Mars Polar Lander Mission Distributed Operations

Paul G. Backes*, Kam S. Tso†, Jeffrey S. Norris*, Gregory K. Tharp†,
Jeffrey T. Slostad* and Robert G. Bonitz*
*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California
†IA Tech Inc., Los Angeles, California

Abstract— The Mars Polar Lander (MPL) mission was the first planetary mission to use Internet-based distributed ground operations where scientists and engineers collaborate in daily mission operations from multiple geographically distributed locations via the Internet. This paper describes the operations system, the Web Interface for Telescience (WITS), which was used by the MPL mission for Internet-based operations. The MPL mission was a lander which landed near the south pole of Mars on December 3, 1999, although communication with the lander was never achieved. WITS was used for generating command sequences for the lander's robotic arm and robotic arm camera, and as a secondary tool for sequence generation for the stereo camera on the lander. WITS was also used as a public outreach tool.

TABLE OF CONTENTS

1. INTRODUCTION

The Mars Polar Lander (MPL) landed near the south pole of Mars on December 3, 1999 and was to perform an approximately three month mission [1]. Unfortunately, communication with the lander was never achieved so commanding the lander was not possible. An artist's drawing of the lander is shown in Figure 1. The lander carried the Mars Volatiles and Climate Surveyor (MVACS) instrument suite, lead by scientists at the University of California at Los Angeles (UCLA). The mission operations center was at UCLA. The mission included searching for near-surface ice and possible surficial records of cyclic climate change, and characterizing physical processes key to the seasonal cycles of water, carbon dioxide and dust on Mars.

The MPL mission was the first NASA planetary mission which utilized Internet-based ground operations to enable geographically distributed scientists and engineers to collaborate in daily mission command sequence generation. Internet-based distributed operations for the Mars Polar Lander mission was done using the Web Interface for Telescience (WITS). This paper describes the ground operations for the MPL mission using WITS, the implementation of WITS, and results of operations using WITS. The results are from commanding the lander in the UCLA testbed.

WITS served multiple purposes for the MPL mission. It was the primary operations tool for visualization of downlink data and generation of command sequences for the robotic arm (RA) and robotic arm camera (RAC). It was also used as a secondary tool for command sequence generation for the Stereo Surface Imager (SSI) stereo camera which is mounted on a lander mast, e.g., for visualizing footprints on the surface where SSI images would be taken. WITS also enabled Internet-based users to generate command sequences for the RA, RAC, and SSI. For example, scientists at the University of Arizona, who were responsible for the RAC and SSI, were able to generate sequence inputs from Arizona so that they would not have to be at the UCLA operations center for the whole mission. Internet-based operations also enabled WITS support engineers at JPL to provide mission support without having to travel to UCLA. This reduced operations costs by requiring fewer support personnel at UCLA. Also, it enabled faster resolution of problems by allowing remote JPL support engineers to work on problems immediately without having to travel to UCLA (an hour or more delay) to address problems. Public outreach is another important application of WITS for the MPL mission. A separate WITS system was provided to the general public for them to download to their home computers to enable them to plan and simulate their own missions.

WITS was originally developed in NASA research programs for use in future rover missions [2], [3], [4]. WITS has been reimplemented based upon rover field test experiences and mission requirements for use in flight missions. The Mars Polar Lander mission was the first flight mission use of WITS. Other examples of Internet-based robot operation can be found in [5], [6].

2. SYSTEM ARCHITECTURE

WITS is a part of the complete MPL mission ground operations system. A simplified diagram of the MPL ground operations system is shown in Figure 2. Downlink data from Mars is processed and put in databases. One of the databases is the WITS database. Sequence generation begins using a sequencing tool called APGEN which generates daily high level sequences for all the lander instruments. The sequences for the different instruments are sent to sequence generation systems specific to each instrument. Included in the sequences are requests which include text descriptions of what needs to be done in each request and how much time, energy, and data volume is allocated for each request. WITS is the sequence

[1]0-7803-5846-5/00/$10.00 © 2000 IEEE

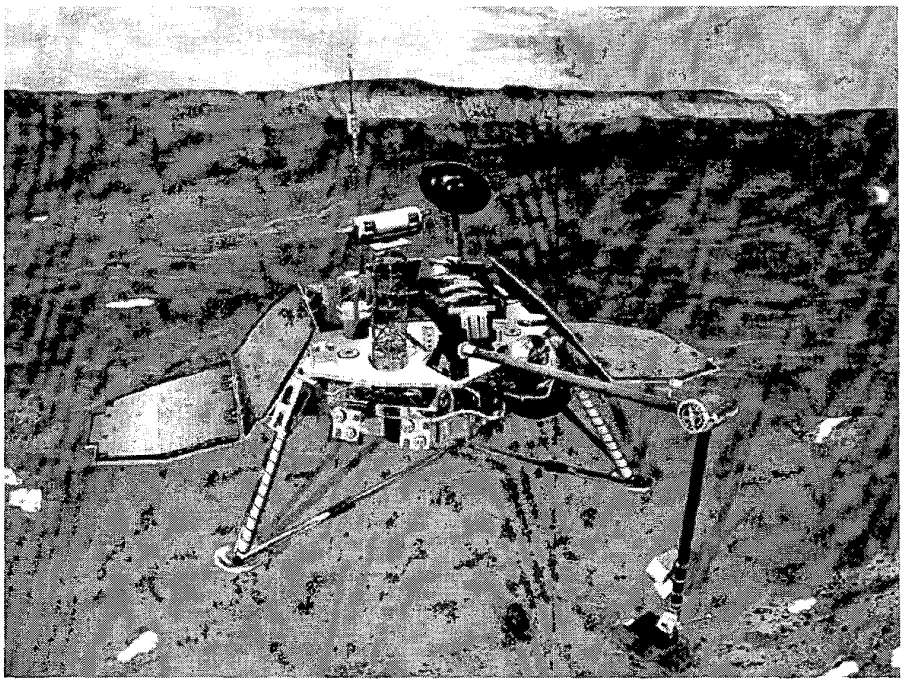


Figure 1. Artist's Drawing of Mars Polar Lander on Mars

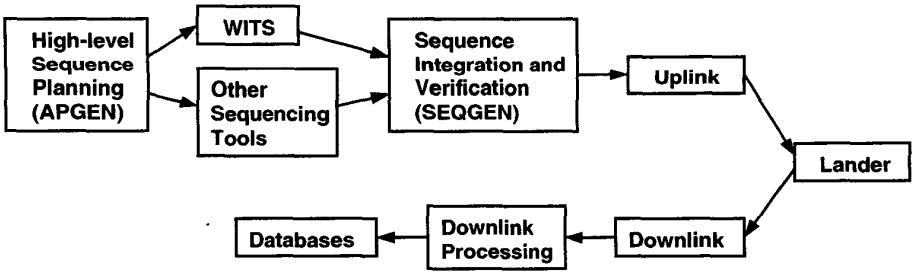


Figure 2. MPL Mission Operations Architecture

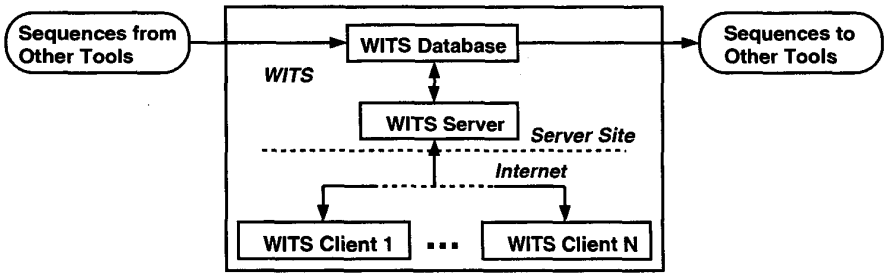


Figure 3. WITS Architecture

generation system for the RA and RAC and some SSI operations. WITS generates the low-level commands to achieve the goals specified in the requests and within the resource allocations specified. The multiple sequencing systems then output their sequences to the SEQGEN planning tool where all the low-level sequences are integrated and resource checking on the integrated sequence is done and final sequence modifications are made to ensure a valid sequence within resource constraints. The final sequence is then sent into the uplink process, eventually to be received at the lander.

The WITS architecture is shown in Figure 3. The WITS Database holds downlink data products and uplink sequence information. The WITS Server provides communication between the WITS Database and the WITS Clients. The WITS Clients are distributed over the Internet and provide the interface to the user to view downlink data and generate command sequences. Sequences from other sequencing tools, e.g., APGEN, are placed in the WITS Database for access by the WITS Clients, and sequences completed by WITS are placed in the WITS Database where they are retrieved by other sequencing tools, e.g., SEQGEN. It would be possible to have other sequencing tools communicate with the WITS Server to send and receive sequences between the sequencing tools, but for this mission it was determined that the most efficient approach is to place sequences in the database for other tools to copy.

3. Internet Security

A critical element in Internet-based mission operations is Internet security. Each day, a large amount of data is received from the spacecraft at the mission operations center and placed into a local database for processing and viewing by mission scientists. To enable collaboration in daily sequence generation by distant, Internet-based, scientists, a secure and efficient way to deliver the data to the remote scientists is needed. Individual scientists may want, or have authorization for, only a specific subset of the downlink data. Also, secure Internet communication is needed to receive inputs, e.g., command sequences, from the Internet-based users. WEDDS, the WITS Encrypted Data Delivery System, was created to provide the required secure Internet-based communication for WITS.

WEDDS is a framework for automatically distributing mission data and receiving remote user transmissions over the Internet in a secure and efficient manner. WEDDS was integrated with WITS for the Mars Polar Lander mission, but is designed to work with any mission application with little modification. WEDDS operates in a fashion that is transparent to the remote user. Files simply appear on the remote user's machine as they become available, and connections are made securely without any additional effort on the part of the user.

WEDDS utilizes a comprehensive approach to Internet security to confirm the identity of its users and encrypt transmissions to and from the mission control center. All WEDDS connections are authenticated using the NASA Public Key Infrastructure (PKI). NASA Ames is currently overseeing the installation of the NASA PKI at all NASA centers [7]. After authentication, all WEDDS communications are made through SSL (Secure Sockets Layer) sockets and are encrypted using the Triple-DES-EDE3 algorithm [8], [9]. Data transferred by WEDDS is nearly impossible to decipher if intercepted, and fooling the authentication protocol would require compromising the NASA Ames Certificate Authority, which is highly protected and forms the backbone of NASA's future approach to security. A copy of the WEDDS client software cannot be activated unless the remote user provides his personal security profile to the system, which is kept on a floppy disk in the possession of the user. Further details on the WEDDS system can be found in [10].

4. Downlink Data Visualization

Downlink data from the lander is provided to the user via various views. Two windows provide the user with available data to be visualized. The Results Tree window (not shown in the figure) displays the available downlink data for the mission by date of downlink. It also has lists of descent imagery. The Plan window displays available Panorama, Overhead, and 3D views for a specific plan. This is usually the most convenient way for a user to specify a view to be opened since the desired views usually have data which are organized for the specific plan. Each of these specific views has a specific set of downlink data it uses, so definitions of these views may be updated each day and put in the new plan. The Results Tree and Plan windows have data displayed in a tree structure whose nodes can be expanded or collapsed. The user opens a view to visualize downlink data by clicking on the item. The various types of views are described below.

The Descent view (not shown in the figure) provides images taken from the spacecraft during descent to the surface and shows the landing location. The Overhead view, shown in Figure 4, shows the area around the lander from above. An optional grid provides angle and range information. Blank (shown in the figure), color-coded elevation map and texture mapped image options are provided. Targets are displayed in the Overhead view, as well as selected points and view objects (targets view objects are described in Section 5). Clicking on a point in the Overhead view causes the x,y position to be displayed at the clicked location. Clicking and dragging causes a ruler to be displayed with the start and end points and distance displayed.

The Panorama view, shown in Figure 4, is a mosaic of images taken by a stereo camera. Selecting a point in an image causes the x,y,z position on the surface to be displayed as well as the ARZ (azimuth angle, range from site center, and z elevation value). The point can be turned into a science target via the Main window Action pull-down menu. The Panorama view can be shown in 1/4, 1/2, and full scale. The view can also be shown in anaglyph stereo via a menu option. Clicking and dragging causes a ruler to be displayed with the start and end points x,y,z values and the distance and azimuth between the points.

The Wedge view displays one image with various options, e.g., left or right image from stereo image pair, anaglyph stereo, and resize. When a user selects a pixel in a Wedge or Panorama image, WITS determines the corresponding x,y,z

Figure 4. Panorama, Wedge, and Overhead views and Sequence and Plan windows

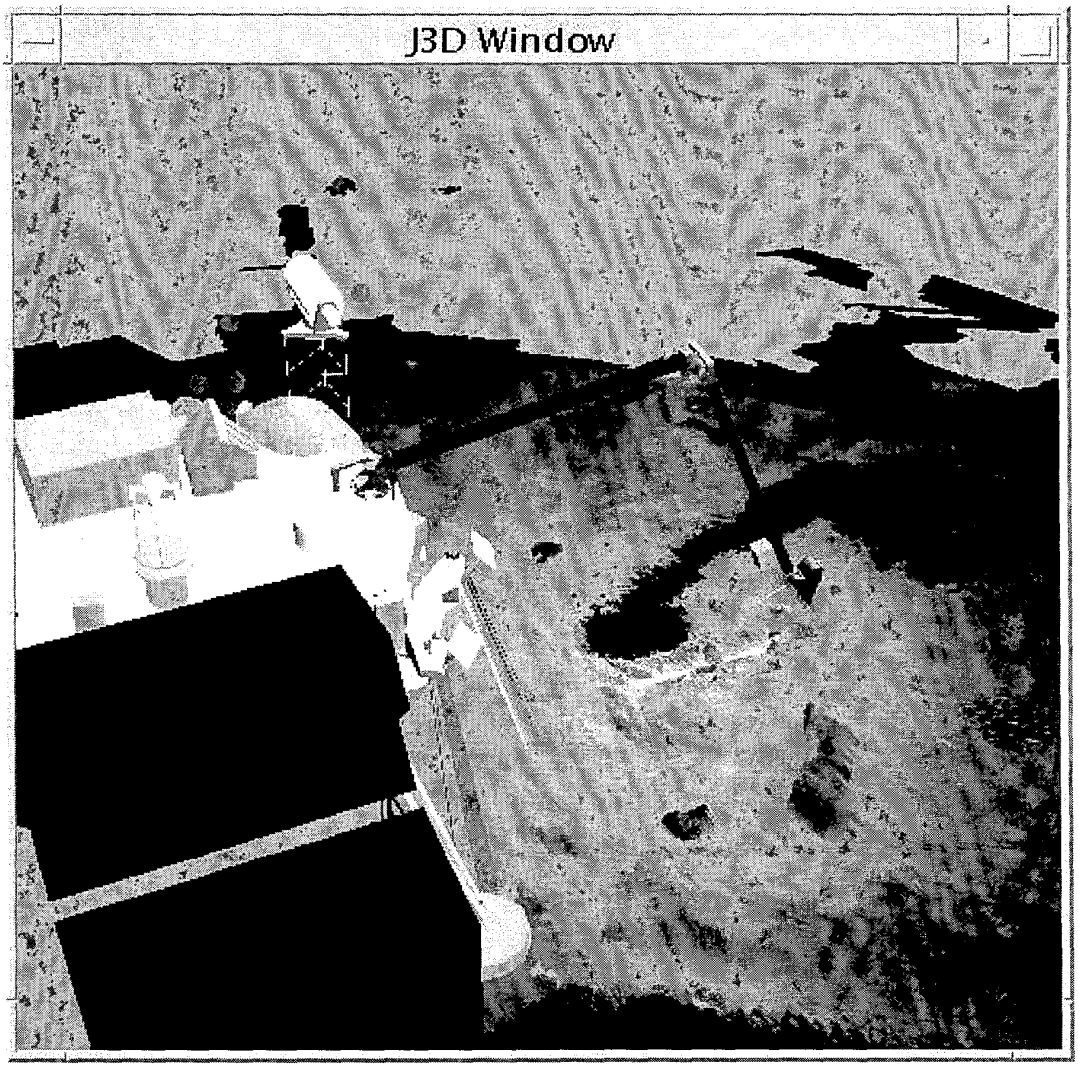


Figure 5. 3D view with lander and terrain visualization

position and surface normal on the terrain and displays the x,y,z position to the user in all views in which this point is visible. This point can be turned into a target, as described below in Section 5. In the Wedge view, the pixel intensity is also displayed.

The Contrast Adjuster view (opened from a Wedge view pull-down menu), shown in Figure 6, enables the contrast to be adjusted for a Wedge view image. The minimum and maximum desired pixel intensities are selected via scroll bars and then the pixel intensity values of the image are linearly stretched to have the selected pixel intensities become minimum (0) and maximum (255). The histogram of the initial and adjusted images are also shown.

The 3D view, shown in Figure 5, provides a 3D solid model visualization of the lander and terrain. Sequence simulation and state is visualized in the 3D view. The Control window (not shown) provides slide bars to set the arm and SSI degrees of freedom directly, as well as selections for setting other visualization parameters such as viewpoint.

5. Sequence Generation

The views discussed above provide a means for visualizing downlink mission data. WITS also provides various windows and features for command sequence generation. WITS enables 3D locations to be used as parameters in commands. As described in the Wedge view description above, a user specifies a 3D point by selecting a pixel in an image in either the Wedge or Panorama views and WITS determines the 3D coordinates and displays them at the selected point. The point can be turned into a target by selecting "Add Target" in the Main window Action pull-down menu. Targets are displayed in the various views as pink circles. WITS keeps track of all specified targets and provides them as possible input parameters to sequence commands.

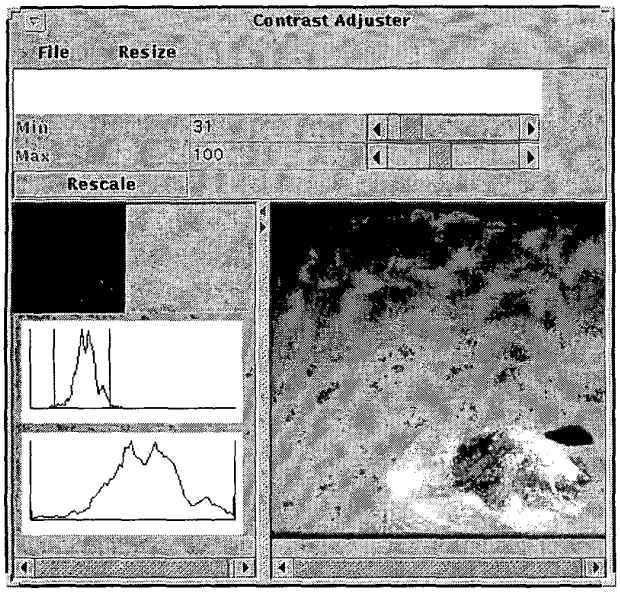


Figure 6. Contrast Adjuster view

The Sequence window, shown in Figure 4, is used to generate a command sequence. A command sequence has a hierarchy of elements. The hierarchy, in descending order, is: Sequence, Waypoint, Request, Macro, Step. There can be any number of elements at a lower level of the hierarchy, e.g., there can be any number of macros in a request. There is only one waypoint for a lander mission, the landing site (the waypoint level is added to enable WITS to support future rover missions as well). A request represents a high-level task. A macro, described in more detail below, is the functional element in WITS by which the user specifies commands and parameters. Macros have expansions into steps. A step is a low-level command that will be uplinked to the spacecraft. WITS can generate various format output sequences. For the flight mission, WITS outputs sequences in the Spacecraft Activity Sequence File (SASF) format (a proprietary format).

The Sequence window shows the sequences in one plan. Multiple sequences can be displayed. A plan generally represents the planning elements to generate one command sequence to be uplinked to the lander. The sequences are shown on the right hand side of the Sequence window. Supporting multiple sequences is useful for integration of subsequences from different scientists or subsequences for different instruments into the final uplink sequence.

A list of macros which can be inserted into a sequence is shown on the left side of the Sequence window. Multiple lists of macros are available; choosing between macro lists is done via the pull-down menu above the macro list. A macro is inserted into a sequence by selecting the location in the sequence for it to be inserted and then double clicking on the macro in the macro list. Double clicking on a macro in the sequence causes the Macro window to pop up. Figure 7 shows the Macro window for the rac_mosaic_target macro. The structure of the macro window is the same for all macros. At the top is a description area. The parameter names, values, and means for specifying their values is in the middle, and

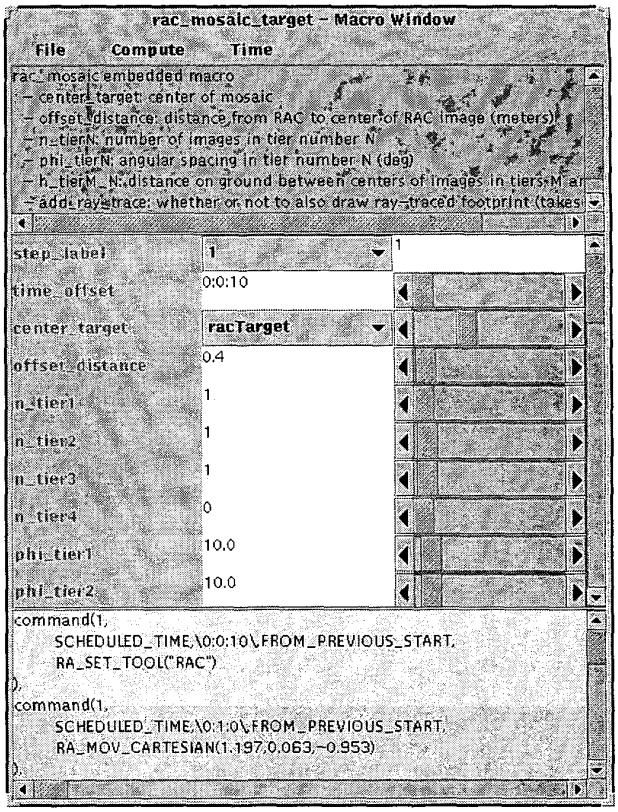


Figure 7. Macro window

the macro expansion into steps is at the bottom. Depending on the type of parameter, text areas, pull-down menus, and slidebars are provided for specifying the parameter values. A macro-specific algorithm converts the parameters into the expansion which can have any number of steps.

A macro can generate view objects which are displayed in the views to indicate what the macro is producing. Figure 4 shows square outlines which are view objects for SSI imaging commands. They represent where images will be taken by the SSI in the current sequence. Above the dump pile is a view object of a RAC image and above the trench is a view object of a three-image RAC mosaic. View objects can also be generated by parsing the steps directly.

There are various sequence editing features in the Sequence window, e.g., cut, copy, paste, and delete in the Action pull-down menu. Additionally, the user can click and drag an item in the sequence to another position in the sequence, e.g., the user can click on a macro and drag it into a different request. A user can drag all the macros from one request into another request as a block of macros.

A valuable feature of the Sequence window is sequence state visualization. When the user clicks on a step in the sequence, then the SSI and Robotic Arm in the 3D view are updated to their states at the end of the selected step.

The WITS Sequence Execution window allows the user to simulate a sequence. The sequence simulation is visualized in

the 3D view. The whole sequence can be simulated, or just individual commands. Simulation-specific macros are provided which can be inserted into a sequence, but are not included in the exported sequence. The SIM_SET_VIEWPOINT macro allows the viewpoint to be specified at a specific point in the simulation. The SIM_SET_VIEWCONE macro allows a viewcone to be turned on or off. A viewcone is a translucent projection from a camera showing its field of view.

Resource analysis and rules checking are important elements of sequence generation. WITS provides resource analysis for a sequence. The duration, energy, and data volume for each step of the sequence are computed and stored along with the cumulative duration, energy and data volume at each step. When a user clicks on a step in the Sequence window, then information about that step is displayed in the text area at the top of the window, e.g., resources for that step and cumulative resources through that step and absolute execution time. Rules checking is important to ensure that a sequence is valid relative to specified sequence rules. An example of MPL mission sequence rules that WITS checks is verification that request resources used are within allocations.

Ephemeris information is provided in WITS to enable simulating the sun's position and generating commands to image the sun or moons of Mars. Ephemeris information is information about the position of celestial bodies. The ssi_sun_mosaic macro generates a subsequence to take a mosaic of SSI images of the sun at a specified time. The sun position is shown in the Panorama view and 3D view and updated at each sequence step during simulation. The sun position is shown relative to the SSI at a distance that is specified with a macro.

6. Distributed Collaboration

Internet-based collaboration is achieved in WITS by providing Internet-based users with daily downlink data and allowing them to specify targets and generate command sequences and save them to the common server. The Internet-based users can see each others targets and sequences and can use target inputs from each-other's sequences. The Sequence window File pull-down menu enables sequences to be loaded from the common server (e.g., located at UCLA), or to be saved to the common server. Users can also save and load sequences to their local computers. One of the uses of distributed operations for the MPL mission was to to have scientists at the University of Arizona generate SSI and RAC imaging sequences and submit them to the UCLA operations site.

7. Public Outreach

An important motivation for the development and use of WITS in a planetary mission is its use in public outreach. During the mission a separate version of WITS was made available to the general public to download and run on their home computers. A subset of mission data was placed in the public WITS database. Since the actual mission commands and flight software are proprietary and could not be used in the public version of WITS, new arm kinematics were written for the public version. Also, new commands were used in the public version. The public version kinematics and command set were functionally equivalent to the flight versions. The public is then be able to view mission data and plan and simulate their own missions. The goal is to provide the public with an engaging mission experience where they use the same tool as the mission scientists to view mission data and generate and simulate their own missions. The site to download the public MPL mission WITS can be found at URL http://robotics.jpl.nasa.gov/tasks/wits/. The public outreach site is hosted at Graham Technology Solutions, Inc.

8. Implementation

The client WITS system which is used by the Internet-based users was implemented using the Java2 and Java3D programming environments. The system is run either as a Java applet using the appletviewer or as a Java application. Users must first download the Java Runtime Environment and Java3D. Communication between the client and server is implemented using Remote Method Invocation (RMI). The WITS database is a structured file system.

9. Example Sequence

An example sequence is shown in the Sequence window of Figure 4. The sequence shows some of the types of macros and steps which were used by the robotic arm operators. A real sequence has many more low-level engineering commands. One request is shown which has many macros. The ssi_mosaic_target macro expands into four steps. The first step moves the simulated SSI in the 3D view to the central target and the next three steps are the actual commands that will be uplinked to the lander. The first rac_mosaic_target macro takes an image of the dump pile. It expands into three steps: it sets the RA tool definition to the RAC tool, it then moves the arm to point the RAC at the dump pile, and then it takes the image. The M_RA_DIG_TRENCH macro expands into the step which is uplinked to the lander to dig a trench with the specified parameters. The M_RA_ACQUIRE_SAMPL macro expands into the step which is used to acquire a sample in the scoop for depositing on the Thermal and Evolved Gas Analyser (TEGA) instrument on the lander deck. The second rac_mosaic_target macro takes three images of the trench. At this point in the sequence, the sample might be dumped in the TEGA.

10. Conclusions

WITS provides a new Internet-based operations paradigm for planetary mission operations. WITS was used in the Mars Polar Lander mission ground operations for downlink data visualization and command sequence generation for the robotic arm and robotic arm camera and as a secondary tool for the stereo surface imager. With the use of WITS, the MPL mission was the first planetary mission to utilize Internet-based ground operations. The integrated visualization, sequence planning, and Internet security features of WITS make it a valuable tool for planetary mission operations. Also, WITS is an engaging public outreach tool which the public can use to visualize mission downlink data and plan and visualize their own missions.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] NASA/UCLA. *Mars Surveyor 98 Lander*, 1999. http://mars.jpl.nasa.gov/msp98/lander/.

[2] Paul G. Backes, Gregory K. Tharp, and Kam S. Tso. The web interface for telescience (WITS). In *Proceedings IEEE International Conference on Robotics and Automation*, pages 411–417, Albuquerque, New Mexico, April 1997.

[3] S. Hayati, R. Volpe, P. Backes, J. Balaram, R. Welch, R. Ivlev, G. Tharp, S. Peters, T. Ohm, and R. Petras. The Rocky7 rover: A Mars sciencecraft prototype. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 2458–2464, Albuquerque, New Mexico, April 1997.

[4] R. Volpe. Navigation results from desert field tests of the rocky 7 mars rover prototype. *International Journal of Robotics Research, Special Issue on Field and Service Robots*, 18(7), July 1999.

[5] Eric Paulos and John Canny. Delivering real reality to the World Wide Web via telerobotics. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 1694–1699, Minneapolis, Minnesota, April 22-28 1996.

[6] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley. Robot teleoperation via WWW. In *Proceedings IEEE International Conference on Robotics and Automation*, May 1995.

[7] NASA-Ames. *NASA Ames IT Security Development Group*, 1999. http://pki.arc.nasa.gov/.

[8] Netscape. *SSL 3.0 Specification*, 1999. http://home.netscape.com/eng/ssl3/.

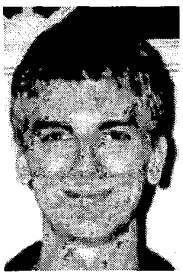
[9] RSA Data Security, Inc. *RSA Labs FAQ- What is Triple-DES?*, 1999. http://www.rsa.com/rsalabs/faq/html/3-2-6.html.

[10] Jeffrey S. Norris and Paul G. Backes. Wedds: The wits encrypted data delivery system. In *Proceedings IEEE Aerospace Conference*, Big Sky, Montana, March 2000.

BIOGRAPHY

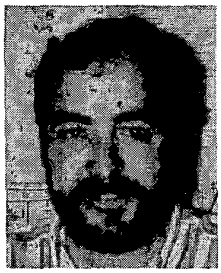
Paul G. Backes *is a technical group leader in the Autonomy and Control section at Jet Propulsion Laboratory, Pasadena, CA, where he has been since 1987. He received the BSME degree from U.C. Berkeley in 1982, and MSME in 1984 and Ph.D. in 1987 in Mechanical Engineering from Purdue University. He is currently responsible for distributed operations research for Mars lander and rover missions at JPL. Dr. Backes received the 1993 NASA Exceptional Engineering Achievement Medal for his contributions to space telerobotics (one of thirteen throughout NASA), 1993 Space Station Award of Merit, Best Paper Award at the 1994 World Automation Congress, 1995 JPL Technology and Applications Program Exceptional Service Award, 1998 JPL Award for Excellence and 1998 Sole Runner-up NASA Software of the Year Award. He has served as an Associate Editor of the IEEE Robotics and Automation Society Magazine.*

Kam Sing Tso *is the cofounder and Vice President of IA Tech, Inc., a small business specializing in Internet technologies, distributed and fault-tolerant systems, intelligent agents, and robotics. He received his Ph.D. in Computer Science from the University of California, Los Angeles, and B.S. in Electronics from the Chinese University of Hong Kong. He is one of the developers of WITS under a contract from the NASA Small Business Innovation Research program.*

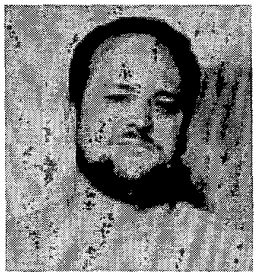


Jeffrey S. Norris *is a computer scientist and member of the technical staff of the Autonomy and Control Section at the Jet Propulsion Laboratory. He specializes in software engineering for telerobotics, distributed operations, machine vision, and large scale interfaces. He received his Bachelor's and Master's degrees in Electrical Engineering and Computer Science from MIT. While an undergraduate, he worked at the MIT Media Laboratory on data visualization and media transport protocols. He completed his Master's thesis on face detection and recognition at the MIT Artificial Intelligence Laboratory. He now lives with his wife in Azusa, California.*

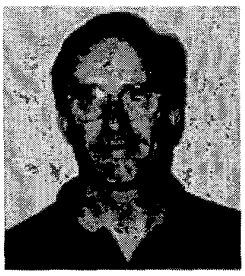
Gregory K. Tharp *received the BSME degree from the University of California, Berkeley, holds the B.A. in economics from the University of California, Santa Cruz, and received the MSME degree from the University of California, Berkeley, in 1989. After a one year staff appointment at the University of California, Berkeley, he worked as a consultant for Fujita Research for three years, studying virtual environment and telepresent display systems for remote manipulation. From*



1993 until 1997 he was a member of technical staff at the Jet Propulsion Laboratory, California Institute of Technology working on graphical user interfaces and image processing for the NASA robotics program. Currently, he is a research engineer at IA Tech Inc. where he is developing a collaborative distributed environment for planning and control of remote autonomous systems.



Jeffrey T. Slostad *is a member of the technical staff in the Imaging and Spectrometry Systems section at Jet Propulsion Laboratory, Pasadena, CA. He was the Operations Manager for the Robotic Arm for the Mars Polar Lander mission.*



Robert G. Bonitz *is currently with the Telerobotics Research and Applications Group at the Jet Propulsion Laboratory where he recently designed and developed the control algorithms and software for the Mars Volatiles and Climate Surveyor Robotic Arm which was also inherited for use on the Mars 2001 Lander Robotic Arm. Previously, he has conducted research in control algorithms for multiple-manipulator robotic systems, robust internal force-based impedance controllers, frameworks for general force decomposition, optimal force control algorithms, and calibration methods for multi-arm robotic systems. He has worked for a variety of industrial companies including Raytheon, TRW, Source2 International, and GTE. He has a PhD in Electrical Engineering from the University of California, Davis.*