Standardization, Transformation, & OneSAF

John Logsdon & Dr. Rob Wittman
OneSAF Program – Program Executive Office Simulation Training and Instrumentation
3045 Technology Parkway
Orlando, FL 32826
USA

John.Logsdon@us.army.mil./rwittman@mitre.org

STANDARDIZATION, TRANSFORMATION, AND ONESAF

This paper will discuss both the role that reuse has played within the One Semi Automated Forces (OneSAF) system design and how the reuse of OneSAF has now become the major factor in the transformation of the United States (US) Army.

The first part of this paper will detail how the three USA Modeling and Simulation Domains collectively guided the OneSAF system design to reuse existing Knowledge Acquisition/Knowledge Engineering (KA/KE) artifacts, physical models and data; and software components from other simulations. Additionally, the paper will detail how the reuse of standards within the OneSAF architecture supports interoperability across a wide range of interfaces including: US Army and Joint Command and Control (C2) systems; Distributed Interactive Simulation (DIS) 2.04 & 2.06; High Level Architecture (HLA) RPR and other Federation Object Models (FOMs); integrated with both the 1.3.6 and 1516 versions of the Run Time Infrastructures (RTIs); and the JC3IEDM. How data modeling standards such as Base Object Model (BOM), eXtensible Markup Language(XML) and SEDRIS are used. Finally, the paper will detail how we are working the Military Scenario Definition Language (MSDL) through SISO/IEEE standardization process as a Product Development Group (PDG), how the MSDL PDG is collaborating with the Coalition-Battle Management Language C-BML PDG and how we intend to bring other recommendations forward into the SISO/IEEE process.

The second part of the paper will discuss how OneSAF has become the basis for standardization within US Army constructive and virtual simulations products (including WARSIM, AVCATT and CCTT) being developed and fielded by PEO STRI; how OneSAF and its components (ERC, C4I Adapter) are being employed as "Common Components" by a range of organizations and programs across the US Army; and how transformation of key future capabilities such as the Future Combat System (FCS) are being enabled across their entire product development cycle (system analysis and design, test and integration, embedded training) through application of a common OneSAF product line. Finally, the paper will discuss how these elements of OneSAF might be applied across a different community, such as NATO, to facilitate higher degrees of commonality and interoperability.

1.0 INTRODUCTION

In January 1996, the Computer Generated Forces (CGF) Assessment Working Group completed an assessment of the various CGF models in the Army and briefed the results to Army leadership. As a result of that assessment, the Army's CGF investment strategy for the future was established. The primary recommendation was that a flexible and composable SAF architecture should be developed that could integrate the best features of existing CGFs such as Modular Semi-Automated Forces (ModSAF) and the

Logsdon, J.; Wittman, R. (2007) Standardization, Transformation, & OneSAF. In *Improving M&S Interoperability, Reuse and Efficiency in Support of Current and Future Forces* (pp. 20-1 – 20-14). Meeting Proceedings RTO-MP-MSG-056, Paper 20. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int.

Close Combat Tactical Trainer (CCTT) SAF. In May 1997, the Deputy Commanding General (DCG), Training and Doctrine Command (TRADOC), approved the Mission Need Statement (MNS) for the OneSAF model. Army Modeling and Simulation Executive Council (AMSEC) guidance to begin the OneSAF project was provided in May 1998. In June 1998, an approved OneSAF Operational Requirements Document (ORD) was published (V1.0). Since then program execution against those requirements has been guided by the OneSAF Training and Doctrine (TRADOC) Program Office (TPO) acting as the Combat Developer representing the Advanced Concepts and Requirements; Research Development and Acquisition; and the Training Exercise and Military Operations (ACR/RDA/TEMO) domains. Senior level overview has been provided by the Army Model and Simulation Office (AMSO) (now G-3) to ensure AMSEC guidance was followed. OneSAF was formalized as an Acquisition program in April 2000. Shortly thereafter, 01 May 2000, PM OneSAF was formally chartered, by the Army Acquisition Executive (AAE), to manage all OneSAF efforts.

The key aspect of the OneSAF Mission, as stated within the MNS is: “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.” [1]

These actions established a unique programmatic basis for the execution of OneSAF.

1.1 PROGRAMMATIC VISION

OneSAF has had standards as part of its program basis of execution from its origination. The execution methodology, contracting approach and guiding technical principles for OneSAF were formalized prior to the formal charter of the program by a special “Red Team” comprised of senior staff at the Simulation, Training and Instrumentation Command (STRICOM). The Deputy Under Secretary for Operations Research DUSA (OR), Mr.Walt Hollis, provided senior Department of the Army guidance to this process. In particular, the DUSA (OR) emphasized that existing standards such as High Level Architecture (HLA), Distributed Interactive Simulation (DIS), eXtensible Markup Language (XML) and SEDRIS be formally incorporated in the acquisition contracts; that engineering analysis be conducted to maximize the reuse of existing data and products; and that the program provide new data and products it developed back out to the community for reuse and consideration as standards. An extensive initial review with continuing oversight has been conducted both by the Battle Command Support Element (BCSE) (formerly the Army Modeling and Simulation Office (AMSO)) and the OneSAF Senior Technical Review Board (STRB) for existing and emerging standards for use in OneSAF system development. In 2004, AMSO emphasized their commitment to standardization by providing start up funding for PM OneSAF to bring the Military Scenario Definition Language (MSDL) forward into the Simulation Interoperability Standards Organization (SISO) process for consideration as an international standard. PM OneSAF remains committed towards continuing both embracing new standards as well as providing its own developmental candidates into the various standardization bodies.

1.1.1 PROGRAM LEVEL REUSE DIRECTION AND GUIDANCE

In 2001, as part of the formal reuse assessment, it was determined that there were a number of areas where reuse was a feasible approach to meet the formal OneSAF requirements, as defined in the OneSAF Operational Requirements Document (ORD) v1.0. After a significant review process headed by the Director, AMSO, he designated the following as formal “Directed Reuse” areas: by the program:

1. Army Material Systems Analysis Activity (AMSAA) Physical Model Algorithms and data.
2. The Environmental Runtime Component and C4I Adapter from the WARSIM program.
3. The Military Scenario Development Environment (MSDE) and PowerSTRIPES After Action Review (AAR) from the Close Combat Tactical Trainer (CCTT) program.

1.2 ARCHITECTURE VISION

The OneSAF architecture as discussed in the context of this paper applies to the overall OneSAF product line architecture and not the architecture of one or two applications, components, or products within the product line. The architecture in this sense is what provides the structure, standards, processes, technologies, and interfaces allowing the individual tools to integrate and interoperate in meaningful ways.

Architectural related discussions, methodologies, products, and architecture focused teams have been of critical importance to the OneSAF development. OneSAF as a software intensive system has relied on architectural-based processes to develop and document stakeholder consensus on system wide design decisions; disseminate and institutionalize system wide design decisions, develop and coordinate priorities on software quality requirements and needs, and provide a description of the "as built" toolset components.

The OneSAF architecture description is manifested not only in the code itself but also described in the 3 volumes of the OneSAF Product Line Architecture Specification (PLAS). Volume I contains the Architecture Overview, Volume II describes the Concept of Operations for a components based perspective, and Volume III Product and Component Specifications provides the API interfaces to all publicly available interfaces. The 3 volumes are available at onesaf.net.[2] The interfaces are parsed from the code base so the architecture description maintains equivalence with the implementation.

OneSAF architecture development technologies leveraged heavily from the Product Line Architecture concepts formulated within the Software Engineering Institute (SEI). Software asset reuse is a cornerstone of the Product Line Architecture methodology where an asset may be a line, module, or framework of code, a test methodology or other process related artifacts, or an architectural description of how product instantiations are formulated from existing components (also considered software assets). The concept addresses a need to quickly and at minimal expense create meaningful variations on a core set of capabilities and assets to support different users or different uses. Because of the broad user community and the resulting broad yet overlapping requirement set the OneSAF technical team settled on a Software Product Line development strategy in support of the OneSAF Program Office.

The architecture description intentionally and specifically created a conceptual and literal home for the reused products, standards, and technologies identified earlier as well as describing the boundaries for how their internal component architectures would have to change to fit within the overall OneSAF toolset. In many cases a high level notion of the interface was provided and the detailed component level API evolved as the software engineering teams worked through their iterative and incremental design and implementation phases. The resulting OneSAF Product Line Architecture Framework (PLAF) as shown in Figure 1 provides the component breakout associated with version 1.1 of the OneSAF toolset. A description of each component is beyond the scope of this paper but is in Volume 1 of the PLAF as well as conference papers presented at the Simulation Interoperability Workshop (SIW) and Euro-SIW events. [3, 4, 5]

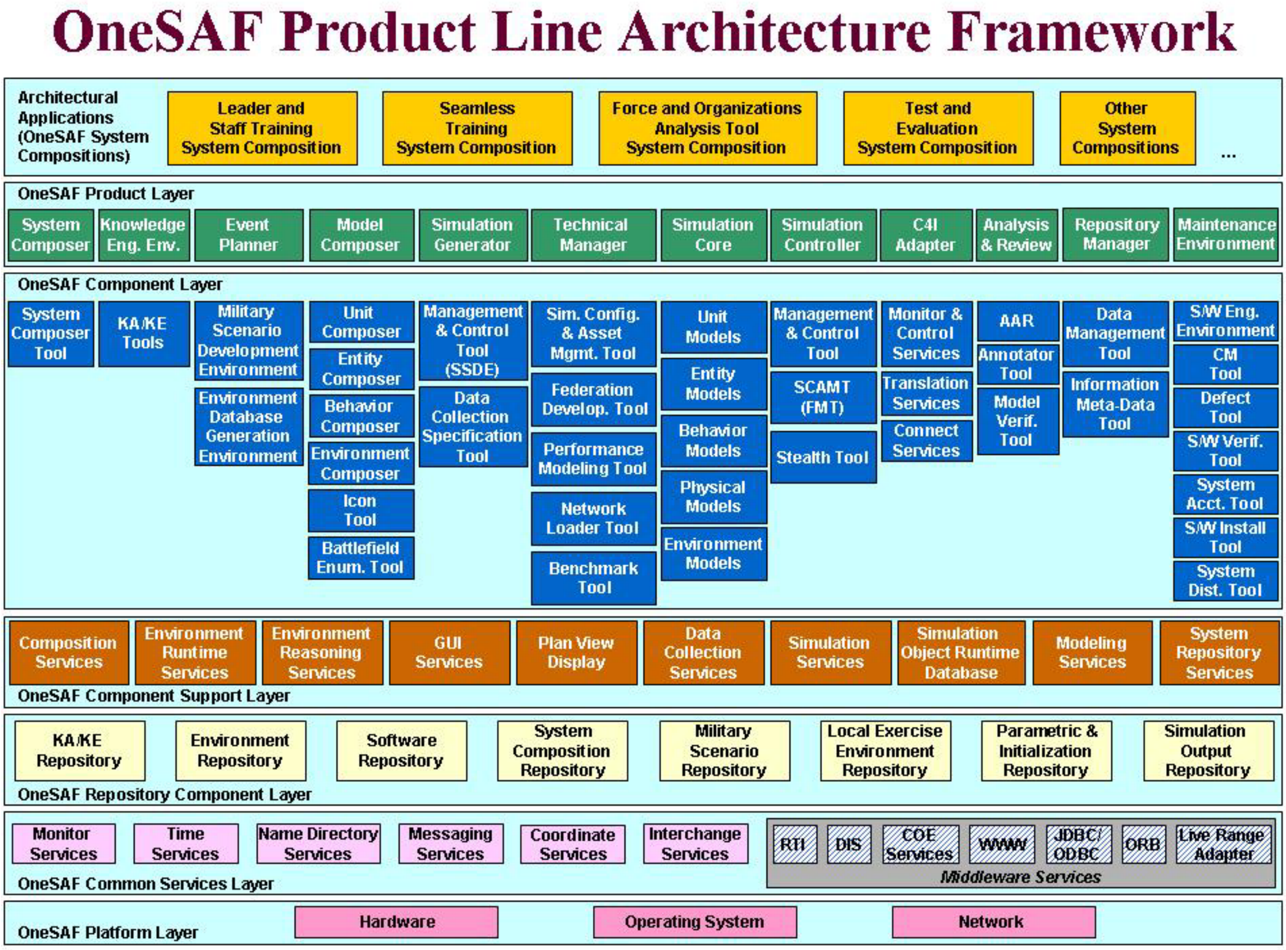

Figure 1: OneSAF Product Line Architecture Framework [3]

2.0 STANDARDS-BASED REUSE

There are two important aspects to standards-based reuse within OneSAF. The first focuses on implementing standards to take advantage of existing investments in other standards-based applications such as those that use the Distributed Interact Simulation (DIS) protocol or the High Level Architecture (HLA) and underlying technologies such as TCP/IP and UDP. The second is centered on performing system and software development in such a way as to enable OneSAF program standards to be proposed and considered for industry wide status in areas where standards currently do not exist but would benefit from standardization as a means to reduce development and cost risk, and by enriching capabilities and performance.

In dealing with implementing existing standards OneSAF capitalized on both “systematic” and “opportunistic” reuse opportunities during development. From the systematic perspective, as mentioned earlier, OneSAF leaders mandated a number of US DoD, commercial, and international standards to lower OneSAF production risk. Additional standards and practices were also adopted by the OneSAF architect team to continue the reuse theme.

2.1 INTEROPERABILITY STANDARDS

The OneSAF architecture and resulting implementation takes advantage of a number of interoperability standards that span the simulation event lifecycle: the pre-exercise, exercise execution, and post-exercise phases. In doing so OneSAF and other standards-based simulations benefit by having the potential to share scenario and environment definition and initialization data, runtime execution data, and data collected during runtime.

As part of the pre-exercise scenario definition and initialization phases OneSAF leverages open commercial standards such as the World Wide Web Consortium's (W3C) eXtensible Markup Language (XML) to hold instance documents that describe military scenario and simulation scenario specific information including entity, unit, and associated behavioural composition information A significant advantage of using an XML-based approach and supporting XML Schema definitions is the availability of open, free, and/or commercial parsers, viewers, data editors, and data verification and validation tools. Early on OneSAF developed and enforced a policy of defining schemas for the data sets associated with military scenarios, composition files, and simulation specific input and output files. As these XML Schema definition files are improved for clarity and conciseness the goal is to promote them through the OneSAF local development process and submit them to international bodies such as the Simulation Interoperability Standards Organization (SISO) for community-wide standardization. A prime example of this vision is being executed as part of Military Scenario Definition Language (MSDL) SISO/Institute of Electrical and Electronics Engineers (IEEE) international standardization effort which will be discussed later in the paper.

OneSAF also leverages the SEDRIS standard, in the pre-exercise phase, for storing environmental data in a simulation independent format, called a SEDRIS Transmittal Formatted (STF) file. STF compliant terrain skin data is compiled with shapefile feature data to create the OneSAF specific environment representation called the Objective Terrain Format (OTF). Alternatively an OTF formatted file can also be directly produced from raw environmental data using an OTF write API provided by OneSAF. The OneSAF Program Office has worked with commercial environmental tool vendors to produce consistent SEDRIS Formatted Transmittal file and OTF formatted files from the same sources. Once these data sets are produced the OneSAF Environment Runtime Component (ERC) uses the OTF files for initialization and execution, while other visualization components and/or external simulations, use the consistent SEDRIS Transmittal Formatted file to initialize their applications. For a complete list of the SEDRIS technologies and standards please refer to http://www.sedris.org. [6]

During runtime OneSAF employs translation-based middleware software to engage with High Level Architecture (Version 1.3, and IEEE 1516) and/or Distributed Interactive Simulation (IEEE 1278) federations. OneSAF can be configured to run in a standalone mode either on a single host or in a distributed, networked mode, across multiple hosts. In networked mode the user/administrator may decide to distribute the models across multiple hosts; to distribute the GUIs (Management and Control Tools) to multiple hosts; or to distribute both the GUI and models to multiple hosts depending on the computational resources available, the entity resolution and entity count required, and the simulation to real-time rate of expected progression. When in networked mode OneSAF uses its Simulation Object Runtime Database (SORD) to share data and ensure consistency between the distributed OneSAF nodes. The contents of the SORD data model were initially based on the Real-time Platform Reference (RPR) Federation Object Model (FOM) with extensions for overlays, command and control messages, and other simulation and application management aspects.

The Joint Consultation Command and Control Information Exchange Data Model (JC3IEDM) is the last international-based standard that we will discuss in this section. [7] Although OneSAF did not originally use the JC3IEDM or its predecessor the Command and Control Information Exchange Data Model (C2IEDM), efforts are now underway to align OneSAF pre-exercise military scenario data and C2 message-based data produced during runtime with the JC3IEDM and associated web-services.

For After Action Review OneSAF stores collected data in XML formatted files as well as a Standard Query Language (SQL) compliant relational database. The OneSAF tool set also comes with a Microsoft Powerpoint tool to extract and visualize data reposed in the SQL database.

2.2 DEVELOPING, MATURING, AND PROPOSING CANDIDATE STANDARDS

In working with SISO to support the OneSAF Program Office's vision of developing and maturing program standards into industry-wide standards a number of OneSAF architectural characteristics stand out as fundamental enablers. MSDL offers a prime example of how these key architectural characteristics help promote broader, community-wide standardization:

- **Separation of code from data**: An MSDL formatted document includes all of the data necessary to define the initial conditions of a military scenario and the initially-planned missions and tasks associated with the organizations defined within the MSDL document Furthermore, an MSDL document is defined using XML and is independent of any other programming language, database, or application.

- **Use of industry standards for data format and content definition**: MSDL is defined using the World-Wide Web Consortium (WC3) eXtensible Markup Language (XML) schema definition. [8] This allows MSDL instance documents to be validated with respect to format and content prior to document use.

- **Application independence**: MSDL documents are fully explained by their accompanying standardized XML schema and accompanying MSDL information products. MSDL documents can be, and have been, populated manually, via database applications, via C4I device output, or via specially designed tools.

- **Separation of concerns:** MSDL's exclusive focus is restricted to information relevant to a military scenario and does not include application specific, training specific, exercise control specific, or other types of simulation initialization type information. This type of information might include the speed of simulation execution, specifying the forces that will be represented either in a live, virtual, or constructive manner, etc.

In addition to MSDL other OneSAF products are aligned with these architectural characteristics and are poised to address industry concerns over a number of issues. The OneSAF products that are being considered important to the international community for standardization include:

- **Objective Terrain Format (OTF):** The OneSAF Objective Terrain Format defines the file structure and content the OneSAF Environment Runtime Component (ERC) uses for initialization. The OTF contains the polygonal representation of the terrain skin for a particular geographic area with attribution for areal, linear, network, and point feature data as well as the Ultra High Building Resolution (UHRB) data. UHRBs contain building interior structure (rooms, walls, doors, stairwells, etc.) and content (desks, chairs, clutter, etc.) data. The feature data and attribution supported within the OTF are defined within a separate data model called the Environment Data

Model (EDM). The OTF package includes a publicly available write API to populate data within the OTF file. Commercial environment data producers are currently using the OTF and associated APIs to produce data sets for OneSAF from raw environment data sets. Alternatively, OneSAF also provides a SEDRIS (for terrain skin) and associated shapefiles (for feature data) to OTF compiler to generate OTF files from SEDRIS Transmittal Formatted (STF) files and associated shapefiles.[9] It is envisioned that the OTF provides a starting point for an industry wide simulation terrain format that can be used across virtual and constructive simulation systems. In so doing substantial cost and time savings could be realized in the environmental data production arena.

- **Simulation Object Runtime Database (SORD):** The SORD provides the consistent distributed object database of all OneSAF entities, units, overlays, command and control messages, simulation control interactions, environmental objects, and all associated interactions. The SORD provides a number of interface layers to support rich client-side filtering and access capabilities. At the lowest level there is the Object Database (ODB) that holds the consistent data representation between OneSAF nodes. As mentioned earlier the SORD also provides the data to translators to allow DIS and HLA based external federates interfaces to the OneSAF system. Sitting on top of the ODB are the Object Database Managers (ODM). These managers automatically collect runtime data and provide client-friendly access to the collected data sets. The current set of data managers include: a Command and Control (C2) Manager; Sides and Forces Manager; Entity Manager; Unit Manager; Control Measures Manager; and an Application Manager. Clients can be models, control or display GUIs, reporting GUIs, etc. that access the managers for particular data sets to visual, report, or control the simulated entities. It is envisioned that the SORD provides a starting point for a community wide runtime distributed simulation service infrastructure and would be supportive of a Service Oriented Architecture (SOA) implementation.

- **Entity, Unit, and Behaviour Composer Language:** The composer languages are defined using XML schemas and support their respective composer tools: the entity, unit, and behaviour composers. The Entity Composer allows a user to construct battlespace entities: tanks, aircraft, individual combatants, etc. from models such as turrets, hulls, sensors, and weapons. The Unit Composer allows the user to create hierarchical collections of composed entities. The Behaviour Composer provides the user with a flowchart-like graphical language to describe an entity's or unit's behaviour over time. The advantage of standardizing these languages would allow sharing of entity, unit, and behaviour compositions between composer language compliant systems. These systems may include simulation-based tools or real-world battle command-based tools to populate data sets that could be used by compliant simulations.

3.0 PEO-STRI ENTERPRISE LEVEL COMMON COMPONENT COORDINATION

PEO STRI has executed a series of actions across the past 5 years to promulgate Common Components in pursuit of improved interoperability and reduced life cycle costs. Leveraging the initial activities by the OneSAF and Live Training Transformation (LT2) programs PEO STRI incorporated requirements into Live-Virtual-Constructive interoperability solutions into the formal Charters for all Project Managers. This was reinforced with the release of guidance for the Configuration Management (CM) of Common Component[10], in 2004 and the Policy on the Use of Common Standards, Products, Architectures and/or Repositories (CSPAR)[11], in 2006. Together these lay out a clear roadmap for both new and existing programs at PEO STRI to follow towards common component and standards based solutions.

3.1 DEFINING COMMON COMPONENTS

By simple definition a "Common Component" is one that is used by more than one program. The aggressive reuse approach taken by OneSAF effectively created a set of initial common components within the PEO by adoption of a range of existing components from the WARSIM and CCTT programs. Perhaps even more importantly, PM OneSAF took on design and configuration management responsibilities for these components for the entire PEO. During the first several years formalization of the coordination of efforts was defined by a series of Memorandums of Agreement (MOAs) executed at the Project Manager level. This philosophy was extended PEO wide by the Configuration Management (CM) of Common Component, Reference (1), Standard Operating Procedure (SOP). It established the policy, processes and procedures for CM guidance, and assigns responsibilities for the implementation and standardization of common components within PEO STRI. It directs that PMs shall seek common opportunities across all programs and adopt a policy to reuse existing components rather than rebuild the functionality the component provides. It establishes the objective of a PEO STRI-wide CM policy for the management of common software and hardware components in order to reduce program development and maintenance costs. In order to define common components it specifically established the following process:

a. PMs will identify common component candidates that are being developed within PEO STRI through the Configuration Control Board (CCB). The development of those components shall be managed by the designated project manager in accordance with the CM policies of their Core Business Unit (CBU) and individual program.

b. The CCB will evaluate common component submitted as potential candidates. The components will be assessed for consideration using appropriate criteria and guidelines for initial screening.

c. Upon approval as a common component, the Common Products and Components (CPC) office will facilitate the creation of a MOA between all CBUs. If disapproved, the CPC office documents the rationale and keeps it in the CC repository index for historical purposes. The MOA clearly address each PM's and supporting organizations' roles and responsibilities for that developed, deployed, maintained, and funded Configuration Item (CI) throughout the life cycle of the component.

d. The CPC office will staff the MOA with affected organizational elements to ensure legality and consistency of agreement narratives, with functional/programmatic requirements/alignments to obtain resource data. Finalized version of the MOA with signatures distributed to all affected parties.

e. PMs must perform validation and verification of the developed common component to ensure it achieves its desired outcome.

f. The CCB will establish an approved configuration baseline for the CI and the common component will be delivered to the CPC office who will take configuration management control of it.

g. The developing PM then transitions a baseline version of the common component to CPC for inclusion in the CC repository, where it is managed and updated through a configuration change management process.

3.2 GOVERNING COMPONENT DEVELOPMENT AND EXTENSIONS

Reference (2) the PEO STRI Policy on the Use of Common Standards, Products, Architectures and/or Repositories (CSPAR) defines policy for the designation and use of common products and the identification of communication and interface standards, data models and architectures which facilitate and ultimately reduce the cost of the integration and interoperability of Live, Virtual and Constructive (LVC) capabilities across PEO STRI. This reference document was established by a committee comprised by the Chief Engineers from each of the PEO Project Mangers. It includes a reference set of recommended standards, protocols, components, architectural approaches and data repositories. It also promotes the establishment of a methodology for the consideration of new common capabilities by evaluating:

- What is the expected Return On Investment (ROI) of a common baseline? How is the ROI calculated for legacy programs, programs in the middle of development, and new programs?
- Who manages and develops the common baseline? Where do their responsibilities begin and end?
- How are new requirements generated?
- How are competing and conflicting requirements handled?
- How will open source requirements be handled and how integrating externally developed enhancements be handled?

Once established as a common component it will be configuration managed in accordance with the above processes as it is extended to meet a more diverse set of requirements. It is anticipated that these policies will lead to an eventual establishment of a full set of common products, data and capabilities that will result in a fully interoperable, seamless set of LVC tools for the community.

3.3 COMMON PRODUCT TESTING AND DELIVERY

As defined by the Configuration Management (CM) of Common Component, Reference (1), Standard Operating Procedure (SOP) responsibility for the testing and delivery of common components resides with the Product Manager who executes the development activities. However, coordination with the users of the common components in order to establish the critical test requirements is key to continued acceptance and use of the common component. This is further complicated with OneSAF common components because they are delivered as source code which can and typically is modified by Co-Developers and returned for re-integration and test with the common component. An excellent example of the challenges with this activity can be found in the C4I adapter. After the decision to reuse the WARSIM C4I Adapter on OneSAF a single combined OneSAF/WARSIM adapter team was created. The team initially worked together supporting both program's requirements while maintaining a single software baseline but eventually the teams separated and two separate code bases existed. During this period engineering level discussions and concept sharing continued with a focus towards keeping common functionality and a relatively consistent software base. Over time the baseline being modified by the original WARSIM team was provided back to the OneSAF team for re-integration into the OneSAF maintained "Common Component" baseline. The lessons learned from the interactions on the C4I adapter and Environmental Runtime Component (ERC) components with the WARSIM team led to the establishment of processes that now form the basis for the Co-Developer Handover process used by the entire program.

3.4 EXISTING COMPONENTS

3.4.1 ENVIRONMENTAL RUNTIME COMPONENT (ERC)

The ERC provides the static environmental representation (Land, Sea, Air, Space); coordinate services, data models (shared); runtime compilers; and environmental effects models (NBC, smoke, dust, dynamic terrain/atmosphere, etc.)

In order to optimize software reuse in the implementation of the OneSAF ERC, the government made a decision to leverage the WARSIM Environment implementation. At a top level this approach was intended to mitigate risk, reduce development cost, and enhance capabilities through direct/near direct reuse of the existing medium resolution WARSIM Synthetic Natural Environment (SNE) software.

This approach was expected to:

- Minimize software development and life cycle maintenance costs through reuse of the WARSIM environment software;
- Facilitate attainment of interoperability requirements between the OneSAF, WARSIM, and CCTT environmental runtime components;
- Maximize commonality between OneSAF SNE interfaces and CCTT/OneSAF Testbed Baseline (OTB) interfaces and WARSIM/Joint Simulation System (JSIMS) Dynamic Terrain (DT) interfaces;
- Provide initial terrain databases to support tool kit activities;
- Enhance WARSIM to support High Resolution/Very High Resolution OneSAF requirements through integration of subsystem components;
- Improve interoperability and reduce life cycle maintenance cost through commonality/correlation between frameworks, models and processes; and
- Minimize cost to other STRICOM programs and external Army and Joint Systems including, but not limited to WARSIM/JSIMS and CCTT.

This approach also leveraged the WARSIM Terrain Common Data Model (TCDM) which was extended into the current Environmental Data Model (EDM) and is based on the ISO EDCS. Both the ERC API and the EDM are provided by OneSAF as publicly available documents on our community web site: OneSAF.net.

The ERC has been reused at this point to varying degrees by a number of programs including WARSIM, Synthetic Environment (SE) Core (AVCATT and CCTT), FCS, IWARS, Combat XXI and a number of others. PM OneSAF intends to make this area its next formal proposal into the standards process.

3.4.2 COMMAND AND CONTROL (C2) ADAPTER COMPONENT

The initial objective by PM OneSAF was to leverage the earlier work within the Command, Control, Communications, Computers and Intelligence (C4I) adapter to create a common solution between the WARSIM and OneSAF programs. Achieving this goal will eventually allow the US Army Joint Land Component Constructive Training Capability (JLCCTC) to reduce from the current set of 4 C2 adapters down to just one. The ultimate objective is to integrate the C4I Adapter within other programs with similar C4I interface requirements.

In order for OneSAF to reuse the C4I adapter changes were made to support the interface to the OneSAF simulation runtime data formats and Application Programming Interfaces (API). Modifications were also necessary to access adapter initialization data prior to runtime as the OneSAF initialization process gets its data from the PAIR (Parametric and Initialization Repository) and which differs from the WARSIM process which receives data via Federation Objects. On the C2 device side because OneSAF focuses at interoperating with brigade and below level C2 devices, while WARSIM focuses on Brigade and above, OneSAF required the addition of an Force XXI Battle Command Brigade-and-Below (FBCB2) interface. Between the Simulation data and C2 device interfaces "mappers" bridge a large part of the gap. These are typical of the interface and data differences that must be addressed for successful implementation of common components.

Since the initial establishment of the Adapter as a common component between WARSIM and OneSAF a number of additional programs have begun activities to utilize this software. Notable among those are the Test Communities C3 Driver program, PM CATTS SE Core program and the Future Combat Systems Tactical Network Gateway (TNG). As these initiatives unfold it expected that they will provide additional capabilities to the Adapter both in terms of simulation data and C2 device interfaces. As this effort moves forward PM OneSAF will focus its efforts both on maintaining currency with the unfolding USA Software Blocking (SWB) requirements and continuing general improvements to the initialization and mapper GUIs, modularization of the architecture to facilitate easier reuse and general performance improvements.

3.4.3 AFTER ACTION REPORTING COMPONENT

A range of tools were initially considered for the starting point for the OneSAF AAR. OnesSAF evaluated the CCTT AAR (PowerSTRIPES), WARSIM AAR, ModSAF/OTB, COMBAT XXI, HLA WARRIOR, Vision 21 and Data Collection and After Action Review System (DCARS). The PowerSTRIPES AAR was selected as the starting point for meeting the OneSAF AAR component. The OneSAF requirement was to provide a lightweight AAR, functionally an equivalent to the JANUS embedded AAR. In conjunction with this approach OneSAF established a robust data collection capability and an external AAR data interface to support heavy weight AAR tools. Both the Vision 21 and DCARS teams were engaged in order to ensure that a full set of data requirements were considered. In 2004, the OneSAF AAR was selected by the Future Combat System (FCS) program to be merged with the Live Training Transformation (LT2) AAR into a singular Training Common Component (TCC) AAR. The Joint Forces Command (JFCOM) selected the OneSAF AAR as the basis for its Joint AAR (JAAR). The Joint AAR (JAAR) effort is an initiative by JFCOM to extended, using defined SEI architectural approaches, the OneSAF AAR data collection and fusion systems abstraction layers (SAL) for HLA, DIS, SORD, and XML. This open integration approach provides JFCOM flexibility in approaches for development and deployment. Both the FCS and JAAR efforts have enabled the AAR to provide far more capability than originally envisioned under base program execution while serving as a potential basis for standardization in this area in the future between JFCOM and the US Army.

3.4.4 MILITARY SCENARIO DEVELOPMENT ENVIRONMENT

A range of tools were also initially considered for the starting point for the Scenario Development Environment (MSDE). OnesSAF evaluated the scenario generation capabilities in WARSIM Scenario Generation, ModSAF/OTB, COMBAT XXI, HLA WARRIOR, CCTT Exercise Initialization Tool (CEIT) and the ASTT Model Based Simulation Composition (MBSC). The decision was made that the MSDE work would leverage the work originally done by the CEIT effort. The CEIT tool provided a capability that was independent of the simulation architecture and provided the scenario definition in an XML-like format. PM OneSAF modified the CEIT tool with a focus on the development of an environment that would allow a user to develop a military scenario that is independent of the simulation to be used. The scenario is specified in Military Scenario Definition Language (MSDL). OneSAF converts the MSDL into a simulation specific scenario definition for run-time execution. Initially the MSDE effort focused on extending the capabilities of the CEIT to meet the OOS requirements and maturing the existing XML schema into the current MSDL standard. Since then a range of user driven enhancements have been made. Finally, the MSDE was selected by the FCS program as the basis for their on-board Scenario Development capability.

4.0 INTERNATIONAL COLLABORATION AND DEVELOPMENT

PM OneSAF is actively engaging the international community through collaborative efforts with the United Kingdom (UK) and the Australia & New Zealand, Great Britain, Canada, and America (ABCA) countries, and through international standardization and simulation workshop activities. The following sections highlight some of the OneSAF international standards activities and initial engagements with the NATO modelling and simulation community.

4.1 SIMULATION INTEROPERABILITY STANDARDS ORGANIZATION

OneSAF is working closely with the SISO/IEEE community to develop the MSDL and other related standards. Two are mentioned in more detail here as they are active Product Development Groups within the SISO organization. These efforts are the MSDL and the Coalition-Battle Management Language (C-BML)

4.1.1 MILITARY SCENARIO DEFINITION LANGUAGE PRODUCT DEVELOPMENT GROUP

OneSAF proposed the MSDL to SISO in early 2005 as the subject of a military scenario language Study Group. The Study Group transitioned to Product Development Group (PDG) the following year with SISO acceptance of the MSDL Product Nomination shortly after the 2006 Spring SIW. Since then the PDG has developed a formal specification and revised the MSDL XML Schema to align with the needs of the international community. The PDG on 7 August 2007 voted to present the specification and schema package to the SISO Standards Activity Committee for approval to ballot. Concurrently the PDG initiated a request for balloting to create the balloting pool that will ultimately vote on the MSDL specification package. The balloting pool request will be open for 30 days to ensure an appropriate balloting pool is established. Following this the formal balloting process will begin and is expected to take between 30-60 days. [12]

The Military Scenario Definition Language (MSDL) intends to provide a standard mechanism for loading Military Scenarios independent of the application generating or using the scenario. Standard MSDL is defined utilizing an XML schema thus enabling exchange of all or part of scenarios between (e.g.) Command and Control (C2) planning applications, simulations, and scenario development applications. XML based scenario representations can readily be checked for conformance against the standard's schema. The scope to MSDL is bounded by the situation, defined at one instant in time, combined with the course of action about to be taken in context to that situation. The intent is for MSDL to include that information which is either core or common to the situation and course of action (COA) of a military scenario. Definition of COA falls under the scope of the Coalition Battle Management Language PDG.

4.1.2 COALITION BATTLE MANAGEMENT LANGUAGE PRODUCT DEVELOPMENT GROUP

Prior to MSDL Study Group established PM OneSAF was coordinating with the Army-Battle Management Language prototype effort. The coordination centered on how OneSAF could leverage the BML prototype. At the time, circa 2003-2005, BML was being prototyped as a computer parse-able and understandable language to task subordinates whether they be simulated, human, or robotic with an initial focus at orders generated at a Battalion level. The results of this early coordination was that the BML language once established should allow OneSAF units and entities to be tasked using the BML language. One avenue for passing the tasks to OneSAF would be through MSDL the other is during runtime through the OneSAF C2 Adapter that currently translates real-would message sets USMTF, VMF messages into the C2 component of the OneSAF Runtime Data Model (RDM). This same two-pronged approach remains intact today with the mid-range plan merging the C-BML and MSDL efforts so that the same language set is used for scenario definition battle message syntax and semantics.

C-BML is also being worked under formal Product Development Group within SISO. They are currently working plans for their specification development and are leveraging the JC3IEDM to guide their initial development. As of this date there have been a number of BML efforts and prototypes but the PDG is still in the process of coalescing all of the efforts and developing the initial specification.

In generally, the C-BML will specify an unambiguous language supporting of all battlespace domains to command and control real, simulated, or robotic forces and equipment conducting military operations; and to pprovide for situational awareness and a shared, common operational picture. For more information on the C-BML activities please see: www.sisostds.org/index.php?tg=articles&idx=More&article=439&topics=102. [13]

4.2 NORTH ATLANTIC TREATY ORGANIZATION

This paper represents one in a series of initial engagements with the NATO M&S WG community in order to provide information concerning the OneSAF program. In particular, we also want to highlight that one of our Co-Developer sites (Aviation and Missile Research and Development Engineering Center (AMRDEC) briefed the OneSAF Conceptual Modeling (CM) approach as the US input to the recent MSG-058 forum.

5.0 CONCLUSION

This paper was intended to show that the OneSAF program has made demonstrable progress towards its original MNS goals of being a catalyst for transformation of the US Army Modeling and Simulation community through its embracement of data and product reuse, standards and an open source business model. The CM brief at MSG-058 (reflection by one of our users that the OneSAF CM process has value), our efforts with MSDL and this paper are all reflections that OneSAF will provide value beyond the base program execution. It also potentially serves as an indicator that OneSAF's work may help serve as a transformation trigger within the larger community. Finally, PM OneSAF hopes both that our efforts will contribute positively to the general body of M&S knowledge and provide a solid basis to support any evaluation of the potential use of OneSAF to meet future NATO objectives.

6.0 REFERENCES

[1] "One Semi-Automated Forces (OneSAF) Mission Needs Statement (MNS)", approved May 1997.

[2] OneSAF - http://www.onesaf.net

[3] OneSAF Architecture and Integration Task Order; CDRL No.: A002. OneSAF Product Line Architecture Framework Specification Vol I; 28 Feb 2006.

[4] Simulation Interoperability Workshop (SIW) – http://www.sisostds.org/index.php?tg=articles&idx=More&article=18&topics=8

[5] European - SIW – http://www.sisostds.org/index.php?tg=articles&idx=More&article=19&topics=9

[6] SEDRIS - http://www.sedris.org

[7] Multilateral Interoperability Programme – http://www.mip-site.org/publicsite

[8] World-Wide-Web (W3) Consortium (W3C) - http://www.w3c.org

[9] Environmental System Research Institute, Inc. ESRI Shapefile Technical Description, An ESRI White Paper – July 1998

[10] SFAE-STRI-PM FF(S)-CPC 18 October 2004; Standard Operating Procedure (SOP) No. 70-3-3 (Interim NTE 18 Apr 05); Subject Configuration Management (CM) of Common Components

[11] SFAE-STRI-PS-E 16 0ct 2006; Memorandum for PEO STRI Project Managers, Executives, and Project Directors; Subject: PEO STRI Policy on the Use of Common Standards, Products, Architectures and/or Repositories (CSPAR)

[12] SISO Military Scenario Definition Language Product Development Group: http://www.sisostds.org/index.php?tg=articles&idx=More&article=440&topics=103

[13] SISO Coalition-Battle Management Language Product Development Group: http://www.sisostds.org/index.php?tg=articles&idx=More&article=439&topics=102