Cassini CAPS Ground System Evolution and Lessons Learned

Judith D. Furman
Southwest Research Institute
6220 Culebra Road
San Antonio, TX 78238
210-522-6040
jfurman@swri.org

Gregory D. Farris
Southwest Research Institute
6220 Culebra Road
San Antonio, TX 78238
210-522-5811
gfarris@swri.org

Charles Zinsmeyer
Southwest Research Institute
6220 Culebra Road
San Antonio, TX 78238
210-522-5018
charles.zinsmeyer@swri.org

Prachet Mokashi
Southwest Research Institute
6220 Culebra Road
San Antonio, TX 78238
210-522-5925
pmokashi@swri.org

David T. Young
Southwest Research Institute
6220 Culebra Road
San Antonio, TX 78238
210-522-5510
dyoung@swri.org

Abstract—Recently the Cassini CAPS team investigated the number of different software tools, programming languages, programs, and major technologies used in building and running its ground system[1][2]. A total of twenty six different tools, programs, and major technologies were used in the downlink portion of the ground system and five different tools, programs, and major technologies are used for the uplink portion. This collection of tools, programs, and technologies were not considered in the initial ground system design at the beginning of operations, but were added as they were needed to address changing requirements.

During the pre-launch phase of Cassini Huygens in 1997, operations were performed in a limited capacity in the calibration facilities. After launch, process automation increased gradually, as the ground system continued to grow. New technologies were incorporated as they emerged, and were added to the ground system, although they may not have been part of the initial plans. This led to some unexpected and sometimes creative inter-connectivity solutions to ensure that they could be integrated with the existing system.

This paper will document the changes in the ground system from the period just before the first instrument check-out in January 1999 through the end of 2008. We will discuss why the technologies, programs, and tools were added. Lastly, we will look at the ground system from a management perspective in an attempt to determine how the system could have been built modularly, considering that emerging technologies and other capabilities could be added at a later date.

TABLE OF CONTENTS

1. INTRODUCTION

NASA/ESA's Cassini-Huygens spacecraft successfully entered into orbit around the Saturnian system on July 1, 2004. Cassini-Huygens is a robotic spacecraft with 12 instruments on the orbiter and 6 instruments on the Huygens probe. The Cassini-Huygens mission is managed by the NASA/California Institute of Technology Jet Propulsion Laboratory (JPL) in Pasadena, CA. Cassini was approved by Congress in 1989 as part of a larger CRAF (Comet Rendezvous Asteroid Flyby)/Cassini program and was planned for launch in 1995 with CRAF following in 1996. However, in 1992 there was a major reorganization that cut the CRAF portion out of the budget and the Cassini program funding was reduced [1, 2]. Due to restructuring, the launch date slipped and launch was performed October 15, 1997. Cassini launched in 1997 and entered the Saturn system on July 1, 2004. The main Cassini mission involved a 4-year prime mission around Saturn. Near the end of the prime mission, the Cassini project negotiated with the National Aeronautics and Space Administration (NASA) for a 2-year extended mission. This extended mission, the Cassini Solstice Mission, was approved by NASA, and is currently being executed. As the Solstice mission has progressed and the scientific value of the data being

[1] 978-1-4244-3888-4/10/$25.00 ©2010 IEEE
[2] IEEEAC paper #1322, Version 3, Updated January 5, 2010

returned is still increasing, the Cassini scientists are back in negotiation with NASA to fund a second extended mission (Equinox Mission) that will last through July 2017. The main science goals of this second extended mission will be to observe seasonal variations, as one (1) Saturn year is equivalent to 29.7 Earth years.

Along the way, there continued to be more budget cuts that were handled in different ways. One way to reduce costs during development was the removal of the high-precision scan platform and turntable that would have de-coupled the interactions between the magnetospheric and plasma science (MAPS) instruments and all the other instruments. It was not until much later, that the full effects of the removal of this platform were discovered [2, 3], including the need to negotiate pointing and spacecraft operations modes, and much later, dealing with the issues arising from over use of the spacecraft reaction wheels. One of the other methods of cost reduction was to implement distributed operations. This choice was made up-front, unlike the Mars Exploration Rover Mission (MER) that started with centrally located operations and later moved to distributed operations[3, 4]. Given that the responsibility for instrument design resided at the Principle Investigator's (PI) home institution, there was no need to have a representative at the Jet Propulsion Laboratory (JPL). Additionally, with such are large design team, there really wasn't room available at JPL to centralize the design work [2]. Hence it was decided to keep the expertise in-house at the PI institution by distributing the operations responsibility. This distribution reduces the communication and approval layers in the operations processes and frees the ground system. An additional reason to choose distributed operations is empowerment. The PIs are now fully responsible for their instrument from the design period all the way through operations, including building instrument commands, instrument monitoring, and archiving [5].

2. CASSINI PROCESSES

There are two processes in the Cassini ground data system, the uplink process (see Figure 1) and downlink process (see Figure 2). The uplink process is an involved set of processes to take the science goals at the start and to have a product to uplink to the spacecraft at the end. The process that integrates and implements all science activities is called the Cassini Science Planning process [6]. The last set of processes involves building an integrated product that includes all of the different instruments commanding, any engineering activities (ie. flight software updates, preventative instrument maintenance, or spacecraft engineering activities), spacecraft pointing commands, and commands to handle the data management and Deep Space Network (DSN) commanding[7, 8]. The final product is clean of any flight rule or other violations.

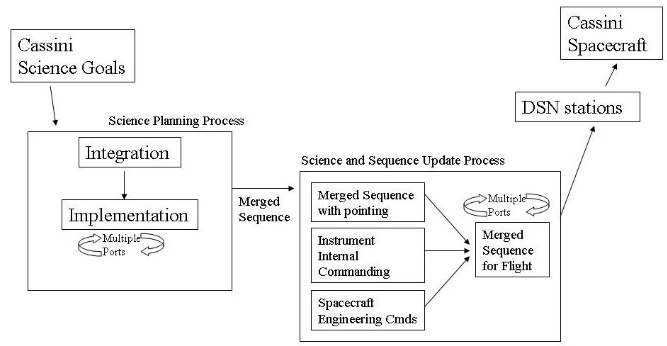


Figure 1 - Cassini Uplink Process

The sequence execution process is where the uplink processing product is transmitted to the spacecraft and executed on-board. This leads to the downlink process where the results of the uplinked product execution are downlinked in the form of science and engineering data. These data are processed on the ground in different ways: instrument and spacecraft health and safety, science analysis, and archiving. Instrument and spacecraft health and safety are first addressed during the real-time downlink of data. Even if an instrument is not viewing in real-time, JPL has a Cassini ACE (Aerospace Communications & Information Expertise [9], a person who sends commands to the spacecraft) that remains in front of a bank of computer monitors during active DSN downlink periods in order to monitor any anomalous readings from the spacecraft or any of the instruments. Additionally, instruments have their own process in place to determine whether their instrument has encountered anomalous behavior. The process for CAPS is not done with the real-time data stream, but is done off-line during the automated nightly processing of science data.

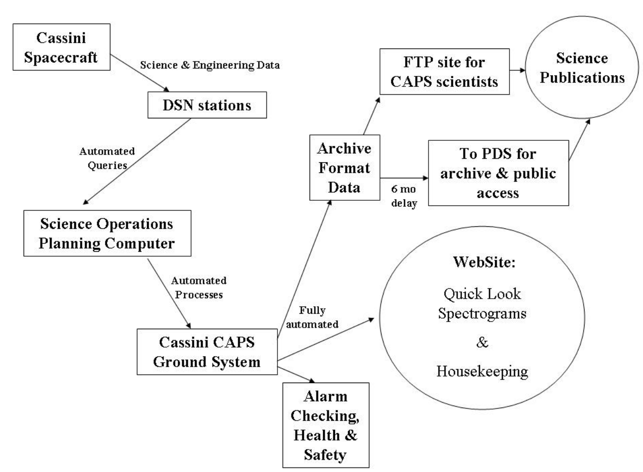


Figure 2 - Cassini CAPS downlink process

For the science data analysis portion, the science teams are more involved. The associated CAPS scientists are provided with a mechanism to retrieve processed data. Using this data, the scientists can perform science analysis, including collaboration with other scientists. As of October 2009, there are over 175 publications with CAPS authors and ~497 conference talks. Collaboration and science analysis is running very nicely.

Lastly is the science data archival process, which is archival of all the instrument data so the general public can access it. This is a requirement for all space missions. There are different locations for instruments to submit their archived data, based upon the function of the science. For fields, particles, and waves, data is submitted to, stored, and can be retrieved from the Planetary Data System (PDS) [10]. Delivery of CAPS data is made to the PDS every quarter. Currently all the cruise and prime mission data is in the PDS, as well as data from the first six months of the Extended Mission.

The complex, yet repetitive nature of the uplink and downlink processes lend themselves to automation. For the downlink process, the goal is for data to flow smoothly from the spacecraft to the scientists without manual interaction by the operations group. For the uplink process, the goal is to have as automated a process as possible such that mis-commanding of the instrument is less likely to happen and that deadlines are not missed. For the instrument teams, as long as requirements and deadlines are met, the Cassini project did not dictate an implementation strategy. Hence, each instrument team was left on its own to come up with how it would respond to this challenge of managing distributed operations and the requirements levied by the uplink and downlink processes.

One Cassini paper of interest was from the Magnetosphere Imaging Instrument (MIMI) that describes how choices made by the Cassini project affected how their instrument team operates within the Cassini operations defined framework [11]. The MIMI operations technical lead discussed how automation was necessary in keeping up with all the required process deliveries in order to avoid commanding errors.

The focus of this paper is not on the automation process, as much as it is a discussion of the changes in the ground system as time progressed on the project and the technologies that are now used in the implementation of the system. Additionally, there will be discussion regarding lessons learned during the ground system evolution, hardware used, informal solutions, failures, and our insight into a way to create a better ground system.

3. CASSINI CAPS GROUND SYSTEM

Very few activities were planned in Cassini's Cruise phase (defined as the time between launch and six months prior to the start of Prime mission). The plan was to perform instrument check-outs for flight software operation and science verification as well as periodic instrument maintenance activities like exercising moving parts. To conserve funding in the earlier phases on the mission, much of the flight software development (both instrument and spacecraft) was deferred until later. For example, CAPS launched with only two internal telemetry modes being defined and implemented and the flight software was updated during the cruise phase to the full compliment of nine internal telemetry modes. While the original plan did not include science activities during cruise, there was a change of focus after and cruise science was added.

Given that the operations processes were not fully in place and that the processes were being implemented much sooner than expected, JPL put in place an initial distributed operations process. This meant that many of the ground system elements were not prepared for full on operations. The basic ground system for CAPS during this period did not involve many tools, programs, or major technologies (and those it did contain are not used today). On the downlink side, a National Instruments™ Labview™ program was used for a basic quick look into data products as they were streamed from the spacecraft, through the system, and into basic flat file storage. As a basic part of the system, it employed shell and python scripting and the use of sockets for transmission between computers and software. CDF (common data format) formatted data files were used to pass along data to the science team. For uplink operations, the commanding was done manually, with the use of basic validation and verification software that was provided by JPL. To avoid commanding errors, multiple CAPS personnel verified commanding and intent. Reuse of files was utilized where possible and rigorous revision control was implemented.

During the first few years of operations and use of the ground system, several problem areas were uncovered that required re-design. It was discovered that working with CDF files was difficult. The overhead of accessing data and transferring it to tools that could process the data for scientific use was large and expensive. Additionally, each of the different science co-investigators had to deal with this hurdle on their own. Also, it was discovered that processing files for uplink manually was a long, tedious process. Making sure that science intent, data volume, and operational modes were being met was of high importance and needed more automation.

As time progressed, there was a decision within the project science group and project management that it would be possible to have more cruise science, specifically to collect science during the Jupiter encounter. This would be more than collecting science to verify proper working of the instrument as it presented a unique science opportunity. Additionally, it would be a good time to test the system with integration, implementation, and instrument commanding all rolled in to the process. The project decided to empower the instrument teams further by allowing them to design and build their prime observations. The Cassini project supplied a new tool to instrument teams for this purpose, called the Pointing Design Tool (PDT).

This new pointing design tool (PDT) and process of implementation brought about a whole collection of new problems, as not all of the flight rules were built into the system, and more iterations of the process were required as the teams responsible for the non-included flight rules had to verify the entire sequence was flight rule free. Additionally, many instrument teams were sharing coordinated observations, which made the process and interactions more complicated. CAPS did not require this level of coordination for most activities, as the instruments that collaborated with CAPS found the pointing designs to be adequate. As part of this process, the deputy PI for operations was handling the CAPS science integration and subsequently built tools that allowed him to visualize how the different spacecraft pointing attitudes affected the science that CAPS would be able to obtain (as well as how the other MAPS instruments science would be impacted by our pointing choice). A few of the other instruments with which CAPS collaborated had also built similar tools that made collaboration much smoother. The technologies used for the tools on the CAPS side were IDL™ and the IDL™ implementation of the SPICE Toolkit [12] (the ICY DLM). This tool is still used today after a few upgrades based on additions to the SPICE toolkit command set and IDL™ upgrades.

As a result of increased science activities a whole host of tools within the CAPS ground system were developed to handle the growing number of interactions with others as well as the increase in the volume of instrument commands and complicated data volume schemes. For CAPS most of the software built was used then and is still used today. However, there was some software that became obsolete rapidly (reading and plotting of CDF files) and new software was built to take its place. After integration, implementation, and execution of the Jupiter encounter, the lessons learned were applied to ground system development.

Growth of the ground system was painful as the system moved from a manual system to a more automated system. Even though it was desirable to move towards more automation, it was difficult to see where the system would end and how useful it would be when it was complete. One of the main concerns was in facing a change from a command line system to a web-based point and click system. After many meetings, an initial system was put in place to deal with the downlink portion of the process. The uplink process tools were planned for later development. Additionally, a system was built to allow the CAPS co-investigators to be more involved in the science planning process to help relieve the work load experienced by the deputy-PI of operations during the Jupiter encounter. The system, called eCARS (electronic CAPS activity request system), would allow the CAPS co-investigators to specify science observations that would be of interest. The goal was to collect scientific measurement requests from the co-investigators and merge the requests together to create a merged sequence of scientific requests for the CAPS instrument that could easily be integrated with the rest of the CAPS system and flow smoothly into operations.

4. MAJOR TECHNOLOGIES DISCUSSION

To give insight into the complexity of the system that was built, discussion of the major technologies used for the design will be undertaken. For the initial design, the major technologies used are as follows:

- The back-end of the system uses Java™ and Servlets.
- Data is stored using an Oracle® database, Structured Query Language (SQL), and Java Database Connectivity™ (JDBC ™).
- The presentation is done using Java™ Server Pages (JSP), HTML, Cascading Style Sheets (CSS), and JavaScript™.
- Web site content was restricted, and required user authentication, which was implemented using a lightweight directory access protocol (LDAP) service

Along the way, there was an additional set of Java-related technologies that were added, for convenience. These were added as they emerged and determined to be a good fit for the system.

- Database connection pooling was added to decrease the overhead of establishing new database sessions.
- Process thread pooling was added to ease the overhead of creating new threads to process tasks.
- APACHE™ Log4J[3] was added to increase the ease and flexibility of logging messages.
- JSP tag libraries were added to eliminate repetitive code in the presentation layer.
- APACHE™ Ant[3], a build tool, was included to make building and deployment of new versions of the ground system easier.
- ION™ (IDL™ on the net) was added which allowed access to IDL™ from a Java™ Virtual Machine.

[3] Apache Tomcat and Apache Struts are trademarks of The Apache Software Foundation.

Additionally, there was some legacy code left over from the earlier ground system, where the processing was done manually. There was complicated C and C++ code that processed the data packets into instrument cycles that was easier to reuse (with difficulty) rather than rewrite (and the budget for a rewrite was not available at this point). Other external tools used were the Navigation and Ancillary Information Facility (NAIF) SPICE Toolkit [12] and the ICY DLM. These tools were used for visualization and data manipulation. The SPICE Toolkit was heavily used in the eCARS system, as it was necessary for the scientists to have knowledge regarding the space environment when designing activity requests.

With all these major technologies, there was a need to connect them together. The "glue" that holds all these major technologies are the following technologies, software tools, or programming languages:

- Cron is used to schedule jobs, including queries, clean-up activities, file passing, data processing, and plot generation.
- Java™ Native Interface (JNI) is used to connect the Java code with C and C++ code, as well as the NAIF Spice toolkit.
- Unix™ shell scripting and Perl™ are used to tie all of the pieces together.
- Java™ Remote Method Invocation (RMI) was used to allow the web server to access the ION™ product. This particular use is an informal solution within the system and remains a part of the current system even though there are IDL™ technologies that allow this solution to be replaced. However, since it works smoothly, there is no reason to replace this solution.

The end result of all these technologies is the CAPS Ground system web site. The daily automated processes that it manages are as follows:

- Data query from servers at JPL.
- Transmission of the file to the data processing machine (as well as cleanup on the machine performing queries so it does not overfill the disk resources).
- Ingestion of the data packets into the database.
- Generation of alarms for instrument health and safety and automated notification sent via email to the operations team.
- Recombination of data from packets into instrument cycles, including data decompression and recognition of current products that are in the system.
- Post processing of data based on in-flight activities, including additional processing of actuator drift cycles.
- Build of the archive products.
- Build of quick-look spectrograms.
- Transfer of files to an ftp site for Co-Investigator access and an internal machine for scientists at Southwest Research Institute.
- Transfer of quick-look spectrograms to the web server.
- Logging for all processes.
- Incremental backups of critical systems.

Offline activities include the ability to look at spectrograms and housekeeping data since the start of operations in 1999, plot any housekeeping channel (or combination of channels) for a specified period of time, and to gain access to raw instrument data. Additionally, data from the Engineering Model can also be accessed through this system.

5. CAPS Uplink Ground System

As the project approached prime mission operations, it was decided to implement some automation on the uplink side, due to the complexity of integration of science data volume among all the instrument teams and internal mode changes. It was not necessary to have a graphical user interface for uplink operations, since there were already many files built (and archived) for performing many of the tasks that were routinely done. Also, the graphical user interface would have required a lot of up-front cost to implement via the SQL-database and to recode tools that were already built to work within the database environment. Additionally, there would be the necessity of building in a large set of commands that were already available via files that had been sent to the spacecraft previously. Integrated sequences could be generated easily by using blocks from current files via cut and paste at the command line. Additionally, there was a change in focus from placing commands into the background sequences to placing commanding within instrument expanded blocks (IEBs) inside the instrument.

After much discussion, it was decided to build tools in such a way that input and outputs from the tools were text based and allowed easy integration into the current uplink processing. The new tools were built in IDL™ and C. These tools could read in CAPS activities, spacecraft data volume, and the current spacecraft pointing kernel and output a rough cut at an IEB that would execute commanding for the 5 week sequence execution period. Final tweaks were done manually.

After building all commanding for the CAPS instrument for a 5-week sequence, the merged sequence file was handed off for final validation. Sequence validation is one of the key uplink requirements for teams. The requirement for validation was handled by a CAPS Simulation Tool that was built using Labview™. Upon ingesting a merged sequence containing CAPS commanding and the commanding that handles data volume, the simulation program runs and generates output files with the state of the different elements of CAPS, as well as data volume estimates, and any commanding errors. This tool allowed CAPS to validate a sequence prior to execution and has caught a few commanding errors prior to submission. Labview™ was used for this tool because there were already tools on the flight software side that were able to read CAPS commanding, IEBs, and the spacecraft file format. Extending this earlier work was simpler (and more cost effective) than starting from scratch in another programming language.

6. Other ground system Opportunities

After operating for a time with a single operations technical lead (OTL), a second OTL was hired and it became necessary to coordinate information between OTLs and processes. To facilitate this coordination, a mechanism for file sharing within the password protected ground system was created. In addition to the ability for the OTL's to share information, this also allowed CAPS scientists to view the sequence that was executing and what science was being integrated into upcoming sequences. This offered greater visibility into the CAPS science plan. These pages made use of current technologies that were already part of the system.

Another extension was the inclusion of meeting pages to keep track of presentations made by scientists at the CAPS team meetings. This facilitated faster file sharing, as it was simple for the files to be uploaded after they had been received at the meeting. This allowed rapid collaboration between scientists, and because the site was password protected, there was no concern about loss of science publication ideas to others outside of the CAPS team.

7. Hardware Used and Obsolescence

The hardware behind the CAPS ground system has served us well over the years. There have been very few failures within the system, and given the possibility of a second extended mission, hardware upgrades are in process. The following is a discussion of the current system, and what machines are being upgraded, and why.

The main CAPS ground system runs on a SunFire Ultra Sparc 4800 Server, with four processors and 32GB of memory with expandability to 16 processors and 96GB of memory. The system was purchased in 2003 and has been running nearly 99.8% of the time since its launch, with downtime only for maintenance. The database is currently housed on a RAID (redundant array of inexpensive disks) consisting of 26 IDE (Integrated Drive Electronics interface) drives running at 7200rpm (revolutions per minute). The connection between the RAID and the server is a SCSI (Small Computer System Interface) connection with a maximum speed of 320Mbps.

In addition to the main system, the CAPS system consists of three (3) other machines:

- A web server running on a Sun Ultra 450. This machine currently has 4 processors running at 250MHz.

- A SPICE Repository running desktop Sun Blade 2000 workstation. Given the other machines purchased, it was necessary to purchase this machine as a low entry point machine.

- An internal to Southwest Research Institute (SwRI) data repository and science processing machine, running on a Sunfire V440, with four processors and 32GB of memory. Additionally, a 2.5TB RAID was added to supply sufficient space.

There have been a few failures within the system over the past six years. The web server has bad memory that generates error messages and takes up space on the system with the logging of the errors. The memory has been swapped out a few times, but the memory problems seem to persist. The web server is scheduled to be replaced with a new high end Sun running at 1.8GHz with a 4 core processor.

The other system that failed was the RAID. The drive backplane in the chassis failed, which caused a catastrophic loss of all data on the RAID. The drive backplane was replaced and the RAID rebuilt. While the system was being rebuilt, the development system was drafted into service to allow the CAPS operations team to remain up-to-date with health and safety monitoring, and continued push of science to the instrument team.

Given that the Cassini mission is likely to be funded for a second extended mission to 2017, it was decided to upgrade a portion of the system. The RAID is being upgraded to a fiber-channel connection to the server with a maximum connection speed of 3Gbps, with SCSI disks running at 15,000rpm and a total useable space of between 12 and 15 TB. This configuration should increase the speed at which the system can run by at least a factor of 2 and potentially up to a factor of 10. The new RAID also triples the useable space.

As discussed earlier the web server is being upgraded due to memory problems and the age of the machine (parts are not easily found anymore). Additionally, the SPICE repository machine is being upgraded, mainly due to the space limitations on the machine but also due to speed increases and life expectancy. The internal SwRI data storage and science processing machine does not need to be upgraded as the original machine still has the speed and the space available for growth as needed for a second extended mission.

8. Informal Solutions and Tools that Failed

For all the technologies included in the ground system and the order in which they were included, there are amazingly few informal solutions. Most of the included technologies or tools were put in place as the best solution possible at the time. One of the informal solutions discussed earlier was in regards to the use of the Java™ RMI. ION™ interfaces with RMI to generate a solution for building images with IDL™ on one machine and displaying those images on another machine via a web-page. Although solution exists to avoid this clumsy handshaking between tools via the informal solution, the decision was made to accept the current solution. The decision was based on cost of the change versus the benefit of making the change. Given that the current solution is functional and that no additional benefit can be achieved by changing the solution, there was no need to spend money on a "cleaner" solution. The choice between keeping informal solutions and reworking functional code is always a trade-off between the benefit of making the change and the cost of leaving it alone.

There were not many failures of tools within the CAPS ground system. There was the false-start of choosing a file format that was too unwieldy to be handled by the scientists and operations personnel. This failure was corrected with help from the deputy-PI of operations in the form of well-formatted binary files and code to read and plot the contents.

Notably the largest failure within the CAPS ground system was the eCARS system. Although the technologies that built the eCARS system were sound, the tool never managed to be accepted by the end user. The goal was to get the external CAPS co-investigators to input their requests for science observations into the system. However, this required the co-investigators to spend the time to learn the system and to use it actively to generate requests. As it turns out, very few of the scientists were able to successfully use the system to build science requests. The biggest reason cited was a lack of time to interface with the tool to generate the desired activities. It was much simpler to have a discussion with the deputy PI of operations and to discuss the types of observations that the co-investigator would like to see implemented

9. Conclusions

Looking back to the ground system at the beginning of the Cassini mission, the system has grown from the manual uplink and downlink processing. The ground system is nearing full automation and the technologies, tools, and programs that are in place are working well together.

In looking back, two questions come to mind: "what would you do differently today, if you know what you do now", "and can you offer advice to the next generation of spacecraft instrument ground system developers"? Firstly, if ground system development could be started over, it would be important to have a better understanding of how the instrument worked. Actually, not only how the instrument worked, but also a better understanding of the telemetry and any complexities associated with partial data products that are an outgrowth of DSN and ground data processing problems. Secondly, writing all the important pieces in the same language would avoid some of the hardships of tying multiple language pieces together to generate one coordinated system.

Another point to take away from the ground system development on CAPS would be that even if a better understanding of the system was available up-front and all the tools were in a common language, changes in the flight instrument could still force informal solutions to be created within the ground system. This was true for the CAPS ground system, as the actuator monitor failed in flight, and a new version of flight software was uplinked to make the actuator function properly. The new flight software required that the position be adjusted for drift, as the actuator was being moved by counting steps. Due to downlink inconsistencies generating partial "home" periods, the calculation of drift has been difficult to incorporate and no straightforward, clean solution has been found.

The last piece of advice that can be offered is that although the ground system is a necessary part of the operation of any flight instrument, the ground system is not necessarily designed while the instrument is being built. Typically the ground system development starts after the instrument has been through calibration and sometimes while the instrument is being commissioned on the spacecraft. At this point, there are already many constraints on the ground system (both uplink and downlink) and the best that can be done is to build the ground system around the constraints. To build the most effective ground system, it would be best to start early with the design and be able to (potentially) influence flight software decisions that would mesh the design of the system with the science goals and instrument operation. This is a change from the standard mode of operation, but may be the wave of the future.

References

1 Pate-Cornell, E., and Dillon, R.: 'Challenges in the management of faster-better-cheaper space missions', 1998 IEEE Aerospace Conference, March 21-28, 1998.

2 Pate-Cornell, M.E., and Dillon, R.L.: ‘Success factors and future challenges in the management of faster-better-cheaper projects: lessons learned from NASA’, IEEE Transactions on Engineering Management, 48 (1), Feb 2001.

3 Mishkin, A., and Larsen, B.: ‘Implementing Distributed Operations: A Comparison of Two Deep Space Missions’, 2006 American Institute of Aeronautics and Astronautics (AIAA) Space Ops Conference, June 19-26, 2006.

4 Wick, J.V., Callas, J.L., Norris, J.S., Powell, M.W., and Vona, M.A., ‘Distributed Operations for the Mars Exploration Rover Mission with the science activity planner’, 2005 IEEE Aerospace Conference, March 5-12, 2005.

5 Lock, P., and Sarrel, M., ‘Distributed Operations for the Cassini/Huygens Mission’, 1998 AIAA Space Ops Conference, Tokyo, Japan, 1998.

6 Paczkowski, B.G., and Ray, T.L., ‘Cassini Science Planning Process’, 2004 AIAA Space Ops Conference, May 17-21, 2004.

7 Vandermey, N., and Paczkowski, B.G., ‘The Cassini-Huygens Mission Overview’, 2006 AIAA Space Ops Conference, June 19-24, 2006.

8 Paczkowski, B.G., Larson, B., and Ray, T., ‘Managing complexity to maximize science return: Science planning lessons learned from Cassini’, 2009 IEEE Aerospace Conference, March 7-14, 2009.

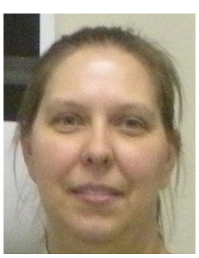
9 Barlow, V., ‘Cassini Project Acronymn List and Users Guide’, Jet Propulsion Laboratory document, 2008.

10 Planetary Data System Website: http://pds.jpl.nasa.gov/.

11 LaValle, D.B., and Turner, F.S.: ‘Automating Cassini instrument commanding for Saturn exploration’, 2005 IEEE Aerospace Conference, March 5-12, 2005.

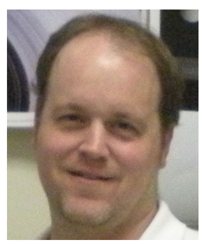
12 NAIF Website: http://naif.jpl.nasa.gov/naif/toolkit.html.

BIOGRAPHY

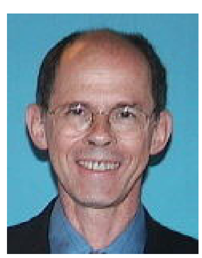


Judith D. Furman started work on the Cassini Plasma Spectrometer (CAPS) instrument that is part of the payload of the robotic spacecraft called Cassini-Huygens in August 1998. After working on CAPS data decompression for about a year, she was promoted to be the operations technical lead (OTL). Moving in to this position offered challenges not only associated with the responsibilities of learning about how to operate the CAPS instrument, but also in navigating the "in-flux" project processes for integrating with the spacecraft and other instruments. This involved heavy use of virtual teams, but also a lot of independent work from the remote sites. After operating the instrument for 10 years, Judy continues to take the lead role in instrument operations, as well as archiving, and general project management. She as a BS in Mathematics from the University of New Mexico and an MS in Mathematics from the New Mexico Institute of Mining and Technology. Additionally, she is working towards an MS is Management of Technology from the University of Texas at San Antonio.

Charles J. Zinsmeyer started work on the Cassini Plasma Spectrometer (CAPS) instrument that is part of the payload of the robotic spacecraft called Cassini-Huygens in December 1994 as the lead flight software engineer. Mr. Zinsmeyer is responsible for the development of the flight software for the main CAPS processor. Mr. Zinsmeyer has been involved in the development of both scientific instruments and spacecraft avionics for several NASA missions including Cassini-Huygens, Deep Impact and IMAGE. Mr. Zinsmeyer has an MS in Computer Information Systems from St. Mary's University in San Antonio and is continuing his graduate studies in Computer Science at the University of Texas at San Antonio.



Gregory D. Farris is a Software Developer in the Department of Space Science at Southwest Research Institute in San Antonio, Texas. He is currently the Lead Programmer for the Cassini Plasma Spectrometer Ground System (CGSS), a web-database system for the CAPS instrument on the NASA Cassini project. Recently, he has also developed components for several other web-database systems for the NASA Cassini project, including the Cassini Information Management System (CIMS) and Ion and Neutral Mass Spectrometer Operations Network (ION). He currently functions as the Lead Programmer for the Cassini CAPS Ground System. He is also pursuing a Master's degree in Computer Science from the University of Texas in San Antonio.



David T. Young's expertise is centered on understanding the chemical composition of solar system magnetospheres, ionospheres, and atmospheres, and on developing a wide range of mass spectrometers that enable their measurement. His interests also include the spinoff of space instrumentation to ground-based applications. Over the past 38 years Dr. Young has contributed to the design and development of thirteen different instruments (seventeen total) on eleven NASA and ESA science missions. He is currently the Principal Investigator for the Cassini Plasma Spectrometer on NASA's Cassini spacecraft, in orbit about Saturn, and is also Principal Investigator for the Hot Plasma Composition Analyzer being developed for Magnetospheric Multiscale, a four-spacecraft mission scheduled for launch in 2014.

Prachet S. Mokashi is a Senior Research Engineer at Southwest Research Institute in San Antonio, TX. He has been involved with space science instrument development, operations and data analysis since 2002. He is the operations lead for the Cassini Plasma Spectrometer (CAPS) on Cassini and the Ion and Electron Sensor (IES) on Rosetta as well as a science planning lead for the Juno mission. He plans and coordinates instrument activities with flight software engineers, instrument scientists and the mission operations teams. Additionally, he is involved in analyzing plasma science data and generation of mission and instrument science support data.

ACKNOWLEDGMENTS

This work is sponsored by NASA contract number NAS703001TONMO710023.