Owning the Technical Baseline with Model-Based Engineering

Tying together Architecture and System Integration

Garrett Wampole (gwampole@mitre.org)
David Campbell (dcampbell@mitre.org)

IEEE Software Technology Conference
In an environment where rapid change is a fact of life, our current capability development paradigm is inadequate. Large, complex programs with industrial-era development cycles measured in decades may become obsolete before they reach full-rate production.

The new disciplines required of our Air Force are that of integrator and synchronizer – directing how and when the elements must fit together.

- U.S. Air Force 30 Year Strategy
Owning the Technical Baseline is SAF/AQ Priority – And It Makes Sense

SAF/AQ Priorities

- Continue to get high priority programs right and keep on track – JSF, KC-46, LRS FoS, Joint Stars Recap*, T-X*, EELV, PAR, C2 (AOC 10.2 / 3DELRR), Protected Comms
- Continually improve relationships and transparency with stakeholders: OSD, Hill, Industry, Labs
- **Own technical baseline for important programs**
  - Continue to implement highest impact BBP2.0 parts for best outcomes
    - Implement “should cost”
    - Strong partnership with requirements
    - Employ appropriate contract types
    - Eliminate unproductive processes and bureaucracy
    - Enforce open system architectures and effectively manage data rights
    - Improve acquisition of Services outside traditional acquisition
  - Build to 2023 and beyond – resiliency to peer competitors – experiment and innovate
- Provide and support an environment for effective program execution...to make the PEOs and programs successful

Our View

- Unambiguous Technical Data
- Product more than Document Centric
- Used to Solve Problems
- Manage System Evolution

Dr. Bill LaPlante
Principal Deputy, SAF/AQ
Oct 13
Some Systems Engineering Philosophy

- Future acquisitions must be different, so let’s do systems engineering differently
- Architecture products should participate actively in verification of designs and implementations
Executable Government Reference Architecture (GRA)
Targeting UML

- Class groups related Functionality
- Container for Interface Ports

- Interface Port ‘provides’ (exposes) an Interface
- Defines ICD characteristics

- Interface defines Events
- Groups required pieces of data to make an Interface

- Event contains data element
- Represents a specific data format

- Statechart defines behavior when Interface is used
- Represents the “service contract”
Model-Based Engineering

- Model is the design
- Useful by broad community (war fighter, engineer, tester)
- Auto-generated documentation
- Executable Government Reference Architecture integrated in SIL
Step 1: Define Interfaces

“DataViz” port provides “VisualizedItems” interface that specifies “OwnshipPosition” event.

“OwnshipPosition” event specifies detailed data format as an argument.

Class has “DataViz” port.
Step 2: Create Business Logic Simulation

"Inside" Class is the business logic simulation

Run this code when an "OwnershipPosition" event is received...

Result: Display position on executable model's Globe display – *if message is formatted correctly*.
Step 3: Analyze Model Execution

Executable GRA showing simulation of processing an OwnshipPosition message.
GRA-Based SIL Concept
SIL Analysis Workflow

- Analysis Tools
- Executable GRA
- Simulations (Sensors)

Avionics Message Bus

- Gateway Service
- 1553 Bus
- Terminal Lab
- Rockwell Collins AN/ARC 210
- Simulations (Comms)
- Data Capture Tools
SIL Analysis Workflow

Step 1, Identify Path Through Architecture

Analysis Tools

Executable GRA

Simulations (Sensors)

Avionics Message Bus

Gateway Service

1553 Bus

Terminal Lab

Rockwell Collins AN/ARC 210

Simulations (Comms)

Data Capture Tools
SIL Analysis Workflow

Step 1, Identify Path Through Architecture

Step 2, Identify GRA Component To Analyze
SIL Analysis Workflow

Step 1, Identify Path Through Architecture

Step 2, Identify GRA Component To Replace

Step 3, Install Service Implementation

Analysis Tools

Executable GRA

Simulations (Sensors)

Gateway Service

1553 Bus

Terminal Lab

Rockwell Collins AN/ARC 210

Simulations (Comms)

Data

Contractor Software Service
Benefits

- Discover defects in the design earlier in the development cycle
- Bridge system and software engineering teams
  - Improve efficiency and communication
- Consistent design/interface documentation
  - Generate documents such as ICDs directly from model
- Improve design/review process during active development
  - Base design reviews around model artifacts
- Decrease time to bring new engineers up to speed for development and support
- Easier technology insertion/upgrade impact analysis over project lifecycle

Need to be committed to process and provide adequate resources to achieve benefits
Couldn’t Say It Better If I Tried

IBM and GM Build Smart Systems for Chevy Volt

This announcement highlights the recent work between GM and IBM, showcasing the Rational software used to develop the software system on the 2011 Chevrolet Volt. GM is one of the first U.S.-based automobile manufacturers to produce a next-generation “plug-in, range extended electric vehicle.” It is directly propelled by an electric motor, for up to 40 miles, with a gasoline engine that will drive the vehicle up to an additional 300 miles.

The Chevrolet Volt: IBM Rational Software Helps...


Ideas to Market

Time to Market

Collaboration
# Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRA</td>
<td>Government Reference Architecture</td>
</tr>
<tr>
<td>ICD</td>
<td>Interface Control Document</td>
</tr>
<tr>
<td>MBE</td>
<td>Model-Based Engineering</td>
</tr>
<tr>
<td>SAF/AQ</td>
<td>Assistant Secretary of the Air Force (Acquisition)</td>
</tr>
<tr>
<td>SIL</td>
<td>System Integration Lab</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language (<a href="http://uml.org">http://uml.org</a>)</td>
</tr>
</tbody>
</table>