Breakout Session 4: Approaches to Achieving Common Architectures and Common Components

Phillip Schmidt
Session Chair
Thanks to our Contributors

- Paul Denning, Mitre
- Keith Hogie, CSC
- Judy Kerner, Aerospace
- Maj Ray Pry, USAF SMC/CWI
- Phil Schmidt, Aerospace

- Lots of audience talent...
Agenda

- Session Themes
- Approaches and Insights
- Issues and Observations
Session Themes

• What do we have to do to make common architectures and/or components really work and be understood?

• Look at technical/non-technical approaches to solve some challenging architectural issues
Use Standards to Achieve Commonality

• They are not a panacea
• They may not meet your needs (e.g., DII COE)
  – Grow stale
  – GOSIP example
  – Their implementations may be incomplete
• They evolve (e.g., TCP, SQL)
• They may reduce your interface design complexity
• They need an architectural context to be effective
• Payoff to convert to them needs justification
Understand how the trends meet the needs

- Architecture is defined once and tweaked forever => driver for flexibility
- Support for reuse less important than support for operational flexibility
- Evolution of new capabilities via new interfaces => driver for interface management
- Leveraging independent component development
- Interface definition without needing source code
- New business models will be necessary
Address the Challenges of Change

- Designing for change requires more effort
- **Interface Management Issues in Component-based Architectures**
  - Architecture representation to clearly manage interface and implementation separation
  - Implications of interfaces as contracts; limits of IDL
  - Policies and practices that manage interface growth
  - Need to map logical architectural representations to deployed components
  - Manage variation points throughout the artifacts
  - Solves Component Deployment Problem
- **Cannot force users into lock-step evolution**
  - Expect multiple versions will get deployed
  - Optional interfaces
Address the Needs of Legacy Systems

• **Budget critical driver**
  – Massive budget cuts force creativity
  – Optimizations focused on out years won’t get funded

• **Mission requirements/operation come first**
  – Need to keep the mission going
  – Can’t afford to rebuild

• **We suffer from rigid designs of the past**
  – Blurs our focus on what’s important and creates “tension” on how to fix it.
    » Modernize vs Rearchitect
    » Network-centric vs embedded
    » One-size-fits all-solution is not necessarily the right solution
  – Move toward plug-and-use is heroic!!
  – Takes years to implement change (e.g. Commercial: 3 years, Govt: 11 years)

• **Will benefit from modernization be overcome by the benefit of commercialization?**

• **Commercialization as a faster, smarter evolution balanced against the influence to shape it**
Collaborate on Reference Architecture for needed Domains

- Idea of Domain Task Force for Space Ground Systems to shape a reference architecture
- It’s starting
  - NRO/NASA SOTG expanded
- Technical: Correctly scoping domain and the components within
- Organizational: Concern for open participation
Create Business Incentives

• **What’s the Product-line business case for an integrator?**
  – Will integrators be valued in this new model?
  – Always need someone to but pieces together in more complex ways than originally intended.

• **Vendor’s have their “product line”**
  – COTS as an architectural driver
  – May not match the Govt definition the domain => architectural mismatches
  – They need to feel they can shape/contribute to the reference architecture

• **We suffer from our rigid business models of the past**
  – Blurs our focus on what’s important and “tension” on how to fix it.
    » Contractor use of derived

• **Preserve contractor’s intellectual property rights**
Leverage the Technology

• **Operating Missions as Nodes on the Internet**
  – Leverages IP as the Network layer to the Satellite

• **Leverage research of IETF community**
  – Debug
  – Deploy
  – Evolution IP4, IP6, etc

• **Web-based architecture directions for SGS**
  – Java, CORBA performance improvements
  – Configurability of JavaBeans

• **Cheap and Easy are attractive**
  – Some successes with DCOM

• **Watch Maturity**
  – Be prudent when moving toward bleeding edge
Issues and Observations

• Declaring a Center of Excellence for Space entails a responsibility to shape the domains in which we wish to excel
  – Cannot be mandated
  – Cannot be closed

• Perspective affects the way we perceive problems
  – Are we riding the elephant or walking underneath it?

• Support flexibility not perfection
  – We’re never done
  – Even reference architectures evolve

• Scope/design issues
  – Common solution <> one size fits all
  – Does common architecture => common mission?
  – Flexibility to support multiple CONOPS

• Methodologies for evaluating architectures needed
  – Interface semantics