Sequence Planning for the FIDO Mars Rover Prototype

Paul G. Backes*, Jeffrey S. Norris*, Mark Powell*, Kam S. Tso†, Gregory K. Tharp†, P. Chris Leger*
*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California
†IA Tech Inc., Los Angeles, California
Paul.G.Backes@jpl.nasa.gov

Abstract— A planetary rover operations system has been developed and verified in terrestrial rover field tests. The Web Interface for Telescience (WITS) provides visualization of downlink data and rover command sequence generation. WITS also provides collaborative sequence generation by distributed Internet-based users. The Parallel Telemetry Processing (PTeP) system processes downlink data from the rover and places the data products in a database. The Multi-mission Encrypted Communication System (MECS) provides communication between the primary operations center and geographically distributed Internet-based users. The operations system was used to visualize downlink data and generate command sequences for the Field Integrated Design and Operations (FIDO) rover in desert field tests. Complex surface operations expected of future Mars rover missions were demonstrated in the field tests.

TABLE OF CONTENTS

1 INTRODUCTION
2 ARCHITECTURE
3 DOWNLINK DATA VISUALIZATION
4 UPLINK SEQUENCE GENERATION
5 CONCLUSIONS

Figure 1. FIDO Rover During Nevada Desert Field Test

1. INTRODUCTION

Mars rover mission operations includes the ground operations system on Earth, communication between Earth and Mars, and surface operations on Mars. The ground operations process starts with receipt of downlink data from the rover. The data is then processed and placed in a database. Users view the downlink data and then generate and validate the next command sequence that will be uplinked to the rover. This paper describes the ground operations system used to command the Field Integrated Design and Operations (FIDO) prototype Mars rover. Technologies are described which have been developed and integrated into the operations system to make rover ground operations more efficient. The ground operations system was used to command the FIDO rover during desert field tests in California and Nevada.

The FIDO rover is a prototype Mars rover similar to the rovers which will land on Mars in the 2003 Mars Exploration Rover (MER) mission [1]. FIDO is used to evaluate the complex surface operations expected of future Mars rover missions, including identification of rock targets, approaching the targets and conducting in-situ measurements, as well as collecting scientific data on distant targets. FIDO field tests exercise MER mission operations concepts and provide science operations team training in complex and real terrains.

Stereo cameras on the FIDO rover's mast and body are used to image the surrounding terrain. An Infrared Point Spectrometer (IPS) on the mast is used to characterize both near-by and distant targets on the terrain. An instrument arm places a microscopic imager on selected surface targets. A coring drill is used to extract rock samples. Figure 1 shows the FIDO rover during the May 2000 field test in Nevada.

The Web Interface for Telescience (WITS) is used for downlink data visualization and command sequence generation for FIDO. An updated version of WITS will be used for science operations in the MER mission and experience from the FIDO field tests has led to focused technology enhancements to benefit MER mission operations. WITS was originally developed to provide Internet-based distributed Mars lander and rover operations and was used to command the Rocky7 research rover [2], [3]. An adaptation of WITS was used for

0-7803-7651-X/03/$17.00/© 2003 IEEE
IEEEAC paper # 1414

Robotic Arm and Robotic Arm Camera command sequence generation for the Mars Polar Lander (MPL) mission which was to begin surface operations in December 1999 [4]. Unfortunately, communication with the MPL lander on the Martian surface was not achieved, so commanding the lander was not possible.

WITS is used with other tools for the FIDO rover operations system. The Parallel Telemetry Processing (PTeP) system processes downlink data from the rover and places the data products in a file system-based database. The Multi-mission Encrypted Communication System (MECS) provides communication between the primary operations center and geographically distributed Internet-based WITS client systems.

New rover operations technologies have been developed to make sequence generation more efficient. Internet-based operations, where users can participate in command sequence generation from their home institutions, enables users to fully participate in daily rover operations without having to travel to the primary operations center. Simulation of sequence commands in various types of views enables users to quickly verify the commands of the sequence. Co-registration of image data sets enables target selection and viewing of data in images taken from any rover position. Integrated resource analysis and rules checking enables fast verification of the sequence. Group collaboration and shared sequence editing enable distributed users to efficiently collaborate in data analysis and sequence generation. Automated terrain traversability analysis shows safe traversal paths to assist a user in planning rover traverse paths. Automated report generation greatly reduces the necessary time to generate reports describing a sequence that was sent to the rover. These capabilities have been developed and integrated into WITS and demonstrated in FIDO rover operations.

This paper describes the ground operations system for the FIDO rover. The PTeP downlink telemetry processing system, the file system database, and MECS Internet communication system are described. The capabilities and use of WITS for downlink data visualization, sequence generation, and distributed operations are described in detail. Examples are given using data from May 2000 and September 2000 FIDO rover field tests.

2. ARCHITECTURE

The operations architecture for the FIDO rover used in the field tests is shown in Figure 2. The architecture has equivalent elements to an actual Mars rover mission. Surface operations are performed by the FIDO rover. Earth to Mars communication is simulated by satellite communication which provides a communication link between the field test site and the primary operations center at JPL. Downlink processing is done by PTeP with results placed in a file system database. WITS is used for data visualization and sequence generation. MECS is used to distribute data to Internet-based users and for communication with Internet-based WITS clients.

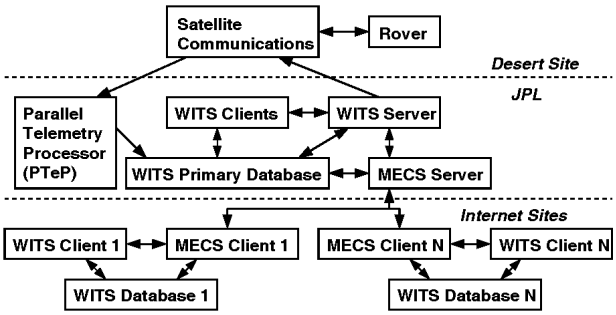


Figure 2. FIDO Operations Architecture

The WITS database is a structured file system. The database includes constant information such as models, uplink sequence generation information such as sequences, and downlink data products. Downlink data is automatically processed and placed in the correct place in the database. Information in the database is used by WITS for data visualization, sequence generation, and for automated report generation.

Downlink Data Processing

The Parallel Telemetry Processor (PTeP) processes all of the data received from the instruments on FIDO into 25 different science data products and automatically places these products in a structured file system database [5]. A prior downlink processing system used Unix scripts to process the downlink data. That system was greatly limited by the fact that it could only process a single downlink packet at a time. In addition, it did not gracefully handle error conditions or provide an intuitive interface to the user.

PTeP is a Java language application capable of processing an arbitrary number (typically 4) of downlink packets from the rover simultaneously, and automatically takes advantage of all of the CPUs that are available on the computer. PTeP's architectural efficiency enables throughput of about three packets per minute. PTeP provides its user with a graphical monitoring interface that illustrates the steps in the downlink processing pipeline for each type of instrument packet and indicates the status of each packet currently being processed. When processing errors occur, the affected packets are moved into an error queue for later review while processing of new packets continues. At any time, a user can access detailed information about a processing error, correct the problem, and resume processing of the packet.

The database structure for the processed FIDO downlink data products is shown in Figure 3.

Client-Server Implementation

WITS uses a client-server architecture to support communications between numerous clients and the one server. The server provides communication between the primary database and the clients. The clients are distributed over the Internet and provide the interface to the users to view downlink data

results
- aerial (images taken from orbit or during descent to the surface)
  - alt-1 (images from one altitude)
  - alt-n
- sites
  - site-1 (one site)
  - site-n
    - 3dView (3D view information for this site)
    - overheadView (overhead view information for this site)
    - panoramaView (panorama view information for this site)
    - position-1 (a position within a site)
    - position-n
      - instrumentName (directory with data for this instrument)

Figure 3. Downlink Database Structure

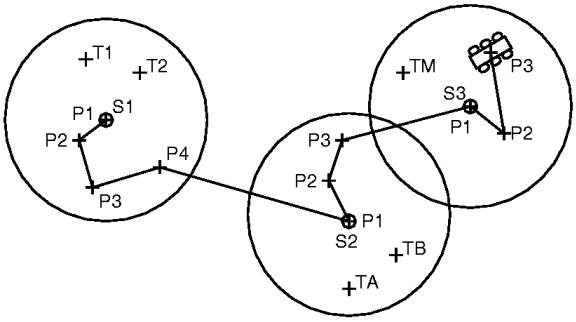


Figure 4. Sites, Positions, Targets

and generate command sequences.

The WITS client and server are implemented using the Java2 platform, including the Java3D and Java Cryptography extensions. Communication between the clients and server is implemented using Java Remote Method Invocation (RMI).

The WITS server is implemented as a Java language application running on a computer at the primary operations center (at JPL for FIDO operations). The WITS client is downloaded from the JPL WITS website. It is usually run as a Java application; it can also be run as a Java applet in a web browser using Java Plug-ins. Users must first download the Java Run-time Environment and Java3D.

Internet-based Operations

Internet-based operations are provided by the client-server architecture of WITS and the data distribution and secure communication capabilities of MECS. An Internet-based remote user is provided all of the functionality available to a user at the primary operations center. Remote users have access to downlink data products almost as quickly as users at the primary operations center. As soon as the downlink data products are verified as valid, then they are distributed to remote users.

Internet-based operations provide various benefits. Use of the public Internet for distribution of mission data eliminates the costs of dedicated leased lines which have been previously used to support remote users. Enabling users to participate in a mission from any location on the World Wide Web reduces both housing and operations center infrastructure costs. Also, scientific experts can participate in the mission who would otherwise not be available due to travel constraints. Pre-mission operations readiness tests (ORTs) can also benefit from Internet-based operations, because participants from other institutions can fully participate in ORTs without the additional cost (both in time and money) of having to travel to JPL for each test. Thus, more tests can be performed which involve all mission participants.

A secure and efficient means to transfer data is needed to enable collaboration in daily sequence generation by Internet-based scientists. The Multi-mission Encrypted Communication System (MECS) was created to provide the required secure Internet-based communication [6]. MECS was integrated with WITS both for FIDO rover operations and for Mars Polar Lander mission operations. MECS operates in a fashion that is transparent to the remote user: Files simply appear as they become available and connections are made securely without any additional effort on the part of the user. MECS provides two types of encrypted communication for WITS: 1) automated delivery of database updates to Internet-based clients, and 2) encrypted communication between the clients and server. Since encryption was not required for the FIDO rover field tests, the communication features of MECS were used without encryption.

MECS connections are authenticated using the NASA Public Key Infrastructure (PKI). After authentication, communications are made through SSL (Secure Sockets Layer) and are encrypted using the Triple-DES-EDE3 algorithm. MECS is implemented as two Java programs, server and client, using the publicly available Entrust Java Toolkit's low-level authentication and encryption capabilities. For each mission, there is typically one server, operating behind a mission firewall, and many clients, one on each remote user's machine.

Internet-based operations was used effectively in the FIDO field tests. In the May 2000 field test near Tonapah, Nevada, operators used WITS and MECS to visualize downlink data and generate command sequences from JPL; Ithaca, New York; Birmingham, Alabama; Flagstaff, Arizona; St. Louis, Missouri; and Copenhagen, Denmark. In the operations readiness tests leading up to the field test, scientists Steve Squyres and Ray Arvidson and their colleagues from Cornell University and Washington University, respectively, were able to lead the tests from their remote locations.

Sites and Positions

The physical locations that the rover traverses to are organized using sites and positions, as depicted in Figure 4. A site is a new area in which to explore. It is generally the area within a new panorama, e.g., an area of about 20 meters radius. A position is a location that the rover has moved to within a site. A new position is defined whenever the

rover wheels have turned and new data has been acquired to be downlinked. Sites are shown in Figure 4 as circles with crosses in them with names starting with S and positions are shown as crosses with names starting with P.

Targets are locations where science data is taken. Targets are attached to the terrain. For example a target is where Infrared Point Spectrometer instrument data is taken and can be distant from the rover. Targets are depicted in Figure 4 as crosses with names starting with T.

Various coordinate frames define the kinematic relationships between locations that the rover has visited. A SITE frame is specified for each site and has its X axis aligned with North. The LANDER frame is the first SITE frame, at the start of the mission. The ROVER frame is a frame fixed on the rover. A POSITION frame is the ROVER frame when a new position is defined.

Science data is defined relative to a POSITION frame. For example, when a stereo image pair is processed and the 3D coordinates on the terrain associated with each pixel are computed, the 3D coordinates are defined relative to the POSITION frame where the rover was when it took the images. This is important because the data relative to the rover's position stays constant, but the estimate of the rover's position is likely to be updated upon further analysis. Updating the estimate of the rover's position (by updating the transformation from the SITE frame to the POSITION frame) does not require updating of processed instrument data.

WITS automatically manages the various coordinate frames and transformations between coordinate frames. The user can then select targets in imagery from any rover position and planning information can be displayed in imagery from any rover position. The user generally sees all data relative to the current site frame, independent of the frame that the data is actually stored relative to.

3. DOWNLINK DATA VISUALIZATION

Downlink data from the lander or rover is provided by WITS by means of various 'views'. Panorama, Overhead, Wedge, and 3D views with data from the FIDO rover May 2000 desert field test are shown in Figure 5. The Results window (Figure 6) enables the user to select data products to be visualized in views, based upon rover positions. The Plan window (Figure 15) provides available views based upon a specific uplink plan. A plan includes view definitions and sequence information for generating one uplink sequence.

The views display various planning information. Information is displayed consistently in all views where it is visible. In Figure 5, the rover is shown as a 3D solid model in the 3D view, in 3D wireframe in the Panorama view, and in 2D wireframe in the Overhead view. The ruler is shown in the Panorama view (it is out of range in the other views). The pointer is shown in the Panorama and Overhead views. Targets are pink circles, waypoints are blue squares, and group markers are green triangles. Yellow footprints are view objects where pancam images will be taken. Small yellow circles are view objects where IPS spectrometer data will be taken, e.g., at ips1 and ips3.

Results Window

The Results window displays lists of available downlink data organized by sites and positions. The Results window is a tab in the Main window, as shown in Figure 6. Selecting a data product causes the associated view to pop up displaying its data. WITS automatically determines what type of data the selection represents and opens the appropriate view to display the data in. The various types of data views are described below.

At the bottom of the Results window is the Reload Database button. Pressing this button updates the display in the Results window. This is used after the database has been updated with new data products. The updated database could be either the primary database for users at the primary operations center or the database on a remote user's computer that has been updated by MECS.

Descent View

The Descent view provides a 2D image taken from the air, e.g., from orbit or during descent to the surface. The Descent view displays the landing location, planning information, and all sites up to the site of the current plan. The sites are named and connected sequentially. The view can be rescaled and resized, and the user can select to toggle display of sites, lines between sites, waypoints, and targets. The user can select a point and can choose to display 3D points in LANDER frame or last site frame coordinates.

An example Descent view, using data from sol9 of the first week of the May 2000 FIDO rover field test, is shown in Figure 7. The figure shows the sites that the rover has traversed to, with the rover currently at site 7. The image also shows the targets (circles) and waypoints (squares) that have been specified for the current plan.

Overhead View

The Overhead view provides 2D overhead images of the area around the rover, as shown in Figure 5. The Overhead view is different from the Descent view in that view objects for a sequence are displayed (view objects are discussed in Section 4). The user can choose between various background images including texture, elevation, contour, and obstacle; a color-coded elevation map is shown in Figure 5. The Overhead view provides location selection and display, ruler tool selection and display, target and waypoint display, view object display, square, radial, and angle grids display, and toggle of display of headings, radial grid, and square grid.

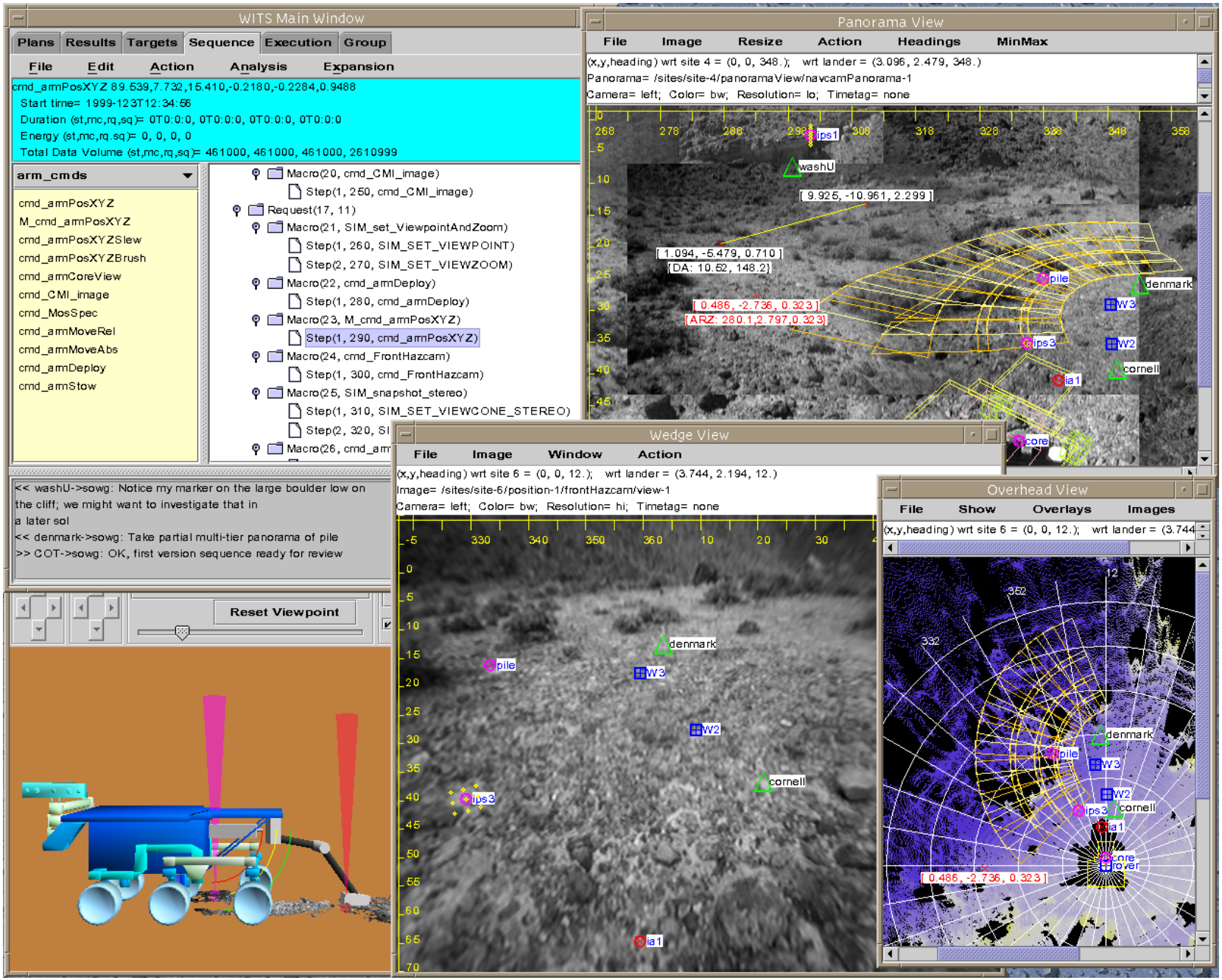


Figure 5. WITS Visualization and Sequence Generation

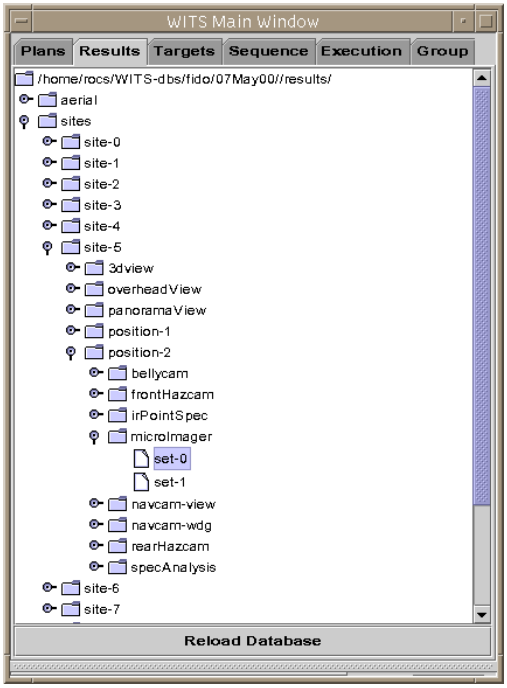


Figure 6. Results Window

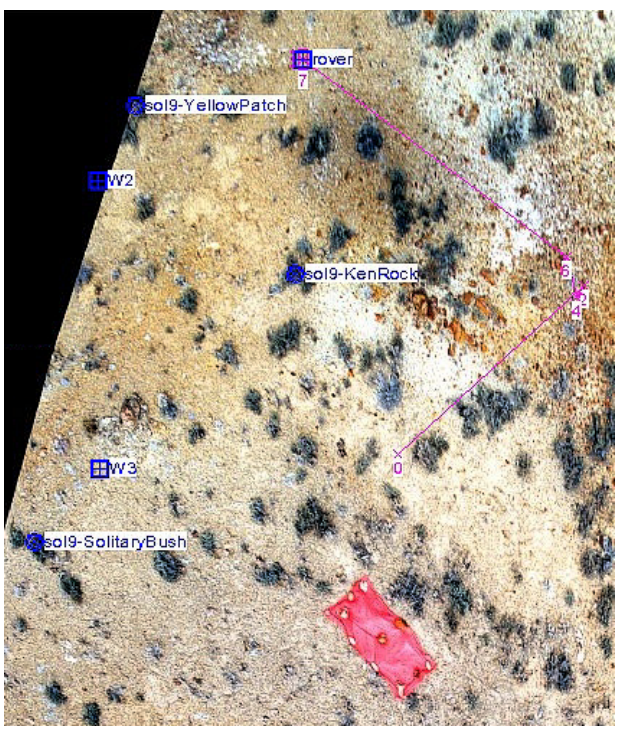


Figure 7. Descent View

Wedge View

The Wedge view displays one image associated with a stereo camera pair, as shown in Figure 5. As in other views, targets are pink circles, waypoints are blue squares, and group markers are green triangles. Small yellow circles are view objects where IPS spectrometer data will be taken, e.g., at ips3.

The Image menu allows the user to select what image product to view. There are various standard image products available. Standard image products for monochrome images include the original left and right images from the stereo pair, versions of those images with elevation data overlayed on them, and a combined anaglyph image which can be viewed with red-blue glasses to see scene in stereo. Standard image products for color images include the composite color left and right images, versions of those images with elevation data overlayed on them, the anaglyph image, and separate images for each camera from the three color bands.

The Wedge view provides location selection and display, ruler tool selection and display, target and waypoint selection and display, and view object display. Azimuth headings are shown relative to North and tilt angles are shown relative to the horizontal. Another menu item allows the user to linearly modify the contrast of the image such that 90%, 95%, or 98% of the pixel intensities are within the total range.

3D view data and Overhead view data are automatically produced for each image stereo pair. 3D and Overhead views corresponding to the Wedge view can be opened via menu items in the Wedge view Action menu item.

Panorama View

The Panorama view displays a mosaic of images taken by a stereo camera, as shown in Figure 5. The rover is shown in wireframe and the ruler and pointer are visualized. Targets, waypoints, and group markers are shown. Image footprints are view objects where mast mounted pancam images will be taken. Small yellow circles are view objects where IPS spectrometer data will be taken, e.g., at ips1 and ips3.

The images of a panorama are automatically placed in the mosaic at run-time. The Panorama view provides location selection and display, ruler tool selection and display, target and waypoint selection and display, and view object display. Azimuth headings are shown relative to North and tilt angles are shown relative to horizontal. The user can change the min and max azimuth and tilt angles to draw the panorama with a convenient angle range. The view can be rescaled to 1, 1/2, and 1/4 size. Scroll bars allow the user to scroll through the panorama both vertically and horizontally. A menu item allows the user to linearly modify the contrast of all images such that 98% of the pixel intensities are within the total range.

When a user selects a pixel in an image, the associate 3D

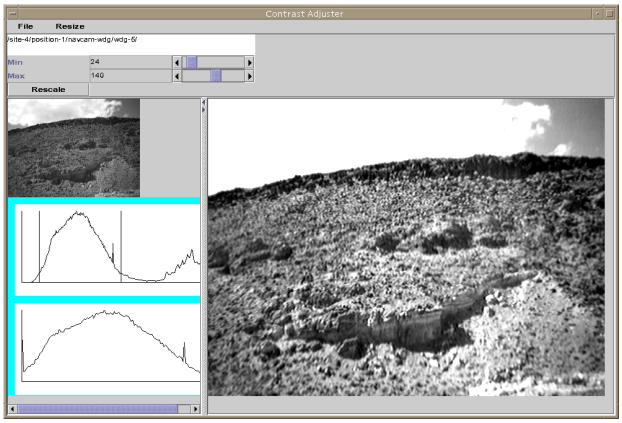


Figure 8. Contrast Adjuster View

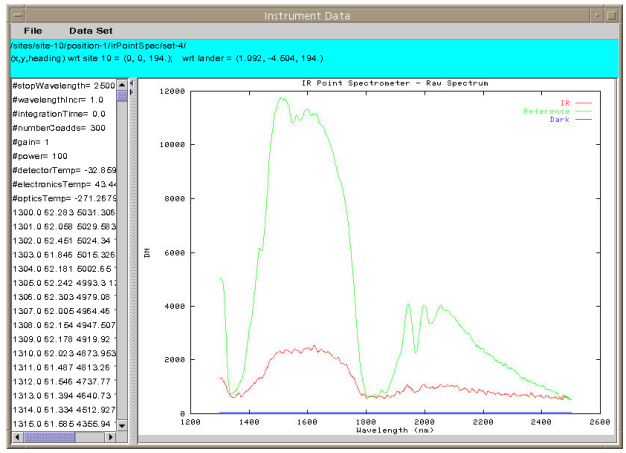


Figure 9. Instrument Data View - IPS

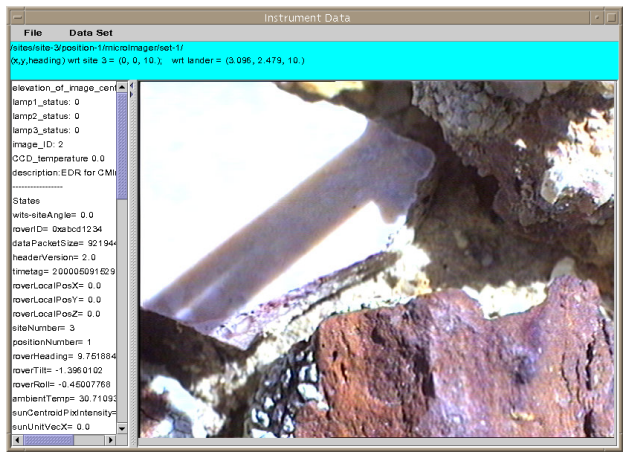


Figure 10. Instrument Data View - Color Microimager

point on the terrain becomes the current selected point and the image that the point is in becomes the currently selected image. A menu item allows the user to open a Wedge view for the selected image.

Contrast Adjuster View

The Contrast Adjuster view is opened from a Wedge view pull-down menu and enables the contrast to be adjusted for a Wedge view image, as shown in Figure 8. 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 initial image and the histograms of the initial and adjusted images are shown on the left and the modified image is shown on the right.

3D View

The 3D view provides 3D visualization of the rover and terrain, and planning information. Targets are pink cones with spheres at the bottom where a target is (the currently selected target is colored red). Waypoints are blue boxes. The user can interactively change the viewpoint and zoom and translate and rotate the viewpoint. In Figure 5, the rover and two targets are shown.

Control Window

The Control window provides interactive control over visualization in the 3D view. Specific tabs are provide for mission-specific hardware elements. For the FIDO rover, the user can specify states of mast angles, mini-corer, body, instrument arm, and camera view cones. Also, the user can select which images to show as 3D terrain in the 3D view.

Instrument Data View

The Instrument Data view displays science instrument results. It is assumed that science instrument data has been processed to generate a jpeg image. The jpeg image is displayed along with the rover state when the data was acquired. An example of IPS spectrometer data from the FIDO field test is shown in Figure 9. An example of Color Microimager data from the FIDO field test is shown in Figure 10.

Pointer and Ruler

The pointer tool enables selection of a 3D point in an image. Clicking on a point in a Descent, Overhead, Panorama, or Wedge view causes a cross to be drawn there with the X,Y,Z values on the terrain for that point with respect to the current site. This location becomes the global currently selected location. An example is shown in Figure 5.

The ruler tool enables drawing a ruler between two 3D locations in the terrain. Clicking and dragging between two pixels in a Descent, Overhead, Panorama, or Wedge image causes crosses to be drawn at the start and end points and the X,Y,Z values at the start and end points to be drawn, with the X,Y,Z values given with respect to the current site. The distance between the points and the azimuth direction from the start to end point are also displayed. A line is drawn between the start and end points. The ruler is stored as a view object and displayed in all Descent, Overhead, Panorama, and Wedge views. An example is shown in the Panorama view of Figure 5.

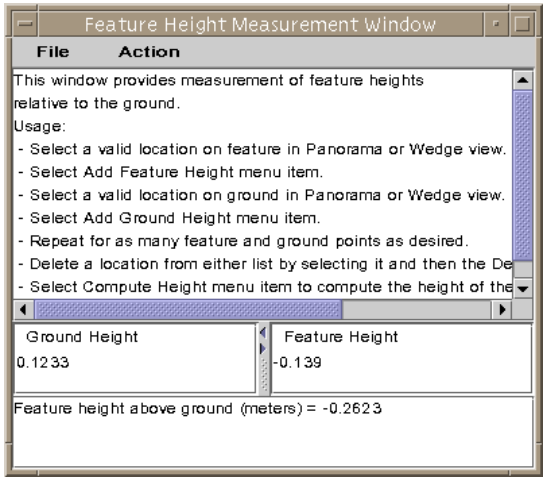


Figure 11. Feature Height Window

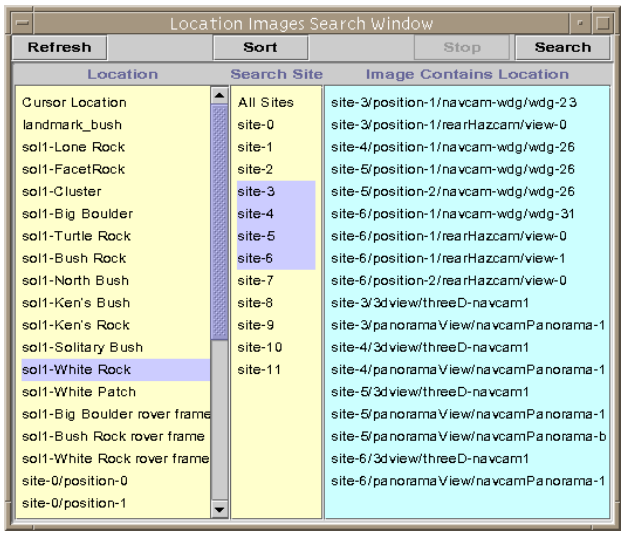


Figure 12. Location Images Window

Feature Height Window

The Feature Height window is used to measure the height of a feature. The user selects points on the feature and on the ground and the average height is displayed. An example is shown in Figure 11.

Location Images Window

The Location Images window is used to find all images and views that a specified location is seen in. An example Location Images window is shown in Figure 12. The available locations are listed in the left column, including the current cursor location, all targets and waypoints, and all positions that the rover has traversed to. The user selects the desired location. The sites to search are selected in the middle column. The user can select one, multiple, or all sites to be searched. The example shows that four sites will be searched. Searching a site means to check all images and views which have data in the specified site. The search results are shown in the right column. Selecting a search result will cause the associated WITS view to open.

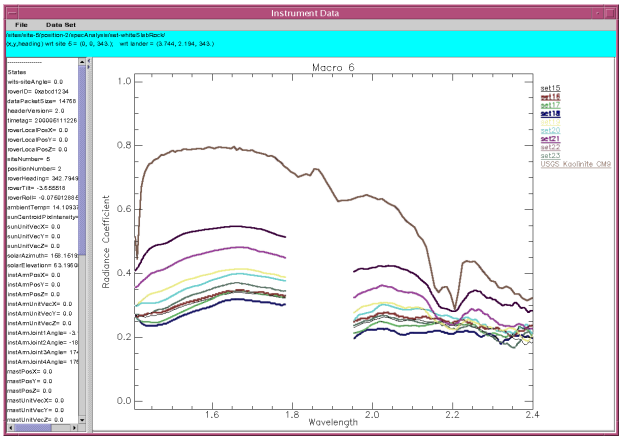


Figure 13. Custom Data Product Example

Custom Data Products

Users are able to create their own data products, place them in the WITS database, and then view them with the Wedge, Overhead, and Instrument Data views. The user generates the data in the correct format and puts it in the database at the desired place. Wedge view, Overhead view, and Instrument Data view custom data are produced by users as jpeg images with a specified naming convention. The new data product automatically shows up in a menu of the associated view. Data placed in the primary database is accessible by all mission participants. This is a convenient means for users to store custom data products in the correct place in the database, to view their data products, and to share their data products with other users.

Figure 13 shows a custom data product viewed with the Instrument Data view. The data product shows a library spectra of the mineral Kaolinite and the spectra from nine points in an IPS grid. It was produced by a remote participant during the May 2000 FIDO field test and saved to the primary database at JPL.

4. Uplink Sequence Generation

Sequence generation starts with users selecting targets where science activities will be performed. Waypoints are selected to specify where the rover will traverse through. Macros are added to a sequence to specify the tasks that the rover will perform. Resource analysis and rules checking are performed to verify that the sequence is valid within specified constraints. The sequence is simulated and then sent to the rover for execution. Features that WITS provides to support sequence generation by a group of users are described below.

Plans, Plan Window

A plan contains information associated with one uplink sequence such as science targets, sequence fragments, and associated views. Plan data is in a directory in the uplink part

uplink
- targetLibrary
  - targetLib-name (one target library file)
- plan
  - plan-name (directory for one plan)
    - viewDefinition-name (a view definition directory)
    - planTargets (plan targets definition file)
    - apgen (directory of sequences to be imported from APGEN)
    - seqgen (directory of sequences to be exported to SEQGEN)
    - witsSeq (directory of WITS sequences)

Figure 14. Uplink Database Structure

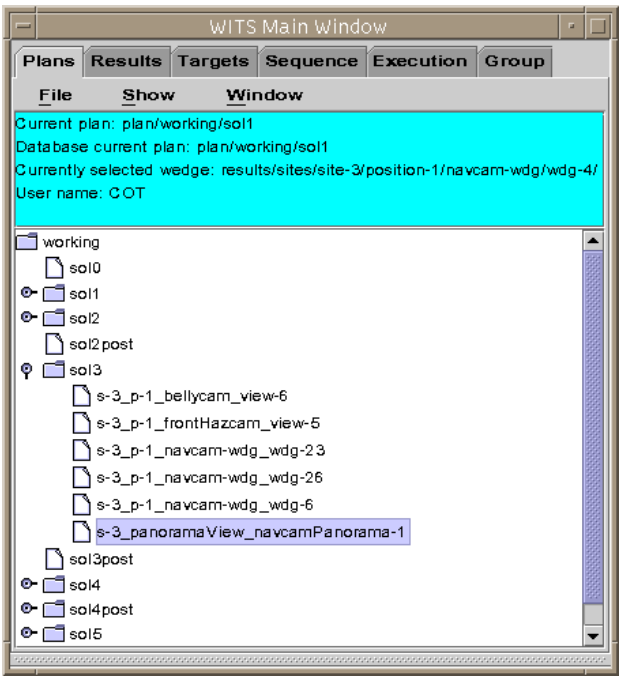


Figure 15. Plan Window

of the database. The uplink database structure is shown in Figure 14.

It is useful to collect all planning information for one sequence into a plan. There is much downlink data available to view, but only a small subset might be of interest for planning one sequence. The set of data and the view definitions to view it are stored with the plan. After the downlink data is available, a user can use the Results window to view any data. An open view that is of interest to the plan is associated with the plan by selecting 'Save to Plan' in the view's File menu. All targets created and saved while in a plan are defined and stored in the planTargets file. The apgen and seqgen directories represent directories where sequences from or to other sequence tools are stored. The apgen and seqgen directories referred to the APGEN and SEQGEN tools in the Mars Polar Lander mission.

The Plan window is a tab in the Main window, as shown in Figure 15, and is used to manage plans. The Plan window displays available Panorama, Overhead, Wedge, and 3D views for a specific plan. This is a convenient way to present available views to the user. Each view definition has a specific set of downlink data it uses, so definitions of views may be updated for each new plan. The Plan window has 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 user changes plans by double clicking on a plan.

The Plan window is also used to provide general interaction with the user. Via the Plan window Show menu the user can toggle whether to show various items in all views including targets, waypoints, pointer, ruler, view objects, and image outlines. Via the Plan window Window menu, the user can open other general windows and views including: Options window, Feature Height window, Location Images window, Synthetic Location window, Target Reachability window, Control window, Check Ephemeris window, Blank Overhead view, and Blank 3D view. The Synthetic Location window enables a user to specify targets and waypoints by typing in the values directly or getting the X,Y values from the Overhead view and specifying the Z value. The Check Ephemeris window provides pointing information to celestial bodies, e.g., the earth, sun or moons of mars.

Target and Waypoint Selection

Targets are locations where science data will be acquired. Waypoints are terrain locations where the rover will traverse to. A user specifies targets and waypoints interactively in the views and they can be used by name as parameters in macros in a sequence.

A target or waypoint is created from the currently selected location. A new selected location is created by selecting a point in a Descent, Overhead, Panorama, or Wedge view. A cross is drawn at the selected location and the corresponding 3D coordinates on the terrain are displayed. This selected location can be turned into a named target or waypoint using the Add menu of the Targets window. WITS stores various pieces of information associated with the location including image in which the location was selected and the surface normal.

A target or waypoint can be moved by clicking on it in a view and dragging it to another location or moving it pixel-by-pixel with keyboard arrow keys. By default a target can not be moved after it has been saved to the database. The ability to edit, which includes moving, a target after saving it to the database can be set in the Options window.

When a waypoint is added, WITS automatically adds a new waypoint-level element to the current sequence. A request element is added with a driveRover macro. The sequence in the Sequence window is automatically updated.

The Targets window provides management of targets and is accessible as a tab in the Main window, as shown in Figure 16. A target can be added, deleted, or edited. There are two types of targets: plan targets and library targets.

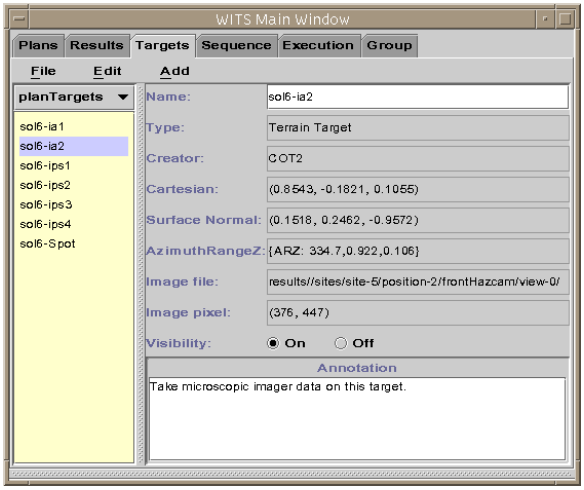


Figure 16. Target Window

Plan targets are targets associated with the current plan. They are stored in the database with the current plan and are not accessible by other plans. Targets are saved to the current plan in the Target window by selecting File/Save.

Library targets are targets that are stored in a target library file in the database. Above the list of targets in the Target window is a pull-down menu of available target library files (the plan target list is also one of the options). The list of targets in the Target window are the targets in the selected target library. Library targets can be imported into a plan and then saved as a plan target. A target is saved to a library file by selecting File/Export. This brings up the Target Library window where the user specifies which library file to put the target in. The user can define a new library file in the Target Library window.

Loading targets with File/Load causes the plan targets to be reloaded from the server. This might be done after a remote user has added a target.

An annotation for a target can be added to explain the purpose of the target, as shown in Figure 16. The visibility of a target can be turned on and off using the Targets window. Turning off the visibility of a target causes it to not be displayed in the views, but it is kept as part of the plan.

Sequence Building

The Sequence window is used to generate a command sequence and is accessible as a tab in the Main window, as shown in Figure 5. 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. 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 Mars Polar Lander mission, WITS output sequences in the Spacecraft Activity Sequence File (SASF) 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 spacecraft. 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.

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 Macro window is used to specify the parameters for a macro. An example Macro window is shown in Figure 17. 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 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.

View Objects

View objects are icons of various types that are drawn in the views. View objects are drawn in all appropriate views. A standard view object is the Ruler Tool view object which is used to visualize the Ruler Tool.

View objects are also used to visualize commands in the sequence. Commands to take pictures with mast-mounted cameras are visualized with footprint view objects. Figure 5 shows footprint outlines of pancam images defined in a pancam panorama command. The footprints can be seen in both the Panorama and Overhead views. Small circles are used as view objects for points where Infrared Point Spectrometer data will be taken, as at target ips3 in the Wedge view of Fig-

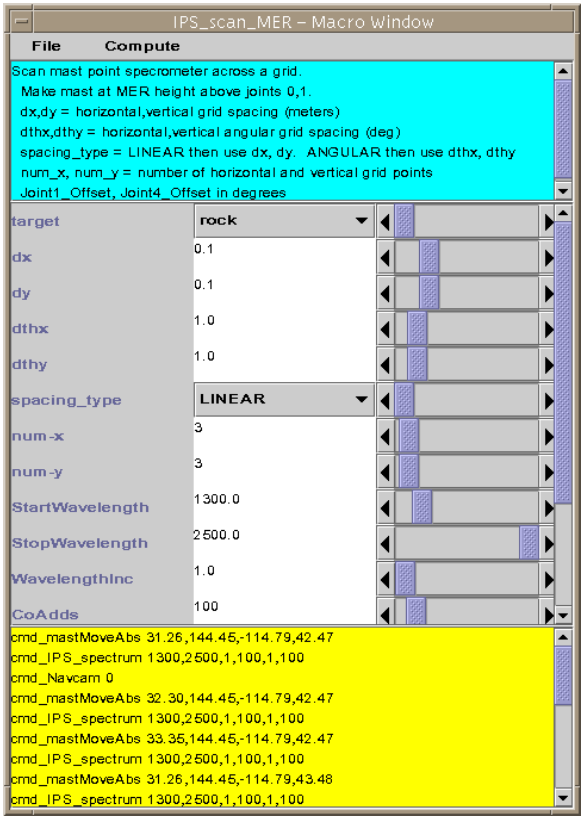


Figure 17. Macro Window

ure 5 and at target ips1 in the Panorama view. When a macro or command is added to a sequence, the associated View Objects are automatically computed and displayed in the views.

State Visualization

State visualization and simulation are used to view the predicted state of the rover at different points in the sequence. When a user selects a step in the sequence in the Sequence window, state visualization updates all the views with the state of the rover at end of the selected step. State visualization is useful when checking the state of the rover, e.g., instrument arm and mast angles, at specific steps of the sequence. The rover states at the end of each step in the sequence are computed at the same time as the resources used by the steps such as time, energy and data volume.

Simulation

The sequence can be simulated using the Execution window, which is a tab in the Main window. The whole sequence can be simulated or the user can select specific commands to simulate. Simulation provides smooth simulation of each step of the sequence, e.g., the motion of the instrument arm during deployment and placement of an instrument on a rock. The user can select to single step through the sequence starting at a selected step. Simulation results are visualized in the various types of open views.

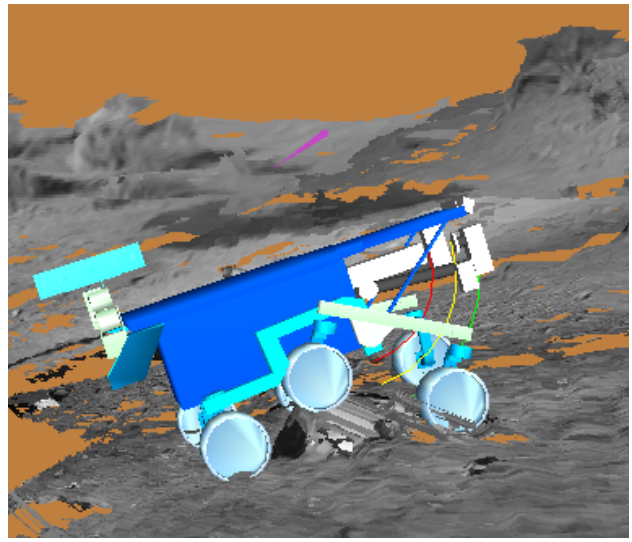


Figure 18. Traverse Simulation

Rover traverse simulation is accomplished by computing and displaying the rover configuration at incremented steps along the path. The rover position, orientation, bogey angles, and resulting rover body tilt are displayed at each step, as shown in Figure 18. The rover heading is known from the path. For each incremented path position, 2D (X,Y) positions of the six rover wheels are determined from the predicted rover position along the path. The elevations of the six wheels are then determined from the terrain information.

The following algorithm is used to compute the bogey angles, body tilt, and body roll at each path increment. A bogey connects the forward two wheels on each side of the rover. A rocker connects the rear wheel and the center of the bogey on each side.

1 The bogey angle with respect to the terrain reference frame for each of the two bogeys is calculated using the difference in heights of the two bogey wheels and the distance between the wheels.
2 The height of each of the two bogey attach points on the rocker is computed using the height of front bogey wheel, the bogey angle, and the mechanical geometry of the bogey/wheel mechanism.
3 The angle of each rocker with respect to the terrain reference frame is calculated from the heights of the rocker wheel and bogey attach point and the rocker geometry.
4 The height of each of the rocker attach points is calculated using the rocker angle, rocker wheel height, and rocker geometry.
5 Because the rockers are attached through a differential linkage they actually represent a single degree of freedom. The rover tilt with respect to the terrain reference frame is the average of the two rocker angles.
6 The rocker angles with respect to the rover are calculated by subtracting out the rover tilt.
7 The bogey angles with respect to the rockers are calculated by subtracting out the rocker angles from the bogey angles.

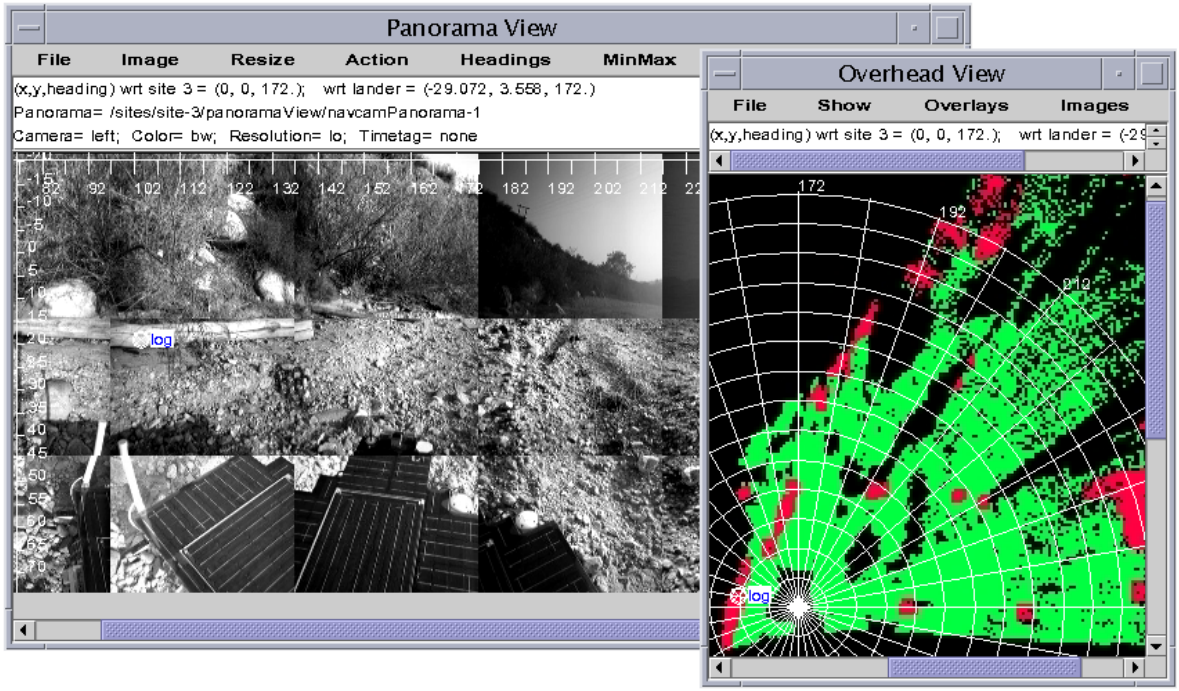


Figure 19. Traversability Visualization

8 The roll of the rover is calculated using the relative heights of the rocker attach points on the rover.

Traverse Planning

Rover traverse planning is a critical part of sequence planning. Rover paths should require minimal time and energy for the traverse while keeping the rover as far as possible from hazardous areas. In the Mars Pathfinder mission, the traversability of the terrain and the best path for the rover were specified by the operator without additional terrain analysis tools [7], [8]. The operator perceived the terrain and planned rover positions in a stereoscopic display of the Martian terrain while wearing stereo goggles. The operator could 'fly' a 3D rover icon through the stereoscopic display. The operator assessed the traversability of the terrain by inspecting the stereo scene and placing the rover icon in various positions within the scene.

While rover traverse planning can be done directly from the operator's perception of the terrain, automated rover traversability analysis and automated path planning are valuable tools to assist in path planning. Automated path planning was available in an earlier version of WITS for the Rocky7 rover [9], but has not been integrated into the FIDO rover version yet.

Automated terrain traversability analysis has been implemented for FIDO operations. As part of the PTeP automated downlink processing for FIDO, data products are produced that overlay terrain and traversability information on images. The elevation-overlay data product displays a color-coded elevation overlayed on an image. This enables the operator to see elevation changes in the scene to help in evaluating the terrain traversability. PTeP also produces a traversability map which is an Overhead view image that shows traversable and non-traversable areas in different colors. A traversability map in an Overhead view using data from the September 2000 field test is shown in Figure 19. Criteria such as obstacle height and slope are used to produce the traversability map.

Rover paths are specified in WITS by creating waypoints, as described above. The waypoints are displayed in all the views. Simulation and state visualization can be used to visualize the planned rover state at each waypoint in all the views.

Hazard zones are areas in the terrain that the rover should not traverse through. Hazard zone specification enables a user to interactively specify hazard zones on the terrain. Hazard collision prediction shows an operator where the currently planned rover path will cause the rover to collide with a specified hazard zone.

A user creates a hazard zone by specifying the 3D center of the zone and then using the Hazard window to create the hazard zone. A basic problem with specifying a point on the terrain in the center of the hazard zone is that the desired point on the terrain is likely to not have terrain information. For example, if the hazard zone is around a boulder, then the point on the ground at the center of the hazard will be occluded by the boulder. The solution offered in this implementation is to specify the 2D center point in the Overhead view and then to specify the height (or Z) component in a special Synthetic Location window. This results in a new 3D selected location in WITS. The user then opens the Hazard window to make the hazard. A hazard zone is defined to be a circle. A user specifies the hazard name and radius and can add text com-

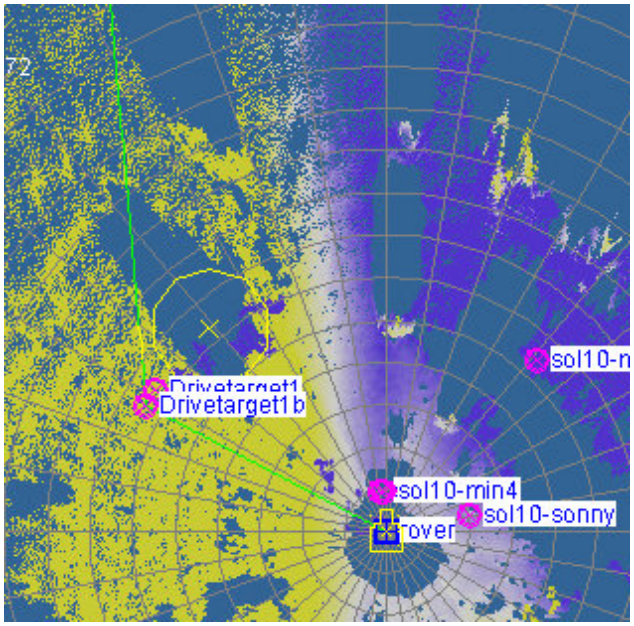


Figure 20. Hazard Collision Prediction

ments about the hazard. The hazard is then drawn as a yellow circle in the WITS views. A hazard is shown in an Overhead view in Figure 20.

Hazard collision prediction determines when a planned rover traverse is likely to cause the rover to collide with a specified hazard zone. A rover traverse is specified by designating waypoints for the rover to traverse through. The planned rover traverse path is the straight line segments between the waypoints. The rover path is shown as green line segments. Collision prediction is done by testing the distance from the rover to all specified hazard zone centers at incremented points along the planned rover path. The difference between the distance from the rover to the center of a hazard zone and the hazard zone radius is computed. This distance is compared to safety distances. Two safety distances are specified: safe and collision. If the difference is greater than the safe distance, then there is no danger and the path is drawn green. If the difference is less than the safe distance but greater than the collision distance, then the path segment is designated risky and colored yellow. If the difference is less than the collision distance, then the path segment is designated as a collision area and colored red. A risky path segment is shown in Figure 20.

The rover traverse path shown in Figure 20 was actually commanded in the May 2000 FIDO field test. The hazard was a bush. The rover collided with the bush at the position predicted by the display (the part of the path colored yellow). Hazard zone specification and hazard collision prediction were not used before the traverse. After the collision hazard zone specification and collision prediction were used to diagnose why the rover collided with the bush, and it became obvious that the reason was that the rover just drove where it was commanded to drive. Use of the hazard collision prediction before commanding the rover could have prevented the collision (a safer path would have been commanded). The rover had automated on-board collision detection and avoidance, but it was developed to detect solid objects like rocks and often fails to detect bushes, as in this case.

Instrument Arm Planning

Instrument arm planning consists of selecting targets on the terrain for the rover's instrument arm to place an instrument on and specifying the instrument commands, as well as planning arm deployment and stowage. Consistent with the plan for the MER mission, it is assumed that all arm motion will be done with the rover in the same position as when it took the images used to plan the arm motion. Arm and arm instrument commands are added to the sequence and edited. Targets are used by name as parameters in the commands. Simulation and state visualization can be used to visualize the planned arm state after each sequence command. Two special arm planning features are provided to assist in arm planning: reachability-overlay image and target reachability analysis.

The reachability-overlay image is automatically produced during downlink data processing by PTeP. It is a front hazcam image with color-coded overlay of areas reachable by the instrument arm. An example of the reachability-overlay image in a Wedge view is shown in Figure 21. With the reachability-overlay image, the operator can see all the possible areas to select a science target on. In Figure 21, the white patch overlays on the front of the rock and on the ground are the reachable areas. The reachable areas on the rock would probably not have been predicted as reachable by an operator without the reachability-overlay image. In the figure, the operator has selected a target in the reachable area on the rock.

Target reachability analysis is provided by the Target Reachability window, as shown in Figure 22. The user selects a target name from a list of targets. If the target is reachable, then target information and the arm joint angles to reach the target are displayed. Also, the arm configuration on the target is shown in the 3D view, as shown in Figure 21. If the target is unreachable, then target information and the reachability violation reason are displayed.

Mast Articulation Planning

Various features are used to simplify planning of mast tasks. Low-level and high-level macros are provided for sequence generation. A low-level command is provided for each command that the rover understands such as commanding joint angles, taking stereo image pairs, and taking an individual IPS spectrum. High-level macros are provided to enable easy specification of complicated tasks such as IPS scans. A high-level macro for the IPS is the IPS_scan macro which causes the rover to take IPS measurements at a grid of points around a target. The central target name, the number of horizontal and vertical points, the horizontal and vertical spacing, and various IPS parameters are parameters for the macro. For the

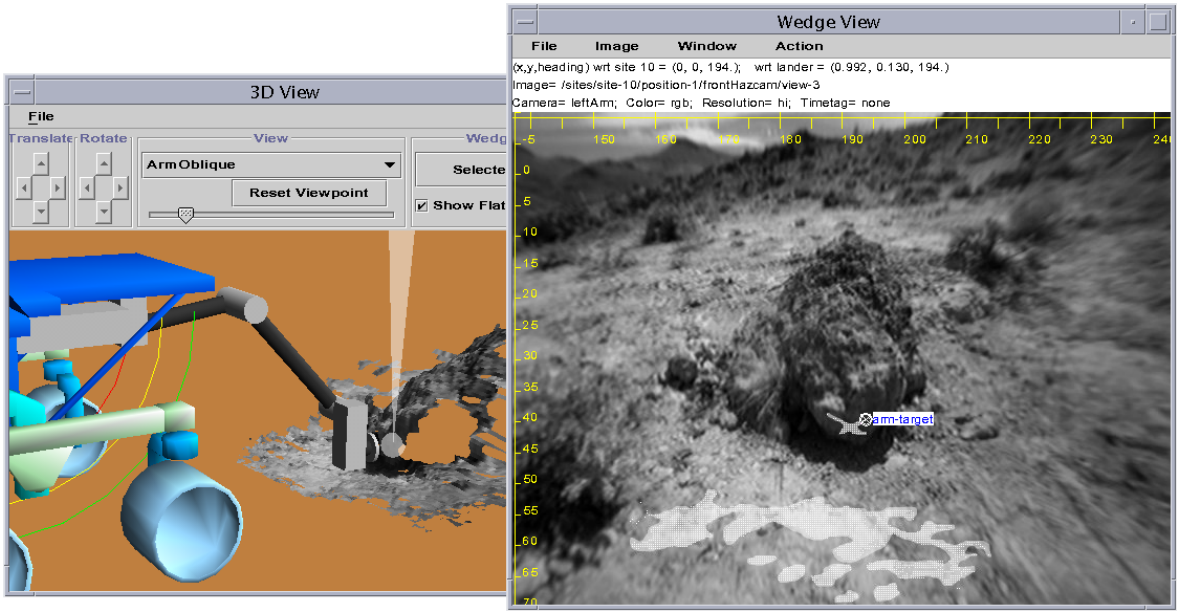


Figure 21. Instrument Arm Planning

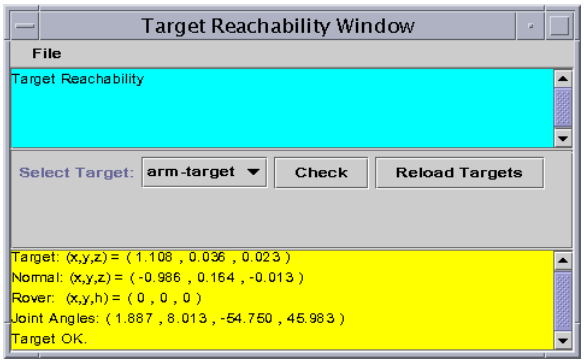


Figure 22. Target Reachability Window

high-level macros, the user inputs intuitive pointing parameters, such as a target name, and the required mast joint angles are automatically computed.

View objects, as described above, visualize the mast tasks. IPS measurement points are shown as small yellow circles, pancam image outlines are shown as yellow footprints, navcam image outlines are shown as blue footprints. Simulation and state visualization can be used to visualize the planned mast state after each sequence command.

Constraints Checking

WITS provides automated resource analysis and rules checking to verify various types of constraints that a command sequence must satisfy. Energy, time duration, and data volume resource utilization were computed for each step of the sequence for the Mars Polar Lander mission. Also, the absolute execution time for each step was computed. For FIDO, only data volume is computed since energy and time models have not been defined yet. Resource utilization is computed using resource models for each low-level command.

Rules checking runs all defined sequence rules. Many types of rules can be defined. Resource constraint rules can check that the actual resources used are within the resource allocations. Time constraints rules can check that commands will execute by a specified time. FIDO has numerous sequence rules such as requiring the mast to be deployed before taking a pancam image or the arm to be deployed before placing the microimager on a target.

Automated sequence repair is provided to automatically repair a sequence which violates sequence rules. Examples are the automatic insertion of arm and mast deployment commands if they are missing. Resource analysis, rules checking, and automated sequence repair are run via menus in the Sequence window.

Group Collaboration

Group collaboration has been implemented in WITS to enable groups of geographically separated users to collaboratively decide on objectives for the rover. The group collaboration features enable distributed users to collaborate as if they were in the same room using the same computer.

Group collaboration features are provided to the user via the Group tab of the WITS Main window, as shown in Figure 23. A user can work independently viewing data products and editing sequences. When the user wants to work with other users to collaborate in viewing data products or generation of a sequence, then the user joins a group. A user can create a new group or use an existing group. A list of all current WITS users, along with the group they are currently in, is shown in the left column of the Group window.

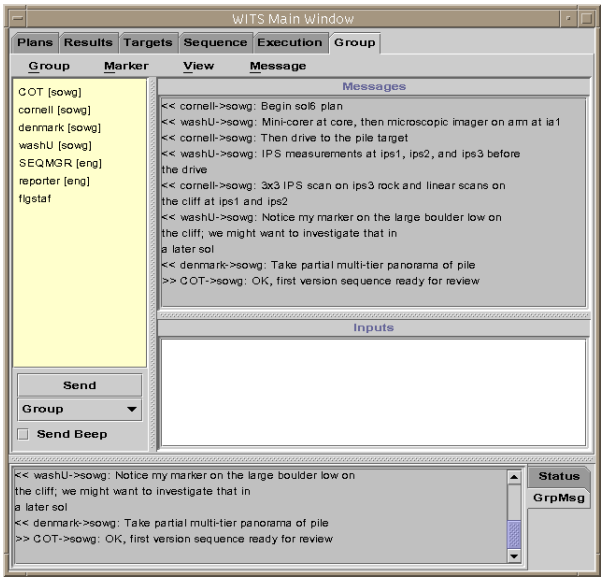


Figure 23. Group Window

To ensure that all group members are viewing the same information, the group view feature was implemented. A view is turned into a group view by selecting the 'Add To Group' item in the File menu of the Panorama, Overhead, Wedge, and 3D views. When a view becomes a group view, it is opened in all group members' clients and the word Group is added to its title.

The Group window 'Marker' menu is used to manage group markers. A marker is a 3D location selected by a group user and displayed on all other group users' client views. Examples are shown in Figure 5. To create a marker, a user first selects a pixel in an image in a Panorama or Wedge view. This location is converted to a marker by selecting 'Add' in the 'Marker' menu. A green triangle is drawn at the location in all Panorama, Wedge, and Overhead views of all group members and the name of the user who created the target is displayed next to the marker.

Group messages are used for textual messaging between group members. It is assumed that during collaboration, telephone-based teleconferencing or Internet telephony are used for verbal communication. All received group messages are displayed in the Messages area of the Group window. To send a group message, a user types the message in the 'Inputs' area of the Group window and then selects the 'Send' button. Who the message is sent to and how it is sent are controlled by the selection in the menu below the Send button. If 'Personal' is selected, then the message will only be sent to the user that has been selected in the users list in the left column of the Group window. The selected user does not have to be in the group that the current user is in. If 'Group' is selected, then the message will be sent to all members of the current group. If 'Announcement' is selected, then the message will be displayed in a pop-up window in all group members' clients. The 'Send Beep' checkbox of the Group window indicates whether the user wants a beep to be sounded on all group members' clients when their message is received.

The server maintains a state for each group. The state includes information on all current client users, all current markers, cumulative group messages, and current group views. When a user joins a group, their client is automatically initialized with the group state, i.e., all the group views are automatically opened, all the markers are displayed, and the group messages are displayed. The group state is also useful for on-going group users. If a group user deletes one of more group views for some reason, then they can automatically reopen them by selecting 'Refresh Group Views' in the View menu of the Group window.

Collaborative Sequence Editing

Collaborative sequence editing is provided to enable users to simultaneously edit a common command sequence. Sequence generation time is reduced by having different parts of the sequence developed in parallel. The collaborating users can be at the primary operations center or distributed over the Internet.

For collaborative sequence editing, ownership of the sequence is specified at the sequence and request levels. The owner of a sequence can modify the structure of the sequence including adding and deleting requests, specifying owners for each request, and specifying resource allocations for requests. The sequence owner cannot modify the macros inside a request that is owned by a different user (although the sequence owner can delete a complete request). Tasks for the different instruments are separated into different requests of the sequence. The owner of a request can add, delete, and modify the macros inside the request, but cannot modify the constraints and resource allocations for the request. A special user called the sequence manager can make any modifications to the sequence.

To collaboratively edit a sequence, a user first checks out the sequence. The sequence is copied from the common server to the user's WITS client. The user then modifies the sequence as appropriate for their ownership. Updating the common server sequence with the user changes is called merging.

To merge a sequence owner's changes to the common sequence at the server, the sequence owner user selects a Sequence window menu item to modify the sequence structure. The original common server sequence is copied and given a version number. Then the structure changes are made to the sequence. All requests in the modified sequence are used in the new sequence. For all requests that were in the original sequence, the macros are copied from the original sequence and placed in the new sequence.

A request owner merges changes to the common server by selecting a menu item to merge their changes to the common server sequence. The original server sequence is copied and

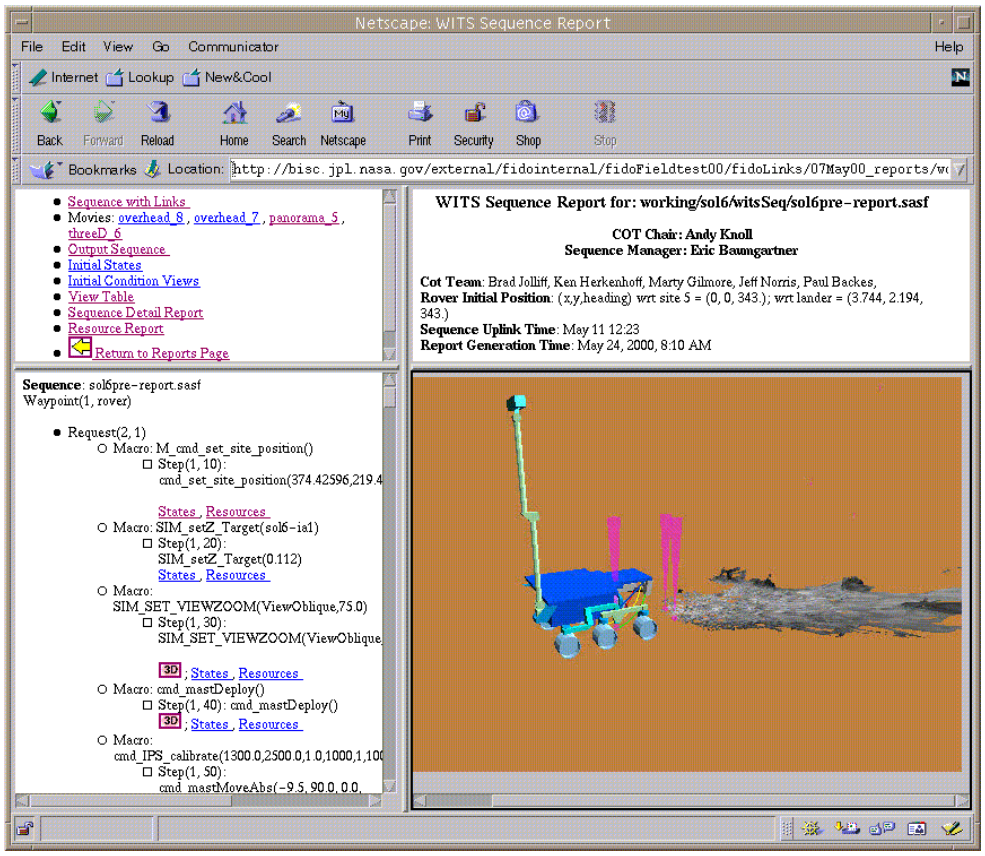


Figure 24. Automatically Generated Sequence Report

given a version number. Then the requests in the original sequence that the current user is owner of are updated with the macros in the modified sequence.

Automated Report Generation

The sequence planning process is not complete with the sequence being uplinked to the rover. Reports need to be produced to document the sequence. An automated sequence report generation capability was developed and integrated into WITS [10]. Sequence reports are automatically generated and provided in a web-browser-based operations report system. An automatically generated sequence report web page from the May 2000 FIDO field test is shown in Figure 24.

The purpose of the automated report generation is to reduce the documentation time for each sequence and to make it easy to access downlink data products and uplink sequence information. Automatic generation of information for the reports reduces the time to generate the documentation, ensures accurate information, and can provide more information than could be generated by hand.

A sequence report provides detailed information for a sequence. The report, consisting of numerous HTML and jpeg image files, is automatically generated via a menu item in the Sequence window. Four frames are used. The upper right frame has static information about the sequence including its name, operators, and rover initial position. The upper left frame has links which control what information is shown in the bottom left frame. The bottom right frame has information as selected in the bottom left frame.

The Sequence with Links page shows the WITS sequence with its hierarchical elements, as shown in Figure 24. Steps have the incremented step labels, commands with parameters, States and Resources links, and links to the view images which relate to the Step. The States link causes the planned rover states at the end of the Step to be displayed in the bottom right frame. The states include rover position and heading, instrument arm angles, and mast angles. The Resources link displays the start time and the resources after the Step including duration, energy, channelized data volume, non-channelized data volume, and total data volume cumulative for the Step, Macro, Request, and Sequence. There are icons for the 3D view, Panorama view, Wedge view, and Overhead view. If a jpeg image of a view was saved for the step, then its icon is listed with the step. The icons are links to the images. Selection of a view icon causes the jpeg image to be placed in the bottom right frame. Figure 24 showns a 3D view selection.

Movies of Overhead, Panorama, and 3D views are provided. Javascript programs are generated for the movies. Selection of the desired movie in the upper left frame causes the movie to the played in the bottom right frame.

Selection of the Output Sequence link causes the output sequence to be displayed in the bottom right frame. Selection of the Initial States link causes a complete list of the rover's initial states to be displayed in the bottom right frame including all states downlinked from the rover. Selection of the Initial Condition Views link causes the jpeg images of all the WITS open views, showing the rover at its state at the beginning of the sequence, to be displayed in the bottom left frame at a constant width. Selection of one of images causes the full-size image to be displayed in the bottom right frame.

Selection of the View Table link causes all of the stored jpeg images of the views to be displayed in the bottom left frame separated by their associated Steps. Selection of one of the images causes the full-size image to be displayed in the bottom right frame. Selection of the Sequence Detail Report link causes the sequence with all of its hierarchical elements to be displayed in the bottom left frame along with view images at each step. Selection of the Resource Report link causes the detailed resource report to be displayed in the bottom right frame.

5. CONCLUSIONS

The FIDO ground operations system incorporates advanced operations technologies to make rover sequence generation efficient. The Parallel Telemetry Processor (PTeP) efficiently processes downlink telemetry to generate numerous data products. The Multi-mission Encrypted Communication System (MECS) efficiently distributes database updates to Internet-based participants and handles communication with Internet-based WITS clients. The Web Interface for Telescience (WITS) enables efficient distributed downlink data visualization and rover sequence generation.

ACKNOWLEDGEMENTS

The work described in this paper was funded by the Mars Exploration Technology Program, the TMOD Technology Program, and the SBIR Program, and performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

REFERENCES

[1] R. Arvidson, Paul Backes, E. Baumgartner, D. Blaney, L. Dorsky, A. Haldemann, R. Lindemann, P. Schenker, and S. Squyers. FIDO: Field-Test Rover for 2003 and 2005 Mars Sample Return Missions, 30th Lunar and Planetary Science Conference, March 15-19, 1999, Houston, Texas.

[2] 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, Proceedings IEEE International Conference on Robotics and Automation, April, 1997, pages 2458-2464, Albuquerque, New Mexico.

[3] Paul G. Backes, Gregory K. Tharp, and Kam S. Tso. The Web Interface for Telescience (WITS), Proceedings IEEE International Conference on Robotics and Automation, April, 1997, pages 411-417, Albuquerque, New Mexico.

[4] Paul G. Backes, Kam S. Tso, Jeffrey S. Norris, Gregory K. Tharp, Jeffrey T. Slostad, Robert G. Bonitz, and Khaled S. Ali. Internet-Based Operations for the Mars Polar Lander Mission, Proceedings IEEE International Conference on Robotics and Automation, April, 2000, pages 2025-2032, San Fransicso, California.

[5] Jeffrey S. Norris, Paul G. Backes, and Eric T. Baumgartner. PTeP: The Parallel Telemetry Processor, Proceedings IEEE Aerospace Conference, March, 2001, Big Sky, Montana.

[6] Jeffrey S. Norris and Paul G. Backes. WEDDS: The WITS Encrypted Data Delivery System, Proceedings IEEE Aerospace Conference, March, 2000, Big Sky, Montana.

[7] B. Cooper. Driving On The Surface Of Mars Using The Rover Control Workstation, SpaceOps98 Conference, June, 1998, Tokyo, Japan.

[8] Andrew H. Mishkin, Jack C. Morrison, Tam T. Nguyen, Henry W. Stone, Brian K. Cooper, and Brian H. Wilcox. Experiences with Operations and Autonomy of the Mars Pathfinder Microrover, Proceedings IEEE Aerospace Conference, March, 1998, Snowmass, Colorado.

[9] Paul G. Backes, Gregg Rabideau, Kam S. Tso, and Steve Chien. Automated Planning and Scheduling for Planetary Rover Distributed Operations, Proceedings IEEE International Conference on Robotics and Automation, May, 1999, pages 984-991, Detroit, Michigan.

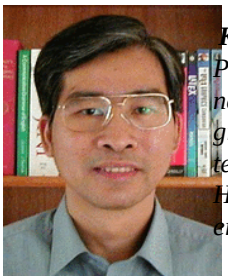
[10] Paul G. Backes and Jeffrey S. Norris. Automated Rover Sequence Report Generation, Proceedings IEEE Aerospace Conference, March, 2001, Big Sky, Montana.

Paul Backes *is a technical group leader in the Mobility Systems Concept Development section at the Jet Propulsion Laboratory, Pasadena, CA, where he has been since 1987. He received a 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.

Jeffrey S. Norris *is a computer scientist and member of the technical staff of the Mobility Systems Concept Development section at the Jet Propulsion Laboratory. At JPL, his work is focused in the areas of distributed operations for Mars rovers and landers, secure data distribution, and science data visualization. Currently, he is a software engineer on the Mars Exploration Rover ground data systems and mission operation systems teams. Jeff 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 lives with his wife, Kamala, in La Crescenta, California.*

Mark W. Powell *has been a member of the technical staff in the Mobility Systems Concept Development section at the Jet Propulsion Laboratory, Pasadena, CA, since 2001. He received his B.S.C.S. in 1992, M.S.C.S in 1997, and Ph.D. in Computer Science and Engineering in 2000 from the University of South Florida, Tampa. His dissertation work was in the area of advanced illumination modeling, color and range image processing applied to robotics and medical imaging. At JPL his area of focus is science data visualization and science planning for telerobotics. He is currently serving as a software and systems engineer, contributing to the development of science planning software for the 2003 Mars Exploration Rover mission and the JPL Mars Technology Program Field Integrated Design and Operations (FIDO) rover task. He, his wife Nina, and daughters Gwendolyn and Jacquelyn live in Tujunga, CA.*



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.*

Gregory 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.*

P. Chris Leger *After realizing he was unlikely to visit Mars in person, Dr. Leger decided that the next best thing would be to travel there vicariously through a robot. With that in mind, he received a BS in Computer Engineering from Carnegie Mellon in 1995, and MS and PhD degrees in Robotics from CMU in 1997 and 1999. He now works on the Mars Exploration Rover mission at the Jet Propulsion Laboratory. His other passions include rock climbing, playing bass, luthiery, and writing.*