Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, California. April 2000

Internet-Based Operations for the Mars Polar Lander Mission

Paul G. Backes*, Kam S. Tso†, Jeffrey S. Norris*, Gregory K. Tharp†,
Jeffrey T. Slostad*, Robert G. Bonitz*, and Khaled S. Ali*
*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. 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. Results are shown from the January 2000 field test in Death Valley, California.

1 Introduction

The Web Interface for Telescience has been developed to provide Internet-based distributed ground operations for planetary lander and rover missions. WITS provides an integrated environment for operations at a central location and collaboration by geographically distributed scientists. Collaboration by geographically distributed scientists at their home institutions enables participation in missions by a greater number of scientists and reduces operations costs. An earlier version of WITS provided some of the features needed for a planetary rover mission [1]. Based upon experience with WITS as an evaluation and public outreach tool in the 1997 Mars Pathfinder mission [2] and rover field tests [3, 4], WITS was redesigned and reimplemented to provide the features needed for planetary mission operations. This paper describes the new WITS system. Other examples of Internet-based robot operation can be found in [5, 6].

The Mars Polar Lander landed near the south pole of Mars on December 3, 1999 and was to perform an approximately three month mission [7]. Unfortunately, communication with the lander was not achieved so commanding the lander was not possible. The lander carried the Mars Volatiles and Climate Surveyor (MVACS) instrument suite, which had its mission operations center at UCLA. Results shown in this paper are from a January 2000 field test in Death Valley, California, using a mockup lander with flight spare components, shown in Figure 1.

Figure 1: Field Test Site

WITS served multiple purposes for the MPL mission. It was the primary operations tool for visualization of downlink image 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 lander mast-mounted Stereo Surface Imager (SSI) stereo camera, 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. This capability would enable them to participate from Arizona for part of the mission. Also, a separate WITS system was pro-

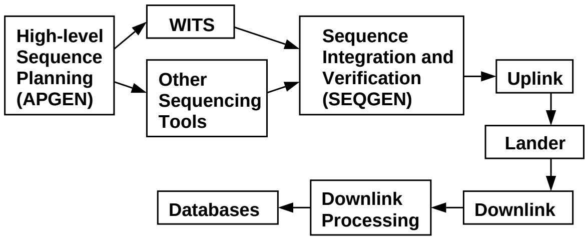

Figure 2: MPL Mission Operations Architecture

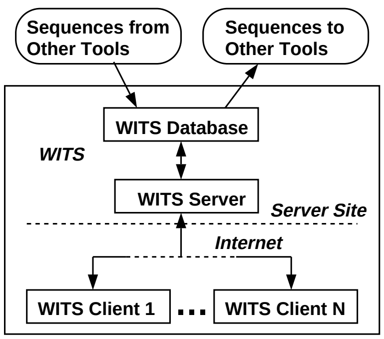

Figure 3: WITS Architecture for MPL

vided to the general public to download to their home computers to enable them to plan and simulate their own missions. Through the use of WITS, the Mars Polar Lander (MPL) mission was the first planetary mission to utilize Internet-based ground operations to enable geographically distributed scientists to collaborate in daily mission command sequence generation.

2 System Architecture

WITS was 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 was processed and put in databases, e.g., the WITS database. Sequence generation began by using a sequencing tool called APGEN which generated daily high level sequences for all the lander instruments. The sequences for the different instruments were sent to sequence generation systems specific to each instrument. Included in the sequences were requests which included text descriptions of what needed to be accomplished and how much time, energy, and data volume was allocated for each request. WITS was the sequence generation system for the RA and RAC. WITS generated the low-level commands to achieve the request goals while remaining within the specified resource allocations. The multiple sequencing systems then output their sequences to the SEQGEN planning tool where all the low-level sequences were integrated, resource checking on the integrated sequence was done, and final sequence modifications were made to ensure a valid sequence within resource constraints. The final sequence was then sent into the uplink process, eventually to be received at the lander.

The WITS architecture is shown in Figure 3. The database holds downlink data products and uplink sequence information. The server provides communication between the database and the clients. The clients are distributed over the Internet and provide the interface to the users to view downlink data and generate command sequences. Internet security was integrated between the server and clients. Communication between the client and server is implemented using Java Remote Method Invocation (RMI). The database is a structured file system. Other tools, e.g., planetary ephemeris tools or other sequence tools, can interact with WITS by reading and writing to the database or by direct communication with the server.

The WITS client and server are implemented using the Java2 platform, including the Java3D and Java Cryptography extensions. The client is run either as a Java applet using a web browser or an appletviewer, or as a Java application. Users must first download the Java Run-time Environment and Java3D.

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 means to transfer the data to and from the remote scientists is needed. WEDDS, the WITS Encrypted Data Delivery System, was created to provide the required secure Internet-based communication for WITS. WEDDS was integrated with WITS for the MPL 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.

All WEDDS connections are authenticated using the NASA Public Key Infrastructure (PKI) [8]. After authentication, communications are made through SSL (Secure Sockets Layer) sockets and are encrypted

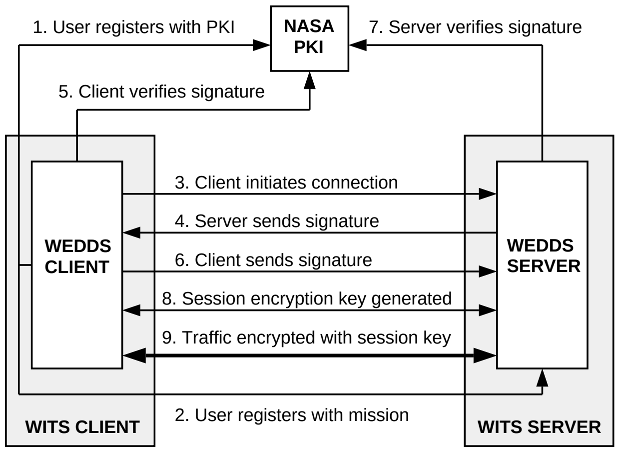

Figure 4: Secure Communication Steps

using the Triple-DES-EDE3 algorithm [9, 10]. The following are some of the advantages of WEDDS for Internet-based data distribution. 1) WEDDS does not require the remote users to request a data update. The data is automatically delivered as it becomes available. 2) The WEDDS administrator can specify on a user by user basis exactly which users will receive a particular type of file or directory. 3) Since WEDDS is written entirely in Java, it can run on most computers without modification. 4) WEDDS provides a high level of data security by using the SSL algorithm to perform authentication and the Triple-DES-EDE3 algorithm for encryption [11]. 5) WEDDS clients can be allowed to transfer files back to the mission control center. Files from users are stored on the server in a compressed, enveloped form that allows them to be scanned for hostile code. Since every client is authenticated using the NASA PKI, unauthorized users can not transmit files to the server.

WEDDS is implemented as two Java programs, a server and a client, using the publicly available Entrust Java Toolkit for its low level authentication and encryption tasks [12]. For each mission, there is typically one server, operating behind a mission firewall, and many clients, one on each remote user's machine.

Figure 4 illustrates the steps necessary for a single WEDDS transaction. Steps 1 and 2 occur once, before the beginning of the mission, while steps 3 through 9 occur for every transaction. In step 1, a remote user must obtain a security profile from the NASA Public Key Infrastructure (PKI), which requires appearing in person at a NASA center security office. A security profile is a set of files on a floppy disk, protected by a password, that contain a user's private key. A user needs his private key to positively identify himself online. The WITS server is also issued a security profile so that it can prove its identity to remote users. In step 2, each user must contact the mission administrator and request that their profile be given access to mission data.

Steps 3 through 9 are repeated for every transmission from the client to the server or from the server to the client. In steps 4 through 7, the server and client exchange digital "signatures", generated from their security profile. They verify these signatures by communicating with the NASA PKI. This process, SSL authentication, relies on the fact that it is nearly impossible for someone to generate another user's digital signature without that user's private key and password. In step 8, the last step in establishing an SSL connection, the client and server use the Diffie-Hellman key agreement protocol [13] to generate a unique symmetric encryption key that will be used to encrypt the traffic for this transaction. This encryption key is never sent as clear-text over the internet, and all subsequent traffic for this transaction is encrypted, making WEDDS transactions invulnerable to eavesdropping or "packet-sniffing" attacks. In addition, every WEDDS transaction is protected by a new encryption key. Further description of WEDDS can be found in [14].

4 Downlink Data Visualization

Downlink data from the lander or rover is provided to the user via various views (Figure 5). Views with Death Valley field test data are shown in the figures. 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. The Plan window (Figure 5) displays available views for a specific plan (a plan includes view definitions and sequence information for generating one uplink sequence). Each of the view instances is constructed using a specified subset of downlink data. New view instances can be specified as new data becomes available. 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 (Figure 5) shows the immediate area around the lander from above. Targets are displayed in the Overhead view, as well as selected points and view objects (view objects are described in Section 5 below). Clicking on a point in the Overhead view causes the x,y

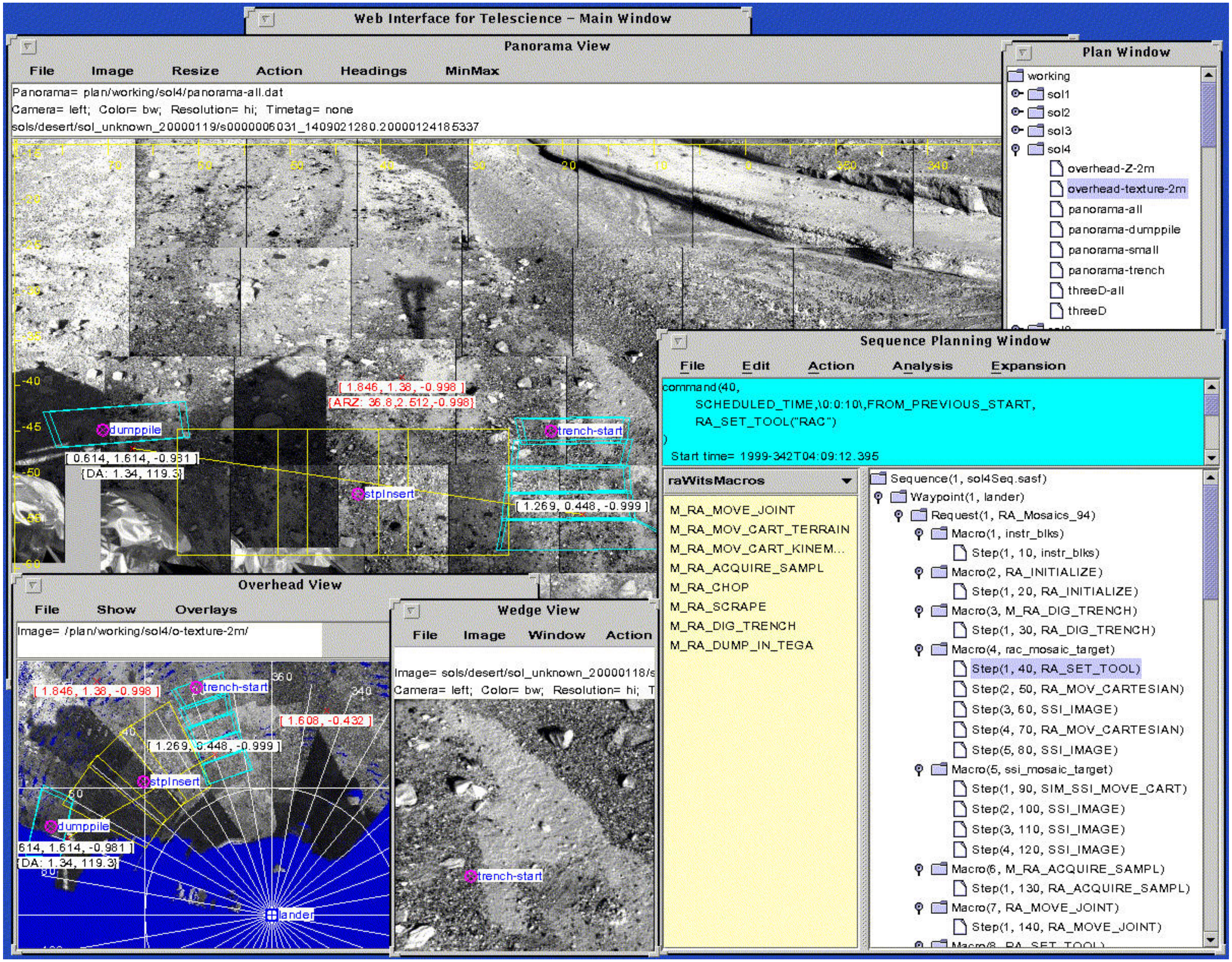

Figure 5: Panorama, Wedge, and Overhead Views and Sequence and Plan Windows

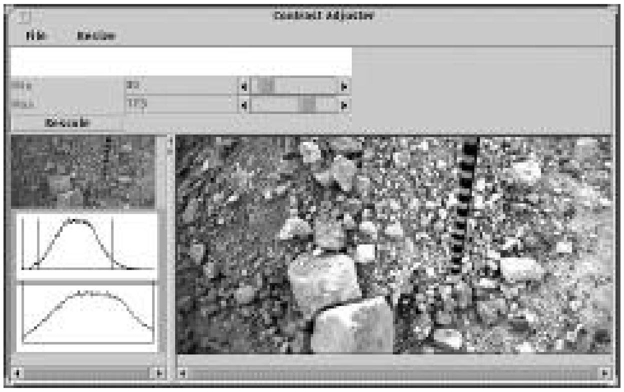

Figure 6: Contrast Adjuster View of Dump Pile

position to be displayed at the clicked location. Clicking and dragging causes a ruler to be shown with the start and end points and distance displayed.

The Panorama view (Figure 5) is a mosaic of images taken by a stereo camera, i.e., the SSI on the mast of the lander. Selecting a point in an image causes the x,y,z position at that point on the surface to be displayed. A target can be created at the selected point via a menu option. 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 of the panorama with various viewing options.

The Contrast Adjuster view (opened from a Wedge view pull-down menu) 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. Figure 6 shows the Contrast Adjuster view with a RAC image of the dump pile during the January 2000 field test in Death Valley.

The 3D view, shown in Figure 7, provides a 3D solid model visualization of the lander and terrain. Animated sequence simulation and state are visualized in the 3D view.

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. 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 on the terrain and displays them at the selected point. The point can be turned into a target via a menu option. Targets are displayed in the various views as pink circles and can be used as input parameters to sequence commands.

The Sequence window, shown in Figure 5, is used to generate a command sequence. A command sequence has a hierarchy of elements, which are (in descending order): 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 provided 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 output sequences in various formats. For the flight mission, WITS outputs sequences in the Spacecraft Activity Sequence File (SASF) format (a proprietary format).

The Sequence window shows the sequences for one plan. 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.

The left side of the Sequence window displays a list of macros which can be inserted into a sequence. 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 its insertion point in the sequence 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, in which the parameters for the macro are specified. Figure 8 shows the Macro window for the rac_mosaic_target macro. A macro-specific algorithm expands the macro into one or more command steps.

A macro can generate view objects which are displayed in the views to indicate what actions the macro is producing. Figure 5 shows square outlines which are view objects for SSI imaging commands. They in-

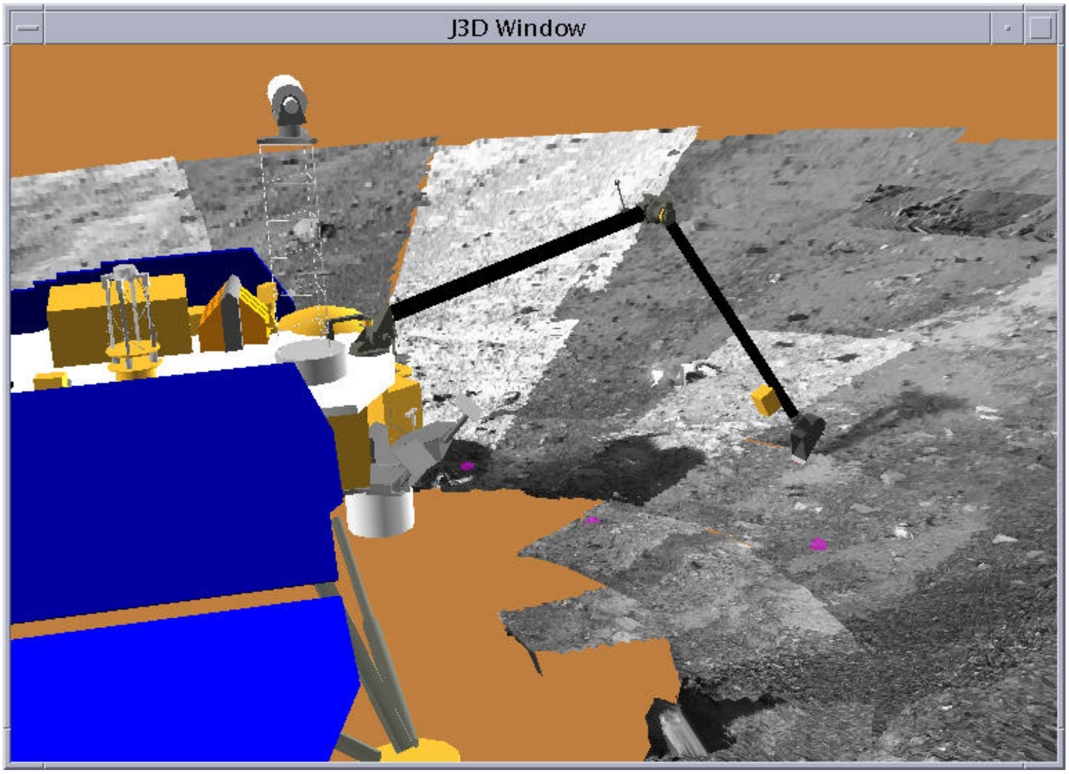

Figure 7: 3D View with Lander and Terrain Visualization

dicate where images will be taken by the SSI in the current sequence. Above the dump pile are view objects of RAC images and above the trench are view objects for a multi-image RAC mosaic that was used in the Death Valley field test (pairs of RAC images are taken to enable the generation of stereo data, e.g., anaglyphs). 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. 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 perform an animated simulation of 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 sets the 3D view viewpoint in the simulation; the viewpoint can be set to a fixed point such as above the lander or to a moving point such as the SSI camera. 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. 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 describes 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

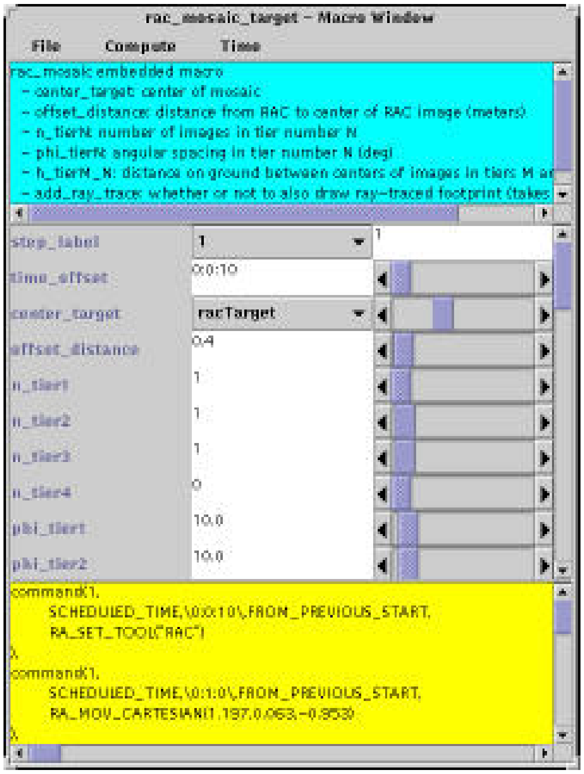

Figure 8: Macro Window

and 3D view and updated at each sequence step during simulation. In the 3D view, the sun position is shown relative to the SSI at a distance specified with a macro.

6 Example Sequence

An example sequence is shown in the Sequence window of Figure 5 and its view objects are shown in the views of the figure. A real sequence has various other low-level engineering commands. One request is shown which has many macros. The instr_blks macro has the resource allocations for the request. The RA_INITIALIZE macro initializes the Robotic Arm. 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 rac_mosaic_target macro takes two RAC images of the dump pile by moving the Robotic Arm and taking RAC images (RAC images are commanded using the SSI_IMAGE command). The ssi_mosaic_target macro expands into four steps which take a three stereo image SSI mosaic around the stpInsert target. The first step moves the simulated SSI in the 3D view to the central target and the next three steps are the commands that would be uplinked to the lander for SSI images. 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. At this point in the sequence, the sample might be dumped in the TEGA.

7 Distributed Collaboration

Internet-based collaboration is achieved in WITS by providing Internet-based users with daily downlink data updates and allowing them to input targets and command sequences and to save them to the common server. The Internet-based users can access targets and sequences created by other users and can use targets from other users' sequences. Users can also save and load sequences to their local computers.

8 Public Outreach

An important motivation for the development and use of WITS in a planetary mission is its use in public outreach. 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, new arm kinematics and new commands were used in the public version. The public is able to view mission data and plan 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 was hosted at Graham Technology Solutions, Inc.

9 Field Test Results

The Mars Polar Lander mockup with flight spare components was taken to a desert site in Death Valley, California for a field test January 17 – 22, 2000 (Figure 1). The three working instruments (Surface Stereo Imager, Robotic Arm, and Robotic Arm Camera) were operated remotely from the Mars Science Operations Center at UCLA via satellite link. Downlinked image data were imported into WITS which was used to develop RA and RAC command sequences. WITS sequences commanded the arm to successfully dig a 50cm long by 15cm wide by 10cm deep trench in rather challenging soil (somewhat hard and rocky). WITS also generated sequences for sample acquisition

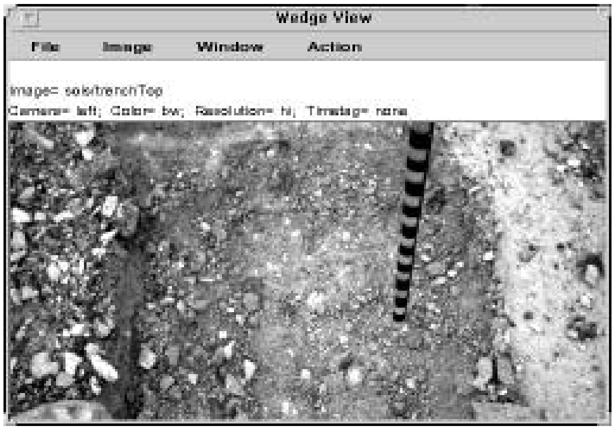

Figure 9: RAC Image of Trench Top

with the RA scoop and RAC mosaics of the terrain, trench and dump pile. The test was a success demonstrating the readiness of the flight components and operations system and all objectives were met. Figures 5, 6, 7, and 9, show data from the field test. A Wedge view of a RAC image of the top of the trench after digging is shown in Figure 9.

10 Conclusions

WITS provides a new Internet-based operations paradigm for planetary missions. 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 work described in this paper was funded by the Mars Exploration Technology Program, the TMOD Technology Program, the SBIR Program, and the Mars Polar Lander Mission, and performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

[1] 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.

[2] Paul G. Backes, Kam S. Tso, and Gregory K. Tharp. Mars Pathfinder mission Internet-based operations using WITS. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 284–291, Leuven, Belgium, May 1998.

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

[8] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

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

[10] RSA Security Incorporated. *RSA Labs FAQ Section 3.2 DES*, 1999. `http://www.rsasecurity.com/rsalabs/faq/3-2.html`.

[11] M. Wiener. Performance comparison of public-key cryptosystems. *CryptoBytes*, 3(3):1–5, 1998.

[12] Entrust Technologies. *Entrust/Toolkit Java Edition*, 1999. `http://developer.entrust.com/java/`.

[13] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22:644–654, 1976.

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