Data Management for Mars Exploration Rovers

Joseph F. Snyder

Jet Propulsion Laboratory

California Institute of Technology

Pasadena, California, USA

David E. Smyth

Oak Grove Consulting, Inc

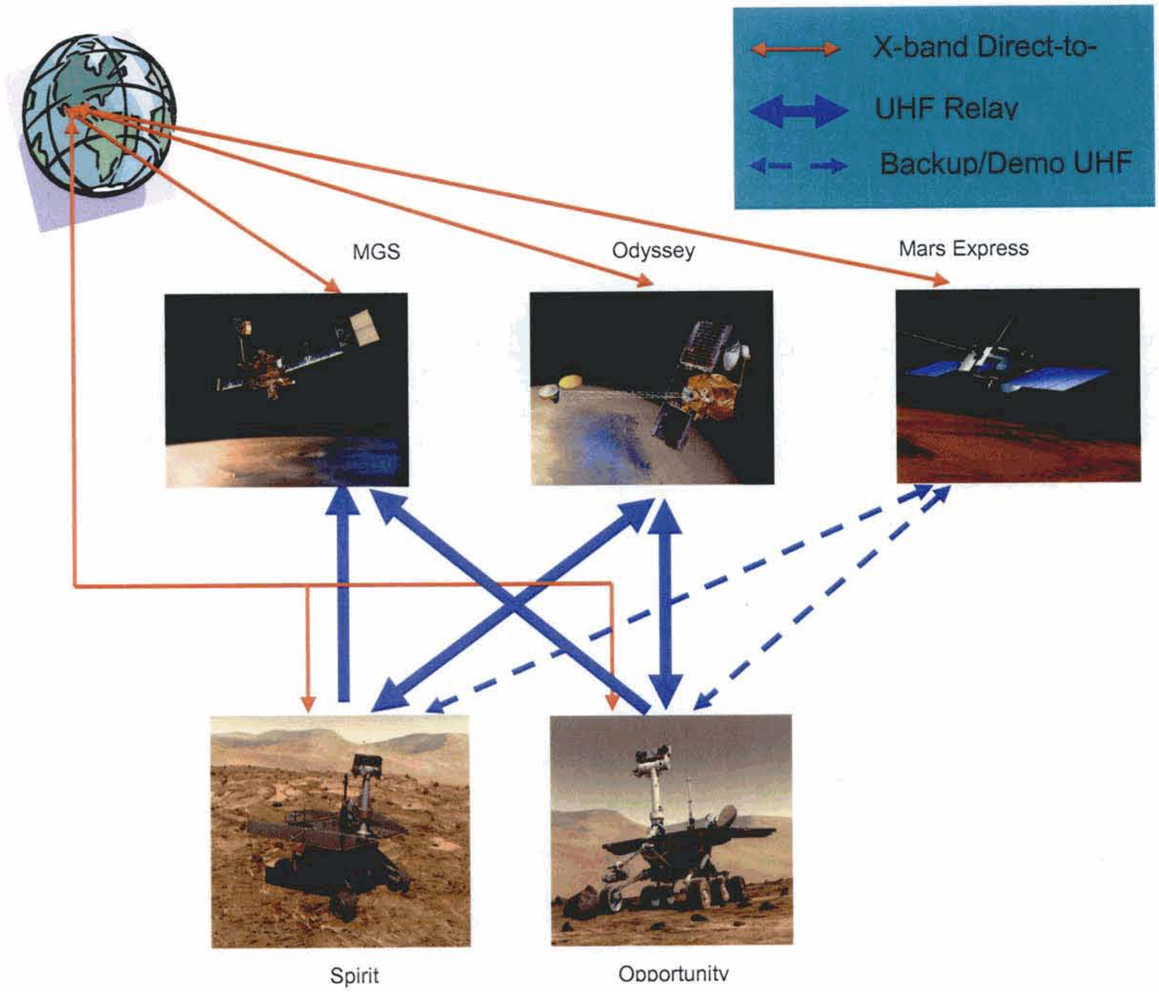
Pasadena, California

Abstract

Data Management for the Mars Exploration Rovers (MER) project is a comprehensive system addressing the needs of development, test, and operations phases of the mission. During development of flight software, including the science software, the data management system can be simulated using any POSIX file system. During testing, the on-board file system can be bit compared with files on the ground to verify proper behavior and end-to-end data flows. During mission operations, end-to-end accountability of data products is supported, from science observation concept to data products within the permanent ground repository. Automated and human-in-the-loop ground tools allow decisions regarding retransmitting, re-prioritizing, and deleting data products to be made using higher level information than is available to a protocol-stack approach such as the CCSDS File Delivery Protocol (CFDP). Such decisions enable effective use of the unique characteristics of each relay asset at Mars as well as effective use of the very limited uplink bandwidth, while reliably gaining timely access to data required for surface operations planning, and to significant science data. By basing the system conceptually on an on-board POSIX file system that becomes mirrored to an infinite hierarchical file system on Earth, tools emerge from across the user community that help automate and provide useful insight into mission level data management.

Introduction

NASA's Mars Exploration Rovers (MER) project landed two rovers, Spirit and Opportunity, on Mars on January 3 and January 24, 2004, respectively, where they continue to operate in extended mission mode as of August 2004. To date, Spirit has operated for over 218 Sols (Martian days), and Opportunity for over 197 Sols. One of the many unique aspects of the MER mission is its consistent use of both a Direct to Earth telemetry/command link and a Relay telemetry link using the Odyssey, MGS and Mars Express orbiters. Using these links, Spirit has transmitted more than 30 gigabits of data back to Earth, and Opportunity has transmitted more than 28 gigabits of data back to Earth.

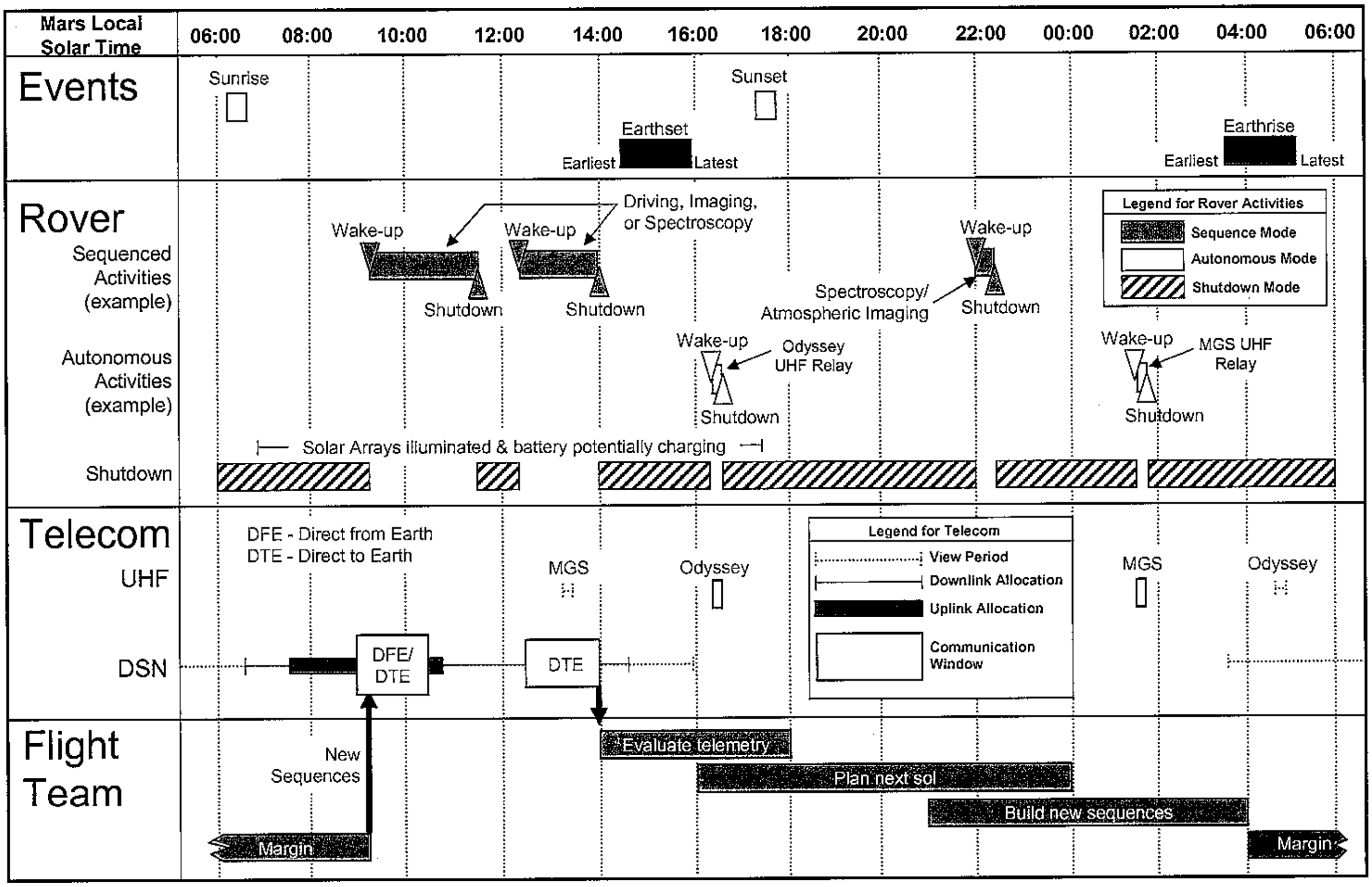


The MER project had a highly compressed schedule that began in July 2000, with launches on June 10, and July 7, 2003. One aspect of the compressed schedule was that the MER effort intended to leverage as much as possible from the successful Mars Pathfinder mission, including the flight computer, flight software and ground data system. However, the scope of the MER mission, particularly after landing, was far more complicated and demanding than the Pathfinder mission. One of the areas that required a significant departure from the Pathfinder architecture was in the area of Data Management.

Data Management means different things to different constituencies. For this paper, the "Data" in data management refers specifically to the "Data Products" generated by the rovers intended for possible telemetry to Earth, as well as the Data Products actually received on Earth. "Management" is more broadly defined, and covers the actual handling of the Data Products on-board, on Earth, and the interface between the two. "Data Product", in turn, refers specifically to a data set meant to be treated as a logically discrete unit on-board for storage and transmission to Earth, as well as for receipt, storage, and processing on Earth. In other words, a Data Product is any set of data that can be treated as a file. This includes, for example, images, spectra, sets of discrete time-ordered measurements, state histories, time-ordered event data, and so on. In the development and execution of the MER mission, Data Management was and is treated as a System, and as such it was intended from the beginning to respond to demands not encountered previously.

Objectives for Surface Operations

The principal driver for MER Data Management was support for Surface Operations. A typical Sol during Prime Mission Surface Operations is depicted in the figure below.



Specifically, the Data Management System must support the following general Surface Operations criteria:

- Once per Sol uplink of a command load: The command load contains all command sequences that direct the rover's activities for the next Sol. This includes all commands related to Data Management.
- Rapid command turnaround: The command upload for tomorrow (the next Sol) can not be created until the critical results from today (this Sol) are received.
- Limited communications opportunities: Continuous communications with either rover are not possible. Uplink and downlink opportunities are restricted due to

rover energy and thermal constraints, orbiter over-flight timing, and Earth-Mars geometry.

- Prime mission duration of 90 Sols: Achievement of all mission objectives within the short prime mission required a very high workload, with full operations teams for each rover working on Mars time. Since the rovers are on opposite sides of Mars, this meant that one vehicle was active on the surface during its Sol, as the planning team for the other rover worked on the command load for its next Sol.

In addition, the Data Management System must respect important MER-specific flight system constraints. These include:

- Frequent Shutdown/Wakeup Cycles: The rovers are solar powered, and thus the majority of activities must be performed during the Martian day, with limited operations at night. Even during the day, the rovers are energy and thermally constrained. These constraints require the rovers to shutdown and wakeup several times each Sol. Because all data in Random Access Memory (RAM) is lost on shutdown, all data products must be maintained in non volatile memory. In contrast, Mars Pathfinder was never planned to support shutdowns.
- Data Generation and Storage Capacity Limitations. The rovers' capability to generate data exceeds the capability to store, stage, and transmit it. The available non volatile memory, FLASH, has about 200 megabytes (1600 megabits) available for data products. The available RAM for staging is about 15 megabytes (120 megabits).
- Asynchronous link bandwidth. The total uplink bandwidth available is tightly limited. Furthermore, only Odyssey can support relay uplink, and is not practical given the Operations criteria described above.

Thus, the single all-encompassing Data Management objective was to accommodate "human-in-the-loop" tactical operations while simultaneously automating standard operations, all the time respecting the unique flight system constraints of MER. A Data Management process must execute every Sol to support the uplink opportunity for the next Sol. This process must cover all data received since the previous time it executed. It must support changes in downlink opportunities, changes in data collection, and changes in data criticality.

This overall objective was handled by meeting the specific objectives described in the following paragraphs.

Insight Into On-board Data

Each rover generates large amounts of data from multiple sources, triggered by both commanded and autonomous events. The amount of data created each Sol is potentially greater than the amount that can be transmitted to Earth each Sol. Therefore, a key MER Data Management objective was to enable the identification of on-board data using some form of a compact synopsis of each on-board data product. Previous missions did not generally have intuitive methods for understanding and managing data once it was designated for downlink.

This was achieved by maintaining an on-board catalog of metadata, or "data about the data." Each data product consists of a file (e.g., image data, robotic arm motion history, etc.), and an associated instance of metadata that described the file. The metadata includes: the generating source of the data (e.g., the left Panoramic Camera); the type of the data (e.g., a compressed full image); the autonomous action or the sequence and command that caused the data to be generated; the time the data was sampled (e.g., the camera shutter time); the size of the file; and a number of other attributes.

Before each telemetry session, the on-board Data Management software creates a data product containing a summary of this metadata, and sends it in telemetry. This catalog data product, called a Data Product Summary Report, provides clear insight into the data on-board the spacecraft.

Explicit Downlink Priority Control

The rovers have a tightly constrained operational planning cycle supported by very specific communications passes. Furthermore, individual data products were expected to have highly dynamic usefulness. For instance, certain data products would be critically important immediately after creation, but would become essentially worthless after the next communications pass. Conversely, certain data products initially considered of low value would become much more important based on science and engineering analyses operating over divergent timelines. Finally, various categories of otherwise identical data would have very different criticality based on the current Sol's activities relative to the next Sol's planned activities. Therefore, a key MER Data Management objective was to provide fine control over data priority (the order that data products are selected for transmission to Earth). Previous missions generally had only coarse priority control capability over telemetry data.

This was achieved by implementing flexible priority control accessible directly by the activity planning cycle. Priority is assigned, at the discrete data product level, via the commands resulting in data product creation, e.g., a command to take an image. This priority is changeable via direct reprioritization commands. Finally, the reprioritization commands can operate on sets of data products (wild-carding) as well as operating on individual data products.

Transmission Path Control

MER is one of the first missions to use both Direct to Earth and Relay telemetry links. The availability, bandwidth, latencies, and reliabilities associated with these links varied considerably. The Direct to Earth link generally had low bandwidth, the least latency, and the highest reliability. The Relay links had much higher bandwidth, fixed availability, and variable latency and reliability. Therefore, a key MER Data Management objective was to provide control over which path data products are sent to Earth.

This was achieved by implementing transmission path control at the individual data product level. Transmission path control includes specifying transmission over a preferred path, multiple paths, or first available path.

Retransmission Control

Given the previously discussed dynamic data criticality and multiple transmission paths, another key MER Data Management objective was to provide flexible retransmission control. As with priority control, previous missions generally had only limited retransmission capability.

This was accommodated in two ways. First, because partially received data products may be critically important, partially received data must be as useful as possible. Therefore, to the greatest extent possible, data products were implemented to be self-identifying and "process-able" at the part level. For example, a partially received image data product can still be completely identified, and usually can be displayed.

Second, retransmission control was implemented to allow missing parts to be reprioritized, and allow for a different transmission path. Finally, retransmission control commands can operate on sets of data products (wild-carding) as well as operating on individual data products. That is, the commands operate on groups of completely "lost" data products, as well as loss of parts within a single data product.

Deletion Control

The final key MER Data Management objective was to provide flexible deletion control. As with reprioritization control and retransmission control, the deletion control capability was implemented such that commands can operate on sets of data products (wild-carding) as well as operating on individual data products.

Development History

The MER mission offered many challenges to the software development team. The mission goals were uncompromising, and requirements were comprehensive. A spin stabilized cruise phase took us to Mars. The critical EDL (Entry, Descent, and Landing) phase was followed by complex deployments during ITE (Impact To Egress). The engineering subsystems require extensive on-board control and fault responses. The telecommunications system supports UHF uplink and downlink relays, as well as direct-to-Earth uplink and downlink communications over fixed low gain and steerable high gain antennae. The numerous and diverse scientific instruments require sophisticated on-board data processing. A fully articulated robotic arm manipulates a fistful of instruments and devices. An instrument mast provides azimuth and elevation pointing for high-resolution imaging and spectroscopy instruments. Machine vision and autonomous navigation enable mobility across Martian terrain.

All of this capability resulted in over ten times as much flight software being developed for MER as for Mars Pathfinder: 360K SLOC on MER, 32K SLOC on MPF. However, the MER development timeline was the same three years as the much simpler Mars Pathfinder mission. The MER processor was the same 20 MIP PowerPC used on Mars Pathfinder. The MER software team was only three times the size of the Mars Pathfinder team.

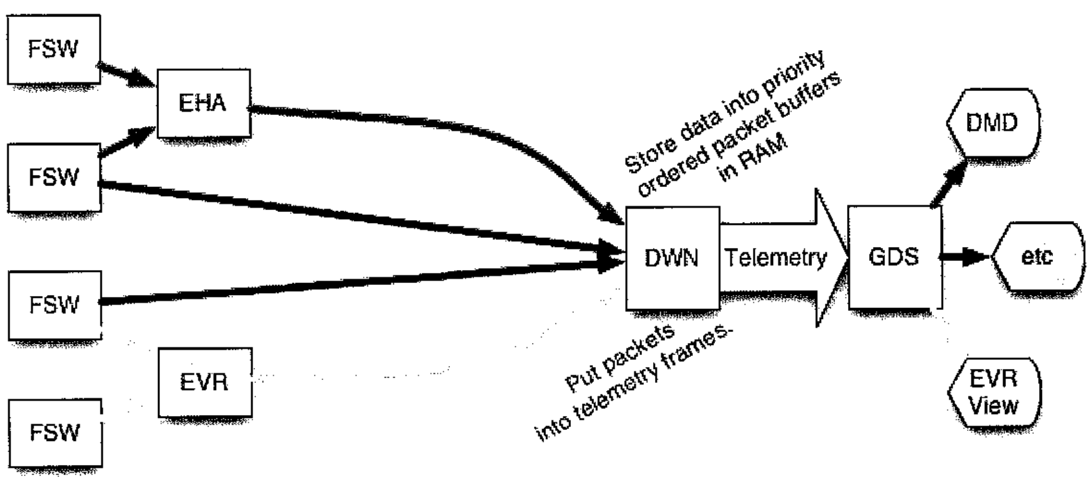
In order to have confidence in meeting the schedule and key requirements, MER started with the Mars Pathfinder software system for both the flight software and the ground data system. This flight software code base was successful the first time, and has proven successful on many subsequent missions including Deep Space One, Stardust, SIRTF (Spitzer), Odyssey, Genesis, and other Lockheed Martin Astronautics missions. The ground software code base, called AMMOS, has been used for every JPL mission since 1990.

Both the ground software and the flight software share architectural similarities: both are built on UNIX or POSIX APIs; both use many client and server tasks with predominately point to point communication links; both are mostly object-oriented; and of course both utilize CCSDS protocols for space communications.

Reuse of the Mars Pathfinder Flight Software Code Base

The Mars Pathfinder flight software code base (MPF FSW) provided many infrastructure subsystems, including EVR (an event reporting service that provided functionality similar to "fprintf(stderr,...)") and EHA (channelized telemetry service) that efficiently utilized CCSDS telemetry and bandwidth and were already integrated into the AMMOS ground data system. The MPF FSW also provides DWN, a CCSDS packet telemetry service with prioritized packet selection. The DWN packet storage was in RAM, and was the persistent storage of (nearly) all telemetry data onboard the spacecraft.

The figure below depicts how Mars Pathfinder DWN (downlink) system insulates clients from the transport layer.



Genesis of MER Data Management System

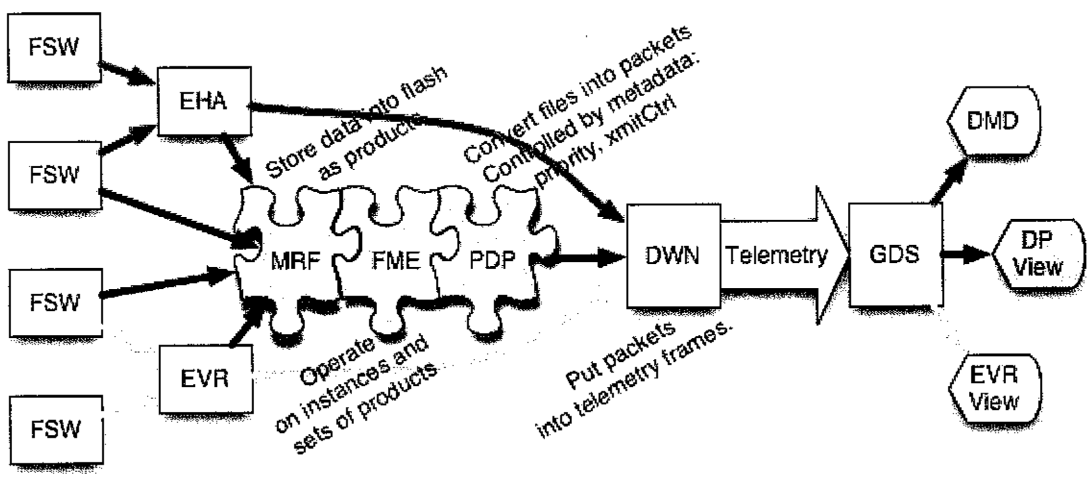
However, Glenn Reeves, the flight software architect on the MER mission, soon realized that the operational objectives and system constraints described earlier, particularly the requirements to shutdown the system at night, and wakeup for nighttime UHF relay passes, would require a fundamentally different approach than that provided by the legacy MPF system. The direct implication of a system that has frequent shutdown/wakeup cycles is that RAM can not be used for persistent storage; science and engineering telemetry data must be stored in non-volatile FLASH memory. FLASH memory was not well suited for use as the underlying media for the MPF packet based system. Furthermore, the intricacies of the FLASH devices were such that a COTS (commercial off-the-shelf) solution for storing data in FLASH was necessary. This was TFFS (True Flash File System), a flash device driver, under the DOS file system provided by the VxWorks operating system.

The "fundamentally different approach" selected became the MER Data Management system. This would be a complete system of integrated flight and ground software. On-board, the Data Management system would be responsible for writing all recorded telemetry data (channelized engineering, EVR events, and all science and engineering files) to persistent storage (FLASH) as files. Metadata would describe each file on-board. A file, with its associated metadata, was defined as a Data Product. During communications sessions, the on-board system also converts the Data Product files to CCSDS packets. On Earth, the Data Management system would be responsible for reassembling the Data Products from the packets, providing simple data product viewing, identifying missing data product parts, and supporting generation of retransmission and deletion commands.

Perhaps the single biggest change migrating from the Pathfinder data architecture to the MER data architecture was to visualize and work with data as Products instead of Packets.

The on-board Data Management FSW consists of three components: MRF (Mirror RAM to FLASH), FME (File Metadata Engine), and PDP (Packetize Data Products).

The figure below depicts how the MER Data Products system supports file and legacy EHA and EVR data.



Almost all client tasks (VxWorks threads) on the MER spacecraft generate data products. MRF provides the client API to the Data Management system. The MRF API is intentionally similar to POSIX fopen(), fwrite(), and fclose() calls. On UNIX, a call to fopen() that creates a new file will cause a certain amount of metadata to be created based on the context of the task calling fopen(), including the user ID, the group ID, the time, the permissions, and so on. Other metadata (the file name) is stipulated by arguments to fopen(). On MER, the call to mrf_open() includes certain context-related attributes from the caller task (the sequence and command to which the task is responding), the validity time of the data (for example, the shutter time of an image), and finally the type and priority of the data. From this information, a Metadata instance is created, and this metadata will remain related to the resultant file. As the client writes data using mrf_write(), the data is (eventually, due to buffering) written to FLASH storage, and the metadata is updated (e.g., the number of records, and the size of the file). The client invokes mrf_close() when done, and MRF then ensures the data is moved to the FLASH device.

The metadata is manipulated by FME, the file metadata engine, on-board the spacecraft. FME supports operations on sets of data products, on individual data products, and on portions of individual data products. Using the metadata, FME selects files to be transmitted via an X-band (direct to Earth) or UHF (relay via Mars Express, Odyssey, or MGS) telemetry pass. FME tells PDP which files need to be packetized for telemetry, one at a time.

PDP uses the metadata for a data product to find the file, and to packetize the file according to record boundaries if they exist. For example, the ICER compression algorithm used for many images generates a record-oriented compressed image file, such that each record is an error containment region: losing one region does not prevent the image from being uncompressed and viewed, but it results in some area of the image being "fuzzy." Each packet created by PDP contains the normal CCSDS packet primary header, the JPL standard secondary header with spacecraft time, and an UPTH: Universal Product Tertiary Header. The UPTH contains sufficient metadata to identify the product to which each packet belongs, and the position within the file the data of the packet resides.

The ground based Data Management software re-constitute files from CCSDS packets that contain UPTHs. Even if some packets are lost, the files are reconstituted, as the UPTH contains enough information to allow "holes" to exist within the re-constituted files. Rather than using timers to determine when a product completes, the MER ground system can detect the end of a product in two ways: (1) by observing the last packet of a product via information in the UPTH, or (2) by detecting end-of-pass. Client software on the ground can subscribe to a JMS (Java Messing System) server for notifications of the arrival of new products (including partial products) of specific types.

Data Management Surface Process

As MER completed the prime mission and moved into an extended mission posture, the MER Data Management process has evolved with changing Surface Operation priorities. Currently, the process performs the following tasks once each uplink cycle (1 per rover per Sol), most of which are completely automated, and all of which are highly configurable.

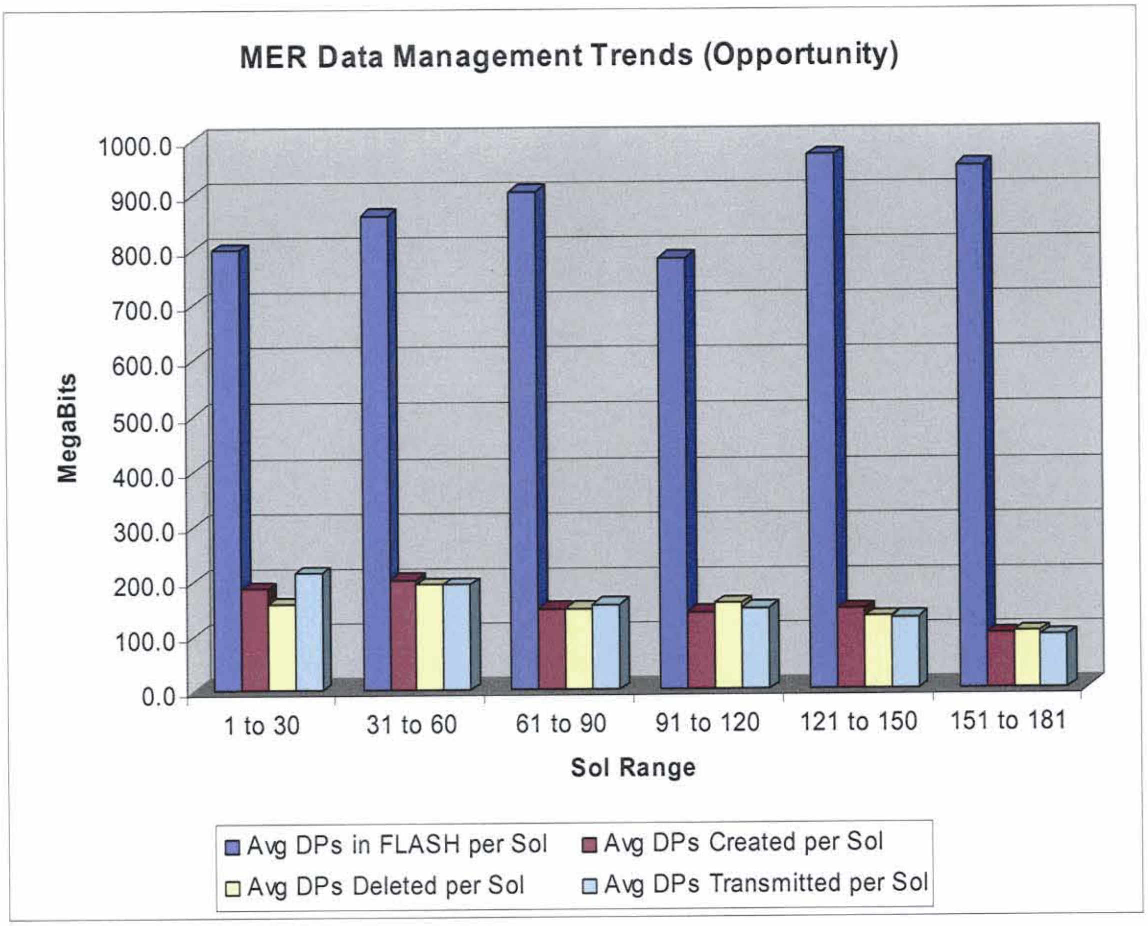
- Identifies all partially and completely received Data Products since the last cycle completed. This includes all data products received from all relay assets and direct to earth links.
- Generates individual Retransmit Commands for each identified missing or partially received Data Product. This typically results in tens of commands, which fit comfortably in the uplink bandwidth. The automatically generated retransmit commands account for Data Products that should never be

retransmitted, and for Data Products that need standard priority changes and standard transmission path changes. Finally, in the unusual case where the number of retransmit commands exceeds a configurable maximum, blocks of commands for completely missing Data Products can be replaced with single "Group" retransmit commands, thus recovering uplink bandwidth.

- Performs automatic corruption checking for certain classes of Data Products. It turns out there is a non-Data Management related MER flight software bug that can corrupt small amounts of data just prior to transmission. It does not affect the source data products. When corrupted data is detected, individual Retransmit Commands for the corrupted parts are automatically generated. When the retransmitted data is received, replacement requests are generated to update the original corrupt Data Products on the ground.
- Generates single "Group" Delete Commands for each non-contiguous transmitted time range. Received data products are clustered with transmission time ranges corresponding to the communications sessions that occurred since the last Data Management cycle. Single group delete command can delete any number of data products. Typically, such commands result in hundreds Data Products deleted per Sol.
- Generates single "Group" Delete Commands for unsent classes of data products that meet configurable age and size parameters. Typically, such commands result in tens to hundreds of Data Products deleted per Sol.
- Generates single and "Group" Reprioritization Commands for certain classes of Data Products that meet configurable age and size parameters.

In addition, because of the visibility into the data products on-board, and the efficiency of the "group" commands, highly custom requests are accommodated nearly every Sol. These include deleting sets of data products deemed worthless, reprioritizing data products from specific sequences, time ranges, or other criteria, and accommodating experimental relay passes without compromising critical data receipt.

The following figure shows a set of average data volumes over 30 Sol intervals for Opportunity.



Conclusions and Lessons Learned

Some of the major highlights illuminated by the development and operational use of the MER Data Management system are listed below.

- Data Management as a function begins during development and continues to the end of mission. Individual Data Products need to be identified in a way that facilitates identification and cross referencing across test and flight environments. Of particular concern is tracking and managing spacecraft time systems as they relate to the large set of time tags generally associated with data.
- Use of a standard stdio-like API greatly facilitated the software development effort. First, it allowed mirroring of on-board file systems such that data products could be compared bit-for-bit at many points in the end-to end data flow, greatly enhancing validation. Second, it allowed for lower level Data Management implementations to be updated or completely replaced with minimal client impact.
- On-board flexibility was greatly enhanced and uplink bandwidth was highly optimized by implementing extended wild-card capabilities across all data products.

- Designing for "human-in-the-loop" from the beginning allowed for maximum flexibility as Surface Operations evolved. The system remained highly automated while still able to respond to unforeseen pitfalls and opportunities.

Operational use of the Data Management design also exposed the need for further improvements.

- The retransmit capability should support multiple part ranges. The current system is limited to one range per data product. This limitation did not consider the impact of periodic data dropouts like those seen on data relayed via MGS.
- Commands operating on individual data products should use compact identifiers to further reduce uplink bandwidth. The current system uses long file name strings, which are inefficient.
- The transmit path control capability should have allowed specification of specific relay paths. The current system only allows specification of Direct-to-Earth or Relay. This resulted in unnecessary complexity when dealing with MGS, Odyssey, and Mars Express communications sessions.
- Data Product identification/naming, and generating sequence/command identification needs to be standardized. The current system assumes MER/MPF specific conventions.
- Wild-card capabilities need to be extended. The current system has impressive wild-card features, but they are still primitive compared to most command line capabilities.

In summary, the MER Data Management System has been a highly successful next step beyond the pure CCSDS packet protocol approach to telemetry data used by most missions to date. In particular, the MER Data Management System highlights the advantages of an Integrated System approach, involving flight software plus ground software, and both uplink and downlink processing. This system effectively flattens the protocol layers, from presentation and application layer to transport layer, to maximize the efficiency of the overall spacecraft-Earth system, while still leveraging the interoperability, re-use, and flexibility of existing protocol stacks from proximity one through CCSDS frames and packets.

The authors believe the MER Data Management System is well suited for future missions. It is naturally complementary to autonomous spacecraft applications such as the Autonomous Sciencecraft Experiment. It has straight-forward interfaces to client flight software. It leverages existing ground data system infrastructure, including important institutional capabilities such as the Multi-mission Image Processing Lab. Finally, and perhaps most importantly, the MER Data Management System is tunable to a specific mission's operational requirements, and accommodates those requirements as they evolve from pre-launch development through prime mission and extended mission lifetimes.

Acknowledgements

The authors would like to acknowledge the MER Flight Software Architect, Glenn Reeves, who conceived the MER Data Management system, and provided the technical guidance and support necessary to make it a reality.

The authors would also like to acknowledge the principal Ground Data System Software Engineers, David Noble and Jesse Wright. Without their dedicated effort, there would be no "System" in the MER Data Management system.

The efforts and research described in this paper were carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

1. Erickson, J.; Adler, M.; Crisp, J.; Mishkin, A.; and Welch, R.: "Mars Exploration Rover Surface Operations", IAC-02-Q.3.1.03, 53rd International Astronautical Congress, October 10-19, 2002, Houston, Texas, USA.`