

Developing the “Right” User Interface for Analyzing Human Behaviors and Soft Factors Across Multiple User Domains

Lowell Baker, Rodney Figaroa

Product Manager OneSAF

Program Executive Office for Simulation, Training, and Instrumentation

3045 Technology Parkway

Orlando, FL 32826

USA

Lowell.Baker@us.army.mil / Rodney.Figaroa@us.army.mil

TRAINING AND ANALYSIS OF HUMAN BEHAVIORS AND SOFT FACTORS USING CONSTRUCTIVE SIMULATION WITH REAL TIME STRATEGY GAMING TECHNOLOGY

Constructive simulations are used more today than ever to research, analyze, and train Warfighters using higher fidelity models of human behaviors and soft factors. The integration of these complex behaviors across the various United States (U.S.) Army Modeling and Simulation (M&S) domains creates an unusually cumbersome user interface for these constructive simulations. The One Semi-Automated Forces (OneSAF) is a composable, next generation computer generated forces (CGF) that can represent a full range of operations, systems, and control processes with a variable levels of fidelity that supports all U.S. Army M&S domains. Incorporating all of the needed functionality from these various domains into a single human graphical interface resulted in a “one size fits no-one” paradigm. An effort was conducted by the program to find a more optimal and intuitive way of displaying information to the user. An initial evaluation concluded that Real Time Strategy gaming interfaces addressed the key concern of an intuitive structure that could be easily used by the operators of the various domains with limited training. This investigation led to the development of an alternative user interface for the OneSAF simulation engine, called Ares. This report will address the current technology; human factors design considerations, operation, and future development of the Ares Graphical User Interface (GUI).

1.0 INTRODUCTION

The United States is engaged in a long war against terror which has changed how the U.S. Army M&S domains simulate real world environments that represent the cultural and behavioral problems that Warfighters encounter on a daily basis. Constructive simulations in particular, have evolved from representing conventional force-on-force conflicts, to an environment in which cultural and behavioral aspects of individuals often produce the vital information on which a commander can base their crucial decisions against. The advancement in simulation research and technology has allowed us to build behavior and cultural representations into complex scenarios that are being used today to train our Warfighters with the knowledge, skills, and experience to handle everything from humanitarian assistance to counter insurgency operations. Also, more than ever, the U.S. Army is relying on cost effective solutions to training that can provide an enhanced training experience, as well as reduce duplication and lower the total life cycle cost of multiple simulations that provide a redundancy in training value.

OneSAF is the US Army's next generation, entity-level constructive simulation that is answering the M&S training need of providing a complex environment capable of modeling human behaviors across all of the M&S domains, thus reducing the amount of constructive simulations needed to be sustained throughout their lifetime. One of the challenges of having a single entity level constructive simulation supporting so many various users, is providing a user interface which can provide the needed functionality in a way in

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE OCT 2010	2. REPORT TYPE N/A	3. DATES COVERED -			
4. TITLE AND SUBTITLE Developing the Right User Interface for Analyzing Human Behaviors and Soft Factors Across Multiple User Domains		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) OneSAF Program Executive Office for Simulation, Training, and Instrumentation 3045 Technology Parkway Orlando, FL 32826 USA		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 See also ADA564696. Human Modelling for Military Application (Applications militaires de la modelisation humaine). RTO-MP-HFM-202					
14. ABSTRACT Constructive simulations are used more today than ever to research, analyze, and train Warfighters using higher fidelity models of human behaviors and soft factors. The integration of these complex behaviors across the various United States (U.S.) Army Modeling and Simulation (M&S) domains creates an unusually cumbersome user interface for these constructive simulations. The One Semi-Automated Forces (OneSAF) is a composable, next generation computer generated forces (CGF) that can represent a full range of operations, systems, and control processes with a variable levels of fidelity that supports all U.S. Army M&S domains. Incorporating all of the needed functionality from these various domains into a single human graphical interface resulted in a one size fits no-one paradigm. An effort was conducted by the program to find a more optimal and intuitive way of displaying information to the user. An initial evaluation concluded that Real Time Strategy gaming interfaces addressed the key concern of an intuitive structure that could be easily used by the operators of the various domains with limited training. This investigation led to the development of an alternative user interface for the OneSAF simulation engine, called Ares. This report will address the current technology; human factors design considerations, operation, and future development of the Ares Graphical User Interface (GUI).					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR	18. NUMBER OF PAGES 10	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

which it is easy to interpret, understand, and use. This is no easy task, due to the diversity of the OneSAF user community. The US Army M&S domain users range from the operational to the engineering/analysis sides of the spectrum, and an interface which fits one community perfectly, will more than likely make another group of users throw their hands up in frustration. To understand this challenge, it is important to understand the OneSAF Program, and to know the missions of the organizations that make up the US Army's M&S domains.

2.0 THE ONESAF ENTITLY LEVEL SIMULATION

OneSAF has a very unique mission as compared to most acquisitions undertaken by the Department of Defense (DoD). Usually, an official acquisition is the result of an identified capability deficiency that cannot be filled by current systems; therefore the DoD develops and deploys this capability to the Warfighters. In the case of OneSAF, the mission need statement aimed at reducing the US Army's life cycle M&S costs. Various studies in the area showed that developing a single entity level constructive simulation to be used by the entire M&S community would drastically reduce total life cycle ownership costs. The programs' requirements were to replace the legacy entity based simulations that were in use. OneSAF was officially released as version 1.0 on Sept 29, 2006 with subsequent releases on an annual basis.

OneSAF is a government owned, open-source, composable simulation toolkit that includes computer-based simulation pre-exercise, exercise, exercise execution, and post-exercise tools. It has the capability of modeling up to a Brigade (BDE), and contains semi-automated behaviors for the echelons of Company (Co) and below. Additionally, a full complement of entities (individual combatants, tanks, rotary and fixed-wings aircraft, etc), semi-automated units (teams, squads, platoons, and companies) and behaviors are included along with tools allowing the user to develop their own unique entities, units and provides the ability to modify existing behaviors. OneSAF gives an operator the ability to run an exercise from the initial planning and scenario generation, thru execution, and to the final After Action Review (AAR) of the exercise. One of the strengths of OneSAF is its ability to model urban operations that focus on the Contemporary Operating Environment (COE), and its ability to model complex human behaviors and soft factors.

OneSAF Capability enhancements are continuously being developed and integrated into the OneSAF baseline which results in yearly version releases. OneSAF has been fielded to multiple US Army users within the three U.S. Army M&S domains as well as other DoD agencies, along with industry, Foreign Military Sales (FMS), and academic locations.

3.0 M&S DOMAINS THAT UTILIZE ONESAF

3.1 Advanced Concepts and Requirements (ACR) Domain

The ACR domain uses OneSAF to conduct experiments that investigate new concepts and advanced technologies. The products of these experiments help to develop operational requirements in doctrine, training, leader development, force design, materiel and soldiers which will better prepare the Army for future operations. The ACR also uses OneSAF to evaluate the impact of horizontal technology integration through simulation and experimentation. To the ACR domain, a user interface would need to allow them to easily change data tables associated with existing entities, add new entities and units, execute multiple runs faster than real time, and collect the data from the simulation in support their essential elements of analysis.

3.2 Research, Development and Acquisition (RDA) Domain

The RDA domain uses OneSAF to investigate the design, development, and acquisition of weapons systems and equipment, conduct basic research, and evaluate system prototypes. The M&S in the RDA domain is used for scientific inquiry to discover or revise facts and theories of phenomena, followed by transformation of these discoveries into physical representations. The RDA domain needs a user interface which would be flexible enough to allow the augmentation of data parameters, implement attributes of high fidelity models, seed certain values to eliminate variables, and collect data that supports their theories.

3.3 Training, Exercises and Military Operations (TEMO) Domain

The TEMO domain uses OneSAF in training events that includes individual, collective, combined arms, joint, and/or combined forces exercises. The mission of the TEMO domain includes rehearsals and evaluations of all phases of war plans. The analysis conducted during the rehearsals or evaluation validates the plan as best as the simulation environment will allow. The TEMO domain needs a user interface that is intuitive for its operators. Many of the interface capabilities required by the RDA and ACR domains are rarely needed for the TEMO operators. The interface needs to be easy to train, resemble the Army Doctrine which the soldiers are familiar with, and be “hardened” to the point where an operator cannot interfere with the stability of the system by accessing a function which does not pertain to their role.

4.0 ONESAF INTERFACES

Until the release of version 3.0, the OneSAF baseline contained only one user interface, the Management Control Tool (MCT). Even though the interface was reviewed by Human Factor experts in the past (10 years ago to be exact), it was becoming difficult to present all of OneSAF’s capabilities onto one single usable user interface. In many cases, it seemed as if a software developer needed to add some new capability to the user interface, so they created a radio button or a drop down box by means similar to throwing a dart at a dartboard. While certain users had no trouble at all with using the MCT to obtain their desired outcome, others were frustrated with things such as the amount of steps that were involved in implementing behaviors, or the amount of fields that required user input to task a unit to move in a particular formation. The need was quickly realized for an alternative interface, or a way to compartmentize the user interface that provided operators with the proper hooks to implement an interface that would better suit their needs.

4.1 Management Control Tool (MCT)

The MCT is OneSAF's primary tool for creating and executing a simulation scenario. Those familiar with constructive simulations will find its appearance to be very familiar. It uses Sun Microsystems' Java Foundation Classes Swing, which is an application programming interface (API) for providing a GUI for Java programs. The interface provides the basic scenario planning functions which enables an operator to create, edit, and run a simulation scenario. It also provided controls that allow an operator to initialize, run, and stop the simulation. The MCT (Figure 1) is made up of four distinct windows that allow you to build task organizations, apply control measures, script a scenario with assigned behaviors and tasks, and run the simulation.

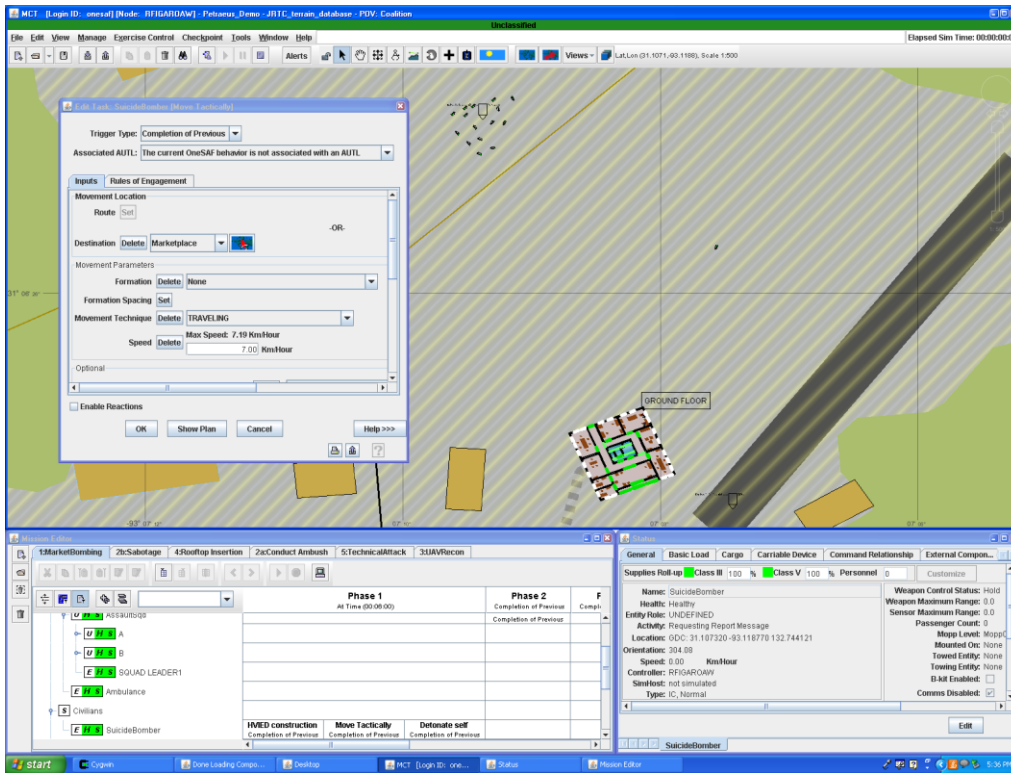


Figure 1: OneSAF MCT.

4.1.1 Plan View Display (PVD) Window

The PVD is a classic 2 Dimensional view of the terrain. It provides a visualization of entities, terrain features, graphic control measures, and the status of entities. Many operators would claim that this is the single most important component in the user interface where they want to instantly visualize their forces, objects, and updated status. A classic complaint by operators about the PVD is the size. Even though half of the screen is utilized by the PVD by default, operators still want a larger, clutter free terrain box.

4.1.2 Mission Editor Window

The Mission Editor Window is an execution matrix, where the units in the Task Organization window are assigned tasks to perform. This allows an operator to script the execution of behaviors on a per entity or unit basis. Behaviors execute at specified times or upon the triggering of certain events (such as the crossing of a certain phase line graphical control measure). For the ACR and RDA domains, this is an essential capability that allows them to configure multiple, scripted repetitions of a particular scenario. The TEMO domain, however, prefers not to use scripting, but instead manually control the actions associated with the entities and units.

4.1.3 Status Window

The Status Window is used to determine scenario outcomes and change properties of units and objects on the map. Depending on the entity in which the status is acquired, this window can return so many attributes, that don't even fit on the default form. For the ACR and RDA domains, this may be needed in order for them to analyze and collect data on certain parameters. For other operators, this is viewed as entirely too much information, because they only care about a few of the parameters, such as health and fuel.

4.1.4 Task Organization Window

The Task Organization Window is used to add and delete entities and units to a simulation. It automatically defaults to two sides: Coalition and Insurgents, but it can be modified to accommodate as many sides as needed. As in many other constructive simulations, this is shown in a hierarchical tree. While this allows for ease in selecting entities and units for small scenarios, it can become cumbersome when the scenario contains several thousand entities. While certain users may find the MCT useful, others find the interface very inadequate. Many operators do not want to be presented with the buttons, menu options, and windows whose capabilities do not pertain to them. The number of steps required to execute behaviors is always a point of contempt between the diverse users. The ACR and RDA communities need the level of behavior input provided in the MCT to properly control all the parameters in their experimentation and system evaluation runs. The TEMO operators do not really care for that level of control and prefer an “intervention” like way to control the battle space, with minimal user input. Another request that has been raised by operators is the length of time required to train an operator on the MCT. Currently, operator training for the MCT lasts approximately two weeks and leaves the operator somewhat confident on the workings of the tool with all the myriad of functions and widgets presented to them. Typically the lead time for an event does not allow ample time to train operators, hence the need for an intuitive interface.

4.2 EVOLUTION OF THE NEW INTERFACE: ARES

Prior to the release of OneSAF version 3.0, the government and development teams were hard at work sifting through user feedback, defining and implementing the interface’s new look and feel. The vision for the new interface was to provide users an interface that:

- They can easily visualize and understand.
- Provides enough information that will allow the operator to make a timely decision on what action he/she wants to take.
- Provide the ability to easily execute those actions as intuitively as possible.

The team came up with three different implementation options which primarily focused on the 2D aspects of the new interface:

- Use the existing GUI framework – This option was faster to implement as the expertise of this working technology resided in-house and would give the operator the accustomed look and feel. Some of the major drawbacks of this option were that it did not offer any performance improvements, did not provide major improvements to the map display and the effort would be bounded by the inherent limitations of the Java Swing API. The time needed to re-factor could be as much as the time needed to start from scratch, if not more.
- Create a new interface with a flexible 2D/3D engine – This option allowed the development team to build an architecturally better product that allowed for future growth. This approach broke away from the current OneSAF look and feel and allowed the development team to gain control over every display aspect on the screen. There were also some possibilities of gaining performance enhancement due to the offloading of processing onto the Graphical Processing Unit (GPU). Of course this approach was risky and presented the team with some drawbacks. Topping the list was the lack of in-house expertise in game interface development coupled with uncertainties associated with the stability and functionality provided by the engine.
- Create a new interface using a Web Browser – This option allowed for complete Operating System (OS) platform independence and extended the use of the new interface over the Internet/Wide Area Network (WAN). An issue with this approach was the real estate requirements for the basic Web browser itself, which takes a good bit of space. This threatened to

put the new interface into the same screen real estate issues associated with the MCT. Other drawbacks for this option included possible performance issues associated with large scale exercises, latency associated with the message relay time between the Web servers and each GUI instance, and security implication that may arise due to use of web services to support exercises of different classifications.

The team also compiled a set of desired capabilities which were folded in with the analysis of alternatives. Through this process it was determined that the second option provided the interface solution the team was looking for. Many of the features attributed to Real Time Strategy (RTS) games could provide a more intuitive interface, limit the number of unwanted controls on the MCT, provide an easier way to visualize 3D display, and greatly reduce the training time needed for the MCT. The results of this investigation into RTS interfaces produced the initial design concept for Ares [1][2][3]. Figure 2 shows the basic screen layout that came out of the analysis.

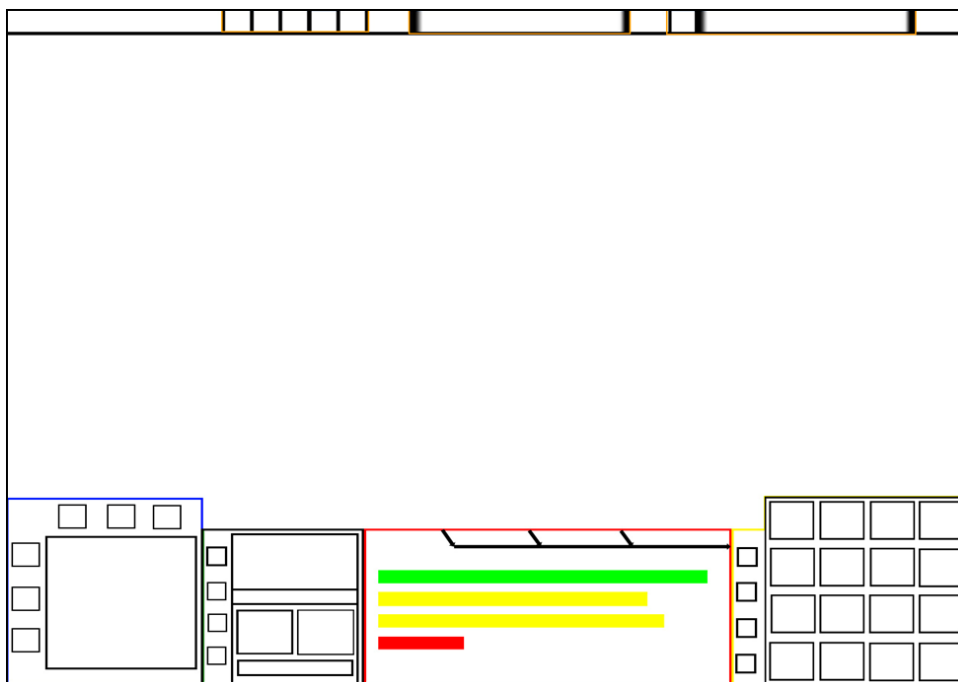


Figure 2: Ares HUD Mockup.

Ares provides the operator with a game-like interface providing full control over entities and a 3D view of the battlespace. Many of the younger Warfighters are familiar with RTS interfaces from such games as Starcraft, Command and Conquer, and Age of Empires III to name a few. Ares was designed to provide the operator with a small, streamlined set of controls on screen at all times while maximizing the 3D view of the battlespace (terrain, features, buildings, and building interiors).

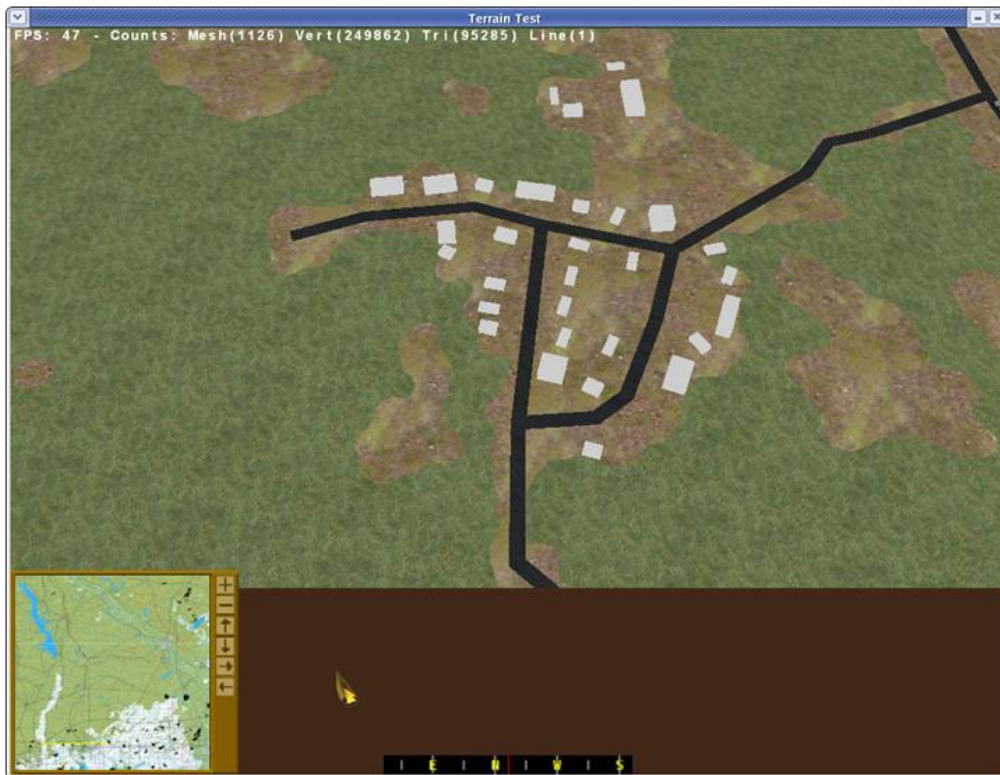


Figure 3: First working Ares prototype.

Figure 3 shows the initial working prototype, which took approximately 4 weeks to complete. This prototype only showed a very limited representation of the feature rich terrain database. One criteria for this interface was the ability to ingest the OneSAF native Objective Terrain Format (OTF). This was essential to eliminate the need for doing any terrain conversions between native OneSAF interfaces and, perhaps the more important benefit, the drive to eliminate virtually all terrain correlation issues. This would also lessen fair fight issues between a top-down and a virtual interface. The early prototype also gave some early insights into the battlespace navigation as well as insight into the battlespace object control aspects of the interface. The early success of the Ares prototype was the result of choosing the right 3D rendering engine. Ares was developed using the java Monkey Engine (jME) a high-performance scene graph based graphics API. In order to develop a user interface which utilizes full 3D, and to keep with current best practices and standards, the GUI was designed using OpenGL standard. Access to OpenGL was provided by the Light Weight Java Game Library (LWJGL), an open source Java software library, provided by jME and commonly used by game developers. Both OneSAF and jME were each other's complement as both promoted open source software, standards and have the ability to be platform independent. Figure 4 shows the current stage of the Ares.



Figure 4: Ares.

The following are a few of the Ares features:

4.2.1 3D Battlespace

Ares provides a larger view of the battlespace than the original PVD. While the details of the 3D graphics are less than that of an RTS computer game, it provides a visual that is easier to interpret than a 2D map with contour lines. The operator can control their view of the battlespace by a number of means, including pointing to the edge of the screen, using the arrow keys, and their altitude by using the page up/page down keys.

4.2.2 Minimap

As in most RTS games, Ares also provides a minimap panel in the lower left hand corner that displays a 2D view overview of the entire terrain database. This allows for the operator to quickly navigate to any position in the terrain box. It also shows the operator which direction he is currently viewing the battlespace in relation to his orientation of the terrain box. The minimap also provides zoom in/out features for the operator to adjust his overall view.

4.2.3 Create Control Measures

Three buttons located above the minimap are allocated to the creation of control measures. The three buttons represent creating points, lines, or areas, which are displayed on the battlespace. Once these control measures are in place, the operator can task the units under his control with behaviors that reason off the control measures boundaries.

4.2.4 Entity or Unit Information Panel

This panel displays the selected entity's name, location, and entity/unit icon. There is also an option in the panel that allows an operator to attach their battlespace view to the perspective of the entity by using the

camera attach/detach buttons. Again, rendering the battlespace in 3D allows for a capability such as attaching a camera view to an entity that the original MCT was not capable of providing.

4.2.5 Tasks Panel

In this panel, all of the orderable tasks associated with the currently selected entity or unit is displayed. These icons allow for the Ares interface to simplify the number of clicks needed to execute a desired behavior. For instance, a move tactically task is now a matter of clicking on the “move tactically” icon, then selecting a point of the 3D battlespace in which you want the entity to move. The numerous fields that allow for the customization of the task, (such as speed, formation, quickest route, orientation, enable reactive behavior, etc.) which is present in the MCT, has been removed and these fields have been set at predetermined default values for current entity/unit. This allows operators to have more manual control over the entities and units under their control, and is the best alternative to an operator who is not interested in scripting his execution, such as the capability provided in the MCT's Mission Editor Window. If the operator feels the need to adjust these predetermined default values and apply a different set to the current behavior they can do so by accessing a sub menu.

4.2.6 Weapon Control Status Panel

Located next to the Tasks Panel, is the Weapon Control Status Panel. This allows an operator to quickly select an entity or unit, and either view or change weapon control status to allow/prohibit the engagement of targets at target locations.

5.0 ARES FUTURE ENHANCEMENTS

Feedback from the user community regarding the Ares interface has been very positive in nature, and many improvements have been proposed for future enhancements. Even though Ares is considered as a prototype in its current state, several co-developers are actively working on new capabilities. A few of these enhancements include:

- Updating the Ares 3D visual model library, as well as improving the articulations and animation of these models.
- The capability to create specific types of point, line and area control measures properties (Assembly Area, No Fly Zones, Critical Friendly Firing Zones, etc.) and 2525B/C symbology.
- Individual Ultra High-Resolution Building (UHRB) component selection.
- Complete all OneSAF current and future behaviors (To the point where the process of adding a new behavior is the same as currently on the MCT).
- A behavior queue that acts similar to the Mission Editor Window.
- Ability to use the Ares tool to conduct mission rehearsal in theatre.
- Increased stability of the Ares interface.

6.0 CONCLUSION

This paper addressed the human factors design considerations, operation, and development of the Ares GUI, and explains how a simulation can implement an user interface which easily falls into the “one size fits no-one” conundrum. When designing the architecture of a constructive simulation, there are some steps that must be taken to ensure the user interface is flexible enough to adapt to the needs of various different user communities. These include:



- Placing a high prioritization of the user interface during the initial architectural design phase of the project.
- Employ human factors and user interface experts early in the design phase and schedule enough time to thoroughly test the user interface to ensure it meets the user’s needs.
- Ensure the proper hooks are documented and developed into the simulation engine that would allow an operator to use additional user interfaces as required to meet their operational goals.

Implementing a Real Time Strategy gaming interface in OneSAF resulted in a much more usable interface to many of our operators, but this may not be a reliable design decision for every simulation. Ideally, the simulation’s architecture would be designed to allow a variety of different types of front ends to interoperate with the simulation engine, essentially componentizing the user interface. It is also important to understand that the interface considerations need to be implemented early during the design phase, because any change in libraries, programming languages, or operating systems can have a very large and expensive impact on the overall system. Hopefully, this OneSAF case study gives the community a good example of how to enhance the usefulness of a user interface when multiple user domains are involved.

7.0 REFERENCES

- [1] Lopez-Couto, S., Figaroa, R. “An Analysis of Game Design to Improve Complex User Interfaces in Constructive Simulations. 2008.
- [2] Dwyer, T. “Usability Study for OneSAF Graphical User Interface”. PM OneSAF Internal Development Document. 2008
- [3] OneSAF. “PM OneSAF – MMI- Decision Brief Phase 1”. PM OneSAF Internal Development Document. 28 January 2008.