Orchestration Research and Assessment at the Naval Research Laboratory

Alex Neihaus, VP Marketing, Active Endpoints
Dr. James Ballas, Ph. D., Naval Research Laboratory
Justin Nevitt, Naval Research Laboratory
Rick Rosenberg, President & CEO, Seros, Inc.
Welcome and Agenda – Alex

Orchestration at the Naval Research Laboratory – Jim and Justin

Seros, Inc. – Rick

ActiveVOS BPMS – Alex

Q&A/Panel discussion – All
Jim Ballas, Ph.D., Head of the Information Management and Decision Architectures Branch, NRL
- Taught at Georgetown and George Mason Universities
- Research interests are developing effective human-system architectures and interfaces using web services technology

Justin Nevitt, Engineering Research Psychologist NRL
- Undergraduate work at the University of Virginia
- At NRL, Justin is conducting research into the areas of service oriented architectures and human-computer interaction.

Rick Rosenberg, President & CEO, Seros, Inc.
- Formerly President, Defense & Intelligence unit at Unisys, Inc.
- Previously with EDS for 18 years
- Seros’s customers include DISA, Office of Naval Intelligence and USMC

Alex Neihaus, vice president, Active Endpoints
- Formerly with Lotus and IBM, worked on messaging systems
- Helped introduce building information modeling to GSA
Some GoToMeeting tips

- Click the double arrow to collapse the panel
- Click the maximize button for the best resolution on your display
- You can enter your Q&A in the chat window…
Or, you can use Twitter to discuss and ask questions

- Use the search term “#activevos”
- We are monitoring the discussion
- We’ll take questions for the Q&A on Twitter as well as in GoToMeeting
Orchestration Research and Assessment at the Naval Research Laboratory

James A. Ballas, Ph.D.
Justin Nevitt

Information Management and Decision Architectures Branch

Naval Research Laboratory
Washington, DC 20375

The views and conclusions contained in this briefing are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Naval Research Laboratory or the US Navy.
“Navy’s C2 systems should be built using a SOA approach. The SOA approach has been developed in the commercial sector for enterprise software systems. To promote reuse and flexibility, it separates out and provides externally callable interfaces to the various components—the data, application logic, user presentation, and orchestration (used to achieve a given work flow).” p. 97.

Partial list of C4ISR Committee:
  John Stenbit, James Hendler, ADM (ret) Archie Clemins, MajGen (ret) Harry Jenkins,
Motivation for Orchestration Technology: 2006 Navy Strategic Plan

• Overarching Global Naval Concept:
  • Distributed, Networked Operations.

• Key tenet: strategic speed and flexible capability.

• Sustain current readiness with exactly the right capabilities for the right cost.

• Build a capabilities based-fleet for the future.
Potential Benefits

• Agility
  • Quickly adapt to warfighter needs.
  • Quickly align Capabilities to current processes and workflow.

• Reuse and Efficiency

• Integration
  • Ensure enterprise governance.
  • Set and enforce standards.
Potential Transformation: Current system

Decision/Dissemination Delay
Minutes
Hours

PPT Preparation Delay
Minutes
Hours
Days

Data Collection Delay
Min, Hrs, Days

Data Entry Delay
Sec, Min, Hrs, Days

Mission Analysts (MAEs) & Subject Matter Experts (SMEs)

Manual Fusion/Correlation

GUI Navigation
Non-Net Data

GUI
GUI
GUI
GUI
GUI
GUI
GUI
GUI
GUI
GUI

Docs
Docs
DBs
DBs
DBs
DBs
DBs
DBs
DBs
DBs

Human Sensors Unknowns

Ground Truth
Potential Transformation: Workflow with SOA and Orchestration

Execution

Decision Makers

Service Orchestration
WSa | WSb | WSc

Decision/Dissemination
Delay
Minutes
Hours

PPT Preparation
Delay
Minutes
Hours
Days

Data Collection
Delay
Min, Hrs, Days

Data Entry
Delay
Sec, Min,
Hrs, Days

Mission Analysts (MAEs) & Subject Matter Experts (SMEs)

Manual Fusion/Correlation

GUI Navigation

Manual Fusion/Correlation

WS = Web Services
(XML, SOAP, HTTPS)

Data

Docs | Docs | DBs | DBs | DBs | DBs | DBs | DBs | DBs | DBs

Human | Sensors | Unknowns

Ground Truth
Avoid the “stupid” Service Oriented Architecture Approach*

- Wrong approach: “Cut across stovepipes, plant some services, so you can get into the data and do cool things.”

- Right approach: “lay out the scenarios you want to carry out and see where they touch stove pipes…Services should be driven by how they will be used”

* 2006 Interview with Grady Booch, IBM Fellow and co-creator of Unified Markup Language, founder of Rational Software (now owned by IBM)
Important Requirements for SOA and Orchestration in Department of Defense

• Non-developer capabilities
• Stability
• Limited Connectivity Environment
• Security
• Thin-client ready
• Support, Maturity and Licensing
• Deployment Flexibility
Overview of NRL Orchestration Projects

• Research on Intelligent Service Orchestration

• Orchestration Technology Assessment
Objective:

• Automatically retrieve relevant information for a Community of Interest (i.e., ad-hoc team).
• Automatically provide secure access to the information for individuals in the team.
• Retrieve real-time and archived data through a single transparent interface.

Approach:

• Intelligent orchestration of web services
  • Extend web services descriptions with semantic information.
  • Adapt computational cognitive techniques to service orchestration.
• Leverage capabilities/experience developed under EVIS project.

Payoff:

“Don’t make the operators search for data we already know they need.”

Col Mark J. Lorenz, Chief, IT Insertion, HQ USSTRATCOM, 29 Aug 06
**Workflow Example: Current technique**

**User 1:** What do you have on XYZ?

**User 2:** I’ll look for something…..

[15 minutes later]

**User 2:** I have found a picture showing XYZ.

**User 1:** What does it show?

**User 2:** I’ll put it onto the shared drive.

[5 minutes later]

**User 2:** OK, its on d:/documents/XYZ.gif.

**User 2:** I also found a document that is relevant. I’ll put it on the shared drive.

[10 minutes later]

**User 1:** I couldn’t get these documents, I don’t have access to the shared drive.
Workflow Example: Orchestrated Services

**User 1:** What do you have on XYZ?

**User 2:** I'll look for something…..

[1 minute later]

**Search service:** 20 pictures and 10 documents have been located and placed on the shared drive.

**Access service:** User 1 and User 2 now have access to the shared drive.

[ 10 minutes later]

**User 2:** Take a look at document 5 and pictures 7 and 8. I think they show what we’re looking for.

[ 1 minute later]

**User 1:** I concur. Let’s make the recommendation.
Workflow Example: Orchestrated Retrieval of Relevant Information

Service Flow
1) Users in a collaboration environment chat with each other.
2) The automated chat content discovery service scans the chat messages in real time.
3) The service submits queries to Federated Search.
4) Federated Search queries its data providers.
5) Federated Search returns results to the automated chat content discovery service and the chat content discovery service does post processing on those results.
6) The automated chat content discovery service displays results to the users through the chat service (inline with the chat or in a sidebar display)
Orchestration Technology Assessment

- Demonstrate the use of service orchestration tools.
  - Used orchestration to quickly develop a workflow that would mediate between data sources and data consumers.

- Orchestration Technology Maturity Assessment
  - Reliability over 30 days
  - Performance and Scalability
  - Performance in Disconnected, Intermittent, Low bandwidth (DIL) Environments
Standards Observed

- WS-BPEL 2.0
- SOAP 1.1
- WSDL 1.1
- XML Schema 1.0
- XML Namespace 1.0
- XPath 1.0
- HTML 4.0.1
- SSL 3.0 (OpenSSL 0.9.7a)
- WSEnumeration
- WSEventing
Demonstrate the use of Service Orchestration Tools.

Used orchestration to quickly develop a workflow that would mediate between data sources and data consumers.
Workflow Design
Workflow Design Simulation
Orchestration Deployment
Execution Monitoring

### Active Processes

<table>
<thead>
<tr>
<th>ID</th>
<th>Process Name</th>
<th>Start Date</th>
<th>End Date</th>
<th>State</th>
</tr>
</thead>
</table>

**Selection Filter**

- **State:** [ ] All [ ] Running [ ] Completed [ ] Compensatable [ ] Faulted
- **Created between:** [ ] and [ ] (yyyy/mm/dd)
- **Completed between:** [ ] and [ ] (yyyy/mm/dd)
- **Name:**

---

Copyright © 2004-2007 Active Endpoints, Inc.
Lessons Learned

• Key outcome: an orchestration can compose web services without requiring any changes in the web services themselves.
• Example messages are essential, especially for schemas that use “wildcard” elements (e.g., anyAttribute) or multiple instances of an element.
  • This required hand editing of xpath code and example messages.
• Simulation in design tools is extremely valuable.
• Going from business process model to BPEL executable, or from one vendor to another is not seamless.
• Key reminder: an orchestration itself is a web service:
  • Is subject to the standards criteria being set for the enterprise (e.g., ws-management, ws-security, ws-addressing, etc.)
  • Will need the engineering that any web service requires (setting up SSL, importing PKI certs in keystores, async vs. sync design, etc.)
Orchestration Maturity Assessment

- Reliability over 30 days
- Performance and Scalability
- Performance in Disconnected, Intermittent, Low bandwidth (DIL) Environments
- Utility: meets requirements
Reliability and Scalability Testing Architecture
DIL Testing Architecture

VMWare ESX Server 3i version 3.5

Orchestration Engine Server

Database Backend

gigabit switch

IPNetSim Satellite Simulator

Packet Capture Device

Grinder Load Tester
Reliability Testing Paradigm: Continuous sampling and monitoring

- Computer resource utilization by all components: engine, database, and grinder:
  - Memory, cpu usage, file sizes
- Key events in logs.
- Network packets using wireshark and subsequent analysis of conversations.
Example 30 Day Data Results: Computer Resource Utilization
Example 30 Day Data Results: Database Growth
Orchestration Issues

- Governance
- Service Availability
- Notification of Changes to Services in an Orchestration
- Controlling Permissions of Published Orchestrations
- Service Granularity
Governance Example: Mediation of Data from one Format to Another

- Mapping of data formats is something that should be governed very carefully.
- Most likely, end users should not be mapping schemas like this.
- If mapped incorrectly, the result be serious: loss of battle space awareness, loss of life or loss of military assets.
- Some elements may be lost or simplified in the translation.
- Some elements that are not in one schema but exist in another will need to be filled out.
  - Could be defaults, user entry, etc.
Service Availability

• Expected vs. Unexpected orchestrations
• Reliability of services is important to their adoption by users.
• Solutions via governance
• Publishing availability metrics in a UDDI
Notification of Changes to Services in an Orchestration

- Orchestrations can be consumed as services.
- Entire orchestrations can be brought down by one faulty or updated service.
- Branching Logic and Partial Consumption
- How does one update/notify the proper personnel of such a change.
Controlling Authorized Usage of Orchestrations

- Access typically controlled via Attribute Based Access Control
- Issue: How to Control Access to an Orchestration that in turn consumes service(s) that should be restricted.
  - User’s Credentials need to be propagated in an assured manner.
  - Propagation may need to get into orchestration logic.
Service Granularity

• Too finely grained – expose too much functionality to the consumer; limit agility.
• Too coarse – stove pipe system masquerading as an SOA; overlapping functionalities and unnecessary rewriting of code.
• No straightforward guidance on this subject from industry or governing bodies.
Important Requirements for SOA and Orchestration in Department of Defense

- Non-developer capabilities
- Stability
- Security
- Limited Connectivity Environment
- Thin-client ready
- Support, Maturity and Licensing
- Deployment Flexibility
Seros Commodity Capability
Implementation Architecture

WS-Management Interface
WS-Management Broker

F5 Big IP IBM DataPower Layer 7 HP BAC MOM Amber-Point Hyperix HQ HP SOA Systinet Mule Galaxy JBoss Virtual Policy Store

Policy Store1

Policy StoreN

To WS-* Interfaces

SOA Dashboard Service

User-Based Orchestration Service

WS-BPEL Interface WS-BPEL Broker ActiveVOS

WS-Notification Interface WS-Notification Broker Sonic MQ or ESB JBoss ESB Apache Active MQ

Information Distribution Service

WS-Metadata Exchange and Search Interfaces

Publishing and Searching Broker Apache Lucene Google Search Appliance

Content Awareness Service

SDK

Customer Machine-to-Machine Clients

Governance and Management Service
ActiveVOS: the BPMS development team love
New ActiveVOS customers Jan-June 2009
Thank you very much

Download a supported, 30-day trial of ActiveVOS at www.activevos.com

Visit ActiveVOS’s blog at www.vosibilities.com (replay of this webinar will be posted there)

Learn more about orchestration: www.activevos.com/indepth.php

Contact info:
Alex Neihaus: alex.neihaus@activevos.com
Jim Ballas: james.ballas@nrl.navy.mil
Justin Nevitt: justin.nevitt@nrl.navy.mil
Rick Rosenberg: rick.rosenburg@seros.com