to the Attack Order, the Air Force simulation would send an Interest PDU to define an "interest area" beyond the normal subscription space. The Interest PDU defined various filters for entity domains and the force type of the target to fine tune the number of entities filtered through the Gateway. At this point, the HLA/DIS Gateway starts sending Entity State PDUs within the prescribed region of interest.

Using the Gateway to Handle Sensor Detections

The HLA federation has incorporated an application called Simulation of the Locations and Attack of Mobile Enemy Missiles (SLAMEM) to simulate real world sensors (Toyon 2007). SLAMEM sends out a "footprint" (Ceranowicz & Torpey 2004) to the federation, which is processed by the simulators. All entities within the SLAMEM sensor footprint then perform their own calculation to determine: whether they were within the sensor footprint and, if they were, if they would be detected (e.g., not obscured by a building or concealed under foliage). This model (called "impainted") was added to all objects that were considered "paintable" (e.g., could be detected by a sensor). Instead of creating PDUs to represent the footprints and distributing them over the DIS network, when the HLA/DIS Gateway received an entity state PDU from the DIS side, it would create a corresponding entity with the "impainted" model on the HLA side. This proved to be an effective method in handling sensor footprints.

Using the Gateway to provide Road Based DR for DIS Entities

Another problem encountered during DIS integration was handling Road Based Dead Reckoning (Road DR). Traditional dead reckoning techniques are insufficient in a dense urban environment because they fail to account for road networks. Subsequently, Road DR had been implemented into JSAF to make the movement of culture-based entities more realistic in the simulated urban terrain. The Road DR algorithm calculates a vehicle's position taking into account: the current road a vehicle is traveling on; the road that the vehicle is heading towards; the vehicle's speed of travel; and time the vehicle entered onto the current road (Moyer 2005). The problem encountered during UR 2015 was that OTBSAF does not incorporate anything similar to Road DR for its dead reckoning calculations. To resolve this incompatibility, the HLA/DIS Gateway was designed to perform algorithm 2 dead reckoning (IEEE 1996). Using this algorithm, the gateway compared the entities' dead reckoned position against the updated position last sent from the HLA network every 100 milliseconds. If the threshold of 1 meter and 5 degrees was surpassed an entity state PDU with an updated position would be broadcast.

Merging Different Movement Models

Another problem encountered during the Integration Milestone phases resulted from the difference in the entity movement models in OTBSAF (DIS simulation) and those used in JSAF's CultureSim. While CultureSim entities recognize terrain road networks and avoid collisions with other entities, OTBSAF entities did not "recognize" the thousands of CultureSim entities and, as a result, would appear to drive through any culture entities in their path which confused the training audience. To solve this disparity and make movement more realistic, the CultureSim entities were programmed to "listen" for any blue entities (primarily provided by OTBSAF) driving along the road network and move off the road. This solution allowed the blue entities to drive past CultureSim entities without any apparent collisions observed by the training audience.

C4I Integration

During UR2015, the training audience collaborated on two Common Operational Picture (COP) systems: the Command and Control PC (C2PC) and the Joint Command Post of the Future (JCPoF). It was critical for these systems to be stimulated by the various simulations. The primary method was to provide simulated entity tracks from the various simulations to both C2PC and JCPoF, which allowed the experiment designers to "paint" a picture to elicit a response from the training audience. For UR2015, the Global Command and Control System (GCCS) was used as the primary conduit for information from the simulation and the training audience COPs.

GCCS was fed simulated ground, air, and surface ship tracks from the various simulations through the Joint Live Virtual Constructive Data Translator (JLVCDT) Prototype. Both simulated Over-The-Horizon Gold (OTHGold) reports and simulated Link-16 tracks were used for this purpose

Using Standard Messages for Reporting Non-Standard Information

Unfortunately, strictly using these real-world tools limit the amount and type of data that can be provided to the training audience because both OTHGold and Link16 are fixed messages containing specific types of information. To provide additional information, JSAF developed a tool to expand the training audience's Situational Awareness (SA). This Situational Awareness Object (SAO) tool allows an operator to quickly enter relevant SA data and share it dynamically with other operators (Curiel, et al 2005). These SAOs inserted contextual information into specific geographic areas using graphic symbols and text. JSAF also has the ability to simulate sensor tracks that are generated by SLAMEM and compiling these tracks in a Track Database (TrackDB). Because these sensor tracks and SAOs were closely related, a capability was developed to "attach" an SAO to a sensor track in the TrackDB. This allowed an attached SAO to move in conjunction with its associated sensor track throughout the terrain.

Another issue was the requirement to transmit this same information into the COP. However, since no predefined messages exist in OTHGold, existing messages (JUNIT) would be used for reporting ground entities. To make these messages useful during testing, an optional "Remarks" field in the JUNIT message was used to maximize the information provided by the SAOs and TrackDB tracks. A format defined the information to appear in the track as it appeared on GCCS. Additionally, a unique naming convention was used to easily distinguish them from standard OTHGold ground tracks on the COP.

Time and Time Again…

Dealing with time differences between simulations and C4I systems proved to be troublesome from the start. Experimentation design dictated that the simulated days (24 total hours) be split into three 8-hour "shifts" spread over 3 actual days. Additionally, TrackDB tracks and SAOs in JSAF needed to be "preserved" and displayed on the COP with the reporting time remaining consistent between days. The problem encountered by this arrangement was that the simulation time from the end of one day to the beginning of the next would be 16 hours behind the current real time displayed on the C4I systems.

To counter this, JSAF development was able to successfully preserve the simulated creation time of the SAOs and TrackDB tracks. Next, the JLVCDT prototype was modified to set the time-stamp of these reports to reflect the current real time (based on C4I Time) minus the delta between the simulated time of the last update to the SAO or Track and the current simulation time. This allowed all JSAF created tracks to be displayed in the COP relative to real time which allowed the training audience and analysts to determine the age of a specific track displayed on the COP.

UR2015 Integration from a Network Perspective

The UR2015 federation consisted of federated applications running from 15-20 remote locations. These applications were bound together using RTI-s in a point-to-point tree topology (Helfinstine & Torpey 2003). The base of the tree structure was located at JFCOM which utilized an OC-12 connection to the Defense Research and Engineering Network (DREN 2007). With this diverse collection of disparate components, the UR2015 simulation team needed a greater understanding of the how the simulation and network interacted than ever before.

Understand the Simulation to Network Relationship

In the past, JFCOM federation integrations consisted of two separate teams that were experts in one particular area: the team of developers who understood the simulation; and the team of network personnel who understood the network hardware which the simulation utilized. Communication between these two groups was often limited to "finger-pointing" whenever problems in simulation communication arose.

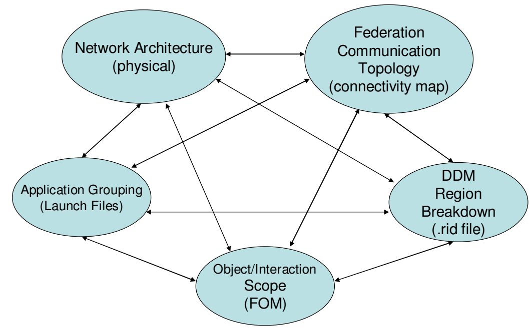

Figure 3: Simulation/Network Relationships

Due to the heightened complexity and scope of UR2015, we created a "Federation Administrator" position whose job was to understand both simulation and network communication layers. This person had to understand both federation object and interactions, and understand how these would be transferred between simulations. This person also needed to understand the physical network architecture so they could design the federation topology and choose which applications would run on which systems. As any major changes were proposed to the simulation/network communication structure, it was the job of the "Federation Administrator" to identify potential adverse effects.

Civilian traffic (created by JSAF's CultureSIM) was a major feature of the UR2015 federation, regularly creating over 200,000 simulated mobile entities. To support this feature, 128 nodes were used on the Scalable Parallel Processor (SPP) system located at Wright Patterson Air Force Base, OH. Using gigabit connections on each node on the SPP allowed for a high level of internal communication. To support vehicle traffic controls (e.g., entities reacting realistically to stop lights and signs) effectively we implemented the simulation of intersections. This greatly reduced and almost completely eliminated vehicle collisions (Speicher & Wilbert 2004). When simulated civilian entities traveled on a road with intersection control, they would transmit their state to the federate simulating the intersection. The internal gigabit connections on each node easily handled intersection traffic over the simulation infrastructure internal to each SPP. The DDM design ensured that the majority of intersection traffic stayed local to the SPP since civilian entities only ran local to that location. The outbound network bandwidth from the SPP was limited to approximately 65 Mbps maximum and usually maintained 50 percent capacity during heavy load times. During the integration events another 128 node SPP in Maui, HI was brought into the federation. The "Federation Administrator" quickly recognized that an operator had inadvertently re-instantiated civilian traffic in identical geographic locations causing the updates to now traverse the WAN links. This additional network traffic saturated all WAN connections and made the federation unusable. While this operator error may have eventually been corrected, having a knowledgeable person specifically trained to react to this type of situation greatly improved integration time.

Certain federates directly or indirectly generated the vast majority of network traffic. In UR2015, SLAMEM generated sensor footprints (Ceranowicz & Torpey 2004) that were sent to all federates simulating entities. These simulations then returned a "sensor detection" object if one of its entities appeared within that sensor footprint. "Sensor detections" typically account for a larger portion of network traffic, and any increase in footprint frequency or size often causes a sizeable spike in simulation network traffic which possibly brings down the network. In this situation, if network personnel simply monitored the traffic levels between simulations, it would appear that the simulator producing the entity caused the problem. However, the entity producer was actually only performing as designed and the change by SLAMEM was the true cause of the network spike.

During the UR2015 integration, we attempted to harden our concept of a federation administration team to bridge the gap between the network, software, and systems engineers. By having federation administrators understand both simulation design and network architecture, troubleshooting times were dramatically reduced.

Add One Piece at a Time the First Time

When integrating a large-scale federation, combining all the components at once is extremely problematic. Experience has shown that many issues can bring down an entire federation in a matter of seconds. However, by formally controlling and serializing the join process during initial integration, fault discovery time was greatly reduced.

During UR2015 we established either video conferencing or voice communications from our technical control center with all participants. As each component joined the federation we used the RTI-s parser to verify that the federate had successfully joined the federation. At this point, we would also observe the network traffic volume going to and from the federate, examine what data streams the traffic was traversing, and simply verify that all metrics appeared to make "sense".

Define the Major Variables on Network Load

Understanding the major factors affecting network load greatly assists in trouble shooting any large federation. By describing these factors thoroughly and discussing their effects with the entire team, focused monitoring can be pre-planned and not simply be reactions to problems that arise.

In the UR2015 simulation there was a direct correlation between simulation time and simulation activity. Visualize the reduced traffic on any urban street at three o'clock in the morning versus the traffic volume encountered at five o'clock in the afternoon during rush hour. Similarly, federation network traffic varied as much as 75 percent based on time of day changes alone. When attempting to predict network traffic levels in an urban simulation, the time of day being simulated must always be considered.

During federation integration load testing it was impossible to duplicate the player audience (e.g., simulation operator) manning expected during the actual experiment. An actively engaged player audience (actively operating a simulation) drives a tremendous amount of network traffic by constantly changing subscriptions, GUI views, and creating objects and interactions. In UR2015, we drastically underestimated the network load that would result from operator interaction with the simulation which accounted for numerous problems that could have been avoided during spiral events.

Many other factors can also cause drastic variances in network load. By attempting to initially define these problems and quantify their effect, we can more accurately predict how many simulation features the network might support.

Set a "Traffic Limit" and Do Not Exceed

During the final trial of UR2015, a number of new remote sites were introduced to the federation with JSAF applications (to permit additional monitoring of specific engagements). Adding these sites was an "emerging" requirement and the available primary J9 DREN bandwidth connection was already near maximum capacity during times of heavy traffic. Despite simulation team warnings that these late additions could cause excessive packet loss and network slowdowns, the third trial proceeded as planned and, as expected, experienced the greatest number of technical problems. In retrospect, the simulation team's warnings may have been too "technical" and were therefore not completely understood by the experiment controllers that generated requirements. This problem my have been avoided by instituting a simple metric, such as a "Traffic Limit", and ensuring all participants understood it's impact and agreed to a maximum threshold.

Have a Process for Isolating Network Spikes

During UR2015 we constantly monitored the network traffic at the network interface for the head Interest Management Processor (IMP) (Helfinstine & Torpey 2003). Since all traffic that would traverse the Wide Area Network (WAN) was routed through this node, it provided the best indication of any spikes in simulation traffic.

When unexplained traffic spikes were observed, the federation administrators would begin a process of utilizing RTI-s provided parser-level commands to display the bytes in and out of the IMP attempting to localize both the publisher(s) of the data and the subscriber(s). Administrators would then traverse the simulation communication structure until they arrived at the source(s) and destination(s) of the traffic spike. Once at these locations, the administrator would use parser commands which break down traffic to a specific stream (Helfinstine 2003) and then attempt to map the stream to the correlated DDM region. With this information, the administrator would then know which developer to notify of the issue.

While the process used for UR2015 traffic isolation was functional, it was not as efficient as we would have wished. There was a tremendous amount of information and statistics available from the parser in RTI-s, however, there was no method available for an application to automatically capture these statistics. Federates could not subscribe to the data and the information could not be queried other than through the parser. Developing an automated tool to query all statistical data available in RTI-s and creating alarm systems would greatly increase the efficiency of federation network troubleshooting. Additionally, developing a capability to automatically map streams to DDM regions would have also sped up troubleshooting procedures.

Test All Network Connections Regularly

During the entire UR2015 experiment, we benchmarked the network's capabilities using a bandwidth measuring tool called Iperf that is available free from the National Laboratory for Applied Network Research (NLANR 2007). As each new site was brought onto the network, we would immediately run a bandwidth test from JFCOM J9 to the remote location verifying both maximum sustained throughput and latency.

Throughput tests consisted of bringing down all simulation traffic, then starting an Iperf server at the remote site and also at J9. Next, we would start up an Iperf client at each location to transmit a constant flow of User Datagram Protocol (UDP) traffic. We would ramp up traffic levels until the maximum sustainable value was discovered. Transmission Control Protocol (TCP) did not make for a good benchmark as its maximum throughput is limited by a function of latency and window size (Eshan & Mingyan 2007). If throughput did not match the results we expected or if data going one direction caused a loss of data in the other direction, we would then use the MTR (My Traceroute) application. MTR combines the features of a standard traceroute and ping and is freely available (bitwizard 2007) under the GNU General Public License (GNU 2007). Executing MTR while traffic was being sent and dropped allowed us to discover the "hop" on which traffic was being dropped and then take the appropriate action to correct the issue.

Network testing was executed weekly and whenever new sites were brought on line. The periodic testing discovered many problems that were introduced from hardware failures and configuration issues. Early in the integration, this process helped uncover numerous system, network card, switch setting, and duplicity issues at the remote sites. By formally defining and performing these network tests weekly, many problems were identified before they had an effect on the simulation.

Conclusion

The number and variety of simulations, sites, personnel, and new concepts for the UR2015 integration pushed the limits of the JFCOM J9 simulation team. This complex situation forced the team to develop new strategies and tactics to address the endless variety of issues that occurred during the entire integration process. Although tremendous progress was made, there is still a great need for improvement of federation troubleshooting tools and procedures.

The UR2015 experiment taught us that fault tolerant architectures can cause unexpected side effects when introduced into a large-scale distributed exercise. We learned that gateways can be used as more than just direct translators and that they can provide Data Distribution Management for architectures that do not directly support the feature. We further determined that integrating movement models across varying architectures may require creative solutions which would not be evident in the initial federation design. We learned that trying to coordinate simulation, real world, and C4I times can be very problematic and require a great deal more attention than might be expected. We also learned that tools for stimulating C4I systems can be used to provide added value for the players within the confines of the defined message passing specification. The experiments taught us that to conduct large-scale distributed simulation you must have personnel that bridge the knowledge gap between network and simulation engineers. Also, we learned that processes must be defined for integrating new applications and locations which verify they meet expected performance benchmarks. Most of all we learned that diligent testing and monitoring are required to integrate a federation of the scale of UR2015.

Significant progress was made during UR2015 in formalizing processes for integrating new components. Also, many lessons were learned that should be shared amongst the simulation community. It is our hope that the lessons learned and expressed in this paper can be used to assist future federation integrations.

ACKNOWLEDGEMENTS

The authors thank Dr. Andy Ceranowicz the JFCOM J9 JSAF Technical Director for his assistance and mentoring. We would also like to thank Steve Bixler for his review and assistance on this paper. Jim Blank for giving us the opportunity to write the paper. Alan Gibson for his assistance on the HLA/DIS Gateways. Bill Helfinstine, Mark Torpey, Dan Speicher, and Laura Dunleavy of Lockheed Martin and the rest of the J9 crew for their continued dedication and support.

REFERENCES

Ceranowicz, A., & Torpey, M. (2004). Adapting to Urban Warfare. *Proceedings of the 2004 Interservice/Industry Training Simulation and Education Conference.*

Speicher, D., & Wilbert D. (2004). Simulating Urban Traffic in Support of the Joint Urban Operations Experiment. *Proceedings of the 2004 Interservice/Industry Training Simulation and Education Conference.*

Helfinstine, B., & Torpey, M. (2003). Experimental Interest Architecture for DCEE, *Proceedings of the 2003 Interservice/Industry Training Simulation and Education Conference.*

Helfinstine, B., Young, C., Brahmbhatt, S., & Levan, M. (2005). The JSAF Control Protocol. *Simulation Interoperability Workshop, paper 05F-SIW-091.*

Moyer, D., & Speicher, D. (2005). A Road-based algorithm for Dead Reckoning. *Simulation Interoperability Workshop, paper 05S-SIW-067.*

Curiel, Jacqueline M., Tran, J., et al. (2005). Developing Situational Awareness Metrics in a Synthetic Battlespace Environment. *Proceedings of the 2005 Interservice/Industry Training Simulation and Education Conference.*

IEEE Standards Committee. (1996) IEEE Standard for Distributed Interactive Simulation – Application Protocols. *The Institute of Electircal and Electronics Engineers, Inc. NY, NY.*

NLANR Iperf Webpage and Iperf license
http://dast.nlanr.net/Projects/Iperf/
http://dast.nlanr.net/Projects/Iperf/ui_license.html

Toyon SLAMEM Webpage
http://www.toyon.com/slemem_gvs.asp

DREN Webpage
http://www.hpcmo.hpc.mil/Htdocs/DREN/

Bitwizard Webpage
http://www.bitwizard.nl/mtr/

GNU Webpage
http://www.gnu.org/copyleft/gpl.html

Eshan, N. & Mingyan, L., *Analysis of TCP Transient Behavior and Its Effect on File Transfer Latency,. Retrieved May 27 from* http://www.eecs.umich.edu/~mingyan/pub/icc03.pdf