

OneSAF: Tools and Processes Promoting and Supporting a Distributed Development Environment for a Multi-Domain Modeling and Simulation Community

Doug Parsons
Program Executive Office-Simulation Training and Instrumentation (PEO-STRI)
12350 Research Parkway, Orlando, FL 32826
407-384-3821
Doug.Parsons@peostri.army.mil

Robert L. Wittman Jr.
MITRE CORPORATION
3504 Lake Lynda Drive, Orlando, FL 32817
321-235-7601
rwittman@mitre.org

Keywords: OneSAF, Computer Generated Forces (CGF), Semi Automated Forces (SAF)

ABSTRACT: *The OneSAF Objective System is the next generation simulation system planned to provide the U.S. Army with an entity-level simulation to serve three modeling and simulation domains. Postured as an open-architecture, open-source application, the OneSAF program will put this software into the hands of a vast number of developers throughout the DoD. A toolkit of composition editors will eliminate the need to work directly at the software code level to build new weapon system entities and units, to include their associated behaviors. Software development of the OneSAF application has been conducted in a highly robust systems engineering environment based on commercial and government best practices. Once fielded, OneSAF will expand this development environment into a distributed development environment leading toward increased integration and improved configuration management. These tools and processes will create unprecedented participation and leverage the talents and resources of developers throughout the modeling and simulation community to include multi-service, international, industry and academia experts. This paper describes those tools and processes.*

1 Introduction

Since its humble beginnings in 1991 Linux has created a revolution in the open source community. What started as a small group of programmers, some considered as fanatics, has become a force of hundreds of thousands.[4] The value of open source computing is leveraging the talents and resources of an entire community. While the OneSAF Objective System (OOS) will not be truly open source, the source code will be distributed with the software. This will allow users to tailor the simulation to suit their needs. In addition, OOS will facilitate the sharing of work and ideas between various user groups. Initial efforts have focused on the development of a capable, robust, and extensible architecture supporting a toolkit that will allow users to grow the baseline. While development of OOS capabilities by PM OneSAF will continue far into the future, the vision is that, after the OOS is fielded, the majority of functionality enhancements will be greatest in the modeling and simulation community (i.e. among

OOS user groups). Within this vision PM OneSAF will primarily be the integrator and maintainer of the baseline. In order to move the vision to reality the OneSAF program is creating tools and processes to support development across a widely distributed community.

1.1 Brief History and Background on OneSAF

The One Semi-Automated Forces (OneSAF) Objective System (OOS) is the U.S. Army's next generation, composable, entity based simulation system. It is being developed to provide an integral simulation service to the Advanced Concepts and Requirements (ACR), Training, Exercises, and Military Operations (TEMO), and Research, Development, and Acquisition (RDA) domains. With requirements ranging from closed-form analytical support to command-level, human-in-the-loop training, OneSAF will be a High Level Architecture (HLA)/Distributed Interaction Simulation (DIS) compliant, entity-level simulation providing a common solution for a broad range of user requirements.[10] The

OneSAF concept originated in January 1996 following an extensive study that came to the conclusion that the Army was caught in a wasteful spending cycle, making identical or similar enhancements to legacy simulations across three different user domains. In May of 1997 the Deputy Commanding General, Training and Doctrine Command (TRADOC) approved the Mission Needs Statement (MNS) for OneSAF which stated:

“The need for OneSAF capabilities is not a response to a specific warfighting threat against the force; the need is driven by the guidance to reduce duplication of M&S investments, foster interoperability and reuse across M&S domains, and meet the M&S requirements of the future force.”

The Army decided the best approach for overcoming the problems associated with the multitude of aging simulations was to create a single general-purpose entity-level simulation.[15] This solution relies on using lessons learned from successful simulation projects like the Modular Semi-Automated Forces (ModSAF) simulation and the Close Combat Tactical Trainer (CCTT) SAF.

1.2 Lessons Learned

The OOS program has drawn many lessons from the ModSAF system. ModSAF traces its lineage to the SIMulation NETwork (SIMNET) program. The official ModSAF project was initiated by the Defense Advanced Research Projects Agency (DARPA) in 1993 as an open architecture, which could be used to create synthetic agents for a variety of Distributed Interactive Simulation (DIS) applications. Like Linux ModSAF’s beginnings were humble. In fact if not for the decision to release the source code to requesting organizations in the Army, the program would likely be relatively unknown; however, today through direct requests, the baseline is distributed to some 300 organizations supporting simulation efforts throughout the military, both within the United States and Internationally. The Army has since created the OneSAF Testbed Baseline (OTB), based on ModSAF 5.0, to support the simulation community until OOS is available and to provide a user-driven distributed mechanism to mature OOS components. ModSAF was retired in April 2002 and replaced by the OTB. Some of the lessons learned from ModSAF and OTB are discussed below.

Characteristics to continue in OneSAF:

- Release the source code. The talents and resources throughout the modeling and simulation community are significant and well worth leveraging. The community includes international organizations, as

well domestic military agencies, industry and academia.

- Provide a means by which the community can communicate. The ModSAF and OTB reflectors have been a tremendous mechanism for users and developers to help each other address questions and solve problems. In addition, PM OneSAF has begun sponsoring User Group meetings, which will expand in scope and frequency. These meetings not only communicate system capabilities and program status, but also allow users to voice concerns and issues and to describe the work they have done with OneSAF

Improvements that should be made to OneSAF:

- The OneSAF architecture should be developed to allow for greater composability and extensibility. While ModSAF and OTB were certainly modular, the applications were monolithic. The all-or-nothing approach does not lend itself to flexibility or tailoring to fulfill a particular user’s needs.
- Reach out to a larger set of users for development. The user community learned quickly that extending the ModSAF and OTB baselines required a high degree of skill in C/C++ programming. While the talent pool of software programmers supporting the military is significant, there are many more that would contribute if they could. A toolkit that could help bridge the gap between a software programmer and a military subject matter expert would provide a leap ahead in capability development.
- Provide a means by which users can contribute and share. The communication provided through the ModSAF and OTB reflectors was a helpful tool; however, there was limited means by which users could share capabilities until the next major baseline release. Contributions include capabilities and fixes/patches.
- Increase the likelihood that externally developed capabilities and fixes will be integrated into the core baseline or accessible to others in the community. While ModSAF and OTB developers exhibited a great deal of software talent, they also programmed with disparate coding styles. Integrating code into the core baseline was sometimes a challenging and resource intensive activity, if it could be done at all.
- Provide a means by which users and external developers can better understand the OneSAF architecture and systems engineering process. Because ModSAF was developed as a DARPA project with no real intention to be widely distributed, there was less attention devoted to formal documentation. The OTB did a better job of providing documentation, but voids still exist.

The remainder of this paper describes the processes and technology that will be applied to address these lessons learned and enhancing the ability for distributed development across the OneSAF community.

2 Processes

2.1 Distribution process and the relationship to Open Source

For the vast majority of organizations that will request and receive the OOS baseline the distribution process will be much like that employed with ModSAF and the OTB. The key distinction between OOS and ModSAF/OTB is that there will be a formal fielding to select Army locations to include New Equipment Training (NET). For the remainder of the community users will need to download and return a distribution agreement.

The PM OneSAF business model will continue to distribute the baseline with source code at no cost.¹ These and other conditions make the OOS *similar* to Open Source software; however, there are key distinctions between the open source and OneSAF distribution models. The Open Source Initiative (OSI) defines Open Source as software providing the following rights and obligations: [14]

- a. No royalty or other fee imposed upon redistribution.
- b. Availability of the source code.
- c. Right to create modifications and derivative works.
- d. May require modified versions to be distributed as the original version plus patches.
- e. No discrimination against persons or groups.
- f. No discrimination against fields of endeavor.
- g. All rights granted must flow through to/with redistributed versions.
- h. The license applies to the program as a whole and each of its components.

¹ This condition may not apply to International users, of which a fee might be applied to off set costs to create distinct international OneSAF baselines.

- i. The license must not restrict other software, thus permitting the distribution of open source and closed source software together.

Of these a, b, c, g and h apply to the OneSAF distribution. Essentially, there are two basic commitments that a user makes when a distribution agreement is signed.

- (1) Authorization to redistribute the baseline is restricted to PM OneSAF. The baseline can not be freely distributed as defined for open source due to security reasons. The domestic version of the OOS, while not classified, has content (data, algorithms) deemed sensitive by the U.S. Department of Army.
- (2) Users who develop new functionality into the OOS baseline agree to provide those capabilities back to PM OneSAF for possible reintegration.

This constraint does have advantages. Facilitating distribution through a single focal point allows PM OneSAF to have knowledge of whom and how users intend to use the baseline. This knowledge will enhance PM OneSAF's ability to identify and integrate useful externally developed capabilities into baseline releases.

Any organization within the DoD with a valid need to use the OOS will receive the baseline free-of-charge. Industry and academia can also request the OOS baseline, but need a US Government sponsor to co-sign the distribution agreement. While OneSAF is focused as an Army/DoD program, other inter-agency organizations (e.g. other services, homeland defense, emergency response groups/police, etc.) can gain access to the application. For these organizations contact should be established through the OneSAF program office.

Users in the international community also sign distribution agreements. Because of international laws their path to receipt of the OTB or OOS baselines has a number of additional steps that must be satisfied. Basically, there are three different means by which international organizations can engage with PM OneSAF (or almost any other Army program office). The three are Foreign Military Sales, Data Exchange Agreements, and Project Agreements.

Foreign Military Sales (FMS) involve a country's request for goods or services from the U.S. Government. This mechanism differs from Data Exchange Agreements and Project Agreements because funds are provided to the implementing agency, which for OneSAF is the U.S. Department of the Army. Initial

discussions are conducted primarily between the U.S. Security Assistance Office (SAO) personnel and Ministry of Defense (MoD) officials in the host country. Follow-on discussions to further define a country's military requirements may include U.S. contractors, U.S. military departments and U.S. Department of Defense (DoD) headquarters representatives. The process usually involves the host country generating a Letter of Request (LOR) for defense articles and/or services. The LOR is then routed either through the U.S. State Department or directly to the Implementing Agency depending upon if the request is for "Significant Military Equipment (SME)" or other non-SME (see reference [3] for further discussion). If appropriate to proceed the Implementing Agency will prepare and return a Letter of Offer and Acceptance (LOA). The Implementing Agency for OneSAF related requests is the U.S. Army Security Assistance Command (USASAC).

Data Exchange Agreement (DEA) (also Information Exchange Agreement (IEA)) is an agreement between the U.S. Government and another nation to exchange mutually beneficial data. DEA's are conducted as *quid pro quo* as typically no funds are involved in the transfer. Any training or technical assistance services desired by that nation from the U.S. in conjunction with the transfer will be requested through FMS. PM OneSAF currently holds DEA's with a variety of nations to include Canada, Australia, and the United Kingdom.

Project Agreements (PA) are initiated between the U.S. Government and another nation's government to collaboratively develop a system. Each party agrees to contribute resources to the project that ultimately benefit both nations. At the time of this paper, PM OneSAF has a Project Agreement with the United Kingdom.

2.2 Communication

The distribution restrictions specified in paragraph 2.1 may be seen as constraints to open communication within the modeling and simulation community. For this reason PM OneSAF will encourage and foster communication among OOS users and developers, of which include electronic and interpersonal methods.

- a. Reflector. The ModSAF and OTB reflectors have proven to be a valuable mechanism for users and developers to address questions, resolve problems and share experiences.
- b. Web tools. Communication through online tools will greatly allow authorized users to take advantage of each others efforts and experiences. Some of the

user capabilities that could be posted to the web discussed below:

- Software fixes and patches,
- Unique entities, units, behavior and system compositions,
- Military and simulation scenarios,
- Terrain databases & associated SEDRIS Transmittal Formats (STF),
- Unique baselines, and
- Problem Trouble Reports and Enhancement Requests.

Users will be able to electronically submit their enhancements/changes with supporting documentation to the OneSAF web site. These submissions will be posted with the results of automated regression tests against those tools and their architecture compliance score. Other users may download these enhancements "at their own risk." The OOS Configuration Control Board (CCB) may then choose some of these along with PMO-created code to incorporate into the next baseline. Once incorporated into the baseline, PM OneSAF assumes responsibility for these enhancements, and they will be removed from the web page.

- c. User Group meetings. PM OneSAF will host regular meetings (tentatively tri-annual; April, August, & December) for both the domestic and international modeling and simulation community. The user group meeting is intended to provide the opportunity to for users to provide as well as receive information relevant to the community, including:

- Status of the OneSAF program,
- Receive demonstrations of new OOS capabilities, both from PM OneSAF and each other,
- Present their own experiences and capabilities, and
- Voice concerns, issues, and recommendations

2.3 Early user engagements

From the first day of program initiation PM OneSAF has encouraged non-traditional methods to develop the system. Elements of Agile Methods [13] and Extreme Programming [2] were tailored into defining interaction with customers/users, communication, development flexibility, and staffing. Relative to distributed development the program's emphasis on continuous user evaluation is most significant. From the first day of development user and domain representatives, as well as the PEO STRI Government engineering team, have

worked side-by-side in the same development environment. These individuals, who represent the OneSAF customers, are not only welcomed but expected to participate in requirement decomposition, peer reviews, tests, and analysis. Because not all user engagement occurs in the OneSAF development facility, tools (primarily web based) were established to allow these widely distributed participants to peer into the ongoing development process and products. These tools and methods will continue to serve future OneSAF users and external developers. The OneSAF Testbed Baseline itself does not allow future users to evaluate the OneSAF Objective System core simulation infrastructure; however, OOS components have been integrated with the OTB and have been bundled with the OTB v2 distribution (released November 2003). These components include the Military Simulation Development Environment (military scenario composer) and After Action Review (AAR) products. Providing these components to the modeling and simulation community will provide PM OneSAF more than two years of extremely valuable user feedback before the OOS is fielded. In addition, PM OneSAF is engaging with high profile organizations that will help align the program with the U.S. Army's Objective Force, Joint/Inter-service community, and International organizations. Some of these engagements include the U.S. Army's Future Combat Systems; U.S. Marine Corps; and the American, British, Canadian, and Australian (ABCA) International working parties.

The **Future Combat Systems (FCS)** is part of the U.S. Army's solution to provide a capability of crisis response with a rapidly deployable, relevant force against a range of threats in a demanding operational environment. The FCS will be comprised of a family of advanced, networked air- and ground-based maneuver, maneuver support, and sustainment systems that will include manned and unmanned platforms. [8] The OOS has been selected as the embedded simulation to FCS training. The OneSAF Product Line Architecture is under assessment to determine an implementation to allow it to work within the FCS System of Systems Common Operational Environment (SOSCOE). While the OTB will not be a participant in the FCS embedded training solution, it is currently a part of the System of Systems Integration Lab (SOSIL) to investigate FCS alternatives. FCS simulation development for both embedded training and analysis involving OneSAF baselines are occurring at distributed locations. PM OneSAF intends to use these experiences to better serve the entire OOS user community.

The **U.S. Marine Corps (USMC)** is devoting resources to integrating OTB (current) and OOS (future) into their Combined Arms Staff Trainer (CAST) Upgrade. The

CAST will be capable of providing realistic fire support training for Marine Air-Ground Task Force up to and including the Marine Corps Expeditionary Brigade level. The USMC has recently chosen to implement the OneSAF Integrated Development Environment (IDE) in order to mimic the OneSAF systems engineering process. PM OneSAF will leverage the USMC products to provide a more capable Joint/ Inter-service simulation. In the spirit of the original MNS to have *ONE* SAF, Marine work is done in close collaboration with the Army team, and their enhancements will be integrated into the single OOS baseline.

The **International community** is also participating in early development of the OOS. Teams from the United Kingdom, Canada, Australia and New Zealand will participate under the American, British, Canadian and Australian (ABCA) Armies' Standardization Program. Discussions are underway to establish OneSAF development environments (similar to what the USMC has already done) to facilitate those teams to develop representations of their weapon platforms, units and associated behaviors. Under a Project Agreement the United Kingdom will engage with PM OneSAF for deeper developmental collaboration.

3 Tools and Technology Base

3.1 The Selection Process

This section starts by introducing quality factors as a means to evaluate and focus the development of a distributed software development environment. Once identified the quality factors provide guide posts or measures for how well specific tools and technologies support the distributed development community. As with other systematic development endeavors it seems reasonable that identifying driving quality factors for a distributed development environment will help the designers to build and/or integrate in the appropriate set of capabilities.

As a short example twelve quality factors heavily influenced the OneSAF architecture development. Quality factors of importance to OneSAF were identified early in the architecture development process and include among others the familiar composability, reliability, maintainability, scalability, and portability and some not so familiar such as integrability - ability to support test-harness like integrations. These quality factors and their importance to the OneSAF architecture are described in detail on OneSAF.net and are listed in Wittman & Courtemanche [15]. Identifying the most

important quality factors early in the process helped to analyze and select technology to meet the OneSAF requirements. For example the selection of the Java programming language was weighed according to its positive support for the OOS quality factors associated with maintainability, portability, composability against its performance limitations. The decision to use Java was based on this assessment and the ability to mitigate Java's performance limitations through efficient programming practices and model composition techniques.

Quality attributes also help us reflect on how well the OneSAF development environment will support distributed software development once the software is fielded. Currently the OOS development team is located in a single facility known as the Integration and Development Environment (IDE); as a result, we will have to use lessons learned from prior experience and community insight to select the most appropriate tools, technology, and techniques to support a distributed development community. These tools must allow easy migration from the formally controlled requirements, design, implementation, and test environment where hallway and regular face-to-face interactions are the norm to a distributed development environment supporting a community of developers where asynchronous web-based interactions occur across software development cultures (from requirements to integration and test). [12] As Scacchi states from his research on developing and understanding open system requirements:

"Developing open software requirements is a 'community building process' that must be institutionalized both within a community and its software informalisms to flourish." [12]

The OneSAF team is preparing for this transition using lessons from ModSAF, OTB, open source community literature, and practical experience. [4, 14, 12] The expectation is once OneSAF is fielded in early 2006, OneSAF source code, data, and executables will be distributed to the OneSAF community. As mentioned earlier this community includes the DoD, academic, and international communities. The following quality factors played heavily in the IDE tool selection process:

- Usability: "2. convenient and practicable for use," [7]
- Accuracy: "1 : freedom from mistake or error," [7]
- Availability: "3. Present or ready for immediate use," [7]

- Completeness: "1 a : having all necessary parts, elements, or steps" Necessary for understanding," [7] and
- Open Source and free of licensing fees.

The following sections describe the OneSAF software development environment in terms of tools and technologies that aspire to the distributed environment quality factors.

3.2 The OneSAF Software Development Environment

The current OneSAF development environment relies heavily on web-based technology even though the vast majority of the OOS engineering work force is currently collocated. This collocation is expected to change dramatically once the software is distributed to the OneSAF community. There are two community efforts worthy of note, FCS and USMC, as they are early users/developers of the OneSAF Objective System and are already beginning to drive a transition to collaborative distributed development using the tools mentioned below. The following sections describe the OOS tools that have been customized from existing commercial or open-source applications or are being developed from scratch to support distributed collaborate OOS software and model development. These tools are evolving with the benefit of active use within the OOS IDE.

The cornerstone of the OneSAF development environment is OneSAF.net. It is a secure Apache httpd-served Web site, accessible via https that houses historical and current programmatic, organizational, and task order structure, and technical information. Technical information from architectural designs down to the Application Programmer's Interface (API) descriptions are also available via linked web-based access. As mentioned earlier, OneSAF.net is focused at providing adequate context and rationalization of existing design and implementation across the development lifecycle. API definitions are provided by automated code scrappers that generate Javadocs on periodic schedule. The following paragraphs provide a small sample of the web-based development and information services available via OneSAF.net.

The government's acquisition strategy for procuring OneSAF is based on appropriately segmenting the OneSAF system and then hiring the right organization to develop each specific product. At the highest level OneSAF.net is consistent with this strategy and is

organized around the task orders involved in OneSAF. OneSAF.net includes those task orders that are actively working on OneSAF and those that have completed their tasks. There are currently thirteen task orders listed including the Architecture and Integration, the After

Action Review, C4I Interface, and Military Scenario Development Environment. For a short description of each task order see Wittman and Harrison [16]. Figure 1 shows the top-level OneSAF web-page.

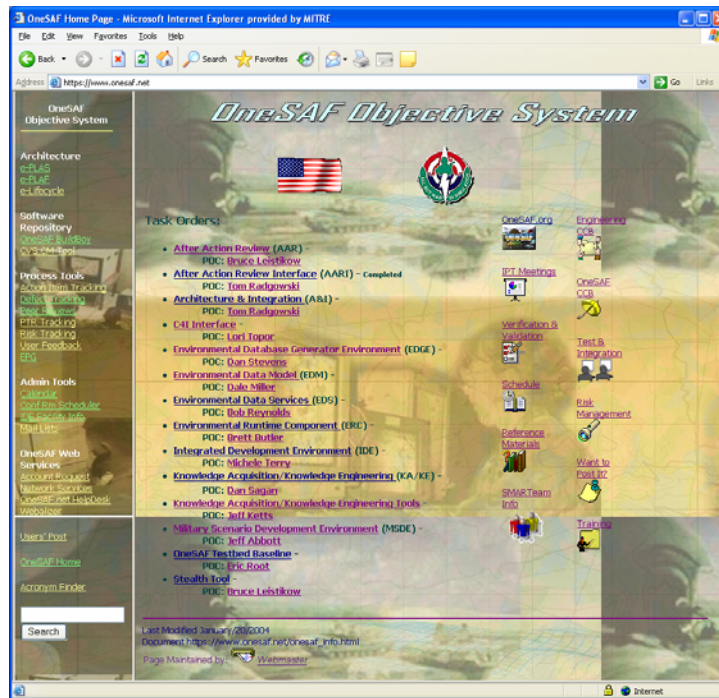


Figure 1: OneSAF.net top level

Each task order publishes their pertinent information for the rest of the OneSAF development community. For example the Architecture and Integration task order has established fully web-linked descriptions and work process artifacts across program control and management, system and software engineering, system integration and test, and software quality assurance. These areas provide historical and current design and implementation decisions in the form of graphical representations, document-based specifications and technical notes, presentations, and other web-based artifacts. The Electronic-Product Line Architecture Specification (E-PLAS) provides a prime example of the web-enabled linkages offered by the A&I's web-pages. The E-PLAS is a fully hyperlinked graphical and textual description of the OneSAF system architecture and includes both static and dynamic perspectives. The linkages allow a user to progress through the architecture description at their own pace and comfort level and offers fully linked access down to the Javadocs describing the OneSAF APIs. [9] Other system and software engineering topics covered within the A&I area include performance tools and models,

data architecture, architecture compliance tools, and training information to allow new developers to quickly learn about and understand the OneSAF requirements, design, and code base structure. OneSAF.net also provides direct and hyperlinked access to the OneSAF System and Software Development Processes. These are contained within the Electronic Process Guide (EPG). Figure 2 shows EPG top-level web-page. The EPG links and provides easy access to appropriate standards, process descriptions, and developmental standards putting this information at the developers' finger tips regardless of their proximity to IDE. The process covers the entire development process from requirements definition, software development, integration and test, finishing with software distribution. A critical piece of the process is the ability to accurately track and manage risks, actions, and defects; as such, OneSAF.net has a customized version of the Web-Request Tracking tool to support each of these areas. The tool allows unique identification of each risk, action item, and defect and tracks the items progress (comments, actions, deadlines, responsibility chain, etc.) through resolution.

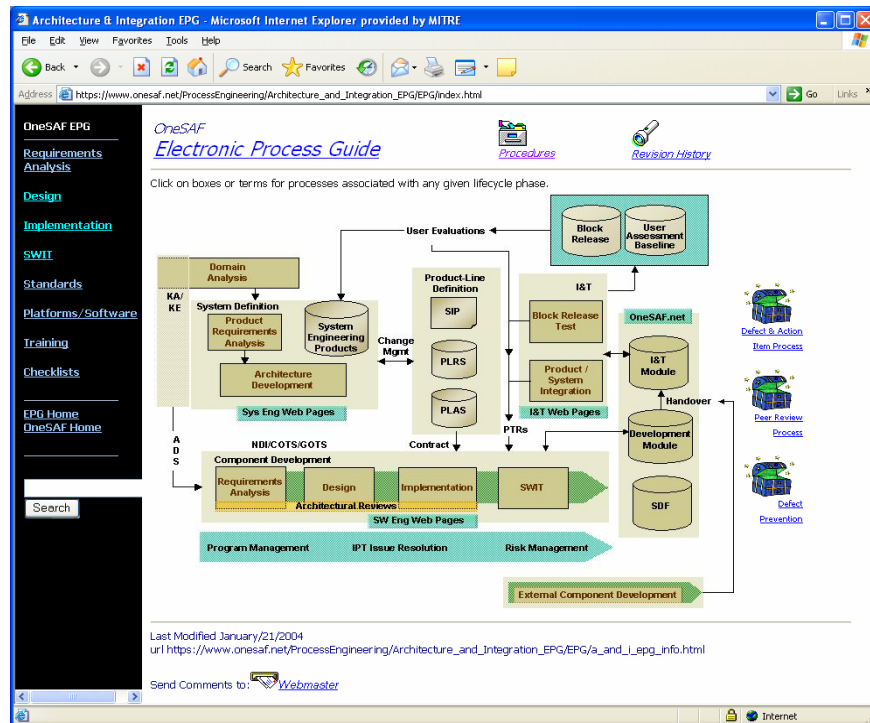


Figure 2: OneSAF EPG top level

3.3 A Web-Based Foundation

As mentioned earlier, web-based technologies provide the cornerstone for information dissemination and tracking within the OOS IDE. The following subsections describe the web-based tools and technologies critical to the OneSAF distributed development environment. The OneSAF development environment is built around a web-based architecture. OneSAF.net employs an **Apache httpd server** to distributed the information described in section 3.2. For more information on the Apache web-server see <http://www.apache.org> [1].

Distributed asynchronous discussions and archiving is provided via e-mail using the GNU free open-source product **Mailman**. Mailman provides an integrated web environment for managing electronic mail discussions and e-newsletter lists. It offers a complement of mail list functionality including built-in archiving, automatic bounce processing, content filtering, digest delivery, spam filters, and web-based list administration. [5, 6] More information on Mailman can be found at <http://www.gnu.org/software/mailman>[5].

Configuration management and revision control processes and services are built around the **Concurrent Versioning System (CVS)**. CVS version 1.10.8 is freely available open-source software and provides revision control for all software development and web-based information developed and used by the OneSAF team. CVS allows documenting changes and managing these changes as part of a specific baseline. Developed as a software development revision control tool OneSAF expands its utility and uses it to manage and track all aspects of OneSAF data as part of its configuration management processes and tools. These qualities are critical for OneSAF development. More information can be found at <http://www.cvshome.org> [11].

Automated support for requirements management and tracking is provided by the **Dynamic Object-Oriented Requirements System (DOORS)**. Although not freely available nor open-source this automated tool was deemed necessary to support the requirements driven OneSAF development process. DOORS version 7.0 provides automated support for OneSAF's rigorous requirements analysis and tracking process and is accessible to all task order participants within the IDE. DOORS allows requirements storage, retrieval, and

maintains linkages between user, system, and implementation requirements. Currently the OneSAF DOORS installation stores the OneSAF Operational Requirements Document (ORD), the OneSAF Technical Requirements Document (TRD), The OneSAF Operational Concepts Document (OCD), the OneSAF Reuse Direction Guide (RDG) and the Product Line Requirements Specification (PLRS). For more information on DOORS please access <http://www.telelogic.com>. [11]

Automated software design and development support is provided by the **Together Control Center** version 6.0. Again Together is neither free nor open-source, but was deemed necessary to meet the managed Software Engineering Institute level 4 software development process in use by the A&I contractor. The Together Control Center allows integrated access to a user-configurable suite of software development tools. These tools span the software development lifecycle from analysis through test. [11] For more information see <http://www.togethersoft.com>.

Automated risk tracking, action item tracking, and defect tracking are handled using the open-source freely available **WebRT** tool. WebRT 1.0.1-4 has been customized to provide a web-enabled tool to track and manage defects, issues, risks, and action items across the OneSAF task order community within the IDE. [11] For more information see (<http://www.bestpractical.com/index.html>).

Java provides a platform-independent development language and development kit to OneSAF. Sun's Java version 2.0 along with the Software Development Kit Version 1.4.1 provides the Java language programming foundation for the OneSAF IDE. OneSAF is taking advantage of many JDK 1.4 features including: design-by-contract, logging, cross-application cut/copy/paste and drag-and-drop, via the Swing Data Transfer Architecture. OneSAF is reviewing the capabilities and schedule for stepping up to next major release of the Java SDK. For more information see <http://www.javasoft.com> [11].

As data architecture and management play a critical role across the pre-exercise, run-time, and post-exercise activities OneSAF is leveraging XML technologies including **XML Spy**. XML Spy version 4.0 provides the OneSAF users within the IDE the ability to create XML Schemas that comply with the OOS Data Interchange Formats (DIF) standards. XML Spy features a format checking and validation tool to cross check a document against its DIF. XML Spy supports both Data Transformation Definitions and XML Schemas. For more information on XML Spy see <http://www.xmlspy.com> [11].

3.4 Tool-based System and Model Development

In order to give users more independence above and beyond access to source code OneSAF is producing two independent classes of composition tools allowing user to independently create new OneSAF system and model compositions. These composition tools are being developed in addition to tools supporting more traditional simulation scenario development, simulation initialization, simulation control, and after action review applications. Both classes of composition tools use Graphical User Interfaces and support processes to remove, to a large extent, the dependency on software experts to develop new simulation system compositions across the training, testing, and concept development domains, as well as new unit, entity, behavior, and environment model compositions.

The first is the **System Composition tool**, shown in Figure 3, used to combine and develop unique instantiations of the OneSAF system. System Composition allows individual OneSAF models, tools, and infrastructure services to be combined to create a complete system level capability supporting specific end user needs. This tailoring means users get, or can compose themselves, only the components they need relieving them of the extra complexity and overhead of unnecessary components. These systems are created using a "JavaBeans" like composition metaphor. The OneSAF composition technique depends on additional software component meta-data that is used by the composition engine to ensure components will work meaningfully together and disallow component connections that are illogical.

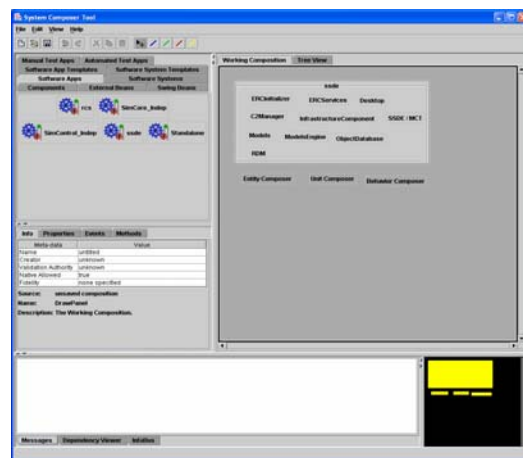


Figure 3: System Composer

The second class of composition tools are called Battlespace composition tools and offer military domain users the ability to create, modify, and extend the set of things that can be represented in the simulation without developing software. The composition tools use and build on existing primitive and composite models to develop new and unique entities (e.g., individual combatants, helicopters, tanks, sensors, weapons, etc.), units (e.g., organizations of entities that behave according to certain sets of rules or doctrine), and behaviors (e.g., move tactically, defend position, etc.) that are associated with units and entities. The construction of these models may include model components that vary across a range of physical and behavioral fidelity (e.g., low, medium, and high). The composition tools will provide model validation checks, using model developer provided meta-data, to ensure the various fidelity levels work appropriately and effectively together. For example a high fidelity mobility model may be coupled with a low fidelity acquisition model. Specific model parameters and then defined as part of the initialization process. The following list describes each of the model composition tools.

Entity composition is handled by the **Entity Composer Tool**. Figure 4 shows the Entity Composer Graphical User Interface (GUI). The composer provides the user with a drag-and-drop capability to develop new OOS entities. The basic idea is to attach the appropriate physical models (mobility, vulnerability) to a platform (body or hull) and then associated specific weapons, sensors, and communications devices to that platform. Once saved the entity can be modified and associated within a unit structure and have behaviors allocated to it. The tool supports the ability to create representations of existing equipment as well as to create experimental entities. New entities can then be distributed and shared among the OOS user community.

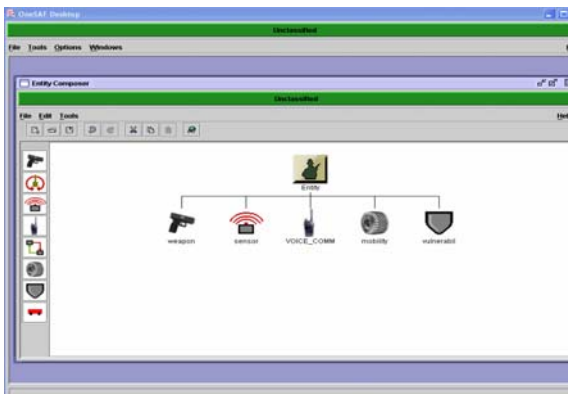


Figure 4: Entity Composer

Unit Composition is supported with the **Unit Composer Tool**. Figure 5 shows the Unit Composer GUI. This tool allows entities to be combined to form asymmetrical friendly, enemy, and neutral type organizations. Both doctrinally correct organizations and new organizations can be developed to support experimentation and concept development efforts. Unit compositions can also be shared among the OOS user community.

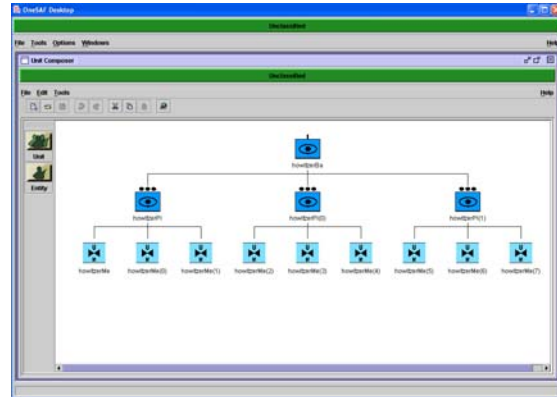


Figure 5: Unit Composer

The **Behavior Composer Tool** allows users to create new behavioral representations that are then associated with units and entities. Figure 6 shows the behavior composer. This tool allows the creation and/or modification of behaviors that entities and units will use to guide their interactions within the simulation. At the top level the behavior composer allows parallel and sequential process flows to be defined. It also support continuous processes that act as background tasks such as “look for enemies” and tasks that are triggered by specific events such as “find cover when fired upon”.

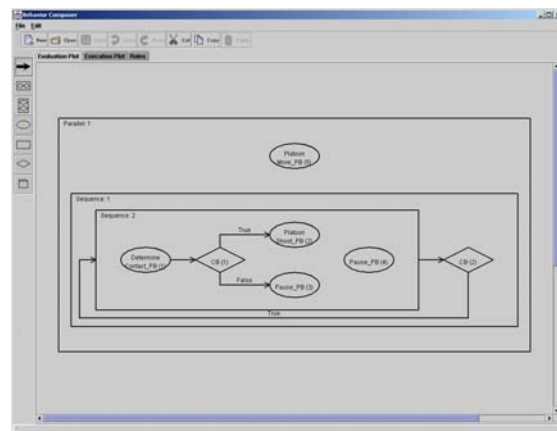


Figure 6: Behavior Composer

These composition tools intend to provide users the ability to extend, enhance, and share OneSAF models without direct interaction and/or support from the OneSAF software developer or the OneSAF Project Management organizations.

4 Summary

This paper reviews the OneSAF processes, tools, and technologies that are being developed to support OneSAF community wide distributed development. It looks at the lessons learned from earlier projects such as the OneSAF Testbed Baseline and open source development endeavors such as Linux and Agile and Extreme programming techniques. Finally it looks at the tools and foundational technologies OneSAF is using to prepare itself to migrate to an open source-like development environment once it is fielded in the 2006 timeframe.

5 References

- [1] Apache.org web site (<http://www.apache.org> & <http://www.apache.org/docs/misc/FAQ.html#what>)
- [2] Beck, Kent; "Extreme Programming Explained – Embrace Change", Addison Wesley Professional, 1999.
- [3] The Defense Institute of Security Assistance Management, The Management of Security Assistance, Chapter Six – The FMS process, <http://www.disam.dsca.mil/pubs/DR/greenbook.htm>
- [4] Hasan, Ragib. History of Linux, July 2002, <https://netfiles.uiuc.edu/rhasan/linux/>
- [5] Mailman web page. (<http://www.gnu.org/software/mailman>)
- [6] Manheimer, Ken; Warsaw, Barry; Viega, John. "Mailman: An Extensible Mailing List Manager using Python", Proceedings of the 7th International Python Conference, November 10-13, 1998, Houston Texas. <http://www.m-w.com/>
- [7] Merriam Webster Online Dictionary. <http://www.m-w.com/>
- [8] Mission Need Statement for Future Combat Systems, 29 August 2002.
- [9] OneSAF Electronic-Product Line Architecture Specification web-page https://www.onesaf.net/Architecture_and_Integrati
<on/SystemsEngineering/SystemArchitecture/PLAF/EPLAF/Graphics/plaf.jpg>.
- [10] OneSAF Operational Requirements Document Version 1.1, Dated 21 May 2001. TRADOC.
- [11] OneSAF.net Software Baseline List: https://www.onesaf.net/Architecture_and_Integration/SoftwareEngineering/Tools/Tool_Set/SEE/SBL.html
- [12] Scacchi, W. "Understanding the requirements for developing open source software systems." IEE Process and Software, Vol 149, No 1, February 2002.
- [13] Turner, Richard; Boehm, Barry, "People Factors in Software Management: Lessons From Comparing Agile and Plan-Driven Methods", CrossTalk Journal of Defense Software Engineering, Vol. 16, No. 12, December 2003.
- [14] Webbink, Mark. Understanding Open Source Software, <http://www.groklaw.net/article.php?story=20031231092027900>, 31 Dec 2003.
- [15] Wittman, Robert L.; Courtemanche, Anthony J.; "The OneSAF Product Line Architecture: an Overview of the Products and Process.", SimTeCT 2002, Paper 38, Melbourne Australia.
- [16] Wittman, Robert L.; Harrison, Cindy J. "The STRICOM Integrated Development Environment". SimTecT 2002 Paper 13, Melbourne, Australia.

Author Biographies

DOUGLAS J. PARSONS is the Lead Engineer of the Intelligent Simulation Systems Team at the U.S. Army Program Executive Office for Simulation, Training, and Instrumentation (PEO STRI). In this role his primary focus is toward the successful development of the One Semi-Automated Forces (OneSAF) Objective System. Mr. Parsons holds a B.S. in Mechanical Engineering from North Dakota State University, a M.S. in Systems Management from Florida Institute of Technology, and a M.S. in Industrial Engineering from the University of Central Florida.

ROBERT L. WITTMAN JR. currently works for the MITRE Corporation supporting the OneSAF program. He has been part of the U.S. DoD M&S community since 1990. He holds a B.S. in Computer Science from Washington State University, a M.S. in Software Engineering from the University of West Florida, and a Ph.D. in Industrial Engineering from the University of Central Florida.