Integrating Legacy Software: Lessons and Hurdles

John Chobany, Associate Director
Vehicle Concepts Department
Architecture & Design Subdivision

Systems Engineering Division
The Aerospace Corporation
2 March 2011
Introduction to Panel Discussion

• General observations based on The Aerospace Corporation’s participation in an on-going “Think Tank” effort that is looking across the National Security Space (NSS) for lessons and hurdles relevant to migrating legacy systems to new ground system architectures.

• These observations are associated with the integration of legacy software in support of migration efforts towards common-service architecture approaches and are being presented in order to spur panel discussions relevant to the challenges and opportunities of harmonizing systems and components for a wide range of stakeholders.
Observations and Lessons Learned

• Observations:
  – *Reuse of legacy software to support new missions is not always compatible with the legacy systems*
    • Undesirable results can include lower performance and missed requirements
  – *Transition costs to go from legacy to new are not always assessed*
  – *Interface complexity plays an important role in determining the impact to legacy software and overall system costs*
  – *Development and maintenance costs of the common services (or shared capabilities) need to be supported by the missions using those services*
    • Not all participants have an equal share of benefit and may resist paying the “tax” or discontinue participation
  – *System closure, performance, and interfaces are not being modeled prior to acquisition*
    • May find out sometime after ATP that it won’t meet requirements
  – *Life cycle costs are not being assessed prior to acquisition*

• Commonality achieved through the consolidation of legacy "stove-pipes" isn't always the best alternative for reducing program costs
Challenges and Hurdles

- Common assumption is there’s not enough time or resources to do a thorough evaluation of alternatives using concept modeling tools
- It’s hard to dispel the notion that consolidation implies cost savings
  - Just as with the fallacy that all software reuse implies cost savings
- Fairness and equality are not traits that are consistently applicable to aerospace software system performance
  - Some missions have performance needs that far exceed the capability of the common services
- How can we implement both a common-service and mission-unique approach within the same ground system architecture?
- Wrapping the legacy code and adding more processors is a neat trick, but at some point we reach diminishing returns on performance
  - Amdahl’s Law
  - Gunther’s law

\[ C(N) = \frac{N}{1 + \alpha(N - 1) + \beta N(N - 1)} \]

- C - relative capacity
- N - number of processors or users
- \( \beta \) - contention
- \( \alpha \) - coherency delay
Opportunities

Follow Good Systems Engineering Practices

• Up-front modeling of the proposed new common-service architectures should be performed pre-acquisition
  – *Modeling to assure system closure (all requirements can be met)*
  – *Modeling to assess performance (latency, throughput)*
  – *Identify test and validation considerations*

• Concept studies enable even earlier programmatic decision making
  – *Rapid yet thorough tradespace exploration of new concepts and block upgrades provides better insight into system needs*
  – *Identify performance and cost drivers*
  – *Determine cost and technical feasibility*
  – *Assess margins and risks*
  – *Refine and validate requirements*
  – *Path pruning*

*Of all decisions affecting life cycle costs, approximately 70% are made during Concept Design*
Example: Concept Design Center

- **Ground Segment Team (GST)**
  - *Designs the Ground Systems Architecture at a conceptual level*
    - Facilities, personnel, processing, communications, and cost estimates
  - **GST Architecture** characterized by a Master Function List (MFL) mapped against a framework of nodes (sites) plus a definition of all possible communication links
    - **MFL indicates whether a function is performed or not at a particular node**
      - Capability-only is an option which typically provides hardware and software functionality, but not staff
    - **Possible functionality includes:**
      - Mission Processing
      - Mission Management
      - TT&C
      - **Ground Control**
      - **Common Services**
      - **Facilities Management**
    - **Communication links include terrestrial and space-to-ground links**
Backup
Multidisciplinary CDC Teams
... and Their Interactions

- **System Architecture Team (SAT)**
  - Constellation design and coverage analysis
  - Top-level element sizing and interface definition
  - Relative cost versus requirements and utility

- **Space Segment Team (SST)**
  - Payload and spacecraft subsystem design
  - Detailed cost and performance estimation
  - Top-level ground segment and software sizing

- **Ground Segment Team (GST)**
  - Facilities, personnel, processing, communications, and cost estimates
  - Top-level space segment sizing

- **Electro-Optical Payload Team (EOPT) & Communications Payload Team (CPT)**
  - Detailed payload subsystem trades
  - Performance and cost estimation
  - Mission requirements implications
  - Top-level spacecraft and ground segment estimation

*Core team members for each study plus additional unique expertise as required*
Master Function List (MFL)

• Master Function List (MFL) is input to the Node Module
  – Defines the functions required by the system in the GST study
  – Communicates system design elements to each of the GST modules
    • Ensures that the GST modules comply with the functions required by the program in the study
    • Deletes functions that are out of scope or GFE’d for the study
    • Requires supporting program / GST study documentation and discussions to interpret correctly for each module
      – Complexity, heritage elements
  – Is tailored for each program to add, modify or delete functions
    • Functions can be
      – Provided
      – Provided and Not Staffed (for example, backup facilities)
      – Not Provided
    • Tailored MFL elements are defined in the GST architecture documentation (report, memo or briefings)
# Sample Master Function List

<table>
<thead>
<tr>
<th>Mission Processing</th>
<th>Ground Command &amp; Control</th>
<th>Support Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Mission Data Capture</td>
<td>• Acquisition &amp; Tracking</td>
<td>• Telemetry Storage and retrieval</td>
</tr>
<tr>
<td>• Mission Data Processing</td>
<td>• Command &amp; Control</td>
<td>• Training</td>
</tr>
<tr>
<td>• Report Dissemination</td>
<td>• Telemetry Processing</td>
<td>• Data Base Management &amp; System Administration</td>
</tr>
<tr>
<td>• User Interface</td>
<td>• Orbit &amp; Attitude Determination</td>
<td>• Data Security</td>
</tr>
<tr>
<td>• Optical Data Processing</td>
<td></td>
<td>• Vehicle Simulation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Development Environment</td>
</tr>
<tr>
<td>Mission Management</td>
<td>Ground System Management</td>
<td>Facility Management</td>
</tr>
<tr>
<td>• Mission Planning &amp; Scheduling</td>
<td>• Communication Connectivity Interface</td>
<td>• Physical and Structural Control</td>
</tr>
<tr>
<td>• Schedule Optimization</td>
<td>• LAN/WAN Management</td>
<td>• Security Control</td>
</tr>
<tr>
<td>• Constraint Analysis</td>
<td>• Ground Terminal Control</td>
<td>• Maintenance</td>
</tr>
<tr>
<td>• Space &amp; Ground Resource Monitoring</td>
<td>• Timing Services</td>
<td></td>
</tr>
<tr>
<td>• Mission Assessment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Task Satisfaction Analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Misc. Functions</td>
<td>Ground System Management</td>
<td></td>
</tr>
<tr>
<td>• Launch and Early Orbit Support</td>
<td>• Communication Connectivity Interface</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Anomaly Resolution</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Operations Management</td>
<td></td>
</tr>
</tbody>
</table>
Key GST Module Interfaces
Functionality of GST Modules

System-Level Modules
- NODE
  - Distributes Master Function List to all Modules
  - Monitors/controls module status
- SYSUMM
  - Repository for system-level characteristics and costs

Staffing
- Specify functional positions and staff type at each position
- Specify number of seats per functional position
- Specify personnel type per seat

Communications
- Analyze connectivity options
- Size data rates, bandwidths
- Network and protocol design
- Determine required equipment

Software
- Identify software functions
- Specify characteristics
  - New/reuse / COTS
  - Effort to adapt / integrate COTS
- Effort for databases, GUI, etc

Information Architecture
- Model flow of information
- Characterize information
  - Nature of data
  - Producers / consumers
  - Data rate
- Characterize network constraints

Processing
- Specify processing equipment
  - Workstations / Servers / PCs
  - Special purpose racks
  - Data archive
  - Hubs / routers / switches
  - Firewalls / guard boxes

Facilities
- Site development
- Site access
- Security
- Space and infrastructure for equipment and personnel
- Antenna facilities incl. radomes

Cost
- COTS H/W
- Staffing
- Facilities
- Software
- Overall wraps
Ground Segment Architecture Framework

Node N
- Facility N
  - Staff @ N
  - Computers @ N
  - SW @ N
  - Terminals @ N
    - X
    - L
    - G
    - A

Node 1
- Facility 1
  - Staff @ 1
  - Computers @ 1
  - SW @ 1
  - Terminals @ 1
    - X
    - A
    - C
    - Y

Nodes 2 thru M

Link 1
Link L
Link J

External

john.chobany@aero.org
Vehicle Concepts Department/Architecture and Design Subdivision