

# **Increases in Efficiency and Enhancements to the Mars Observer Non-stored Commanding Process**

**Robert N. Brooks Jr.**  
**Mission Profile and Sequencing Section .**  
**Mars Observer**  
**Planning and Sequencing Team Chief**

**J. Leigh Torgerson**  
**Spacecraft Systems Engineering Section”**  
**Mars Observer**  
**Deputy Spacecraft Team Chief for Operations**

**Abstract .** The Mars Observer team was, until the untimely loss of the spacecraft on August 21, 1993, performing flight operations with greater efficiency and speed than any previous JPL mission of its size. This level of through-put was made possible by a Mission Operations System which was composed of skilled personnel using sophisticated sequencing and commanding tools.

During cruise flight operations, however, it was realized by the project that this **commanding** level was not going to be sufficient to support the activities planned for mapping operations. The project had committed to providing the science instrument principle investigators with a much higher level of commanding during mapping. Thus, the **project** began taking steps to enhance the capabilities of the flight team. One mechanism used by project management was a tool available from Toml Quality Management (TQM). This tool is known as a Process Action Team (PAT).

The Mars Observer PAT was tasked to increase the capacity of the flight team's non-stored commanding process by fifty percent with no increase in staffing and a minimal increase in risk. The outcome of this effort was to, in fact, increase the capacity by a factor of 2.5 rather than the desired fifty percent and actually reduce risk. The majority of these improvements came from the automation of the existing command process. These results required very few changes to the existing mission operations system. Rather, the PAT was able to take advantage of automation capabilities inherent in the existing system and make changes to the existing flight team procedures.

---

\* Jet Propulsion Laboratory/California Institute of Technology  
4800 Oak Grove Dr.  
Pasadena, CA 91109  
The work described in this paper was carried out at the Jet Propulsion Laboratory/California Institute of Technology under contract with the National Aeronautics and Space Administration.

This paper will describe in detail the enhancements recommended by the PAT for the non-stored command generation process on Mars Observer. This will be contrasted with the process used by the flight team prior to implementation of these improvements. Finally, there will be a discussion of the applicability of the techniques devised by the PAT for enhancement of the non-stored command process to present and future projects.

## INTRODUCTION

The Mars Observer project had as its goal the complete mapping of the Martian surface in several spectral regions. Some areas were to be mapped in extremely high resolution. This was going to be accomplished by following a flight and operations strategy which used the following design principles.

- The spacecraft would be a relatively simple device which would act as an orbiting platform from which to perform remote sensing of the planet's surface and atmosphere.
- The spacecraft would be placed in a low altitude (378 km), near circular, near polar orbit.
- The science instruments would be Nadir pointed with the remote sensing science instruments mounted on a rigid platform.
- Any and all instrument articulation would have to be performed internal to the instrument and be of a non-interactive, non-interfering nature.
- All control of the instruments was to be managed and commanded by the remotely located science instrument teams. The JPL flight team was to be a "poll" through which commands moved, but were not interfered with.
- The flight team staffing was only normal working hours.

These six basic design principles were intended to reduce complexity of operations, increase the autonomy of the Principle Investigators over their instruments and, ultimately, reduce costs by reducing flight team workload and staffing requirements. Unfortunately, a multitude of factors influenced the designers of the operations processes and true autonomy was not attained at the time of launch in 1992. Though the thrust of this discussion is not to elaborate on these factors, it should be sufficient to point out that, at the time of launch, all were legitimate concerns and, therefore, causes for conservatism on the part of the operations designers.

However, after launch it was discovered that many of the aforementioned concerns were no longer problematic. Steps had been taken by various parties to mitigate the problems and a less conservative approach was deemed appropriate. In addition, it became abundantly clear to management, the science teams and the operations team that the level of science commanding necessary to accomplish mission goals was not going to be possible given the conservative operations techniques used by the flight team. A totally new approach would be necessary to satisfy these needs.

The tool which project management decided to use for accomplishing this goal was a standard tool available from Total Quality Management (TQM). This tool is called a Process Action Team (PAT). The PAT assembled by the project manager was charged with determining the best method for increasing efficiency and through-put of the processing of Non-interactive Non-stored Commands (NINSC). This paper will discuss the concept of a PAT,

describe the original NINSC process as it existed at launch and the streamlined NINSC commanding process which resulted from the deliberations of the PAT. Finally, a brief discussion of the application of these operations strategies to future projects will be given.

#### ORIGINAL NON-INTERACTIVE NON-STORED COMMAND PROCESS

The Mars Observer spacecraft design allowed for command execution immediately upon receipt or for the storage of a series of time-tagged commands that would autonomously execute at the appropriate time. These stored commands were referred to as "sequences," and the spacecraft was capable of simultaneous execution of several stored sequences.

As the Mars Observer spacecraft normally flew with one or more stored sequences on board and executing, non-stored commands were scrutinized carefully to assess the possibility of adverse interaction with current sequences, spacecraft configuration or power and thermal conditions.

The spacecraft was specifically designed to minimize the interaction of the science instruments with the power, thermal or dynamic states of the spacecraft bus. A small number of payload commands could cause the power consumption of the payload suite to significantly increase and these were deemed "Interactive" commands. The majority of the payload commands were "Non-Interactive," and the design intent was to allow the science instrument operators maximum freedom to send non-interactive commands to their instruments in real-time without submitting command requests for scrutiny by the flight team, as was necessary in the case of interactive payload commands.

These were termed "Non-Interactive Non-Stored Commands," or NINSC's.

A basic innovative concept behind the Mars Observer operations strategy was that the science teams were located at their home facilities, with command requests and science instrument data communicated electronically through computer networks. A central Project Data Base (PDB) was established at the JPL facilities in Pasadena, with appropriate security measures in place. Each science team had electronic access to current spacecraft health and status data, science data downlinked from the spacecraft, and a repository for placing files that contained NINSCs they wished sent to their instruments. Each science team had their own secure database "bin" for command requests and science data.

There were two parts to an instrument command. Part one was the binary file or files containing the actual commands to be sent to the spacecraft, and part two was the command request which detailed the purpose of the commands, the desired time of transmission, or, if several files needed to be sent in a specific order at certain times, a radiation plan for the Mission Control Team (MCT) to follow. The science team would put these items in the PDB, and notify the Experiment Representative at JPL via E-Mail that a command request was ready for processing.

Processing these requests involved the steps summarized in figure 1. The command file containing the commands for the science instrument to execute had to be

- a. Checked for valid instrument ID and opcodes.
- b. Merged with spacecraft commands which would pass the

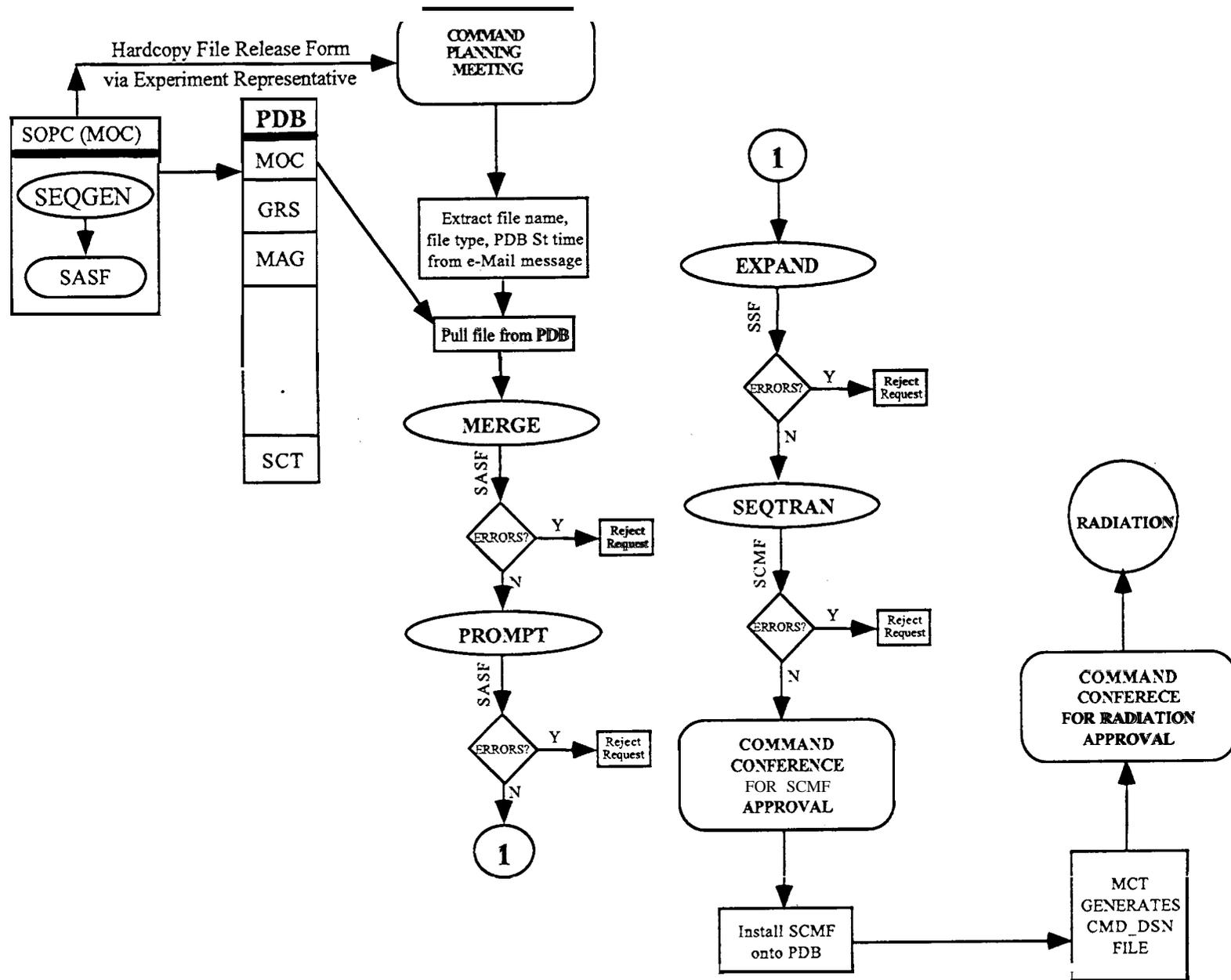


Figure 1: Original NINSC Process

payload commands through to the appropriate instrument.

- c. "Wrapped" with a header which provides information to the DSN about which spacecraft to send the command to and at what time.
- d. Converted to the actual binary file to be sent to the DSN for radiation.

Each of these steps were conducted by different people and several separate pieces of software were required to generate the intermediate files and reports. To limit development costs, much of the software used was taken from other projects and modified to suit the needs of Mars Observer, resulting in a multi-stage process.

With each of these steps there was much paperwork generated, manual Quality Assurance (QA) operations to insure that errors were caught and management scrutiny to see that the commands were indeed non-interactive. In parallel with this process, a series of meetings were conducted to sign off the QA process, coordinate with the Mission Control Team (MCT) on when the commands were to be sent, and to apprise the flight team of the intended command activity.

This process embodied the conservatism necessary to avoid problems which might be brought on by inappropriate commanding, and served the project well for the first few months of Mars Observer flight operations. It was, however, far from the "real-time" commanding expected by the science community, and the process promised a significant workload during mapping, where as many as six NINSC requests per day were expected. Extrapolation to the mapping scenario showed that the original NINSC

process would have taken 34 work-hours per day and produced 120 items of paperwork per day.

## PROCESS ACTION TEAMS

The basic concept behind a Process Action Team (PAT) is that the owner of some process assembles a group of people familiar with the process to study it in detail and then to recommend ways to achieve a set of specific objectives and measurable goals with respect to that process. The PAT uses a formal methodology, and has both a schedule to adhere to and a set of deliverables. A facilitator from outside the project is brought in to aid in objectivity, and a Quality Council panel of senior managers (some from outside the project) periodically reviews the work of the PAT.

The Mars Observer (MO) Uplink PAT was established by formal charter by the project manager, and had the task of reevaluating the uplink process and to establish revised procedures to fulfill several objectives, including:

- . Improved responsiveness to science command requirements
- . Increased command volume without risk
- . Streamlining of the entire uplink process.

These improvements were to be made without any increase in command-processing workforce, and as a goal, the resulting process was to provide at least a 50% increase in command generation capacity by the existing workforce.

The PAT was to deliver a defined set of products which included revised project policies, procedures, forms, interface

agreements and any other documentation necessary to describe and control the revised uplink process.

The activities of a PAT are conducted in a structured, 4-part methodology described by the acronym "FADE", which stands for "Focus", "Analyze", "Develop" and "Execute".

The Focus phase is to decide on exactly what the problem is, and to narrow the focus of the team's work so as to avoid attempts to either solve too much or solve the wrong problem. The result of the Focus phase was a Problem Statement which described the current state of the uplink process, the impact to the customer, and the desired state. The MO Uplink PAT focused on the NINSC process.

At the completion of each phase, the Quality Council reviews and approves the work of the PAT before the commencement of the next phase. This is to avoid the possibility of designing a solution to a problem which, in the eyes of management, may not exist.

The Analyze phase is designed to investigate and quantify the process to shed light on just where the problem areas are. The phase involves deciding what data are necessary, collecting these data to baseline and identify trends, and to finally determine which factors are the most influential. The MO Uplink PAT studied the NINSC process, and did a detailed accounting of the time and energy required to complete each step of the process and determined what "value-added" there was for each step or process output.

During the Development phase, the improvements to the process are developed. These improvements include not only a new process to implement, but also an

implementation plan to smoothly transition from the old process to the new. The MO Uplink PAT found paperwork and reports generated which had no "customers", found several areas where inexpensive automation could replace manual checks, and identified new command categories which would allow achievement of science objectives without increasing either risk or team size.

The final phase is to Execute the solutions defined in the Development phase. The first step is to obtain management and team support for the solutions - a task made infinitely simpler by the objective data and thorough methodology of the preceding three steps. Next is to implement the new process, and to monitor its effectiveness using the same metrics and methods used in the Analyze phase. In the case of the MO Uplink PAT, management and team acceptance of the new process was obtained, some of the new procedures were implemented and monitored, but the unfortunate loss of the spacecraft prior to mapping precluded a full evaluation of the new process.

The following section details the new NINSC process recommended by the MO Uplink PAT.

## DESCRIPTION OF RESULTS

The final outcome resulting from the deliberations of the MO Non-interactive Non-stored Commanding Process PAT was a set of recommendations which would increase the through-put for Non-interactive Non-stored Commands from the current one hour or more per command file to a maximum of fifteen minutes per file. This increase in efficiency was to be accomplished by altering the existing process in three specific ways.

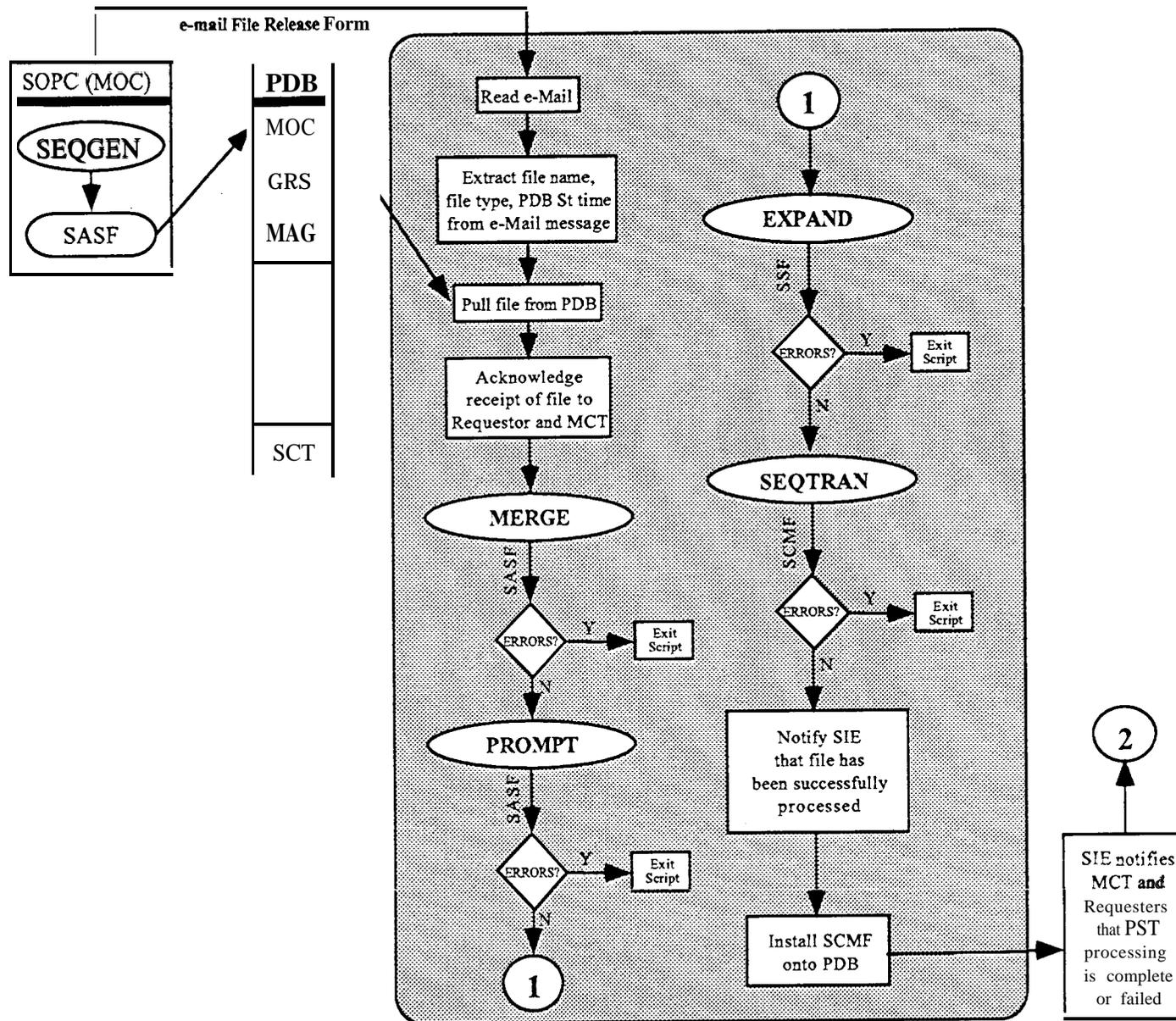


Figure 2: Planning and Sequencing Team (PST) Script

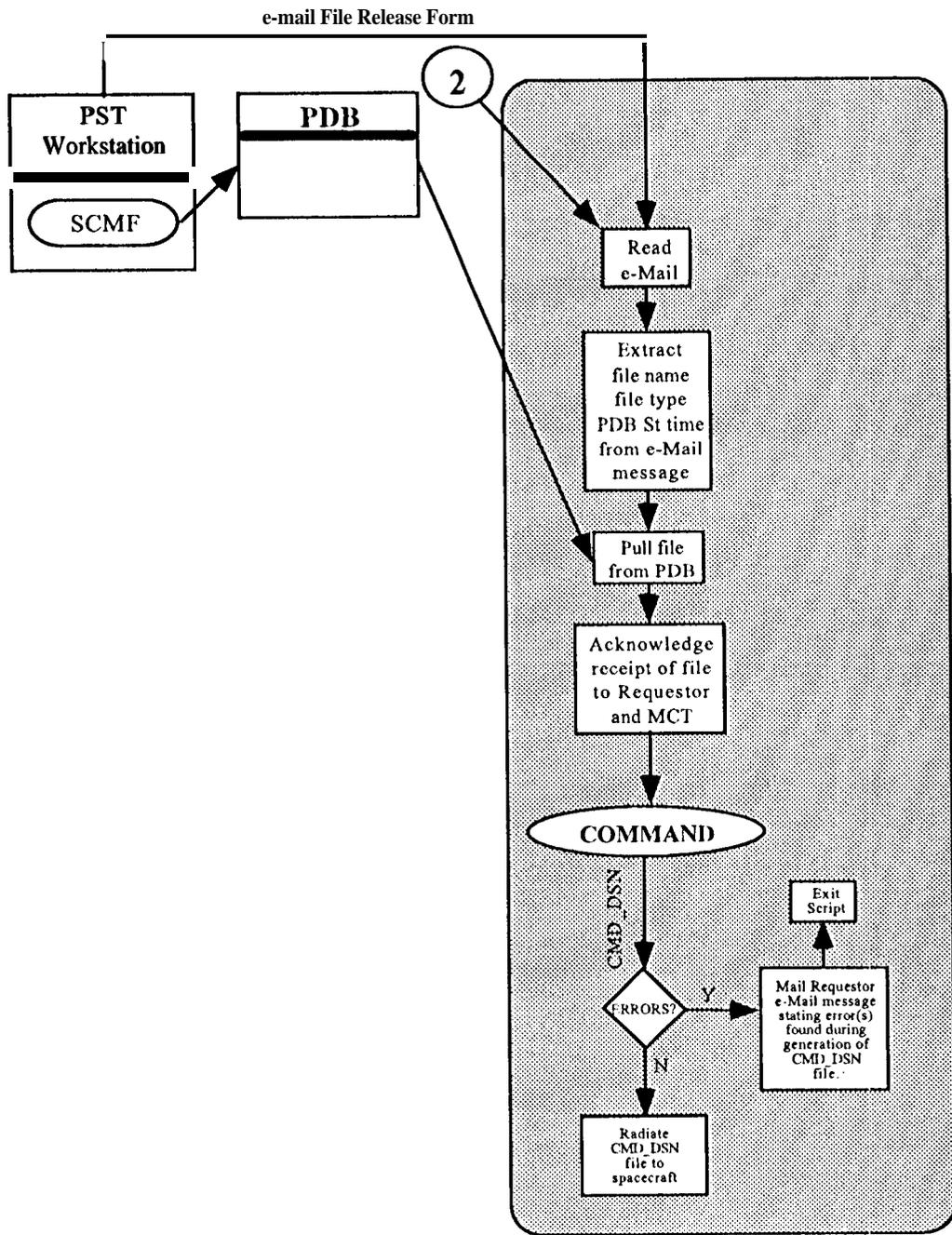


Figure 3: Mission Control Team (MCT) Script

The first problem identified by the PAT as hindering the processing of NINSCs was excessive management scrutiny of the command requests. This scrutiny was felt to be necessary to prevent erroneous commands from being sent to the science instruments. The elements of the command request which were scrutinized included purpose of the requested commands and correctness of the data contained in the request. After some study, the PAT found that such intense scrutiny was totally unnecessary. This was based on the fact that the spacecraft and science instruments had been built so that such commands could not compromise spacecraft health or safety. Furthermore, much of the syntactical checking was already being performed by the ground software system and, therefore, did not need repeating by management. The PAT therefore recommended that all such scrutiny of NINSCs be stopped.

Another problem which was identified by the PAT was excessive amounts of paperwork associated with this type of commanding. Every command request processed required between ten and twenty pages of paper, depending upon the number of commands in the original request. Completion of this paperwork became an intense burden on the flight team. The PAT recommended that NINSCs be exempt from the large amounts of paperwork associated with other types of commanding.

This leads to the third change recommended by the PAT. At the time of launch all NINSCs had been classified together as one large group.

Flight team and management procedures treated all of these commands with equal conservatism and caution. However, as the flight team gained more experience flying the spacecraft, they found that approximately 85% of these commands were genuinely non-interactive in the truest sense of the word.

These commands required no spacecraft resources or significant ground resources. This led the PAT to recommend that a new class of NINSCs be defined which required no coordination beyond any incorporated within the file as it was submitted by the requester. Their processing was to be heavily automated and very rapid. This new class of commands would be referred to as **Express** commands.

The automation of the Express NINSC process was fundamental to the successful increase in efficiency. This automation would be accomplished by using two scripts written in UNIX, PERL and awk. These scripts were divided along team functional lines. The Planning and Sequencing Team (PST) used a script which would execute all necessary and appropriate software, automatically checking each file for errors as it was processed. After each file had completed its PST processing, it would be retrieved by the Mission Control Team (MCT) using their script and processed into a CMD-DSN file for radiation to the spacecraft. What follows is a detailed description of the Express NINSC process as implemented on Mars Observer.

#### DETAILED DESCRIPTION OF FINAL IMPLEMENTATION

The EXPRESS NINSC command process would begin with each requester who required commanding installing their request Spacecraft Activity Sequence File (SASF) onto the PDB in the appropriate PDB bin. At the same time that the requester installed their SASF(s) onto the PDB, they would send an e-Mail "File Release Form" (FRF) to both the PST and the MCT. These two tasks were to be completed by 10:00 am Pacific time for the file(s) to be considered for same day processing.

Flight team processing of Express NINSCs required very minimal human interaction (at only the beginning and end points of the scripts). This interaction was of a process management and instigation nature. Actual file processing, execution of sequencing software and error checking were performed internally by the script. Figures 2 and 3 are graphical representations of the Express NINSC process.

Beginning at 10:00 am Pacific time every weekday, the PST would instigate execution of the EXPRESS NINSC script. This instigation would be authorized by the Sequence Integration Engineer (SIE) and actual script execution initiated by the Software Operations Engineer (SWOE). Each file would be processed by the script, one file at a time in the order that the e-Mail file release forms were received by the PST, until processing was complete.

The script would begin by reading the e-Mail FRF submitted by the requester. This FRF adhered to a specific format and contained data necessary to verify file origin and location. The script extracted from the FRF all of the above described data. The script used these data to extract the SASF from the PDB and install this SASF onto the PST workstation being used to process NINSCs. The script then sent an e-Mail acknowledgment of receipt of the SASF to the requester and the MCT. This acknowledgment allowed these two groups to track the status of those files being processed.

The script executed the MERGE software. This software correlated requesting group and destination instrument. The latter was accomplished by comparing the file type provided in the FRF with the instrument OPCODE provided in the SASF.

The script would then execute a general purpose error detection program. This piece of

software used other program's runlogs as input to check the success of those runs. In this case, it used the MERGE program's runlog as input.

As is obvious from figure 2, during execution of other parts of the script other program's runlogs would be used as input for this program. Any errors detected during execution of this software caused immediate exit from the script and a failure message, containing file name and failure details, to be sent by e-Mail to the SIE. The SIE then determined which was the best resolution of the error. At the discretion of the SIE, this may have included rejection of the file or contacting the requester to help in correction of the error. In any case, an erroneous file was not guaranteed same day readiness for transmission to the spacecraft,

This was followed by the script executing the PROMPT software, which would verify syntax, data field value limits and SASF format, the EXPAND software, which converted the SASF into a Stored Sequence File (SSF). The SSF can be thought of as the "source code" for the commands requested in the SASF. This SSF was used as input to the SEQTRAN software in the next step and finally the script would execute the SEQTRAN software. This software converted the SSF generated by EXPAND in the previous step into an Spacecraft Message File (SCMF, the actual binary representation of the data in the original SASF).

Upon successful completion of all preceding steps in this script, the script would notify the SIE that the file had completed processing and would automatically write the SCMF for the file to the PDB.

The final step of PST processing was the responsibility of the SIE (not the script). This was the notification of the requester and the MCT by e-Mail that the file completed

processing and was available on the PDB. This e-Mail message contained a PST FRF. This FRF was formatted in a specific way and contained information needed by the MCT to begin their processing.

The PST would repeat the above steps for each file for which an FRF was received, until all files submitted for that day had been processed,

Immediately upon receipt of the PST e-Mail File Release Form (FRF), the MCT would initiate its script to process SCMFs into CMD\_DSN files (the files which is formatted to be transmitted through the Deep Space Network). The first step in this script was to retrieve the e-Mail FRF and extract the SCMF file name and other pertinent data. The script would use the information provided by the PST FRF to extract the appropriate file from the PDB. The script would then verify the file's authenticity. The script then executed the uplink window computation software to determine the available uplink windows for the file being processed.

After determining all available uplink windows in the preceding step, the script would execute the COMMAND software, which converted an SCMF into a CMD\_DSN file. Though an SCMF does contain the actual bits to be loaded onto the spacecraft, it is not properly formatted so that it can be radiated through the Deep Space Network (DSN). The COMMAND software formats each SCMF and produces a CMD\_DSN file.

As was the case with the PST script, the MCT script checked the COMMAND runlog for errors encountered during execution. Any errors detected in the runlog would cause immediate termination of the script and a failure message, containing file name and failure details, to be sent by e-Mail to the MCT member responsible for running the script. The

MCT member would then determine which was the best resolution of the error. At the discretion of this MCT member, this may include rejection of the file or contacting the PST or requester to help in correction of the error. In any case, an erroneous file was not guaranteed same day readiness for transmission to the spacecraft. If no errors were found during the above check, then the MCT script would queue the CMD\_DSN for radiation to the spacecraft at the time determined by the uplink window computation software above.

Upon successful completion of all preceding steps in this script, it would notify the responsible MCT member that the file had completed processing and would automatically write the CMD\_DSN to the PDB for archival purposes.

The final step of MCT processing would be carried out by the responsible MCT member (not the script). This would be the notification of the requester by e-Mail that the file completed processing and was queued for radiation. This e-Mail message contained an MCT FRF. This FRF was formatted in a specific way and contained information which unambiguously identified the CMD\_DSN file. The MCT repeated the above steps for each file for which an FRF was received from the PST, until all files submitted for that day had been processed.

#### APPLICATION OF RESULTS TO FUTURE FLIGHT OPERATIONS

The results of the Uplink Process Action Team promised broad application to other non-stored processes used by Mars Observer as well as to other JPL flight projects, both current and future. In fact, experience from Mars Observer indicates that risk is actually reduced when these types of commands are not scrutinized but rather the process by which they are

generated is scrutinized and verified and then is automated in such a manner as to prevent circumvention unless approval is given.

In general, present missions can benefit from these results by scrutinizing and analyzing their processes and identifying all unnecessary (little or no value added) 'human interaction' steps. These steps should then be eliminated if possible or automated when still needed. Prime candidates for this type of automation would include checking of printouts for errors and 'checking' of paper forms for errors. The latter of these two items represented an enormous amount of time spent by managers on MO which slowed down the process. Few if any errors of these types were ever encountered for the NINSCs processed.

Future missions can benefit from this effort by accepting the precept that rigorous analysis of processes and automation of these processes leads to increased efficiency and, hence, either increased productivity or decreased staffing levels. Mitigation of risk is accomplished by scrutinizing and validating the automation tools before they are used in operations. In the case of Mars Observer, the tools in question had been used in actual flight operations for several months and had been well validated. In addition, the team procedures used to define the NINSC process had been well practiced and, when necessary, modified or corrected to eliminate error sources. Finally, the tools used in this processing had been developed in a 'modular' sense and to allow command line control of all software elements. These two characteristics of the software permitted the operations teams to modularize their procedures and break them down into easily understood and automated functions.

Planetary Exploration Through Year 2000, A Core Program: Mission Operations, A Report by the Solar System Exploration Committee of the NASA Advisory Council, U. S. Government Printing Office, 1986.

Varghese, Philip (Team Lead), et al, Final Report of the Mars Observer Uplink Process Action Team, Mars Observer Project Document, 19 November 1993

Quality Action Teams, Workbook for Team Members, Organizational Dynamics Incorporated, 1990.

## REFERENCES