Ground System Architectures Workshop

Session 11A

Architecture-Centric Evolution (ACE) Working Group 2010

Sheri Benator, Sergio Alvarado, Phil Schmidt, Eric Dashofy, John Arcos, Mark Nixon, Scott Hendrickson, The Aerospace Corporation; Jeff Estefan, Jet Propulsion Laboratory
Session Goals

• Eighth of a GSAW series
  – Forum for software-intensive system experts, users, developers & researchers to collaborate and elucidate high-level recommendations for improving software architectures representation, development, & analysis

• Topic
  – Innovative Approaches to Software Architecture Development and Analysis

• Presentations & panel discussions
  – Focus on innovative approaches, both applied (including experiences and lessons learned) and in early / research stages.
Presenters/Panelists (1)

- **Innovative Approaches Applied in Practice**
  - Evolution of the GPS Control Segment as Related to Software Architecture
    - Alex Polack and Mike Campbell, The Aerospace Corporation
  - Role of Software Architecture as Part of the NASA Study on Software Complexity
    - Dan Dvorak, NASA JPL
  - Evolution of a Service-Oriented Architecture (SOA) Command & Control (C2) System
    - Ryan Telcamp, The Boeing Company
  - Use of New Software Development Tools/Strategies to Enhance Ability to Deploy Ground Systems from Product Lines
    - Michael Klug, Chris Newton, Northrop Grumman
  - Fault-Tolerant Architectures: Discussion of Techniques Ensuring System Operation during Periods of High Loading and/or Component Failure
    - Stephen Harrington, Booz Allen Hamilton
• Innovative Approaches from the Research Community
  – Domain-Specific Design Analysis and Code-Generation Frameworks
    • George Edwards and Nenad Medvidović, University of Southern California
  – Lessons Learned in Current Applications of Model-Driven Engineering
    • Stephanie August, Loyola Marymount University
  – Survey on Model-Based Software Development Techniques
    • Yongjie Zheng, University of California, Irvine
Key Points

- **Innovative approaches applied in practice**
  - More sophisticated ground systems demanded to support evolving system needs and diverse users
  - Software systems are dynamic and must deal with change
  - Information assurance must be designed in
  - Quantify adaptability, extensibility, scalability through key scenarios
  - Software is a system’s “complexity sponge”
  - Architecture reviews elicit rationale capture and stakeholder decisions
    - One study cited a 12 fold return on its cost
  - Platform neutral strategy gives you bargaining power
    - SOAs simplify the development platform neutral architectures
  - Architecture frameworks such as OSGi are a key enabler of software product lines, but configuration management is an open question
  - Degraded operations are the key consideration for fault tolerance and availability
Key Points

- **Innovative approaches from the research community**
  - Many ways to accomplish Model Driven Engineering (MDE) that vary in goals
    - Domain-specific and architecture-centric approaches show promise
  - MDE most effective in well understood domains
  - Greater tool maturity and training needed for MDE to be effective
  - Variation of stakeholder technical skills affect use of & communication via models
  - Ground system requirements such as real-time data processing & robust fault-tolerance drive need for analyzable & executable models
  - No “one-size-fits-all” language; drives use of domain-specific solutions
  - Building model interpreters to analyze and generate code from domain-specific models is hard; transformations are also hard
  - Model fidelity introduces new challenges for validation and “correctness”
  - DSLs are not being applied in a broad scale in ground systems
Conclusions

• We are addressing ever-more complex systems
  • Emerging approaches (model-driven and domain-specific approaches) provide new ways to manage increasing complexity
• Perennial problems remain, but we are attacking them at increasing scales
  • Reconciling granularity and fidelity
  • Effective approaches to architectural evaluation
  • Tension between generic, well-supported approaches and high-value domain-specific niche approaches
• New challenges also emerging
  • How to address issues like open-source, information assurance from an architectural perspective

Architecture improvement is evolutionary – not revolutionary