
Josh Choi and Antonio Sanders
Jet Propulsion Laboratory
California Institute of Technology
SMAP Mission Overview
SMAP Mission Overview

- **Earth satellite mission**
  - Directly measure Earth surface soil moisture and freeze-thaw states
  - L-band radar and radiometer instruments
  - Low-Earth, sun-synchronous orbit (LEO, SSO)
  - Recommended by the National Research Council’s Earth Science Decadal Survey Panel to NASA in 2007
  - Budgeted: ~$700 million

- **Baseline mission: 3 years**

- **Managed, developed, and operated by JPL**
  - Currently in Phase C

- **Launch in November 2014**

Note: The SMAP mission has not been formally approved by NASA. The decision to proceed with the mission will not occur until the completion of the National Environmental Policy Act (NEPA) process. Material in this presentation related to SMAP is for information purposes only.
SMAP Passes

• **Short passes: 5-10 mins**
  – 20-25 passes per day

• **Goddard Space Flight Center’s (GSFC) Near Earth Network (NEN) and Space Network (SN) ground stations**

• **Raw science data downlink via X band**
  – Using NASA’s Earth Observing System (EOS) Data and Operations System (EDOS)
  – 130 Mbps downlink rate

• **Engineering telemetry and commanding via S band**
Real-Time Operations During a Pass

- Establish real-time engineering telemetry downlink connection
  - Then establish uplink interface

- **No station monitor data** = determine closed-loop connection via commanding and telemetry

- Monitor spacecraft health (via EHA and alarms, EVRs)

- Radiate uplink products
  - Routine: Ephemeris updates, retransmission/deletion commands
  - Infrequent: Background sequence loads, activation commands, on-board file management commands, orbit trim maneuver (OTM) commands, critical process commands
  - Special: Flight software loads/patches, instrument look-up tables (LUTs)
  - Ad hoc commands
Activity Between Passes

• **NEN/SN station connection**
  – Based on pass list
  – GDS telemetry and command subsystems ready 5 mins prior to start of pass

• **Analyze reports**
  – Science telemetry gap reports
  – Pass reports

• **Pull and process recorded engineering telemetry**
  – Files in form
SMAP Ground Data System (GDS)

- SMAP GDS consists of numerous subsystems that provide specific and necessary services to support the mission
  - Interfaces with external systems: the flight system and the Science Data System (SDS)

- Some of the functions provided by subsystems:
  - Mission planning
  - Sequence generation
  - Flight system modeling
  - Station scheduling
  - Schedule distribution
  - Spacecraft analysis
  - Orbit determination
  - Trajectory analysis and planning
  - Maneuver design
  - Instruments analysis
  - NAIF SPICE kernels generation/distribution
  - Time correlation
  - Level 0 data capture/distribution
  - Science data gap reporting

- Generation of products for science support
- Common data exchange platform
- ... and more

- Real-time operations functions:
  - Capture, process, display, and generate reports of engineering telemetry
  - Translate and radiate flight system commands
  - Generate recorded engineering data gap reports and generate retransmission commands based on them
  - Track on-board system parameters

- Scope of real-time operations automation:
  - Engineering Data Processing Element (EDPE)
  - Command Processing Element (CPE)
  - Automated Operations Element (AOE)
GDS Architecture Functional View
AMMOS Mission Data Processing and Control System (AMPCS)

• Telemetry processing (including alarms), display, storage, query, reporting, and *automation* subsystem
  – Provides full-featured uplink user interface and basic uplink capabilities (for test environments)
  – Multimission
  – Java and Python-based

• MPCS Test Automation Kit (MTAK)
  – Python API framework providing programmable access to AMPCS features
  – Send commands, SCMFs, file loads, and raw data files
  – Receive EHAs, EVRs, and data products
  – Log user-generated messages in AMPCS session log
Automation Strategy

• Keep costs low by using combination of pre-existing features in AMPCS and MTAK API

• Generic Python scripting with MTAK
  – Check, start, and stop AMPCS processes
  – Send *No Operations* commands, monitor incrementing command counter telemetry
  – Initiate radiation of routine uplink products (ephemeris updates and retransmission commands)

• AMPCS user interface for real-time monitoring
  – Human operator maintains full visibility of automated operations

• AMPCS off-line processing
  – Data playback feature allows recorded engineering telemetry to be processed from files on demand

• AMPCS command-line features
  – Applications/tools configurable and executable via CLI
  – Timed execution via Linux *cron* job scheduler
Categorical Breakdown of Routine Real-Time Operations Automation

Telemetry & Command Automation

is a composite of...

- Pass-Prep Automation
  - Script to Verify/Initiate Station Interfaces
  - automatically executes...

- Recorded Engineering Data Processing Automation
  - Script to Fetch Recorded Engineering Data From Station & Initiate Processing
  - can manually execute...

- Pass List Fetch Automation
  - Script to Fetch Latest Pass List from Schedule Store & Schedule Upcoming Invocations of Pass-Prep Automation

- Miscellaneous Automation
  - Script to Do A
  - Script to Do B
  - Script to Do C
  - Script to Do X
  - Script to Do Y
  - ...

Operator
AMPCS Downlink Processor GUI

The image shows the interface of the AMPCS Downlink Processor GUI, displaying real-time data and status. The GUI includes a message log with timestamps and message types, such as Internal Log, Start Of Test, Internal Log, Connect, Raw Data Summary, and Log. It also shows statistics like In Sync Frames, Out-of-Sync Count, Invalid Frames, Idle Frames, Database Queue Length, Message Queue Length, and Product Queue Length.
Example AMPCS Monitor GUI
Automation Strategy (cont.)

- **AMPCS session logs**
  - Allow identification of anomalies and range affected
  - Interleaved with telemetry and commands

- **Objective: Reduce operator staffing**
  - Standard 5 days a week, 8:00 a.m. to 5:00 p.m. work days

- **Automated science data retransmission – how important?**
  - Historical station performances show 96% science data completeness requirement can be satisfied without retransmission capability

- **Drawback: Fragmented automation control**
  - Different activities, different timetables (e.g. regular intervals vs. pass schedules)
  - Meta-automation
  - User-unfriendly management
Toward “Lights-Out”

- Jason-1 and WISE missions demonstrated “lights-out” real-time operations
  - Automation is monitored 24/7 by an on-console flight controller
  - Jason-1 annually reports data completeness of over 99%, with 93% of the data via automated tracking passes
  - SMAP inherits Jason-1 and WISE missions’ automation approach

- Improving automation using an automation controller
  - Fragmented automation control can cause greater operations cost hit when things go wrong
  - An automation engine could centrally manage scheduling, execution, enabling/disabling runs, failure response, and configuration

- Dedicated user interface for automation
  - Enhance quality of information provided to users
  - Allow users to easily manage automation activities, configuration
  - Assist faster failure recovery
Context Diagram Showing Automation Engine & Interface
Tack! Questions?