MSIAC JOURNAL

March 2008 Volume 3 Issue 1

Modeling & Simulation Information Analysis Center

Following the Pathways:
Information in Modeling and Simulation

Constructive Simulation Versus Serious Games – A Canadian Case Study

Urban Resolve 2015: Technical Integration Lessons Learned

The HLA Federated Compliance Testing and Certification Program

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **MAR 2008**	2. REPORT TYPE	3. DATES COVERED **00-00-2008 to 00-00-2008**
4. TITLE AND SUBTITLE **MSIAC Journal, Volume 3, Issue 1, March 2008**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Modeling and Simulation Information Analysis Center (MSIAC),1901 N. Beauregard Street Suite 500,Alexandria,VA,22311**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**27**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

TABLE OF CONTENTS

The Modeling and Simulation Information Analysis Center (MSIAC) Journal is now available as an automated service. Simply send an email to journal-subscribe@lists.dod-msiac.org to be added to our mailing list. This list is for the MSIAC Journal only and will not be used for any other purpose.

To unsubscribe from our mailing list please send an email to journal-unsubscribe@lists.dod-msiac.org.

If you would like to submit a technical paper for consideration for our Journal, please email it along with contact information to msiachelpdesk@dod-msiac.org.

FROM THE DIRECTOR

Welcome to the March edition of the MSIAC Journal. As modeling and simulation technologies are reaching ever further towards representing elaborate and divergent human behaviors and neural processes, we have chosen to dedicate this issue of the Journal to following these pathways of information sharing. There are a myriad of new techniques for modeling both the complexity of human decision-making and the dynamic sharing of supporting information. Be it through new training modalities, experimentation, or the high level architecture, we believe that following these pathways of information is vital to expanding the effectiveness of M&S.

This issue of the MSIAC Journal presents three distinctive but interconnected articles examining the ways we share M&S information. The paper by Dr. Roman and Mr. Brown highlights simulations that are reaching beyond the battlefield to the representation of command and control centers in training systems. The article by Mr. Williams and Mr. Smith evaluates pathways of information in the process of rethinking network relationships and knowledge sharing in the US Joint Forces Command (USJFCOM) exercise Urban Resolve 2015. Finally, the paper by Mr. Crooks explores enhancements in interoperability provided by the High Level Architecture (HLA) Compliance Testing and Certification Program.

Each of these papers examines how the processing of information directly influences the way in which that information is shared. And we are pleased to be sharing the information in this issue of the MSIAC Journal with you.

Constructive Simulation Versus Serious Games - A Canadian Case Study

By: Dr. Paul A. Roman
Associate Professor, Department of Business Administration, Royal Military College of Canada
Synthetic Environment Advisor, Land Forces Doctrine and Training System Headquarters
Paul.roman@rmc.ca; Roman.pa@forces.gc.ca

Mr. Doug Brown
Manager, Canadian Army Synthetic Environment Repository
Directorate of Land Synthetic Environments
Brown.dg@forces.gc.ca

Abstract

As military forces around the world embrace modelling and simulation as a fundamental enabling technology necessary to help meet training requirements, the impressive characteristics of video game technology and the advent of serious games are increasingly becoming an important part of the training tool kit. The Canadian Army's Directorate of Land Synthetic Environments (DLSE) is charged, in part, with the conduct of command and staff training that is typically supported with a constructive simulation. In addition to simulating the battle, the simulation also stimulates the go-to-war command and control (C2) systems such that the headquarters staff (as the primary training audience) can be immersed in the tactical scenario by performing their usual battle procedures in a mock-up Command Post. After 11 years of conducting exercises in this manner, DLSE supported it's first serious game based exercise in October of 2006. Exercise Winged Warrior is the culminating activity at the end of the Advanced Tactical Aviation Course, intended to train pilots to perform as aviation mission commanders and air liaison officers. This paper takes a critical look at the similarities and differences between exercises primarily supported by constructive simulation versus those supported by a serious game. It also introduces the concept of a training needs framework upon which decisions regarding the most appropriate type of tool to support a training objective can be planed.

1 INTRODUCTION

Few would argue that the pedagogical advantages and impressive levels of resolution offered by the latest in video game technology make it clear that serious games have a role to play in military training. Even if one chose to argue, it would be an uphill struggle as the application of this technology is occurring bottom up as trainers close to the front lines have started adopting and adapting these tools to meet real and urgent training requirements. In the Canadian Forces, several training establishments are using their own budgets to acquire these surprisingly affordable software programs. There is no shortage of choice either as the video game industry comes to appreciate, what from their perspective might be perceived as a niche market, an opportunity to differentiate their products to meet the special needs of military training market. Free trial licences and a willingness to accept feedback and make improvements are good business practices for these companies as they incorporate the needs of military users into products that as a result of the increased realism appeal to a much broader audience. One need look no further than the recently conducted Serious Games Summit held in October 2006 to realize that the serious games are definitely growing in popularity and that training policy makers and planners had best start to figure out where they fit in as part of an overall training strategy. As is the case with the adoption of any new technology, however, there is likely to be some resistance to the change as those comfortable with applying constructive simulation tools based upon time tested training doctrine need to adapt to the changes implied with the adoption of game technology.

In a provocative presentation at the Defence Simulation and Training Conference , Helsdingen suggested that analogous to the way that John Gray characterized the different personalities of men and women, one might consider that "Gamers are from Mars and Trainers are from Venus". To reinforce this analogy, Helsdingen provided the following list of demands in Table 1 that contrast the gamer's preferences (Mars) to those of the trainer (Venus):

Gamer Preferences	Trainer Preferences
Entertainment	Learning process
Emotion Player Control Free Play Unpredictable turn of events Fantasy No Boundaries Social Interaction Surprise Risk Suspense Art and Beauty	Structure Learning Goals Instructor Control Standardization Realistic problems Effective and Efficient Transfer of Training Validity Fidelity

Table 1. Comparison of Gamer and Trainer Preferences

A review of these preferences reveals stark differences suggesting that for the games to be applicable in a military training environment either the games themselves (designed for gamers), or the trainers will have to adapt. This was the challenge faced by the Directorate of Land Synthetic Environment (DLSE) when military trainers with eleven years of experience applying constructive simulation tools to support command and staff training were faced with a different training event that would be better served by a visual gaming environment. The event marked a significant milestone for DLSE, which in addition to the training role, is also charged with the development of pan-Army simulation policy. Winged Warrior offered the opportunity to critically evaluate the application of game technology as a means to assist in the development of an appropriate policy on their effective use. This paper will emphasize the differences and similarities between exercises supported primarily by constructive simulation and this one exercise supported by a commercial off the shelf game. The training needs framework will also be presented as a means to help training planners map which tools are best suited to which requirements.

2 THE TRAINING NEEDS FRAMEWORK (TNF)

One challenge that improvements in simulation technology have created is the introduction of a new lexicon to describe the tools. As natural as the distinctions between live, virtual and constructive simulation may seem to those who are familiar with them, games and serious games in particular have begun to blur the lines between them. Combining these three types of military simulation into synthetic environments built to support a particular training event has also made the distinctions between them even less important.

Those responsible for training policy and planning are far less interested in the tools than they are with the outcomes achieved through their use. The tools are a means to an end and not an end unto themselves. The training needs framework (TNF) was created as a way to map how any tool or set of tools can be applied to produce a particular outcome as part of an overall training plan intended to certify troops for a specific deployment. In the Canadian Army, this training progression has come to be described as the road to high readiness. The culminating activity for a battle group identified for an operational tour is a confirmation event conducted as a live training exercise at the Canadian Manoeuvre Training Centre (CMTC). Achieving confirmation, however, depends on the effectiveness of the up to two years of preparation that occurs prior to the event. While on the road to high readiness, units will go through a training progression that sees them perfecting their individual skills, working in detachments, small teams, combined arms teams and eventually as a full battle group in the context of a brigade level operation. Canadian training doctrine describes seven levels of training, from individual (level 1) to a brigade headquarters collective (level 7) that correspond to this progression. The training needs framework presented in Figure 1 portrays the seven levels and theatre mission specific (TMST) collective training on the left hand side of the matrix and the corresponding training outcomes on the right hand side. Across the top of the TNF the normal progression from skills-based training through discreet vignettes (convoy operation for example) and finally continuous scenarios (a series of vignettes where the trainee must recognize which vignette he is in) portray the increasing levels of context upon which the road to high readiness depends.

Constructive Simulation Versus Serious Games - A Canadian Case Study

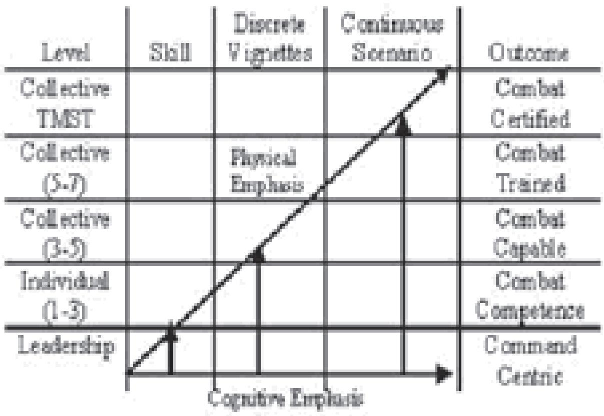


Figure 1: The Training Needs Framework

The term combat in the outcome column of the framework is intended to include a broad definition as appropriate to the mission for which the BG is preparing and could include humanitarian, peacekeeping and other peace support roles as appropriate in the contemporary operating environment. The current Canadian Forces emphasis on being a command centric force and the resultant reliance on leadership is also included somewhat separately from the levels, however, its inclusion is fundamental since the outcome will not be achieved unless there is an appropriate degree of emphasis on leadership and decision making throughout the preparation phases towards certification. It was the lack of a tool that could provide a certification level event within a continuous scenario for a complex tactical aviation mission that lead to the selection of Steal Beasts for the conduct of Exercise Winged Warrior.

3 EXERCISE WINGED WARRIOR

The training arm of DLSE charged with the planning and conduct exercises consists of a group of retired military officers with an average of approximately 28 years of military service followed by up to nine years of military exercise development planning and execution. Prior to this exercise, the emphasis has been on command and staff training conducted through the use of constructive simulations used to stimulate the live command and control systems of the headquarters being exercised. The scope of these exercises has ranged from pre-deployment theatre specific battle group and multinational Brigade exercises to division level exercises in support of the Canadian Forces Land Command and Staff College. Up until the conduct of Winged Warrior, traditional constructive simulations, Janus, JCATS, the ABACUS Command and Staff Trainer, or role players without computer simulation supported all of these exercises.

By contrast, Exercise Winged Warrior had traditionally been a live exercise. Its aim is to test tactical aviation helicopter pilots in their role as aviation mission commanders during the planning end execution of complex missions. Typical missions include:

- Reconnaissance and surveillance
- Direction and control of fire
- Provision of fire support
- Combat airlift/tactical transport
- Logistical transport
- Communications support

To achieve this level of training in a live fire event required the deployment of at least eight utility helicopters and the associated pilots, flight engineers, maintainers, logisticians, operations and command staff. As the primary role of the Canadian Air Force's tactical helicopters is to support the Land Force, the exercise also required the participation of army units with hundreds of ground troops with artillery supported by attack helicopters and jet aircraft. Typically, this was achieved by conducting Exercise Winged Warrior concurrently with an Army exercise. In addition to testing the students, it created a venue to train tactical aviation units' personnel collectively with land force units.

Given the size and complexity of the missions required to achieve the aim of the course, and the very high operational tempo of the Canadian Army, it has become very difficult to bring together all participants needed to provide a realistic training environment and there is no flexibility to add the additional training objectives required by the aviation course. As a result, simulation is now viewed as the only feasible alternative.

In addition to being the only feasible alternative, however, simulation provided several additional benefits to the exercise: friendly land forces can be much larger; they can face a realistic and credible enemy force; supporting

Constructive Simulation Versus Serious Games - A Canadian Case Study

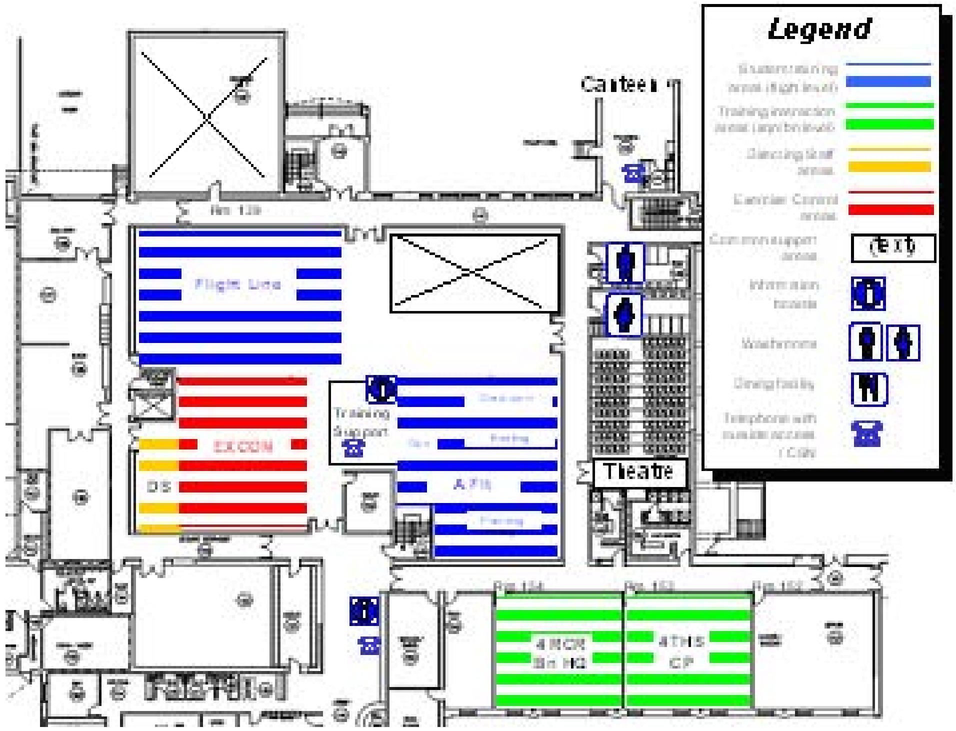


Figure 2 – Winged Warrior Layout

forces like fighter jets, attack helicopters, airborne command and control aircraft and Unmanned Arial Vehicles all can be included. All in all, it gives a much richer tactical environment to support more complex missions, and frees the trainers from having to conduct all of the administrative tasks associated with the coordinating a live exercise.

At the heart of the exercise is the requirement for aviation mission commanders to take part in the planning and execution of the mission. Execution occurs while airborne so the mission commander will make decisions during the mission based on the tactical situation as assessed through both radio information and the visual environment. The limited capabilities of the current fleet of constructive simulations were assessed as inadequate to provide the necessary rich visual environment. The aviation training school had already purchased an adequate number of Steel Beasts licences and so it was selected as the most cost-effective means to meet the requirements of the exercise. Approximately 30 stations were required for the pilots, the directing staff and the exercise controller.

4 PLANNING AND EXECUTION

Preparations for Winged Warrior began only 3 months before with a series of meetings that established the exercise aim, scope and training objectives. Approximately one month prior to the exercise, work commenced on preparing the simulation for use and developing the terrain models.

The layout for Winged Warrior included 98 computers in total, 28 loaded with Steel Beasts, 44 with Sim Radio (a home-grown simulated radio application using Voice Over Internet Protocol (VOIP)) and the remainder as workstations loaded with various applications including Falconview and Microsoft Office. The large training area was set up using dividers, to create a flight line, briefing and planning areas, headquarters areas and an exercise control area as depicted in Figures 2 and 3. The functions of exercise control were to control the simulation, provide inputs from supporting troops, synchronize all activity and provide situational awareness to instructors and assessors.

Constructive Simulation Versus Serious Games - A Canadian Case Study

The majority of the Steel Beast machines were deployed in the flight line area. Two Steel Beast machines were used per simulated helicopter cockpit, one for the flying pilot, and the other for the non-flying pilot. In each simulated helicopter cockpit there were also two Sim Radio machines to emulate the communications networks. In total there were six CH-146 Griffon, two Chinook and two AH-64 Apache cockpits simulated plus a station used as a ground vehicle for a liaison officer and as an Unmanned Arial Vehicle (UAV). The remainder of the Steel Beast machines were located in Exercise Control, one of which was designated as the server. The enemy used two machines, while the others controlled the remainder of the blue and neutral forces. The other non Steel Beasts computers were used in the staff planning process and during exercise execution to aid in ensuring that all radio nets were manned with exercise players in simulated command posts. Each of the Steel Beasts machines had Pentium 4 processors (3.0 GHz), one GB of RAM and 256 MB NVidea graphics card. This hardware configuration turned out to be very suitable for the demands of the exercise.

Figure 3 – Exercise Control

As is the case for any constructive simulation-training event, the Steel Beasts terrain model was constructed from source Digital Terrain Elevation Data (DTED) of the area as well as from VMAP feature data. Steel Beasts has the ability to directly ingest this data and create its own corresponding terrain representation. A lot of detail was added to portions of the terrain to support the various aviation missions that were to be flown. A city was constructed that acted as the main base for the helicopter operations throughout the exercise and several other towns and villages were also constructed if they had an impact on the exercise play.

As a result of the current limitation of an 80x80 Km terrain model for Steal Beasts, two separate terrain models had to be created to accommodate the exercise. A so-called "south" and "north" map were created with considerable overlap. Coordinating the same visual look and feel of both maps over the same terrain area became quite difficult and will be avoided in future exercises. Beyond this limitation, there were only two significant technical challenges to be resolved before the exercise could run. Client machines were dropping out of (crashing) the exercise and the graphics performance was unacceptably slow.

The maker of Steel Beasts, ESim games, was very responsive at helping to resolve these issues. Over a period of 48 hours they provided 3 successive new builds of the simulation each of which progressively addressed the issues described above. Implementation of the final build on a dedicated network with several network services (including firewall) disabled, resulted in Steel Beasts performing flawlessly throughout the exercise period. This was despite running what was from a Steel Beasts perspective, a very large exercise with a very large terrain model.

Graphics performance is something that DLSE is not accustomed to worrying a lot about. For the most part, constructive simulations do not tax the graphics capability of modern PCs. Of course, it became very clear very quickly, that the only thing that mattered in the simulation for this particular exercise was the graphics. A lot of tweaking was done to get a good compromise between scene realism, graphics performance (frame rates) and terrain size. Steel Beasts had acceptable performance when the terrain model was restricted to 60x80 km, which was large enough to run individual helicopter missions. The display was set to be 1024x768 pixels in size. This was also arrived at after a significant process of trial and error.

Tactical aviation is arguably the most difficult (military) case for a serious game considering terrain models and graphics performance requirements. In addition to being relatively fast movers capable of covering large geographical areas, helicopters fly at low altitude demanding a high degree of visual detail. Aircraft flying fast and high can get by with a low-resolution picture draped over a DTED skin. Knowing this, the exercise writers constrained the operations areas considerably. Figure 4 depicts a representative screen capture from the exercise.

Figure 4 – Steel Beasts Screen Capture

Two aviation missions were run each exercise day, one from 10:00 to 12:00 the second at 17:30 to 19:30. This allowed for time before each mission for the control staff to attend the rehearsals and prepare and rehearse their own activities for the next mission. Preparation for each mission included modifying the Steel Beasts scenario with the appropriate forces to properly represent the activities that each mission entailed as well as enemy and neutral forces as appropriate. Again, if the activity did not have a visual impact, observable from the helicopters, then it did not need to be represented in the simulation.

5 TRAINING ASSESSMENT

As the first serious game application for a significant training event conducted by DLSE, exercise Winged Warrior is seen as a milestone for the Canadian Army in terms of the application of this technology to real training events. Many lessons were learned as both technical staff, exercise developers, controllers and directors brought the skills they have employed for constructive simulation exercises to bear on exercise Winged Warrior. This section assesses the effectiveness of Steel Beasts (designed for gamers) against the training preferences described in Table 1. In this section, trainer preferences from Table 1 are presented in bold text, and gamer preferences are presented in bold italics.

The Learning Process and learning goals as applied to this exercise were identical to the processes that would have been employed had the exercise been supported through constructive simulation. There were, however, a few enhancements as a result of the entertainment value provided by the visual effects of the game. Trainees (players) found the out of the window view provided by the simulation to be realistic, interesting and captivating. The degree of immersion achieved was impressive as pilots (controlling flight with a keyboard) were observed leaning into their turns. The simulated enemy force allowed for the creation of realistic problems that affected the trainees on an emotional level that also lead to a perception of free play that included a degree of suspense and corresponding risks that gamers have come to appreciate and demand. In reality, the exercise controllers were, for the most part, in complete control. Maintaining a high degree of control might be even more important when using a game, because of the importance of keeping inputs realistic and to ensure training aims are being served.

Unlike constructive simulation exercises, the training audience interacts directly with the simulation as opposed to a command support system being stimulated by the simulation. As a result, there is less opportunity to make corrections or cover-up any flaws that may occur in the simulation during runtime. The visual effect generated is immediately available to the trainees and must be credible enough to maintain the validity of the exercise from the trainees' perspective. Creating a perception of Fantasy with no boundaries must be avoided as it will likely detract from the effectiveness of the training should the players come to doubt the degree of realism, and this could put the achievement of the training objectives at risk.

Ensuring a standardized and consistent visual representation during runtime became the single most important task for exercise controllers. This requires a high degree of coordination between all parts of the exercise control staff, technical controllers, enemy, fire support, other friendly players, etc. This turns out to be a far higher degree of coordination than would be required for the average exercise control cell using a constructive simulation. Exercise staff rehearsals before mission execution and before critical events were essential to ensuring that the correct visual effect was generated at the right time. Inadvertent pilot reactions that resulted early on in the exercise due to a number of visual bloopers were subsequently avoided as the result of the high degree of coordination. On the efficiency side, however, controllers quickly learned that if an activity did not contribute to the visual scene presented to the pilots, then it did not have to be simulated. After several false starts, recognition of this fact saved a considerable amount of effort. On the other hand, this also implies a great deal of knowledge in the application of the simulation. Very often, what has to be visually generated is not explicitly represented by the simulation and work arounds have to be found to create the correct visual result.

Steel Beasts, while having a good visual representation of a helicopter, does not claim to represent flight dynamics well. Indeed, during the exercise, "collisions" were turned off, so that helicopters could not crash into buildings, trees, mountains or each other. The pilots even used the keyboard and mouse to fly the helicopter, rather than a joystick. This low fidelity implementation was assessed as valid to meet the training aims of the exercise. The objective was not to teach a pilot how to fly a helicopter, rather this exercise was all about training a pilot to think about a tactical situation while a mission was unfolding. A more realistic cockpit simulation was certainly possible, however, had the pilots been presented with joysticks, they might have been more interested in the flight characteristics of the simulation rather than the tactical mission upon which they were required to focus. In this case, a lower fidelity flight model was deemed appropriate given the cognitive training objectives of the exercise.

In reviewing the training assessment details above, it is apparent that the trainers have incorporated many of the advantages of the game (8 out of the 12 gamer preferences) while at the same time being cautious not to include those that might detract from or add little value to the training. All of the trainer preferences with the exception of transfer of training are also addressed in the assessment above. This does not mean that the skills learned will not transfer to actual missions, merely that based upon the exercise alone, this is difficult to assess. As the alternative was not to conduct any exercise at all, any training transfers from this exercise would be better than no training transfer, assuming no negative training occurred. Furthermore, this issue was examined in some detail through one of the other training preferences not listed by Helsdingen in Table 1: After Action Review (AAR).

6 AFTER ACTION REVIEW

In addition to the standard debriefing emphasis of the AAR, students and staff were asked to assess the effectiveness of the exercise. Despite their admitted negative bias going into the exercise, students, staff and supporting aircrew all gave enthusiastic reviews on completion of the exercise. They claimed that the representation of the challenges in planning and executing tactical aviation missions was superior to the live versions of the exercise that participants had experienced in recent years. From the perspective of acceptance, the virtual version of Winged Warrior was rated as highly effective in meeting the exercise aims as evidenced by the decision to conduct the exercise in the same manner in the future.

There is of course room for improvement and two primary areas are being addressed for future iterations of the exercise. Despite the rich tactical environment provided by Steel Beasts including an active enemy with effective shoot-down capabilities, terrain more appropriate to operational deployments, etc., the simulation lacked several of the key decision support elements available in actual cockpits. These include electronic navigation information, communication systems, threat warning systems and countermeasures, door guns, a rear crewman station and the aircraft sensor package. None of these potential information inputs were included in the simulation. Furthermore, the simulation had a limited spectrum of visualization models for vehicles, human entities and cultural aspects of the environment. Airborne weapon systems and the range of ground-based weapon systems are limited and not always realistic in their effects. Improvement in both of these areas is planned for future iterations of the exercise.

7 SUMMARY AND CONCLUSION

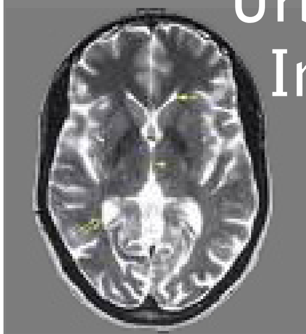
Although not the first application of a serious game for military training by the Canadian Army, Winged Warrior was the first time the professional training staff at DLSE employed a commercial off the shelf game for the conduct of a training event with a significant command and staff component. The lack of an appropriate simulation tool to meet the level 3-5 continuous scenario requirements was highlighted in the Training Needs Framework in Figure 2. The trainers proved adept at adapting the advantages afforded by the gamers preferences while ensuring the overall process was tailored to the aim, scope and training objectives of the exercise. Several opportunities for improvement have been identified but all concerned appear to agree that serious games are a welcome addition to the simulation supported training toolbox. The exercise also clearly demonstrated that exercise staff experienced with constructive simulation can easily adapt their skills to effectively meet training objectives that may be better served with gaming technology.

This paper was originally published in the proceedings of the Spring Simulation Multiconference as part of the Military Modelling and Simulation Symposium, Norfolk VA, March 2007. Reproduced with permission.

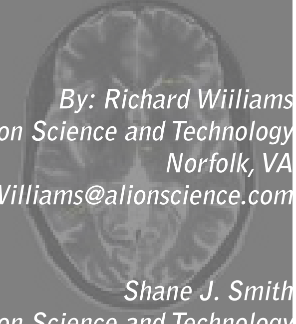
ABOUT THE AUTHORS

Paul Roman, CD, PhD, is an Associate Professor of Management Science and Operations Management at the Royal Military College of Canada and an Associate Professor at Queen's University School of Business. After serving 20 years in the Army as a Signals Officer, Paul became a professional engineer with a successful consulting practice applying his skills in business process re-engineering to projects ranging from the development of a combat development process for the Canadian Army to enhancing the relationship between operations and maintenance at Syncrude Canada Ltd. For the past several years, he has been assisting the Army develop and implement a Synthetic Environment Strategy helping to formulate policies and plans regarding the application of modelling and simulation as the primary enabling technology necessary to help the Army meet it's strategic objectives. In this capacity, Paul leads as the Army's Synthetic Environment Coordination Office that result in his interaction with the M&S organizations throughout the Department of National Defence and with Canada's major Allies. Dr. Roman has several publications to his credit and has chaired multiple simulation and training conferences in Singapore, the UK and Canada.

Doug Brown has been employed at the Directorate of Land Synthetic Environment since 1996. He has been responsible for the creation of the simulation support to hundreds of Canadian Army exercises. He has a Computer Science HBSc from the University of Western Ontario 1981. From 1981 to 1996 he was employed as a Defence Scientist in Military Operational Research, including a posting at the SHAPE Technical Centre.



Urban Resolve 2015: Technical Integration Lessons Learned



By: Richard Williams
Alion Science and Technology
Norfolk, VA
RLWilliams@alionscience.com

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

1. Introduction

In the world of software development, integration is often considered a formalized and strictly defined process. In 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 briefly describes the Urban Resolve 2015 (UR2015) simulations and their locations. It then discusses lessons learned in integrating the applications and locations from both a software and network perspective and examines both the successes and failures.

2. 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.

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 was 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.

3. 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.

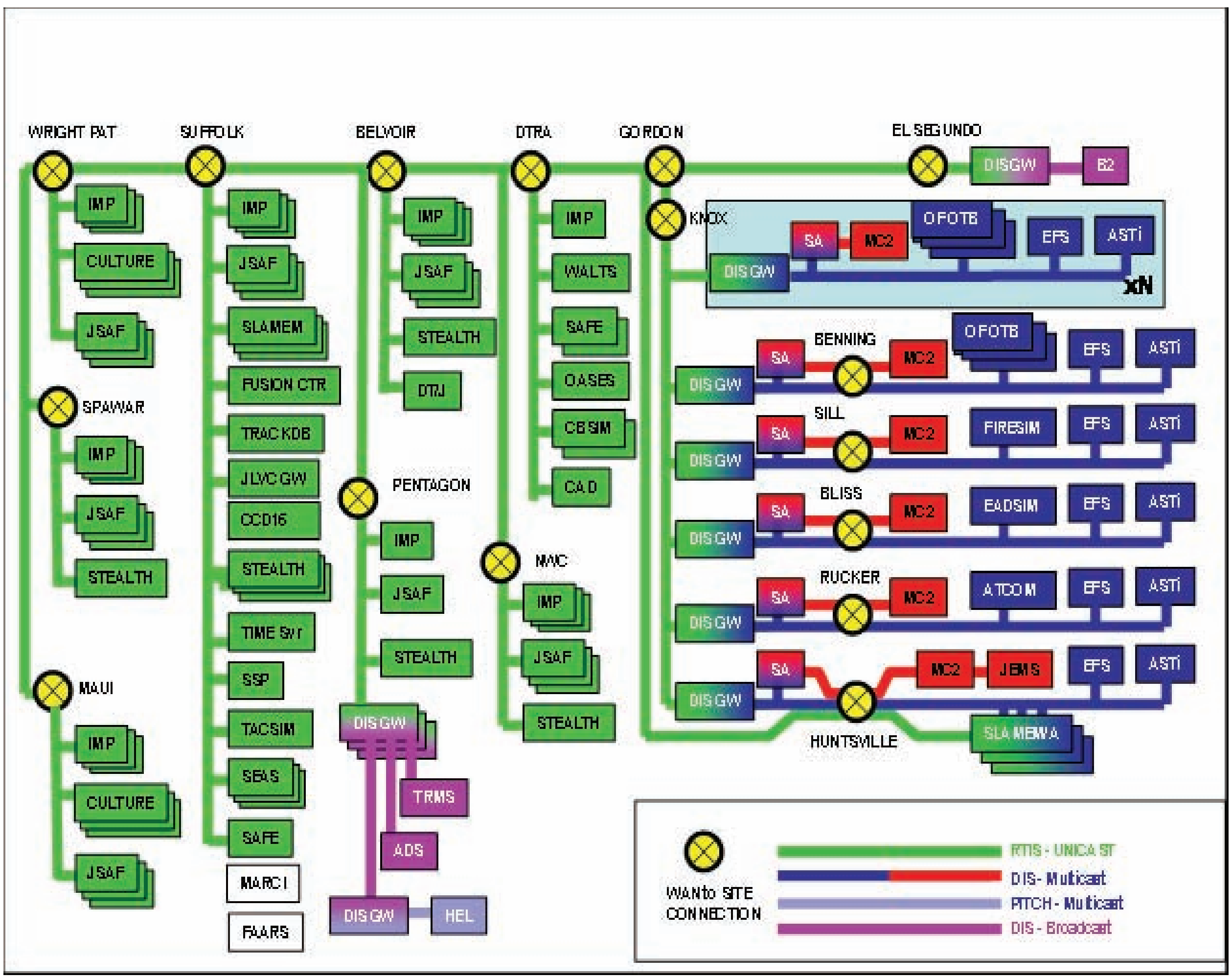


Figure 1: UR2015 Simulation Architecture

3.1 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 repeat-

PARTICIPANT	INTERFACE
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
OTBSAF	DIS
FIRESIM	DIS
EADSIM	DIS
ADS	DIS
ATL	DIS
TRMS	DIS
HEL/B2	DIS
CPOF	GCCS
GCCS	C4IGW
CCD16	C4IGW
C2PC	GCCS

Table 1. Participants in Urban Resolve Phase 3

ing process, crippling the entire federation and forcing a restart. Several solutions were investigated including implementing Data Distribution Management (DDM) 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.

3.2 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 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.

3.3 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. Figure 2 provides a high level view showing the HLA/DIS Gateway applications and the various DIS Networks to which they were connected.

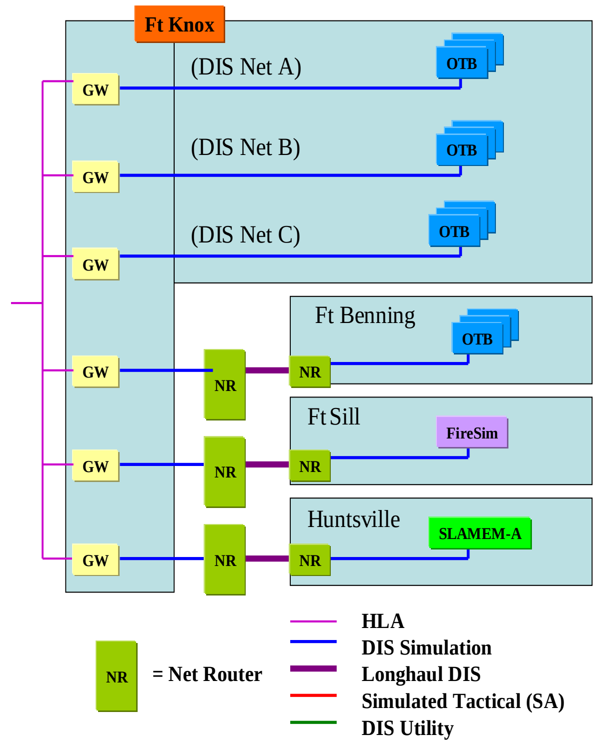


Figure 2: Army Simulation Connectivity

4.2 Data Distribution Management through Gateways

At the time of the UR2015 experiment, the simulations 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.

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 an entity state PDU hits its 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.

3.5 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.

3.6 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 & Speicher 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.

3.7 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.

4. 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 (OTH-Gold) reports and simulated Link-16 tracks were used for this purpose

4.1 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.

4.2 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.

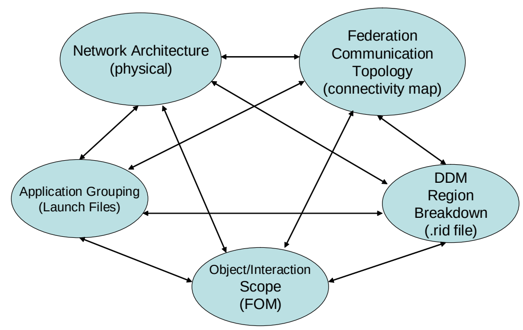


Figure 3: Simulation/Network Relationships

5. 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.

5.1 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.

Due to the heightened complexity (as seen in Figure 3) 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.

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 caused a sizeable spike in simulation network traffic which could possibly bring 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.

5.2 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".

5.3 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.

5.4 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 its impact and agreed to a maximum threshold.

5.5 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 & Torpey 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.

5.6 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.

6. 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.

7. 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, and 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.

8. 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.

The paper was adopted from I/ITSEC 2007 and is reprinted with permission.

9. References

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

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.

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

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.

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

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

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.

Bitwizard Webpage

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

DREN Webpage

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

GNU Webpage

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

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

HLA Federate Compliance Testing and Certification Program

By: Mark Crooks
Alion Science and Technology

INTRODUCTION

Modeling and Simulation (M&S) has become an integral part of the development process for many technological systems, ranging from defense applications to medical and scientific research and development. The High Level Architecture (HLA) is a general-purpose architecture for simulation reuse and interoperability. The HLA was developed under the leadership of the Department of Defense (DoD) to support interoperability and reuse across the large numbers of different types of simulations developed and maintained by the DoD. In addition to the basic HLA infrastructure, the DoD also saw a need to develop a suite of tools to support the implementation, reuse and interoperability of simulations adopting the HLA. One of these tools is the HLA Compliance Test suite, developed to test compliance to the HLA Standard.

HISTORY OF HLA AND COMPLIANCE TESTING

The HLA process is an important step towards present and future simulation interoperability within the DoD and private sectors. The United States DoD established the High Level Architecture (HLA) as a M&S interoperability standard in 1995. The original HLA Standards were titled DoD High-Level Architecture Standard Version 1.3 and were published in 1998[1-3]. In 2000 the Institute of Electrical & Electronics Engineers IEEE adopted revised HLA Standards in the 1516 series [4-6]. Currently these IEEE Standards are in the process of being updated.

Early on, it was clear that the application of the standard would be difficult without a dedicated methodology (a development and execution process), a set of associated supporting tools and an efficient compliance certification process. The DoD initiated the HLA Federate Compliance Testing process in 1997. Initial developmental work for the HLA Federate Compliance Test System was performed by Georgia Institute of Technology and Georgia Tech Research Institute [7]. Today, Johns Hopkins University, Applied Physics Laboratory (JHUAPL) is the HLA Compliance Test tool developer for the DoD [8].

As stated above, the original Federate Compliance Test Tool (FCTT) was fielded in 1997. These early tools served their purpose, but were difficult to use and maintain and could not keep pace with the new and varied ways the HLA specifications were being implemented. With the adoption of the HLA Standards by the IEEE, the FCTT was totally redesigned and could now test both of the HLA standards (1.3 and IEEE 1516). As of December 2007, the MSIAC has tested 289 federates for the DoD and other non-DoD organizations. (Although not covered in this paper, it should also be noted that under agreement with the NATO Modeling and Simulation Office, HLA Compliance Testing is now also available in France, Spain and Sweden).

As simulation technologies continue to evolve, it will become increasingly necessary to incorporate an interoperability component between various distributed simulation systems in order to conduct testing, improve functionality and even promote peripheral system development. HLA Compliance Testing provides an accepted set of tests under which a federate can achieve a level of compliance, based on its stated capabilities, and its ability to meet a defined set of standards. Such baseline standardization ensures that various simulation systems can communicate, interact with and access the capabilities of other simulations. Though this level of baseline interoperability may not completely satisfy all of the interoperability concerns of a federation manager, it does establish a level of assurance that:

- The federate produces and consumes data in accordance with its Object Model
- The federate manages itself consistent with it's stated capabilities
- The federate can call and receive call-backs from a verified Run-Time Infrastructure (RTI)

PROCESS TESTING COMPONENTS

HLA Compliance testing is conducted by an independent organization (MSIAC) and employs a standard set of test tools and Q&A sessions. Testing can be conducted over the Internet for unclassified simulations or onsite, for classified implementations. The HLA Federate Compliance Test System consists of two major components.

The first is the Federate Test Management System (FTMS). This is an HTML based web application that provides the interface into the compliance testing process for the Customer, the Federate under Test (FUT) and the Certification Agent (CA). Essentially, FTMS serves as an information-gathering database for the federate to be tested.

The second component of the test system is the FCTT. This JAVA application performs both pre-runtime and runtime functions. During the pre-runtime phase, the FCTT performs Simulation Object Model/Conformance Statement (SOM/CS) consistency checks and produces a file that either identifies errors in consistency or validates the consistency of the submitted files. One of the files that is submitted is the Federation Execution Document (FED) file. The FED file is a representation of a federate's SOM or Federation Object Model (FOM). The tool also performs a SOM/FED check that verifies that the FED file describes the same class hierarchy that is contained in the SOM. During runtime the FCTT is a federate that joins an established federation and monitors and records the FUT's activity through receipt of HLA Management Object Model (MOM) Report Service Invocation (RSI) interactions. After the FUT has demonstrated a sufficient subset of the capabilities documented in its SOM and invoked the services expected in its CS, the FCTT produces an RSI log and runtime results files which provide proof of compliance with the HLA standards.

In an effort to support the broadest segment of the DoD M&S community, current testing supports both versions 1.3 and IEEE 1516 of the HLA Specifications. A customer requesting HLA Compliance testing for a federate must submit a test application at the following website (http://hlatest.dod-msiac.org:8080/ftms/index.jsp). The testing steps are described below. The FUT must progress sequentially through the 6 steps of compliance testing to achieve certification. The Federate Compliance Testing Process serves as a mechanism to verify at runtime that the expected HLA services are being properly implemented and called in the correct sequences for compliance with the HLA specifications. An explanation of this testing process follows:

Step 1: Complete test application

1. A first time user of the HLA FTMS must register via a certification office web page to be able to submit federates for testing. The initial application requests the following information:

a. Name

b. Address to include country

c. Phone number to include country code

d. Fax number

e. Email address (becomes the user login)

f. Password (assigned by user)

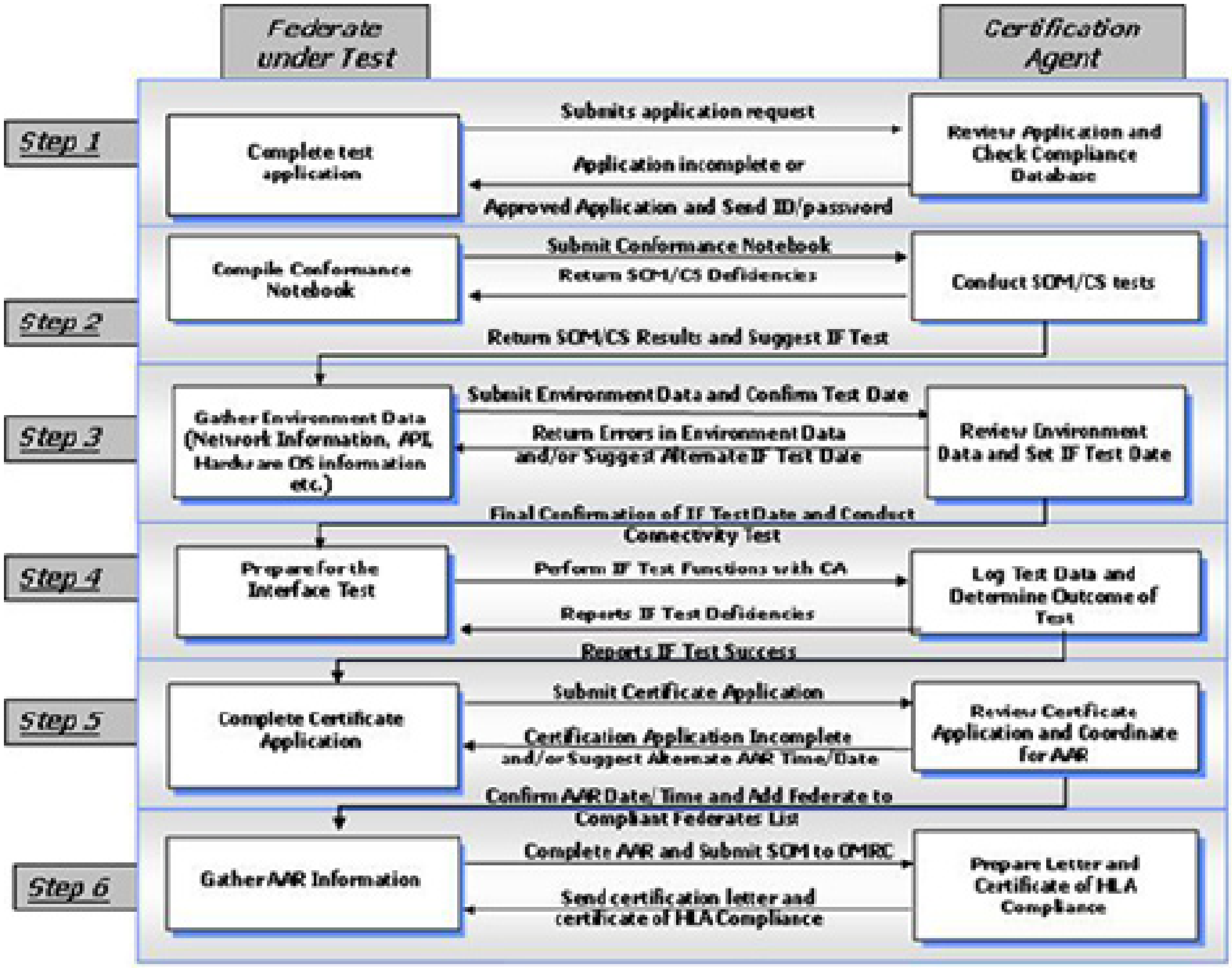
g. Language (currently English or French)

2. After the customer is approved as a new user, the details for a new federate to be tested need to be introduced.

3. Application for Testing is achieved via a certification office web page. Information needed to complete the application includes:

a. Point of Contact Information

HLA Federate Compliance Testing



THE PROCESS

b. Sponsorship Information

c. Contract Number

d. Federate Name, Version, and Brief Description

Step 2: Compile Conformance Notebook

1. The federate developer submits the following files via the web site for the FUT. These files are:

a. Simulation Object Model (SOM)

b. Conformance Statement (CS) document

c. Federation Execution Document (FED) File

2. The CA conducts three tests based on the SOM and CS. These are the CS Dependency/Quality Check, the SOM Parseability Test, and the SOM/CS Cross-Check. The Certification Agent will notify the federate developer that the FUT either passed the three tests or did not, and will show problems. Once the FUT successfully passes Step 2 the FUT owner is notified to proceed to Step 3.

Step 3: Gather Environment Data

.

In preparation for the IF test the following information is requested, i.e.

a. HLA Specification Version (US DoD1.3 or IEEE 1516)

b. Possibility to Test via the Internet (yes or no)

c. RTI Version (verified using the US M&SCO/DMSO RTI verification process, (https://www.dmso.mil/public/transition/hla/rti/statusboard)

d. RTI Configuration File

e. API name, Hardware, and Operating System used

f. RTI Execution hostname and Internet Protocol (IP) address

g. Federation Execution hostname and IP address

h. Runtime interface data (RID), Federation Planners Workbook (FPW), Object Model Template (OMT) and other files as needed

i. Whether or not a firewall is in place

j. Additional Comment Section

Step 4: Prepare for the Interface Test

1. The CA and the federate developer agree upon a test schedule. The IF Test requires the FUT to demonstrate every service and the SOM capability in a predetermined test sequence, which is designed to represent a subset of the complete capability of the FUT.

2. The Interface Test (I/F Test) has three parts:

a. The Nominal Test, which ensures that the FUT can invoke and respond to all services for which it is capable, according to the CS and

b. The Representative SOM (RepSOM) test, which ensures that the FUT is capable of invoking and responding to services using a range of data contained in its SOM.

c. The CA will log service data from the test, analyze the data, generate results, and return a Certification Summary Report (CSR) to the federate developer. The CSR is the official record of HLA compliance for the specific version of the federate code tested.

Step 5: Complete Certificate Application

Once the I/F test has been successfully completed the developer/test candidate and designated recipients will be notified that they have passed the certification process and that the federate is HLA Compliant. A federate that successfully completes the federate compliance test process receives a Certificate of HLA Compliance. The customer must make the request for a Certificate to the Certification Agent via the FTMS. All recipients have to be registered via the FTMS to receive the final certificate.

Step 6: Gather AAR Information

The final part of the certification process is the After Action Review (AAR) and paperwork to document the federate's certification of compliance with the HLA.

1. The CA provides a blank After Action Review form to the Customer.

2. The Customer and the CA coordinate when to conduct the AAR, for instance by completing the questionnaire during a phone conversation. The time that is required for compliance testing and certification will vary based upon the organization's progression through the steps of compliance testing, simulation knowledge, priority for testing, whether or not the federate is classified or unclassified and/or the time and resources necessary to overcome system integration/connectivity obstacles that may arise during the interface testing process.

Currently, there is no charge for compliance testing, but this could be conducted under a fee for service program in the future.

CONCLUSION

The HLA Compliance Testing and Certification process offers the capability to federate managers to reduce their interoperability risks. The process has successfully helped over 200 federate managers in the development of federates for an HLA federation. The evolution of the process and the tools continues to bring improvements. With multiple RTIs now available, the test tools have been continually updated to ensure their operation with all of them. Finally, the certification testing process continues to yield valuable information about federate developers' use patterns of HLA capabilities.

REFERENCES

[1] US Department of Defense, "High Level Architecture Rules Version 1.3," 20 April 1998

[2] US Department of Defense, "High Level Architecture Interface Specification Version 1.3," 20 April 1998

[3] US Department of Defense, "High Level Archi¬tecture Object Model Template Specification Version 1.3," 20 April 1998

[4] IEEE 1516-2000: "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules"

[5] IEEE 1516.1-2000: "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification"

[6] IEEE1516.2-2000: "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template (OMT) Speci¬fication"

[7] Simulation Interoperability Workshop, paper #00F-SIW-036, "Improvements to the HLA Fede¬rate Compliance Testing Process", by Margaret M. Horst, David Rosenbaum, Kyle A. Crawford, Georgia Tech Research Institute, September 2000

[8] Simulation Interoperability Workshop, paper #07F-SIW-075 "NATO and NATO/PfP Nations HLA Compliance Certification: History, Lessons Learned and Proposed Enhancements" by Jean-Louis Igarza, Randy Saunders, Roy Scrudder, Régis Mauget, Martin Adelantado, Mark Crooks, September 2007

AUTHOR INFORMATION

Mark Crooks is currently the Program Manager for the HLA Compliance Testing/HLA Helpdesk for the Modeling and Simulation Information Analysis Center (MSIAC). He has been in this position since 1999. He was a member of the Project Development Group for the original IEEE 1516 specifications. He also served as a member of both NATO MSG-011 and MSG-025. He received his BA from The Citadel in 1977 and a Masters from Troy State in 1987.

Acronyms

AAR	After Action Review
CA	Certificate Agent
CSR	Certification Summary Report
DoD	Department of Defense
FCTT	Federate Compliance Test Tool
FTMS	Federate Test Management System
FUT	Federate under Test
FED	Federation Execution Document
FOM	Federation Object Model
FPW	Federation Planners Workbook
HLA	High Level Architecture
IEEE	Institute of Electrical & Electronics Engineers
IF (I/F)	Interface
IP	Internet Protocol
JHUAPL	John Hopkins University Applied Phys ics Laboratory
MOM	Management Object Model
M&S	Modeling and Simulation
MSIAC	Modeling and Simulation Information Analysis Center
OMT	Object Model Template
RepSOM	Representative SOM
RSI	Report Service Invocation
RID	Runtime interface data
RTI	Runtime Interface
SOM/CS	Simulation Object Model/Conformance Statement