Lessons Learned in the Livingstone 2 on Earth Observing One Flight Experiment

Sandra C. Hayden [1] Adam J. Sweet[2]
NASA Ames Research Center (ARC), QSS Group, Inc, Moffett Field, CA, 94035, USA

Seth Shulman[3]
NASA Goddard Space Flight Center (GSFC), Honeywell, Greenbelt, MD, 20771, USA

The Livingtone 2 (L2) model-based diagnosis software is a reusable diagnostic tool for monitoring complex systems. In 2004, L2 was integrated with the JPL Autonomous Sciencecraft Experiment (ASE) and deployed on-board Goddard's Earth Observing One (EO-1) remote sensing satellite, to monitor and diagnose the EO-1 space science instruments and imaging sequence. This paper reports on lessons learned from this flight experiment.

The goals for this experiment, including validation of minimum success criteria and of a series of diagnostic scenarios, have all been successfully met. Long-term operations in space are on-going, as a test of the maturity of the system, with L2 performance remaining flawless. L2 has demonstrated the ability to track the state of the system during nominal operations, detect simulated abnormalities in operations and isolate failures to their root cause fault. Specific advances demonstrated include diagnosis of ambiguity groups rather than a single fault candidate; hypothesis revision given new sensor evidence about the state of the system; and the capability to check for faults in a dynamic system without having to wait until the system is quiescent.

The major benefits of this advanced health management technology are to increase mission duration and reliability through intelligent fault protection, and robust autonomous operations with reduced dependency on supervisory operations from Earth. The work-load for operators will be reduced by telemetry of processed state-of-health information rather than raw data. The long-term vision is that of making diagnoses available to the onboard planner or executive, allowing autonomy software to re-plan in order to work around known component failures.

For a system that is expected to evolve substantially over its lifetime, as for the International Space Station, the model-based approach has definite advantages over rule-based expert systems and limit-checking fault protection systems, as these do not scale well. The model-based approach facilitates reuse of the L2 diagnostic software; only the model of the system to be diagnosed and telemetry monitoring software has to be rebuilt for a new system or expanded for a growing system. The hierarchical L2 model supports modularity and expandability, and as such is suitable solution for integrated system health management as envisioned for systems-of-systems.

1 Introduction

Livingstone is a model-based diagnosis tool developed at NASA's Ames Research Center. It uses a model of a system, such as a spacecraft, along with the commands to and observations from the system, to determine the system's current state [7]. The original Livingstone was written in LISP, and implemented several research algorithms in model-based diagnosis. It was used on several applications projects, most prominently as the Mode Identification and Recovery component of the Remote Agent Experiment (RAX) that flew on Deep Space 1 in 1999 [5].

Livingstone 2 (L2) was developed from the algorithms in the original Livingstone, but written in C++ and with the new capability of diagnosing and tracking multiple faults over a time history [4]. L2 also has been used on several applications projects, including monitoring and diagnosis of the liquid propulsion feed

[1] PI/PM, Computational Sciences Division, NASA ARC, M/S 269-3.
[2] Research Engineer, Computational Sciences Division, NASA ARC, M/S 269-1.
[3] EO-1 Flight Operations Team Technical Lead, Flight Control Systems, NASA GSFC, M/S 428-2.

system for the X-34 reusable launch vehicle [8]; and of the electro-mechanical actuators on the X-37 vehicle [9]. These X-vehicle programs were cancelled before reaching flight; however, the X-34 L2 experiment was completed using high-fidelity simulation data.

The current work, the Livingstone on Earth Observing One (EO-1) experiment, is the first flight experiment of Livingstone since the Remote Agent. The experiment leverages the X-34 and X-37 work, and also NASA JPL's Autonomous Sciencecraft Experiment (ASE) [6], to accomplish deployment within tight schedule and resources. ASE consists of JPL's Continuous Activity Scheduling, Planning, Execution and Replanning (CASPER) planner; the science event detection software and the Spacecraft Command Language (SCL) from Interface and Control Systems (ICS). The experiment architecture and details of technical issues encountered during integration and adopted solutions are described in [1]. This report documents the final results of the experiment, and explores the potential for the Livingstone technology to improve current fault protection methods in use at GSFC. Gaps and shortcomings in Livingstone are also identified, to feed back into research and development of next-generation diagnostic technologies.

2 Final Results of Experiment

The LEO-1 experiment met and surpassed the pre-defined minimum success criteria (MSC). The MSC were defined as the set of diagnostic capabilities which establish progress over previous work. These capabilities were demonstrated by successful completion of 4 key scenarios on the spacecraft. Flight testing continued on to successfully validate 16 of the 17 defined scenarios. In addition, the LEO-1 experiment completed a long-duration test covering many days of spacecraft operation. Thus, the LEO-1 experiment met the MSC, and furthermore demonstrated the maturity of the diagnostic technology. Table 1 summarizes the experiment's results.

Table 1: L2 on EO-1's demonstrated functionality compared with previous Livingstone experiments

Functionality	Remote Agent (L1)	PITEX (L2)	L2 on EO-1
Spacecraft Hardware in the Loop	Yes	No	Yes
Multiple Hypotheses	No	Yes	Yes
Multiple Hypotheses with Backtracking	No	Yes	Yes
Diagnosis During Transients	No	Yes	Yes
Separation of Code and Model	Yes	No	Yes
Number of diagnostic scenarios	2	24	17
Long-term space operations	No	No	Yes

2.1 Minimum Success Criteria (MSC)

The MSC, set out in [1], are a set of requirements for the experiment to establish progress over previous work. Briefly, the MSC are:

1) Spacecraft Hardware in the Loop
   The L2 experiment shall be deployed on-board EO-1, and shall demonstrate monitoring of nominal operations and diagnosis of anomalies in the spacecraft subsystems.
2) Multiple Hypotheses
   Multiple alternative fault candidates shall be presented in the failure diagnosis, with an indication of relative likelihood.
3) Multiple Hypotheses with Backtracking
   In light of new evidence, the list of diagnostic fault candidates shall be revised. This may entail a revision of the most likely fault candidate [4].
4) Diagnosis During Transients
   Diagnosis of a failure of one component shall not be delayed by concurrent commanding of other independent components on the spacecraft. Livingstone in general must wait for a system to achieve a steady-state after a command was issued in order to perform diagnosis; in L1 the entire system must be quiescent to do a diagnosis. This decreases the precision with which the state of

the system is tracked. In the worst case, if commands are being issued continually, there will never be an opportunity to diagnose the system.

In further work with L2 on PITEX, the ability was developed to diagnose one subsystem while another independent subsystem is transitioning. Drawbacks of this Real- Time Interface (RTI) approach were error-prone time segmenting and embedding of domain-specific information in the RTI code. Unnecessary complexity was introduced to support a policy of controlling the timing of forwarding information to L2 and of performing diagnoses. A new solution eliminating this obfuscation was developed for L2 on EO-1. The approach is a combination of a transient modeling methodology and a simple, domain-independent RTI, and is discussed further in [1].

5) Separation of Code and Model

All L2 code shall be independent of the diagnostic model. The idea of the model-based approach is to maintain a separation between the diagnostic engine and the model. If not achieved, any model change would require corresponding updates to the source code. In the case of EO-1, a source code change requires an upload of the entire ASE/L2 software to the spacecraft, which takes over a week.

These criteria were all met by the successful completion of the scenarios defined in the next section.

2.2 Scenario Validation

The scope of the EO-1 L2 model is a subset of the spacecraft components most relevant to the science data collection sequence: the two imaging instruments, the Hyperion Science Instrument (HSI) and the Advanced Land Imager (ALI), and the solid state data recorder, the Wideband Advanced Recorder Processor (WARP). The L2 model of EO-1 and its development are described in [2]. Scenarios were created for each fault mode in the L2 model. Since we cannot count on actual spacecraft failures, and any failures that do occur would be intercepted by flight software, faults were injected into the nominal operations timeline by altering telemetry. This was done by withholding satellite telemetry from L2 or by substituting a minimal number of telemetry values with values indicative of the fault. In both cases, the fault simulation has minimal footprint for the most realistic tests.

Table 2 below lists all seventeen scenarios which were executed, and their results. Two of the scenarios involved nominal operations and fifteen were fault scenarios. Sixteen of the seventeen scenarios executed successfully. As expected, scenario FS01 did not pass, as the 1 second exposure time for the dark calibration image is too fast for L2 to receive the data-gate telemetry change. A similar situation, elaborated further in [1], was experienced on the flight hardware testbed, nonetheless it was decided to proceed with flight testing this scenario.

To recap, scenario FS01 required that L2 detect that the ALI data-gate failed to enable upon command. However, the operational sequence involved “blind” commanding, in which commands are sent in rapid succession without awaiting their telemetry responses. In this case, the ALI data-gate was commanded enabled, then immediately commanded disabled again before the "enabled" telemetry could be received. As a result of transition delays which are set according to telemetry rates, the data-gate component did not have time to enter the steady state “enabled” mode, and remained in an unconstrained transitional mode until disabled. Since the “enabled” mode was never entered, the observation that the data-gate failed to respond and remained disabled generated no conflict. The final “disabled” mode was consistent with the unchanged data-gate “disabled” telemetry. L2 assumes no faults exist until evidence to the contrary is received; in this case, this results in a missed diagnosis or false negative for FS01, during both ground and flight tests.

Table 2: Scenario Validation Test Results

No.	Scenario ID	Sub-system	Fault Injected	Final Diagnosis Candidate(s) and Component Fault(s)	Passed?
1	DCE		None. Nominal scenario.	no component faults	Y
2	Dual		None. Nominal scenario.	no component faults	Y
3	FS01	ALI	Data-Gate Failed Disabled	no component faults	N

4	FS02	ALI	Data-Gate Failed Enabled	data-gate failed enabled data-gate unknown fault	Y
5	FS03	ALI	Mechanism Power Failed Disabled	mechanism power failed disabled mechanism power sensor unknown fault	Y
6	FS04	ALI	Mechanism Power Failed Enabled	mechanism power failed enabled mechanism power sensor unknown fault	Y
7	FS05	ALI	Mechanism Power Sensor Failed	mechanism power sensor unknown fault	Y
8	FS06	ALI	Aperture Cover Failed Closed	aperture cover failed closed LED 08 unknown fault aperture cover stuck transitioning	Y
9	FS07	ALI	Aperture Cover Failed Open	aperture cover stuck open LED 08 unknown fault LED 09 unknown fault	Y
10	FS08	ALI	Aperture Cover Failed Intermediate	aperture cover failed intermediate LED 08 unknown fault	Y
11	FS09	ALI	LED 09 Failed	LED 09 unknown fault	Y
12	FS10	ALI	LED 08 Failed	LED 08 unknown fault	Y
13	FS20	HSI	Aperture Cover Failed Open	aperture cover stuck open aperture cover sensor unknown fault	Y
14	FS21	HSI	Aperture Cover Failed Closed	aperture cover stuck closed aperture cover sensor unknown fault	Y
15	FS23	HSI	Electronics Error	electronics error electronics unknown fault	Y
16	FS24	HSI	Aperture Cover Sensor Failed	aperture cover stuck open aperture cover sensor unknown fault	Y
17	FS35	WARP	Failed To Record	aperture cover sensor unknown fault	Y

2.3 Coverage of New Activities for Extended Flight Operations

The first upload of L2 was as part of the ASE R2 build. The extended runs of L2 were performed with the uplink of the ASE R3 build. CASPER now controlled several new activities of the spacecraft, and monitoring and diagnosis was expanded to support these new R3 activities.

Support for the new activities did not require full modeling. Although it was very much desired to grow the model during this phase, project resources were inadequate for the task. There was a high risk of over-expanding the model beyond our ability to properly develop and test. To avoid blemishing the good record established thus far, a conservative approach was adopted and a set of minimal model changes with SCL support was implemented.

On expanding coverage to these 7 new activities, the main issue encountered was observability. As L2 was running as part of ASE on a secondary processor, any commands coming from the C&DH flight software or from the ground (Absolute Time Sequence or ATS load) were not visible to L2. L2 only had access to the commands issued by ASE when the planner was in control of the spacecraft. Also, EO-1 has developed sequences of commands, called Relative Time Sequences (RTSs) for common command combinations. One RTS can then be invoked to accomplish a sequence of operations on board the satellite. However, the commands within the RTS are not visible to L2, and expansion of a single RTS command to

its constituent commands cannot be mimicked within L2 itself. When ASE extended their functionality to issue RTSs as well as component commands, the SCL software was modified to issue the RTS to the spacecraft, but break out the individual commands in order to send them to L2.

The following activities required expansion of the L2 model of EO-1, as well as support from SCL:

a) Hyperion Science Instrument (HSI) Solar Calibration

The HSI's imager is occasionally re-calibrated by taking an image of the sun, whose spectral characteristics are well-known. However, imaging the sun directly would damage the instrument, and so the aperture cover moves to a special solar calibration mode to accomplish this action. The HSI aperture cover's solar calibration mode, its transient mode, and the transitions to/from the new modes were added to the L2 model. SCL forwarded the new YCVRTOCAL command to L2, and monitored the HSI's cover position sensor to send the SOLAR_CAL observation to L2.

b) Advanced Land Imager (ALI) Outgassing

The ALI instrument is occasionally commanded to perform an outgassing action to remove contaminants from the device. The ALI aperture cover is left in a partially open state while the ALI is heated, driving the contaminants off into space. RTS 197 and 196 are used to partially open and then close the ALI aperture cover. The ALI aperture cover's intermediate position mode, its transient mode, and the transitions to/from the new modes were added to the L2 model. SCL broke out these new commands from the RTSs and forwarded them to L2. No new observations were required.

The following activities required no expansion of the L2 model, but additional support from SCL:

a) ALI Lamp Calibration

The ALI contains a set of lamps whose spectral characteristics are known; these are occasionally used for taking calibration images similarly to a solar calibration. In this activity, the aperture cover is closed while the lamps are commanded on in succession. The ALI lamp calibration is done by commanding a macro defined within the device, named i_fl_cal. Similarly to the RTSs, L2 did not have access to the subcommands in this macro. SCL broke out the commands that the macro sends to the ALI datagate, and forwarded these to L2. No new observations were required.

b) ALI Lunar Calibration and HSI Lunar Calibration

Similarly to solar calibration, occasionally images are taken of the moon to calibrate the ALI and HSI imagers. However, unlike the sun, imaging the moon cannot damage the instruments. No special modes are required for a lunar calibration. The ALI/HSI Lunar Calibration activity includes a lamp calibration activity. Besides the implementation of the lampcal described above, no further changes were required to support these activities.

c) Bandstripping

Bandstripping is an activity done in support of the autonomous science operations of the ASE experiment. It is extremely CPU intensive, and it is likely that the diagnostic software would be starved of CPU during this activity. Hence, model expansion was not attempted for this activity. SCL prevents bandstripping commands from reaching L2, inhibiting WARP hardware and software telemetry if they correspond to bandstripping.

d) X-Band downlink contacts

RTS 202 and 52 are used at X-Band acquisition and loss of signal. The L2 model already included modes and transient modes corresponding to these activities. SCL broke out the commands from these RTSs, forwarding the WARP Xband hardware and software commands (BCMMODEPB, WRMSXOUT, WRMEXOUT) and WARP X-band telemetry (PLAYBACK, XBPB) to L2.

2.4 Results of Extended Flight Tests

As shown in Table 3, the longest run thus far is 55 days. Cumulative operations time in space is at least 143 days. Average spacecraft activity was about 15 Data Collection Events (DCEs) per day. During these 143 days, there were no failures on the spacecraft within L2's diagnosis scope defined for this experiment. This is fortunate for the spacecraft and its mission, but means that L2 did not have an opportunity to diagnose actual spacecraft faults. L2 did report two false positives due to timing issues, explained below.

Table 3: L2 Extended Flight Test Results

Extended Test Identifier	Period of Operation	False Positives/ Negatives or Software Bugs	Outcome
ASE R2 #1	22 days. 12/1/2004 to 12/22/2004	False Positives. No other issues reported.	L2 diagnosed false positives during the run as ASE was scheduling DCE and X-band activities concurrently.
ASE R3 #1	49 hours. Day-of-year (DOY) 2005-012	No problems with L2.	Success.
ASE R3 #2	23 days. DOY 2005-106 to 2005-129	No problems with L2.	Success for L2. ASE system was halted by an anomaly, requiring a cold start of the WARP processor. The problem was caused by the removal of a file from the ramdisk during processing by CASPER.
ASE R3 #3	41 days. DOY 2005-140 to 2005-181	No problems with L2.	Success. ASE system was halted to perform a Formation Flying maintenance maneuver and Pulsed Plasma Thruster (PPT) operations.
ASE R3 #4	55 days. DOY 2005-192 to 2005-246	False Positives. No other issues reported.	L2 diagnosed false positives at the end of the run as SCL failed to notify L2 of a command that was executed in a timely manner. SCL was delayed due to CPU starvation.
TOTAL:	143 days		

For extended run ASE R2 #1, ASE was not turning L2 off when a supported DCE activity overlapped with an unsupported X-band activity. Specifically, a data collect commenced before loss of signal for an X-band contact. This was a known issue for R2, as L2 did not receive X-band commands in R2, but was receiving the warp telemetry changes which result from X-bands and data collects. Normally, X-bands are completely masked for L2. However, when a DCE overlaps with an X-band activity, the WARP telemetry resulting from the X-band is seen but the initiating commands are not, resulting in a false positive.

In the ASE R3 #4 false positive, the problem turned out not to be L2. L2 behaved correctly with the information it received. From the log files, the effect of a command to enable power to the ALI mechanism was seen before the command itself. This is a timing issue with SCL, which notifies L2 of commands and telemetry responses. In this case, SCL sent the command and its resultant observation out of order:

observation,test.ali.IHSKP_MTR_PWR,ENABLED,1125804977,967542278
command,test.ali.mechanismPower.command,I_MECHPOWER,1125804978,276990646

The cause was CPU starvation; an SCL task got starved due to the heavy processor load, and so did not inform L2 of the command to turn on the ALI mechanism power in time, before the motor power ENABLED effect was seen. After playing back the CPU load telemetry at the time of the false positive, CPU usage was at 100% during the time that SCL delayed notification of the command to L2. CPU starvation of ASE has been noted in the past several times; root cause has not yet been determined however the low CPU availability margin is a known issue.

The detail of the mechanism at work is that L2 must be notified by the rule that monitors changes on test_ali_mechanismPower_command. Once the SCL i_mechpower_cmd script issues an embedded (i.e. pre-compiled) command, it will almost certainly be pre-empted by a higher priority task before setting the test_ali_mechanismPower_command database item to I_MECHPOWER. Telemetry processing is done at a higher priority than script execution, so although the outgoing command telemetry is unaffected, it is the notification to L2 that is getting delayed because the task executing i_mechpower_cmd is getting starved

due to processor load. This is an unreliable notification mechanism: if this experiment was continued, the SCL preemption mechanism would be reexamined.

By comparison with L2's cumulative operations time in space of 143 days, L1 on Deep Space One ran for about a day. The RAX 2-day scenario ran well for 20 hours before it was noted that plan execution had ceased. Troubleshooting tracked the cause of the problem to a race condition between two threads in the executive (EXEC). The follow-up 6-hour scenario experienced loss of a critical monitor value, somewhere between the DS1 flight software and RAXM (the RAX Manager). This led L1's estimation of the thruster state to diverge from the true state. The discrepancy proved to be benign and execution of the scenario was completed.

The long flight times achieved by L2 on EO-1 attest to the maturity of the diagnostic engine and robust handling of real-time issues in the model and the RTI. The L2 software appears to be free of race conditions, dead-locks and memory leaks. With the exception of the noted timing error, the SCL software bridge has also functioned well, with no lost monitor values.

3 Overview of GSFC Fault Protection Technology

Missions operated by Goddard typically use a combination of Telemetry Statistical Monitors (TSMs) and Fault Detection and Correction (FDC) for fault protection. These operate at the spacecraft or system and subsystem level respectively. This discussion is in the context of the Power Subsystem Electronics (PSE) [11], a subsystem on EO-1. How L2 improves upon TSM/FDC shortcomings is later discussed.

3.1 Automated Spacecraft Response: Telemetry Statistical Monitors (TSMs)

The Telemetry Statistical Monitor (TSM) system is resident within the Mongoose V Command and Data Handling (C&DH) Processor. It provides fault protection across all subsystems, including PSE. Two levels of automatic protection are handled by TSMs. If these first two levels of protection do not prevent the progress of the fault, a third level of automatic protection is provided by the FDC actions at the PSE subsystem Remote Service Node (RSN). There are four TSMs and RTS responses in the automatic protection system for power. The TSM actions will take place when out of limit conditions are experienced for Battery Differential Voltage, Battery Low Voltage, Battery High Temperature, and Battery Low State of Charge. For example, TSM #050 checks for low Battery State of Charge (BSOC):

If BSOC < 85% issue event message

If BSOC > 30% and BSOC < 70% call Safe Power (RTS #19)

When the parameter being monitored by the TSM exceeds the first (most sensitive) level of limit violation, the TSM will issue a limit violation event message after three consecutive violations. If the parameter telemetry value continues to degrade and the out of limit reading exceeds the second limit level (more severe violation), the TSM will initiate a series of preprogrammed commands to attempt to protect the spacecraft and correct the limit violation. This series of pre-programmed commands is a Relative Timed Sequence (RTS). RTSs are designated individual sequence numbers to identify them; for TSM #050, RTS #19 initiates the loadshed and safehold sequence.

3.2 Automated Subsystem Response: Fault Detection and Correction (FDC)

FDCs provide health management local to each subsystem. The PSE RSN processor continuously monitors the power system spacecraft telemetry and automatically responds to certain telemetry limit violations in a similar but not identical fashion as the automated TSM response. The limits for the FDC are set to more forgiving levels than the stricter TSM limits, the idea being that the FDC will kick in should the TSM fail to act. The logic flow chart for the 3 PSE FDCs is shown in Figure 1 below.

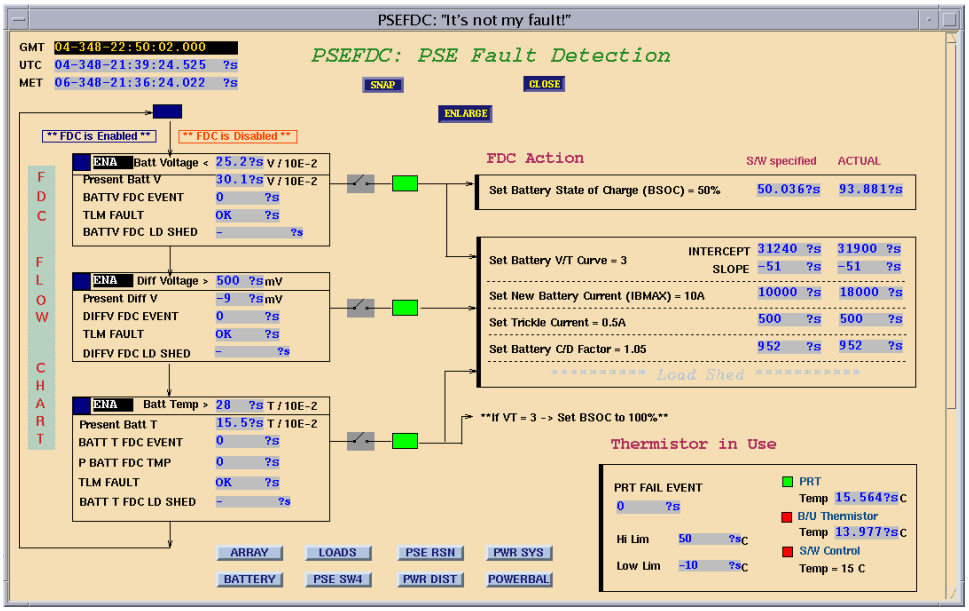

Figure 1: EO-1 FDCs for the Power Subsystem Electronics

TSMs, with their stricter redlines, normally perform all loadshedding and safehold, since the PSE FDCs cannot load shed other subsystems. The PSE FDCs serve as a backup should the TSMs fail to trip. Each trigger on the left is linked to the corrective action(s) on the right. For example, the first FDC corresponds to the low BSOC TSM #050. This FDC will trigger two actions if battery voltage drops below 25.2V: first setting BSOC to 50% to force triggering of TSM #050 if it did not already trip for some reason. Second, loadshedding local to PSE is implemented by lowering the battery charge rate by setting the Voltage/Temperature curve to 3 (normally the charge rate is 4.5). In addition, several parameters are set to their normal values: battery maximum current to 10A to prevent overcharging; trickle current set to nominal 0.5A; and the battery C/D (charge/discharge) factor is set to 1.05, reflecting the fact that the battery is not perfectly efficient.

3.3 Evaluation of the TSM / FDC Approach

Ground Operations for EO-1 are conducted at the GSFC Mission Operation Center (MOC). The command and control software in the MOC is the Advanced Spacecraft Integration and System Test (ASIST). The matrix of all TSMs flying on EO-1 is shown in the ASIST workstation display in Figure 2.

Advantages of the TSM approach are:

- TSMs are simple to write and inspect, as they usually only operate on one telemetry point (also be seen as a limitation).
- TSMs can be individually enabled and disabled. If a TSM often generates false positives, it can be disabled until a patch to the spacecraft software can be uploaded. This can also be used to allow the operators of the spacecraft to perform actions that were not envisioned at design time, but are later determined to be necessary and safe. For example, one TSM is tripped when the spacecraft instrument covers are open and the spacecraft begins to point toward the sun, which will damage the instruments. However, this maneuver is allowable when the spacecraft is in the Earth's shadow. When the maneuver is necessary, this TSM is disabled, then re-enabled after the maneuver is complete.

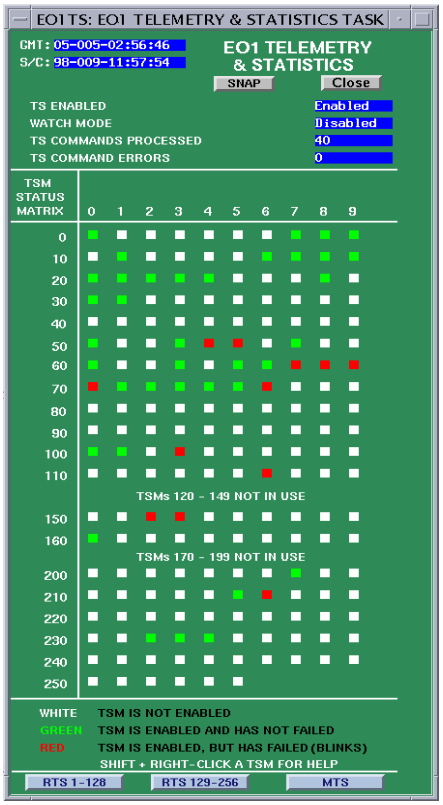

Figure 2: EO-1 TSM Matrix

Weaknesses of the TSM approach:

- TSMs do not identify *root cause*. A common failure e.g. power, would cause the display to light up like a Christmas tree, as in the multiple red fault indicators in this display.
- TSMs do not detect *sensor failures*.
- TSMs do not show *ambiguity* i.e. alternate failures which exhibit the same symptoms, such as a component or its associated sensor failing.
- TSMs do not indicate the *probability* of the diagnosis, where there is ambiguity as to the actual fault.
- TSMs do not indicate the *time* the failure occurred.
- TSMs do not monitor across *multiple telemetry points*. Each TSM is defined for a single telemetry point, with linkages between TSMs manually inserted.
- No cross-checking of the TSMs is performed to ensure *global consistency*.
- The number of TSMs implemented for EO-1 was limited due to development effort and cost. *Scalability* is a challenge for rule-based systems, due to the lack of structure in the set of fault protection rules.
- *Non-intuitive user interface* – unless the operator knows the TSM matrix very well, it does not provide insight at a glance. The operator must examine each effected TSM to determine the problem.

Advantages of the FDC approach are:

- *Redundancy* - the FDC automated response was designed so that spacecraft protection was still available from the subsystem RSN in the event the TSM protection at C&DH was malfunctioning.

Weaknesses of the FDC approach are:

- *As for TSMs* - since FDCs are similar to TSMs, FDCs suffer from many of the same limitations as TSMs described above.
- Potential for *TSM/FDC conflict* - it is quite possible that an FDC limit violation may occur after a valid TSM limit response was issued. In this case TSM and FDC responses will be operating in parallel. The FDC and TSM responses had to be designed so that there are no conflicts between the two systems.
- *System-wide response* - FDC corrective actions are limited to the scope of the subsystem, and any impact on other subsystems is handled by TSMs. If system-level response is required, a TSM must be tripped to initiate wider recovery of the spacecraft. In this way the FDC which sets BSOC = 50% will trip TSM #50 to do load shedding and safehold. This relationship is implicit and if improperly managed could result in unexpected behavior with potentially severe consequences.

4 Value Added by the L2 Model-Based Diagnosis Approach

The TSM/FDC fault protection technology has several weaknesses which have been identified in the previous section. This section discusses L2 solutions to these issues, both in the context of the EO-1 experiment and of previous work [3].

4.1 *Root Cause Analysis:*

Whereas TSMs and FDCs do not perform *root cause analysis*, L2 has the capability to determine the root cause of the fault, exonerating dependent components which exhibit related symptoms but have not

themselves failed. TSMs/FDCs on the other hand are rule-based. It is possible to construct a higher-level rule that reasons over a set of rules, to figure out common cause. However the individual TSMs would still fire and present as multiple faults, since exoneration of a fault requires additional reasoning. This is all handled naturally by L2, with no special support required by the model. The analysis is encoded in the domain-independent L2 algorithm and involves reasoning about the relative likelihood of multiple simultaneous faults and their interdependencies.

Root cause analysis is demonstrated in the L2 test scenarios for EO-1, for example in one of the imaging component main aperture cover models. The ALI aperture cover opens and closes to allow the component to take an image. The aperture cover and its supply power are both modeled. If there is a failure in the supply power component, then the aperture cover will not move when commanded, although it is working nominally. This corresponds to experiment fault scenario #5. As it is expected that the cover will not move when there is no power, L2 does not implicate the aperture cover in this scenario and instead diagnoses the failed supply power component.

In the event that there is no root cause, L2 is also able to diagnose multiple faults, occurring simultaneously or separately. The diagnostic scenarios #8 and #9 show this feature of L2. They each contain two candidates, one of which contains multiple faults.

4.2 *Diagnosis under Uncertainty:*

In a complex system with multiple points of failure, there is always uncertainty - due to noise in the sensor data, and due to restricted observability. Spacecraft tend to be under-sensed, from a health management perspective, to keep weight down at launch. L2 is resilient to gaps in sensor coverage, able to track the state of the spacecraft and diagnose faults even if missing some sensor information. This is due to the fact that L2 uses a conflict-driven search, triggered by evidence of conflict with predicted behavior. The Real-Time Interface (RTI) and Transient Modeling methodology developed for EO-1 [1] also support continued mode estimation during transients, prior to receiving sensor responses to commands or whilst dynamical transients in the physical system are settling down.

Ambiguity is represented by L2 in an ambiguity group of multiple diagnostic candidates, with a likelihood assigned to each candidate. Most of the diagnostic scenarios of the experiment involve multiple candidates. Usually this expresses the possibility that a component could have failed, or a sensor monitoring the component failed. Since all sensors are modeled with at least one failure mode (indeed all components are), detection of *sensor failures* is well-supported by L2.

L2 also maintains a history of alternate trajectories, which allows fault hypotheses to be revised given new information. Two of the fault scenarios, #11 and #12, illustrate this. The initial diagnosis of each contained two candidates. As the scenario progressed, one of the candidates was found to be inconsistent with later observations and eliminated from the diagnosis, improving fault isolation.

For decision support, to enact recovery or to schedule maintenance for rapid turnaround, L2 reports the *likelihood of each diagnostic candidate* in an ambiguity group. This is a relative ranking based on *a priori* component failure probabilities. Where a single candidate contains multiple failures, the rankings are summed to give the rank for the candidate. In addition, the *time of the failure* is reported.

4.3 *Model-Based Operations:*

Another advantage of the L2 model-based diagnosis approach is that it lends itself to model-based operations. Taken to the limit, model-based operations could encompass ground and flight segments as well as the communication links between them. The flight segment may be expanded to satellite constellations such as the EO-1 and Landsat-7 formation. For this experiment, only flight operations of the EO-1 spacecraft were in the scope of the model.

The strength of model-based operations is that the application, procedures or software which implement the required functionality are generic and not domain-specific. There is a separate model or set of models which describes the instance to which the software applies and must reason about. This model is specific to the domain, such as a spacecraft, launch vehicle, or ground station. Model-based approaches have been developed both for planning and diagnosis, providing control and feedback for operations.

For EO-1, a single model was used for the spacecraft, with the subsystems contained as modules therein. Unlike TSMs which have no cross-checking to ensure *global consistency*, the constraints within the L2 model enforce consistency checks both within the subsystem and across subsystems at the system level.

Unlike TSMs which generally are defined for a single telemetry point, L2 uses a first-principles diagnostic model of the system, which has broader scope and models the reality of *multiple telemetry points* associated with components in the model.

Scalability is an advantage for L2; a single, consistent L2 model conveniently encapsulates the error checking functionality of several TSMs. Scalability has not been proven for L2 by modeling a very large system using only L2 and only rules, and comparing the two approaches. However, scalability is known to be a weakness of rule-based systems. For example, on EO-1 only a restricted subset of TSMs was implemented as the originally required set demanded greater engineering effort than could be afforded. Ongoing work is underway with International Space Station (ISS), modeling the C&DH subsystem using L2 [12]. As the ISS is a large-scale complex system, this domain will serve as a stress test of L2's capability to scale-up to large systems.

Currently on EO-1, corrective actions may be taken by both TSMs and FDCs. Interactions between TSMs and FDCs are not automatically tracked or managed, making recovery procedures difficult to verify. The L2 deployment on EO-1 with a single model does not have the duplication of TSM and FDCs. This avoids the potential for *TSM/FDC conflict*, a kind of race condition in which both the system and subsystem are attempting to resolve a problem. However, the single model approach does not have redundancy in the event of malfunction of ASE/L2 fault protection or communication with ASE/L2. It is possible to break up an L2 model into constituent subsystems and deploy each of these separately, with the subsystems communicating status to the system-level model. A distributed application of L2 has yet to be developed. Performance may require a distributed fault protection architecture. For instance, in a hard real-time critical subsystem requiring millisecond response, fault detection, isolation and recovery should be local to the subsystem. If a system-wide response to a subsystem fault is required, the system-level implements fault protection on receipt of the subsystem fault information.

In contrast with the TSM matrix used on EO-1 (Figure 2), L2 supports an *intuitive user-interface*. The L2 presentation of a fault can use a schematic of the spacecraft as shown in Figure 3 below. In this example for the X-34 propulsion feed system, the red valve component has failed. This type of situation display is very easy to grasp and allows quick situation assessment. For the current deployment on EO-1, this was not implemented as the L2 Ground Processing Unit (GPU), which provides this graphical display, was not integrated into the MOC. The L2 telemetry page used instead on EO-1 is presented in [1].

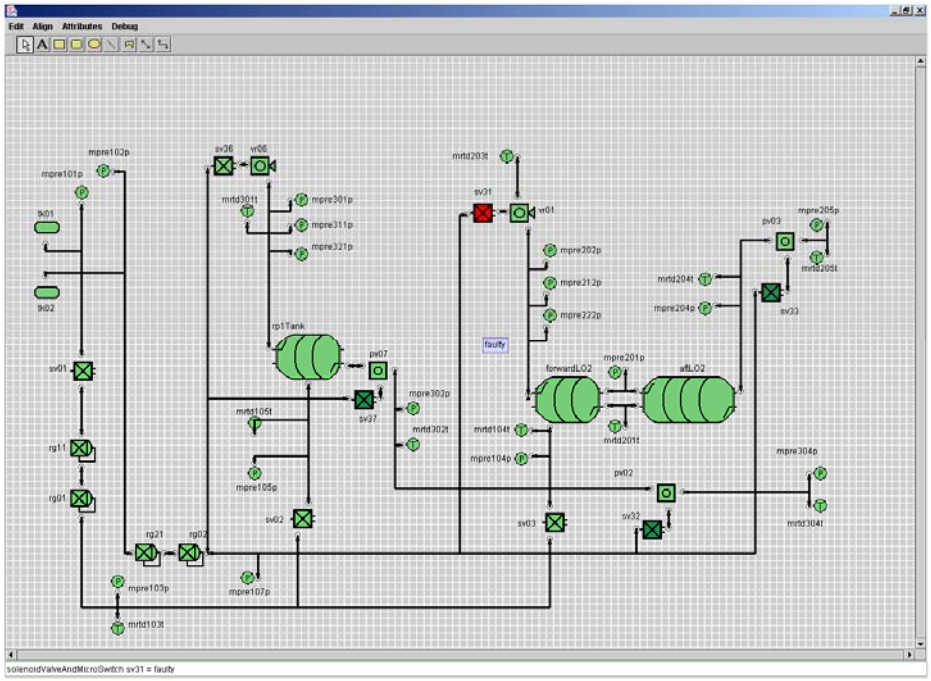

Figure 3: L2 Diagnostic Display of X-34 Feed System

In addition to flagging failed components, L2 monitors nominal mode transitions of components of the system, and these mode changes can also be shown on the schematic display. Mode transitions are

included in the EO-1 telemetry downlink, and the L2 telemetry page displays the mode of the last component which transitioned. This may mean that should several transitions occur quickly, only the last transition is visible on the stack. A history stack of mode transitions was not downlinked to conserve telemetry bandwidth.

L2 also has an explanation capability which can be queried for causal chains of inference, justifying the diagnosis. This is not supported by the current EO-1 deployment, as explanations are provided by the L2 debugging classes which are not part of the flight code. In addition, explanation telemetry requires a reasonable amount of bandwidth as well as definition of a new telemetry packet, of which only a limited number are available, as well as requiring support for uplinking queries from ground operators.

5 Weaknesses of the L2 Model-Based Diagnosis Approach

Previously, general strengths and weaknesses of L2 were identified in [3]. The deployment on EO-1 exposed several areas of improvement for L2. Lessons learned and known weaknesses are categorized in the areas of diagnosis, fault coverage, recovery and operations support.

5.1 Diagnosis:

a) L2 supports discrete models only, with no support for continuous modeling. L2 can diagnose continuous systems, such as a tank being overfilled, by modeling a discrete abstraction of the system and externalizing the continuous processing in the monitor software. However this becomes cumbersome if a significant amount of continuous behavior must be captured to diagnose the system, with the binning strategy and interface between model and monitors requiring careful design.

This limitation affected the scope of the EO1 model. The components chosen for inclusion in the model consisted almost entirely of those with discrete behavior. Although the spacecraft is largely a discrete system with discrete telemetry, attitude control, power and the HSI cryocooler exhibit continuous behavior, and as such were deemed out of scope.

b) L2 does not support autonomous mode transitions, which are spontaneous mode changes not initiated by controller commands. For example, a relief valve will open when its crack pressure is exceeded, with no controller command. The design decision not to support autonomous mode transitions had benefits of simplicity and speed. However, to detect certain changes, a more general conditional trigger is required that is based on both commands and observations and initiates the transition whenever the component is in that mode and the conditional expression evaluates to true. Autonomous mode transitions were not encountered in the scope of the EO-1 model.

c) L2 does not support modeling of controllers, specifically prediction of the commands that they issue. Several EO-1 instruments, such as the ALI, have on-board controllers which perform requested services such as lamp calibration. The problem is that their internal commands do not go out on the spacecraft bus, and lack of visibility to internal commanding impairs L2's ability to predict behavior of the component. To address this, either L2 must have the ability to predict the hidden controller commands (e.g. from a command which initiates the sequence) or the system needs to be designed to give diagnosis tools the observability they require. However, it is common for diagnosis to be developed after the spacecraft's design, rather than the best practice of designing-in diagnosis as an early and integral part of the development process. For advanced diagnosis research tools such as L2, this is even more so the case, being integrated into the spacecraft after deployment. A workaround for these issues for L2 on EO-1 was provided by SCL generation of the internal command sequences.

d) There is no support for timed transitions in the L2 model, to allow for real-time delays in issuing a command and acquiring the response sensor data. Currently a combination of Real-Time Interface (RTI) software and the transient modeling methodology supports this. This weakness increased the amount of development required to deploy L2 on EO-1. The RTI software from PITEX [8] was revised for EO-1 to support timeouts for real-time delays, and the transient modeling methodology requires many additional transient modes and transitions in the model.

5.2 Fault coverage:

a) L2 is able to identify abrupt faults, where the evidence of the fault is either immediately apparent or is revealed by a command e.g. opening of the stuck valve. Most of the faults on EO-1 are of this nature. L2 has no support for prognosis of incipient faults, although this could be provided by external monitors. There is also no support for identifying cascading faults, where the initial fault is causally linked to subsequent faults e.g. a regulator fails and the pressure surge blows a downstream relief valve / O-ring; or domestic object damage to nearby objects should a turbine blade be thrown. Fault modes involving prognosis and cascading faults were not identified on EO-1.

b) To date, no demonstration of hard real-time fault diagnosis (on the order of milliseconds) has been made by an integrated L2 application. This was also not achievable on EO-1 due to saturation of the 8 MIPS CPU. On EO-1, sensor sampling rates range from 1 to 4 seconds, with Hyperion telemetery at 1 second and WARP/ALI telemetry at 4 second intervals. However, telemetry processing by the SCL software bridge often lags, causing L2 and ASE to utilize much longer real-time delays to await incoming telemetry. These delays range from 8 to 40 seconds for L2. Telemetry processing lags are due to the fact that the WARP CPU is severely constrained, flying both the WARP flight software and the ASE/L2 experiments. Soft real-time performance can be claimed for the experiment, as we can guarantee meeting these longer delays.

5.3 Recovery:

a) L2 has no support for recognizing spontaneous recovery from an intermittent fault, as this implies an autonomous transition from an anomalous to a nominal mode e.g a wire shorting on contact under vibration.

b) L2 models have not yet been developed that can recognize that functionality has been preserved through switching to a backup on failure of a component, even though the component remains failed. This might be accomplished through a functional model. This was not explored on EO-1 as the spacecraft is essentially single string.

c) L2 models do not indicate severity or time to criticality of a fault. Impact, time to criticality and probability all influence the recovery action to be selected and enacted. It may be possible to devise a model which captures this, but L2 has no built-in representation or reasoning about the severity of faults.

d) L2 has limited ability to recognize transition back to nominal operations, in the case of fault recovery through human or controller intervention. Supported recoveries must be commanded and recovery transitions explicitly modeled. In addition, this recovery recommendation is generated by a separate instance of L2. Since this adds significant complexity to the architecture and integration, and little redundancy exists on the EO-1 spacecraft to demonstrate recovery, this functionality was not flown on EO-1.

5.4 Operations support:

a) L2 has the limitation that the initial configuration of the model, say $S_0 = \{s_0, s_1, \ldots s_n\}$ where s_n is the state of component/subsystem *n*, must match the initial configuration of the target system at startup. There is no ability to discover the initial state of the system from sensor observations. In fairness, to determine the initial state from sensor observations would require that the system be fully instrumented, which for an under-sensed system is an indeterminate problem. Also, starting from a known state has the benefit that mode identification is resilient to unknown sensor values, identified as a strength of L2. The weakness must be traded-off against the strength it enables.

To accommodate, the initial state of the L2 model was chosen to be EO-1's "idle" configuration. When running the L2 experiment scenarios, checks in the L2 startup procedure used by the EO1 flight operations team allow L2 to start successfully only if the satellite is in the idle mode. If L2 were to be started when the satellite is in a different mode, false positives would result as the satellite would be making state transitions not possible from the initial mode.

b) On EO-1, L2 only acted as a monitoring and diagnosis agent. The current modes of the system are tracked and whether faults have occurred or not, but L2 gives no indication of the criticality of faults nor recommendations to recover from them. L2 does not clear fault alerts, since the recovery instance of L2 is not flying on EO-1. Once a component has failed, L2 will report that a component fault occurred at time *t* until L2 is shut down, even if the fault was recovered from or a backup system was activated. The only circumstance that the diagnosis is cleared is in the event of the

search space becoming exhausted. This may occur in a situation of massive failure, in which case there is no diagnosis which can explain the multiple simultaneous faults within the parameters of the search algorithm (including maximum likelihood).

6 Conclusion

This experiment has successfully deployed Livingstone 2 on-board EO-1. By meeting the minimum success requirements, L2 has demonstrated advances over previous work, most significantly:

- Capability to track multiple diagnostic hypotheses and revise hypotheses given new evidence, important in any complex system;
- Capability to monitor the spacecraft state and diagnose faults during transients, both under partial observability (before telemetry responses are seen) and whilst the physical dynamics of the system are settling out.

After meeting these criteria, work continued on to verify a series of scenarios and give complete coverage of the imaging model. The number of scenarios tested was increased, from 2 for RAX to 17 for the L2 on EO-1 experiment. L2 was shown to be capable of long-term space operations, with the longest extended run thus far of 55 days with no false positives. Cumulative operations time in space is at least 143 days. This is a significant increase over Livingstone's previous longest flight time of 20 hours, achieved by Remote Agent on Deep Space One.

Finally, L2 has demonstrated that it has potential to add-value to flight operations in general, and in particular that it has the ability to improve on several weaknesses identified in the TSM/FDC fault protection approach. Major strengths are:

- Deeper diagnosability, including root cause analysis and reasoning about ambiguity.
- Model-based operations can be grown from model-based diagnosis.

Weaknesses in the L2 approach have also been identified, and these form new requirements for the next-generation Hybrid Diagnostic Engine (HyDE), which is currently under development at Ames.

7 References

1. S. Hayden, A. Sweet, S. Christa, D. Tran, S. Shulman. "Advanced Diagnostic System on Earth Observing One". *Proceedings of AIAA Space 2004 Conference and Exhibit, San Diego, California, Sep. 28-30, 2004.*
2. S. Hayden, A. Sweet and S. Christa, "Livingstone Model-Based Diagnosis of Earth Observing One". *Proceedings of AIAA 1st Intelligent Systems Conference*, Chicago, 2004.
3. Sweet, A. and Bajwa, A. "Lessons Learned from Using a Livingstone Model to Diagnose a Main Propulsion System", *JANNAF 3rd Modeling and Simulation Subcommittee Meeting*, Colorado Springs, CO. (Dec 2003).
4. J. Kurien and P. Nayak. "Back to the future for consistency-based trajectory tracking". *Proceedings of the 7th National Conference on Artificial Intelligence* (AAAI'2000), 2000.
5. N. Muscettola, P. Nayak, B. Pell and B. Williams, "Remote Agent: To Boldly Go Where No AI System Has Gone Before". *Artificial Intelligence,* Vol 100, Best of IJCAI 97
6. S. Chien, R. Sherwood, D. Tran, R. Castano, et al., "Autonomous Science on the EO-1 Mission". *Proceedings of the Seventh International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Nara, Japan, 2003.
7. B. Williams and P. Nayak, "A Model-based Approach to Reactive Self-Configuring Systems". *Proceedings of Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, Portland, Oregon, 1996.
8. Meyer C., Cannon H., et al, "Propulsion IVHM Technology Experiment Overview", *IEEE Aerospace Conference*, Big Sky, MT (2003).
9. M. Schwabacher, J. Samuels, and L. Brownston. "The NASA Integrated Vehicle Health Management Technology Experiment for X-37", SPIE AeroSense 2002, Florida.
10. Livingstone 2 Open Source at http://opensource.arc.nasa.gov/

11. NASA Goddard Space Flight Center, “Earth Observing-1 Contingency Document Section 4: EO-1 Power Balance Contingencies”, November 2000.
12. Peter Robinson, Charles Lee, et al, “Strider: Towards Model-based Mission Operations for the International Space Station and Beyond”, to be published in *IEEE Aerospace Conference*, Big Sky, MT (2005).

8 Biographies

Sandra Hayden is a Group Manager/Computer Scientist for a government contractor, QSS Group Inc, working in the Intelligent Systems Division at NASA Ames Research Center. Since ’89, she has worked on software engineering of large-scale safety/mission-critical systems demanding high reliability, such as Canadian Automated Air Traffic System (CAATS), the next-generation distributed air-traffic control system now deployed to cover all of Canadian airspace (a million plus lines of Ada). She was principal investigator/project manager of the Livingstone 2 (L2) on Earth Observing One (EO-1) flight experiment, which received the 2005 NASA Group Achievement Award. L2 demonstrated model-based monitoring and diagnosis of EO-1 under autonomous control of a planner during earth imaging activities, as well as long-term operations in space.

She participated in Integrated Vehicle Health Management (IVHM) design and development, in the Next Generation Launch Technology and Space Launch Initiative programs. As a member of the NASA/Honeywell Architecture Team, she helped define the IVHM architecture and worked with Pratt and Whitney on a demonstration. Other projects include health management for the X-34 main propulsion system, under the NITEX project, and its successor the PITEX project, which she managed at Ames for a year during which the transient real-time interface was designed and implemented; the WhiteSands non-toxic reaction control system was modeled and automated software engineering V&V technology was applied to model-based diagnosis. She defined system requirements, architecture and interfaces for the VxWorks real-time flight system and the ground processing unit. Other experience includes a submarine’s 1553 system data bus and operations software for undersea diamond mining. Founded the contractor Software Process Improvement Network and conducted CMMi pre-appraisals of selected projects. She has an MS in Computer Science from Simon Fraser University, Vancouver, Canada.

Adam Sweet is a research engineer in the Computational Sciences Division at NASA Ames Research Center, under contract to QSS Group Inc. He graduated with an MS in Mechanical Engineering from UC Berkeley in 1999, and has since worked at Ames modeling and simulating physical systems. His focus has been in robotics, hybrid system simulation, and model-based diagnosis.

Seth Shulman is the EO-1 Flight Operations Team (FOT) Technical Lead at NASA GSFC under contract to Honeywell Technology Solutions Inc. (HTSI). He graduated with a BS in Electrical Engineering from the University of Maryland and has worked since 1987 at GSFC supporting both the Flight Dynamics Facility (FDF) and EO-1 Mission Operations Center (MOC). His focus has been in Gravity Modeling, Flight Dynamics, and Flight Operations. Other missions supported include: Upper Atmosphere Research Satellite (UARS), Extreme Ultraviolet Explorer (EUVE), X-Ray Timing Explorer (XTE), Tropical Rainfall Measurement Mission (TRMM), Landsat-7 and Total Ozone Mapping Spectrometer - Earth Probe (TOMS-EP).