CUSTOMIZING THE JPL MULTIMISSION GROUND DATA SYSTEM: LESSONS LEARNED

SUSAN C. MURPHY
JOHN J. LOUIE
ANA MARIA GUERRERO
DANIEL HURLEY
DANA FLORA-ADAMS

Operation Engineering Lab
Jet Propulsion Laboratory
California Institute of Technology
MS 301-345
Pasadena, California 91109-8099

ABSTRACT

The Multimission Ground Data System (MGDS) at NASA's Jet Propulsion Laboratory has brought improvements and new technologies to mission operations. It was designed as a generic data system to meet the needs of multiple missions and avoid re-inventing capabilities for each new mission and thus reduce costs. It is based on adaptable tools that can be customized to support different missions and operations scenarios. The MGDS is based on a distributed client/server architecture, with powerful Unix workstations, incorporating standards and open system architectures. The distributed architecture allows remote operations and user science data exchange, while also providing capabilities for centralized ground system monitor and control. The MGDS has proved its capabilities in supporting multiple large-class missions simultaneously, including the Voyager, Galileo, Magellan, Ulysses, and Mars Observer missions.

The Operations Engineering Lab (OEL) at JPL has been leading Customer Adaptation Training (CAT) teams for adapting and customizing MGDS for the various operations and engineering teams. These CAT teams have typically consisted of only a few engineers who are familiar with operations and with the MGDS software and architecture. Our experience has provided a unique opportunity to work directly with the spacecraft and instrument operations teams and understand their requirements and how

the MGDS can be adapted and customized to minimize their operations costs. As part of this work, we have developed workstation configurations, automation tools, and integrated user interfaces at minimal cost that have significantly improved productivity. We have also proved that these customized data systems are most successful if they are focused on the people and the tasks they perform and if they are based upon user confidence in the development team resulting from daily interactions.

This paper will describe lessons learned in adapting JPL's MGDS to fly the Voyager, Galileo, and Mars Observer missions. We will explain how powerful, existing ground data systems can be adapted and packaged in a cost effective way for operations of small and large planetary missions. We will also describe how the MGDS was adapted to support operations within the Galileo Spacecraft Testbed. The Galileo testbed provided a unique opportunity to adapt MGDS to support command and control operations for a small autonomous operations team of a handful of engineers flying the Galileo Spacecraft flight system model.

INTRODUCTION

The Multimission Ground Data System (MGDS) at NASA's Jet Propulsion Laboratory has brought improvements and new technologies to mission operations. The development of a generic data system to meet the needs of multiple missions was intended

to avoid re-inventing capabilities for each new mission and thus reduce costs. The traditional mainframe-based data systems of the past were expensive to modify and their proprietary architectures did not facilitate incorporation of new technologies. The MGDS is based on a distributed client/server architecture, with powerful UNIX workstations, incorporating standards and open system architectures.

The MGDS system provides a mature, relatively stable set of software for real-time command and control operations and for off-line engineering analysis. The system is based on a table-driven approach with simple user-oriented languages for specifying processing and display functions that allows the addition of new missions without extensive reprogramming. The standard Sun/HP/UNIX end-user workstations are part of a distributed operations system that places a powerful, flexible, and extensible set of operational capabilities at an analyst's fingertips. When properly configured, these workstations greatly increase the efficiency of spacecraft operations.

ADAPTABLE SYSTEMS

The Multimission Operations System Office's (MOSO) understanding of the MGDS design was that multimission capabilities would be delivered to allow the users to customize, adapt, and tailor the system for their individual use. MOSO was responsible for developing, installing, and maintaining the multimission hardware and software for the operations teams, but customizing its multimission software was up to the project. However, the system has become so powerful with over 1.5 million lines of code that its 'configurability' and 'extensibility' can potentially overwhelm users rather than benefit them. The MGDS user guides currently stand over one foot high on end. In addition, the users don't often refer to the user's guides because they don't want to know how to use a tool, they want to know how to accomplish their operations task within the MGDS environment.

The MGDS Workstation Training Group had been frustrated for several years trying to train users on workstations which bore little resemblance to the configuration the users would find in their operations environment. Often, there was no standard project configuration in the end-user environment and users were on their own to transform their blank screens into a mission operations system. Each user worked individually and project-specific files needed for telemetry processing and display were passed in an ad-hoc manner among team members. However, how well a system is tailored for end users is often the most important factor in determining the degree of system operability and efficiency improvements that come from new technologies.

It has become clear that the MGDS system and its documentation cannot simply be delivered to a project for them to adapt for their needs. Adapting the MGDS software has become a complex task with a high learning curve. This makes adaptation an expensive task for individual projects, especially since operators within the same project will have different needs and interfaces with the system. The adaptation of MGDS for a power subsystem engineer may benefit more from knowing how a power subsystem engineer on another project customized the multimission system rather than how an instrument engineer on the same project would do it. Thus, the learning curve can be made cost-effective if it can be re-applied to several projects, with an adaptation team supporting multiple missions simultaneously. As an additional benefit, a multimission adaptation team will bring knowledge and improvement ideas to bear on future development and customization of MGDS for new projects.

OPERATIONS ENGINEERING LAB

Automation and advanced user interfaces can help reduce costs only if they are focused on the people and the tasks they perform. New technologies may only bring minimal cost savings if the new system functions much like the old one. This often happens since the users who write the requirements aren't always familiar with the capabilities of new technologies and simply use their existing

system as a model. For example, the JPL mission controllers asked for a scrolling screen that displayed telemetry values representing the latest value of the spacecraft clock. This was the way the old system allowed them to determine whether there were any data outages. The developers gave them their scrolling display and operators continued to stare at these displays watching for outages. An important opportunity was lost to automate this process and improve the efficiency of operations.

To solve these types of communications problems, the Operations Engineering Lab (OEL) was created four years ago to merge operations and development activities for the Space Flight Operations Section. The OEL builds scheduling, command, control, and analysis software and currently delivers over 500,000 lines of code. The development philosophy is characterized by iterative development with active participation of the end-users. Our approach has been successful because we involve users and trainers throughout development, focus on automating essential, time-consuming operations tasks, and get implementations in the hands of users early. We also have operators work in the OEL and developers work in operations in order to maintain close contact with our users and understand the problems that need to be solved. By working closely with users, we have learned how to use new technology to change the way they do business, not just automate the old way of doing business. For example, we have built a smart alarm tool to automatically perform the data outage task described earlier and improved mission controller efficiency by over 30%.

CUSTOMER ADAPTATION TEAM (CAT)

At the request of the trainers and project teams, the OEL developers began to work closely with mission controllers and spacecraft engineers to adapt and configure the workstation and MGDS software to meet the individual user needs. The project configurations were then transferred to the trainer workstations to allow more meaningful training. This adaptation task, started as a grass-roots effort, has evolved

into a more formal Customer Adaptation Team (CAT). A small team of OEL developers and operators have supported the adaptation of MGDS for the Voyager, Mars Observer, and Galileo Spacecraft and Instrument Operations Teams. The OEL CAT provides direct project support in developing workstation configurations, customized processing and display tables, automation and analysis tools, and a common user interface for the project.

The workstation configuration and user interface is designed to provide an integrated system view from which a project team can operate a mission. The approach was to provide the flexibility for both advanced and novice operators to run the system to meet their individual needs without their having to know how to integrate across multiple tools and interfaces. We knew that different operators would use the system in unique ways. For example, 24-hour mission controllers want a system that is oriented to an analyst monitoring real-time data, working interactively at their workstation. On the other hand, the spacecraft engineers seldom need to view real-time data. They typically want hard copy plots and tabular printouts of telemetry parameters available overnight.

When the CAT team first started customizing the ground system for the spacecraft team, it became obvious that the system design forced the user to learn many tools and software interfaces to perform their analysis task. For example, to plot telemetry data, they had to use database query tools to retrieve their telemetry files, process the data through the telemetry processing software, export a processed telemetry parameter file and import it into a graphical plotting tool, set the axis correctly, and print the hard copy plot. The operator needed a single, integrated user interface that minimized operator interaction with the workstation and allowed each subsystem engineer to automate their analysis tasks.

The CAT team built a non-real-time telemetry toolkit and user interface that integrated the existing generic tools in the MGDS. The interface design was based on providing

graphical and command-line interfaces that freed the users from knowing the intricacies of querying, retrieving, accessing, and processing telemetry parameters and eliminated the need to know the intermediate file interfaces across various tools. With one simple command line, a user can ask to plot a telemetry parameter for a given time period without any knowledge of the tools needed to perform that task. Command line interfaces are especially important for users who prefer to have their processing done off-line. Graphical interfaces are provided for users who prefer interactive tools. The spacecraft engineers would set up overnight queries that would produce plots automatically for their review when they arrived each morning. The cost to implement this system was minimal since it was built on top of existing multimission capabilities. The interface was built using a GUI-building tool (OELSHELL) and a powerful scripting language (PERL) developed at JPL. There are no licensing costs and no compilation of C code is required for the graphical or command line interfaces.

A new MGDS subsystem was designed by the OEL to deliver these types of end-user tools and interface shells. It provides tools to fill in the gaps in missing capabilities that are discovered after MGDS is delivered, including project-specific adaptations and unique processing requirements. As a result, it is a subsystem that is continually evolving and has grown to be one of the largest MGDS subsystems.

This effort has been very successful because the CAT team works in the operator's own environment, configuring the workstations on their desks, building scripts to automate their tasks, and designing interfaces to integrate and organize the many software tools. In addition, OEL developers do not have the significant learning curve facing analysts getting familiar with the use of workstations, Unix, and MGDS software. We also provide hot-line and on-site support services for end-users, emphasizing quick response time in order to meet the real-time operations needs. Our multimission experiences mean the lessons learned from one project will be transferred to benefit

another. Also, if we find a missing capability in the system, we know who to contact to modify existing software or we will build and deliver a new MGDS capability ourselves.

LESSONS LEARNED

We have learned many lessons in adapting and customizing the MGDS system for end-users. The coordination between the OEL, the mission operations engineers, training personnel, and system administrators greatly improved system operability for the users. The following are some lessons learned in our adaptation activities.

Distributed systems are essential to provide the flexibility needed for incorporating new technologies and capabilities required for missions of the future. Compared to the mainframe-based centralized systems of the past, the distributed nature of modern systems require a more disciplined approach to configuration procedures to ensure consistency among all system nodes. End-users have control of their own workstations and can easily modify processing and display parameters. However, this flexibility can cause traditionally-structured organizations to adopt strong, centralized configuration management tools and procedures to prevent any potential problems. Often this leads to software deliveries that are monolithic, irreversible installations with too much bureaucratic overhead involved in making even small changes. The software delivery process needs to be amenable to simple improvements and fixes in non-critical software. Configuration control of end-user tables, scripts, configuration files, and simple tools for specific project use need to be handled separately from the core system. It must be flexible and be controlled by the operations teams.

The MGDS design recognizes that every user has a unique need and the system should allow for individual customization of tools. However, there was a lack of management understanding of the need to staff a CAT team for the extensive work required to customize a distributed system for users. Initially, there was no official follow-on support after a system was delivered to

operations and hence MGDS operability was rated poor by users. After the CAT team work began, the users' perception of operability was dramatically improved even though the core system was unchanged. We are viewed by projects as the group that makes the MGDS system work for users. There are big payoffs in providing project-specific customization, tools, and interfaces, supplemented with on-site support. Distributed systems require extensive customization to meet the specific needs of users and this should not be left to the device of each individual user or project.

Once a system is customized and automated for the end user, the usage of the system can significantly increase. For example, because we had made the off-line telemetry query and analysis process so easy, a much greater number of operators than originally estimated began to use the system extensively. This created serious network loading and disk storage problems.

Automation must be focused on changing the way we fly spacecraft, not just automating the old way of doing business. The greatest cost reductions can be realized if more attention is paid to the operators and the tasks they perform in order to eliminate tedious, labor-intensive processes and to assist in improving the reliability of critical tasks.

The users also want training geared to their work in the operations environment. The trainers need to know how the users might actually use the system in operations. In addition to providing standardized configurations on training and on project operations workstations, the CAT team developed specialized follow-on training classes focused on the project-specific configuration and use of MGDS capabilities.

CONCLUSION

JPL's Multimission Ground Data System has provided a powerful, adaptable and extensible set of operational capabilities at an analyst's fingertips. With more emphasis on a Multimission Customer Adaptation Team providing integrated systems with customized configurations and interfaces, success has been shown in improving system operability and reducing cost in operations for individual projects.

ACKNOWLEDGMENTS