One Semi-Automated Forces (OneSAF)

Capabilities, architecture, and processes

John Logsdon, Deputy PM, OneSAF
Dave Nash, Ph.D.
Mike Barnes
### Report Documentation Page

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

<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 MAR 2008</td>
<td>N/A</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
<th>5a. CONTRACT NUMBER</th>
<th>5b. GRANT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>One Semi-Automated Forces (OneSAF) Capabilities, architecture, and processes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(S)</th>
<th>5c. PROGRAM ELEMENT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</th>
<th>8. PERFORMING ORGANIZATION REPORT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRADOC Analysis Center</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</th>
<th>10. SPONSOR/MONITOR’S ACRONYM(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>12. DISTRIBUTION/AVAILABILITY STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved for public release, distribution unlimited</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13. SUPPLEMENTARY NOTES</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>14. ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15. SUBJECT TERMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
<th>17. LIMITATION OF ABSTRACT</th>
<th>18. NUMBER OF PAGES</th>
<th>19a. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT</td>
<td>b. ABSTRACT</td>
<td>c. THIS PAGE</td>
<td></td>
</tr>
<tr>
<td>unclassified</td>
<td>unclassified</td>
<td>unclassified</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UU</td>
<td>44</td>
<td></td>
</tr>
</tbody>
</table>

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18
User communities

OneSAF is chartered to serve three communities, each of which has its own requirements and priorities:

- Advanced Concepts and Requirements (ACR)
- Research, Development and Acquisition (RDA)
- Training, Exercise and Military Operations (TEMO)

In addition to these, PM OneSAF also directs support that is to be provided to various joint and civilian organizations, and international customers.
The Simulation Lifecycle

1: Knowledge Acquisition / Engineering
2: Product Line Development
3: Product Line Deployment & Install
4: Event Planning
5: Database Development
6: Software Development
7: Model Composition
8: Scenario Generation
9: Simulation Configuration
10: Systems Test & Verification
11: Simulation Execution
12: Post-Execution Analysis/AAR
13: Archival
14: Retrieval
15: Product Line Maintenance

* The lifecycle phases do not intend to restrict the use of common tools across many phases. Users can transcend phases.

Legend:
- 1-3, 15: Product Line Development, Deployment & Maintenance phases
- 4-8: Pre-Event phases
- 9-11: Event phases
- 12-14: Post-Event phases

* Iteration back and forth between phases not shown
Composable Product Line Concept

- OneSAF is not a single system or product.
- OneSAF is a set of:
  - Components,
  - Products comprising OneSAF components,
  - System compositions comprising OneSAF products
- Tools are provided to create compositions of various types (including system compositions).
The Product Line Architecture Framework

### Architectural Applications (OneSAF System Compositions)
- Leader and Staff Training System Composition
- Seamless Training System Composition
- Force and Organizations Analysis Tool System Composition
- Test and Evaluation System Composition
- Other System Compositions...

### OneSAF Product Layer
- System Composer
- Knowledge Eng. Env.
- Event Planner
- Model Composer
- Simulation Generator
- Technical Manager
- Simulation Core
- Simulation Controller
- C4I Adapter
- Analysis & Review
- Repository Manager
- Maintenance Environment

### OneSAF Component Layer
- System Composer Tool
- KA/KE Tools
- Military Scenario Development Environment
- Unit Composer
- Entity Composer
- Behavior Composer
- Environment Composer
- Icon Tool
- Battlefield Enum. Tool
- Management & Control Tool (SSDE)
- Data Collection Specification Tool
- Sim. Config. & Asset Mgmt. Tool
- Federation Develop. Tool
- Performance Modeling Tool
- Network Loader Tool
- Benchmark Tool
- Unit Models
- Entity Models
- Behavior Models
- Physical Models
- Environment Models
- Environment Database Generation Environment
- Simulation Core
- Simulation Controller
- C4I Adapter
- Analysis & Review
- Repository Manager
- Maintenance Environment

### OneSAF Component Support Layer
- Composition Services
- Environment Runtime Services
- Environment Reasoning Services
- GUI Services
- Plan View Display
- Data Collection Services
- Simulation Services
- Simulation Object Runtime Database
- Modeling Services
- System Repository Services

### OneSAF Repository Component Layer
- KA/KE Repository
- Environment Repository
- Software Repository
- System Composition Repository
- Military Scenario Repository
- Local Exercise Environment Repository
- Parametric & Initialization Repository
- Simulation Output Repository

### OneSAF Common Services Layer
- Monitor Services
- Time Services
- Name Directory Services
- Messaging Services
- Coordinate Services
- Interchange Services
- RTI
- DIS
- COE Services
- WWW
- JDBC/ODBC
- ORB
- Live Range Adapter

### OneSAF Platform Layer
- Hardware
- Operating System
- Network
MSDE / AAR

- **Military Scenario Definition Environment (MSDE)**
  - PowerPoint-based
  - Creates scenario definition (units, command relationships, dispositions) in Military Scenario Definition Language (MSDL)
  - MSDL is imported into a OneSAF scenario for execution

- **After-action Review (AAR) Tool**
  - Generates slides of the simulated scenario
  - Statistics of “snapshot” data items
Composability

• Composability:
  – The ability to assemble reusable software elements ("components") into customized software.
  – A central design theme of OneSAF.

• Fairly recent capability in software development enabled by:
  – increased computing power,
  – success in developing reusable software libraries, and
  – the development and distribution of software and software components over the internet to different computer systems.

• Instead of developing complete computer programs for each computer system, software developers develop platform independent components to be assembled on the user’s computer.
Composition nomenclature

- Use of the term "composition" is ubiquitous in OneSAF, and usually requires context to know exactly what is being described:
  - System composition: a JAR file consisting of a set of system components
  - Component composition: an XML file containing pointers to various Java classes; used to imbue actors with capabilities such as:
    - Vulnerability, mobility, and sensing (physical components)
    - Responsiveness to orders, ability to process fire missions (behavioral components)
  - Actor compositions:
    - Entity composition: an XML file containing metadata and pointers to component compositions defining a single, indivisible battlespace object
    - Unit composition: an XML file containing metadata and pointers to component compositions defining a single, indivisible battlespace object
  - Behavior composition: an XML file containing metadata, inputs, (in the case of composite behaviors) outputs, and control definitions defining the effect of a behavior
OneSAF System Components

- A OneSAF system composition is an assembly of system components. In some ways a system composition is analogous to an executable program.
- A system component is a modular software artifact that makes specific functionality available to the system composition.
- Some examples:
  - The Formation Editor is a system component providing a graphical tool that may be used to interactively design formation data files.
  - The formationEditor system composition consists of the Formation Editor and the Desktop components.
- OneSAF components are JavaBeans with extensions; e.g.:
  - OneSAF specific component metadata such as releasability, dependencies, and fidelity
  - Non-Java code support to allow reuse of code written in other languages.
- Components can contain any subset of:
  - Data
  - Software source code
  - Software executable code
  - Documentation
System Composer

- The System Composer is a graphical tool used to create OneSAF system compositions:
  - The System Composer is based on JavaBean application builders but works with OneSAF extended JavaBeans (OneSAF system components).
  - The System Composer creates a JAR file for each composition (the composition and JAR file have the same name) containing:
    - Information the runtimeloader script uses to run the composition.
    - Composition metadata; e.g.,
      - Description
      - Specification and Implementation Version
      - Validation Authority and Date
  - The System Composer provides the software “glue” connecting components.
- It is executed from the command prompt by running the composer script within SWR/bin/services/sys/composition
  - This is the only stand-alone application within OneSAF; all other applications are started using the runtimeloader script in the same directory.
System Composer
Entity Composer
Using the Entity Composer

• System composition = composers, therefore $BINDIR/runtimeloader composers
• Typical components
  – Physical
    • Sensors (eyeballs, FLIR, etc.)
    • Weapons
    • Mobility
    • Vulnerability
  – Behavior
    • Direct fire controller
    • Operations
    • Intel
    • Supply
Unit Composer
Behavior Composer
The Management and Control Tool
Sides and Forces

• A force is a collection of entities, units, or forces.
• A side is a collection of entities, units, forces, or sides.
• Sides have relationships which determine certain behaviors.
Variable fidelity

• Support for variable levels of fidelity makes it possible to tailor the simulation in order to maximize satisfaction of diverse use cases.

• Accurately and effectively represents activities within the Army warfighting functions:
  
  • Intelligence
  • Movement and maneuver
  • Fire support
  • Protection
  • Sustainment
  • Command and control
Environmental Runtime Component (ERC)

- Obscurants / contaminants
- Features
- Weather
- Terrain
- Line of Sight
- Illumination
Distributed Simulation

- OneSAF is a distributed simulation system.
- Workstations may be assigned different computational roles, such as:
  - Simulation core (SimCore)
  - Management and Control Tool interface
  - C4I adapter
- Distributed design allows:
  - Individual workstations to balance loads
  - System recovery of individual workstations without interrupting a simulation in progress
Interoperability

- OneSAF instances interoperate via:
  - Internal protocols
    - OneSAF messaging services
    - C4I Adapter
  - External protocols
    - HLA
    - DIS
Overview Summary

• OneSAF is a highly configurable suite of systems for the generation of semi-autonomous forces.

• Simulation behaviors and models are data driven to the maximum extent possible, to support customization and “what-if” analysis.

• System capabilities are exposed by powerful visual tools.

• High degree of interoperability through DIS and HLA, and usage of common formats including XML for state.
Introduction to the OneSAF architecture
The OneSAF Agent Architecture

Battlespace Participant
a.k.a. Actor

Actor Software Component

Blackboard

Modeling Infrastructure

Posted to Blackboard on Initialization

Physical Capability

Trigger

Behavior Agents (Controller)

1..*

Behavior Models

1..*

Physical Models

Data

use

Discover method invocation

Data

use

Composite Behaviors

Primitive Behaviors

Actors use & are controlled by

Unit

Entity


 use

Triggers

1

DMSC-24
Battlespace participants (actors)

Battlespace Participant a.k.a. Actor

- Unit
- Entity

Actor Software Component

Blackboard

1..* Triggers

Physical Capability

Modeling Infrastructure

Posted to Blackboard on Initialization

Composite Behaviors

Primitive Behaviors

Triggers

Behavior Models

Physical Models

Actors use & are controlled by

Data

Discovered method invocation

use

use

use

use
Battlespace participants (actors)

• Actor:
  – An entity or unit
  – A simulated thing that can be instantiated in a battlespace and has a location

• Entity:
  – The smallest discrete, stand-alone actor.
  – Implemented as a composition of physical components and behavioral components.

• Unit:
  – An organized collection of actors and their capabilities.
    – Alternatively … “a collection of actors”
  – A simulation object representing the combined command and control of a collection of subordinate actors (entities and sub-units).
    – Alternatively … “a collection of components”
Components

Battlespace Participant a.k.a. Actor

Actor Software Component

Blackboard

Behavior Models

Physical Models

Triggers

Data

Composite Behaviors

Primitive Behaviors

Unit

Entity

Actors use & are controlled by

Physical Capability

Posted to Blackboard on Initialization

Discovered method invocation

Use

1..*

1

1..*

1..*

Behavior Agents (Controller)

Physical Agents (Controller)

Data

Components
Components

• Agents:
  – Behavioral agents provide command and control capabilities, such as planning, plan execution, and situation assessment.
  – Physical agents are the “middlemen” between behaviors, the physical world, and physical models.

• Models:
  – Behavioral models answer behavior agents’ questions and represent the reasoning of agents.
  – Physical models provide physical capabilities, such as mobility, weapons, vulnerability, sensing, and communications. They represent the effectors and perceptors of simulated platforms and the physics of the simulated world.
Example Agents

- **Behavioral Agents:**
  - Command
  - Scheduler
  - Intel
  - Message
  - Operations
  - Driver
  - Direct Fire Weapons Controller
  - Fire Direction Center
  - Caller For Fire
  - ADA Target Handoff

- **Physical Agents:**
  - Weapon
  - Radio
  - Sensor
  - Mobility
  - Transport
  - Vulnerability
Behaviors

Battlespace Participant a.k.a. Actor

Actors use & are controlled by

Composite Behaviors

Primitive Behaviors

use

Data

Actor Software Component

1..*

Blackboard

 posted to Blackboard on Initialization

Triggers

Behavior Models

1..*

use

Data

Physical Capability

use

Data

Discovered method invocation

DMSC-30
Composite behaviors

- Composite Behaviors:
  - Represent tasks and missions
  - Are composed of primitive and other composite behaviors
  - Are created with the Behavior Composer.

- Temporal organization:
  - Sequential
  - Parallel

- Conditional Branching

- Looping
Primitive behaviors

• Primitive behaviors:
  – Provide chunks of functionality from which more complex behavior models are built.
  – Are parameterized with inputs, and may have outputs
  – Interact with behavioral agents.

• Are implemented as Java classes, with an XML wrapper to allow composition
**State transitions**

- **Battlespace Participant (a.k.a. Actor)**
  - **Unit**
  - **Entity**
  - **Composite Behaviors**
    - **Primitive Behaviors**
      - **Data**
  - **Trigger**
    - **Physical Models**
      - **Behavior Models**
        - **Behavior Agents (Controller)**
        - **Physical Agents (Controller)**
          - **Physical Capability**
    - **Triggers**
      - **Actor Software Component**

- **Blackboard**
  - 1..* (1..* indicates a many-to-one relationship)

- **Posted to Blackboard on Initialization**

- **Modeling Infrastructure**

- **Actors use & are controlled by**

**Discovered method invocation**
Content carriers

• Event
  – Information object that is sent from an actor to the external world over the network.
  – Events are “sent” on the EventRouter.
  – Also referred to as a “Simulation Event.”

• Trigger
  – Information object sent internal to an actor. May be sent/received by agents and behaviors.
  – Triggers are “posted” on the blackboard.
  – Also referred to as a “Command.”

• Directive
  – A special event that can also function as a trigger.
  – The Directive may be “sent” or “posted.” When it is received as an event, it will automatically be posted as a trigger on the blackboard.
  – Also referred to as an “Intervention” (on the MCT).

• Fact
  – Information object that is stored in the blackboard.
  – Facts may be simple “Information Facts” that store information or they may be “Fuzzy Facts” that can evaluate its stored information and modify it over time.

• Message
  – A class of events used by the communication framework.
The Blackboard

- The Blackboard is an Agent control mechanism:
  - Agents subscribe to triggers on the blackboard.
  - When the blackboard receives a trigger, it notifies that trigger’s subscribers (and only those subscribers).

- Blackboard:
  - “World Model” for an actor containing perceived truth.
    - Send and receive (internal) triggers
    - Contains facts
    - Access physical capabilities
  - One instance per actor; cannot be shared with other actors.
OneSAF processes
Capability Development
Process Flow

Requirements

Systems Engineering

Conceptual Modeling
Knowledge Engineering

PLRS

Implementation
Software Development

Integration & Test

A&I Baseline

Build k
Build k+1
Build k+2

Version 2.1

Change Request

Requirements

Systems Engineering

Conceptual Modeling
Knowledge Engineering

PLRS

Implementation
Software Development

Integration & Test

A&I Baseline

Version 2.1

Change Request

DMSC-37
Simulating the real world is difficult and error prone.

A well-conceived, consistent intermediate model eliminates many problems by providing a model of the battlespace usable by all participants (customer, domain expert, developer, and user).
Conceptual Modeling and Knowledge Engineering Process

Main products:
- Capability Description Document (CDD)
- Domain Behavior Description (DBD)
- Physical Knowledge Acquisition Document (PKAD)

Activities

- SE/CM/KE kickoff
- Publish draft DBD
- Review draft DBD
- Publish refined CDD + DBD
- Review refined CDD + DBD

Build week
Implementation Build

Conceptual Modeling
Knowledge Engineering

- CDD
- DBD
- PKAD
- RIB

Systems Engineering

RA
2 Weeks

- Update CDD
- SRS-PLRS
- Mapping
- RA Review

Design
2 Weeks

- Change Request updated
- UML Diagrams
- GUI Mock-ups
- Peer Design Review
- TPO Design Review

CUT
3 Weeks

- Code
- Composites
- CUT Review

SWIT
3 Weeks

- Test Thread
- Test Thread Scenarios
- SWIT Review
- Create or Update DocBooks
- Documentation

Integration & Test
Co-developer handover process
General handover criteria

- **Integration with OneSAF**: It is expected that handovers are submitted for the purpose of integration into OneSAF, and that they are ready for integration.
- **Development baseline**: Capabilities must be developed against the most recent generally available release.
- **Architectural compliance**: Handover products must comply with architectural and design guidelines, protocols, language compatibilities, code structures, and documentation standards.
- **Testing**: Capabilities must be tested during development to ensure that they operate as designed are not injurious OneSAF.
- **Classification**: At present, all handover products must be unclassified.
Co-developer handover process

- To be considered for inclusion with a baseline release, co-developers participate in several activities:
  - Registration (co-developer)
  - Initial review (OneSAF Architecture and Integration)
  - Handover package submission (co-developer)
  - Engineering Change Control Board review
  - Peer review (OneSAF Architecture and Integration)
  - Baseline integration and test (OneSAF Architecture and Integration, with co-developer participation desired)

- Required artifacts:
  - Source code and data
  - Design documentation (use cases, sequence and class diagrams)
  - Domain documentation (PKAD / DBD / CDD)
  - Usage instructions
  - Unit tests and system test cases
  - Architectural compliance assessment
Questions?