Sequence System Building Blocks: Using a Component Architecture for Sequencing Software

Barbara Streiffert  
Taifun O’Reilly  
Jet Propulsion Laboratory,  
California Institute of Technology  
Pasadena, CA

Barbara.Streiffert@jpl.nasa.gov  
Taifun.Oreilly@jpl.nasa.gov

SpaceOps 2006 Rome, Italy June 19 – 21, 2006
Agenda

• Introduction
• Next Generation Uplink Planning System (NGUPS) & Ensemble
• Multi Mission Planning and Sequencing
• In-Situ Collaboration
• Conclusion
Introduction

Current Project Characteristics:
• Spacecraft Complexity
  • Multiple processors (Instruments, Attitude Control, Command/Data Handling)
  • Commands with more autonomy
• Desire for more data return for science

Current Goals for Operations:
• Faster more efficient ground operations that are more cost effective
• Low risk ground operations

Aids for Ground System Software:
• New Software Methodologies and Tools
  • Agile Development
  • Rich Client Platform/Frameworks
  • Components/Features
  • Refactoring
Next Generation Uplink Planning System (NGUPS) & Ensemble

NGUPS & Ensemble:

• Funded by the Mars Technology Program
  • Goals:
    1. To utilize new technology for the Mars missions
    2. Infuse proven technology into project software
    3. To promote a more efficient ground system for the various Mars missions in particular Mars Science Laboratory.

• Performed a study of Mars Exploration Operations (MER)
  • Determined:
    1. Need for a more cohesive ground system
    2. Too many User Interfaces representing the same data differently
    3. Desire to reuse sequences
NGUPS & Ensemble:

The new software methodologies and tools were used to implement the proposed MSL ground system including:

1. Agile Development
2. Rich Client Platform (Eclipse)
3. Components/Features (groups of components or legacy tool)
4. Refactoring (primarily of legacy tools)
5. Use of Java utilities including Java Messaging Service
Next Generation Uplink Planning System

MSL Ground System Software

External Model Application Interface

User Interface & Activity Planning

Activity Resources & Targets

Activity Modeling

Constraint Propagation

Interactive Timeline & Table Editing

Sequence Modeling

Activity Expansion/Command Editing

Mobility/Arm Sequencing

Eclipse RCP

Not Used by MSL

Editors

Modeling

3rd Party Software
Multi Mission Planning & Sequencing (MPS)

MPS:

• Funded by Multi Mission Ground Systems and Services
  • Goals:
    1. Provide Reusable Ground System software
    2. Provide Project adaptation capability (mechanism for project dependent aspects)
    3. Provide multi mission aspects in software (i.e. ability to check flight rules)
    4. Provide Projects a lower cost for Ground System development (only one organization implements, documents and tests)
• Holds User Group meetings to determine project needs
  • Recent meetings concluded:
    1. Need for common input (XML)
    2. Need for utilizing laptops (Linux environment)
    3. Need for refactored Activity and Sequence modeling to exclude User Interface
Multi Mission Planning & Sequencing (MPS)

MPS:

The new software methodologies and tools were used to upgrade the MPS ground system tools including:

1. Rich Client Platform (Eclipse including the Eclipse Bridge for legacy tools)
2. Components/Features (groups of components or legacy tool)
3. Refactoring (primarily of legacy tools)
4. Use of Java utilities such as Java Messaging Service (JMS), Java XML Binding for input, etc.
NGUPS, Ensemble & MPS:

The three groups have common lists of new capabilities with the exception of using Agile Development:

1. Rich Client Platform (Eclipse including the Eclipse Bridge for legacy tools)
2. Components/Features (groups of components or legacy tool)
3. Refactoring (primarily of legacy tools)
4. Use Java Utilities such as Java Messaging Service (JMS), Java XML Binding for input, etc.
5. Agile Development (NGUPS & Ensemble)

This Ground System was needed for the upcoming In-Situ missions of Phoenix and MSL so

In-Situ Collaboration was born
In-Situ Collaboration

In-Situ Ground System Software

External Model Application Interface

User Interface & Activity Planning

Activity Dictionary Editor

Activity Resources & Targets

Interactive Timeline & Table Editing

Eclipse RCP

Constraint Propagation

Activity Modeling

External Model Application Interface

Mobility/Arm Sequencing

Activity Expansion/Command Editing

Sequence Modeling

expansion, models, rules

Configuration/Communication

Editors

Modeling

3rd Party Software
In-Situ Collaboration

Challenges:

• Four teams involved:
  • NGUPS (represents MSL)
  • Ensemble
  • MPS
  • Phoenix

• Work and Schedules must be coordinated
• Prioritization includes four separate teams
• Phoenix and MSL have different schedules with different need dates
• All four teams have their own meetings
• Work must be allocated to various teams
In-Situ Collaboration

Solutions:

• Tasks have been allocated based on each mission’s priorities and requirements.
• Team members work for multiple teams.
  • Example: Taifun O’Reilly (a co-author of this paper) works ½ time for MPS and 1/5 time for NGUPS.
• Selected System Engineers and Developers attend MPS, NGUPS and Phoenix meetings.
  • Example: I work for NGUPS but I attend the Phoenix activity dictionary meetings, the NGUPS activity dictionary meetings and the In-Situ status meeting.
• Prioritization has been based on need date. Phoenix launches in 2007 and MSL launches in 2009. However, MSL’s focused technology money ends FY06.
• Work has been allocated to various teams based on the type of work
Project Specific/Project Independent

Joint In-Situ Project

- MSL
  - Project Specific:
    - Activities
    - Flight Rules
    - Mission Rules
    - Commands
    - Sequences
    - Project Models

- Phoenix (PHX)
  - Project Specific:
    - Activities
    - Flight Rules
    - Mission Rules
    - Commands
    - Sequences
    - Project Models

Multi-Mission Planning and Sequencing (MPS)

Project Independent:
- XML Readers/Writers
- Flight Rule Checking Infrastructure
- Activity Infrastructure & Expansion
- Sequence Infrastructure & Expansion
- Command Handling & Checking
- Resource Checking Infrastructure
- Constraint Propagation Infrastructure
Samples of the Software

Timeline Editing and Display:

<table>
<thead>
<tr>
<th>Name</th>
<th>Duration</th>
<th>Energy</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>MECA Science</td>
<td>01:36:00</td>
<td>9.79 Wh</td>
<td>181.21 Gb</td>
</tr>
<tr>
<td>NECA_AFMI</td>
<td>00:21:00</td>
<td>3.66 Wh</td>
<td>5.33 Gb</td>
</tr>
<tr>
<td>NECA_OMI</td>
<td>01:00:00</td>
<td>5 Wh</td>
<td>175.78 Gb</td>
</tr>
<tr>
<td>NECA_TECPI</td>
<td>00:15:00</td>
<td>1.12 Wh</td>
<td>102.6 Mb</td>
</tr>
<tr>
<td>Engineering</td>
<td>04:00:00</td>
<td>200 Wh</td>
<td>0</td>
</tr>
<tr>
<td>CPU_ONI</td>
<td>04:00:00</td>
<td>200 Wh</td>
<td>0</td>
</tr>
</tbody>
</table>

Downlink Browsing:

Note: No real data has been used in these displays.
Samples of Software

Activity Type Editing:

Target Selection:

Note: No real data has been used in these displays
Conclusion

The Ultimate Goal: On-Line Shopping for a GDS

GDS Tool Shopping

- Visualization
- Sequence Modeling
- Timeline & Table Editing
- Activity Expansion/Command Editing*
- Mobility/Arm Sequencing
- Activity Resources & Targets
- Turn Model
- Activity Modeling
- Constraint Propagation

The Ultimate Goal: On-Line Shopping for a GDS