

# Using Measures of Effectiveness to Analyze and Improve Mission Operations

Trevor C. Sorensen<sup>\*</sup>

*University of Hawaii at Manoa, Honolulu, HI, 96822*

*and*

Richard R. Champion<sup>†</sup> and Todd C. Probert<sup>‡</sup>

*Honeywell Technology Solutions, Inc., Columbia, MD, 21046*

**A major factor in the cost of spacecraft ground support is the effectiveness of the mission operations process. An ineffective, error-prone and labor intensive process will most likely result in increased cost, risk, and reduced customer satisfaction. In order to determine the effectiveness of how mission operations are performed and to determine areas of improvement, measures of effectiveness should be identified. The metrics obtained through these measures of effectiveness can then be empirically and subjectively analyzed to determine the areas of the operation that should be improved or automated to increase efficiency. For a science mission, effectiveness factors for the mission operations include percentage completion of science objectives; cost of operations (comparison of actual versus projected costs); response time and flexibility of the mission planning an operations process; and efficiency (cost/data collected). Some metrics that can help measure the effectiveness of science mission operations were also identified and discussed. To effectively generate, track, and use these metrics, they should be incorporated into the mission operations process. To test the methodology and practicality of obtaining and using this process, three recent space missions were analyzed, and the results reported within this paper. An example of where this**

---

<sup>\*</sup> Professor, Hawaii Space Flight Laboratory, 1680 East-West Road POST 501, Associate Fellow AIAA.

<sup>†</sup> DataLynx Mission Operations Manager, M.S. A-116, Member AIAA.

<sup>‡</sup> Program Director, Near Earth Networks Services, 7515 Mission Dr., Lanham, MD 20706.

**operations improvement process was incorporated into the design of a ground segment is also discussed.**

## **I. Introduction**

A famous general (attributed to Prussian Field Marshal Helmuth von Moltke) once said “No plan survives first contact with the enemy.” This is also true for spacecraft operations. Even the most carefully planned mission operations or support plan will not survive first contact with reality. If the mission operations system has not been designed with the flexibility and built-in processes to recognize problems or anomalies, analyze them, and provide a feedback loop to introduce improvements back into the mission operations process, then it will be very difficult and costly for the system to adapt. It is far better and cost effective to design these process improvement features into the system than to try re-engineering the system after launch. There are many case histories where this is true. Several missions, including the Hubble Space Telescope, have attempted to introduce automation and process improvements into the system after launch, and have had a very difficult time doing so.[ref] It is difficult to do this without disrupting or risking the ongoing operations, and when it is possible, it is usually very costly. The golden rule of process improvement is: design the process of process improvement into the system from the very beginning so that it will appear in the design requirements.

This paper presents some means that can be used for helping with the mission operations process improvement, from the determination of suitable metrics, methods to collect and analyze them, determine solutions, and then feed the solutions back into the system. Although the actual metrics and methods that are best suited for a particular mission might be different, the general principles stated herein are the results of experience obtained on several missions, including Low-Power Atmospheric Compensation Experiment (LACE), Clementine, MSTI-3, and others. The process improvement methodology described in this paper was implemented into the design of Honeywell’s DataLynx Satellite Control Network, which is discussed in Section IV.

## **II. Defining and Using Measures of Effectiveness as Metrics for Process Improvement**

A major factor in the cost of spacecraft ground support is the effectiveness of the mission operations process. An ineffective, error-prone and labor intensive process will most likely result in increased cost, risk, and reduced customer satisfaction. In order to determine the effectiveness of how mission operations are performed and to determine areas of improvement, measures of effectiveness (MoEs) should be identified. The metrics obtained

through these measures of effectiveness can then be empirically and subjectively analyzed to determine the areas of the operation that should be improved or automated to increase efficiency.

For a science mission, effectiveness factors for the mission operations include:

- 1) Percentage completion of science objectives (e.g., number of science experiments successfully executed, coverage obtained by imaging, quality of data, quality and quantity of calibration data obtained)
- 2) Cost of operations (comparison of actual versus projected costs)
- 3) Response time and flexibility of the mission planning and operations process
- 4) Efficiency (cost/data collected)

#### **A. Definition of Measures of Effectiveness for Mission Operations**

Some metrics that can help measure the effectiveness of science mission operations include: error tracking, exceptions (complexity) factor, rush factor, effort factor, response factor, fatigue factor, and morale factor. These MoEs were first identified in post-mission analysis of the Clementine lunar mission and were very useful in determining where the mission operations process was successful and where it needed improvement. They were subsequently used in the analysis of other historical missions before being designed into a recently operational commercial mission operations system (Honeywell's DataLynx).

##### *MoE #1: Error Tracking.*

This MoE tracks all the ground source errors that reach the spacecraft during the mission (although we are using the spacecraft as the end "victim" system, this MoE could be equally applied to other systems that receive external data that could cause errors in its execution). Most of the errors that reach the spacecraft are generated by the mission operations process or allowed to pass through it to the spacecraft. Spacecraft commanding and operations errors that affect accomplishment of mission goals may include:

- Planning and timeline/schedule errors – these are the errors introduced in the first steps of the mission operations process before actual commands are generated. For example, a timeline or schedule might direct that the spacecraft to go into a data dump mode before the tracking station is in view. The source of this error is usually human (the mission planner), but could also be a result of incorrect mission rules (requirements), an experiment design fault, or use of erroneous data, such as an out-of-date ephemeris.

- Command script/sequence errors – these are errors that are introduced after taking a timeline or schedule and turning it into a command sequence (although usually still in a human-readable rather than spacecraft readable form). The source of these errors is also usually human. They are especially likely to occur if a manual copy r cut and paste method is used to convert the timeline into a command script. This area is particular suited for automation or constraint checking.
- Instrument or spacecraft pointing errors – these are errors in determining or specifying the correct direction to point some apparatus on the spacecraft, whether an instrument, an antenna, or the spacecraft bus itself. The source of these errors is usually human or software. A pointing error can be introduced from the mission or experiment plan formulation phase all the way through the generation of the command script.
- Commands/script testing errors – many command scripts, after translation in the machine-readable form, are tested on a software simulator or a software/hardware testbed. Sometimes discrepancies between the planned command sequence as expressed on the timeline or script and the actually executed command script escape the notice of the testers, whether human or computer. However, sometimes command errors can even be introduced in this phase as “corrections” to the command script without full realization of the consequences of the changes. The testers might also have an erroneous configuration set up which does not match that which the command script will see on the spacecraft. This is one of the errors that resulted in the spin- up failure of the Clementine spacecraft that caused the loss of the asteroid encounter of the mission.
- Ground system errors – after the script has been tested it is passed along to the real-time or ground operations subsystem for delivery to the ground station for upload to the spacecraft. Errors can occur in this process (e.g., the wrong file is sent or at the wrong time). Included in the ground system errors are any errors that occur at the ground stations (hardware, software, and personnel errors). Hardware outages such as a transmitter or receiver failure at the ground station can affect the FOT's ability to send and collect data from the spacecraft.
- Real-time operations errors – any real-time commanding of the spacecraft during a pass or contact is prone to human errors, especially if constraint and command checking is not provided in the real-time commanding software.
- Spacecraft hardware errors – these are errors caused by faults in the onboard hardware of the spacecraft and are sometimes beyond the control of the ground operations personnel. However, many times problems with the

onboard hardware can be resolved either by using workarounds or by making adjustments to the onboard system or configurations.

- Software errors (ground and flight) – this can be a major source of errors, especially in the initial phase of a mission before the system reaches a certain level of maturity. The “faster, better, cheaper” missions, because of their fast-track development cycle, are often launched before the ground or space software has been fully completed and tested. These missions often rely on a certain basic level of software to the basic essential operation of the spacecraft, but rely on software developed and tested during the mission itself for implementation of higher or more sophisticated functions. The use of software that is not fully developed and tested on an operational spacecraft can have dire consequences (e.g., the “spin-up” and effective loss of Clementine while testing some new asteroid encounter software—this was in conjunction with the testing error described earlier).
- Miscellaneous errors (communication links, ground segment hardware) – this is a catchall category of unlikely or rare sources of errors. If any of these elements become a significant source of errors (e.g., communication link), then it should probably tracked as a separate error. These errors can be either human or machine.

#### *MoE #2: Complexity/Exceptions Factor*

This MoE is a measure of the complexity of a mission “event” (e.g., pass, observation, or experiment). If there is a “standard” sequence for spacecraft operations, then this is the number of “exceptional” events being added to that sequence (e.g., special operations added to mapping). Metrics for this MoE are chosen to meaningfully reflect complexity (e.g., number of commands or activities required).

#### *MoE #3: Rush Factor.*

This MoE is a measurement of time between timeline and/or script completion and script execution on spacecraft. The Rush Factor MoE is inversely proportional to the time, i.e., the less time, the higher the Rush Factor. Elements involved in the mission operations process that may affect the Rush Factor include time required for testing of scripts on simulator/testbeds and time required for queuing and upload to spacecraft. The Rush Factor should be low (days, not hours). However, in order to be responsive to the science team or customer in a dynamic

mission, the Rush Factor may by necessity remain high, i.e., the higher the Rush Factor, the more responsive the operations team is although it is at a cost of putting strain on the team and processes.

*MoE #4: Effort Factor.*

This MoE is a measurement of the number of man-hours expended per mission event. It can be measure of complexity, but it is complicated by the efficiency of the process as well as by the level of automation. The Effort Factor is desired to be low to reduce costs and possible sources of errors. Automation can reduce the Effort Factor (-for operations personnel, but increase it for software engineers and programmers--).

*MoE #5: Response Factor.*

A trade study should be done to determine whether the decreased Effort Factor by operations personnel during the lifetime of a mission warrants the increased effort by the software developers to develop, implement, and test automation. Generally speaking, the larger the mission, the more worthwhile the software development effort will be. This MoE is an inverse function of the measurement of time between the customer's (e.g., science team) request for a mission event and its execution. The Response Factor should be weighted to account for complexity of the requested event. This factor should be maximized (i.e., the time between requests and execution minimized).

*MoE #6: Fatigue Factor.*

This MoE is a measurement of the tiredness of the operations team (e.g., hours worked). The short-term Fatigue Factor is based on shift length, while the long-term Fatigue Factor is measured over weeks or months. Other factors (e.g., complexity, rush, effort, and response) can affect the Fatigue Factor. It may be determined by subjective data (e.g., questionnaires) and the number of errors generated.

*MoE #7: Morale Factor.*

This MoE is a measurement of the satisfaction and optimism level of the operations team, but is difficult to quantify. It is mostly subjective, but some metrics can be collected to help in its determination. The possible metrics includes the turnover rate of personnel and the number operations personnel of complaints received by the

operations management. It might also be possible to use routine surveys of operations personnel, but it has to be determined subjectively as to how accurately these surveys reflect the true morale of the personnel.

## **B. Metric Collection Process**

In order to effectively generate, track, and use these MoE metrics, they should be incorporated into the mission operation process. Due to the limited record keeping typical in many of today's faster, cheaper, and better missions, it is often difficult if not impossible to reconstruct these metrics accurately, either to generate historical test cases or to determine retroactively how the MoE factors have changed over the life cycle of a current operations process. However, steps can be taken in the design of a new operations process or to implement changes in an existing system to collect these metrics.

At each step in the process two logs should be generated and kept. An automatic on-line log should record the time that each event starts and stops in a sub-process (e.g., recording the time that a timeline enters the script generation step and the time that generated script leaves this step to be sent on to the next step in the process, usually testing). This automatic log should also record errors detected by the computer system, especially of errors that were detected in the input data, as well as any significant decisions or substeps. A manual on-line electronic log should also be kept. This log is to record any errors found and corrected or changes made by the operator, along with the decision rationale. Both logs should be archived with the files for that particular pass or event and sent automatically to the operations director or analyst for review and analysis.

The following table shows possible measurement methods for each of the seven MoEs that have been identified in this paper.

## **C. Metrics Analysis Process**

The operations director or mission operations analyst should regularly collect and review metrics to identify problem areas. Trending software is of particular use to see how the factors change over time. The most useful plot is the cumulative errors plot, which shows on the same chart the cumulative total errors and each of the separate errors over the life of the mission or other designated time period. The cumulative number of errors is not so important, but the slope of the line is (i.e., the derivative of the cumulative errors with respect to time). By correlating the slopes of the line (steep slopes are bad, while flat or gentle slopes are good) to the seven MoE tracking charts, causes of the change in errors occurring on the spacecraft can often be identified by type. Steps can

then be taken to analyze the details particular MoEs to determine the root cause of the problem (or conversely, the lack of problems that indicates something good was happening).

#### **D. Feedback Implementation**

Once a sub-process has been identified as needing improvement, total quality methods should be used to involve operations personnel in the solution. They can help in both the identification of the root cause of the problem as well as to help determine how to rectify it and work out a way to implement the solution into the operations process. Methods and metrics to determine the success of the implementation should also be identified. In some cases it might be necessary to include mission or program managers, and or customers (e.g., principal investigators or chief scientists).

#### **E. Reporting Mechanism and Dissemination**

Any meetings involved in the operations process improvement process should be documented to leave a documentation trail of decisions made with rationale. This record is both important for historical purposes and to document decisions that might have to be reviewed at some later time, for instance, either to solve another similar problem, or (hopefully not) as evidence needed by a board of inquiry. Any reports or minutes of these meetings and decisions should be put into the operations archive and a copy sent to the mission manager or director, chief scientist, or other relevant entity. MOEs can be very helpful in help to determine when and where to add automation to mission operations.

#### **F. Discrepancy Tracking and Archive**

As is true for other aspects of mission operations, all discrepancy tracking, metrics collection and analysis, problem resolution and decisions should be archived. Any feedback implementations that have been decided upon should be put into the formal discrepancy tracking system and followed by the operations director until the implementation has been fully completed and tested.

### **III. Mission Case Studies**

#### **A. Low-power Compensation Experiment (LACE) Mission**

## *1. Mission Overview*

The primary purpose of the LACE satellite mission was to provide an orbiting instrumented target board capable of measuring the effects of active compensation of a ground-based laser beam propagated through the atmosphere. To fulfill a secondary purpose, the LACE satellite carried onboard the Ultraviolet Plume Instrument (UVPI), which was designed to collect images of rocket plumes in the ultraviolet waveband and to collect background image data on Earth, Earth's limb, and celestial objects. Background object imagery collected by the UVPI included day and night Earth limb air glow, aurora, sunlit and moonlit clouds, solid Earth scenes with varying solar illumination, cities, and stars. [Smathers et al 1993]

The LACE satellite was designed and built by the Naval Research Center (NRL) in Washington D.C. The satellite (Fig.XX) was launched on February 14, 1990 into a circular orbit at an altitude of 292 nmi and a 43° inclination. The spacecraft weighed 3175 lb. The body of the spacecraft was box-shaped, 4.5 ft by 4.5 ft and 8 ft high. Gravity gradient stabilization was provided by a 150-ft retractable boom with a 200-lb tip mass emerging from the top of the spacecraft. The LACE satellite had no orbit adjustment capability. Pointing for the UVPI was done by using a gimballed mirror.

The LACE satellite was designed to support its mission for 30 months. NRL operated three ground stations to communicate with and control the satellite. Two of the three ground stations were transportable (one was located in Maui, Hawaii for the entire mission and the other moved between Malabar, Florida and Vandenberg Air Force Base in California to support various observations). Each transportable ground station (TGS) was built by the NRL and designed around two 18-ft truck trailers. One trailer housed the telemetry, command, and communications equipment; the other provided an uninterruptible power supply and work area. The third ground station was permanently located in Maryland. [ibid] The central control for the LACE mission was located in a NRL facility in Alexandria, Virginia.

## *2. Mission Operations Analysis*

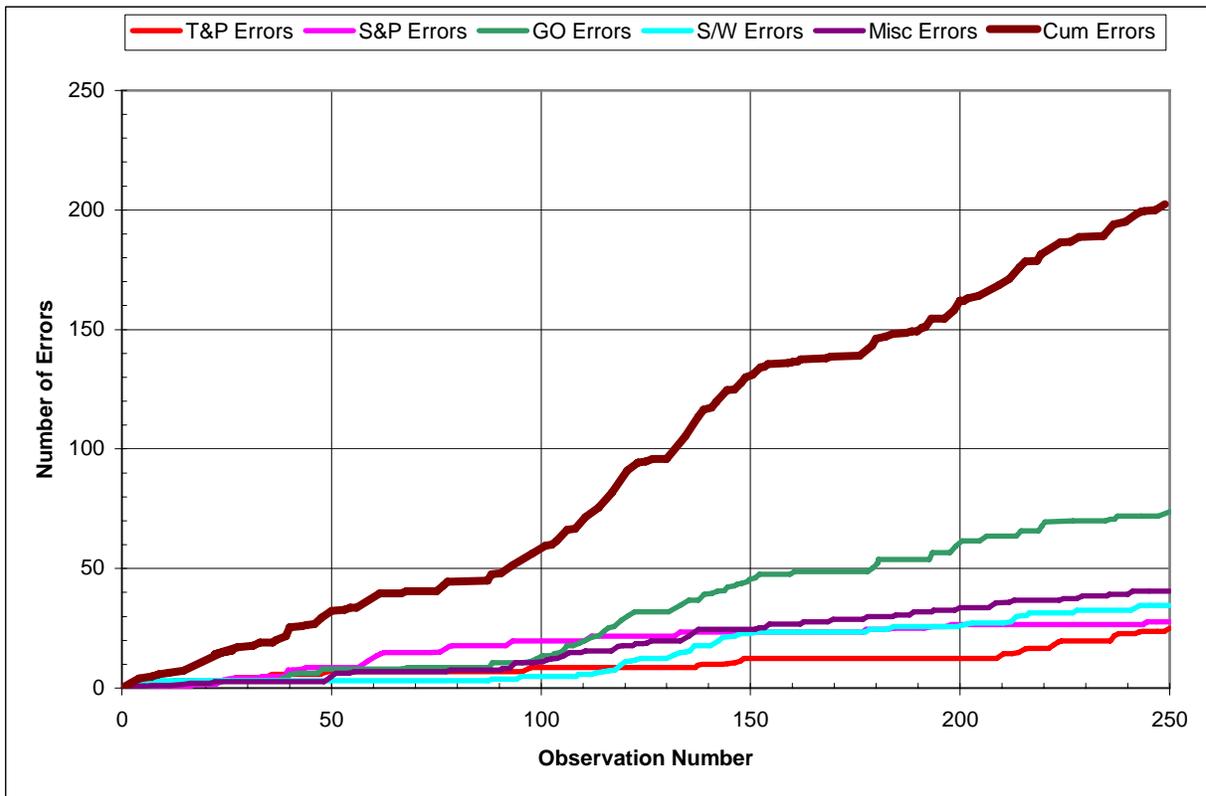


Figure 1. Errors Affecting LACE/UVPI Operations

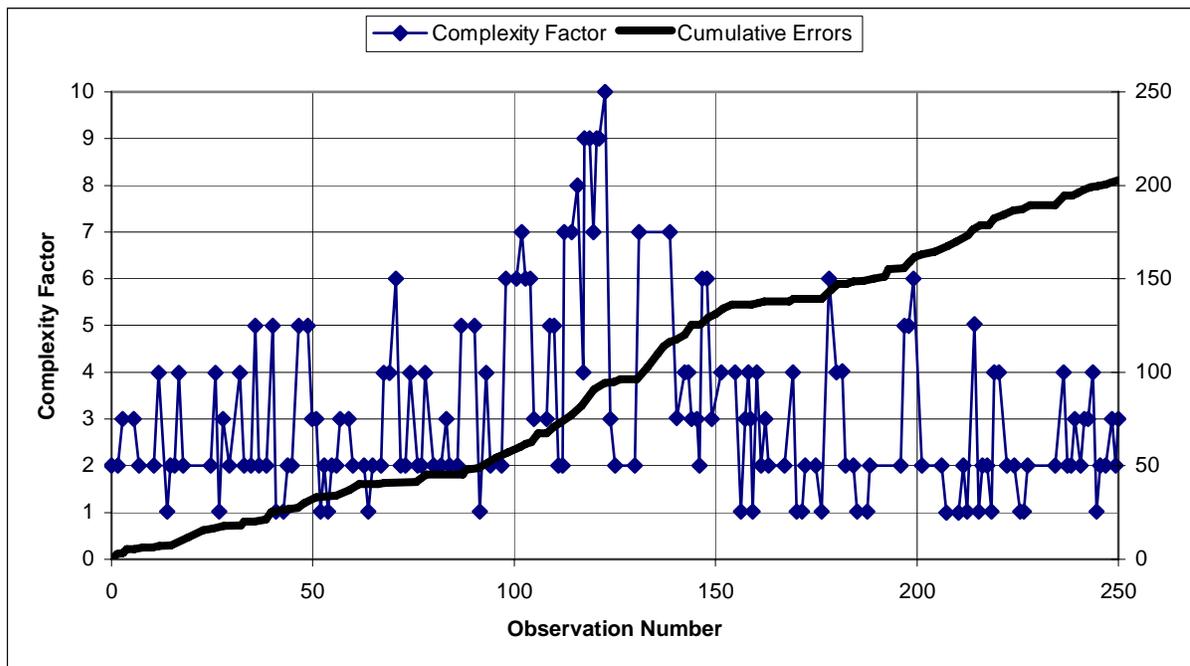
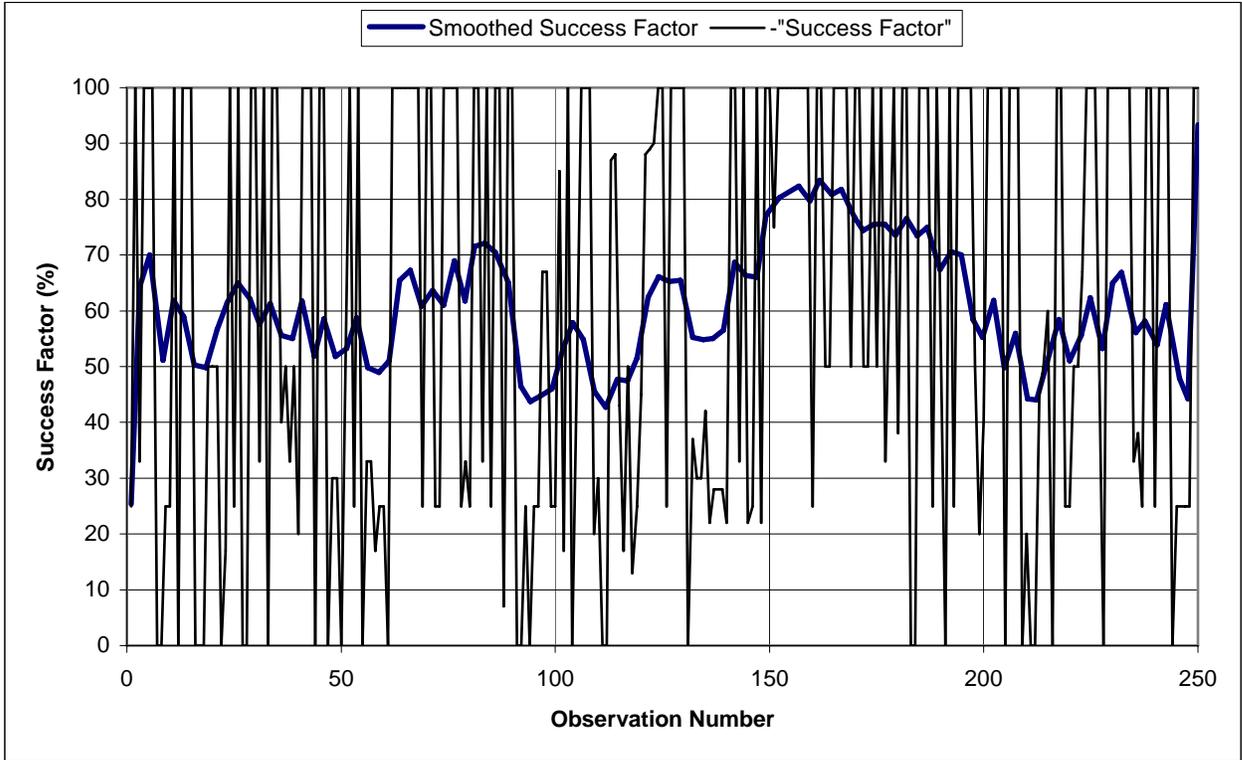


Figure 2. LACE/UVPI Complexity Factor

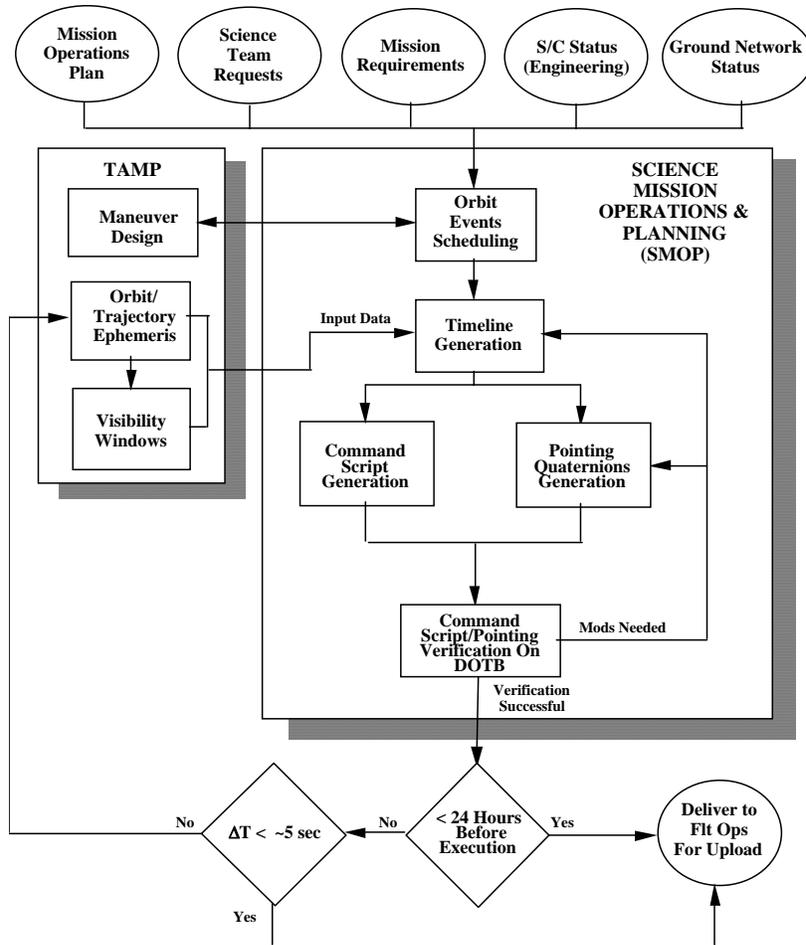


**Figure 3. LACE/UVPI Observations Success Metrics**

## **B. Clementine Mission**

### *1. Mission Overview*

Clementine was a technology demonstration mission of the Deep Space Program Science Experiment (DSPSE) jointly sponsored by the Department of Defense (DOD) and NASA that was launched on January 25, 1994. Its principal objective was to use the Moon, a near-Earth asteroid, and the spacecraft’s Interstage Adapter as targets to demonstrate lightweight sensor performance and several innovative spacecraft systems and technologies.[ref] The design, development, and operation of the Clementine spacecraft and ground system was performed by the Naval research Laboratory. For over two months Clementine mapped the Moon, producing the first multispectral global digital map of the Moon, the first global topographic map, and contributing several other important scientific discoveries, including the possibility of ice at the lunar South Pole. New experiments or schedule modifications



**Figure 4. Mission planning and script generation process.** Flowchart showing the process used by the Clementine FOT for lunar operations from scheduling of events; generating timelines, scripts, and pointing; testing of scripts; to the delivery of the script for upload.

were made with minimal constraints, maximizing science return, thus creating a new paradigm for mission operations. Clementine was the first mission known to conduct an in-flight autonomous operations experiment.

While in lunar orbit communication with Clementine was done using the NRL 100-ft diameter S-band antenna located in Pomonkey Maryland and the 34-ft diameter antennas of the NASA Deep Space Network stations in Goldstone California, Madrid Spain, and Canberra Australia.

## 2. Mission Operations Analysis

The control center for Clementine was the DSPSE Mission Operations Center (DMOC) located in the NRL facility in Alexandria, Virginia. The Science Team was housed in this facility which enhanced cooperation between the Science and Flight Operations teams. Although the major activities of the 350 lunar orbits were defined before

launch, detailed timelines were only developed for the first nine lunar orbits. The mission planning process for supporting operations was designed to maximize responsiveness to Science Team requests by developing detailed timelines no more than two days in advance of execution, and uploading the command sequence for each orbit only once the previous orbit's activities were complete and the spacecraft was downloading data. The ability to make last minute changes to the command scripts before testing and uploading was a fundamental feature of this process. The following process was designed to plan lunar mapping timelines and generate spacecraft command language (SCL) command scripts for upload (figure xxx):

- Mission Operations Plan - consult the detailed mission activity operations plans (AOPs) developed prior to launch.

- Mission (Operational) Requirements - these were the approved changes to the mission plan as requested by internal DMOC groups or external groups and agencies. These requests were defined, evaluated, and implemented using one of the following two methods:

(1) Nominal Operations Mode

This was for standard day-to-day operations in the DMOC. Mission operational requests (e.g., addition of or change to observations) that were internal to the DMOC including external groups (e.g., LLNL, Science Team) with personnel in the DMOC. The mission manager was responsible for the disposition of most operational requests. Requests with significant mission risk or programmatic issues were referred to the Director of Operations to be handled as special requests. Changes in sensor settings or tests, etc. were done by the Sensor Analysis Group without approval of the mission manager.

(2) Special Operations Mode

This was for all mission operational requirement requests from sources external to the DMOC (e.g., BMDO) or for special events or requests with significant mission risk or programmatic issues. All requests submitted to the Director of Operations for consideration. The aim was for no effect on daily DMOC operations unless the Director of Operations determined the request should be pursued. Examples of special mode requirements: lowering the lunar orbit for special observations; fundamental changes in operational philosophy.

- Spacecraft Health & Welfare Status, Resources Schedule, and Ground Network Status - the current operational factors external to the science mission that may affect the mission plan.

- Orbit Events Scheduling Activities - a daily meeting of the SMOP, Trajectory and Maneuver Planning (TAMP) and Flight Ops groups planned the general spacecraft timeline and schedule for a couple of days in advance using the mission plan with the above inputs. Analyses were performed by groups as required to optimize or define activities needed for a detailed timeline. Orbit maneuver requirements (if any) were sent to the TAMP trajectory group for design.

- Timeline Generation - a detailed timeline of spacecraft events was generated using a computer program. Input data for the timeline generator included applicable orbit / trajectory ephemeris or propagated data and visibility/shadow windows. Orbit data were provided by TAMP. Output data included:

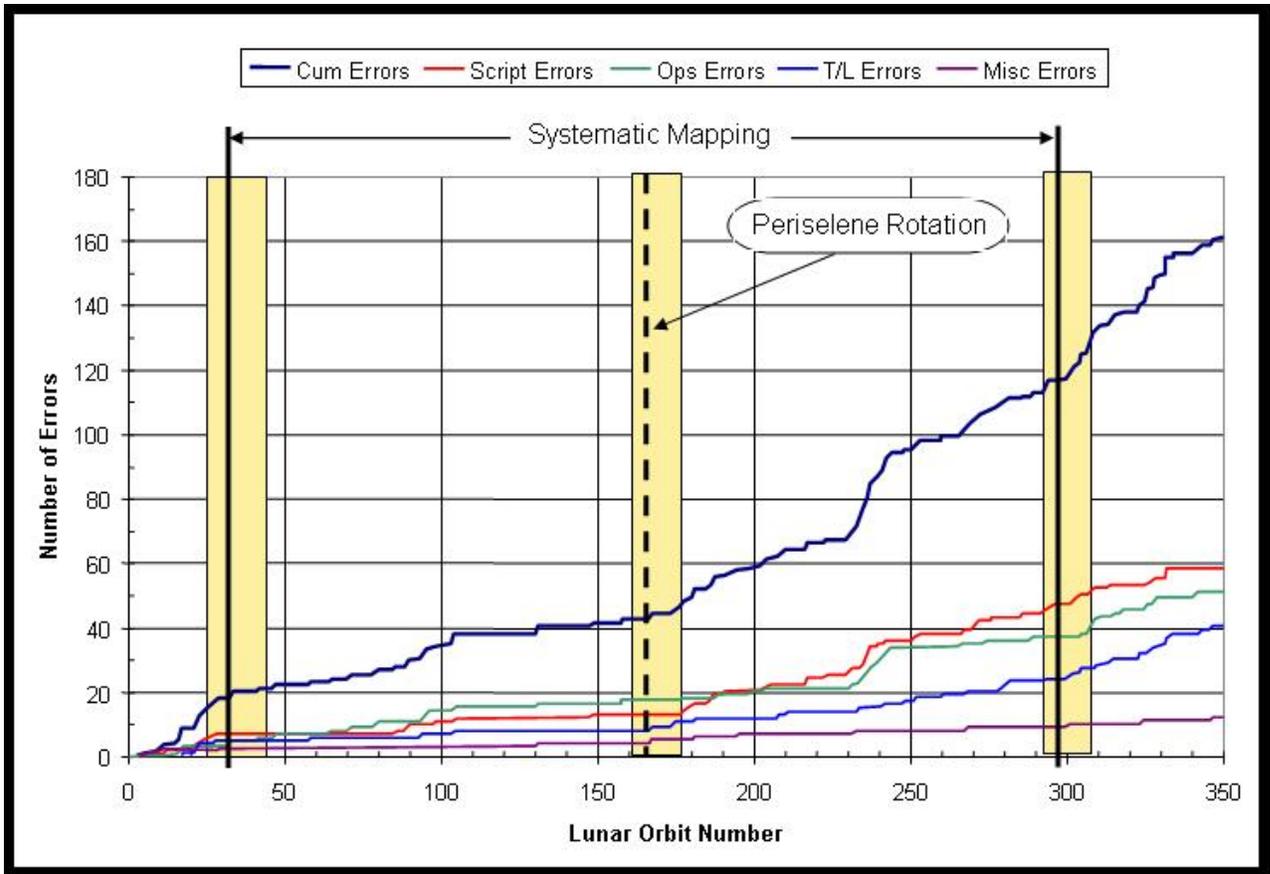
1. A hard copy of the spacecraft commands and associated events/activities.
2. A data file of the timeline to be used as an input to the lunar operations program (LUNOPS) that is used for operations support in the mission control room.
3. A timeline suitable for generating a spacecraft command script.
4. Spacecraft-to-target pointing requirements for generation of pointing functions.

- Command Script and Pointing Functions Generation - converted the timeline into SCL script suitable for compilation and upload to the DOTB or to the spacecraft. Output consisted of spacecraft and sensor commands, and spacecraft pointing quaternions (single or in table).

- Command Scripts Integration - the scheduler combined the command scripts from the SMOP, TAMP and Flight Ops groups into a single integrated command script. The integration process identified major conflicts between the contributing scripts. Conflicting scripts were revised by or in consultation with the contributing groups.

- Integrated Command Script/Pointing Verification on DOTB - a complete set of commands and pointing functions planned for upload was tested and verified end-to-end on the DOTB before being released for upload. Testing and verification for the lunar mapping phase were done in compressed time mode to allow sufficient time for all uploads to be tested in a 24-hour operation.

- Any problems encountered with the command script or pointing functions were corrected, and the modified upload set was tested again on the DOTB. If the execution of the verified uploaded set was due to start more than ~24 hours prior to its execution, the timeline and upload set were regenerated using the latest orbit/trajectory ephemeris approximately 24 hours before the execution. This upload set was retested and verified, after which it was sent to the DMOC for upload to the spacecraft, which occurred at least an hour before execution.



**Figure 5. Spacecraft operations errors during lunar orbit.** Plots of the cumulative errors for each of the major error sources as well as the total number of errors showing the relationship to error occurrence to major events that happened during lunar orbit.

The Clementine mission demonstrated how effective a small, but highly qualified group of mission planners and operations personnel can be in achieving complex science objectives. The Clementine spacecraft obtained nearly two million digital images of the Moon at visible and IR wavelengths covering >99% of the lunar surface. These data are enabling the global mapping of the rock types of the lunar crust and the first detailed investigation of the geology of the lunar polar regions and global topographic figure of the Moon. The BSR experiment that was added to the schedule after lunar insertion, yielded data consistent with the presence of ice in deep craters at the lunar South Pole.

Although the Clementine lunar mission with its six imaging sensors, laser altimeter and complicated slewing scheme was as complex as comparable NASA mapping missions of that period, the Clementine team that provided science mission planning and operations did so with far fewer personnel and resources than in the NASA teams to perform the same functions (Sorensen, et al 1995). The work load on the Clementine operations team during the

lunar mission was excessively high due mostly to the incomplete software tools, such as the Command Script Generator, which were designed as an integral feature of the mission planning and operations process. However, the responsiveness of the process to requests for changes and additions to the scheduled events enhanced the science return by providing a fast feedback loop to optimize sensor settings and procedures, and the mechanism for obtaining additional data.

### **C. Miniature Sensor Technologies Integration (MSTI-3) Mission**

#### *1. Mission Overview*

The MSTI program performed the first on-orbit functional demonstrations of low-cost integrated sensor technologies that support theater missile launch detection and tracking. The MSTI program transitioned from the Ballistic Missile Defense Organization (BMDO) to the Air Force in 1994. The program demonstrated that small spacecraft can be placed on orbit for less than \$50M within 12-24 months from a decision to proceed.

MSTI-3 was the third, and most advanced, satellite developed by the MSTI program to demonstrate more sophisticated sensors and data gathering capabilities. MSTI-3 was launched from Edwards AFB aboard a Pegasus on 16 May, 1996 for a nominal one-year mission and placed eventually into a 425 kilometer circular orbit, inclined at 97°. The MSTI-e satellite was deorbited on December 11, 1997.

MSTI-3 collected data in the short wave infrared (SWIR) and medium wave infrared (MWIR) bands. It surveyed the Earth collecting data to support analysis of ground features, such as terrain and bodies of water, and atmospheric features, such as clouds and aurora. The purpose of this analysis was to determine how the appearance of these features in the infrared varies with season, time of day, and aspect angle. This basic research supports the design of infrared Earth observation satellites. The primary mission of MSTI-3 was to gather extensive MWIR background clutter statistics at sufficient resolution to resolve whether tracking theater ballistic missiles in the coast phase of their trajectories against a warm earth background is achievable. In addition to providing the truth data, the visible imaging spectrometer gathered environmental data to support environmental and ecological analysis. [<http://www.globalsecurity.org/space/systems/msti.htm>]

The MSTI-3 satellite had three instruments: an SWIR camera, an MWIR camera, and a visible wedge spectrometer camera. All of the instruments shared a single telescope. Each infrared camera had seven filters. The

visible wedge spectrometer camera provided comparison data to verify observations made by the infrared cameras. The wet weight of the satellite was 466 pounds.

The Air Force Satellite Control Network (AFSCN) provided the primary tracking, telemetry, and control for the MSTI-3 satellite, while payload data were also received at the NRL TGS that had been located at North Pole Alaska. Flight operations were performed by the USAF, while mission planning and scheduling, and payload operations were conducted in the NRL facility in Alexandria, Virginia (the same one used for the earlier LACE and Clementine missions).

## *2. Mission Operations Analysis*

MSTI-3 operations were successful. During nominal operations, the operations team consistently took four to five 20-minute payload operations and brought back an average of over 8500 images each day. An impressive statistic in its own right but doubly so since original spacecraft design specifications called for only two operations a day. Unfortunately, as all good news bad news stories go, the MSTI operations team had a number of challenges to overcome before reaching its current efficient and responsive state.

Most of the problems encountered in the mission can be attributed to the “faster, better, cheaper” nature of the program. More specifically, programmatic constraints associated with a tight launch manifest schedule immediately after the Pegasus return to flight certification, forced an abbreviated system level test period. Subsequently, the operations team completed much of the final ground test sequences; including payload timing and radiometric characterization, on orbit.

Faced with this challenge the operations team expected changes to the systems operations concept as part of the on-orbit characterization period and set up a system flexible enough to respond. The cornerstone of this system was a detailed error recording and tracking program to measure the effectiveness of the operations process. These metrics served as valuable management tool specifically in the allocation of resources and the identification of areas for process automation.

For the MSTI program an error is defined as any event that causes the loss or partial loss of any scheduled operational event. Each error is cataloged according to the spacecraft or payload event it affected; the corresponding date, time and orbit; that events stored program command uploads; and the stored state of health downlink file containing the telemetry from the spacecraft for that event. A weekly (daily during launch and early orbit phase

(LEOP) and anomaly operations) meeting was held to review all spacecraft activity for that period and every error was assigned an action officer who was responsible for troubleshooting cause and to suggest a resolution for each of the errors.

These errors were also grouped in the following five categories for trending and correlation analysis:

- 1) Spacecraft Errors - actual spacecraft anomalies preventing successful completion of an event;
- 2) Mission Planning Errors - errors generated in the planning process often timing or geometry mistakes;
- 3) Command Script Errors - errors generated in stored program command sequences;
- 4) Real-time Errors - errors committed during real-time contact with the spacecraft; and
- 5) Miscellaneous Errors - catch all category often used for ground data recovery problems resulting in actual loss of data (more on data recovery problems later)

Spacecraft errors were included early on in the tracking process because of characterization requirements forced on the operations team prior to launch. This worked out extremely well because recurring spacecraft errors became

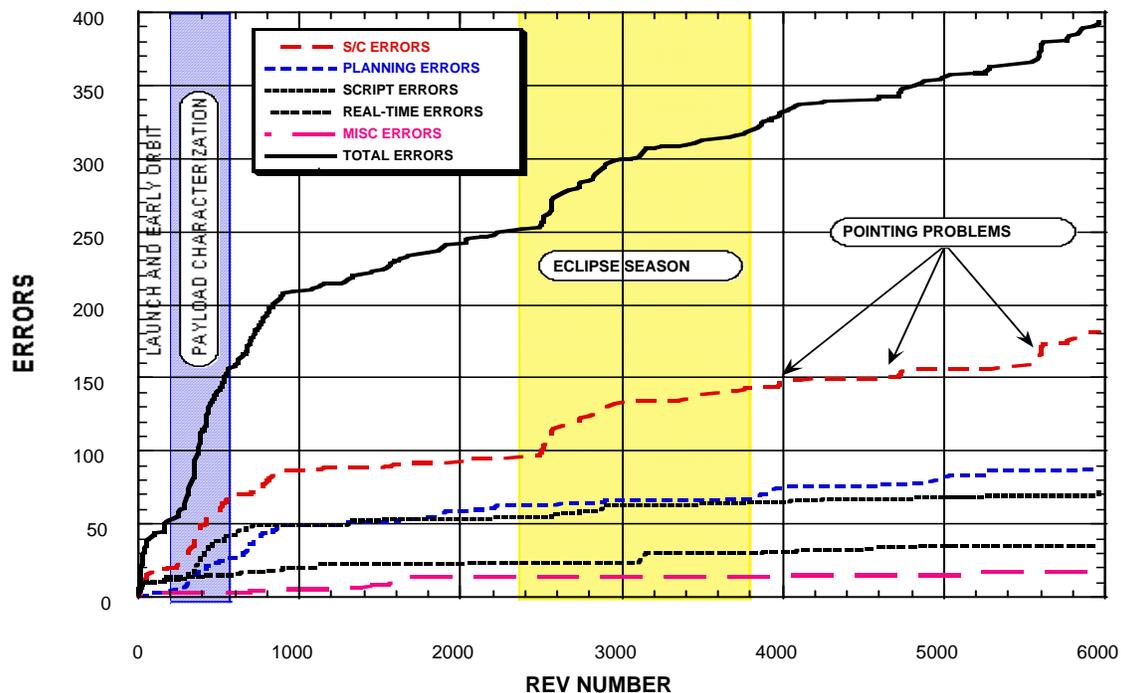


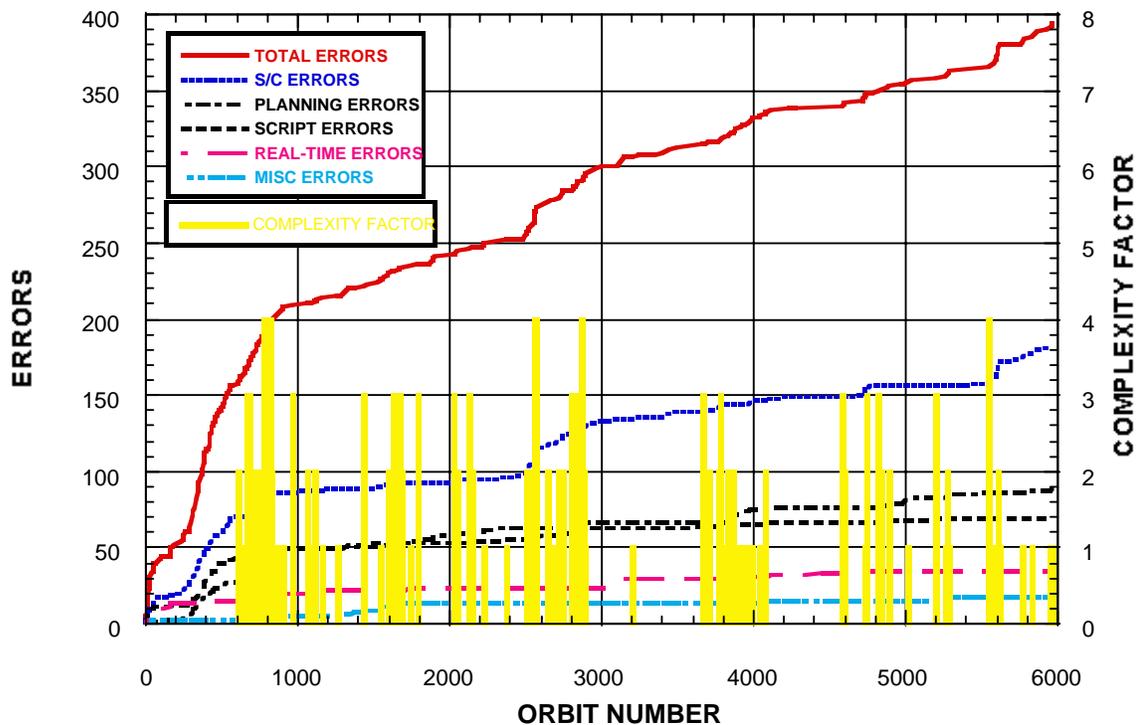
Figure X.1 MSTI-3 Mission Operations Errors

very similar to mission planning or command generation problems where workarounds required redefinition of procedures and creative operations practices.

Figure X.1 is a cumulative history of MSTI operational errors from the start of the mission. Launch and early orbit is marked with a steep learning curve that is typical in most one-of-a-kind research and development missions. This steep error rate extends through the payload characterization phase as the operations team struggled through a payload characterization phase without the aid of numerous (much needed) pre-launch ground tests. Throughout this phase command script and mission planning errors are seen to lag spacecraft errors as knowledge gained through analysis of spacecraft behavior is slowly incorporated into planning and script generation processes.

MSTI diverse mission science requirements actually dictated a significant amount of automation in the mission planning and script generation processes to allow for multiple experiment objectives to be satisfied with one data collection event. After launch, this automation matured; not only through the incorporation of newly learned spacecraft and payload operational constraints; but also through the addition of more flexible constraint checking algorithms that allowed workarounds to future errors to be more readily incorporated in the automation processes themselves. The results of this are easy to see during the eclipse period where spacecraft errors jump dramatically but through close error tracking and incorporation new planning and script processes are incorporated with relatively little problems.

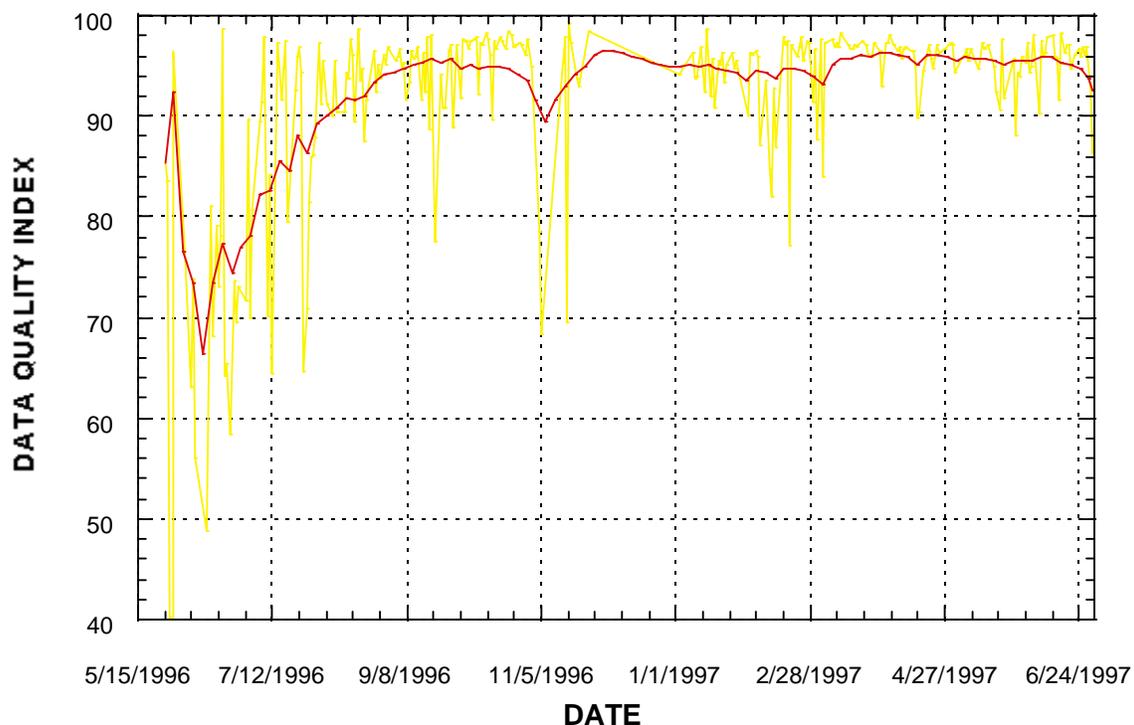
Figure X.2 overlays a complexity factor to the cumulative error history. For MSTI-3 the complexity factor is a somewhat subjective figure of merit numbering from zero to four outlining the difficulty of the overall operations



**Figure X.2 MSTI-3 Errors vs. Mission Complexity Factor**

effort at a given time. The complexity factor is higher for special data collection events or other operations activity that required more than nominal effort by the operations team.

Analysis of this metric adds additional insight the effectiveness of the MSTI error tracking. Notice that the command script errors tend to damp out once automation could efficiently turn around error tracking resolutions. Mission planning however did not always benefit from this type of incorporation of lessons learned as the planning errors clearly show. When new/different types of observations were attempted mission planning errors are often shown to increase. This particularly true for the period when payload pointing control experienced hardware problems and the mission planning team tried a series of complex work-arounds to regain mission capability.



**Figure X.3 MSTI-3 Data Quality During the Mission**

Figure X.3 illustrates an additional metric that proved very useful for MSTI-3 operations. As with most science missions MSTI-3 was seen as (and indeed was) downlink bandwidth limited by the science team. MSTI-3 has an on-board storage capacity of one gigabyte with downlink throughput of only 1 megabit per second with station contacts lasting only 10 minutes or less. With the push from the science team to maximize data collection, the on-board data management concept was set up to be very aggressive with no more than six hours after downlink before data was overwritten by a new data collection.

The ground team set up a quality control program to monitor the status of each downlink, therefore allowing responsive management of the whole data recovery process. This data quality index also allowed the team to generate statistics on every ground station used for data recovery, which was used to work with Air Force Satellite Control Network engineers to troubleshoot ground network problems. Like the spacecraft side of the problem, the ground network needed a good amount of time early in the mission to optimize data downlink cleanliness and

periodically had to take stations off line during the course of the mission to fix quality control issues identified by this index.

#### **IV. An Application of Mission Operations Process Improvement**

DataLynx is Honeywell's commercial offering of spacecraft tracking and mission operations. Beginning operation in November 2000, Honeywell Technology Solutions Inc. (HTSI), designed and built the DataLynx Operations Center (DOC) located at the HTSI headquarters in Columbia Maryland Figure x-1. From the DOC operators can control both spacecraft and the ground tracking antennas.



Figure x-1. The DataLynx Operations Center in Columbia Maryland

In contrast to the previous examples of spacecraft operations the DataLynx example will examine the operations of the ground tracking stations. DataLynx operates an 11 meter and a 7.3 meter X and S-Band antenna located at the Poker Flat range near Fairbanks Alaska. Except for the aperture diameters, the two antenna systems have similar characteristics and in fact share some common hardware. The overall concept of operations for the DataLynx ground tracking network is for the routine operations of the antennas to be primarily autonomous with scheduling, monitoring, and troubleshooting performed by operators, 24/7, located in the DOC facility in Maryland. One full time and one part time employee in the Fairbanks area provide onsite maintenance and repair on an 8/5 schedule.

[Note: in January 2005 DataLynx took control of three additional antennas at Poker Flat Alaska formally operated by the NASA Wallops Flight Facility. These antennas are not the subject of this example].

From the beginning, the DataLynx control center and tracking network was designed to measure and track system performance to provide for fault detection and investigation and as a means to measure performance and provide feedback to improve operations. The operations of the DataLynx tracking antennas is autonomous beginning with the receipt of customer schedule requests and associated orbital elements and sending these to the ground station, to the initialization of the ground station electronics, taking the pass, and the delivery and storage of the spacecraft data to the user. At each step in automated process data files that are received or generated are stored with their associated time tags that allow for reconstruction of each step that lead to successful or erroneous results. Analysis of these files and the associated time tags can be used to identify bottlenecks, sequencing errors, and where efficiencies can be made in the process. Figure X – 3 shows a data flow diagram tailored for the ground network operations.

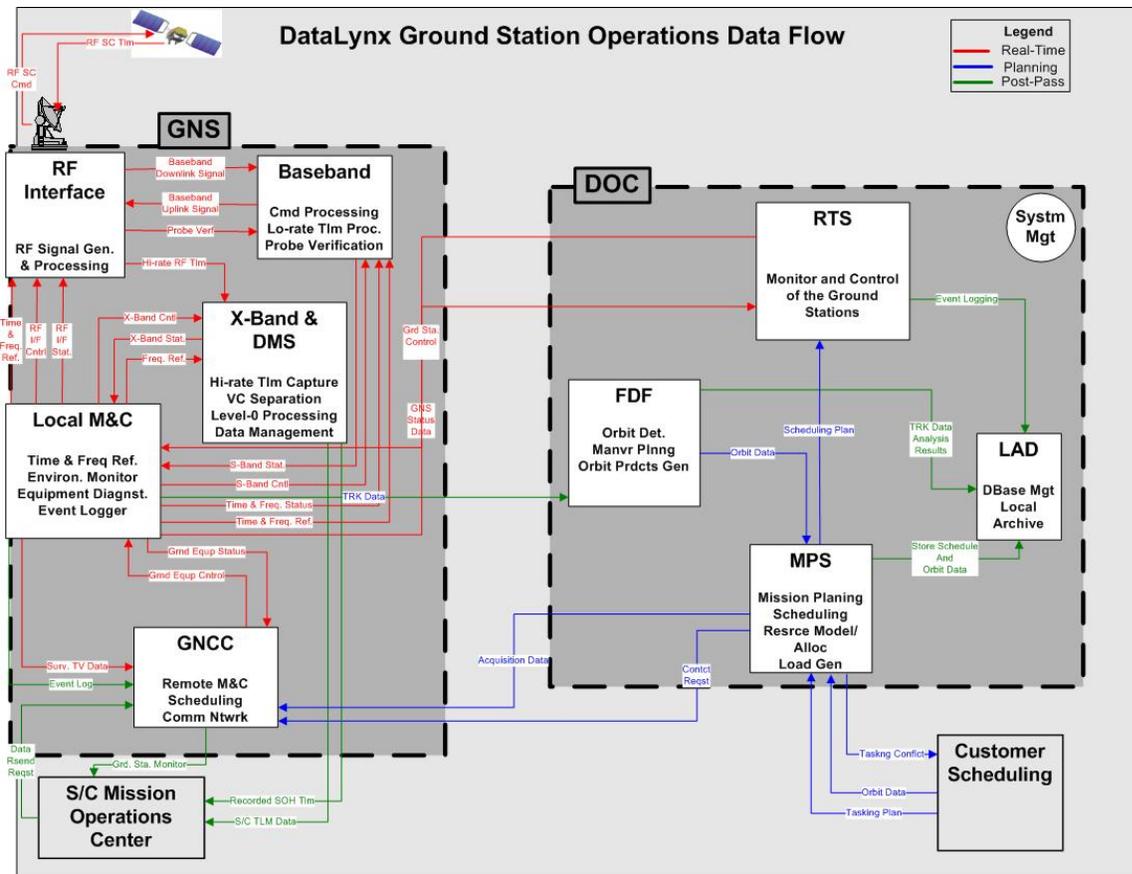


Figure X-3. Data Flow and Control of the DataLynx Ground Station Network

Each of the major functions in the DataLynx Operations Center (DOC) accepts, processes, and generates data for the routine operation. At each function the data is recorded and is used for both trouble shooting and for operations improvement. The Mission Planning System (MPS) accepts both schedule requests and orbit data from the outside community and generates conflict free schedules that are sent to the ground stations for implementation. The Flight Dynamics Facility (FDF) takes in the tracking data from the ground station, performs orbit determination and analyzes the tracking data to monitor antenna performance. The Real Time System monitors and controls the operation of the remote ground stations in Alaska. Both the time tagged commands and the monitor data from the ground station are recorded and are available for analysis to diagnose errors or to evaluate operations performance and make improvements to the operations. The Local Archive Database is the primary long term storage for all the data collected and generated by the DataLynx system. For example, a long term analysis of tracking data angle

means revealed that two distinct populations of values were being generated. See figure X-3. On close examination it was revealed that there was a difference between the X-band tracking and the S-Band tracking biases caused by an offset of the different feeds. This prompted a change to the tracking data software processes that applies separate X- and S-Band angle data corrections differently.

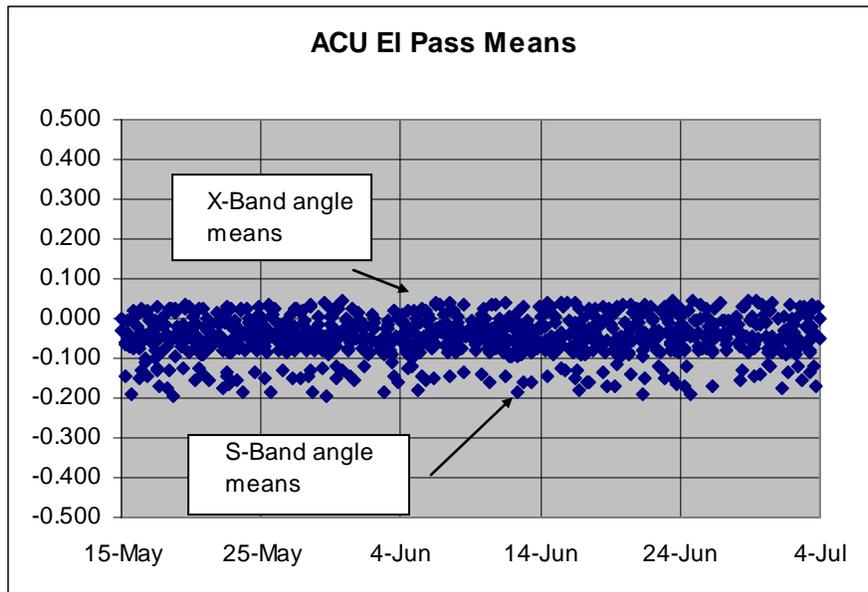


Figure X-3. Antenna Elevation Angle Pass Means for both X- and S-Band Tracking

Just as important to the automated file and data recording are the operator reactions and comments. If an operator discovers or suspects a problem or discrepancy in the operations, the operator is authorized to initiate a discrepancy report. These reports are primarily used to identify, describe, and track suspected problems but these reports, held in a database, are also a useful source of process improvement. Figure X-2 shows the number of discrepancy reports initiated from March 2004 to October 2005.

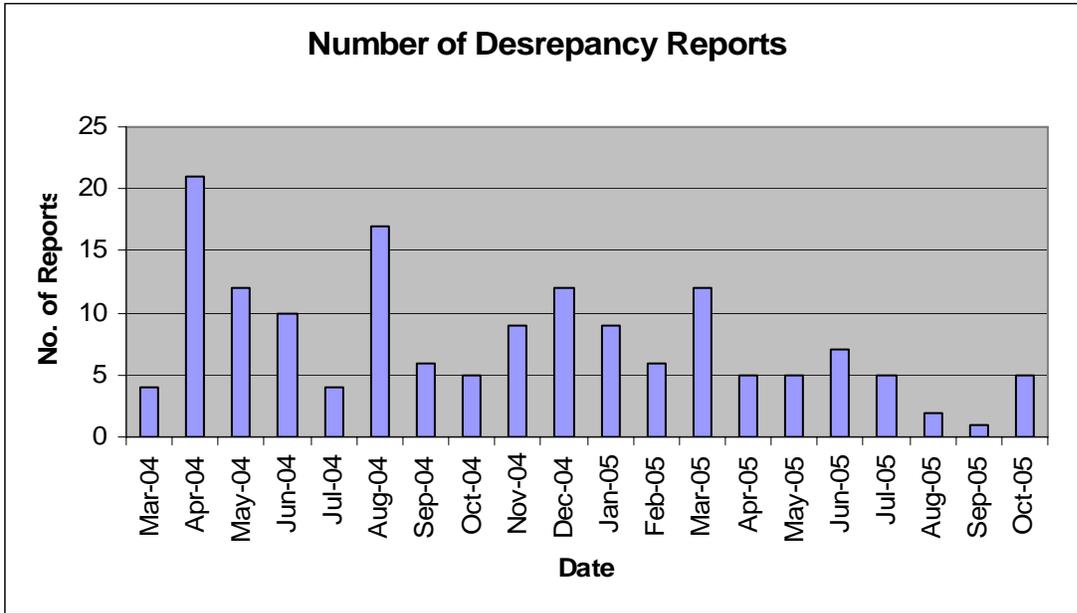


Figure X-2. The Number of Reported Discrepancies over Time

An analysis of these reports can yield useful process improvement information. Although the raw number of reports, the trend over time, can indicate the need for additional investigation, a detailed breakdown of the discrepancies by type and even by time of day is needed to point to specific opportunities for improvement. For example the following chart shows percentage of discrepancies as a function of subsystem.

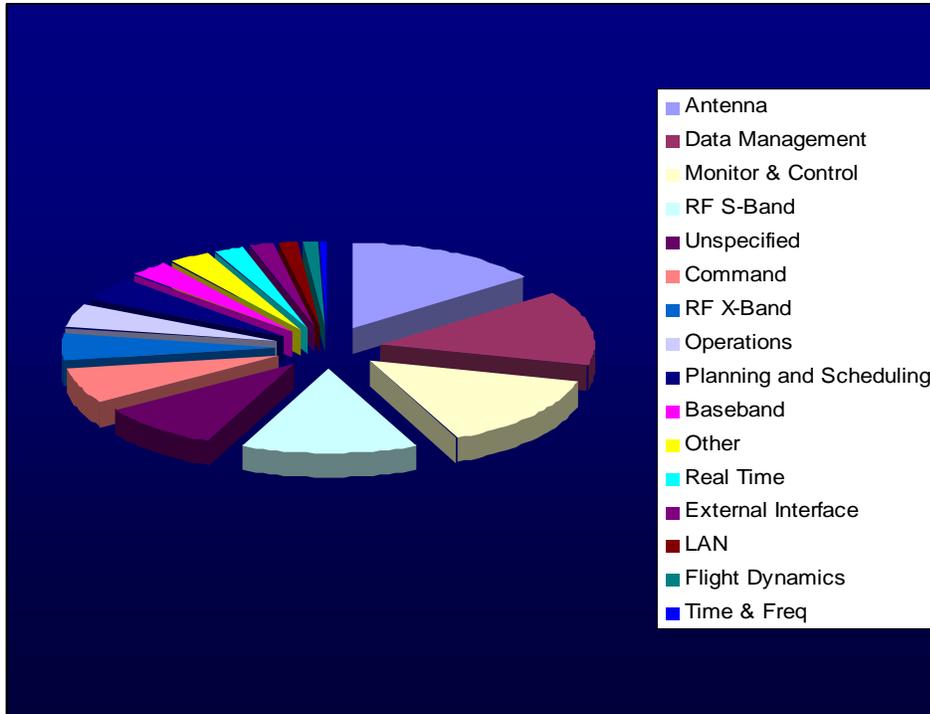


Figure X-3. Discrepancy Reports by Subsystem

It can be seen from this representation of the data that more than half of the reported problems with the system are in four areas: antenna, data management, monitor & control, and the S-Band RF subsystem. From this plus an analysis of the affect that these discrepancies have on the system, management is able to apply the engineering effort to where they will have the greatest impact on performance improvement.

Finally, an additional source of information used for troubleshooting and operations performance analysis is the operations log. The operations log is an electronic log (MS Word document) where the operations personnel keep a narrative of events that occur during each shift. These logs not only tell what happened but also tell why something happened. Summarizing conversations with space mission operations or with onsite maintenance personnel has given insight into the reasoning behind a course of action that was taken and can help lead problem investigations and improve operation efficiency.

## V. Conclusion

TBS