The Challenges of Applying Service Orientation to the U.S. Army's Live Training Software Product Line

Jeremy Lanman, Ph.D.
U.S. Army PEO STRI
12350 Research Parkway
Orlando, FL 32826
+1 407 384 5307
Jeremy.Lanman@us.army.mil

Rowland Darbin
Jorge Rivera
General Dynamics
12001 Research Parkway, Suite 500,
Orlando, FL 32826
+1 407 275 4820
Rowland.Darbin@gdc4s.com
jorge.rivera@gdc4s.com

Paul Clements, Ph.D.
Charles Krueger, Ph.D.
BigLever Software
10500 Laurel Hill Cove
Austin, Texas 78730 USA
+1 512 426 2227
pclements@biglever.com
ckrueger@biglever.com

GENERAL DYNAMICS
C4 Systems

Approved for public release, distribution is unlimited
1st Generation SPL Management Component Reuse

**Challenges**
- Control
- Increased Costs
- Recurring Efforts
- Lost Bug-Fixes
- Product Quality
- Loss of Trust
- Customer Resistance to New Releases

**LT2 Perspective**
- 80+ Components
- 2.8M Lines of Code

**Reuse Results in Complex CM Branching and Merging**

- Example Component e.g. AAR
- Product Teams Pull Components & Add Features
- Site Specific Tailoring Increases Variations
- Field Bug Fixes Further Complicate The Issue
- DRTS
- Lt2 Portal
- Unmanageable Fix & Feature Merges

---

Approved for public release, distribution is unlimited
2nd Generation LT2 Product line Management

Results

- Configuration Control
- Variation Management
- Assets Leveraged by Programs
- Improved Product Quality
- Increased Customer Trust

To Date Over $350M Cost Avoidance
Why SOA

• Increase ROI for LT2
  – Needs of LT2 users are changing
  – Army is moving toward Cloud and Mobile-based technologies

• Training as a Service (TaaS)
  – “Always On” training capabilities

• Align development and technology with customer goals
### Apply SOA design principles to meet **strategic** goals

<table>
<thead>
<tr>
<th>Design Principle</th>
<th>Benefit to Training Community</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standardized Service Contracts</td>
<td><strong>Easier</strong> to integrate services developed by others</td>
</tr>
<tr>
<td>Service Composability</td>
<td>Create new capabilities from reusable services</td>
</tr>
<tr>
<td>Service Loose Coupling</td>
<td><strong>Easier</strong> to deploy new capabilities (less dependencies)</td>
</tr>
<tr>
<td>Service Abstraction</td>
<td><strong>Easier</strong> to reuse services (supports loose coupling)</td>
</tr>
<tr>
<td>Service Reusability</td>
<td>Reduces redundant implementations of same service logic</td>
</tr>
<tr>
<td>Service Autonomy</td>
<td><strong>Improved</strong> Reliability, Availability and Maintainability (RAM)</td>
</tr>
<tr>
<td>Service Statelessness</td>
<td><strong>Improved</strong> availability/scalability, supports composability</td>
</tr>
<tr>
<td>Service Discoverability</td>
<td><strong>Easier</strong> to find services developed by others</td>
</tr>
</tbody>
</table>

### Strategic Goals

- Increased Federation
- Increased Intrinsic Interoperability
- Increased Vendor Diversification
- Increased Business and Technology Domain Alignment

### Strategic Benefits

- Increased ROI
- Increased Organizational Agility
- Reduced IT Burden
Using loosely coupled, well defined services that address the business goals of the system. Services are composed into orchestrations that change to meet the future needs of the live training community.

<table>
<thead>
<tr>
<th>Capability</th>
<th>CTIA Current</th>
<th>CTIA 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>Monolithic Middleware</td>
<td>Scalable Service Oriented</td>
</tr>
<tr>
<td>Composability</td>
<td>Plug-and-Play Components</td>
<td>Composable Services</td>
</tr>
<tr>
<td>Service Interface</td>
<td>Dedicated IDL APIs</td>
<td>Reusable Service Contract</td>
</tr>
<tr>
<td>Usability</td>
<td>Thick Applications</td>
<td>Mobile Enabled Web Clients</td>
</tr>
<tr>
<td>Reusability</td>
<td>Component Level</td>
<td>Service Level</td>
</tr>
<tr>
<td>Business Logic</td>
<td>Static; Tightly Coupled to Components</td>
<td>Composition driven; Defined by Consumer</td>
</tr>
<tr>
<td>Integration</td>
<td>Static</td>
<td>Composable</td>
</tr>
<tr>
<td>Configuration Management</td>
<td>Single, Monolithic Release</td>
<td>Individual Services</td>
</tr>
<tr>
<td>Governance</td>
<td>Component Agreement</td>
<td>Service Contract &amp; Service Agreement</td>
</tr>
<tr>
<td>Backwards Compatibility</td>
<td>Dependent on Middleware; Compile-time only</td>
<td>Service Mediation</td>
</tr>
<tr>
<td>Testing</td>
<td>Unit / Component / System</td>
<td>Service / System, Automation, Coverage</td>
</tr>
</tbody>
</table>

Aligning technology to business goals – Leveraging SOA methodology
Challenges to adopting SOA

• Ensuring reuse
  – How do we meet current capabilities in a SOA environment
  – Define autonomous services

• Business process
  – CITA doesn’t follow archetypal service orchestrations
  – Phases of system use changes active service inventory

• Deployment / Cycle Time
  – Periods of high use with no change
  – Moving to cloud for always on capability

• Security
  – Changing interfaces is expensive
  – System boundaries are not flexible

• Technical Concerns
  – TPMs and performance constraints
Challenges managing a SOA Architecture

Lack of governance is the number one reason for SOA failure in organizations

Managing your SOA in a Product Line

- Re-using existing Product Line processes
  - Variation management
  - Common baseline
- Using LT2 Portal for Service Inventory and static discovery
- Community approves changes

Product line governance process didn’t change due to SOA
Product alignment development strategy

CTIA Transition Architecture 1
Tracking Prototype

CTIA Transition Architecture 2
Full Soldier Tracking Use Case

CTIA Transition Architecture 3
Full TRACR Use Case

CTIA Transition Architecture 4
Full HITS Use Case

CTIA Transition Architecture 5
Supports Regional Training Center Use Case

SOA Infrastructure
Web-Based 2D Map
Participant Definition Tool
Tablet Support
Alerts, Graphics, Battle Roster, Instrumentation
Engagements, Targets, Exercise Planning, Complete 2D Map
Tactical Voice, Replay, & AAR
Supports full CTC-IS, Distributed Training, Deployable SOI
Current Transition Architecture

CTIA 3.0 and LT2 Components

CTIA 3.0-4.0 Gateway

Damage State Controller Instrumentation Controller TSPI Source Engagement Source

Player Unit Controller Services

Mux

Player Units

Services

Geoloc
DIS Enum
Entity Org
Exercise
Runtime Persistence
Rule Engine
Damage State Coord
Graphics
Exercise Element
Rule Notification
Battle Roster

GUI Components

RESTEasy
OWF
Smart GWT

JBoss AS

AMQP Messaging
PostgreSQL

TIGR Services

T1 Service
T2 Service

Player Unit Gateways

Graphics
Battle Roster

JBoss AS

Entity Org
Exercise
Rule Engine

AMQP Messaging
PostgreSQL

TIGR Services

T1 Service
T2 Service

Player Units

Mux

Player Unit Gateways

Current Transition Architecture

Approved for public release, distribution is unlimited
Conclusions

- LT2 is a proven and mature product line with over $350M in cost savings already realized
- An established software product line aligns with SOA migration
- Major barriers to managing SOA are already solved in a software product line
- Supports product needs and technology evolution