Evolution of the Standard Simulation Architecture

Dr. Jeffrey S. Steinman
Chief Science and Research Officer
RAM Laboratories, Inc.

Douglas R. Hardy
Scientist
SPAWAR Systems Center, San Diego
Benefits of a Standard Simulation Architecture

- **Reduce software development cost**
  - Modeling tools, constructs, and support utilities
  - Improve reliability by reducing code written by developers

- **Facilitate interoperability, composability, and reuse**
  - Entity and component repositories with object composition tools
  - Abstract interfaces decouple software implementations
  - Layered architecture supports technology insertion

- **Provide high performance**
  - Scalable parallel and distributed computing

- **Maximize configuration flexibility**
  - Flexible assignment of models to hardware platforms
Historical Evolution of the Standard Simulation Architecture

Late 1980’s
- SIMNET
- JTC
- TWOS

Early 1990’s
- DIS
- ALSP
- SPEEDES

Late 1990’s
- HLA
- SMF DSMS

Today
- HLA
- SSA
- Entity & Component Repository

Simulation Technology
Government & industry partnership
- Investments made by both government and industry
- SPEEDES, CCSE, WarpIV

Layered architecture supports inclusive development
- Academic R&D
- Research laboratories
- Multiple industry technology vendors

Success requires government participation
- Standards organizations
- Government programs
Layered Architecture: The Standard Simulation Architecture

<table>
<thead>
<tr>
<th>Direct Federate</th>
<th>Abstract Federate</th>
<th>HLA Federate</th>
<th>HLA Federation</th>
<th>External System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CASE Tools</td>
<td>HPC-RTI</td>
<td>HLA Gateway</td>
<td>External HLA Modeling Framework</td>
</tr>
</tbody>
</table>

- **Entity Repository**
- **Component Repository**

- SOM/FOM Translation Services
- Distributed Simulation Management Services
- Standard Modeling Framework
- Time Management
- Event Management Services
- Standard Template Library
- Persistence
- Rollback Utilities
- Rollback Framework

<table>
<thead>
<tr>
<th>Internal High Speed Communications</th>
<th>External Distributed Communications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Communications</td>
<td></td>
</tr>
<tr>
<td>Utilities</td>
<td></td>
</tr>
<tr>
<td>Threads</td>
<td></td>
</tr>
<tr>
<td>System Services</td>
<td></td>
</tr>
</tbody>
</table>
Interoperability and the Standard Simulation Architecture

- **Network Communications**
  - SSA Federates
  - HPC-RTI Federates
  - Parallel SSA Federate
  - Sequential SSA Federate
  - Any HLA Federate

- **HLA RTI**

High Performance Interoperability with Legacy Systems

Models Interoperate

Legacy Models ↔ New Models

Federate HPC RTI

SSA SimEngine

High-Speed Communication

Federate SSA SimEngine HPC RTI

Events processed by both engines

Optimistic

Conservative
Internal High Speed Communication is Critical for SSA Performance

HP - 1 Byte Message Throughput

Number of Messages Per Second

Number of Nodes

AsyncUnicast Messages/Sec, AsyncMulticast Messages/Sec, AsyncBroadcast Messages/Sec, CoordUnicast Messages/Sec, CoordMulticast Messages/Sec, CoordBroadcast Messages/Sec
Internal High Speed Communication is Critical for SSA Performance

HP - 64 Kbyte Message Bandwidth

Number of Bytes Per Second

Number of Nodes

AsyncUnicast Bandwidth
AsyncMulticast Bandwidth
AsyncBroadcast Bandwidth
CoordUnicast Bandwidth
CoordMulticast Bandwidth
CoordBroadcast Bandwidth
System Composability and IPC Overheads

<table>
<thead>
<tr>
<th>Communication Unit</th>
<th>IPC Mechanism</th>
<th>Granularity (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLA Federate</td>
<td>RTI, CORBA</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>SSA Federate</td>
<td>TCP/IP LAN/WAN</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>Machine</td>
<td>Beowulf Cluster</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>Node</td>
<td>Shared Memory</td>
<td>$10^{-6}$</td>
</tr>
<tr>
<td>Thread</td>
<td>POSIX Threads</td>
<td>$10^{-8}$</td>
</tr>
</tbody>
</table>
Model Composability

- UML: Type Of
- UML: Composed Of

HLA Federation

- HLA Federate

SSA Federation

- SSA Federate

- HPC-RTI Federate

Models

- Entity

Sub-models

- Component

Sub-sub-models

- Component

Publish local FOs
Subscribe to remote FOs
Interoperability

- HLA provides standards for interoperability between multiple simulations
  - Coarse grained interoperability

- The Standard Simulation Architecture provides standards for interoperability for models at three levels
  - Between **Federates** within an HLA federation
  - Between **Entities** within a (parallel or sequential) simulation
  - Between **Components** within an entity

- The Standard Simulation Architecture also provides standards for technology insertion
  - Layered architecture compartmentalizes functionality
Architecture Rules for Model Interoperability

- Must preserve the abstraction that an entity may reside on any node when running in parallel, or within any federate when executing in an HLA federation
  - Entity state exchanged with other entities must be provided exclusively through **Federation Objects**
  - Entities interact with other entities exclusively through HLA-style **Interactions**

- **Entities behave like miniature federates…**
  - DSMS Layer provides HLA functionality between entities
  - Operator overloading in C++ automates distribution of attributes
  - Interest management automatically operates on attributes
Hierarchical Composability

**Entities contain Components**

- SimObj
- Entity
- Model
- Root FoMgr

**Components contain other Components**

- Component
- SubModelA
- SubModelB

*Hierarchical FoMgrs*
Federation Object Management

- **Interest Management is automatically provided between Entities**
  - Filtering is automatically performed on attributes as they change
  - Hierarchical grids supports multi-resolution scalability in parallel

- **Interest Management is automatically provided between Components**
  - FoMgrs filter FOs based on Component subscriptions
  - Special component for range-based filtering
Entity1 sends a Detonation Interaction

**Entity1**

**DSMS**

**FedGateway**

**Entity2**

**Detonation Interaction sent through RTI**

**RTI**

**Entity3**

**DSMS**

**FedGateway**

**Entity4**
Double Abstraction Barrier Principle

- **Interactions between HLA Federates (~milliseconds)**
  - Federates do not know which other federates have subscribed
  - Federates do not know how interaction is processed

- **Interactions between Entities within an SSA federate (~microseconds)**
  - Entities do not know which other entities have subscribed
  - Entities do not know how interaction is processed

- **Polymorphic methods between Components within an SSA entity (~nanoseconds)**
  - Components do not know which classes have registered
  - Components do not know which methods are registered
The **Process Detections** polymorphic function allows the Radar Scan process to invoke the **Fuse Detections** polymorphic method of the Track Fusion component without requiring access to its pointer.

- **Register the Fuse Detections method as a polymorphic method**
- **Double-Abstraction Barrier**
- **Call polymorphic Process Detections function**
- **Invokes polymorphic Fuse Detections method**
Standardization Encourages Three Business Models

- **Government Off The Shelf (GOTS)**
  - Development by government laboratories
  - Government provides life-cycle maintenance

- **Open Source**
  - R&D by research institutions and universities
  - Successful R&D feeds into real programs

- **Commercial Off The Shelf (COTS)**
  - Development by industry
  - Users buy software licenses with support contracts
Standardization Steps

- **Government sponsorship and oversight**
  - Establish SISO working group to study standards issues
  - Management of the standardization process
  - Requires appropriate level of funding and commitment

- **Architecture Participants**
  - **Engineering Team** comprised of proven simulation technologists to refine standards
    - Industry, Government, and University
    - Prototype standardized interfaces and services
    - Joint development of unit and system test suites
  - **Technology Panel** of specialists review individual layers
  - **User Group** generates feedback on services
Summary and Conclusions

- The Standard Simulation Architecture addresses critical needs of DoD simulation community
  - Interoperability between federates, entities, and components
  - Facilitates object composability
  - Layered architecture promotes technology infusion
  - High performance computing
  - Portability and flexibility
  - Reduces software development costs while improving reliability

- Requires government sponsorship and oversight
  - Commitment to standardize and implement the SSA layers
  - Programs must focus on model and component reuse
  - COTS, GOTS, and Open Source business models for technology insertion
WarpIV Simulation Kernel

- WarpIV provides prototype development of the SSA

- RAM Laboratories is currently offering WarpIV to:
  - Universities
  - Research Laboratories
  - Government Programs
  - Industry

- For more information about WarpIV, see our website:
  - http://www.ramlabs.com