Background and Goal

- Implementing a Service-Oriented Architecture (SOA) on a large program such as GPS-OCX brings out unique issues.

- The same issues seem to emerge across different programs of this size.

- What follows is a sample of those issues and their resolutions to help your Service-Oriented Architecture be successful, especially on a large program.
GPS-OCX Overview

- What is GPS-OCX
  - The Operational Control Segment for the Next Generation Global Positioning System meant to
    - Improve the accuracy and availability of GPS signals
    - Provide increased capacity for satellite support
    - Move to a more secure and extensible modern architecture

<table>
<thead>
<tr>
<th>Specifics</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Basis</td>
<td>JROC CDD, Sept. 2009</td>
</tr>
<tr>
<td>Major Users</td>
<td>AFSPC, Military and Civil Users</td>
</tr>
<tr>
<td>Contractor</td>
<td>Raytheon – Aurora, Col.</td>
</tr>
<tr>
<td>Major Subcontractors</td>
<td>ITT – Bloomfield, N.J.</td>
</tr>
<tr>
<td></td>
<td>Boeing – Aurora, Col.</td>
</tr>
<tr>
<td></td>
<td>Braxton – Colorado Springs, Col.</td>
</tr>
<tr>
<td></td>
<td>Infinity – Colorado Springs, Col.</td>
</tr>
</tbody>
</table>
Why SOA for GPS-OCX?

▪ Two principal drivers for SOA on GPS-OCX – Net-Centricity and Architecture Tenets

▪ OCX has a Net-Centric strategy to support evolving existing interfaces to more efficiently share data
  – The program has requirements for Net-Centric Capabilities but not explicitly a Service-Oriented Architecture
  – The program must satisfy the Net-Ready Key Performance Parameters

▪ OCX has design requirements and a set of Architecture Tenets for non-functional aspects of the system

▪ A SOA was chosen as a means to satisfy the architecture tenets and align with the OCX Net-Centric strategy
Relevant SOA Standards

**Presentation**
- JSF 2.0
- Portlet API
- WSRP 1.0

**Workflow/Orchestration**
- BPMN
- BPEL 2.0

**Service**
- WS-I Basic Profile 1.1
- JAX-WS 2.1
- JAX-WS 1.1
- SOAP 1.1

**Data**
- XML 1.1
- JAXB 1.1
- XML Schema
- JPA (EJB 3.0)

**Messaging / Mediation (ESB)**
- HTTP/HTTPS
- HTML 4.0.1
- JMS 1.1
- JSR-914
- XSLT 1.0
- Xpath 2.0
- Xquery 1.0

**Governance**
- UDDI 3.0.2

**Security**
- LDAP 3
- RFC 3377
- WS-Security 1.1
- SAML 1.1

**Legend**
- Standard
- DISR Compliant
SOA Challenges

- **Integration** - governing service interfaces and interactions
  - OCX has complex interactions across many services
  - Agreeing, designing, documenting, and integrating the different types of services requires discipline

- **Mediation** – ownership and use of SOA shared resources
  - Should individual teams have their own service bus and manage mediation functions or should these functions be coordinated and managed by a common team?

- **Performance** – monitoring and managing the overhead of web services and the associated security controls
  - How can performance be designed into the SOA infrastructure and services, validated and monitored to ensure an operationally usable system?
Addressing the Challenges (1)

- **Direction, Documentation, and Discussion**
  - A robust set of guideline documents to drive compliance to standards and processes
  - An Interface Repository captures interface documents (WSDL, XSD)
  - A Developer Forum trains, discusses, and agrees guidelines (e.g. the use of shared resources)

- **Service Identification Methodology**
  - Identify Mission Services from DODAF artifacts, model the interactions and elaborate to the appropriate granularity
  - A custom set of UML modeling stereotypes (similar to the SOA modeling language) documents interface attributes
    - Attributes drive understanding and facilitate other modeling (e.g. performance)
Addressing the Challenges (2)

- Design Pattern Repository
  - Collection of Design Patterns to be followed by the SOA Services

  - Contains
    - Core Service Integration Design Patterns
    - Net-Centric Design Patterns
    - Legacy / Reuse Application Design Patterns
SOA Recommendations (1)

- Create a SOA expert team
  - Responsible for all things SOA on the program
    - SOA Architecture and Guidelines
    - SOA Services and Interfaces
    - SOA Implementation and Issues
  - Organization should match your SOA goals

- Eliminate “stove-pipe” services
  - Distinguish between “cross-team” and “intra-team” services but
  - Services should abide by the same standards and guidelines
    (always exceptions)
SOA Recommendations (2)

- Design orchestration into the Architecture
  - Not just the mechanism but the mission service compositions

- Monitoring and Performance
  - Capture performance attributes as a part of interface definition (best if based on an operational profile)
  - Have monitoring and performance guidelines ready for development from the start of design
Questions?