

Flight Software Effects on Ground Systems

Nathaniel Parsons
Associate Professional Staff
Embedded Applications Group
Space Department
(443) 778-1612

Nathaniel.Parsons@jhuapl.edu

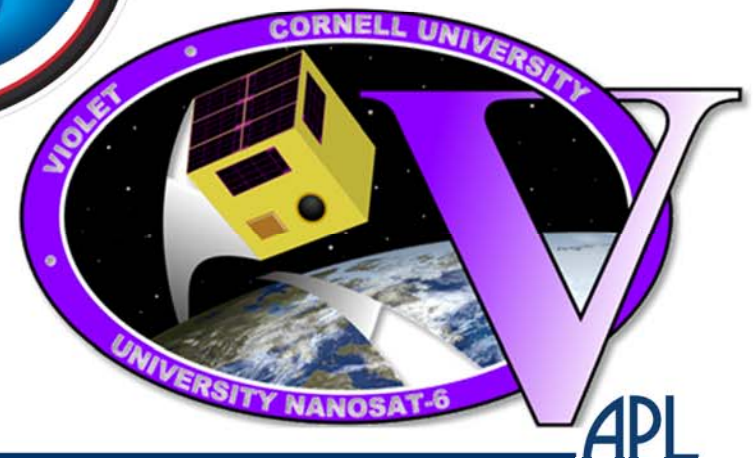
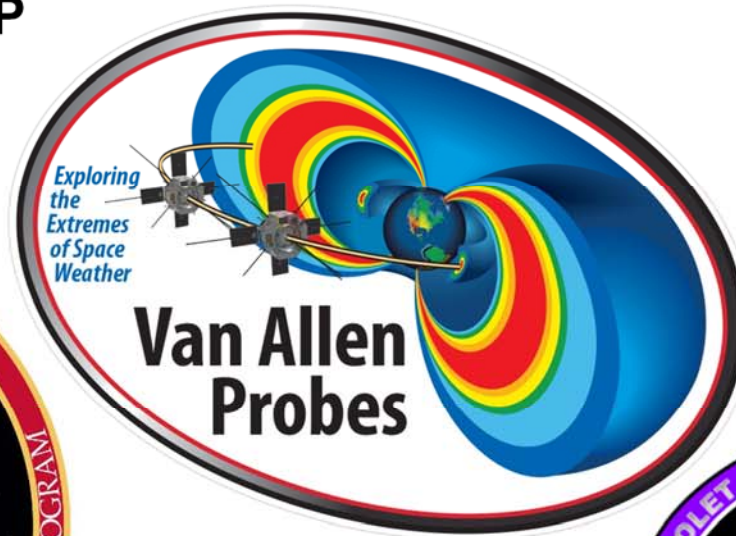
GSAW 2013

The logo for Applied Physics Laboratory (APL) at Johns Hopkins University, consisting of the letters 'APL' in a large, bold, blue sans-serif font.

JOHNS HOPKINS UNIVERSITY
Applied Physics Laboratory

Project Experience

- **2007-2010 CUSat: Cornell University Satellite Project**
- **2009-2010 Violet (Cornell University)**
- **2010-2012 Van Allen Probes (JHU/APL)**
 - Formerly RBSP



Project Experience

- ***CUSat, Phase C/D***
 - Ground Systems Lead
 - Telemetry & Commanding Subsystem
 - Software Systems Lead
 - InControl Software Lead
- ***Violet, Phases B - C***
 - Flight Software
- ***Van Allen Probes, Phases C - E***
 - Flight Software System Testing
 - Flight Software Independent Acceptance Testing
 - Integrated Electronics Module (FC enclosure) Box-level Testing
 - Mission Operations

Idea 1: Early Integration is Good

- ***Reveal problems earlier***
 - Interface Description
 - Interface Implementation
 - Interface Design
 - User (Operator) Interface/Experience
 - Before FSW or GSW design is frozenx
- ***Mini-MOC***
 - Package end-to-end ground system in 1 rack
 - Initial delivery in Phase B
 - FSW Developers use same interface as Ops
 - Test script reuse as operations scripts

Idea 1: Early Integration is Good

- ***Initial Suggestion: Apply principle to other tools***
 - Mission Planning
 - Analysis
 - ...
- ***Some tools may not be useful to devs as-is***
 - Overkill for short tests -> separate relevant portions
 - Too cumbersome -> simplify and streamline
- ***Involve Ops in FSW/GSW testing***

Idea 2: Compensating for FSW

- ***FSW often forced to simplify implementations***
 - “There’s still enough information in telemetry for the ground software to reconstruct what’s happening.”
 - If GSW doesn’t do this automatically, Ops must fill gap manually
 - But Ops should always be thinking in the problem domain
- ***Examples***
 - CUSat parameter upload tool
 - Command sequences stored as binary chunks
 - Open loop G&C
 - Non-streaming data (e.g. memory objects)
 - On-board file systems
 - *Allocation and overfilling are major concerns for Ops*
 - *Ability to recognize time critical/high priority data is key*
 - *Filenames mean little without knowledge of recording configuration*
E.g. 11000000550000000000_1_H

Idea 2: Compensating for FSW

- ***Initial Suggestions***
 - Keep special track of these trades
 - Involve GSW and Ops
 - Verify GSW adequately reconstructs necessary info
 - Goal: hide FSW compromises from operator
 - Ask FSW devs to deliver testing tools

Idea 3: Debugging/Error Handling

- ***At first sign of anomaly, Ops must determine location and nature of fault***
 - Flight systems (simpler, harder to debug)
 - Ground systems (more complex, easier to debug)
- ***Relevant fault scenarios***
 - Working example: connection loss at data rate change
 - Misconfiguration
 - *Systems behaving as required, but not as expected*
 - *Data rate change scheduled improperly by “operator error”*
 - GSW error
 - *FSW responds correctly to incorrect GSW behavior*
 - *GSW misformats data rate change command*
 - FSW error
 - *FSW responds incorrectly to correct GSW behavior*
 - *E.g. failure to make data rate change or unrelated safing*

Idea 3: Debugging/Error Handling

- *Initial Suggestions*
 - Provide clear audit trail from operator intentions to FSW behavior