Session Goals

• **Topic**
  – SOA (Service Oriented Architecture) based approaches for architecting satellite ground systems in a net-centric environment – Where we are today

• **Presentations & panel discussion**
  – Focus on sharing experiences in determining when (and if) a SOA-based architecture applies and experiences and lessons learned in developing or prototyping a SOA-based architecture.

• **Panel questions related to topic areas**
  – Software Architecture Considerations and Tradeoffs
  – Successes, Challenges, and Lessons Learned in Applying SOA
  – Implementation Considerations
  – Systems Interoperability
  – Standards and Core Services
  – Future Predictions
• **Acquisition and Oversight Perspective**
  – Michael Kramer, Aerospace
  – Major Steve Paine, USAF
  – John Arcos, Eltefat Shokri, Aerospace

• **Development Perspective**
  – Magdi Carlton, NASA JPL
  – Morris Brill, Northrop Grumman

• **Research Perspective**
  – Scott Tilley, SEI/CMI
  – Richard Taylor (UCI), Nenad Medvidovic (USC), Eric Dashofy (Aerospace)
Key Points

- **Definition of Service-Oriented Architecture**
  - Terminology overload: SOA is all things or “Are we drinking Kool-Aid?”
    - SOA cannot be purchased
    - Requires change of culture/paradigm;
  - Misconceptions on SOAs
    - Net-centric == SOA
    - Legacy applications can be easily integrated into SOA
      - consider using SMaRT for legacy migration
  - SOA is not a technology….it is a paradigm
    - How do you standardize a paradigm?
**Key Points**

- **Experiences- Lessons Learned**
  - Need to align business operations and goals - 75% fail on 1st try
    - Cost of aligning business logic with IT
  - Agility in SOA framework is more important than immediate ROI
    - Reuse drives the long-term business case
    - Industry ROI is typically not seen for the first 2-5 years
  - Start small, fail small, build when you succeed!!
  - Tension between architectural purity and user needs
    - Selection and development of services should be driven by user-centric scenarios

- **Current Status and Appropriateness of Standards**
  - Reference architecture helpful in capturing diverse SOA viewpoints
  - Reference model captures core concepts and relationships to understand essence of SOA
  - Evolving SOA standards
  - Proliferation of competing standards and standards groups
Key Points

• Implementation Considerations – Development
  – Applied to: mission control, data management, S/C analysis, environmental information systems
    • Successful applications were those best suited to a SOA approach
  – Service ownership decentralization and independent evolution entail loss of control
    • Who pays?
    • Who’s responsible?
    • New contract-business incentives evolving
  – Don’t forget
    • Security
    • Training
    • Licensing dependencies
  – Legacy System challenges:
    • Architectural mismatches
    • Operational mismatches,
    • Tool availability
    • Separation of Concerns
  – CMU-SEI SMaRT available to analyze viability of legacy component migration
  – Service granularity
Key Points

• Where are we going – Research
  – Consider WWW view
    • RESTful service architectural style vs SOAP-based Web services
      – exposing info via URLs vs methods
  – Semantics of service definition is a challenge
    • How to model and describe services?
    • QoS contract guarantees
  – Trends in SOA
    • Major Concerns:
      – From hiding Heterogeneity to Standards-based Interoperability to Integration
    • Performance
      – Like a local application to Some Real-Time to Predictable to Predictable +
Conclusions

• Despite existence of an industry standard reference model (OASIS) on SOA, there is significant disagreement on what SOA is

• Decentralization means loss of control
  – Trust is a requisite component

• Successful SOA projects in progress
  – User-centric view
  – Importance of training
  – Essential to understand and align business operations and goals for achievable expectations