SYSTEMS ENGINEERING INTERFACES: A MODEL BASED APPROACH

Elyse Fosse, Christopher Delp
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, California USA
Context

- **AMMOS:**
  - Advanced Multi-Mission Operations System
  - Product line: Adaptable tools and services for operating NASA’s robotic missions
  - Key advantage: Cost and Risk

- **Operations Revitalization Task**
  - Enhance, extend multi-mission operations

- **MOS 2.0**
  - The Next-Generation Mission Operations System
The Flight Ground Interface

- Key interface for Mission Operations Systems Engineers (MOSE)
- Focuses on the interaction between:
  - An operational flight system
  - The MOS located on the ground
- The MOSE needs to understand:
  - Allowable information exchange
  - Allowable behavior between the flight system and the MOS
  - Operational constraints: timing, quality of product
  - To name a few
Flight Ground Interface Description

• An MOSE relies on documents and diagrams to describe the specification of the Flight Ground Interface.

• Ops Rev created a framework consisting of:
  • Interface-specific language (extending from SysML)
  • Patterns for modeling interfaces and instances of interaction
  • Viewpoints for addressing specific concerns related to interface engineering
    • The MOSE can focus on the systems engineering work related to interfaces and the needed document/presentation artifacts can be generated from the model.

• Ops Rev implemented to framework to describe the Flight Ground Interface
Interface Engineering with SysML

Provides a precise model-based representation for specifying interfaces

Interface: System boundary presented by a system for interaction with other systems

Interface Specification: Describes the nature of the system boundary in terms of properties and functionality
Interaction Engineering with SysML

Provides a precise model-based representation of integration of parts through interfaces

Interaction: An instance of an interface

Interaction Specification: Describes how two or more system boundaries can connect and effect one another
MMOS delegates its functionality to the ground domain in order to interact with the operational flight system.
Interface Object Flow View

What are the flows between parts of the system?

<table>
<thead>
<tr>
<th>MMOS-Flight System Interaction Specification</th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>-MMOS-Flight System Director Interaction</td>
<td>Flight System Command History</td>
<td>Flight System Ephemeris Loads</td>
</tr>
</tbody>
</table>
Performance and Limitations View

What are the expectations and limits of the given interaction

<table>
<thead>
<tr>
<th>Agreement</th>
<th>Producer</th>
<th>Consumer</th>
<th>Products</th>
<th>Responses</th>
<th>Frequency</th>
<th>Duration</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command and Telemetry Agreement</td>
<td>MMOS</td>
<td>Operational Flight System</td>
<td>• Flight System Commands</td>
<td>• Flight System Response</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOS Directing Flight System Configuration Agreement</td>
<td>MMOS</td>
<td>Operational Flight System</td>
<td>• Flight System Configuration Commands</td>
<td>• Flight System Response</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Flight System Configuration File</td>
<td>• Telemetry</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Function Occurrence View

How do functions occur between parts of the system?

<table>
<thead>
<tr>
<th>Event (Function Occurrence)</th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receive Flight System Telemetry</td>
<td>Telemetry</td>
<td></td>
</tr>
<tr>
<td>Update Flight System Commands</td>
<td>Flight System Commands</td>
<td>Flight System Response</td>
</tr>
</tbody>
</table>
Interface Delegation View

MMOS delegates to Mission Services for functionality fulfillment

Mission Services are discipline-specific functionality groupings
Interface patterns used for MMOS and Flight System are applied at the Mission Service level as well
Other Interface Layers

The focus of Ops Rev and its interface engineering implementation is on the business layer.

The same methodology and Viewpoints can be used to express the specifications of the other layers:

- **Data**: elaborates information identified in the business layer into the actual data to be transferred to the Flight System.
- **Protocol**: translates into the standards-compliant protocol stacks.
- **Hardware**: allocates software functionality to specific hardware.
Approach Advantages (1 of 2)

Ask questions of the model and get reportable answers

<table>
<thead>
<tr>
<th>SE Verification</th>
<th>Model Interrogation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is the connection between 2+ parts valid with respect to information?</td>
<td>Check that all SysML ports connected to each other of the same (or specialized from the same) type?</td>
</tr>
<tr>
<td>How many interfaces exist without connections to other parts?</td>
<td>Find all interfaces with no connectors</td>
</tr>
<tr>
<td>Is the connection between 2+ part valid with respect to the framework patterns?</td>
<td>Check that all SysML ports are either connected by a constraint property (agreement) or have parent port that are connected by a constraint property (agreement)</td>
</tr>
</tbody>
</table>
Approach Advantages (2 of 2)

- **Automation**
  - The interface model can be queried and replicated in an automated fashion to provide instances of interaction

- **A core set of reusable Viewpoints**
  - Allows for document and presentation artifacts to be generated directly from the model
  - The MOSE can focus on the engineering content of the model
  - The reviewer can focus on how the Mission-specific view responds to the concerns of the Viewpoint.
Approach limitations

• In SysML 1.3 there is not a good way to trace functionality of a system across its interactions
  • The sequence diagram elements have no direct connection to the interface ports
• SysML 1.4 was released and introduces refinements to port specifications
Summary and Next Steps

Currently:
- Ops Rev developed and maintains a framework that includes interface-specific language, patterns, and Viewpoints
- Ops Rev implements the framework to design MOS 2.0 and its 5 Mission Services
- Implementation de-couples interfaces and instances of interaction

Future:
- A Mission MOSE implements the approach and uses the model based artifacts for reviews
- The framework extends further into the ground data layers and provides a unified methodology.