Beyond playing games: Federating game-based technology with large-scale simulation systems

Daniel Scolaro, Jennifer McNamara
BreakAway, Ltd.
Hunt Valley, MD
dscolaro@breakawayltd.com
jmcnamara@breakawayltd.com

John Little
Nova Technologies
Orlando, FL
little@novatechnologies.net

ABSTRACT

Game-based technologies have entered the traditional modeling and simulation (M&S) community to offer new, lighter-weight augmented or alternative solutions to military M&S problems. Increasingly, to fully leverage their potential contributions, game-based simulations are required to federate within the Live, Virtual, Constructive (LVC) paradigm by providing geographically separated instrumented operational equipment, human-in-the-loop simulators, and wargame simulations. These systems can include legacy M&S systems, vehicle and instrument simulators and even other game-based products. In this paper, we discuss the federation of game-based simulations via the High Level Architecture (HLA) within a large scale government M&S program. Specifically, we draw upon our experience federating a game-based simulation within the prototype for the Joint Terminal Control Training and Rehearsal System (JTC TRS).

This paper draws upon our team's experience providing the visualization framework for a lightweight, laptop-deployable, training solution for Joint Terminal Attack Control (JTAC) trainees. The resulting system satisfied the goal of integrating distinct systems, including existing Department of Defense simulations and custom components to support the JTC TRS requirements. The systems that were integrated represent varying levels of simulation fidelity and have a diverse range of functionality. The work done on this project has demonstrated that: 1) a game-based? technology simulation can act as a member of an HLA federation, 2) as a member of a federation game-based technology products can both visualize inputs from other COTS and GOTS solutions and generate outputs back to them and, 3) game-based simulations can provide an acceptable level of simulation fidelity for training while offering a deployable lightweight simulation. We draw upon direct experience to share lessons learned and make recommendations for successful integration of game-based simulations within the LVC environment.

ABOUT THE AUTHORS

Daniel Scolaro is a Senior Software Engineer at BreakAway, Ltd. Dan is responsible for overseeing all client-focused development activities including federating our game-based products with a variety of other simulations. Most recently, Dan has been the lead engineer on the JTC TRS program. He has over seventeen years experience in systems analysis and design, GUI and interactive simulation programming. He holds a B.S. in Information Systems from Drexel University and has formal DMSO training in HLA.

Jennifer McNamara is a Director at BreakAway, Ltd. In this role, she directly supports customers in creating serious games and simulations for training and experimentation using mōsbē™, BreakAway's simulation development platform. Working closely with organizations actively engaged in serious game development provides Jennifer with an insider's view of the needs and experiences of customers and the demands they place upon available simulation technologies. Jennifer holds a B.S. in Cognitive Psychology from Drexel University and a M.Ed. in Instructional Systems from The Pennsylvania State University.

John Little is a Vice President at Nova Technologies. John is a vision holder and architect for military training simulation programs. He holds a B.S. in Computer Science Applications from Arkansas State University and a M.S. in Information Systems Management from the University of Phoenix. John has over 35 years experience working in modeling and simulation based training systems, most recently leading our efforts on JTC TRS.

Beyond playing games: Federating game-based technology with large-scale simulation systems

Daniel Scolaro, Jennifer McNamara
BreakAway, Ltd.
Hunt Valley, MD
dscolaro@breakawayltd.com
jmcnamara@breakawayltd.com

John Little
Nova Technologies
Orlando, FL
little@novatechnologies.net

NEW APPLICATIONS FOR SERIOUS GAMES

The use of game technology for non-entertainment purposes has increased steadily over the past ten years and proven useful when applied to diverse market needs including: military training and simulation, corporate training, teaching complex political and social concepts, and even marketing consumer products. Early growth in the serious games market was fueled by consumer response to repurposing entertainment titles or leveraging game design to engage players to achieve other agendas/goals outside of the game environment.

Specifically, the military has turned to game companies with the intention of utilizing game technology to create stand-alone serious games and training simulations. This iterative process has met initial expectations with a reasonable degree of success as serious games have clearly found a place among traditional stand-alone, computer-based, distance, and simulation training solutions.

The use of serious games has evolved as military training, simulation, and experimentation customers have become savvier in their understanding of the contributions from the video game industry. As these customers have continued to hone their requirements through early experiences, it has become clear to game developers that the aspects of a great game, such as the suspension of disbelief, the pursuit of objectives, and the player engagement are not necessarily the most desirable features to be leveraged from games to meet military simulation needs.

The military needs flexibility and agility to develop and deploy simulations quickly, with fewer resources and at a lower cost than is customarily involved in building a traditional simulation solution. They want to select from pre-existing modeling and simulation (M&S) components, Semi-Automated Forces (SAFs), game technologies, distance learning components and modules, hardware and software simulators and actual operational systems to rapidly compose effective training and experimentation simulation solutions.

Tools for end-user customization, support for iterative development processes, and accelerated development timelines are all specialties of the games business. The industry has been creating end-user modification tools (known as “game mods”) for years, turning over aspects of game control to the players, who create their own experiences from the game’s assets. Given these adopted user trends and technical advances in the game industry, it seems a logical fit to apply the processes and practices from this industry to solve some of the military M&S community’s needs for flexible and agile development.

In order for video game technologies to move from stand-alone applications to play a role in integrated simulation solutions, they need to become participants in the iterative development process. Beyond this requirement, they must further be made capable of exchanging information following the standard techniques adopted throughout traditional M&S. In this paper, we describe one of the first major initiatives to take a game-based solution rooted in the game industry’s real-time strategy genre and incorporate it as a key system within an integrated M&S solution to prove the viability of the approach.

The JTAC training need

New innovations typically come along by recognizing shortcomings of systems currently in use. These shortcomings are typically not of a lack of effort but of effectiveness. Such was the case for a system needed to satisfy the training needs of joint terminal attack controllers (JTACs). The current training applications were proving ineffective at various levels even with many prior attempts to satisfy the need.

JTACs are defined as ground-based controllers that integrate US, Allied or Coalition air power with ground unit fire and maneuver by controlling close air support (CAS) against enemy targets during joint combat operations (Brown, 2005). The task is inherently joint as it touches individuals from multiple services requiring joint coordination of assets during execution. Further, CAS has become a standard aspect of operations in both the current and future anticipated battle spaces.

From the Joint Close Air Support (JCAS) Action Plan Memorandum of Agreement (MOA) 2004-01 (Joint Close Air Support Executive Steering Committee, 2004) the following statement defines the case for the JTC TRS need:

"Issue 2 of the 2003 JCAS Action Plan contains four actions designed to standardize training of JTACs throughout the Services, USSOCOM, and other DoD agencies. Completion of these actions will improve joint force interoperability and reduce the potential for mishaps and fratricides."

The Action Plan referred to in the JCAS MOA contains several specific activities which are expected to occur in support of the JTAC proficiency and organization. One of the items is a requirement to correct training deficiencies which resulted from each service or agency that employ JTAC's using independent and non-interoperable training systems.

In 2006 the Joint Terminal Control Training and Rehearsal System (JTC TRS) prototype program was created (hereafter referred to as "the program"), designed to replace five prior attempts to complete a training capability to support the multi-service, multi-faceted interoperable training requirements. Key to the success of any new development was an understanding of what caused the prior attempts to be measured as "ineffective" and reflection on the measures necessary to correct those shortcomings including the technology involved to provide the necessary structure.

Setting the stage for selection of a game-integrated solution

Due to the number of prior attempts made to serve the JTAC training needs and the existence of many different solutions across the DoD services, the program had an allowance to do initial research investigating the prior system developments, as well as completing an investigation as to the system operational issues within the user community. Significant effort was placed into a process which was designed to ferret out both the positive and negative aspects of the legacy systems, and with inputs from active users as well as developers, a profile emerged which reflected the shortcomings that would need to be overcome in any new development efforts.

The three aspects that emerged from the operational analysis, with the greatest need for emphasis from any new development, were:

- simplicity of user interface
- management of targets
- management of controlled assets

From the long term supportability and extensibility analysis, there were a number of issues that arose from the community, not the least of which was that there was no consolidated or common support plan in place or under consideration- for the future. The lack of a support plan resulted in a mixed configuration set with variable support mechanisms, most of which limited the overall long term expandability to new configurations to fit the evolution of the system, as anticipated. The focal points of any new solutions would be to:

- develop in a supportable, modern and extensible language
- provide an extensibility that would allow for future upgrades and interchangeable components
- develop a method of managing the interface to both the user and other interoperable simulations

Analysis was completed to see if any of the required system components were currently available as non-developmental items or from other obtainable government owned baselines. This analysis revealed that in past efforts little focus had been placed on creation of a common and quickly adoptable user interface mechanism that would be extensible enough to solve the training complexity demanded by the device. Along with this critical missing component, there was little to no allowance for the varying levels of users, both instructors and trainees, given in either the design of the user interface or core functionality of the system.

The team understood that providing the required level of involved solution which was both user friendly and adaptable to a variety of application and user requirements would be a significant challenge. Using simple analysis of the user community for the future systems, certain characteristics of the new interface management system were defined, however the overarching goal was to be as "user friendly as a modern video game". Given the lack of experience in the "normal" defense systems development groups for this type of interface, it was determined that a game developer would be a logical choice to provide assistance to address the design and development for this specification.

Requirements were gathered and initial functionalities were defined for development of a "framework" which would provide the infrastructure upon which the modularity and extensibility would be formed. The framework would also form the bridge between the game based components and the "typical" developed components from standard DoD production. There are well vetted processes and known methods for developing quality applications within the mainstream DoD training systems industry. However, it was decided that development of a system which could be considered extensible enough coupled with providing user adaptability would require a development group versed in both the art of the "walk-up-and-play" entertainment approach to interface development as well as the user functionality of the necessary high fidelity simulations.

From these points, the decision was made to contract a game company to develop a framework for applications which would allow the user interface as well as the - applications underlying them to be managed during execution, as well as allow for adaptations for alternate configurations.

The project challenged the development team to push the limits of the integrated component set and its underlying game-derived technology. The end solution needed to be scalable, to act as a member of a federation sharing entities and effects among multiple applications via HLA; be configurable to run on multiple user stations providing the appropriate end-user tools and perspective views; be interoperable by integrating into existing training modules; and be modular, by using existing COTS and GOTS software solutions.

In addition to user interface adaptability, user management of the virtual environment, specifically target placement and actions, was an issue in the initial analysis. Traditional simulation solutions for the military integrated elements are referred to as Semi-Automated Forces (SAF) systems. These systems have very complex levels of functionalities, and accompanying complex human interfaces. The selected game company had an existing military mission planning game-based simulation which provides the following system required aspects:

- drag-and-drop scenario editing
- distributed, multi-client system with centralized server
- 3d visualization with free-fly and attachable cameras
- library of 3d models
- entity dynamics with scriptable, semi-autonomous behaviors

The system also has a requirement for extended capability for recording and playback functionalities. These features all existed within the game-based environment and allowed selection of one application to serve the combination of several functions while providing a simple user interface for managing entities.

The training solution

Project requirements were to provide a lightweight, laptop-deployable, acceptable-fidelity training solution for trainees with the goal of integrating systems -- including existing DoD simulations and custom components to support the training requirements. Out of necessity this hardware restriction provided a significant challenge, requiring optimization and multi-tasking of some elements of the system,, differing greatly from standard paradigms of design.

To facilitate this the gaming technology toolkit was tasked to support multiple functions. The game was selected to provide a visualization virtual perspective for the Instructor function of the final system solution. Integrating the traditional SAF and Stealth functionality into a single application made hosting the system on a single laptop for the instructor operationally possible.

While the game selected had features directly applicable to the program needs, it had never been connected to or federated with an external system. Not only had the game never acted as part of a federation, but it had been designed and developed without any knowledge or anticipation of federating as a potentially desirable feature. For this reason, effort was necessary to enable the game to act as part of a federation. It was determined that this work was justified to enable the newer approach to serving the JCAS community training needs.

To successfully develop the overall integrated system, an incremental process was begun in early 2007 to first make the game High Level Architecture (HLA) compatible, and then secondly to create the definition of the adaptable framework to include both the game and user defined capabilities into a single refined application. Several iterations were made through the potential end state capabilities as well as an analysis of alternatives for user interface mechanization.

The Phase One goal was to build a solution that operates on laptops for both an instructor and a trainee. The instructor workstation provides overall simulation management and control as well as providing a platform for orchestrating integrated and interoperable exercises. The overall instructor station definition is depicted in Figure 1.

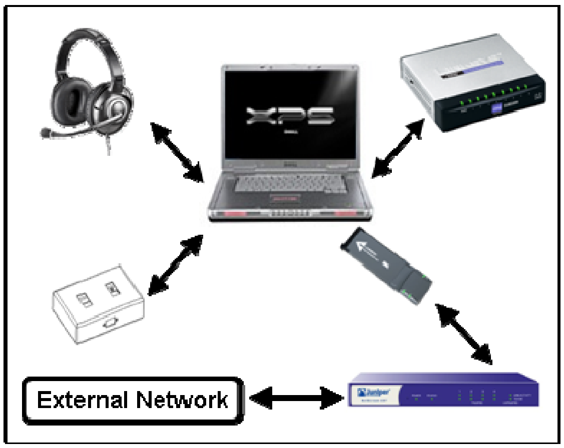

Figure 1 - The Instructor Station and Components

The trainee workstation provides the platform for a trainee to interact with both the virtual environment and simulations of equipment to support task training for equipment management. The trainee work station is depicted in Figure 2.

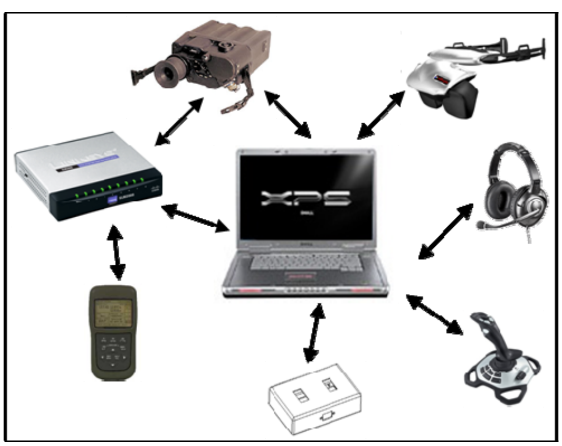

Figure 2 - The Trainee Station and Components

The framework of the user interface and the integrating architecture are deployed within both the instructor and student infrastructures. Each station is uniquely defined using common mechanisms and shared interface structures. The system components loaded on a given station are selected based on its configured role or "purpose" for the current exercise. Key mission functions employed in prior simulations were repackaged using the integration architecture provided by the game-based system.

MAKING A GAME HLA COMPATIBLE

In order to understand the significance of the work performed for federating via HLA, it is useful to understand a few high-level concepts and terms. The HLA defines a communication and simulation management framework that is easily adaptable to a variety of distributed simulation needs. The various simulation component applications ("federates") are organized into a Federation. The data exchanged in the federation is defined in the Federation Object Model (FOM). Additionally, many concepts and data standards drawn from the Distributed Interactive Simulation (DIS) protocol have been incorporated into HLA and many of the FOMs in use for military simulation. The HLA is implemented as the Runtime Infrastructure (RTI) by various vendors and government agencies. Many of the RTI products adhere to the Simulation Interoperability Standards Organization (SISO) Dynamic Link Compatible (DLC) Application Programming Interface (API) making it possible to choose the best RTI for a federation's needs without rebuilding the application.

For a more detailed understanding of these concepts we recommend reviewing:

- U.S. Department of Defense, High Level Architecture Interface Specification Version 1.3, Draft 11, 20 April 1998;
- IEEE 1516-2000 - Standard for Modeling and Simulation High Level Architecture - Framework and Rules;
- SISO-STD-004-2004 - Dynamic Link Compatible HLA API Standard for the HLA Interface Specification Version 1.3;
- SISO-STD-004.1-2004 - Dynamic Link Compatible HLA API Standard for the HLA Interface Specification (IEEE 1516.1 Version);
- IEEE 1278.1-1995 - Standard for Distributed Interactive Simulation - Application protocols; and
- IEEE 1278.1A-1998 - Standard for Distributed Interactive Simulation - Application protocols.

References to find these documents are provided in the references section of the paper.

The process we followed to make the game HLA compatible presented challenges that yielded valuable lessons learned for the federating of an existing game via HLA.

Federation

The client-server architecture of the game system allowed for the creation of an interface to the HLA federation in the server application without affecting the existing network interfaces between the server and the clients. This interface is implemented as a runtime selectable adapter layer that can be configured for different FOMs. The adapter layer handles translation between game system and FOM data.

The first step in creating the HLA interface for the game system was the identification of the key data which must be communicated to/from the federation. The data is divided into three categories:

- entity object management (create/update/ remove)
- munition effects interactions (firing/ detonation)
- simulation control interactions (pause/resume/ time)

The Real-time Platform Reference (RPR) FOM v2.0D17 was selected because of its clear representation of these concepts.

The prototype adapter layer was developed in two phases. The first phase concentrated on getting data from the external world into the game system and having the game entities react appropriately to it. This included visualization of the external entities and munitions effects as well as their detection by game entity sensor systems. In order to select the appropriate simulation and visualization models for the external data, a mapping system was created to match the incoming entity type (represented by DIS enumerations in the RPR FOM) with the closest matching equivalent in the game system. Requests to create, update or remove entities are passed from the adapter layer to the server application where they are translated into game system messages and forwarded to the clients.

In the second phase, the data for the game system entities was published to the federation. The same mapping system created for selecting game models to represent objects in the federation was used in reverse to select the appropriate DIS enumeration for the outgoing data. By the end of the second phase, the game system entities were able to interact fully (detect, react, engage, etc) with the other entities in the federation.

Record/Playback

The project issued a requirement to develop a capability to record HLA network traffic and support playback both post-exercise and during a live run (in-exercise Situational Awareness review). In the first phase the record/playback (RP) mechanism was embedded in the game system server application but this was deemed too restrictive. In the second phase, a standalone federate was created to encapsulate the RP functionality. While this arrangement was more flexible it did introduce several complications.

Since the RP mechanism was no longer an embedded component of the game system, a new, remote User Interface (UI) component was needed in order to control the chapter marking (record) and pause/resume and chapter selection (playback) functions. The RP function was required to co-exist with other applications that produce entities for use in the simulation. Playback with these federated applications executing would result in a duplicate set of entities being displayed in any visualization process. In order to facilitate pausing a live exercise for review, the FOM was extended to include interactions used to alert applications that they should temporarily retract (un-publish) their objects to avoid conflicts with those in the recorded data. Similarly, when the review is complete, the recorded objects are retracted and the federates are notified that they should re-publish their objects in the state they were in before the system was paused.

Challenges

Prior to this project, the game system did not interact with externally generated entities. All entities that would exist throughout the exercise were included in the scenario file and could be accounted for at load time. The project required that the game system be able to accept not only dynamically created external entities, but also the injection of new game system controlled entities based on user input through a “white cell” feature.

Incorporating externally controlled entities into the game-system posed a number of problems. The movement model that normally controlled animation of game entities was reworked to provide dead reckoning (DR) between object updates. The “AI” that controls entities needed to be suppressed to prevent unwanted behaviors such as external entities "running off' after perceived enemies.

The game system uses an effects-based combat system which pre-calculates hit probability and damage when the weapon is fired. It does not create an object representing the projectile (i.e. no flyout). Reconciling this system with that used by the other federates (notification of weapon firing, projectile object flyout and finally detonation) required creation of special projectile "vehicles" that "attack" their detonation location in order cause damage to game system objects and display the detonation animation. The damage model was modified to include partial damage. Also, the relative strengths of weapons and armor were adjusted to bring the game system entities and munitions in line with the other simulations in the federation.

Results

This effort resulted in successfully developing a flexible HLA interface for the game-based system. It was proven capable not only of participating in a federation with other, more traditional federates, but also of providing both a robust basis for the instructor interface and an advanced record/playback capability.

PROVIDING AN ADAPTABLE FRAMEWORK

The program needed an architecture that would allow for the management of different modules and components but also support training from a common doctrine. The simulations to be integrated represent varying levels of simulation fidelity and have a diverse range of functionality. The critical key to the success of the integrated application definition was the establishment of a common operating environment which could integrate the gaming platform, legacy traditional simulations and newly developed applications.

During the course of the system design process, a common operating framework was developed for allowing the integration of components, both legacy and new, to co-exist in a common operating space. To facilitate this, a set of "rules of engagement" were defined that provided a definition of both data and management processes necessary to facilitate the integration. Developed during this process was the concept of a "purpose" where a collected and grouped set of applications were collaborated onto a single hardware platform. The oversight method and environment in which a purpose operates needed to be developed.

Given that this environment would be tightly integrated with the overall human interface, and that human interface is one of the primary challenges in existing systems, the task of developing the overarching management process and the structure of the human interface was assigned to the same game group that had developed the federated gaming system.

The rules of the interface

Several considerations were made during the development process given that the final configuration for the user interface would be a collaboration of legacy and newly developed applications. To facilitate this "normalization" of functionality, several rules were defined for the development or alterations required to support the interface. The parameters of the rules were as follows:

- User Interfaces and core application functionality would be segregated. To allow for clear lines of control as well as extendibility of the working applications for functions, the UI and Applications were not to be imbedded within the same applications executable.
- Loadable Libraries: The segregation of the UI from the application supported another rule for the framework implementation. The use of Dynamic Linked Libraries (DLL's) which could be independently managed and defined for integration allowed the configurability of a purpose.
- Common Data Reference Methods: To allow for normalization of interactions between application components, standards were provided to the developers that defined the overall protocol for the integration. There were two key architectural methods employed: a) Synchronized Port Definitions (for transmission and reception point to point) and b) XML structured messages being passed through user defined communication messaging. Within an application, any methods which would facilitate best communication (shared memory, port integration or other methods) were allowable. Using these standards allowed for common interaction methods which simplified the overall development challenge.
- Common Application Management: This process required several elements to be defined, not the least of which was a standard method for starting and stopping applications. There was also a requirement for the development of a set of services which

allowed integration of specific functions (such as location requests) into the overall data structure.

The human interface

To allow for the facilitation and use of the applications infrastructure, a standard display method and segmentation of the user display was completed leveraging the overall game human interface techniques, in use and proven effective by the game industry. Utilizing colors, common locations and methods as well as simplicity in architecture, the display construct completed the overall framework definitions. The final configuration for the user interface of the Instructor Station definitions are reflected in Figure 3.

The overall elements of the display management screens are independent from the underlying applications they control. The allowance for clearly demarked interfaces gave the flexibility of managing the interfaces separately from the management applications underlying each functionality, giving the ability to have varying control systems for common functions across purposes. An example of this would be for Mission Management, where one application would be used for execution, with only those controls and displays necessary to support that station brought forward to the user. This would allow the instructor and the student to both have a capability, but allowing the instructor to have possibly more control or visibility that would not be allowed for a student.

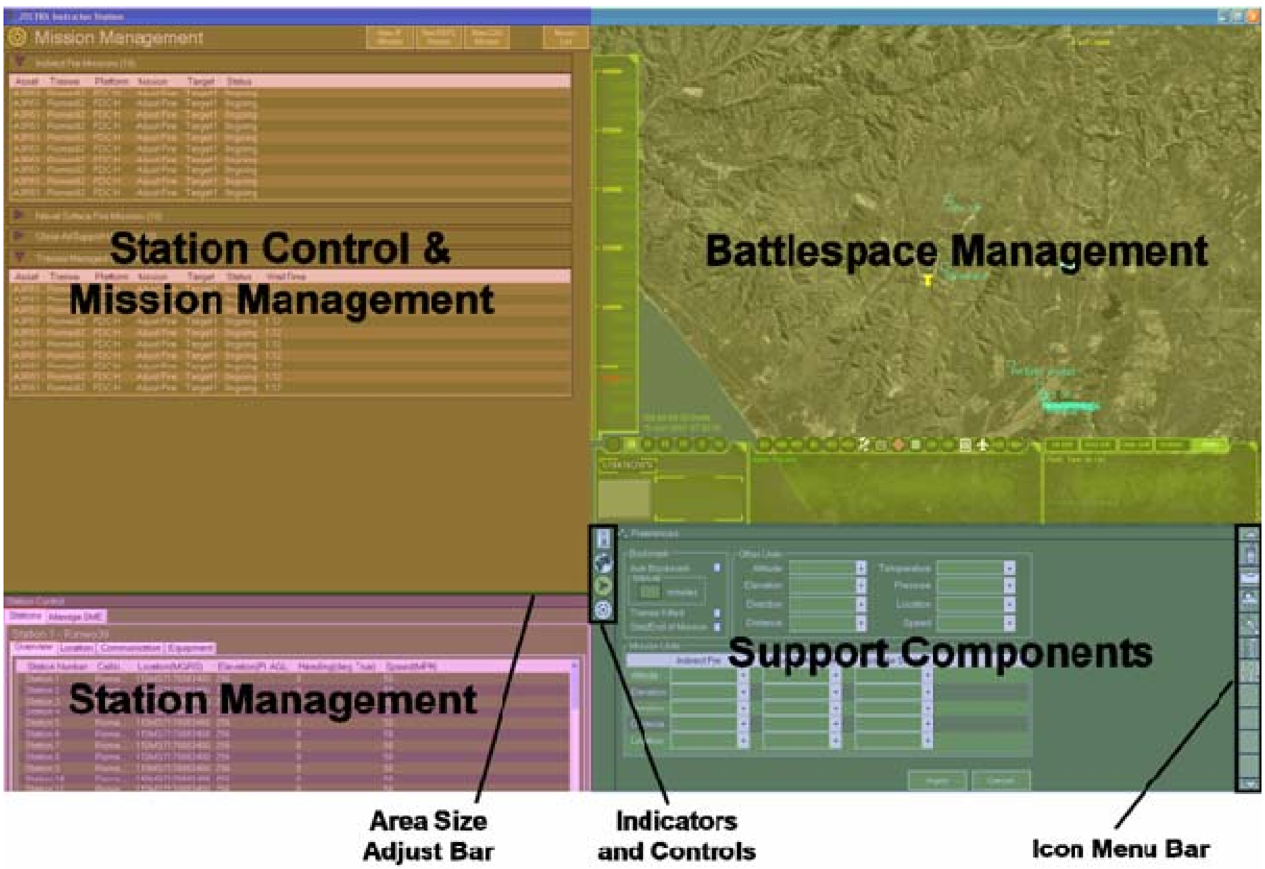

Figure 3 - User Interface Structure Example

THE ROLE OF THE GAME

The key requirement for the games application to this development effort was to make displays and controls of the game correlate with other system components by federating via HLA and exposing external programmatic control of the application.

Using the developed HLA interface, the game was able to represent entities and events which are being defined and controlled by the game system and provide reactions to all accepted data from other systems at the same level of fidelity as an event native to the game. The internal record/playback capability of the game

was also used to capture events and interactions communicated via HLA from the federation.
Instead of switching between multiple stations during an exercise, the game's visualization and control capabilities provide one common station for an "instructor-in-the-loop" exercise. For this project, the game provides white cell and scenario editing capabilities allowing the instructor or manager of the simulation exercise to create the training scenario, monitor the simulation as it evolves, and add and delete entities to force training or experimentation requirements. The game was able to replace multiple simulation programs eliminating the need for an instructor to rely on multiple work stations by incorporating all necessary functionality into one application.

The game generates the entities that populate the trainee station – in the program the visualization of these entities for the trainee is handled by an external image generator (IG). The IG is visualizing all entities in federation for the trainee. The game is visualizing all the entities for the instructor while also generating entities, and controlling them, in the federation. To assure database correlation a database format was chosen which was consumable by both visualization systems. In support of the effort, the development team also modified the game so that it can be embedded in other applications and systems, and be launched and controlled by external systems. This allows information to pass back and forth programmatically between the external applications. The external application can run or shut down the game, tell the game what scenarios to run, ask for data from the game and send messages into the game. This capability can be easily extended to include access to additional capabilities.

CONCLUSION

The development effort undertaken by the team demonstrated that the flexibility and modularity of game-based technology can provide a high fidelity, low overhead simulation environment. The work done on this project has produced significant contributions for the M&S community by demonstrating:

- that a game-technology based simulation can act as a member of an HLA federation
- that, as a member of a federation, game-based technology products can both visualize inputs from other COTS and GOTS solutions and generate outputs back to them
- that game-based simulations can provide an acceptable level of simulation fidelity for training while offering a deployable lightweight simulation

The solution created as part of the development project has applicability for any integrated training project. As demonstrated in this program, the game industry's approach to exposing high levels of end-user simulation control can revolutionize simulation development. In this way, a game can fit into a larger training picture enabling operational leaders, mission planners, instructors and scenario designers to share a common operating picture with strategic commanders and individual trainees without building custom simulations and tools at each level in the training process.

ACKNOWLEDGEMENTS

The JTC TRS prototype effort was a designated program defined by JFCOM, managed by PEO-STRI (PM/STS) and led by Rockwell Collins, STS. We thank the program leadership for believing that a game-based solution could meet the program needs.

REFERENCES

Brown, D. R. (2005) JTAC: MOA vs. JTTP. *FA Journal Jan-Feb 2005*. Retrieved, June 07, 2008 from http://findarticles.com/p/articles/mi_m0IAU/is_1_10/ai_n13717112

Drake, S., Granowetter, L., Möller, B., Wuerfel (Eds.) Simulation Interoperability Standards Organization Dynamic Link Compatible HLA API Product Development Group. (2004). SISO-STD-004-2004 - Dynamic Link Compatible HLA API Standard for the HLA Interface Specification Version 1.3. Orlando, FL: Simulation Interoperability Standards Organization, Inc.

Granowetter, L., Möller, B., Wuerfel (Eds.) Simulation Interoperability Standards Organization Dynamic Link Compatible HLA API Product Development Group. (2004). SISO-STD-004.1-2004 - Dynamic Link Compatible HLA API Standard for the HLA Interface Specification (IEEE 1516.1 Version). Orlando, FL: Simulation Interoperability Standards Organization, Inc.

IEEE 1278.1-1995 - Standard for Distributed Interactive Simulation - Application protocols (1996). New York, NY: Standards Committee on Interactive Simulation (SCIS) of the IEEE Computer Society.

IEEE 1278.1A-1998 - Standard for Distributed Interactive Simulation - Application protocols (1998). New York, NY: Standards Committee on Interactive Simulation (SCIS) of the IEEE Computer Society.

IEEE 1516-2000 - Standard for Modeling and Simulation High Level Architecture - Framework and Rules (2000). New York, NY: Standards Committee on Interactive Simulation (SCIS) of the IEEE Computer Society.

Joint Close Air Support Executive Steering Committee. (2004, 9 1). Joint Close Air Support (JCAS) Action Plan Memorandum of Agreement 2004-01. Retrieved from: http://www.ewtgpac.navy.mil/newt/TACAIR%20WEBPAGE%20FILES/03%20MOAs/JTAC/JTAC_MOA_1Sep04_Final.pdf.

Shanks, G. (Ed.). (2003). Real-time Platform Reference Federation Object Model (RPR FOM) Version 2.0D17 10 September 2003. AMS Donibristle, UK.

Shanks, G. Simulation Interoperability Standards Organization DIS Enumerations (2006). Enumeration and Bit Encoded Values for use with Protocols for Distributed Interactive Simulation Applications SISO-REF-010-2006 12th May, 2006. Orlando, FL: Simulation Interoperability Standards Organization, Inc.

Simulation Interoperability Standards Organization Standards Development Group. (1998). U.S. Department of Defense High Level Architecture Interface Specification Version 1.3, Draft 11, 20 April 1998. New York, NY: Institute of Electrical and Electronics Engineers, Inc.