

# SATELLITE MISSION OPERATIONS BEST PRACTICES

ASSEMBLED BY THE

BEST PRACTICES WORKING GROUP  
SPACE OPERATIONS AND SUPPORT TECHNICAL COMMITTEE  
AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS



**Don't Reinvent The Wheel**

**Point of Contact:**

Ray Harvey  
RayHarvey@jhuapl.edu  
Johns Hopkins Applied Physics Laboratory  
11100 Johns Hopkins Road  
Laurel, MD 20723

April 20, 2001

**ASSEMBLED BY THE  
BEST PRACTICES WORKING GROUP  
AIAA SPACE OPERATIONS AND SUPPORT TECHNICAL COMMITTEE**

(April 20, 2001)

CONFIGURATION MANAGEMENT	Steve Paine/SMC-TEOC	April 13, 2001
TRAINING AND CERTIFICATION	David Fuller	April 10, 2001
PROCESS IMPROVEMENT	Trevor Sorensen/ Univ. of Kansas	March 30, 2001
GROUND SYSTEM DEVELOPMENT	Evan Eller/Honeywell	April 12, 2001
PRE-LAUNCH SPACECRAFT OPERATION DEVELOPMENT AND TEST	Scott Williams/Stanford	December 27, 2000
FLIGHT DYNAMICS	Ken Galal/NASA-Ames	March 30, 2001
OFF-LINE SPACECRAFT PERFORMANCE ASSESSMENT	Ray Harvey/JHU-APL	April 13, 2001
SINGLE SCREEN SATELLITE ALARM LIMIT DISPLAY REQUIREMENTS	Brian West/SMC-TEOC	July 25, 2000

DISCLAIMER

The AIAA Space Operations and Support Technical Committee (SOSTC) cannot accept responsibility for any successes or failures that may have resulted from the use of these Best Practices. They are simply recommendations, suggestions, and rules of thumb based on Lessons Learned. In using these Best Practices, please consider how each recommended Best Practice would be implemented in your particular application and determine for yourself whether or not they seem appropriate. (12/26/00)

**SATELLITE MISSION OPERATIONS BEST PRACTICES  
BEST PRACTICES WORKING GROUP  
AIAA SPACE OPERATIONS AND SUPPORT TECHNICAL COMMITTEE**

**TABLE OF CONTENTS**  
(February 1, 2001)

	<u>Page</u>
Table of Contents	i
Foreword	iii
Best Practices Working Group	v
Acknowledgements	vi
1. Operations Management (TBS)	
- Schedules	
- Documentation	
- Staffing	
- Reviews	
2. Configuration Management for Satellite Operations (Paine Input)	1
3. Training and Certification (Fuller Input)	5
4. Process Improvement (Sorensen Input)	10
5. Data Transport (Welch TBS)	
6. Ground System Development (Eller Input)	17
7. Spacecraft Development and Test Support (Williams TBS)	
- Operability Enhancement	
- Spacecraft Integration and Testing	
8. Pre-Launch Spacecraft Operation Development and Test (Williams Input)	23
- Ground System Testing	
- Simulations	
9. Launch and Early Operations (TBS)	
- Launch Support	
- Spacecraft Activation and Checkout (Williams, TBS)	
10. Flight Dynamics (Galal Input)	28
11. Mission Planning and Scheduling (TBS)	
- Event Planning	
- Station Scheduling	
12. Station Contact Operations (TBS)	
- Pre-Pass Spacecraft Operations	
- Station Acquisition	
- Pass Termination	
13. Real-Time Control Operations (Harvey/Fuller, TBS)	
- Telemetry Monitoring	
- Real-Time Commanding	
14. Recorded Payload Data Recovery (TBS)	
15. Stored Command Sequence Generation and Execution (TBS)	

## TABLE OF CONTENTS (CON'T)

	<u>Page</u>
16. Off-line Spacecraft Performance Assessment (Harvey Input)	35
17. Single Screen Satellite Alarm Limit Display Requirements (West Input)	41
18. Payload Data Processing (Marinacchio TBS)	
19. Contingency Planning and Support (TBS)	
20. Flight Software Maintenance (TBS)	
21. Ground System Maintenance (TBS)	

TBS—To Be Supplied

**SATELLITE MISSION OPERATIONS BEST PRACTICES  
BEST PRACTICES WORKING GROUP  
AIAA SPACE OPERATIONS AND SUPPORT TECHNICAL COMMITTEE**

**FOREWORD**

(4/12/01)

The effort of compiling a collection of Best Practices for use in Space Mission Operations was initiated within a subcommittee of the American Institute of Aeronautics and Astronautics (AIAA) Space Operations and Support Technical Committee (SOSTC). The idea was to eventually post a collection of Best Practices on a website so as to make them available to the general Space Operations community. The effort of searching for available Best Practices began in the fall of 1999. As the search progressed, it became apparent that there were not many Best Practices developed that were available to the general community. Therefore, the subcommittee decided to use the SOSTC Annual Workshop on Reducing Space Mission Costs as a forum for developing Best Practices for our purpose of sharing them with a larger audience. A dedicated track at the April 2000 workshop was designed to stimulate discussions on developing such Best Practices and forming working groups made up of experienced people from various organizations to perform the development. These groups were solicited to help outside the workshop to bring this effort to fruition. Since that time, bi-weekly teleconferences have been held to discuss the development of the Best Practices and their posting.

One set of Best Practices that did exist was the result of a NASA Goddard Space Flight Center activity. The Satellite Operations Risk Assessment (SORA) Team produced some Best Practices based on research into a problem with SOHO operations. This set was available to us and we used it as a model. In addition to the SORA report, we started with a list of topics and functions involved in Mission Operations. Members of the Best Practices Working Group volunteered to lead the development of Best Practices for particular topics. We scheduled the telecons such that particular topics were to be discussed on particular days. The leader for that topic would send out the draft of Best Practices to the group via email. This was the basis for discussion during the telecon. Following the telecon, the leader would incorporate the various comments received. The telecons were very informal. Announcements with a proposed agenda were sent out prior to the day of the scheduled telecon (sometimes the day before) and minutes were kept and emailed to the group for those who could not attend (unfortunately not always in a timely manner). Action items were assigned as appropriate. The end results of these discussions are the sections presented within this document.

There are many reasons why this effort has been possible. One in particular was used as a selling point to the development group. First of all, we could! These are simply recommendations and rules of thumb; not declarations of what you "shall do". These are NOT Standards and would not go through the years of review often required of Standards. This is a way that real experienced people can do something to help their fellow Mission Operations team members and possibly help shape future Mission Operations. It is stressed in the "disclaimer" that these Best Practices are simply recommendations based on Lessons Learned. Many times when we think of our Best Practices, we are looking at things we did right in the past and would do again the next time.

These are Lessons Learned-applied! This is our way of sharing with the community those things we did right so they may be able to take advantage of past experiences.

This effort could be construed as another attempt to foster the “Faster, Better, Cheaper” paradigm in that it may facilitate re-use of proven “processes”; but it was really put forth for another purpose. The underlying objective was to provide someone who has not done this before with some insight into what has worked in the past, and give them guidance as to how they may want to implement their Space Mission Operations related application. It is this underlying principle that forms the basis of the SOSTC Best Practices Working Group (BPWG) logo. In case you have seen it (perhaps it is on the cover page) and don't quite understand: Our "Rookie" Mission Operations Manager is trying to reinvent the wheel. We don't want to see that happen. The BPWG is trying to reduce this type of occurrence by making our Best Practices available to anyone; especially to the "Rookie" Mission Operations Managers!

In closing, there is one main reason why this effort has been as successful as it has been and it must be acknowledged here. It is the time and effort of the people on the BPWG. I was somewhat surprised at the dedication and hard work these folks put in to a “zero budget” effort. It has really made me appreciate what experienced professional people can do if they have a focussed goal. My thanks go out to the members of the team who have “suffered” through the “every-other” Friday telecons. As of the end of April 13, 2001, the only bi-weekly telecon starting with the one that was held on April 28, 2000, that did not occur, was the one that would have occurred the day after Thanksgiving 2000. No one argued when it was decided to not schedule that one. As of the Spring of 2001, this effort is ongoing. We are always looking for new members to take on some of the topics we have not touched on. If you are interested in helping out or wish to comment on what we already have, please contact me at:

[Ray.Harvey@jhuapl.edu](mailto:Ray.Harvey@jhuapl.edu)

Whether you are considered a Ground System Administrator, Spacecraft Operator, Principle Investigator, Program Manager, Chief Scientist or, in particular, a Rookie Mission Operations Manager, we hope you find the information contained within beneficial. Please remember that these are recommendations, suggestions, and rules of thumb. They are not guaranteed to bring you success, but they may help you avoid some trouble.

Ray Harvey  
Johns Hopkins Applied Physics Laboratory  
11100 Johns Hopkins Road  
Laurel, MD 20723

**Best Practices Working Group  
Space Operations and Support Technical Committee  
American Institute of Aeronautics and Astronautics**

Ray Harvey, Chairman

Roger Brissenden  
Evan Eller  
David Fuller  
Ken Galal  
Leigh Gatto  
Tom Hail  
Michelle Lutz  
Christine Marinaccio  
Annan Navo  
Bill Nickum  
Bernie Schwartze  
Jeff Shirer  
Trevor Sorensen  
William W. Vaughan  
Dave Welch  
Brian West  
Scott Williams

**Space Operations and Support Technical Committee  
American Institute of Aeronautics and Astronautics**

James Grady, Chairman

## **Acknowledgment**

(04/12/01)

The SOSTC BPWG would like to thank the former group members and reviewers who have graciously and valuably contributed to this effort. Those members are recognized below:

### **Former Members**

TBD

### **Reviewers Providing Feedback**

Matt Bille  
Connie Golden  
Larry McNeese  
Dave Recce  
Michael Vinyard

We would also like to thank the members of the SOSTC who reviewed the document and found it to be so good, they didn't need to provide comments. Seriously - thank you for taking the time.

**SATELLITE MISSION OPERATIONS BEST PRACTICES  
BEST PRACTICES WORKING GROUP  
AIAA SPACE OPERATIONS AND SUPPORT TECHNICAL COMMITTEE**

\*\*\*\*\*

**CONFIGURATION MANAGEMENT FOR SATELLITE OPERATIONS**

**STEPHEN C. PAINE  
AIR FORCE SPACE AND MISSILE SYSTEMS CENTER**

April 13, 2001

1.0 Introduction

Configuration Management (CM) is handled in different ways at different levels. From the operations perspective, the goal of CM is to produce reliable results when conducting satellite operations. The real reason for CM is that there is a single point of control that is held responsible for satellite operations.

*Configuration Management is the act of controlling all mission-impacting aspects of the satellite operator's environment. CM introduces organizational control into satellite operations. A properly controlled environment will produce predictable results, and allows the Program Manager to assume total ownership and responsibility for program success or failure. In some cases, this ownership may be held by the Operations Manger (OM). A real-life example is listed below:*

*Operational procedures established the process by which a change to the real time environment was allowed. CM managed the change request, tracking, disposition of the approval authority and audit processes. Once CM approved a change, that change could be made to software/hardware, and made ready for operational use. However, only the Operations Manager could remove/fallback to previous configurations without CM actions. The Operations Manager functioned as an approval authority of one. No one else got a vote. So CM could approve a change, but if the OM had any concern, the change would never reach the operational floor. Change of operational procedures was the domain of the Operations Manager and he delegated this authority to subordinates for execution.*

Configuration Management in this definition applies to the use of hardware, software and procedures. Contrast this with the definition of CM given by the newsgroup *comp.software.config-mgmt*:

*There are a number of different interpretations. For purposes of this newsgroup, we are talking about tracking and control of software development and its activities. That is, the management of software development projects with respect to issues such as multiple developers working on the same code at the same time, targeting multiple platforms, supporting multiple versions, and controlling the status of code (for example beta test versus real release). Even within that scope there are different schools of thought:*

- *Traditional Configuration Management - Checking/checkout control of sources (and sometimes binaries) and the ability to perform builds (or compiles) of the entities. Other functions may be included as well.*
- *Process Management - Control of the software development activities. For example, it might check to ensure that a change request existed and had been approved for fixing and that the associated design, documentation, and review activities have been completed before allowing the code to be "checked in" again.*

While process management and control are necessary for a repeatable, optimized development process, a solid configuration management foundation for that process is essential.

This definition introduces the concepts of Configuration Management, Process Management, Problem Management and Requirements Management. This newsgroup is concerned with software development, but their approach could easily apply to any development process. This CM approach handles the development environment, but does not handle the additional strain of an operational system. Since development and operations are often asked to co-exist, the overall CM process should be defined at the program level, and not controlled by either the operators or the engineers.

## 2.0 Configuration Management Tools

There are a number of commercially available tools to help simplify the Configuration Management task. They include revision control software, requirement management and tracking software, and others. Don't be fooled into letting a software tool define your process! The CM process is larger than the tools that help carry it out, and hence specific tools are unlikely to be named as an important part of these Best Practices. The application of different tools may provide insights, but the underlying process is more important.

## 3.0 Best Practices

- 3.1 *ABSOLUTELY EVERYTHING* Must be Covered Under the CM plan!!! Neglecting seemingly un-important aspects will introduce ambiguity, invites "judgement calls", and creates headaches for everyone. At one time, the CM process at CERES only applied to the core mission software. We eventually realized that configuration

scripts, passplans, and even procedures had a great impact on the success of our missions, and we incorporated these areas into our overall CM process. It is easier to start out doing this, rather than trying to get your staff to implement and conform to a more restrictive CM process after significant development has occurred in an uncontrolled environment. If not practical to implement everything under CM, then a careful evaluation must be made of the areas not covered to assess their possible mission impacts. Some form of change control should be followed. For example, for changes to products or databases used in real-time, change authority could be given to the "shift leaders", or to other lead individuals for other areas.

- 3.2 Implement a Default CM Process and Leave no Gray Areas. The CERES approach to CM is that any change needs to be covered by a process. A change is anything that changes bits on the hard drive, or any physical configuration. This includes plugging in a network cable, or powering up a workstation. In more complex systems; however, a distributed change authorization process may be necessary in order to make the system manageable. The CERES default process is the Change Request (CR) process. Each CR must be approved by the Requirements Screening Panel before it is worked, and this panel consists of both peer review and organizational buy-in. Now, this is obviously too restrictive to be feasible. The loophole is that anything can be pulled out of the CR process, but only if another *approved* process is created to cover this activity. This still allows leadership to manage the configuration by approving the way things are to be done, but the actual working level has the opportunity to do things the way they want to do it. Examples of things that CERES has put under separate processes are maintenance actions, system administration procedures, orbit analysis procedures, real-time operations procedures and mission planning activities. Remember that most real-time activity results in data being generated or modified on your system, and it is wise to consider what exactly is happening and what the impacts might be.
- 3.3 Include Procedures in the CM Process. Procedures are developed and approved as a method of controlling how the satellite mission is conducted. Once procedures are put in place, any changes should also be approved at the same level as the initial procedure. Otherwise, the organization loses the ability to accept responsibility for mission success or failure. The program lead can make conscious decisions to delegate approval authority to an appropriate level, but this delegation should be clear and specific. This also includes any products associated with the procedures such as scripted command files, memory loads, and telemetry displays. Date and revision numbers, as well as a history of the changes to the product should be a part of the product itself.
- 3.4 Document Your CM Process. Having a CM process that is undocumented, and learned through OJT is an easy trap to fall into. This is even truer if you rely heavily on software tools to handle your CM. It is hard to hold people responsible for following the process when it is not clearly spelled out, and this problem is only compounded by personnel turnover. CERES has found that not only are there fewer deviations from the CM policy when it is documented, but the staff is also quicker to learn the process, and more willing to follow it.

- 3.5 Allow for an Accelerated Path Through the CM Process. It is never acceptable to ignore the CM process in an emergency. If the process does not allow emergency database updates in an anomaly, or quick recoveries from catastrophic system failures, then fix the process, but don't ignore the process thinking it will save you time. This means only including steps, checks, and decision points in your process, which truly are important. The approval authority for each of these steps should be available whenever operations are being conducted, so this means having a documented backup in case the original person cannot be reached. This allows a change to be pushed through out-of-cycle, while reserving all decisions for the appropriate position or level. On the flip side, there are very few, if any at all, changes which must be made immediately. The current configuration has already been tested, approved, and baselined, and if it has worked for the last few months, it will probably continue to work at least as well for the next few hours or days.
- 3.6 Consider Implementing Audits. Audits ensure that changes, which have been approved, are actually incorporated into the operational environment.
- 

END

**SATELLITE MISSION OPERATIONS BEST PRACTICES  
BEST PRACTICES WORKING GROUP  
AIAA SPACE OPERATIONS AND SUPPORT TECHNICAL**

\*\*\*\*\*

**TRAINING AND CERTIFICATION**

**DAVID FULLER**  
**david.fuller@macconnect.com**

**April 10, 2001**

**1.0 Introduction**

A training program should have two purposes. One is to shape the culture, or behavior, of the Flight Operations Team (FOT) as it interacts with internal and external interfaces. The second is to teach the FOT how to operate the ground and flight systems. Training that encompasses both of these concepts will help ensure mission success.

In training, one of the most important concepts to remember is "Train as you fly, fly as you train." In other words, define your operations culture early. Then develop a training plan that best creates that culture. Follow through by operating the satellite in the same manner as you have trained.

**2.0 Training Goals**

The ultimate goal of a training program is to reduce Personnel Errors (PEs). To achieve this goal some basic principles should be kept in mind.

2.1 A Mission Operations Control Center is a unique environment, and the training program should include teaching behavior patterns that ensure effective behavior in that environment. A training program that includes this concept will ensure consistent behavior among the members of the FOT.

2.2 The key to proficiency in any activity is practice and more practice. Simulations and rehearsals of routine and contingency operations prepare the team members for all situations and lessen the chance of PEs during real-time activities.

2.3 Many of the systems used for satellite control are unique to the mission, and often require specialized instruction and practice to make the members of the FOT proficient.

### 3.0 General Training Program Requirements

This list is by no means definitive, but the members of the SOSTC Best Practices Working Group have found these requirements very useful in setting up a training program.

- 3.1 Develop a Training Plan. A poorly planned training program will be reflected in the team members that come out of it. Have an experienced FOT member write it or work closely with the training personnel. At the very least, thoroughly review it before release. Include clear goals, expected results, and schedules for completion.
- 3.2 Develop a Skills/Knowledge Description for Each Position. This should be done in as much detail as possible, along with initial and recurrent training, and certification requirements. Such details make it easier to judge if the trainee has met the requirements, and helps the trainee to understand what goals to strive for.
- 3.3 Review and Update Training Plans Periodically. This will ensure relevancy with current mission requirements. If the operations have developed in a different direction than anticipated, be sure the training plan also evolves with it. Ensure that the FOT is involved in all reviews.
- 3.4 Maintain Complete and Accurate Training and Certification Records. This should be done for each individual, and make them easily accessible to both trainer and trainee.
- 3.5 Staffing Levels Should be Adequate. This is necessary to allow some team members to be in training so that attrition does not leave operations understaffed and at risk. Also rotate FOT members into trainer positions to ensure distribution of the knowledge base. This will provide breaks from continuous console work and reinforce knowledge that is not used frequently.
- 3.6 Training Should be in the Form of Computer Based Training (CBT). This training should include training material, lessons, and self-tests. The training software should reside in a central server for ease of maintenance and to ensure that only the latest, approved version is used for training.
- 3.7 Training Modules or Sections. These should focus, as much as possible, on simple skills. Once the simple skills are learned, they can be combined into more complex activities.
- 3.8 Involve FOT Members. This is important for the success of spacecraft and ground system Integration and Test (I&T) processes. This not only provides good training in system idiosyncrasies that may not be adequately documented by the design team, but also helps to promote an operations oriented point of view during the I&T process.
- 3.9 Ensure that Design/Testing Knowledge is Documented. This documentation should be passed on to FOT. This can be accomplished by involving, as much as possible, key members of the FOT early in the design and testing process.
- 3.10 Develop Rehearsals/Simulations. Anomaly/Contingency scenarios are essential in preparing the FOT to handle emergency situations. Each crew should experience these and conduct "crew reviews" (peer reviews) of their actions and possible consequences. These should exercise both nominal and contingency operations. This

will provide excellent feedback for procedure development, and help desensitize the FOT to emergency situations and reduce panic responses. Even fatal, non-recoverable scenarios can be useful in this regard. Multiple rehearsals allow for repetitive training as well as specific focused events. Rehearsals can include: Communications rehearsals between the launch center, the operations center, and the factory; Launch and deployment rehearsals; and Day-in-the-Life rehearsals.

#### 4.0 Initial Training

- 4.1 Identify mission requirements and develop training modules to address them.
- 4.2 Train and certify the FOT before launch. Orbit raising and in-orbit test should not be a period for training of the FOT.
- 4.3 Train core skills first, then cross train. Ensure that a complete FOT is prepared to fly the mission, then cross train members.
- 4.4 Ensure that new hires have basic space operations training as well as mission specific training. This will ensure a common knowledge set for the FOT.

#### 5.0 Crew Resource Management

All air carriers have trained their flight crews in Crew Resource Management (CRM) skills since it was shown in the 1980's that it reduces PEs. It has also been shown to be effective in nuclear power control rooms and medical operating theaters. Satellite controllers work in a similar, real time environment, and the following skills, if practiced by the FOT, will help reduce PEs.

- 5.1 Leadership/Followership and Teamwork. Knowing how to lead and follow are important parts of teamwork. Leaders must know how to distribute tasks, keep track of the overall situation, and direct the team's attention as needed. Just as important, leaders must listen to their team members and utilize their expertise and talents. Followers must be able to react to their leader's direction, but also know how to help the team leader choose the correct path.
- 5.2 Communications. Many PEs can be related to poor communications. The use of standard terminology lessens the risk of misunderstanding in both internal and external interfaces. Failure to initiate communications has also been shown to a significant factor in many incidents.
- 5.3 Situational Awareness. Simply put, knowing what's going on and when it's going to happen. Situational awareness is especially important when the FOT is focused on a problem, and other problems go unnoticed until its too late.
- 5.4 Task Prioritization. Mission management should establish clear priorities of tasks to help the real time controllers manage their workload during normal operations and especially during contingency operations.

- 5.5 Event Logging. Keeping an accurate and timely log of events is invaluable in not only tracking the day's activities, but also reconstructing those activities and actions of the FOT weeks or months later.
- 5.6 Workload Management. Task overload can occur quickly during contingencies. Not only must the team leader be aware of the distribution of workload, but individual team members must be able to recognize overload and ask for help.

## 6.0 Resources

There are many resources available to help train the FOT. Each member of the team as well the managers should regard everything as an opportunity to learn more about the systems on which they will work.

- 6.1 Vehicle Assembly, Integration and Test. One of the best ways to ensure that system idiosyncrasies are passed on to the operations team is to get the FOT involved with assembly, integration and testing of the satellite
- 6.2 Manuals, Specifications, and "As Built" Documents. After ensuring that the "as built" documentation and manuals are as accurate as possible, these may be the only reference source once the vehicle is launched.
- 6.3 Lectures and Classes. These will provide a good basis of knowledge, and help the FOT to begin working as a team.
- 6.4 Simulators and/or Rehearsals. These are the only way to practice nominal and contingency operations, and to refine procedures, without risk to flight hardware. They also build teamwork and help desensitize the team to contingencies.
- 6.5 Mentors and On-The-Job-Training (OJT). No matter how well trained by manuals and rehearsals, new team members should be assigned a mentor who will show them the ropes, help integrate them into the team, and evaluate progress.
- 6.6 On-going Operations. Visiting existing satellite operations centers prepares the FOT for the operational environment and provides insight into actual satellite operations and operational paradigms.

## 7.0 Recurrent Training

Training should be considered an ongoing process. Recurrent training should be based on frequency of performance and criticality of performance of the activity. Activities that are performed on a routine basis are continually reinforced and do not require the same amount of training, as do activities that are seldom performed. Recurrent training should be developed with the following points in mind.

- 7.1 FOT Membership. This will change through attrition, promotion, and transfers.
- 7.2 Team Members. They should be cross-trained in other positions.

- 7.3 New Technology, New Procedures, and New Systems. These require that the FOT be familiar with them.
- 7.4 Routine Procedures. Those that are not used frequently should be trained on a regular basis.
- 7.5 Critical and Contingency Procedures. These should be trained on a routine and continual basis to insure the desired response by the FOT.

## 8.0 Certification

Certifying an individual to perform the functions of a given position means that the individual can do those tasks without direct supervision. Requiring re-certification to work in a position helps ensure that the individual is fully capable to perform those functions, and helps motivate the individual to stay current. Levels of certification also motivate and encourage continued growth in the knowledge of satellite and ground capabilities and characteristics.

- 8.1 Define the level of knowledge required for each position.
- 8.2 Decide the time period of certification: semiannually, annually, etc.
- 8.3 Develop computer-based self-tests for personnel.
- 8.4 Evaluation should be by the team leader/supervisor as well as the training officer.

---

END

**SATELLITE MISSION OPERATIONS BEST PRACTICES  
BEST PRACTICES WORKING GROUP  
AIAA SPACE OPERATIONS AND SUPPORT TECHNICAL COMMITTEE**

\*\*\*\*\*

**PROCESS IMPROVEMENT**

**TREVOR SORENSEN**

Sorenst@ukans.edu

**UNIVERSITY OF KANSAS**

**March 30, 2001**

1.0 Introduction

A famous general (attributed to Prussian Field Marshall 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. 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 section will look at 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.

## 2.0 Defining and Using Measure 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:

- 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)
- Cost of operations (comparison of actual versus projected costs)
- Response time and flexibility of the mission planning and operations process
- Efficiency (cost/data collected)

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).

2.1 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 or 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 be tracked as a separate error. These errors can be either human or machine.

- 2.2 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).
- 2.3 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.
- 2.4 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--).
- 2.5 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).

- 2.6 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.
- 2.7 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.

### 3.0 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.

<u>MoE</u>	<u>Measurement Method</u>
Error Tracking	Logs (automatic and manual) kept for each step in process to record any errors found and corrected
Complexity Factor	Determine from timelines/schedules
Rush Factor	Log the time of completion for each step in process including execution on the spacecraft
Effort Factor	Log the time spent by each person on each mission event being processed
Response Factor	Determine from times in the log and the complexity rating of each mission event
Fatigue Factor	Determine from hours worked
Morale Factor	Use routine total quality surveys of personnel and note the turnover rate and the complaint rate

#### 4.0 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).

#### 5.0 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).

## 6.0 Reporting Mechanism and Dissemination and Further Implementation

- 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.

## 7.0 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.

---

END

**SATELLITE MISSION OPERATIONS BEST PRACTICES  
BEST PRACTICES WORKING GROUP  
AIAA SPACE OPERATIONS AND SUPPORT TECHNICAL COMMITTEE**

\*\*\*\*\*

**GROUND SYSTEM DEVELOPMENT**

**EVAN ELLER**  
**Evan.Eller@csconline.com**  
**HONEYWELL**

**April 12, 2001**

**1.0 Introduction**

This section describes Best Practices for ground system development, including both the development process and ground system design. Development process practices include such items as who should be involved, the reviews conducted during development, the design process, component selection, and component delivery, testing, and control. Ground system design guidelines include such items as on-line access to mission information and using TCP/IP communications, open and upgradeable systems, common hardware platforms throughout as much as possible, user configurable capabilities, and providing automation for monitoring and non-mission-critical control capabilities with robust paging capabilities. Much of this material originated from lessons learned identified by the EUVE team at UC Berkeley.

**2.0 Development Process**

**2.1 Staff and Reviews**

- 2.1.1 Operations staff/engineers should be involved early in the ground system design.
- 2.1.2 The ground system design team should include a mixture of those experienced in space operations ground systems and those with recent information technology training.
- 2.1.3 Require that system users, spacecraft engineers, ground system developers, and maintenance personnel work closely together in order to facilitate the development and maintenance process, minimize unnecessary delays, and ensure that the system meets user requirements and needs. Whenever possible, collocate mission team members.

- 2.1.4 Continued access to software developers (not just maintainers!) is critical for the rapid and reliable implementation of software enhancements or modifications.
- 2.1.5 Early internal project reviews are very important and beneficial. They allow for design changes, evaluation of the process and the team members, the suggestion of alternatives, and the identification of relevant drivers and risks. Conduct at least one thorough design review of the Control Center in order to achieve technical consensus and focus. Two such reviews are better: a preliminary one to review and critique the operations concept/plans and to raise relevant issues and concerns, and a final review to demonstrate satisfactory resolution of those same issues and concerns.
- 2.1.6 External reviews with independent review panels should also be conducted. For new missions, these should be conducted as subsystem design reviews coordinated with overall mission reviews, such as Functional Design Review, Preliminary Design Review and Critical Design Review. Review panels should be small and composed of people with directly related experience.

## 2.2 Design Process

- 2.2.1 Clearly define the intended users and customers for the ground system. Don't attempt to serve so diverse a set of customers that it is difficult to define consistent requirements. Categorize "needs" vs. "wants" and focus on the former first. Well-defined objectives and requirements are critical to successful development.
- 2.2.2 Early in the design phase promote extensibility by keeping as much generality of function in the design as possible.
- 2.2.3 Create common conventions for all interfaces: command, telemetry, etc.
- 2.2.4 Organize software applications into functional packages. This allows a modular design, which provides the flexibility of replaceable components.
- 2.2.5 Plan from the beginning for future system extensions as technology changes.
- 2.2.6 Always define the key interfaces for any system or subsystem before starting the implementation. Break down the large systems into subsystems with well-defined interfaces.
- 2.2.7 Build functionality into stateless libraries with strictly defined interfaces. This avoids duplication of code, simplifies maintenance, and reduces development and testing time.

- 2.2.8 Set up an infrastructure that will lend itself to adding automation. Introduce automation first that has been proven on other missions or does not involve mission safety.

### 2.3 Component Selection

- 2.3.1 Whenever possible, use commercial hardware. Make arrangements with vendors for quick supply of critical items, even for redundant systems. If you must use customized systems, obtain, in advance, spare parts to avoid work delays or system downtime during either development or operations.
- 2.3.2 Investigate and implement Commercial Off-The-Shelf (COTS) solutions wherever they meet program requirements, including such characteristics as reliability as well as functional and performance requirements. COTS applications are generally already operational, well documented, are easy to use, and come with some level of technical support (the more the better). These attributes may make COTS significantly cheaper to use in the long run, and easier to manage than in-house software development, albeit at the sacrifice of some level of flexibility. However, if a COTS evaluation determines that it does not meet all your requirements, the level of effort required to supplement it for full support of your operations must also be evaluated.
- 2.3.3 Avoid using languages and tools with a small user base that are not open source.
- 2.3.4 When possible, use "standard" languages (e.g., ANSI C, html) for portability purposes.
- 2.3.5 Always take the time to search the Internet for open source software that either does what's needed or can be easily modified. For example, UCB has made good use of non-UCB-developed freeware UNIX shell utilities; they have also provided a number of homegrown ones on their WWW site (<http://www.cea.berkeley.edu>) that have been used by other projects (e.g., Gravity Probe B).

### 2.4 Component Delivery, Testing, and Control

- 2.4.1 Maximize the use of Configuration Management (CM) and control mechanisms (e.g., via the UNIX MAKE and SCCS/RCS utilities) for source code, documentation, procedures, etc. All items should be under CM before testing begins.
- 2.4.2 Establish a well-defined software release mechanism, which will instill organization, control, and tracking, albeit at the cost of a little extra (value-added) bureaucratic overhead.

- 2.4.3 Make software releases distinct from software installations. The EUVE Project at UCB implemented this by developing the Flexible Software Installation (FSI), a UNIX-based freeware package (which is available on their WWW site).
- 2.4.4 Documentation should be in a standardized and portable format and be easily maintainable. The development and maintenance of WWW-based (e.g., html and text) documentation is relatively inexpensive, and greatly simplifies the cross training of personnel.
- 2.4.5 Use a problem or bug tracking system. Freeware packages exist (e.g., GNATS) that may be useful.
- 2.4.6 Recognize the importance of the spacecraft simulator in the timeline. It can be an important part of testing and training.
- 2.4.7 Consider the use of a "project definition" policy for all software development requests. Late in the EUVE mission UCB implemented this policy due to the continuous in-house requests for additional software tools. The policy required that all requests be formally written up and submitted for review. The small amount of extra-required overhead served to filter out unimportant requests, while ensuring that requests were clearly thought out in advance of submission for review. If approved, requests were then prioritized for development. This should be part of your CM process. (See paper on Configuration Management)
- 2.4.8 Allow for the testing of the ground system with the spacecraft as early as is prudently possible. These tests should evolve to the point that the actual operators and engineers are running operational procedures in a mission like environment using the ground system in all mission phases. This includes launch and ascent, activation and checkout, and normal operations procedures. If possible, continuous multi-day testing can expose unforeseen problems during the development process.
- 2.4.9 Consider phasing in any major changes (e.g., automation). Multiple phases not only allow people to adjust to incremental changes, but also allow for the implementation of the easy things first.

### 3.0 Ground System Design

#### 3.1 General

- 3.1.1 Consider providing on-line organized access to all mission telemetry that makes it extremely easy and convenient to perform any on-demand science or engineering investigation. With today's computer technology, and the decreasing prices for storage media, this strategy may well provide an excellent return on investment in terms of data analysis and results.

- 3.1.2 Choose fast, reliable, flexible, open-ended, and proven data storage systems whose capacity can be upgraded to handle a great deal more data than the mission originally may envision.
- 3.1.3 Maximize the use of technologies like RAID that promote reliability and minimize downtime. Mission critical hardware and software (e.g. command servers) should have hot backups or other technologies that promote reliability and minimize downtime such as RAID (Redundant Array of Independent Disks). All ground systems should be configured to simplify backup procedures by using centralized data storage.
- 3.1.4 Use a highly integrated operating system that is reliable, powerful, flexible, and customizable. The EUVE team at Berkeley recommends Unix for these qualities, and its shell scripting capabilities alone have allowed all personnel—not only programmers—to implement incremental yet significant improvements across all areas of the EUVE Project. The downsides they experienced have been the need for better user training and the relative high expense of, and poor support for, UNIX versions of various common software applications.
- 3.1.5 Try to limit the number of computer hardware platform and operating systems in use (i.e., only one, if possible) in order to simplify and minimize network complexity and associated maintenance.
- 3.1.6 Missions which include data distribution among multiple locations should ensure that their networks can handle a great deal of Internet traffic. This is particularly important given the recent expansion in use of the WWW, and in the general migration for satellite operations and data transmission and delivery via the Internet. Pay close attention to NASA security and network bandwidth issues when purchasing routers and other network equipment.
- 3.1.7 Missions should use a common standard computer communications protocol. This will preclude the need for proprietary protocols and their associated hardware/software and will greatly simplify system development, implementation, operation, and maintenance. The current ubiquity of TCP/IP makes it an obvious candidate for the near future.
- 3.1.8 The use of relational databases or object-oriented databases is extremely valuable for managing data such as command and telemetry definitions and long term engineering trending statistics. However, it is crucial that these databases be properly designed and implemented by a knowledgeable database programmer. If poorly implemented such databases can lead to major maintenance headaches and expenses. Also, database software will typically add overhead time to processes that use them.
- 3.1.9 Make maximal use of the WWW for any project requiring diverse geographic data distribution, as it can greatly simplify global communications. Its inherent ease of use and platform-independent nature make it an ideal means of on-line

communications, and a great way to save money (e.g., paper, phone, and mailing costs). A local WWW server does, however, require some maintenance time, but this can be minimized if well managed (e.g., via the use of some up-front and consistent internal standards and controls).

- 3.1.10 Use programming languages and tools that are appropriate for the task at hand, and do not dictate the use of a particular language and/or tool without consideration for the specific task.
- 3.1.11 Provide user-configuration capabilities, including command line access. It is often convenient to temporarily modify the monitoring rule parameters (e.g., limit values) or to screen pages for expected conditions (e.g., non-standard payload configurations). This should be implemented within an overall configuration management structure that establishes rules for what can be changed under different levels of authority.

## 3.2 Automation

- 3.2.1 User interface tools for an automated system should be focussed on providing a means to determine the operations current status and to interrupt the automation when necessary. Automation does not need to provide routine display of all spacecraft telemetry, since the purpose of such a system is to replace manual monitoring. Such detailed displays can degrade overall system performance and require significant extra development and maintenance costs.
- 3.2.2 Automated telemetry monitoring system can be greatly simplified by not including capabilities to send commands to the spacecraft. Such immediate ground-based commanding has never been required for EUVE; on-board automated safety mechanisms (e.g., TMON/RTS or built-in safe modes) are typically used instead. The main focus of the system should be to detect anomalies and page someone who will then investigate and take corrective action. At this time this strategy is still much cheaper and more reliable than trying to build a smart system that can on its own diagnose problems and take corrective action.
- 3.2.3 Implement a method of persistent paging (i.e., at regular intervals) that requires an external acknowledgment to turn off. The EUVE Project at UCB uses multi-level paging -- to primary (i.e., the EUVE ACE), secondary (i.e., select engineers), and "other" (i.e., everyone) groups -- in order to ensure that pages are received by someone within a reasonable period of time.
- 3.2.4 The system should, preferentially, have some way to group together related problems for paging.
- 3.2.5 It is useful and recommended that there be a separate stand-alone system set up with the sole purpose of monitoring the primary telemetry monitoring system.

---

END

**SATELLITE MISSION OPERATIONS BEST PRACTICES  
BEST PRACTICES WORKING GROUP  
AIAA SPACE OPERATIONS AND SUPPORT TECHNICAL**

\*\*\*\*\*

**PRE-LAUNCH SPACECRAFT OPERATIONS DEVELOPMENT  
AND TEST**

**SCOTT WILLIAMS**  
Scott.williams@stanford.edu  
**STANFORD UNIVERSITY**

**December 27, 2000**

1.0 Introduction

This section describes Best Practices for the pre-launch spacecraft operations development and test process. A key risk mitigation for any mission is to:

***Test the spacecraft and instruments as they will be operated and operate the spacecraft and instruments as they were tested.***

Process practices include who should be involved, systems engineering, development management, personnel development, spacecraft integration and test support, and operations testing and training. The purpose of testing is to validate the systems, procedures, timelines, and personnel for flight operations. An effective means to prepare systems and personnel for flight is for the operations team to operate the spacecraft and instruments during integration and test.

2.0 Operations Development Management Process

Use an integrated team approach to mission design and operations development following good systems engineering practices.

2.1 Provide early feedback to the project, spacecraft manufacturer, and instrument teams concerning the impact of the spacecraft hardware & software design on meeting requirements for both ground test and flight operations. Where necessary provide recommendations for changes to design and/or requirements.

2.2 A review of the Operations Concept should be included as an integral part of all formal mission reviews beginning with the Systems Requirements Review, both at the system as well as the element level (e.g., spacecraft, instrument, ground).

- 2.3 Evolve the spacecraft / instrument design and the operations concept in parallel during the development phase.
- 2.4 Conduct thorough design reviews of the control center in order to achieve technical consensus and focus: a preliminary review to critique the operations concept/plans and to raise relevant issues and concerns, and a final review at which satisfactory resolution of those same issues and concerns is demonstrated.
- 2.5 Support the development of a spacecraft simulator for validating procedures, developing operations timelines, and supporting operations team training.
- 2.6 Working with the development team, prepare operations manuals and training materials which describe spacecraft and instrument systems, define the procedures for normal operation, identify processes for recognizing mission threatening conditions, and highlight the contingency responses to spacecraft and instrument anomalies. Document failure modes for each subsystem, event tables, Standard Operating Procedures (SOP), contingency procedures, and command scripts.
- 2.7 Conduct pre-launch meetings with spacecraft developer, subsystem discipline engineers, instrument team members, and I&T engineers to specifically define on a mission phase, subsystem, instrument, and special event basis what parameters are the most important to be monitored in real-time (on a spacecraft and instrument configuration / state basis) and what one should be looking for in those parameters.

### 3.0 Operations Team Development Process

Include the operations team and science team members early in the instrument design process.

- 3.1 Include the operations team in the science team discussions of mission changes and development of new procedures after launch.
- 3.2 Cross training and multiple job responsibilities is essential to low-cost operations. Operations engineers should be controllers, schedulers, and planners, as well as ground systems and operating systems experts.
- 3.3 If possible, include the operations team in integration & test (I&T) planning and implementation. Effective training can be achieved by using the operations team to operate the spacecraft and instruments during the I&T process.
- 3.4 Cross train personnel from integration and testing to support launch and early orbit operations. They can provide surge operations team staffing and helpful engineering support for launch and early orbital operations.

#### 4.0 Operations Systems Usage

Ensure a stable operations development environment following good systems engineering practices.

- 4.1 Recognize and take advantage of the potential for cost savings from using common systems for flight operations and for integration and testing. The use of a common system could avoid software translations and transfers, decrease validation requirements, reduce risk and lower cost. Particular advantages can be found in using common command and control software environments. Early planning can allow the use of ground support equipment (GSE) developed to support integration and testing for operations support. I&T GSE can provide a quick, low-cost, and proven capability for monitoring instrument or spacecraft performance.
- 4.2 Plan to use pre-launch and testing phases as training opportunities. Ensure that design/testing knowledge is documented and passed on to operations team.
- 4.3 Ensure the flight and ground software is stable, under configuration control, well documented, and thoroughly tested well before launch.
- 4.4 Implement increasing levels of configuration control by development and/or mission phase as appropriate.

#### 5.0 Spacecraft Test & Simulation Support

An effective means to prepare systems and personnel for flight is for the operations team to operate the spacecraft and instruments during integration and test.

- 5.1 Support spacecraft sensor verification tests to validate understanding of sensor geometry and performance.
- 5.2 Support ground system compatibility tests and verify telemetry conversion values.
- 5.3 Use the I&T process as an opportunity to test and verify all operations modes before launch. The exercise of flight procedures and timelines are excellent ways of verifying spacecraft capabilities.
- 5.4 Operations simulations are an excellent means for testing software and associated user interactions. Carefully specify and develop spacecraft and instrument simulators with the highest possible fidelity within program constraints. Include the ability to test anomaly response by using simulators to inject faults.
- 5.5 Conduct simulations of key orbit maneuvers, spacecraft modes, and contingency plans to verify software and procedures.

## 6.0 Operations Testing

The purpose of testing is to validate the systems, procedures, timelines, and personnel for flight operations.

- 6.1 Base integration and testing on operations plans and procedures. Combine the integration, testing, and operations test plans. Use operations procedures in the test environment and capture systems responses and behaviors during integration and testing. This will avoid duplicate tests and procedures and ensure that systems are tested as they will be used.
- 6.2 If possible, use the operations team to conduct tests with developers in support.
- 6.3 Combine as much as possible validation and readiness testing for flight operations with the integration and test of the ground and space elements.
- 6.4 Allow for the hands-on control of the spacecraft by the operations team as early as possible. For example, begin monitoring spacecraft telemetry during all powered integration and test activities as soon as the ground systems are capable of doing so.
- 6.5 Allow for the testing of the ground system with the spacecraft as early as is prudently possible.
- 6.6 An end-to-end system testing philosophy from the spacecraft to the science data processing software will ensure that delivered systems are robust and reliable, and that operations personnel are well trained in the usage of the system.
- 6.7 Use the flight data processing facility to process and archive data acquired during integration and testing and simulated activities.
- 6.8 During flight operations, test and commands and procedures with simulators prior to use. This will validate the procedures and familiarize the operators with expected performance.
- 6.9 Use the operations team to perform pre-launch calibration tests, to process the test data into calibration parameters, to implement the calibrations in the telemetry database, and develop limit monitors and values. The familiarity with calibration procedures gained by the operations team will reduce risk for any on-orbit calibration activities. In addition, the knowledge gained about the contents of the telemetry will improve the operations team ability to respond to out-of-limits conditions.
- 6.10 Exercise limit monitoring in ground systems during integration and test activities to gain experience with out-of-limits conditions and responses.

- 6.11 Identify, define, and document those remaining spacecraft and / or instrument failures requiring time critical ground intervention and document clear and concise recovery procedures for each.
- 6.12 Identify, define, and document those remaining spacecraft and / or instrument failures requiring time critical ground intervention and document clear and concise recovery procedures for each.
- 6.13 An end-to-end system testing philosophy from the spacecraft to the science data processing software will ensure that delivered systems are robust and reliable, and that operations personnel are well trained in the usage of the system.
- 6.14 Use the flight data processing facility to process and archive data acquired during integration and testing and simulated activities.
- 6.15 During flight operations, test and commands and procedures with simulators prior to use. This will validate the procedures and familiarize the operators with expected performance.
- 6.16 Use the operations team to perform pre-launch calibration tests, to process the test data into calibration parameters, to implement the calibrations in the telemetry database, and develop limit monitors and values. The familiarity with calibration procedures gained by the operations team will reduce risk for any on-orbit calibration activities. In addition, the knowledge gained about the contents of the telemetry will improve the operations team ability to respond to out-of-limits conditions.
- 6.17 Exercise limit monitoring in ground systems during integration and test activities to gain experience with out-of-limits conditions and responses.
- 2.6 Identify, define, and document those remaining spacecraft and/or instrument failures requiring time critical ground intervention and document clear and concise recovery procedures for each.

---

END

**SATELLITE MISSION OPERATIONS BEST PRACTICES  
BEST PRACTICES WORKING GROUP  
AIAA SPACE OPERATIONS AND SUPPORT TECHNICAL COMMITTEE**

\*\*\*\*\*

**FLIGHT DYNAMICS**

**KEN GALAL**  
**kgalal@mail.arc.nasa.gov**  
**NASA AMES RESEARCH CENTER**

**March 30, 2001**

**1.0 Introduction**

Flight Dynamics support consists of all analysis and operations necessary to determine and control the orbit and attitude of a spacecraft. Operations entail the generation of a number of products, including definitive orbit and attitude solutions, orbit and attitude predictions, event predictions (e.g. station contacts, eclipses, etc.), and orbit/attitude control system command data (e.g. orbit state vectors, star catalogs, sensor biases, maneuver times, etc.). Flight Dynamics analysis consists of a variety of pre/post-launch analysis topics involving orbit/attitude mission design and spacecraft maneuver planning. As Flight Dynamics functions are an important part of spacecraft operations, there should be close coordination between analysts performing these functions and Flight Operations Team (FOT) personnel who control the spacecraft. Consequently, it is often advantageous to perform as many of these capabilities as possible within the Control Center.

**2.0 Requirements Analysis**

Consistent with the general recommended best practice of involving the operations team early in the requirement definition process, Flight Dynamics personnel should take an active role in defining operations concepts and influencing mission design. Feedback should be provided to the project/science team concerning the impact of requirements on mission operations costs and risks, and recommendations provided on ways to minimize each.

- 2.1 *Pre-Launch Error Analysis*: Perform pre-launch error analysis to assess the ability of a proposed spacecraft sensor complement (e.g. attitude sensors, GPS, etc.) and processing algorithms to meet onboard attitude and orbit determination requirements.
- 2.2 *Orbit Data Requirements*: Establish tracking data requirements and orbit determination/prediction processing requirements necessary to achieve desired orbit accuracy.

- 2.3 *Cost Analysis*: Perform a cost analysis with regard to meeting project Flight Dynamics requirements and provide feedback on possible cost savings that could be achieved by relaxing tight accuracy requirements that necessitate intensive ground operations (e.g. post-processing of telemetry, in-flight sensor calibration, more frequent parameter uploads, etc.).
- 2.4 *Onboard Processing*: When possible, perform definitive attitude and orbit determination onboard to minimize operations costs.
- 2.5 *Product Delivery Schedules*: When possible, negotiate product delivery requirements that promote flexibility in terms of staffing (e.g. weekly deliveries vs. daily). Also, if possible, automate product delivery or make it available on the Internet for user access as needed.
- 2.6 *Orbit Maintenance Requirements*: When possible, negotiate orbit maintenance requirements that minimize the frequency of maneuvers (e.g. +/- 20-km altitude tolerance vs. +/- 10 km). Re-evaluate and update orbit maintenance requirements as necessary throughout the vehicle life based on propellant remaining and changes in expected mission lifetime.
- 2.7 *Spacecraft Autonomy*: Where practical, recommend the use of autonomous spacecraft capabilities (e.g. momentum management, solar array slewing, initiation of contingency modes) in order to reduce operations life-cycle costs.

### 3.0 Spacecraft Design Evaluation

Provide early feedback to the project and spacecraft manufacturer concerning the impact of spacecraft hardware and software design on the ability to meet orbit/attitude operational requirements. Where necessary provide recommendations for changes to designs and/or requirements. As a rule, straightforward operations should always be a goal in designing a spacecraft. It is recognized, however, that in some cases tradeoffs may warrant more complicated operations in the interest of meeting difficult requirements and reducing spacecraft cost/complexity. In such cases, the project must be made aware of the operational impact and risk associated with these tradeoffs.

- 3.1 *Telemetry Parameters and Rates*: Verify adequacy of key attitude and orbit control telemetry data and rates for use in ground support of spacecraft (e.g. propulsion system temperatures/pressures and thruster history for orbit maneuver planning/calibration, attitude sensor data for calibration, etc.).
- 3.2 *Attitude Thruster Disturbances*: When possible, ensure attitude thruster firings occur in pairs with no net translational force imparted.
- 3.3 *Delayed Command Capability*: Ensure the ability to initiate key spacecraft events (e.g. orbit insertion maneuvers, attitude mode changes, etc.) via delayed command from memory.

- 3.4 *Thruster Qualification*: Verify that thrusters are qualified to fire over the duration of all possible burns without heat soak-back or plume-impingement concerns (consider also contingency cases where longer burns might be required, e.g., a thruster failure requiring longer burns on remaining thrusters).
- 3.5 *Battery Sizing*: Verify that batteries can meet expected eclipse conditions throughout the transfer, nominal mission, and extended mission orbits.
- 3.6 *Solar Radiation Torque Profile*: When possible, ensure spacecraft surface symmetry relative to center-of-mass to minimize momentum buildup/dumping resulting from solar radiation torques.

#### 4.0 Spacecraft Test and Simulation Support

It is crucial that operators take advantage of any opportunities to test their software directly with the spacecraft prior to launch. Whenever possible, manufacturer specifications on telemetry conversions, command functions, units, and spacecraft hardware configuration and performance should be verified (see Pre-Launch Spacecraft and Operations Development and Test Best Practice). This is particularly true for spacecraft hardware supported by Flight Dynamics functions. However, in addition to tests with the spacecraft, simulations should also be carried out to verify Flight Dynamics operations plans, procedures, and timelines. While it may be possible to rehearse certain capabilities (e.g. orbit maneuver design and product delivery) without the use of a high-fidelity spacecraft simulator, other Flight Dynamics capabilities (e.g., attitude determination, sensor calibration, orbit determination) do require sophisticated modeling of spacecraft sensors and orbit/attitude geometry for realistic simulations. High-fidelity simulators can also be very helpful in support of spacecraft operator training, particularly for spacecraft with complicated attitude maneuver profiles. Typically, orbit determination capabilities are exercised using generic (i.e. spacecraft-independent) simulators with access to simulated spacecraft trajectories.

- 4.1 *Sensor Alignment*: Support spacecraft attitude sensor alignment verification tests to validate understanding of sensor mounting geometry and polarity.
- 4.2 *Sensor Performance Characteristics*: Where possible, participate in spacecraft tests in order to collect data on sensor performance (e.g. gyro bias temperature sensitivity may be observed during thermal vacuum testing).
- 4.3 *Telemetry Conversion Values*: Support ground system compatibility tests and verify attitude/orbit telemetry conversion values supplied by spacecraft manufacturer.
- 4.4 *Engineering Units*: Verify the use of consistent units between spacecraft designers, software developers, and operations personnel.
- 4.5 *Simulations*: Conduct simulations of key orbit maneuvers, spacecraft attitude control modes, and contingency plans to verify software and procedures.

## 5.0 Mission Analysis

Perform the following mission analysis activities as required for each mission, making sure that project, systems engineering and affected subsystem personnel (e.g. attitude control, thermal, power) are aware of results.

- 5.1 *Launch Window Determination (time of day and day of year)*: Based on mission requirements and propellant budget constraints, establish the available size of the launch window.
- 5.2 Launch Vehicle Analysis
  - 5.2.1 *Launch Vehicle Selection*: Provide recommendations to project on candidate launch vehicles that meet spacecraft mass and deployment orbit characteristics.
  - 5.2.2 *Consistency Verification*: Verify consistency of coordinate systems, units, and trajectory modeling parameters with launch vehicle personnel through the use of test cases.
  - 5.2.3 *Separation Vectors*: Establish requirements for the delivery of separation state vectors as needed.
  - 5.2.4 *End to End Modeling*: Ensure that the launch vehicle provider assumes responsibility for modeling any transfer orbit injection stage when possible.
- 5.3 *Propellant Budget*: Ensure an adequate spacecraft propellant budget that can meet all expected maneuver requirements (including transfer orbit and mission orbit maintenance maneuvers, and attitude control thrusting) and dispersions (both launch vehicle and spacecraft propulsion system) with sufficient margin (e.g. 10%).
- 5.4 Propulsion System Modeling
  - 5.4.1 *Blowdown Characteristics*: Obtain propulsion system "blowdown" characteristics (e.g. thrust vs. pressure and ISP vs. pressure curves) from manufacturer for maneuver planning.
  - 5.4.2 *Maneuver Performance Modeling*: Account for the effect of attitude control thrusting on orbit maneuver performance (e.g. thruster off-pulsing during orbit maneuvers for attitude control).
  - 5.4.3 *Orbit Disturbance Modeling*: Account for the effect of long-term attitude control thrusting on orbit evolution.

## 5.5 Spacecraft Launch Date and Deployment Orbit

- 5.5.1 *Launch Parameter Update Requirements:* Establish requirements for updates to launch vehicle and spacecraft parameters for each candidate launch date (e.g. injection state, injection stage ballast masses, pre-loaded separation or maneuver attitudes, etc.) and prepare data in advance.
- 5.5.2 *Launch Data Validation:* Generate required Flight Dynamics predicted products for all planned launch dates and across the entire launch window on each date to verify station schedules, shadow profiles, and sensor visibility/interference profiles.
- 5.5.3 *Performance Requirements:* Provide launch vehicle manufacturer with the nominal and three-sigma deployment orbit requirements.
- 5.5.4 *Dispersion Analysis:* Ensure that three-sigma deployment orbit altitude is sufficiently high that drag will not significantly impact the mission lifetime in the event of delays in spacecraft operational timelines.
- 5.5.5 *Risk Reduction:* Design deployment orbit and attitude geometry to minimize risk to the spacecraft in the event of delays in ground contact with the spacecraft. This includes a spacecraft deployment geometry with solar array orientation in a power-positive state and with antennas pointing in the direction of upcoming station contacts.

## 5.6 Transfer Orbit Design

- 5.6.1 *Maneuver Visibility:* When possible design maneuvers to occur in view of a ground station.
- 5.6.2 *Backup Station Coverage:* For key maneuvers (e.g. planetary orbit insertion) schedule backup station coverage if possible.
- 5.6.3 *Eclipse Analysis:* For geosynchronous and planetary transfer orbits, verify that transfer orbit does not unexpectedly pass through Earth shadow cone.
- 5.6.4 *Maneuver Modeling:* For large maneuvers, ensure that tracking data supplied to ground stations has maneuver Doppler characteristics modeled in order to prevent station from dropping lock.
- 5.6.5 *Station Selection:* For large maneuvers, consider using 26-meter stations with autotrack capability when link margins permit (i.e. within 0.01 AU of Earth) as a measure against dropping lock.
- 5.6.6 *Independent Verification:* Verify critical orbit maneuver planning conditions using independent software and/or personnel.

## 5.7 Mission Orbit Design and Maintenance Requirements.

5.7.1 *Mission Orbit Design*: Design mission orbits with minimum maintenance requirements for the given science objectives and launch capacity.

5.7.2 *Orbit Maintenance Requirements*: Negotiate orbit tolerances (e.g. on altitude, eccentricity, inclination, argument of periapsis and ascending node rotation) that maximize the time between maneuvers in order to minimize fuel use and operational risk (see FD best practice 2.6).

5.7.3 *Maneuver Scheduling*: When possible schedule maneuvers to occur early to mid-week to permit execution, validation and any contingency measures to be completed prior to the weekend (when internal/external supporting elements may be staffed down).

5.7.4 *Maneuver Database*: Maintain a database of maneuver and propellant conditions (maneuver date, pre/post maneuver orbit/attitude state, fuel remaining, thrusters in use, tank temperature/pressure, etc.) and update after each orbit/attitude maneuver.

5.7.5 *Maneuver Calibration*: Perform a calibration of the propulsion system performance following each orbit maneuver, and solve for thrust parameters to be used in the next burn (taking into account attitude offsets, tank pressures, temperatures, mass properties, thruster selection, etc.).

## 5.8 End-of-Life De-Orbit Requirements

5.8.1 *Propellant Reserves*: Ensure sufficient fuel reserves to meet de-orbit requirements.

5.8.2 *De-Orbit Planing*: Prepare a de-orbit plan prior to launch with re-entry conditions that minimize risk to life and property.

5.8.3 *De-Orbit Initiation Criteria*: Establish any control system failure criteria that should trigger de-orbit operations by control personnel.

5.9 *Orbit Determination and Acquisition Data Generation*: Develop an orbit determination and acquisition data generation plan for early mission and nominal mission support.

## 5.10 Contingency Planning

5.10.1 *Initiation Criteria*: Establish trigger points for entering into orbit/attitude contingency modes.

5.10.2 *Orbit Maneuver Contingencies*: Prepare orbit maneuver contingency plans that address, among others, ...

- Failed thrusters
- Delayed burn
- Attitude errors
- Premature burn termination

5.10.3 *Attitude Maneuver Contingencies*: Prepare attitude maneuver contingency plans that address, among others, ...

- Actuator failures (e.g. momentum wheel, thruster, etc.)
- Attitude sensor failure (e.g. gyro, sun/earth sensor, etc.)

5.11 *Attitude Sensor Calibration Plan*: Prepare an attitude sensor calibration plan, as dictated by accuracy requirements, for in-flight computation of sensor alignments and biases which can be commanded to the spacecraft (for improved onboard attitude determination/control), or used in ground software (to improve attitude knowledge).

---

END

**SATELLITE MISSION OPERATIONS BEST PRACTICES  
BEST PRACTICES WORKING GROUP  
AIAA SPACE OPERATIONS AND SUPPORT TECHNICAL COMMITTEE**

\*\*\*\*\*

**OFF-LINE SPACECRAFT PERFORMANCE ASSESSMENT BEST  
PRACTICES**

**RAY HARVEY**  
Ray.Harvey@jhuapl.edu  
**JOHN HOPKINS UNIVERSITY**  
**APPLIED PHYSICS LABORATORY**

**April 13, 2001**

**NOTE:** This also contains Real-time Monitoring Best Practices in some areas.

**NOTE:** Housekeeping telemetry refers mostly to that which is not science related; particularly things pertaining to health and status of the Spacecraft subsystems and instruments. This includes currents, temperatures, voltages, pressures, configuration telltales, attitude information, etc.

**1.0 Introduction to Off-line Spacecraft Performance Assessment Section**

Off-line Spacecraft Performance Assessment refers to the state of health monitoring, performance evaluation, and long term trending done outside of real-time satellite contacts with the ground station. It also includes the detection of anomalous behavior, which may have occurred outside of a real-time contact. Some aspects of real-time control are included here because of their applications in the off-line assessment environment. This section includes a list of practices that have been applied in the past in constructing an off-line spacecraft performance assessment system. The Best Practices described are based on a system that has been demonstrated to work very efficiently with a small number of staff members on a highly complex satellite.

**2.0 Maintain on-line Archive of all Raw Housekeeping**

Keep all spacecraft and instrument (if necessary) housekeeping data in raw format - on-line for easy access - for the entire mission, if possible. With the latest data storage technologies available, it is not as expensive as you would think. Make an approximation of how much housekeeping telemetry there will be; based on your data rates and the duration of the mission. Then add some margin to determine your storage requirements. One caveat may be that you are limited by budget constraints that may restrict you from maintaining all of the housekeeping on-line. In this case, try storing older data off-line; however, ensure that there is accessibility to this data, obviously making the retrieval as quick as possible.

### 3.0 Maintain a Critical Housekeeping Telemetry Data Base

Keep a critical subset of the housekeeping telemetry processed and stored in engineering units in a separate database so that you can perform ad-hoc queries of this data. Have several of these databases if necessary so as to include everything you feel you may like to run queries on. This capability is useful in anomaly investigations when trying to correlate several occurrences of the same situation. This also provides the ability to quickly respond to sponsor requests of "... what is the monthly average number of stars identified by the star tracker?" Or to a request by the Power Subsystem Engineer of "...how many times the Battery Depth of Discharge went below 50%?" Without having this type of database, in order to gather this type of information you would need to process a significant amount of raw data and perform manual searches, unless it could be imported into a relational database allowing queries. In any event, not maintaining this type of database capability greatly increases the amount of time involved in gathering this type of information.

### 4.0 Output Data in ASCII Text

The software tool used to extract and process the data into engineering units should output the data in ASCII text format. From there it can be easily imported to many different types of commercially available plotting packages including MS Excel, Pvwave, DaDisp, Matlab, etc. as well as a text editor. However, it should be kept in mind that large amount of data in ASCII format may cause the file to be extremely large and potentially unwieldy to move around. In these cases, it may prove more prudent to output and store the data in binary format. The majority of off-the-shelf packages, including those above, support binary format as well.

### 5.0 Compute Statistical Data

Ensure that the software, which processes the data into engineering units, is able to compute statistics such as min, max, average, and orbital average. Additional capabilities should include a Fast-Fourier Transform function and moving averages (such as one week, two-month, etc...).

### 6.0 Routine Plotting/Trending Recommendations

6.1 Generate and Output Plots Autonomously. The system should do this on a routine basis and at a convenient time for the assessment team to review them as you attempt to get them the latest and greatest data for reviewing. Other trending data products should be produced in a consistent format on a regular basis. The capability should exist for multiple X-axes. This allows an analyst to overlay a previous period with a current period to identify similarities and differences between the sets (see section 7.1).

6.2 Data Sets and Products. These should be defined for short (orbit by orbit, daily, weekly) and long term (several months to years) trending. The data sets and products should be updated, as necessary, as the spacecraft configuration changes due to failures, etc.

- 6.3 Trending Data. Ensure they are reviewed, analyzed, and found acceptable by knowledgeable individuals (preferably subsystem / instrument engineers) and/or key members of the FOT. If the data is reviewed by the FOT only, they need the knowledge to know when there is a problem. If possible, including nominal trending plots as reference for FOT members may help them notice when a problem arises.
- 6.4 Unexpected Trending Results. These should be further analyzed, with potential impact evaluated across the full operations team, and procedures updated to reflect required operations changes to track future conditions relating to the unexpected results.
- 6.5 Ensure That Results are Analyzed. This is necessary for potential incorporation into other operational missions and / or development missions to lower risk and aid in reliability engineering.
- 6.6 Have Different Frequencies of Trending. The capability should exist to generate plots at different frequencies depending on the parameter being trended to allow both short and long term trending. Have plots come out daily, weekly, monthly, quarterly, annual, and/or other rates depending on the frequency at which you would need to see these parameters. For example, you may want to see the solar array current for a time period that is synchronized to the orbit precession rate so that you see a complete cycle. It is common practice that you have the same plots come out at various frequencies so that you can see it at different resolutions and for recognizing short term and long term trends.
- 6.7 Real-Time Plotting Capability. The capability should exist to provide real-time plotting of telemetry for use during Real-time contact with the spacecraft. This tool should also be available for use off-line in replaying old telemetry.
- 6.8 Ability to Plot X-axis Other Than Time. The capability should exist to allow correlation between two parameters as opposed to just time. This allows, for example, correlating a thermal parameter with a portion of the orbit, or a particular spacecraft axis relative to the sun.

## 7.0 Multiple Axis Plotting Capabilities

- 7.1 Multiple Y-axis Plotting Capability. The capability should exist to create plots with multiple y-axes. Up to three is a minimum. This allows you to plot related items on the same plot so that you can see the relationships more easily. For example, battery depth of discharge and battery pressure should track pretty closely for a nickel-hydrogen battery. With them on the same plot, you can see how well they do track and can develop a substitute method of trending should one of the sensors fail.

- 7.2 Multiple X-axis Plotting Capability. The capability should exist for multiple X-axes. This allows an analyst to overlay a previous period with a current period to identify similarities and differences between the sets. This alleviates the need of holding the two pages back-to-back up to a light.
- 7.3 Make Any Parameter Easily Interpretable. The capability should exist to display any parameter that is uplinked and/or downlinked from the spacecraft in a format that is human readable, i.e. converted to engineering unit. This is to avoid the need for bit busting. At a minimum, the raw must be output as well to check the engineering units' conversion, but it is much easier and understandable for the MOT if they can interpret it quickly. This especially includes parameters loaded and dumped as data structures. These types of things are not routinely loaded or dumped, but when they are it is likely a critical time period.
- 7.4 Ad Hoc Plotting Capability. Ensure the system allows the user to plot parameters that are not otherwise routinely plotted. This is useful in anomaly investigation and resolution.
- 7.5 Evaluate Commercially Available Plotting Packages. PvWave has been used at Applied Physics Laboratory (APL), along with DaDisp. Ensure they meet your requirements. APL has also generated its own plotting tools. These were written in C and Visual Basic.

## 8.0 Telemetry

- 8.1 Engineering Telemetry Remote Access. The capability should exist to perform an automated transfer of routine "engineering files" to an unclassified server, where members of the MOT and engineering staff could log-on and download telemetry for their use in off-site (or at their desk) debugging of anomalies or routine performance assessment. Ensure the engineering team defines what data they most likely will need in the "engineering files".
- 8.2 Derived Telemetry Parameters. These are also referred to as "pseudo-telemetry." The capability should exist to combine parameter comparisons in defining a higher, better-defined state. You could conceivably "derive" a Spacecraft Top Level Health parameter, which if all its sub-parameters were considered "green", would indicate that the entire health of the spacecraft is "green". This capability should also exist in the Real-time environment. Whether this capability is accomplished through a rule-based system or an Object Oriented (OO) design depends on the application and those performing the implementation and maintenance. The "OO" approach is easier to maintain.

## 9.0 Alarm Processing

- 9.1 Clear Description of Each Key Parameter. A clear description and the significance of its data readings and trends are required. Limits or "alarms" should be assigned to each key parameter with specific instructions provided for MOT handling of out-of-limit or "alarm" occurrences. This should also be a real-time requirement.
- 9.2 Alarm Dependencies. Require that the "alarm" software allow for dependencies of other parameters being in a specific "mode" or "state". This allows the "alarm" to be more specific to a particular condition. This processing should also occur in the Real-time environment. The ability should exist to change these or add to them as the spacecraft evolves.
- 9.3 Alarm Trigger Count. Require that the "alarm" software allow for a "trigger count" where the condition must exist for a specified number of samples before it will actually signal the alarm. This processing should also occur in the Real-time environment.
- 9.4 Process "Alarms" for Data Recorded On-Board. Require that the "alarm" software used in the Real-time environment to be able to be run on the on-board recorded telemetry after it is downlinked. This allows for alarm checking of telemetry outside station contacts. Try to make it as quick as possible to expedite the process. This process should output a summary report of the alarms encountered during the span of the data analyzed by the alarm processing.
- 9.5 Prioritized Alarm Processing. In cases of simultaneous alarms, ensure all alarms may be easily detected, interpreted, and prioritized. In future Miss Operations Centers where "light-out" operations becomes more of the norm, this functionality will be more critical as the "system" will need to be able to prioritize alarms to determine the appropriate response.

## 10.0 Reports Maintenance

- 10.1 Spacecraft Configuration Change Log. Maintain a database or just a text file of Spacecraft Configuration changes. Include information such as the date and time of uplink or execution of the change. An example of this type of change may include the changing of the Battery Charge/Discharge ratio, or the reaction wheel gains, etc... This allows you to go back and make a correlation between a change and other effects, which may not be noticeable for several weeks. This is also a good thing to include in a summary of Events or a Sponsor Status report.
- 10.2 Maintain a Database of Anomalies. It is highly critical that the system maintains a database of anomaly reports that include a description and the resolution. This saves time in correlating similar problems and leads to quicker resolutions of subsequent occurrences. Create a standard naming convention such that ad hoc queries of similar problems is possible. In the DOD world, the complications of classification should be considered in the maintenance of an anomaly database.

10.3 Relay Information Back to The Spacecraft Manufacturer. Performance information and Lessons Learned from applications should be transferred back to the manufacturer for their information and assistance in improving their products and functions.

10.4 Periodic Assessment Reports. Periodic reports should be generated that discusses the trending analysis and highlight any areas of potential concern. These should be reviewed by a senior member of the technical or system engineering staff, with appropriate feedback provided to the FOT.

10.5 Plots embedded in E-reports. In posting Performance Assessment reports on servers or other electronic media, plots are better than columns of numbers to convey more information. One method is to embed plots in the ASCII text reports as encapsulated post-script. This will require that anyone printing the report need a post-script printer for the plots to come out in a readable fashion; however, it can add significant detail to the report for those who have such a printer. More modern methods include placing the plot on the Web in HTML format. ASCII reports are more desirable from a historical perspective because you will always be able to access this type of format.

---

END

**SATELLITE MISSION OPERATIONS BEST PRACTICES  
BEST PRACTICES WORKING GROUP  
AIAA SPACE OPERATIONS AND SUPPORT TECHNICAL COMMITTEE**

\*\*\*\*\*

**SINGLE SCREEN SATELLITE ALARM LIMIT DISPLAY  
REQUIREMENTS**

**BRIAN WEST  
Bian.west@JNTF.OSD.MIL  
AIR FORCE SPACE AND MISSILE SYSTEMS CENTER**

**July 25, 2000**

1.0 Introduction

COBRA currently displays alarms as a color change in the mnemonic/value pair. This type of alarm display system only displays the fact that a telemetry parameter is out-of-limits and an operator can only view the alarm condition by being on the telemetry screen that displays the telemetry parameter. In order to ensure an alarm condition is observed, the operator must be on the telemetry screen at the time the telemetry point is out-of-limits. Operationally this means an operator must go from screen-to-screen and/or a hierarchical drill down display process must be created.

Another method of displaying alarms is to create a screen, or display process, where all parameters that are out-of-limits are displayed on a single screen. This display process displays several pieces of information about an alarm event and displays several alarm occurrences on a single display. In addition, a single screen is used for all alarm processing for all supports in the Control Center. There are many advantages to this type of alarm display; the operator is notified as an alarm occurs, the operator gets a chronological list of the alarms, several critical pieces of information about an alarm condition are displayed in one place and multiple alarm instances are displayed only once. Another advantage is that the operator can immediately associate, or disassociate, a set of alarms with a particular event. By determining when a set of alarm occurred with respect to the time of an event, it is easy to determine in Real-time whether the alarms are associated with an event.

The single screen concept lends itself towards automation because the operator does not need to be physically up on the support to "monitor" alarms. A process runs continuously that monitors the data flow in the control system (or the data is broadcast and picked up by this process). If an alarm condition occurs it is displayed on the alarm screen (or terminal). The system will display alarms from any pass and from any satellite controlled by the Control Center.

## 2.0 Requirements

2.1 Parameter Display List. The following is a list of parameters that should be available for display on the satellite alarm display.

2.1.1 Parameter Mnemonic and value.

2.1.1.1 Display parameter value at time of first occurrence.

2.1.2 Time of first alarm occurrence.

2.1.2.1 Display time actual alarm occurred.

2.1.3 Number of alarm occurrences.

2.1.3.1 See filtering below.

2.1.4 Value of limit.

2.1.4.1 Display the value of the alarm limit that was exceeded.

2.1.5 Support Identification Information.

2.1.5.1 Display support identification.

2.2 Display and Print Field.

2.2.1 The display and print field should be in landscape layout (or selectable by user).

2.2.2 The display and print field should allow for at least 15 (20 preferable) alarm instances on one screen or one page.

2.2.3 A print function shall allow printing of all alarms in the queue, even alarms not on the screen.

2.2.4 Alarms will be displayed from top to bottom with the most recent alarm on the bottom.

2.2.5 The alarms will be displayed until deleted from the display.

2.2.6 Alarm data will be recorded.

2.2.7 A method should be employed so the user can view all of the alarm conditions even if the display field is full, for example a scrolling screen or another alarm page or window.

2.3 Filtering. It is not operational useful to display each occurrence of an alarm when a telemetry parameter is dithering in and out-of-limits. The preferred method of displaying this information is to display the alarm condition once, and then display the number of times the alarm occurred. Filtering specifies exactly under what conditions a new alarm field is generated for a dithering telemetry value and further enhances displayed alarm information. Specific requirements for filtering follows:

#### 2.3.1 Number of Alarm Occurrences.

2.3.1.1 This parameter displays the number of times a telemetry parameter dithers in- and out-of-limits, subject to the filtering parameters discussed below.

#### 2.3.2 Filter Parameter 1.

2.3.2.1 This parameter, set by the user, is a time limit within which alarm occurrences are filtered.

2.3.2.2 This parameter should have a minimum range of 10 – 10,000 seconds.

#### 2.3.3 Filtering 1.

2.3.3.1 The alarms are filtered by time between occurrences.

2.3.3.2 Any instance of a specific alarm that occurs within the filter parameter increases the “Number of Alarm Occurrences” for that telemetry parameter.

2.3.3.3 Once an alarm occurs outside of the filter parameter, the filter parameter is reset. A subsequent alarm condition would then be displayed separately.

#### 2.3.4 Filter Parameter 2.

2.3.4.1 This parameter, set by the user, is based on the number of decimal counts of the telemetry parameter over or under the limit threshold.

2.3.4.2 The parameter should have a minimum range of 0 – 255 counts.

### 2.3.5 Filtering 2.

2.3.5.1 The alarms are filtered by a value whose level is set by the value of the initial alarm condition and whose range is determined by filter parameter 2.

2.3.5.2 Any instance of a specific alarm condition that occurs within the range of counts set by filter parameter 2 increases the "Number of Alarm Occurrences" for that telemetry parameter.

2.3.5.3 A subsequent alarm condition that occurs outside of the range of counts determined by the first alarm condition is displayed separately. This new alarm condition sets a new level, but not a new range, for filter parameter 2.

2.3.6 Filtering 1 and 2 should be compatible with each other.

### 2.4 Single Screen Requirement.

2.4.1 The single screen concept would probably require a continuously running process. All support alarm data is fed to a single screen (or terminal) that is centrally located in the operations room. The single alarm screen picks up all alarm data, regardless of support or satellite.

---

END