Common Gunnery Architecture - Issues in Developing a New Software Product Line by Extending an Existing Software Product Line

Henry Marshall	Dean Runzel Andrew Richards	Darren Law	Gary Green
US Army Research, Development and Engineering Command, Simulation and Training Technology Center	US Army Program Executive Office for Simulation, Training and Instrumentation, Program Manager for Combined Arms Tactical Trainers	Science Applications International Corporation	Institute for Simulation and Training (IST), University of Central Florida (UCF)
Orlando, FL	Orlando, FL	Orlando, FL	Orlando, FL
Henry.A.Marshall@us.army.mil	Dean.Runzel@peostri.army.mil Andrew.B.Richards@peostri.army.mil	Darren.R.Law@saic.com	ggreen@ist.ucf.edu

ABSTRACT

The Common Gunnery Architecture (CGA) is a new software product line being developed by the Project Manager Combined Arms Tactical Trainers (PM CATT) at the Program Executive Office for Simulation, Training and Instrumentation (PEO STRI). The CGA objective is to reduce duplication between gunnery training systems in part by maximizing the advantage of common components such as the OneSAF Objective System (OOS). Using requirements derived from different gunnery training systems as a baseline, the CGA team analyzed the OOS Product Line Architecture Framework (PLAF) to determine OOS extensions required to support precision gunnery training. This analysis determined that the OOS PLAF would need to be extended or modified to include unique gunnery training components such as crewstation interfaces, image generators (IGs), curriculum and student management, simulation controllers and physical and behavioral models. By extending the OOS PLAF, CGA could heavily leverage the existing OOS infrastructure investment. As the CGA analysis progressed, the Research Development and Engineering Command (RDECOM) Simulation and Training Technology Center (STTC) was developing an embedded simulation demonstration using OOS to drive both the constructive simulation and virtual ownship simulation. This was closely related to the direction of CGA which encouraged the teams to leverage their efforts. As part of the risk mitigation for the CGA product line development, the CGA team partnered with the STTC to develop a functional prototype of a gunnery training system to verify assumptions concerning the use of OOS as the building block for the CGA product line. This paper provides an overview of the CGA program and identifies issues and lessons learned during the gunnery prototype development. Examples of the many issues include simulation engine latency, bi-directional interfaces and high fidelity models.

ABOUT THE AUTHORS

Henry Marshall is a Principal Investigator for Embedded Simulation Technology at the Research, Development and Engineering Command (RDECOM) Simulation and Training Technology Center (STTC) where he manages the Embedded Combined Arms Team Training and Mission Rehearsal Army Technology Objective (ECATT-MR ATO). Prior to this assignment he worked at the Simulation, Training and Instrumentation Command (STRICOM) where he spent 11 years as lead for the computer generated forces/semi-automated forces (CGF/SAF), High Level Architecture (HLA) and Linux porting developments on the Close Combat Tactical Trainer (CCTT) system in addition to being a OneSAF team member. He received a BSE in Electrical Engineering and an MS in Systems Simulation from the University of Central Florida.

Dean Runzel is a software engineer at the Program Executive Office for Simulation, Training and Instrumentation (PEO STRI). He currently supports software development for training devices for the Program Manager for Ground Combat Tactical Training including the Common Gunnery Architecture. During his career, he has supported software development for several program managers at the PEO covering the entire spectrum of live, virtual and constructive training devices. He has received a BS in Engineering from the University of Central Florida and a MS in Software Engineering from Monmouth University. He has been certified as a Software Architecture Professional by the Software Engineering Institute at Carnegie-Mellon University.

Darren Law is a Chief Scientist at Science Applications International Corporation (SAIC). He currently supports the Embedded Combined Arms Team Training and Mission Rehearsal Army Technology Objective, Common Driver Trainer, Common Gunnery Architecture, and Synthetic Environment Core (SE-Core) programs. His prior experience at SAIC includes performance-oriented work on the OneSAF Objective System and Close Combat Tactical Trainer (CCTT). He received baccalaureate degrees in Computer Science and Math from the University of Florida and master's degrees in Computer Science and Modeling & Simulation from the University of Central Florida, where he is a doctoral candidate in Computer Science.

Andrew Richards is a student of Computer Science at the University of Central Florida, expecting to graduate with a Bachelor's Degree in December of 2005. He is currently working in a co-op position with Program Manager for Ground Combat Tactical Trainers at PEO STRI through the college. He has worked there part-time while attending classes since May of 2002. His efforts while employed by the PEO include Information Assurance work in support of the accreditation of several training systems developed by the Program Manager for Combined Arms Tactical Training (PM CATT). He has worked closely with the CGA program since its inception and has been responsible for scenario generation and testing efforts on the CGA testbed.

Gary Green is a research associate at the Institute for Simulation and Training (IST), University of Central Florida. He has over 25 years of experience in management and research of training and simulation programs. His recent experience includes eight years managing research projects exploring embedded simulation and embedded training issues in support of Army research and development. He is currently the principal investigator for IST's Embedded Simulation Technology work with US Army RDECOM STTC. His MS in Operations Research is from the US Naval Postgraduate School.

Common Gunnery Architecture - Issues in Developing a New Software Product Line by Extending an Existing Software Product Line

Henry Marshall	Dean Runzel Andrew Richards	Darren Law	Gary Green
US Army Research, Development and Engineering Command, Simulation and Training Technology Center	US Army Program Executive Office for Simulation, Training and Instrumentation, Program Manager for Combined Arms Tactical Trainers	Science Applications International Corporation	Institute for Simulation and Training (IST), University of Central Florida (UCF)
Orlando, FL	Orlando, FL	Orlando, FL	Orlando, FL
Henry.A.Marshall@us.army.mil	Dean.Runzel@peostri.army.mil Andrew.B.Richards@peostri.army.mil	Darren.R.Law@saic.com	ggreen@ist.ucf.edu

THE CASE FOR A COMMON GUNNERY PRODUCT LINE ARCHITECTURE

The Army inventory includes numerous precision gunnery training systems that have similar functions. However, since most of these systems are based on "stove piped" proprietary designs there is little or no commonality that can be shared in life cycle maintenance or future system development. To overcome this lack of a common component framework for building gunnery trainers, the Program Manager for Combined Arms Tactical Trainers (PM CATT) at the Program Executive Office for Simulation, Training and Instrumentation (PEO STRI) began developing a Common Gunnery Architecture (CGA) Product Line Architecture. The goals of the CGA can be seen in Figure 1. The Army's gunnery training systems currently use four separate software architectures and are now encountering a cost benefit wall while trying to maintain concurrency with the weapon systems and adapting to the Contemporary Operating Environment (COE). A cost benefit analysis shows a clear advantage for CGA over the high maintenance costs of the current architectures. CGA also offers the potential for lower costs in fielding future systems and in support of new COEs such as urban terrains. Other advantages include shared upgrades and common user/instructor interfaces.

The Program Manager for Ground Combat Tactical Trainers (PM GCTT) gunnery training device team researched potential solutions to overcome the duplication of costs and development efforts in gunnery trainers. Two candidate solutions were identified as potential building blocks for CGA. The first was the emerging OneSAF Objective System (OOS) with its requirements and architectural processes, composable architecture, enhanced synthetic natural environment, models and target behaviors, and network services that would support C4I adaptors and gateways. (PM OneSAF, undated) The second was the Synthetic Environments (SE) Core, which will support future virtual domain designs. The SE Core program is likely to provide substantial opportunities for reuse of virtual simulation at various points in any product line, but is in a very early stage of development. For CGA development, the team chose to extend the OOS constructive simulation software product line for use in the virtual simulation domain. To minimize duplication of effort and promote further cost efficiencies, the CGA team will continuously engage the SE Core program.

The first step for the CGA team was to develop a Product Line Requirements Specification (PLRS) covering all the gunnery training systems identified for possible integration of CGA (Figure 1). Requirements documents were collected for each of the software baselines and merged into a consolidated requirements list. Based on this consolidated requirements list, the CGA team drafted a Product Line Architecture Framework (PLAF) with emphasis on leveraging existing OOS components as much as possible. Figure 2 shows the current OOS PLAF with components that may be modified to support CGA identified with their top right corner folded. In some cases, particularly for models and environmental areas, these modifications could be extensive. Some of the requirements identified by CGA were clearly beyond the planned capability of OOS. These are shown in Figure 3 as additional components that would be required in the CGA PLAF to support unique gunner training requirements.

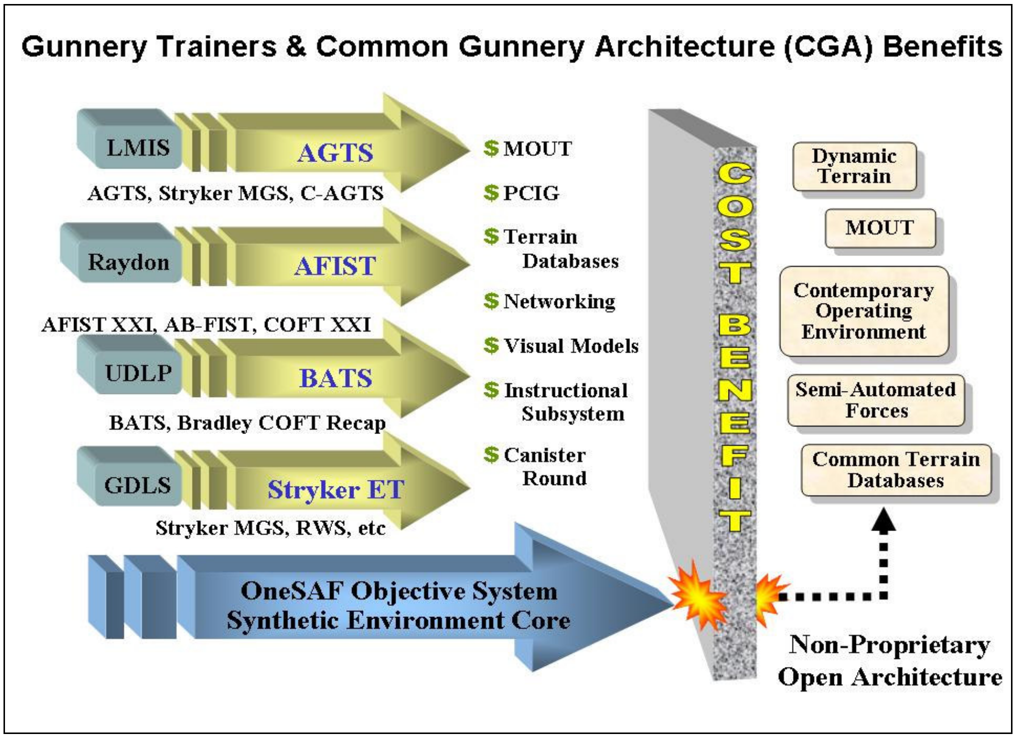

Figure 1: CGA Gunnery Systems and Goals

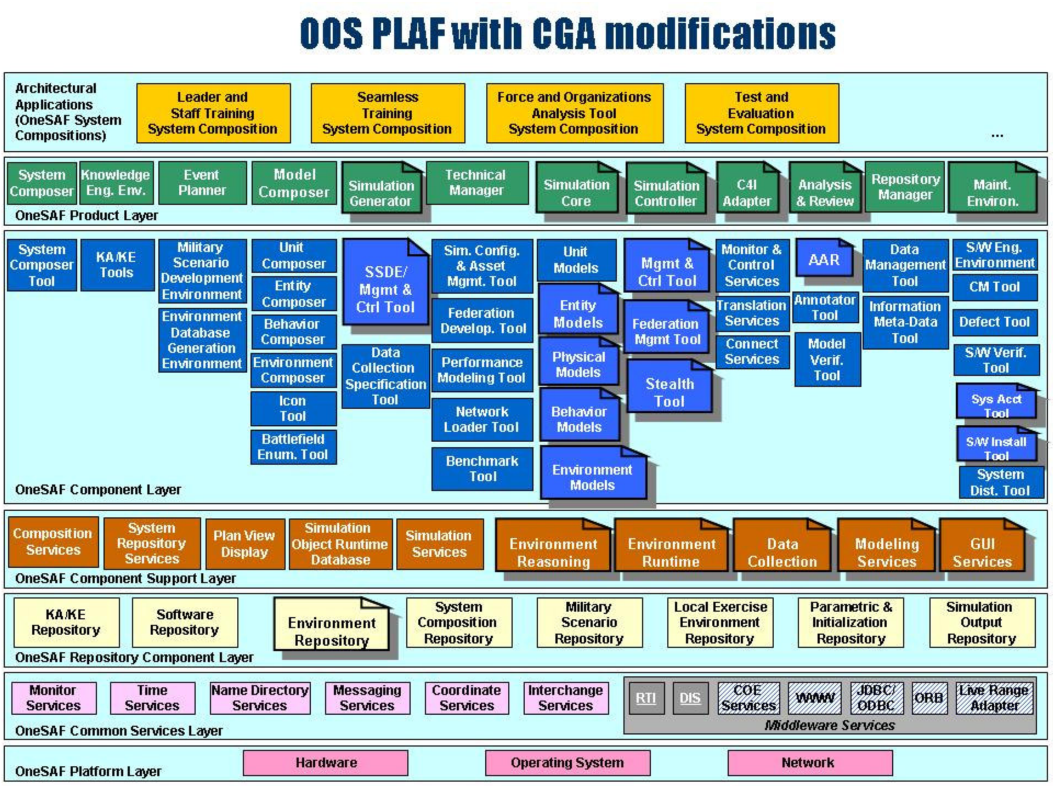

Figure 2: OOS Product Line Architecture Framework - Components Requiring Modification to Support CGA Have a Folded Corner

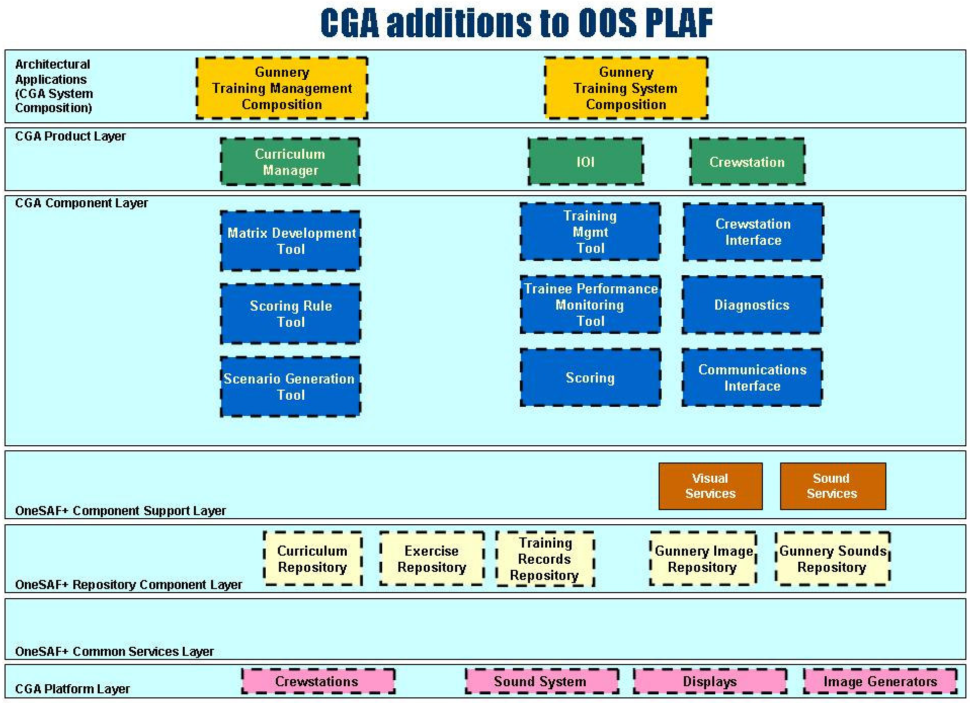

Figure 3: CGA Additions to the OOS PLAF

These components generally relate to the gunnery crew station, the instructor/operator station and training management and were not in the original OOS scope.

It is worth noting that it was never envisioned that CGA would be part of OOS. Rather, OOS was to be a building block for CGA. This is consistent with other developments such as the Close Combat Tactical Trainer that share numerous components between manned simulators and Computer Generated Forces (CGF) systems.

The CGA PLRS and PLAF were developed from paper reviews of existing gunnery training systems and Common Components such as OOS that could be used as building blocks for the CGA product line. The missing factor was an actual demonstration of a gunnery training system using OOS and supporting human-in-the-loop (HITL) virtual simulation components. Typically issues and requirements that appear similar or trivial under paper comparisons are revealed to have numerous issues during the actual system design. This evaluation process and its results are the focus on the remaining parts of this paper.

LEVERAGING A RELATED RDECOM DEVELOPMENT - ONESIM

As the CGA program was beginning its initial analysis, the Research Development and Engineering Command (RDECOM) Simulation and Training Technology Center (STTC) was performing research on possible solutions for a new embedded simulation engine for the Future Combat System (FCS) program. (SAIC, 2005) Known as OneSIM, this research was carried out under the Embedded Combined Arms Team Training and Mission Rehearsal (ECATT-MR) Army Technology Objective (ATO). The focus of this research was to develop an embedded simulation engine that could reside in the limited hardware footprint expected in FCS manned vehicles. (Marshall et. al., 2004). After an evaluation of alternatives the ECATT-MR ATO manager elected to investigate a single host simulation approach based on Semi Automated Forces (SAF). The single host approach used the Stryker specification for embedded training, which at the time had the embedded training system operating on a limited single host footprint. This required that major simulation tasks such as image generation and SAF processes reside on the same processor.

To virtually simulate the ownship, the SAF was modified so that a SAF entity could accept control signals from a human using a joystick rather than the usual high level control functions from the SAF. In this design the operator could take control of a SAF entity and direct its movement in the simulated environment using a joystick attached to the SAF computer. The initial prototype used the OneSAF Testbed (OTB). This research was demonstrated to the FCS Lead Systems Integrator and Program Manager (PM) OneSAF. PM OneSAF suggested that rather than continuing with OTB, the second phase of the development move to OOS which was presently in development. It was also suggested that the joystick be replaced by a more relevant control interface such as a crewstation for an FCS vehicle.

PHASE 2 OF ONESIM

To accommodate a more relevant control interface in OneSIM the ECATT-MR ATO developed an FCS-like Infantry Carrier Vehicle (ICV) crewstation simulator. The basic architecture is based on an A-Kit/B-Kit concept used by the embedded simulation software of the Tank Automotive Research Development and Engineering Center (TARDEC) for their Crew integration and Automation Testbed Advanced Technology Demonstration. In this design the A-Kit typically includes controls that are unique to a vehicle (in this case the ICV simulator) and the B-Kit includes common embedded simulation items such as the SAF and image generator. The two kits communicate using a well defined interface consisting of a set of A-Kit/B-kit messages.

After reviewing available architecture and middleware alternatives for use with OneSIM based on OOS, the ATO manager decided to use the Distributed Interactive Simulation (DIS) protocol, the A-Kit/B-Kit framework for the FCS ICV simulator and the Simulation Object Runtime Database (SORD). The preferred alternative, the FCS System of Systems Common Operational Environment (SOSCOE), was very immature at the time and the team was concerned that integration issues caused by this immaturity would distract from the basic research of building an embedded simulation engine.

To support the HITL inputs coming from the A-Kit in the ICV simulator, the ECATT-MR team worked with the FCS OOS team in developing some common simulation components for OOS. The basic components to support a HITL simulation prototype can be seen in Figure 4. One of the strengths of OOS is that it allows developers to create “compositions” of OOS components that include or exclude OOS components as needed. Some of the components that were developed for the HITL application include:

- An input/output (I/O) manager that routes the HITL data to the SORD and to OOS actors such as mobility and weapons as a way to manage the agents that could affect a given entity
- A “set data” PDU middleware component that is used to send control positions and states to the I/O Manager
- A Graphical User Interface (GUI) external control that allows an operator to cause an entity to stop accepting constructive control from the SAF and begin accepting HITL control.

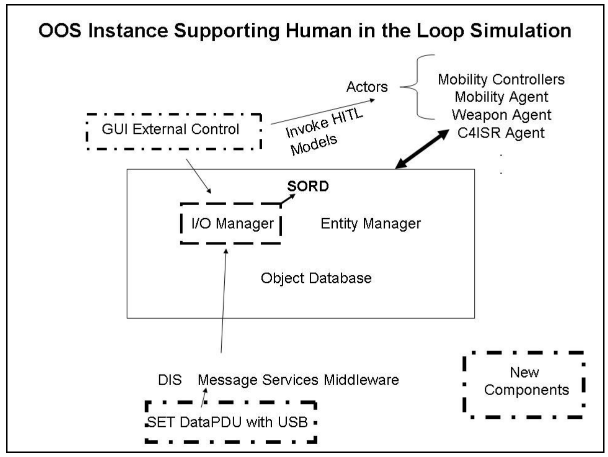

Figure 4: OOS Components for HITL Control

Originally the team attempted to use existing SAF agents to execute the HITL control. In general, OOS SAF models lacked the resolution or control interfaces necessary for HITL control and were difficult to use to generate a sufficiently immersive experience for the operator. In the CGF or constructive mode, many of the agents, for example the mobility agent, can be broken down to sub-agents that act as driver, gunner or commander and simulate steering, collision avoidance and route following. Interfaces to lower level agents accept inputs from controlling agents and simulate the physical movement of the system. Both the mobility and gunnery SAF agents had to be modified to accept HITL inputs in their lower behavioral structure where inputs such as throttle position that are normally provided by higher level behavioral components in the constructive mode are now provided by a yoke and throttle manipulated by a human. It is assumed that with HITL, all control is deterministic based on operator inputs rather than being determined by various high level behavioral agents. Agents used for HITL were modified to accept the HITL inputs, overriding any high level SAF control functions.

In some instances, OOS models simply did not provide the high resolution modeling required for HITL gunnery. For example, the low resolution direct fire weapon model that OOS implemented to support SAF weapon fire was replaced with a ported version of the munitions flyout model that CCTT used for HITL simulation. This permitted the visual feedback of "tracer rounds" to be displayed to the gunner and allowed obstructions along a more accurate ballistic arc to be detected as impact points.

To demonstrate OneSIM development to the simulation community, the ECATT-MR team developed the OneSIM proof of concept that was demonstrated at the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2004 in the PM OneSAF booth. The framework for this demonstration is shown in Figure 5.

During this demonstration the ICV operator's actions generated set data PDUs such as changes in steering and throttle position. These PDUs were sent to the OOS DIS middleware that routed the information to the HITL agent controllers. The output of these agents was modified based on the simulator's entity state such that, for example, a damaged system was not able to fire its weapon. The end product was an entity state PDU that was received by the FCS ICV simulator and used to control the virtual vehicle orientation and state in the virtual environment. In addition, the system also used the constructive OOS capabilities and scheduler to control other CGF entities at the same time that the ICV operator was controlling a HITL controlled entity.

As a result of the successful I/ITSEC demonstration and several ECATT/CGA meetings during this period it became apparent to both the CGA and ECATT/MR teams that the OOS HITL elements in OneSIM could provide risk mitigation for the CGA program. The two teams decided to leverage efforts and began planning and working together, with weekly meetings to discuss and track issues.

TIME MANAGEMENT

One of the areas of greatest concern to CGA was the ability of the OOS scheduler to support the 30 Hz update rate required for gunnery training. OOS provides three different time managers that maintain simulation time as independent, synchronized or sequential. Each of these time managers has specific requirements derived from expected OOS use cases.

- **Independent time management** is used in a single or multi-node environment where the advancement of time is synchronized to node-local wall-clock time. That is, the advancement of one second of simulation time is slaved to the advancement of one second of real time. The processing of scheduled simulation events might not occur at the scheduled simulation time if the system is overloaded or models are not well-behaved. With independent time management, fidelity of the simulation and causality may be sacrificed to allow a more graceful (and possibly imperceptible) degradation of the system's behavior under high load conditions.

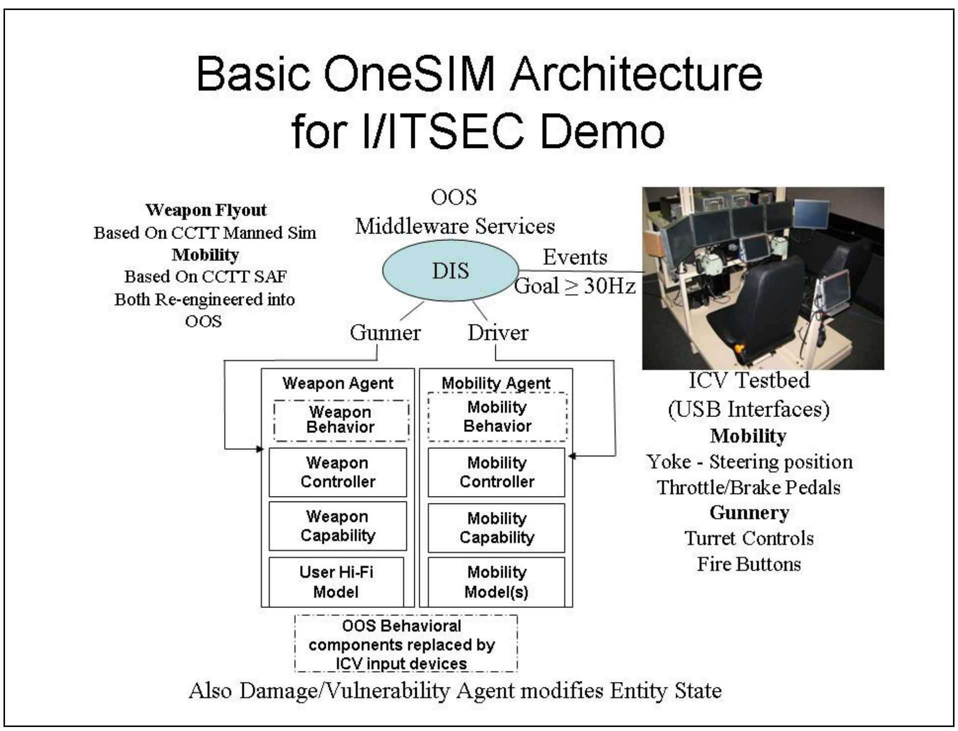

Figure 5: I/ITSEC 2004 Demonstration

- **Synchronized time management** is used in a multi-node environment where the advancement of time is tightly synchronized between all nodes. The major use case for OneSAF for this mode is inter-operating with other constructive simulations. Synchronized time management is usually much less efficient than independent time management, but provides greater repeatability. Repeatability insures that all trainees see the same events for a given scenario. This is important for comparing and scoring Soldier results in gunnery training.
- **Sequential time management** is used on a single node environment, specifically to support use cases where repeatability of the simulation is of the utmost concern. This time management mode allows alternative variables or courses of action to be evaluated at accelerated run rates to support experimentation via simulation.

It is anticipated that a CGA product line, using OOS, will need to modify the OOS scheduler to support hard fixed time frames. The system will also have to synchronize to external time baselines, such as interrupts from image generators or other hardware components. The introduction of externally-driven time advancement would allow the constructive OOS to better support the needs of the virtual domain by servicing external devices to promote increased trainee immersion. The OOS team felt that this presented no unique technical challenges, but creating an OOS fixed frame scheduler would represent additional work that is not currently included in the OOS work plan and should be considered part of the cost for CGA development.

During the evaluation of the OneSIM prototype the system operated near 30 Hz using independent time management with minimal optimization. The OOS development team has identified no barriers to the addition of a fixed frame scheduler as part of the CGA product line architecture development.

CGA TESTBED

The CGA and ECATT-MR teams determined that in order to evaluate the full scope of the CGA product line architecture they needed a testbed that used the OOS Management and Control Tool and could focus on prototyping various capabilities in the CGA PLAS, specifically gunnery exercise generation. ECATT-MR agreed to develop a low cost, reconfigurable testbed to satisfy the CGA requirement. To speed development and keep the costs low, the CGA testbed used the same architecture as the ECATT-MR ICV testbed. It also reused many software components from the ECATT-MR ATO testbeds. Figure 6 shows the CGA testbed.

Figure 6: Driver and Commander in the CGA Testbed

The initial CGA testbed configuration is that of a Stryker Mobile Gun System (MGS) turret crew station. The Gunner, on the left, has a weapons sighting display and Stryker Gunner's Control Handle to control the turret, aim and fire the weapon. The Commander, on the right, has an OOS Plan View Display (PVD) for situational awareness, representing the Stryker's Force XXI Battle Command Brigade and Below (FBCB2) system. The Commander has a Stryker Commander's Control Handle used in conjunction with the Commander's Display Unit shown top right to scan for targets. He can also override the Gunner and control the turret and fire the weapon using the Commander's Control Handle. The center display represents the Stryker's Center Display Unit, with touch screen buttons and switches to select ammunition type and other weapon system management functions.

The testbed hardware architecture was designed to take advantage of available hardware and rapid development. It is powered by five computers of various types. Two computers are used for image generation with Carmel Applied Technologies XIG image generators. A separate computer runs OOS and two additional computers serve as hosts.

The CGA testbed software architecture features the A-Kit/B-Kit design and is based on a central Process Interface Unit (PIU). (Institute for Simulation and Training, undated) The PIU is the sole communication channel between simulation components. It allows multiple programs to interact asynchronously in the simulation. PIU generally hides details of where other simulation processes execute and inter-process communication is handled implicitly. All CGA testbed processes use PIU.

CGA TESTBED OBJECTIVES

The testbed has allowed the CGA team to perform gunnery scenario tests using OOS as the simulation controller, something which previously had not been possible. This resulted in proof that OOS could be used as the simulation control in gunnery exercises and also led to the discovery of issues in OOS that could be resolved before the actual implementation of CGA. There were several objectives in the tests performed on the testbed involving ownship control and exercise generation.

Objective 1: Can a SAF entity be controlled via human-in-the-loop control?

One of the first CGA testbed tests was to verify the system's ability to place an entity within an OOS database and then control that entity using typical gunnery physical controls. The previous efforts of the ECATT-MR ATO on the ICV testbed had already implemented this HITL control method, and it was mirrored on the CGA testbed. Once a SAF entity is under HITL control, a set of control handles mimicking those in a real Stryker are used to rotate the turret and fire the weapon. Turret movement is reflected in the gunner's 3D display and the Commander's PVD. Since gunnery trainers do not typically include the ability to drive the vehicle, this driver functionality was not included in the CGA testbed.

Objective 2: Can a SAF entity controlled as the ownship engage and destroy target entities?

The next step in testing was to determine whether SAF entities could be created that behaved like gunnery targets and whether they could be engaged and destroyed by the ownship vehicle. To test this, a simple "shooting gallery" scenario was created with a stationary ownship and three target tanks that moved from and returned behind cover.

After working out some minor issues, the CGA team completed this scenario. The targets successfully appeared and disappeared from behind buildings at set intervals as intended, and the human-controlled ownship could fire on and destroy the targets.

Objective 3: Can OOS's Management and Control Tool be used to create an approximation of a typical gunnery exercise?

After completing the simple scenario described above, the next task was to create a larger, more complex scenario which closely approximated the look and feel of a gunnery exercise. This new scenario would contain multiple engagements consisting of several hostile targets, as well as routes for the ownship to traverse between engagements.

Using OOS route functionality in conjunction with tasks in the mission editor, the CGA team was able to direct the ownship along set paths between engagements. A phase line placed perpendicular to the route was used to trigger target movement. Figure 7 shows an OOS PVD view of this demonstration. In this scenario, target vehicles were triggered to move into view after the CGA ownship reached a designated point. As in the simpler scenario, targets, when activated, drove from behind buildings and waited for a specified amount of time before returning to cover. Once all engagement targets were either destroyed or had returned to their hidden positions, the ownship followed the next route segment to the next engagement under CGF control.

OOS issues discovered in the creation of this scenario involved the mission editor and its use in creating engagements. Working with OOS, the CGA team resolved enough of these issues to create a working scenario as described above. However, the team continues to investigate issues related to PVD representation of turret/hull alignment, dismounted infantry, and correlated terrain databases.

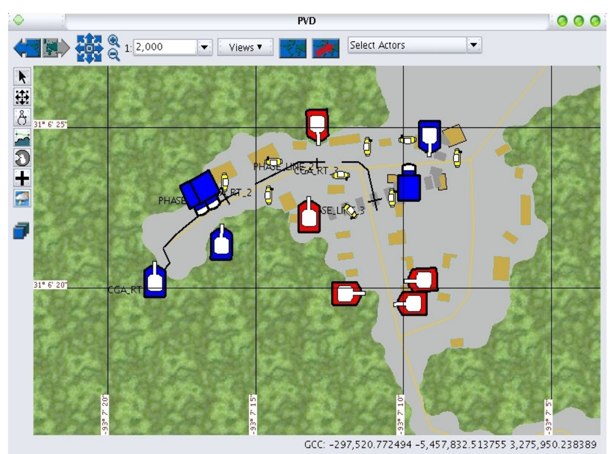

Figure 7: Demonstration of CGA OOS Stryker Gunnery Scenario

Objective 4: Can a gunnery exercise created with OOS be expanded with more entities, such as noncombatants?

The current testing effort involves expanding the gunnery exercise with additional engagements against hostile targets, more friendly entities and the use of noncombatants. The objective of this effort is to push the number of exercise entities beyond the number of entities in typical gunnery exercise, and then observe the performance of the system.

Noncombatants are of interest. Placing noncombatants along the sides of the path of the ownship would better simulate an urban operations environment in the COE. Tests have been performed using routes to move the noncombatants in order to have more moving entities during exercise execution, but these tests have so far been inconclusive. OOS is currently implementing a "milling" behavior for Dismounted Infantry that should provide this functionality.

The CGA team will continue adding to the current scenario in order to test OOS's use in gunnery exercise generation. As new builds of OOS are released, the testbed is upgraded to these newer versions in order to take advantage of new features and bug-fixes. The testbed is currently in the process of loading Build 25 for use in testing.

Objective 5: Future testing – bi-directional feedback, scoring, etc.

In the future, the testbed can be used for further testing of other CGA components outside of exercise generation. One of the next efforts will involve implementation of bi-directional feedback. Currently, the testbed only allows control to be sent to OOS through the input devices. Bi-directional feedback will allow OOS to report status back to output devices like the Center Display Unit. This will enable functions such as the switching of ammunition types.

Another area that may be tested is the scoring component. Automated scoring is a common and critical element in individual and crew training environments but must be integrated with the system's SAF capability to maximize automation. CGA scoring will probably rely heavily on OOS's data collection feature. This, along with the Data Collection Specification Tool, could be tested using the gunnery scenarios previously developed.

ONESAF ISSUES DISCOVERED WITH THE TESTBED

It should be noted that during this effort the OOS program was in the process of full scale development, which typically means frequent new versions (or builds) and identification of issues or problems that are addressed through Program Trouble Reports (PTRs). Early CGA testbed use of OOS identified a number of issues that were provided to OOS via PTRs. These included entities on the map display whose turrets did not rotate and dismounted infantry that could not be killed. Many of these PTRs have been or are in the process of being corrected by OOS. Differences between the various builds also required that the CGA team regenerate scenarios for each new build. The CGA testbed has been a very useful tool to identify OOS issues such as those described below.

- HITL vs. CGF control - During the development of the gunnery scenario described earlier it became apparent that the driver and gunner agents in the CGA ownship would need to be specified independently as either HITL or CGF controlled. In the scenarios, since there was no driver, the ownship vehicle needed to operate via CGF control while the gunner used the HITL control. Ideally this would provide a repeatable gunnery scenario so that gunnery scores could be compared with each other. OOS did not provide this capability. This was corrected by providing a menu in the unit instantiation that allows different agents to be specified as either CGF or HITL control. Likely this concept will need to be extended to other platform agents to allow unmanned elements in the CGA simulator to be role played by the either CGF or by an operator. Repeatability of the agents with the OOS time management (i.e. driver agent travels the same speed each execution) is related to the CGA HITL vs. CGA control issue and will be addressed at a future time.
- Soft Real Time Scheduler – CGA requires a 30 Hz update rate with an objective of 60 Hz. Presently the CGA testbed runs OOS on a 3.2 GHz machine and has been able to run the HITL controls in the gunnery scenarios near 30 Hz using independent time management and about 30 target entities. This is considered to be the typical maximum number of targets in current gunnery training exercises although some COE scenarios may extend the entity count requirement much higher. There have been delays noted when starting new behaviors and scenarios. High entity counts definitely affects update rate but at this time it is uncertain whether this is caused by the testbed IG or OOS. The plan is to continue optimizing OOS to support the HITL control and examine fixed frame time issues.
- Bi-directional instrumentation and controls – During the system development it became apparent that manned simulators operate with a significant number of inputs and outputs (I/O) that

are not represented in current OneSIM design and models. For example, when the operator moves the transmission selector to a new position, data is sent to the model indicating the desired effect. However, this is a unidirectional activity. When assuming HITL control of an entity, there is no bi-directional I/O from the models/agents in OOS to tell the operator the current status of the engine such as RPM, temperature or transmission selection. The ability of the simulation components to support bi-directional feedback will be the subject of future work.

- Support for the COE - Urban environments such as those being encountered today require numerous human entities including dismounted Soldiers and civilians representing multiple sides. OOS plans to include a wide variety of individual combatant and civilian models and behaviors that have been identified by the gunnery community as critical to the current COE. These will be evaluated and included in CGA testbed gunnery scenarios as they evolve from OOS.
- Multiple Ownships from one process - The current OneSIM design assumes that only one entity will need HITL control. When considering possible use cases it becomes apparent that a framework supporting multiple HITL platforms will be needed. Examples include a platoon level gunnery exercise and robotic systems controlled or tele-operated by the ownship.
- High fidelity models – High fidelity models of interest include damage models, flyout models, and environmental models for wind, temperature, etc. Most of the high fidelity models are not yet included in the OOS system. The plan is to implement several higher fidelity models in the testbed to explore the architectural issues of including such models.
- Degraded mode gunnery - This is an important training issue that causes the models to react properly for training when elements of the gunnery system are damaged, either from enemy actions or because the damage is planned as part of the training scenario. This will be looked at in the future.
- Curriculum manager and scoring tools – These are not in the current OOS baseline but will be considered in future work.

CONCLUSIONS

The composabilty of OOS and the ability of others such as the CGA team to create new behaviors using that capability indicate that even if OOS doesn't support specific requirements out of the box, users can add additional functionality themselves. They don't have to wait for OOS to address their issues.

The OOS product line architecture has proven to be a very good starting point for CGA, allowing the CGA and ECATT-MR teams to build a robust demonstration for several gunnery COEs in a very short time. Having the PEO STRI CGA and RDECOM STTC teams work together during the CGA concept development and risk reduction phases has been a very effective and efficient means to initiate a CGA product line architecture. In addition, the CGA testbed has been a very useful tool to educate team members and validate concepts. Testbed use helped identify numerous unforeseen developmental needs that were not identified in the CGA PLAS/PLAF paper exercises. Demonstrations of the CGA testbed have received positive comments from several gunnery subject matter experts. During this collaboration new CGA team members from different contractors have joined the weekly progress meetings and have quickly became educated on CGA and OOS.

For future work, OOS has agreed to make the additional CGA Simulation Common Components part of their OOS baseline releases. This will save the development team the effort of rolling the CGA additions into each new OOS release. The major issue with merging product lines will likely be coordination of OOS builds and test plans. Plans are to continue this joint development and technical exchange until the CGA product line development is under contract.

REFERENCES

US Army Program Manager OneSAF, Undated, *OneSAF Objective System,* http://www.onesaf.org/

Institute for Simulation and Training, University of Central Florida, 2005, *FCS Embedded Combined Arms Team Training / Mission Rehearsal (ECATT/MR) Testbed Architecture* http://dart.ist.ucf.edu/etmr/

Marshall, et al, Development of a Next Generation Embedded Simulation Engine for FCS, *Interservice/Industry Simulation, Training and Education Conference*, Orlando, FL., Dec 2004

SAIC Corporation, *Embedded Systems Final Scientific and Technical Report for Embedded Architecture Development*, 31 March 2005