N94-23865

MAGELLAN SPACECRAFT AND MEMORY STATE TRACKING; LESSONS LEARNED, FUTURE THOUGHTS

Allen W. Bucher

Martin Marietta Astronautics Group, Denver, Colorado 80201, U.S.A.

ABSTRACT

Numerous studies have been dedicated to improving the two main elements of Spacecraft Mission Operations: Command and Telemetry. As a result, not much attention has been given to other tasks that can become tedious, repetitive, and error prone. One such task is Spacecraft and Memory State Tracking, the process by which the status of critical spacecraft components, parameters, and the contents of on-board memory are managed on the ground to maintain knowledge of spacecraft and memory states for future testing, anomaly investigation, and on-board memory reconstruction.

The task of Spacecraft and Memory State Tracking has traditionally been a manual task allocated to Mission Operations Procedures. During nominal Mission Operations this job is tedious and error prone. Because the task is not complex and can be accomplished manually, the worth of a sophisticated software tool is often questioned. However, in the event of an anomaly which alters spacecraft components autonomously or a memory anomaly such as a corrupt memory or flight software error, an accurate ground image that can be reconstructed quickly is a priceless commodity. This study explores the process of Spacecraft and Memory State Tracking used by the Magellan Spacecraft Team, highlighting its strengths as well as identifying lessons learned during the primary and extended missions, two memory anomalies, and other hardships encountered due to incomplete knowledge of spacecraft states. Ideas for future state tracking tools that require minimal user interaction and are integrated into the Ground Data System will also be discussed.

Keywords: mission operations, state tracking, memory tracking, Magellan.

1. THE MAGELLAN SPACECRAFT

The Magellan Spacecraft is a National Aeronautics and Space Administration's (NASA) planetary mission designed to obtain high-resolution Synthetic Aperture Radar (SAR) images of Venus. Magellan was launched aboard the Space Shuttle Atlantis on May 4, 1989 and arrived at Venus on August 10, 1990. Magellan began SAR imaging the Venusian surface on September 15, 1990 and to date has produced high-resolution imagery of over 99% of the planet's surface.

The Magellan Spacecraft was designed using many spare parts primarily from the Voyager and Galileo spacecraft. The basic design of the craft includes a gyroscope/three-axis stabilized Attitude Control subsystem, a Ni-Cd battery and solar array Power subsystem, a mono-propellant Propulsion subsystem, an S and X Band Telecommunications subsystem, a SAR and Altimeter, and on-board electronics and computers for controlling the craft and peripherals.

The spacecraft contains two main subsystems for controlling its operations, the **Command and Data Subsystem (CDS)** and the **Attitude and Articulation Control Subsystem (AACS)**. The following paragraphs contain a brief description of the CDS and AACS subsystems. These descriptions provide the reader with information regarding the on-board computer systems integral to the subsequent state tracking discussions.

1.1. Command and Data Subsystem

The CDS computer system is the distributed data system of the spacecraft. This system controls the operations of Magellan by providing a central interface to all components, Figure 1. The tasks performed by the CDS include interpretation of uplink messages, execution of all spacecraft command sequences, fault protection monitoring, and collection and formatting of engineering and science data for transmission to Earth.

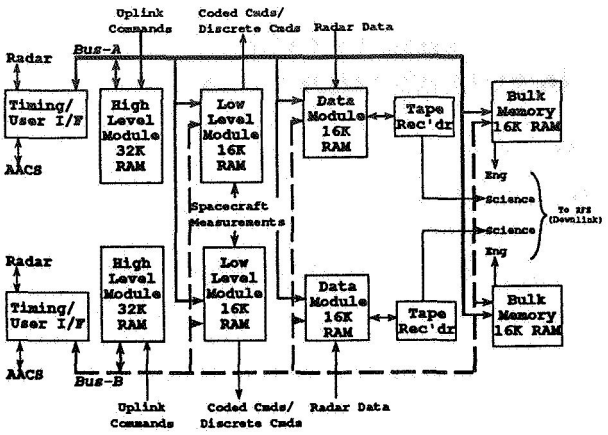

Figure 1 CDS Functional Block Diagram

The CDS computer system is composed of two computer strings which operate in parallel. Only one string is required to be operational for Magellan to function. Thus, the strings are redundant to provide fault tolerance. Each CDS computer system contains two 8 Bit RCA 1802 microprocessors and 80 kilobytes (80K bytes) of active Random Access Memory (RAM) and 80K bytes of redundant RAM for a total of 160K bytes. Approximately 48K bytes of active RAM is used for flight software executable code and parameters. (The CDS flight software contains approximately 600 commandable parameters for software performance customization.) The remaining 32K bytes are used for Command Sequence/Data Storage. The redundant RAM is not normally used to supplement active RAM and the CDS contains no Read Only Memory (ROM).

1.2. Attitude and Articulation Control Subsystem

The AACS is a computer controlled subsystem that performs the tasks required to maintain Attitude and Articulation Control. Tasks performed by the AACS include desired reference generation, attitude determination, momentum management, sequence and scheduling, and fault protection.

The AACS is an assembly of electronic components as shown in Figure 2. The AACS subsystem is a block redundant system with cross-strapping capability between the major components including the sensors, reaction wheels, solar array drive assemblies, and propulsion devices to provide fault tolerance. Each AACS computer contains one 16 bit Applied Technology Advanced Computer (ATAC) microprocessor with 32K words (64K bytes) of RAM. All 32K words of RAM are used for flight software executable code and parameters. The AACS flight software contains approximately 3000 commandable parameters for software performance customization. Each memory also contains 1K of Read Only Memory (ROM).

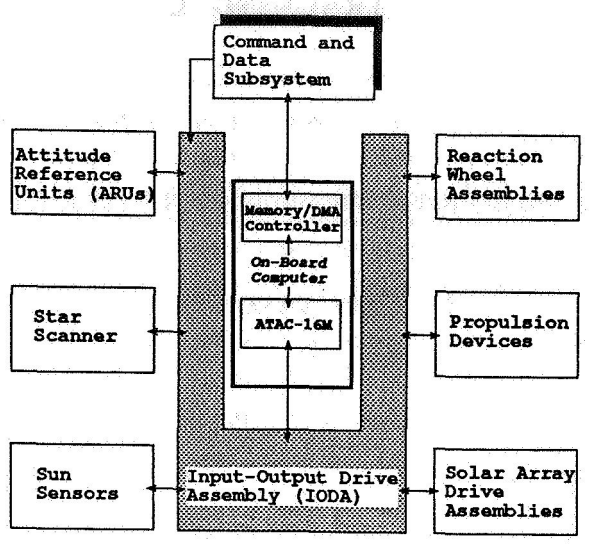

Figure 2 AACS Functional Block Diagram

2. STATE TRACKING OVERVIEW

The two types of state tracking being emphasized in this study are the tracking of critical spacecraft hardware or software states (*Spacecraft State Tracking*) and the tracking of on-board computer memory states (*Memory State Tracking*).

2.1. Spacecraft State Tracking Overview

Spacecraft State Tracking is typically thought of as tracking critical spacecraft components to ensure that the configuration of the spacecraft is known at all times. The spacecraft states that are of interest are those states that may not be readily available via spacecraft telemetry, states that are critical to long term sequence and mission planning, and states that are utilized by the spacecraft fault protection system. For example, the different cross-strapping configurations of AACS hardware are not always directly derivable from spacecraft telemetry due to the usage of switchable primary and secondary hardware relays. These configurations must be derived from the last known state and the state altering commands issued either from the ground or by spacecraft fault protection. The status of critical components must be carefully tracked to assure compatibility with the current sequence of spacecraft activity and the current configuration of the spacecraft fault protection system.

During the life of a given mission, the spacecraft is likely to encounter a variety of mission phases requiring the spacecraft configuration to be altered to meet the goals of that phase. The sequence of events for each phase is planned weeks or months prior to actual implementation on the spacecraft. Mission planners need to plan and create a sequence of events for the spacecraft to perform in the future. Knowing the configuration of the spacecraft in advance presents two unique problems; 1) the current mission phase may have the spacecraft configured differently than future mission phases require, and 2) hardware and/or software failure resulting in fault protection intervention may alter the state of the spacecraft prior to the implementation of future sequences. Knowing the current spacecraft state and the ability to determine the differences between the current state and the state required for future sequences is the goal of effective spacecraft state tracking.

2.2. Memory State Tracking Overview

Memory State Tracking is the process by which the on-board computer memory contents are managed on the ground to maintain an accurate representation of the content of on-board computer memory. Memory is usually partitioned into segments representing flight software code, flight software parameters, and command sequence storage.

The objective of effective memory state tracking is to maintain knowledge of the content of on-board computer memory at any given time. The tracking methods need to supply an effective way to track and report changes, archive historical memory states, and provide interfaces to spacecraft commanding and test facilities. In contrast to spacecraft state tracking, memory state tracking deals only with the content of the on-board computer memory.

3. MAGELLAN STATE TRACKING

3.1. Magellan Spacecraft State Tracking

Spacecraft State Tracking for Magellan is primarily a manual process. Many sequence constraints and spacecraft state configurations are documented in the Spacecraft Block Dictionary (Ref. 1). The Block Dictionary contains the requirements for all spacecraft sequence "blocks". A sequence block is designed to "define the necessary block sequences of commands that will result in execution of spacecraft subsystem events to satisfy required spacecraft operations" (Ref. 1). The group of commands required to desaturate the spacecraft reaction wheels is an example of a sequence block. The Block Dictionary also contains a reference state table used for determining block initial and final conditions including fault protection system configurations. Although the Block Dictionary contains sequence requirements and a reference state table, the implementation of the required spacecraft state configuration is accomplished manually by the spacecraft team. The current spacecraft state is manually determined from telemetry, the state for future sequences is determined manually, and the system fault protection is manually reviewed to insure compatibility with the desired spacecraft state. These manual processes rely heavily on human expertise and discipline to verify all states and configurations. This process worked well while the spacecraft team was at the maximum staffing levels and while the "experts" remained part of the team. Hardships are encountered however, as the spacecraft team size continues to decrease and the experts leave the team.

Another difficult task during major mission phase changes is the preparation of future states for the System Verification Laboratory (SVL). SVL is a high fidelity software and hardware simulator of the Magellan spacecraft designed for flight software testing, sequence verification, and anomaly resolution testing. The SVL has to be cognizant of the actual spacecraft state changes to insure accurate spacecraft state representation and must be properly configured with future spacecraft states to accommodate future sequence testing. When the spacecraft team does not perform all required state changes, the SVL is able to determine the majority of the sequence versus spacecraft configuration mismatches. The SVL is not able to catch all inconsistent states because not all spacecraft components are modeled. With most of the emphasis focused on creating and tracking valid states in SVL to insure valid sequence testing, the SVL has become the default implementation of a spacecraft state tracking and propagation tool for Magellan. Although this method works, the SVL still has to manually implement changes to maintain compatibility with the current spacecraft state. Also, there is not an easy method for the spacecraft team analysts to retrieve information from SVL regarding past or future spacecraft states.

All critical component state tracking was also accomplished by engineers from each subsystem manually recording their individual subsystem states in a spacecraft state table. However, soon after launch, the spacecraft team developed the ability to autonomously track all spacecraft states that are telemetered. The spacecraft team created a Unix script entitled "Moose Query" which queries all

Magellan telemetry from the central database and produces a summary report of all states that are derivable from telemetry. Although this report is still manually compared to a table of expected states, it greatly simplifies collection and recording of the approximately 1000 spacecraft states. In addition to the Moose Query script, the spacecraft team also uses the Data Monitor and Display capability to dump the latest available data (LAD) buffer for spacecraft state tracking. This LAD dump capability is a useful analysis tool when it is not necessary to determine the exact time the state changed but to have a snapshot of the spacecraft states at a specified time. The Moose Query and LAD dump reports are archived to provide a permanent record of the spacecraft state.

All state changes that are accomplished as part of the nominal mission are carefully planned and reflected in the sequence development cycles. Many hardships result from last minute changes to sequence plans due to fine tuning of mission requirements or from unanticipated spacecraft configuration changes. For example, the spacecraft sequence designed to perform reconditioning of the Magellan batteries had been planned and constructed assuming use of transmitter A. However, due to degrading performance of transmitter A, the spacecraft team was forced to switch to transmitter B prior to activation of the sequence. This switch was not reflected in the nominal sequence planning and generation process. The sequence was radiated to the spacecraft and when execution commenced, the spacecraft fault protection, enabled by the sequence, detected that transmitter A was not producing an output signal (because it had been previously commanded off). The fault protection software, assuming failure of transmitter A, issued commands to autonomously switch to transmitter B. The fault protection activation resulted in several telemetry alarms and some short-lived excitement. After analysis it was determined that the commands issued by the fault protection software had no effect because the transmitter A to B swap had already been accomplished by ground command.

This is one example where the spacecraft team was forced to react in an emergency mode because the sequence did not match the current state configuration of the spacecraft. Fortunately the result of the configuration mismatch was benign. The incident did however raise some questions regarding the spacecraft team's ability to manually guarantee that the configuration of the spacecraft is consistent with the current spacecraft sequence and fault protection configuration.

3.2. Magellan Memory State Tracking

Memory State Tracking for Magellan is accomplished by a suite of software tools designed to perform the tasks of flight software loading, flight software parameter tracking, and memory compare and analysis. The software tools are broken into three computer programs designed to carry out the required tasks. These programs are defined as:

1) *Flight Software Loading* - **VMFLOAD** is designed to interface flight software loads from either the flight software development environment, the spacecraft test facilities, or the mission operations environment to the command system. VMFLOAD accepts hexadecimal flight load images and reformats them into uplink messages for processing by the sequence translation software and subsequent radiation to the spacecraft. These "uplinkable" images are also compatible with the SVL to allow testing prior to actual command radiation.

2) *Flight Software Parameter Tracking* - **VMPTRK** is designed to track commanded changes to flight software parameters. VMPTRK reads the Spacecraft Event File (SEF) and extracts commanded changes to tracked parameters and records the new parameter value, the time of the change, and the identification of the SEF that altered the parameter in the VMPTRK Tracked Parameters Database. The Tracked Parameters Database contains a historical archive of all parameter values and the information regarding each update.

3) *Memory Compare and Analysis* - **MEMANAL** is a program designed to read a file representing a computer memory image and file of spacecraft Memory ReadOut (MRO) and identify any miscompares between the ground memory image and the MRO data. Spacecraft MRO data is generated by issuing a spacecraft command designed specifically to "Read Out" a selected area of memory and place the result in the spacecraft telemetry stream. MEMANAL has the ability to compare and analyze any selected portion from either computer system.

After the Magellan flight software and associated parameters were loaded and the spacecraft was configured for launch, all on-board computer memories were read-out and verified against the final flight software builds using MEMANAL. The results of these verifications were baselined as the first ground memory images. The contents of the first VMPTRK Tracked Parameter Database was also verified. These initial configurations were important as all future configurations are derived as deltas to the initial baselines.

The process of memory state tracking was accomplished by tracking commanded changes to parameters via VMPTRK, tracking changes to flight software loads via VMFLOAD, and verifying changes via MEMANAL. Periodically the on-board computer memories are "read-out". The resultant MRO is compared to the latest ground memory image(s) and any miscompares are verified. Once the new MRO is verified, it is archived and placed under configuration management as the current state of the on-board memory. This method provides for periodic recording and validation of the memory state. As an independent check the tracked parameter database is also compared to the MRO. This accomplishes two objectives: It verifies that the content of the on-board parameters is as expected, and it verifies that the content of the tracked parameter database is correct and can be trusted as an accurate recording of critical flight software parameters.

To provide examples of the use of these tools, resolution of two memory anomalies on the Magellan spacecraft will be discussed. The following paragraphs give a brief description of each memory anomaly and the actions taken by the Magellan spacecraft team to resume normal spacecraft operations.

3.2.1. CDS Memory Read Parity Error

On December 31, 1989 the CDS processor detected a memory read parity error indicating that the parity reading from the memory was different from what the processor expected. This is considered a privileged error by the CDS resulting in the spacecraft placing itself in a safe state. To recover from the safe state, the spacecraft team decided to utilize the redundant RAM associated with the suspect CDS processor. The redundant RAM however, was still configured with the launch flight software and parameters and would have to be reconfigured to the current memory state before it could be used. The first step was to read-out the primary RAM to determine the up-to-date configuration and read-out the redundant RAM to determine its exact content. The two memory read-out images were compared, using MEMANAL, to determine the memory image configuration differences. The differences determined the changes required to restore the state configuration of the redundant RAM. Once the required changes were determined, commands were built to update each individual parameter that was out-of-date. Each command was compared with the configuration managed memory image and the VMPTRK tracked parameter database to insure correctness. After the commands were verified, they were tested in the SVL and then radiated to the spacecraft. After successful radiation the memory image was again read-out and verified, using MEMANAL, before the processor was returned to operation using redundant RAM.

3.2.2. AACS Memory Address Failure

Following Venus Orbit Insertion the off-line AACS computer experienced an intermittent memory address bit failure resulting in an area of off-line RAM that was not accessible by the CPU. Realizing that this configuration was not desirable, the spacecraft team configured the AACS on-board computers to indicate that the off-line memory and CPU were not usable while they established the probable failure mode and solution. Also as a precautionary measure, the entire off-line memory was "back-filled" with a alternating pattern of all 0000's followed by all FFFF's at 16 word boundaries. The 0000's represent no-ops and FFFF's represent traps to the CPU, both of which were benign should the intermittent failure occur again.

The back-fill was accomplished by creating a memory image with the desired 0000/FFFF pattern. VMFLOAD was used to generate a series of uplink messages. The uplink messages were processed by the sequence translation software, tested in the SVL, and then radiated to the spacecraft to neutralize the memory while trouble-shooting could continue.

Spacecraft experts were able to determined a probable failure scenario and also determined that if the problem diagnosis was correct, the problem would eventually "heal" itself (Ref. 2). With the off-line memory loaded with the 0000/FFFF pattern, the AACS was essentially operating in a single string mode. Therefore, the project was anxious to get an operational load into the off-line memory to restore the AACS to redundant dual string operation.

The first step in the process was construction of a valid memory image. The memory image was constructed by utilizing the last configuration managed on-line image and overlaying it with the up-to-date parameters from the VMPTRK tracked parameter database. VMFLOAD was used to generate a series of uplink formatted messages from the memory image. These uplink formatted messages were processed by the sequence translation software to generate the actual uplink files. These uplink files were tested in the SVL and then radiated to the spacecraft. Following successful uplink radiation, the memory image was read-out and verified via MEMANAL. After successful verification the off-line memory and CPU were returned to normal operation.

4. SUMMARY

Magellan State Tracking is primarily a manual function allocated to the sequence design process and the individual subsystem analysts. High level groundwork for the state tracking process was initiated by the sequence block dictionary. This implementation did not, however, facilitate the continuation or automation of the state tracking process required for mission operations. The sequence generation process provides a mechanism to track component states via the initial conditions/final conditions scheme. The states reflected by this scheme also require manual updates whenever state changes occurred in a non-sequence fashion. Each sequence assumes a certain configuration and the spacecraft team did not have an efficient way to implement changes in the latter part of the sequence cycle. The burden of late configuration changes was always borne by the Upload Preparation Group who worked tediously implementing manual edits to the final sequences reacting to late or unanticipated configuration changes.

The level of manual activity required for state tracking was not a concern when the spacecraft team was at maximum staffing levels and contained experts dedicated to analyzing major state changes and the fault protection implications. As the team size continues to shrink and critical personnel leave the team, the worth of a more automated state tracking mechanism appears more attractive. However, at this stage of the project the creation of such a tool is not feasible. Given the budget constraints encountered by the mission operations designers and the repetitive nature of the mapping mission this was probably the correct approach. It is difficult to quantify the savings that an automated state tracking tool would have achieved.

The spacecraft team did develop a method to autonomously track after-the-fact spacecraft states. This method can track current states based on spacecraft telemetry but does not project future state configurations for sequence planning or fault protection configuration purposes. This tool is used primarily to record spacecraft states during critical periods or spacecraft anomalies.

The process of memory state tracking including flight software code and parameters has been a tremendous asset to the Magellan program. The ability to track critical parameters and reload partial or entire on-board computer memories with minimal effort has proven to be an integral part of anomaly resolution and recovery. The three tools discussed above, VMFLOAD, VMPTRK, and MEMANAL, all utilize common input formats and are compatible with the formats used by the sequence translation software used to generate and translate command sequences. The input and output formats are also compatible with the formats used by the SVL and the flight software development processes. This format commonality has allowed the tools to provide the flexibility needed for mission operations.

5. FUTURE THOUGHTS

Spacecraft and Memory State Tracking as ground analysis tasks are an integral part of mission operations. Spacecraft designers should strive to place all spacecraft state information in telemetry. Memory contents and states can be tracked based on a known configuration and updates to that configuration. The majority of spacecraft and memory state changes are accomplished by the command or sequence process and can be determined via telemetry. An effective state tracking tool must provide an automated and integrated method to:

- Interface with the command system to determine commanded state changes,
- Interface with the telemetry system to determine autonomous changes to states and provide a check-and-balance mechanism with the intended states from the command system,
- Provide the capability to determine required state changes based on the current state and the state required to support the desired sequence or mission phase,
- Interface spacecraft and memory states with the spacecraft test facilities, and
- Provide effective reporting of states at both the management and analyst levels.

With such a large impact on the sequence design and generation process, spacecraft and memory state tracking should be an integral part of the sequence design and generation tools.

6. REFERENCES

1. Spacecraft Flight System Block Dictionary, VRM-SE-025, Rev: Final, Feb. 1991, pages 1-4.

2. Kasuda, Rick, "ISA 8899 AACS Memory B Bit 4 High Addressing Failure", January 4, 1992.

7. ACKNOWLEDGEMENT

The work described in this paper was accomplished by Martin Marietta under contract to the Jet Propulsion Laboratory, California Institute of Technology and sponsored by the National Aeronautics and Space Administration.