Ground Data System Analysis Tools
To Track Flight System State Parameters for the Mars Science Laboratory (MSL) and Beyond

Daniel Allard, Lloyd R. DeForrest
Jet Propulsion Laboratory, California Institute of Technology
SpaceOps, May 5-9, 2014, Pasadena, CA, USA
Overview

- Background
- About Parameters and Tracking Them
- Tracking Tool Implementation
- Issues
- Conclusions
Parameter Tracking Background

• In 2006, Mars Global Surveyor (MGS) was lost due to spacecraft commanding against an invalid fault mode parameter configuration\(^1\)
  – Situation led to non-Earth orientation of HGA as well as rapid, fatal power drain
  – Clearly parameter state must be tracked or risk invalid flight system behavior

• Mars Exploration Rovers (MER) parameter experience led to design and implementation of an MSL Parameter Management Tool (PMT)

• So… What are parameters?
  – Configurable system values residing in volatile and persistent memory
  – E.g. Thermal and power targets, fault monitor behavior…
  – Specified in dictionaries of one sort or another

• Why parameters vs. configuration tables?
  – More fine tuned control, can potentially modify down to a single parameter at time
  – However, they are significantly more complex to track

• What questions do we ask?
  – “What is the current state of parameters as of time X” (Snapshot Query)
  – “What parameter changes have occurred between time X and Y” (History Query)
### Battery Control Board (BCB) Module Parameters

#### MON Parameter Group

Parameters specifying fault monitor configuration

**Group Copies:**

- `batt_offline_at_boot_bcb1`
- `batt_offline_at_boot_bcb2`
- `batt_offline_imminent_bcb1`
- `batt_offline_imminent_bcb2`
- `fail_to_communicate_with_both_bcbs`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Default</th>
<th>Units</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>detection_enabled</td>
<td><code>Bool (in gbl/gbl_pub.h)</code></td>
<td><code>TRUE</code></td>
<td><code>none</code></td>
<td><code>FALSE : TRUE</code></td>
<td>Whether or not detection is enabled for this monitor</td>
</tr>
<tr>
<td>response_enabled</td>
<td><code>Bool (in gbl/gbl_pub.h)</code></td>
<td><code>TRUE</code></td>
<td><code>none</code></td>
<td><code>FALSE : TRUE</code></td>
<td>Whether or not response is enabled for this monitor</td>
</tr>
<tr>
<td>error_persistence</td>
<td><code>U32</code></td>
<td><code>1</code></td>
<td><code>none</code></td>
<td><code>1 : 0xFFFFFFFF</code></td>
<td>Error count at which monitor becomes YELLOW</td>
</tr>
<tr>
<td>fault_persistence</td>
<td><code>U32</code></td>
<td><code>1</code></td>
<td><code>none</code></td>
<td><code>1 : 0xFFFFFFFF</code></td>
<td>Error count at which monitor becomes RED</td>
</tr>
</tbody>
</table>

Parameter identifier generated from module, group copy (if present) and parameter
Enabling a Snapshot Query

Module Parameter Evidence (Dump File, Checksum, etc).

Group Copy Evidence

Cross-Module Snapshot

Spacecraft Event Time

Snapshot times
A parameter value over time

MSL parameter evidence is downlinked as per module or for a specific group copy. But evidence specification in telemetry is not consistent across missions.
Implementation

• Designed a parameter tracking web service with a back-end database
  – Start of coding green lighted 3 months prior to MSL launch
  – Flight software (FSW) had been in development for over 5 years by this time
  – Three half time developers, all supporting other critical software development
    • MSL data management, relay planning systems..
  – Implemented on top of MSL Mission Processing and Control Systems (MPCS) software

• Technologies and Standards
  – Coded in Java and Python
  – ReST for server interactions
  – ReSTlet Java web server
  – Hibernate for information persistence
  – HTML and Velocity templates for data visualization

• Deliveries
  – Early cruise version
  – Multiple surface ops versions
    • Including recent ops version to deal with design issues found along the way
Visualization

Parameter Management Toolkit

Snapshot Report

<table>
<thead>
<tr>
<th>SCET</th>
<th>Venue</th>
<th>RCE</th>
<th>Persistence</th>
<th>Template</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Surface</td>
<td>A/B</td>
<td>Memory</td>
<td>snapshot.html</td>
</tr>
</tbody>
</table>

Filter

- Category
  - OR -
  - FSW Module
  - Group
  - Group Copy
  - Parameter

Parameter Snapshot Report

Venue: Surface
RCE: A/B
Module: bcb

<table>
<thead>
<tr>
<th>Module</th>
<th>Group</th>
<th>Group Copy</th>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
<th>Last Update</th>
<th>From Evidence</th>
<th>Evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>bcb</td>
<td>mon</td>
<td>bar_offline_immune_bcl1</td>
<td>detection_enabled</td>
<td>True</td>
<td></td>
<td>2014-09T03:48:04.435</td>
<td>BcbPams_0449595466-30997-1.dat</td>
<td></td>
</tr>
<tr>
<td>bcb</td>
<td>mon</td>
<td>bar_offline_immune_bcl1</td>
<td>response_enabled</td>
<td>True</td>
<td></td>
<td>2014-09T03:48:04.435</td>
<td>BcbPams_0449595466-30997-1.dat</td>
<td></td>
</tr>
<tr>
<td>bcb</td>
<td>mon</td>
<td>bar_offline_immune_bcl1</td>
<td>error_persistence</td>
<td>1</td>
<td></td>
<td>2014-09T03:48:04.435</td>
<td>BcbPams_0449595466-30997-1.dat</td>
<td></td>
</tr>
<tr>
<td>bcb</td>
<td>mon</td>
<td>bar_offline_immune_bcl1</td>
<td>fault_persistence</td>
<td>1</td>
<td></td>
<td>2014-09T03:48:04.435</td>
<td>BcbPams_0449595466-30997-1.dat</td>
<td></td>
</tr>
<tr>
<td>bcb</td>
<td>mon</td>
<td>bar_offline_immune_bcl2</td>
<td>detection_enabled</td>
<td>True</td>
<td></td>
<td>2014-09T03:48:04.435</td>
<td>BcbPams_0449595466-30997-1.dat</td>
<td></td>
</tr>
<tr>
<td>bcb</td>
<td>mon</td>
<td>bar_offline_immune_bcl2</td>
<td>response_enabled</td>
<td>True</td>
<td></td>
<td>2014-09T03:48:04.435</td>
<td>BcbPams_0449595466-30997-1.dat</td>
<td></td>
</tr>
<tr>
<td>bcb</td>
<td>mon</td>
<td>bar_offline_immune_bcl2</td>
<td>error_persistence</td>
<td>1</td>
<td></td>
<td>2014-09T03:48:04.435</td>
<td>BcbPams_0449595466-30997-1.dat</td>
<td></td>
</tr>
<tr>
<td>bcb</td>
<td>mon</td>
<td>bar_offline_immune_bcl2</td>
<td>fault_persistence</td>
<td>1</td>
<td></td>
<td>2014-09T03:48:04.435</td>
<td>BcbPams_0449595466-30997-1.dat</td>
<td></td>
</tr>
</tbody>
</table>

Simple HTML web interface

Multiple data views configurable via Velocity templates
Parameter Management Toolkit Architecture

MPCDS Downlink Processor (chill_down)
Telemetry
Flight System

Parameter Dump Products
MPCDS Database
Parameter State at time X
Query Parameter dump products

Parameter Tracker Database
Parameter Service
PMT Admin Tool
Parameter Structure, Definition, Initial Values
PMT Dictionaries
- Manage Venues
- Trigger data collection
- Load dump products

Parameter Reports (snapshots, history, commands)
Supports automated post-pass data collection

PMT Report Tools
Challenges and Issues

• Challenges
  – Flight software developed for years ahead of PMT implementation, and fully locked down by the time it started
    • Any design issues were the problem of the PMT implementation
  – Always lower priority to MSL’s data management systems
    • Iterative design updates well up through Surface

• Primary Design Issues
  – Lack of standards across dictionaries and downlink data leading to issues in parameter identification/mapping of parameter data out of telemetry
  – Revision management approach in the face of out-of-order data receipt
A snapshot at any time is merely a query for ‘most recent prior context’, with pointer references auto-resolved via Hibernate. Made for a very fast query.

At each point in time where parameter evidence is found, a context revision is generated, including that data and pointers to the most recent evidence for every other parameter.
### Some Design Issues

#### Out of Order Data Handling

<table>
<thead>
<tr>
<th>Context 1</th>
<th>Context 2</th>
<th>Context 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 1</td>
<td>Module 1</td>
<td>Module 1</td>
</tr>
<tr>
<td>Group</td>
<td>Group</td>
<td>Group</td>
</tr>
<tr>
<td>Group Copy</td>
<td>Group Copy</td>
<td>Group Copy</td>
</tr>
<tr>
<td>Parameter</td>
<td>Parameter</td>
<td>Parameter</td>
</tr>
<tr>
<td>Parameter</td>
<td>Parameter</td>
<td>Parameter</td>
</tr>
</tbody>
</table>

Even though there is a 1-1 mapping between parameters in dictionary and telemetry, the identifiers are not consistent between the two, requiring error-prone substring matching to put back together. Dictionary ID may be A_B_C_D, telemetry may be B_C_D_E. Varies from one FSW module to another.

A 1-1 unique identifier between dictionary and telemetry evidence would make this aspect of the problem trivial.

#### Identification Mismatches

- New group dump arrives, earlier SCET than existing dump
- Later group reference not updated with new group pointer

Required substantial additional meta-data management to solve.
Conclusions

Lessons Learned

• While we did eventually implement a working version of the tool, the late start and resulting inability to affect interfaces or otherwise address integration issues at the design level resulted in overall higher than expected costs. Design and integration of these and similar functions must be addressed much earlier in the mission development lifecycle.

Continued Work

• Version recently implemented and released for Soil Moisture Active-Passive (SMAP)
• MSL updates to improve revision management and to track evidence of parameter “truth” using memory checksums and command set counters
References


Acknowledgements

The author would like to acknowledge present and former JPL software developers David Noble, Paul Wang, Nicole Ameche, Hayk Arutyunyan, Pearl Haw and Frank Hy for support and contributions to the effort.

The work described in this paper was carried out at the Jet Propulsion Laboratory (JPL), California Institute of Technology (Caltech), under a contract with the National Aeronautics and Space Administration (NASA). © 2014 California Institute of Technology. Government sponsorship acknowledged.