Flight System Operability Issues

Steven Wissler
Jet Propulsion Laboratory
California Institute of Technology

Low-Cost OPS Guidelines

- Flight System conforms to existing standard uplink/downlink interfaces and GDS capabilities.
  - Develop a standard re-usable flight software architecture.
  - Don't re-engineer an off-the-shelf existing GDS. Saves in Systems Engineering, documentation and validation costs.
- Early inclusion of OPS Systems Engineering to scope operability issues and develop groundwork for Flight/Ground interfaces.
  - Saves on expensive re-work and cost uppers later in development.
- Get the mission system ready for launch, not just the flight hardware.

Flight System Characteristics Which Enable Low-Cost Mission Operations

- The “simple” spacecraft is generally expensive to fly.
  - Example: Body-fixed solar arrays and high-gain antennas.
    - The spacecraft must be turned in order to do pretty much anything…
  - Sun avoidance.
    - A spacecraft which can turn from attitude A to attitude B, without violating any constraint is much easier than one which must go through extensive ground checks for every maneuver to prevent safing.
- Good margins for telecom, power and thermal.
  - Reduces need for complex operational strategies and subsystem modeling.
- Simple data management architecture with good margin on data collection / storage / downlink capabilities.
  - Use on-board file system for data storage.
  - Instruments generate deterministic data as files in file system.
  - Standard mechanisms for uplink/downlink of files (i.e. CFDP).
- High level, functional commanding.
  - Flight operations teams should not need to be assembly language programmers
  - Reduces need for system state modeling and flight rule checking.

Command Interface Issues

- Command structures are often designed by flight software engineers without regard to operations needs.
  - Parameters which change frequently should be isolated from parameters which never or rarely change.
    - Example, a command which contains control settings, such as gains, as well as the flag to turn the function on/off, requires operations to set the control settings every time the command is sent. This can result in the wrong values being sent, as these values may change over time.
  - Naming conventions: command and parameter names should be clear, unambiguous and consistent across all S/C subsystems.
    - Example: no double – negative commands
  - Parameter units should be consistent, there should not be multiple methods for changing the same parameter, with specification of different units.

Command Interface Issues

- Instrument pass-through commanding
  - Affects of instrument commands are invisible to spacecraft MOS
  - Flight rules related to instrument operations are found late in ATLO.
  - MOS must invent methods to figure out what instruments are doing.
  - Commands should provide functionality at the proper level
    - EXAMPLE: separate commands which set LOW and HIGH order bits in order to set a single functional value.

Command Interface Issues

- All FSW parameters which need to be updated as a function of Mission activity should be set through a command interface.
  - Don’t use activation of pre-compiled “configuration” files in a sequence
  - Don’t use generic parameter set commands which command parameter numbers (i.e. memory loads) , Always use mnemonic representation of parameter name :
    - Example:
      - setParameter (Parameter=10,Value=60).
    - Should be:
      - setThermalRWATempLimit(Value=60)
- Rational:
    - This type of S/C operations hides the details from the MOS/GDS and makes it difficult to know what the sequence will actually do.

Telemetry Interface Issues

- Telemetry structures are often designed by flight software engineers without regard to operations needs.
  - Critical data may be spread around large packets, containing non-essential data, which makes low-data rate operations difficult.
  - Operations should be able to construct new data packets without redelivering flight software.
- Spacecraft engineering data necessary for science data reconstruction.
  - Example: instrument should monitor attitude telemetry and insert attitude data into instrument data headers, rather than have MOS re-construct attitude at sample time from sparse or non-existent spacecraft engineering telemetry.

National Aeronautics and Space Administration
Jet Propulsion Laboratory
California Institute of Technology

Poorly Conceived Autonomy

- On-board autonomy is often inserted as a localized technology demonstration without proper systems engineering to determine the affects on other parts of the system.
- The cost to model and validate the autonomy, in order to convince operations management that it will not harm the spacecraft can be very expensive and must be included in the cost to implement the autonomy.
- Example: on-board image compression
  - If non-deterministic, requires:
    - Proper data prioritization strategies to ensure capture of all critical science data.
    - Keeping sufficient margin to ensure no data loss may result in less data captured than not using compression.
- Example: Turn device on, wait until temperature=XX, then continue.
  - Non-deterministic spacecraft operations can be very expensive for the MOS to operate. Difficult to model timing interactions to ensure spacecraft health and safety and science data return.