The OneSAF Objective System Fits Individual Simulation Needs

Chelene Fortier-Lozancich
CrossTalk

Can battlefield simulation software be everything to everyone? The challenges faced by the One Semi-automated Forces (OneSAF) Objective System found out the answer when faced with the Army's modeling and simulation needs, and in the process set a new standard for what they did and how they did it.

Computer modeling software must meet many requirements: there must be common pieces, the components must be flexible for different requirements and be able to meet a user's particular need, and the software needs to be *everything to everyone.*

This challenge was faced by the U.S. Army's modeling and simulation division: how to address a broad range of requirements for a flexible simulation battlefield modeling architecture with a supporting set of components, tools, and services that allows individual users to compose a simulation to meet their individual needs.

This is where the One Semi-automated Forces (OneSAF) Objective System (OOS) comes in. The OOS is composable, next-generation Computer Generated Force (CGF) modeling software that represents a full range of operations, systems, and control processes from the individual combatant and platform level to brigade levels. The OOS accurately and effectively represents specific combat, combat support, combat service support, and command, control, communications, computers, and intelligence activities. "OOS provides a complete simulation environment that supports the entire simulation life cycle from simulation and model development through scenario generation and execution to after-action analysis and review," said Tom Radgowski, program manager for OOS Architecture and Integration. "To meet diverse domain requirements, OOS is developed as a composable line of individual products. Users can combine different products within the OOS product line to meet their individual needs."

As an example of what composability provides to the user, Surdu asked Team OneSAF how the peer-to-peer architecture could be scaled to handle hundreds of thousands of simulated entries. "They told me that the network services layer was architected to allow the simulation to operate in its peer-to-peer mode or it could be run on a single multiprocessor server with shared memory," said OneSAF project manager, U.S. Army Lt. Col. John Surdu. "Due to their layered architecture of OneSAF, this different mode of operation would be completely transparent to the rest of the simulation."

The OOS is designed for use across the three Army modeling and simulation domains: Advanced Concepts and Requirement; Training, Exercises and Military Operations; and Research, Development, and Acquisition. "The requirements for this simulation were that it meet the needs of sophisticated analysts who need high fidelity – as well as staff trainers – who need low fidelity and high entity count," said Surdu. "Team OneSAF has done an exceptional job of creating a scalable, flexible, extensible, composable architecture that is technically the best simulation architecture I have seen in several years of working under the hood in military simulations."

"Peer reviews occur at all phases of the development ... Our metrics indicate that these reviews identify more than 90 percent of all defects."

Team OneSAF Structure

The Team OneSAF approach incorporates government managers, contractor development teams, and end users into a single organization. The U.S. Army Program Executive Office for Simulation Training and Instrumentation awarded a series of task orders to hand pick a set of contractors that could best build the individual pieces of the OOS. They also enlisted on-site representation from the end-user community and reach-back access to a wider group of users for the development process.

"The OOS development effort is characterized by an unparalleled level of cooperation between the government team and the various contractor teams working on the program," said Radgowski.

Science Applications International Corporation (SAIC) served as the OOS Architecture and Integration Task Order lead, and established a comprehensive process set for software and system development. "Our processes are tailored from general SAIC processes that have been externally certified as Capability Maturity Model® Level 4," says Radgowski. "OOS processes are documented in a Web-based electronic process guide that is available to all OOS developers. Compliance to these processes is monitored by independent quality assurance audits and tracked by software development metrics. Peer reviews occurred at all phases of the development to ensure timely defect prevention and optimal product quality. Our metrics indicate that these reviews identify more than 90 percent of all defects."

An Integrated Environment

The core development team is collocated in a single facility and is supported by the OOS Integrated Development Environment (IDE). The IDE is a comprehensive Web-based management and development environment that enables an efficient and effective interchange of ideas and concerns, and facilitates the swift resolution of issues as they occur.

The IDE also provides support services for OOS participants (such as beta site testers) who participate in the program at geographically diverse locations. Access to the IDE is provided throughout the OOS Web site <www.onesaf.org> providing access to numerous tools that help manage action items, peer review artifacts, problem trouble reports, risks, and help desk requests. The BuildBoy application routinely builds and automatically regression tests new OOS software, and publishes build results and status on its Web site.

The Web site is configuration managed, which enables secure, distributed development. The IDE capabilities are a combination of commercial off-the-shelf (COTS) products, custom-developed products, and customized configurations of COTS products, and represent an open network architecture that is capable of scaling large numbers of development machines, rapidly introducing new resources and providing a stable, secure development environment.

OOS Quality Build Methods

The OOS IDE provides automated tools to collect and report technical and management metrics that are reviewed on a regular basis using a formally defined Quantitative Process Management and Software Quality Management process to support the OOS' formal metrics plan. Bi-monthly meetings are held to analyze trends and identify areas where improvements can be made. This allows program management personnel to drill down and examine productivity or quality issues in detail, according to Radgowski.

The OOS is built using a spiral development methodology and extreme programming (XP), and is designed to be hardware-platform and operating-system independent. The developers build and integrate their software on Windows, Linux, and Solaris systems and formally test the results after every development spiral.

"The OOS requirement to integrate a significant portion of directed reuse components into the end product is enabled by the application of XP concepts," said Radgowski. For example, the OOS uses a succession of small, rapid, build cycles to integrate frequent releases, therefore avoiding the problems of a single integration. The process begins with overall four-block (A, B, C, D) planning: a development process where user feedback is incorporated into the final product and tested at sites across the country. Currently, Blocks A and B have been distributed to select organizations within the Army, Navy, and Marine Corps.

Block A was developed to be an initial implementation of the OOS architecture with the corresponding tools, components, and services to allow it to execute entire simulation life cycles. Block B contains a comprehensive set of current OOS components, including the system, unit, entity, and behavior composers; the command, control, communications, computers and intelligence adapter; the military scenario development environment; the 3D viewer; the after action control component; the environmental runtime databases; the data repositories; and the initial software application compositions for execution.

Team OneSAF is characterized by unparalleled cooperation between government and contractors.

"Each of these four blocks is deployed to selected sites for evaluation and comment," said Radgowski. "We provide a user feedback tool so that beta site evaluators can provide comments back to the development team. Senior OOS staff individually evaluate each comment ... If they find a bug, or give us insight on how to make OOS better, we can react very quickly to their comments."

The process begins with overall block planning, which determines the goals for the respective block, and allocates the goals into eight-week builds for individual software development teams. Each team performs detailed planning for a given build four weeks prior to the beginning of the build. Each build contains requirements analysis, design, code and unit test, and software integration phases. Once a development team completes these phases, it formally hands its software to the Integration and Test team, which conducts an independent test of the code. If the code passes this test, it is nominated to the Test Working Group for designation as a user assessment baseline (UAB). If approved, the UAB is then made available for user evaluation and demonstration. This continuous integration process helps ensure that independently developed OOS components remain in sync.

"The use of radical programming has been of significant benefit. Every eight weeks, the program executed Integration and Test and a review of the current state of the software by engineers and, most importantly, the users," said Gregory Miller, senior engineer, Alion Science and Technology, support to TRADOC Project Office OneSAF.

Because of the combined programming methods, any upgrades or fixes are promptly made and the engineers get immediate feedback from users on how to make the system better. "Team OneSAF is manned by people recognized as gurus within the modeling and simulation field. The composable architecture … creates a unique solution and has made fans of skeptics," said Surdu.

Cost

The OOS program was delivered extremely close to cost and time estimates. Since the software is designed to allow all users to interact on the same software, the government only has to maintain one system instead of several. Implementing standardization methods also saves time in training and sharing of information.

One impressive recommendation for the software is the expressed desire for other major programs such as the Marine Corps Combined Arms Staff Trainer, and the Army's Synthetic Theater of War and Program Executive Office for Simulation Training and Instrumentation Common Gunnery Architecture programs to use the OOS software in their development efforts. As well, the Army's Future Combat Systems program has designated OOS as a training common component.

The Army's investment is already paying off. "On quarterly earned value reports, it is amazing for programs to be within 1 percent cost and schedule variance," said Surdu. "Typically the OOS team is within 0.5 percent — and the program has *never* been restructured. I have a strong technical background, but the engineers working on OOS amaze me daily with the strong technical decisions they make. The contractors working on Team OneSAF take the long-term view to make sound technical decisions that are right for the customer."◆

Project Point of Contact

Tom Radgowski
SAIC
12901 Science DR
Orlando, FL 32826-3014
Phone: (321) 235-7739
E-mail: tradgowski@
ideorlando.org