Software Architecture for Satellite Ground System Development

Jeff Garland
Principal Consultant
CrystalClear Software, Inc
Nature of Ground Systems

- Highly Complex, Large-Scale, Mission Critical
- Distributed / Concurrent systems
- Many COTS components -- require integration
- Heterogeneous components
  - Multiple languages
  - Multiple storage techniques etc
Key Requirements of the Approach

- Need an approach that scales to huge systems
- Need to be understandable by large range of project participants
  - minimal use of modeling elements (e.g., complex UML)
- Need to map to 'real things'
- Needs a focus on 'interfaces'
- Needs to include quantitative and other annotations
Modeling Software Architectures

- UML -- Unified Modeling Languages
- IEEE 1471 -- Views and Viewpoints
  - Viewpoint -- template for a view (purpose, applicability, stakeholders)
  - View -- instance of a viewpoint
- 14 Viewpoints in all
- Key Elements to Model
  - Components and Interfaces
  - Subsystems
  - Processes and Databases
## Top 5 Views

<table>
<thead>
<tr>
<th>Viewpoint</th>
<th>UML Diagram Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>Use Case</td>
<td>Show the external system actors and the system under design.</td>
</tr>
<tr>
<td>Component</td>
<td>Component</td>
<td>Illustrate component communications.</td>
</tr>
<tr>
<td>Component Interaction</td>
<td>Interaction</td>
<td>Interactions among components.</td>
</tr>
<tr>
<td>Layered Subsystem</td>
<td>Package</td>
<td>Illustrate layering and subsystems design.</td>
</tr>
<tr>
<td>Deployment</td>
<td>Deployment</td>
<td>Mapping of software to hardware for distributed systems.</td>
</tr>
</tbody>
</table>
**Component Viewpoint**

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Describe runtime component connectivity and communication. Can be applied to performance analysis and later the process interaction design.</th>
</tr>
</thead>
<tbody>
<tr>
<td>When Applicable</td>
<td>During system design and development, as analysis views and subsystems are identified.</td>
</tr>
<tr>
<td>Stakeholders</td>
<td>Architecture Team, Subsystem Developers, Test Team, Software System Engineering Team, Systems Engineering Team, Project and Development Managers (to a lesser degree)</td>
</tr>
<tr>
<td>Scalability</td>
<td>Drawn with scenario or component focus, Can make use of composite components.</td>
</tr>
<tr>
<td>Relation to Other Views</td>
<td>The Component Views should be consistent with components shown in the Process and Deployment Views.</td>
</tr>
</tbody>
</table>
Component View Example

<<external component>>
Web Browser  
{0..10}

<<component>>
Web Server  
{ quantity = 1..5, jsp adapted }

<<component>>
Windows Interface  
{1..5}

<<component>>
Orbit Analysis

<<component>>
Orbit Projection
{CORBA}

<<component>>
Telemetry Processor

<<component>>
Telemetry History

<<RDB>>
Telemetry Archive

<<RDB>>
Orbit Data

<<external component>>
Telemetry Feed System  
{one per satellite}

<<external component>>
Data Store

<<component>>
Message Protocol Name

<<component>>
Raw Telemetry

<<component>>
Output Port

<<component>>
Input Port

Mechanism Annotations

orbit projection
200 - Orbit Projection
{CORBA}

300 - Burn Data  
{corba event channel}

900 - Burn Data

procedural interface

raw telemetry

{soap/http}
# Component Description Table

## Example

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teller Client</td>
<td>Provide a user interface tuned for the needs of bank tellers.</td>
</tr>
<tr>
<td>Teller Server</td>
<td>Provides services for the use of tellers. This includes administrative functions.</td>
</tr>
<tr>
<td>Session Manager</td>
<td>Provides transaction and session id support.</td>
</tr>
<tr>
<td>Customer Info Server</td>
<td>Provides service interfaces that provide access to basic customer information.</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Deployment View Example
Where Has this Approach Been Applied?

- Genesis was development of Iridium Ground-System
  - Was pre-UML at the time
  - Refined on projects since then
- Framework documentation for another satellite system
- Other large distributed systems
  - financial, communications, enterprise systems
- Other satellite ground systems
Systematic Elaboration of Architecture

Use Cases

Component Views

Component Interaction Views

Interface Definition
Evolution

- **Runtime**
  - Understand the runtime evolution requirements and constraints

- **Buildtime**
  - Subsystem dependencies is key

- **Be Aware of Hard to Evolve Parts of Architecture**
  - Data == concrete
  - COTS difficult to upgrade
Evaluation

Evaluation Issues
- Complexity of systems is a barrier
- ‘architectural view’ doesn’t reflect implementation

Recommendations
- Focus on key scenarios
- Use appropriate views to focus
- Stay as concrete as possible
- Get an experienced software architect
Conclusions and More Info

- Systematic development of software architecture is achievable
- Tools are a problem
- UML issues

- More info at www.largescalesoftwarearchitecture.com
- Email: jeff@crystalclearsoftware.com