The Science Activity Planner for the Mars Exploration Rover Mission: FIDO Field Test Results

Paul G. Backes*, Jeffrey S. Norris*, Mark W. Powell*, Marsette A. Vona*, Robert Steinke*, and Justin Wick[†]
*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California
[†] Cornell University
Paul.G.Backes@jpl.nasa.gov

Abstract—The science activity planning process for the 2003 Mars Exploration Rover (MER) mission was exercised in a ten day terrestrial rover field test in August 2002. A version of the MER mission Science Activity Planner (SAP) tool was used for downlink data visualization and uplink plan generation. The Field Integrated Design and Operations (FIDO) rover was at an undisclosed location in the Arizona desert. Data from the rover was sent via satellite to scientists and engineers at JPL who used SAP to generate uplink sequences.

Figure 1. Field Test Operations Architecture

TABLE OF CONTENTS

1. INTRODUCTION

Two rovers will land on Mars in January and February of 2004 as part of the Mars Exploration Rover (MER) mission. About fifty scientists will be participating in the commanding of each rover. New operations technologies will be used to maximize the influence that the scientists have on the rover operations. The Science Activity Planner (SAP) will be the primary science operations tool in the MER mission for scientists to visualize downlink data and specify desired uplink activities.

SAP is the name of the adaptation of the Web Interface for Telescience (WITS) that will be used for the MER mission. WITS was the primary command generation tool for the Mars Polar Lander mission robotic arm and robotic arm camera [1]. Unfortunately, communication with the MPL lander on the Martian surface was not achieved, so commanding the lander was not possible. WITS has also been used to command the terrestrial Field Integrated Design and Operations (FIDO) rover in field tests in 2000 and 2001 [2]. The many

new science operations requirements for the MER mission resulted in a major redesign and new development of the WITS tool. WITS is run as a Java application on Linux, Windows, and Solaris operating systems. WITS is implemented using Java2D, Java3D and Java Advanced Imaging.

The updated version of WITS was used in a ten day field test of the FIDO rover in August 2002. The rover was at an undisclosed location in an Arizona desert and was commanded from JPL by about fifty MER mission scientists. The field test served to introduce scientists to planetary rover operations, to test MER mission science operations processes, and to test the capabilities of the updated WITS/SAP tool. This paper describes the ground data system for the FIDO rover field test and the capabilities and design of the updated WITS/SAP tool as used in the field test.

The FIDO rover is a prototype Mars rover similar to the rovers that will land on Mars in the MER mission [3]. Stereo cameras on the FIDO rover's mast and body are used to image the surrounding terrain. The Pancam camera pair on the mast provides color images. The Navcam camera pair on the mast provides monochrome images. The front and rear Hazcam camera pairs mounted at the front and rear of the the rover's body provide monochrome images. 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.

2. FIELD TEST OPERATIONS ARCHITECTURE

The field test operations architecture is shown in Figure 1. The rover was at an undisclosed desert location. Satellite

Figure 2. Geochemistry/Mineralogy Theme Group using WITS

communications were provided for communication between the operations center at JPL and the rover in the desert. Downlink data from the rover was processed by the Parallel Telemetry Processor (PTeP) system [4] which placed resulting data products in the primary database. The database was as structured file system. When all data for one downlink session was complete, the Multimission Encrypted Communication System (MECS) automatically distributed the new data products to all client computers [5]. All client computers had copies of the data products from the primary database that were needed by their client WITS applications. For the field test there were eight client computers each running MECS and WITS clients (the computers were Linux workstations). One of the WITS client applications was used for final uplink command sequence planning and it produced the command sequence that was sent to the rover via the satellite communication. For the field test all WITS clients were at JPL, but architecturally this was not necessary. WITS clients could have been running anywhere on the Internet and MECS clients would have provided the needed data distribution capabilities.

3. OPERATIONS PROCESS

The roles for people participating in the field test are briefly described below.

Mission Planner: Provided mission resource availability information, e.g., data volume.
Mission Manager: Provided high level field test management including both engineering and science issues. Did not get information from the field beyond normal downlink data.
Test Director: Provided overall direction for the field test. Made sure that rules of field test were followed. Was the only person besides the engineering team that was in communication with people in the field.

SOWG Chair: Led the SOWG meeting.
SOWG Documentarian: Took notes during the SOWG meeting.
Theme Group Documentarian: Took notes during the theme group meetings.
Theme Group Member: A scientist in a theme group.
Rover Uplink Lead: Performed uplink sequence generation.
Rover Engineering Lead: Lead person for the engineering team.
Rover Engineering Team: Engineers analyzing the health and status of the rover.
WITS Support Staff: WITS experts who answered scientists' questions on use of the WITS tool.
SOWG WITS Operator: Operated WITS during the SOWG meeting.
Human Factors Observer: Observers who took notes on human factors issues.
Headquarters Representative: Representative from NASA headquarters, e.g., to decide whether mission success criteria were met.

The science team was organized into the following theme groups: Soil, Atmosphere, Long Term Planning, Geochemistry/Mineralogy, and Geology. Each theme group had a workstation running a MECS and a WITS client, as indicated in Figure 1. A photograph of the Geochemistry/Mineralogy theme group using WITS is shown in Figure 2. All of the scientists from all theme groups made up the Science Operations Working Group (SOWG).

Separate from the science team was the engineering team, WITS support staff, mission manager, test director, and others listed above. The engineering team managed the health and status of the rover and provided information on the health and status of the rover to the science team. The WITS support staff answered scientists' questions about the use of WITS

and provided the WITS operator during the SOWG meeting. The scientists operated WITS on the theme group workstations.

The SOWG meeting area is shown in Figure 3. At the bottom left of the picture is an engineer from the engineering team. At the right are two WITS support staff with the person in front of the two monitors being the SOWG meeting WITS operator. At the head of the center SOWG table is the SOWG chair. The Soil theme group is on the far left and the Atmospheric theme group is at the far right beyond the WITS operator. During the SOWG meeting, scientists from the other theme groups came from the adjoining room where the other three theme groups were located and joined the rest of the operations team for the SOWG meeting.

The field test ran for ten consecutive days. Two equivalent MER mission sols of operations were performed per day. One sol is one Mars day. For each sol, the following operations steps were performed in the order shown and with the indicated durations in minutes.

15 min, Uplink Sequence Review: In the SOWG meeting room, the WITS operator explained the command sequence that was uplinked to the rover for the previous sol by displaying the sequence on the projection screens and stepping through the sequence. The rover configuration at each step of the sequence was displayed.

45 min, Downlink Assessment: The DTE data was available at the start of this process step. The downlink data was analyzed and scientists began developing rover activities in their theme group plans using WITS.

15 min, Science Assessment Meeting: All scientists met in the SOWG meeting area to discuss the state of the rover and what types of activities they wanted the rover to perform on the next sol. The purpose was to get agreement on the types of activities the rover would perform.

15 min, Prep for SOWG Meeting: Scientists returned to their theme groups and finished their theme group plans by specifying desired science activities in their theme group plans.

45 min, SOWG Meeting: All scientists met in the SOWG meeting room to generate one SOWG activity plan. Activity plans from all theme groups were merged using WITS to generate the one output SOWG plan. The WITS operator performed the activity and plan editing as requested by the SOWG chair.

15 min, Sequencing Team Meeting: A subset of scientists and engineers met in the uplink room to review the SOWG activity plan.

60 min, Build and Validate Sequence: A limited number of scientists and engineers in the uplink room used WITS to refine the SOWG activity plan to generate the uplink command sequence to be sent to the rover.

14 min, Review and Approve Sequence: The uplink command sequence was reviewed and approved.

1 min, Uplink Sequence: The sequence was sent to the rover via the satellite communications.

90 min, FIDO Rover Field Operations: The rover autonomously executed the uplinked command sequence.

Two types of downlink communication from the rover were used to simulate the two types of communication that will be available for the MER mission: DTE and UHF. DTE is direct to Earth communication where the rover communicates directly to Earth. UHF is where data is relayed to Earth via a satellite orbiting Mars. DTE and UHF data were downlinked from the rover and made available to the operations team at the following times for each sol.

0 min: DTE data available
120 min: UHF data available

Additionally, there was one hour-long science meeting at the end of each actual day. The equivalent meeting will occur for each sol during the actual MER mission, but there was not time to have the full meeting for each sol during the field test.

The operations process shown here is similar to the process that is planned for the MER mission, but there are differences due to the reduced operations time per sol and the decision to not exercise in detail all of the operations steps. The test focused on the science operations process that includes the process steps through the SOWG meeting. The steps after the SOWG meeting will use different tools for actual MER operations than were used in the field test.

4. DOWNLINK DATA PROCESSING AND DATA DISTRIBUTION

The PTeP tool was used to process downlink data and produce resulting data products in the database [4]. Data products sent from the rover were called Engineering Data Records (EDRs). The resulting data products that PTeP generated from EDRs were called Reduced Data Records (RDRs). An example of an EDR is a stereo image pair from the Navcam cameras on the rover mast. Example RDR products that were derived from the an EDR are the extracted left and right images, the updated camera models, range maps, and 3D terrain maps.

5. DOWNLINK DATA VISUALIZATION

Interaction in WITS is done within WITS browsers. The browsers have a selection tree on the left and a work area on the right. There are two primary browsers: the Downlink Browser and the Uplink Browser. These browsers automatically open when a user logs into WITS. The Downlink Browser is shown in Figures 4, 5, 6, and 7. The Uplink browser is shown in Figures 8 and 9 and described in Section 7.

The Downlink Browser is used to select and view downlink data products. Just like a Web browser can have a list of bookmarks on the left side of the window and the remaining space is for viewing a Web page, the downlink browser arranges links to data products in a tree on the left and creates a view

Figure 3. SOWG Meeting

on the right when a link is selected.

Select and Open a Data Product

The downlink selection tree on the left of the downlink browser contains all of the data products in the rover database. The tree is arranged initially by site, position, and instrument name. The tree may be reorganized by selecting the *Browser⇒Configure Tree* menu, and choosing from several combinations of sol, site, position, and instrument ordering.

Clicking on the turnbuckle icon to the left of a folder will expand the folder into the folders and data products that it contains. Double-clicking on a data product in a folder opens the selected data in its default view. Right-clicking on a data product causes a popup menu to appear that enables selection from all of the available types of views for that data. For example, right-clicking on an image allows opening it in the Image view, Image Cube view, or 3D view.

Data products from instruments like the Microscopic Imager are labeled as "set-0", "set-1", etc. Image data products that are parts of a panorama are named "wdg-0", "wdg-1", etc., where "wdg" is short for "wedge," or a section of a panorama. Image data products that are from the Navcam or Pancam that are not part of a panorama are labeled "view-0", "view-1", etc., since these are individual observations.

The collections folders in the tree contain groups of data products, for example all of the images in a panorama. When collection data products are viewed, all the data in the collection is shown in a single view, such as the Panorama view or the 3D view.

Configure the Downlink Browser

Selecting data products from the tree to view causes them to appear in the area on the right called the view grid. At first, the view grid takes up all the space in the browser except for the tree. The tree can be hidden by dragging the bar that separates it from the view grid to the left. The topology of the grid can be changed by partitioning the viewing area into several parts by selecting a topology option from the *Browser⇒View Grid Topology* menu, at any time. There are various topologies to choose from, such as 1 by 1, 2 by 2, and 1 over 2. When the view grid has multiple view panes, each new view that is opened will appear in one of the empty panes. When there are no more empty panes, the least recently used pane will be re-used, and any view that was there will be sent to the Trash tab (described below). If all the views are occupied and it is desired to open a view in a specific pane, then clicking the "X" icon at the upper right of the pane will close the pane and the next opened view will be displayed in that pane.

At first, the downlink browser has two tabs: "Tab 1" and "Trash". Tabs are useful for organizing groups of views. A new tab is created by right-clicking on a tab area above the view grid and selecting *Insert New Tab*. Clicking on a tab causes all the views in that tab to be displayed. Tabs can also be renamed.

Figure 4. Downlink Browser with Navcam Panorama, Overhead, and 3D views
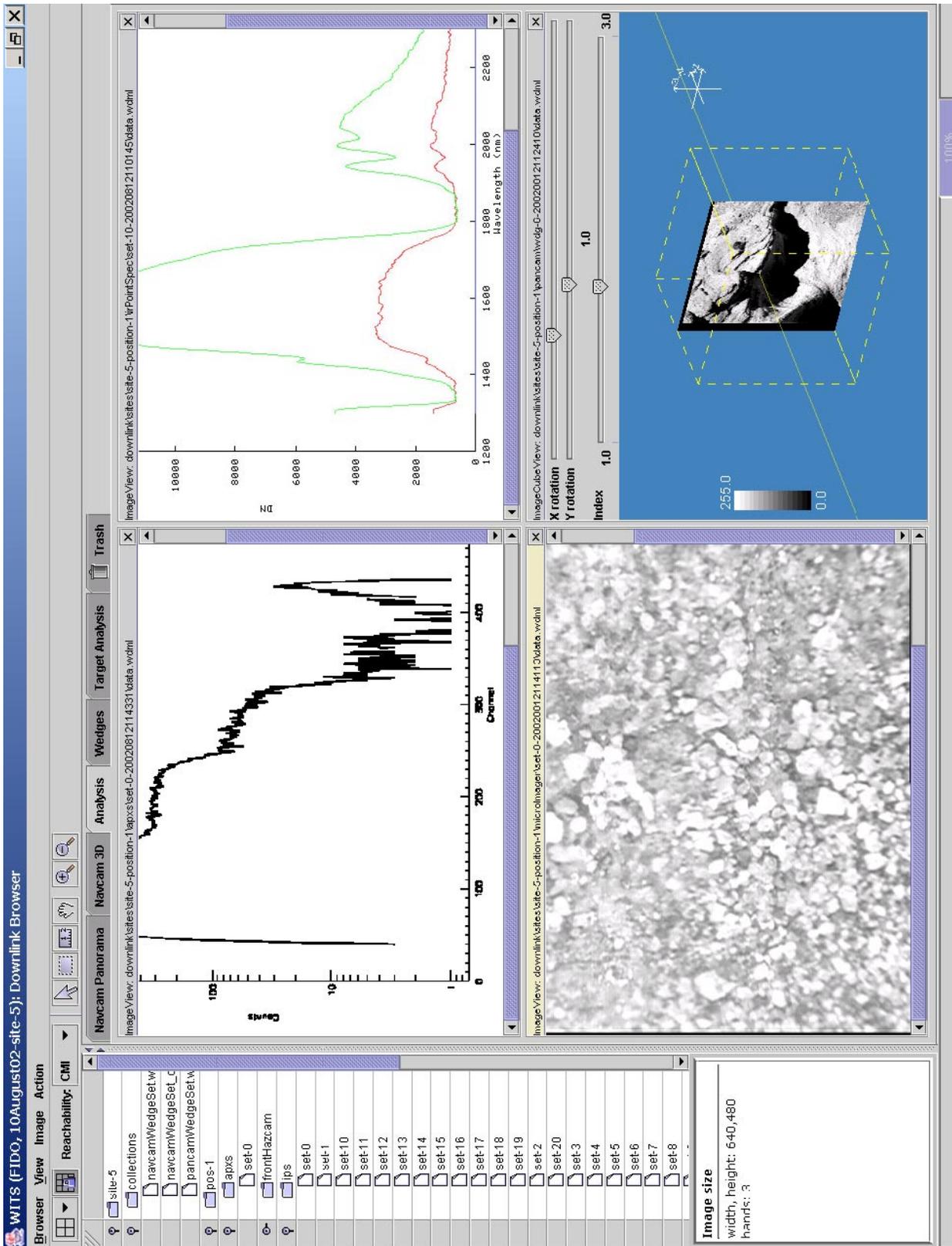
Figure 5. Downlink Browser with APXS, IPS, Microscopic Imager, and Pancam Image Cube Views
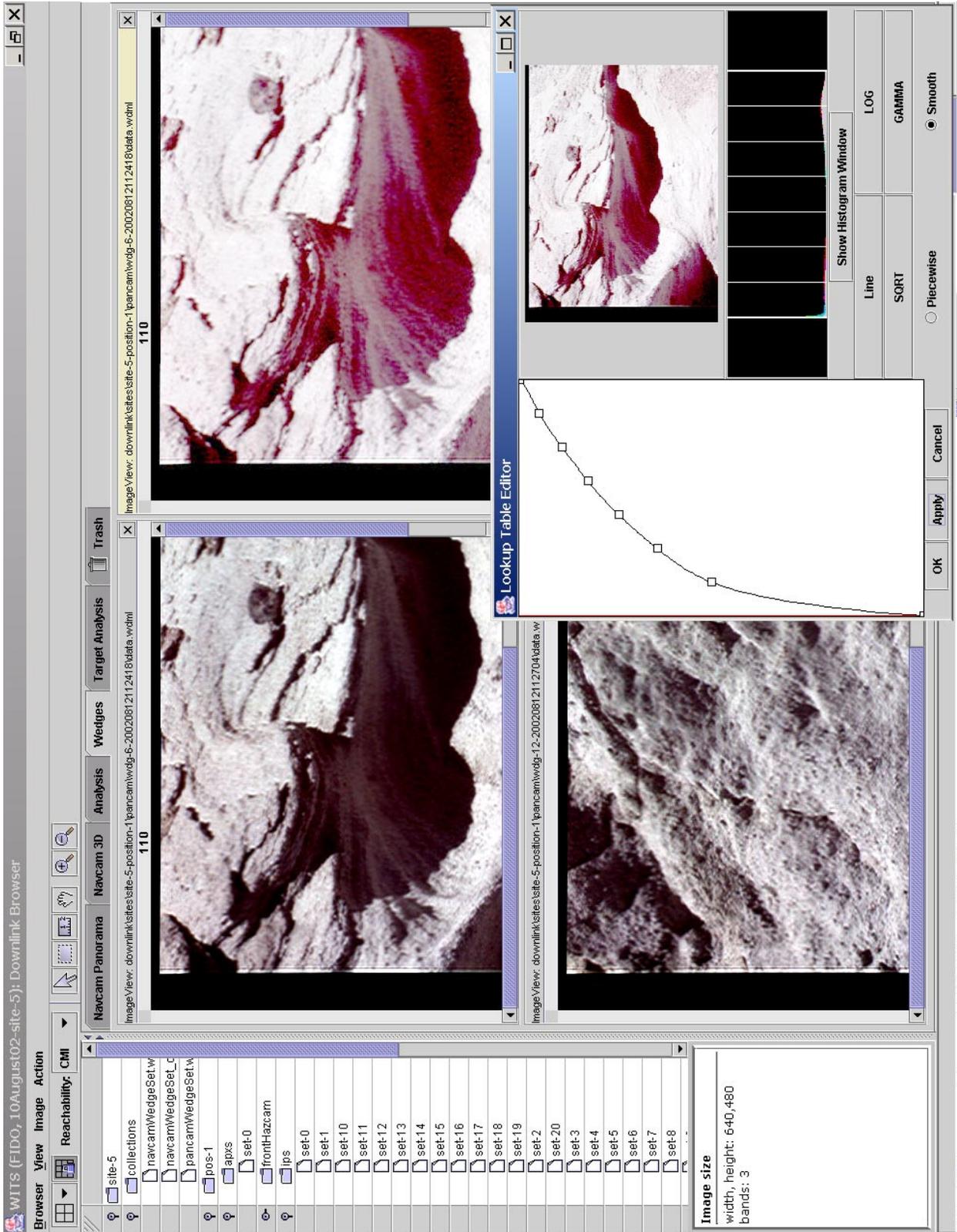
Figure 6. Downlink Browser with Pancam Images in the Image view, Contrast Adjuster Dialog, and the Pancam Image Modified using the Contrast Adjuster Dialog
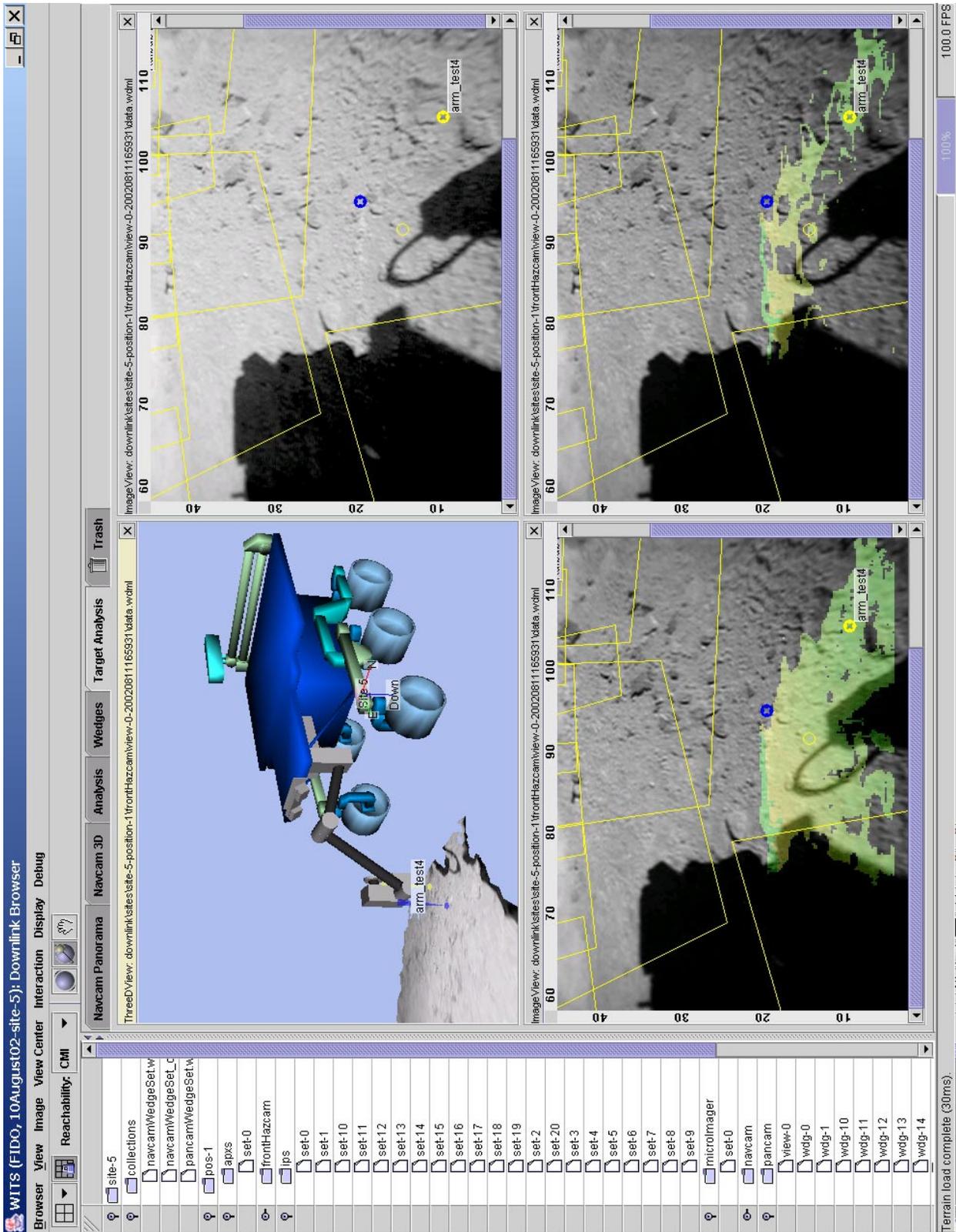
Figure 7. Front Hazcam Image in 3D View and Image View Showing Original Image and Original Image with Mossbauer (Left) and CMI (Right) Reachability Overlays

The Trash tab is a special tab: it is where views go temporarily when they are closed or replaced by new views. The Trash tab holds onto the four most recently closed or replaced views, and when the Trash tab fills up, the oldest views are discarded.

View Types

Different types of views are provided for viewing data. The Panorama view, shown in Figure 4, automatically reads in a collection of images and mosaics them together [6]. Azimuth and elevation angles are displayed. Many features are provided with the Panorama view such as zoom in and out, conversion to anaglyph stereo, and image processing capabilities such as median, low and high pass, gaussian, and edge filters. Glyphs are drawn on the Panorama view representing different planning information. The yellow lines in Figure 4 are footprints representing commanded images in the current plan.

The Image view is used to display a single image, e.g., one image of a stereo pair or an individual image. Figure 5 shows a Microscopic Imager instrument (located at the end of the robot arm) image. Figure 6 shows left images from the Pancam camera pair. Figure 7 shows Front Hazcam images in the Image View. The Image view provides anaglyph and image processing capabilities of the Panorama view.

The 3D view displays the 3D terrain and rover, as shown in Figure 4 [7]. Targets, features, and clicked points are also displayed. The 3D view automatically loads terrain segments from a collection of images, e.g., the collection of images in a panorama. Automatic level of detail switching improves display performance. Various navigation capabilities are provided.

The Pancam Image Cube view, shown in Figure 5, enables a user to quickly switch between the various bands in a multi-spectral image. For the FIDO rover there were three filters in the Pancam so images from the three filters are loaded into the Pancam Image Cube view. The view is opened for a Pancam image by right clicking on the image in the downlink selection tree and then selecting the Pancam Image Cube view.

The Contrast Adjuster dialog is used to perform some types of stretching operations on an image or collection of images. It can be used for an image in an Image view, as shown in Figure 6, or for a whole panorama in a Panorama view. It shows a histogram of the image and allows for selection of various types of stretching operations including Line, LOG, SQRT, and GAMMA. The result of the stretch operation is shown at the top of the Contrast Adjust dialog and the operation can be applied to the specified image or collection by selecting the Apply button. In Figure 6, the upper left Pancam image is the original image and the upper right image is the resulting image after the stretch operation from the Contract Adjuster dialog has been applied to it.

6. FEATURE AND TARGET SELECTION

To command the rover to drive to or place an instrument on a particular location in the terrain, targets are created. Targets are 3D locations on the terrain that are selected from stereo image pairs that come from the Hazcam, Navcam or Pancam instruments. Features are also 3D locations on the terrain, but they are not used as parameters in activities. Features represent objects in the terrain and targets are associated with features.

After clicking on an image in an Image view or Panorama view, WITS will try to look up the range of the point in the image. If there is range for the point, then a circular blue annotation, or glyph, is drawn at the clicked point and also in every other open view that contains that point. Glyphs are annotations that are drawn on top of images, like the blue circle, targets, and image footprints.

If a selected point has range data, then a target or feature can be created there via the *Action⇒Add Target* and *Action⇒Add Feature* menu items. A dialog prompt will appear for entering a name for the target or feature. For targets, the user also associates the target with a feature by selecting a feature from a pull-down list of features in the dialog prompt. Targets and features appear listed in the Targets view of the Uplink Browser.

In order to perform target selection for instrument arm placement, a user needs to know what areas on the terrain an instrument can reach. A user does this by displaying a Front Hazcam image in an Image view, as shown in Figure 7. Then, via an Image view pull-down menu, the user can select to overlay the areas that a specific instrument can reach. The bottom left image in Figure 7 shows the overlayed area where the Mossbauer instrument can reach and the bottom right image shows where the Color Microscopic Imager can reach. (The Mossbauer instrument was not actually on the rover arm; it was modeled in WITS to simulate a multi-instrument arm as will be available in the MER mission.) The user can then select a target in the color-overlayed area and know that the instrument should be able to be placed on the target.

7. UPLINK PLAN GENERATION

The Uplink Browser is used to create and edit activity plans, as shown in Figures 8 and 9. Activity plans consist of targets, observations, and activities, and are stored in the Rover Markup Language (RML) format, which is based on XML.

The left side of the Uplink Browser is an uplink selection tree that allows the user to load a previously saved RML plan. This selection tree is organized by sols and theme groups. A plan is opened from the selection tree by double-clicking on it.

The right side of the Uplink Browser is a WITS view grid. Like the Downlink Browser's view grid, the topology of this

Figure 8. Uplink Browser with Details Dialog and Three Plans Loaded
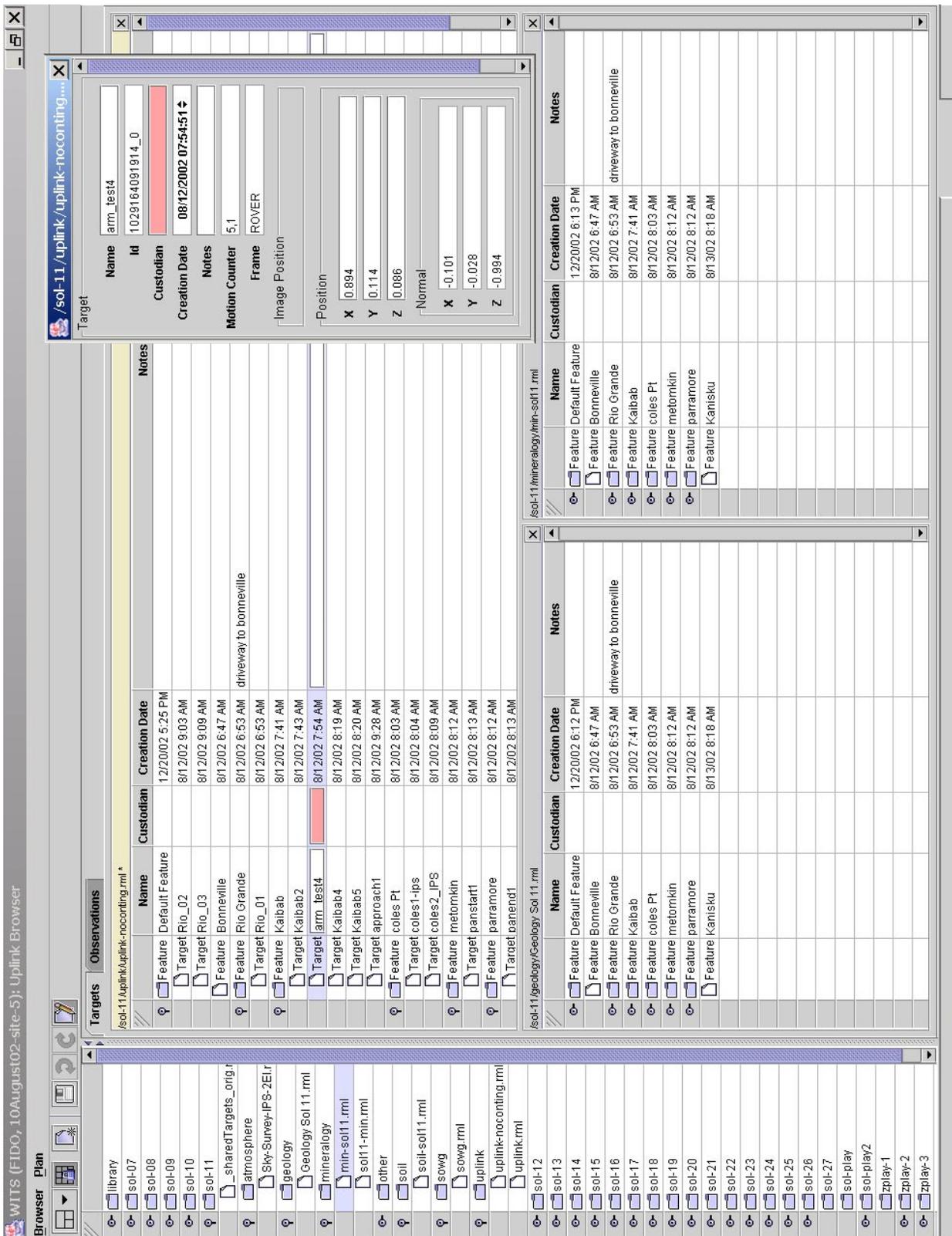
Figure 9. Uplink Browser with Targets Tab Selected

area can be arranged in a variety of ways by clicking on the first (furthest-left) icon on the toolbar and selecting a topology. By default, the view grid is configured to show just one plan at a time. The selected topology shown in Figure 8 has one view pane on top and two view panes below. Different plans can be opened in the different view grid view panes.

The tabs above the view grid allow the user to view the different sections of the currently open plans. Currently there are two tabs: "Targets" and "Observations," but more tabs will be added in the future. These tabs change the currently viewed section for all open plans. If the user clicks on the targets tab, the targets defined for all plans will be displayed. If the user selects the observations tab, then the observations of the plan will be displayed. The arrangement of the plans in the view grid stays the same on different tabs, so the top area will always represent the same plan, regardless of what tab is currently selected.

When the user clicks within a plan in the view grid, its title bar turns yellow and that plan becomes the currently selected plan. When a plan is selected, its glyphs are displayed in the downlink views.

When the user clicks on an item within a plan (for instance, an activity), details on that item are displayed in a smaller floating window called the Details Dialog (see Figure 8). The details dialog allows the user to edit attributes of the currently selected item. The Details Dialog can be shown or hidden using the icon on the Uplink Browser toolbar that looks like a piece of paper and a pencil.

The targets tab in the Uplink Browser shows the targets that have been defined for the currently open plans. Targets are built within downlink views, as described in Section 6, but can be viewed and edited in the Uplink Browser. The targets listed in the targets tab can be used as arguments to activities built within that plan. Target glyphs are displayed in the downlink views for the targets in the currently selected plan.

Activity Plan Editing

Activity plans are built within the Observations tab of the Uplink Browser. A user can load a previously stored plan by double clicking on it in the uplink selection tree at the left. A new plan can be created by selecting *Browser⇒New Plan* in the Uplink Browser. The hierarchy of the activity sequence in a plan is shown below.

- Plan
  - Observation
    * Activity
      ○ Expanded Activity

New observations are added to the plan by selecting *Entry⇒Build New⇒Observation*. An observation can be moved in a plan by clicking and dragging the observation.

When the observations tab is selected, the activity dictionary appears in the bottom left corner of the Uplink Browser. The activity dictionary has all of the available activities that can be added to an observation. The activities are organized into various sets. A pull-down menu above the activity dictionary (to the right of "Instrument:") enables selection of the the various sets of activities. The idd set is selected and displayed in Figure 8. An activity is added to the currently selected plan by selecting where in the plan to insert it and then double clicking on the activity in the activity dictionary list. To specify the parameters for an activity, a user selects the activity in the plan to cause its parameter list to be displayed in the Details Dialog window. The user then specifies the parameters for the activity. Figure 8 shows the CMI_Wedge_Step_Down activity selected and its list of parameters displayed in the Details Dialog window. One of the parameter is the Location parameter with the arm_test4 target selected as its value. Selecting the arm_test4 parameter box causes the list of available targets to be displayed there and the user can select another target for the Location parameter value.

The activity dictionary is stored as a file (in an XML format) and contains activity definitions including activity parameter definitions, default parameters, and expansions for activities. Activity expansions define how a higher level activity expands into one or more lower level activities. Higher level activities are provided to enable users to plan activities at as high a level as possible. Planning using the lowest level activities, the activities that are actually sent to the rover, would be very cumbersome to users. Users can insert high and low level activities, as needed. Activity expansions can be displayed by selecting *Plan⇒Show Expansions*.

Plan merging is accomplished using observation and activity click and drag. Observations and activities can be copied from one plan into another plan by clicking on an element and dragging it into another plan. When an observation is dragged from one plan into another plan, all of its activities are copied with it. In the SOWG meeting, each science theme group provides their own activity plan and the plans from all theme groups are merged into one SOWG meeting output plan. Figure 8 shows the SOWG plan at the top and the Geology and Mineralogy theme group plans below. The theme group observations and activities can be dragged into the SOWG plan for plan merging.

The click and drag feature is also used when copying observations and activities out of library plans. The library plans are stored in the theme group libraries under the library element in the uplink selection tree, as shown in Figure 8. A library plan generally has observations that are templates of types of tasks that a user expects to have the rover do in the future. A user can load a library plan and update the observations in it and save the changes back into the library. A user can also load a library plan and then drag observations and activities out of the library plan into the plan that they are generating for that sol. Users can quickly generate sophisticated obser-

Figure 10. Resource Plot

vations in this way.

Activity Glyphs, Simulation, and Resource Modeling

The activities in the plan are visualized in the Panorama and Image views when possible using activity glyphs. For imaging activities, the activity glyphs are footprints on the terrain where the images will be taken. Glyphs for the images in the current plan are shown in Figure 4 with yellow outlines. Single image footprint glyphs are shown and a series of glyphs are shown for a planned Pancam panorama. Glyphs for a vertical IPS scan are also shown in Figure 4. The IPS glyphs are yellow circles with the same angular extent that the IPS has.

State simulation is also provided to assist in evaluating a plan. In state simulation, when the user selects an activity in the plan, the rover position and configuration is updated in the 3D view with the predicted rover state at the end of that activity. In Figure 4, the rover position and configuration is shown for when the user selected the Move_Inst_Standoff activity in the plan of Figure 8. A user can quickly click through the plan to get an idea of what the rover is going to do.

The rover has very limited resources when executing the up-linked sequence. Resource modeling is provided in WITS to compute the time duration, energy, and DTE and UHF data volume resources that will be consumed by the rover while executing the sequence. The user can plot these resources as shown in Figure 10 (plotted as a percentage of total available). This plot was used in the SOWG meeting to ensure that a plan did not require more resources than were available. The plot was used again in the Build and Validate Sequence step to ensure that the sequence to be uplinked did not require more resources than would be available to the rover.

8. RESULTS

The science team was able to command the FIDO rover using the WITS/SAP tool to achieve all of their field test success criteria. With the science team at JPL and the rover at the undisclosed desert location, they successfully acquired a 360 degree color panorama, traversed 202 meters, performed mast-based remote sensing using color stereo imaging and the IPS, performed instrument-arm based in situ measurements, and excavated two trenches using rover wheels. Lessons learned from the field test will be used to further improve the tool for the MER mission.

9. CONCLUSIONS

Many visualization and activity planning capabilities for rover mission operations have been developed as part of the WITS tool. These capabilities were successfully used to command the FIDO rover in a ten day desert field test. Results of the field test indicate that SAP, the MER mission version of WITS, will provide the needed downlink data visualization and uplink science activity planning needed by the MER mission science team for rover science operations.

ACKNOWLEDGEMENTS

REFERENCES

[1] 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. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 2025–2032, San Francisco, California, April 2000.

[2] Paul G. Backes, Jeffrey S. Norris, Mark Powell, Kam S. Tso, Gregory K. Tharp, and P. Chris Leger. Sequence planning for the fido mars rover prototype. In *Proceedings IEEE Aerospace Conference*, Big Sky, Montana, March 8-15 2003.

[3] 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. In *30th Lunar and Planetary Science Conference*, Houston, Texas, March 15–19 1999.

[4] Jeffrey S. Norris, Paul G. Backes, and Eric T. Baumgartner. PTeP: The parallel telemetry processor. In *Proceedings IEEE Aerospace Conference*, Big Sky, Montana, March 2001.

[5] Robert Steinke, Paul G. Backes, and Jeffrey S. Norris. Distributed mission operations with the multi-mission encrypted communication system. In *Proceedings IEEE Aerospace Conference*, Big Sky, Montana, March 9-16 2002.

[6] Mark W. Powell, Paul G. Backes, Marsette A. Vona, and Jeffrey S. Norris. Visualization of coregistered imagery for remote surface operations. In *Proceedings IEEE Aerospace Conference*, Big Sky, Montana, March 8-15 2003.

[7] Marsette A. Vona III, Jeffrey S. Norris, Mark W. Powell, and Paul G. Backes. Challenges in 3d visualization for mars exploration rover mission science planning. In *Proceedings IEEE Aerospace Conference*, Big Sky, Montana, March 8-15 2003.

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.*

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 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.*

Marsette A. Vona, III *is a computer scientist, software engineer, and electromechanical hardware developer. He is a member of the technical staff of the Mobility Systems Concept Development Section at the Jet Propulsion Laboratory. At JPL, his work is currently focused in the areas of high-performance interactive 3D data visualization for planetary exploration, user-interface design for science data analysis software, and Java software architecture for large, resource-intensive applications. Marsette received a B.A. in 1999 from Dartmouth College in Computer Science and Engineering, where he developed new types of hardware and algorithms for self-reconfigurable modular robots. He completed his M.S.*

14

in 2001 at MIT's Precision Motion Control lab, where he developed a high-resolution interferometric metrology system for a new type of robotic grinding machine. Marsette was awarded the Computing Research Association Outstanding Undergraduate Award in 1999 for his research in Self-Reconfigurable Robotics.

Robert Steinke is a computer scientist and member of the technical staff of the Autonomy and Control 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, distributed data consistency, and communication support for group collaboration. He received his B.S. in 1995 and M.S in 1997 in Computer Science from U.C. Santa Barbara, and his Ph.D. in 2001 in Computer Science from the University of Colorado at Boulder. As a graduate student he completed his Master's thesis on a serializable lazy update scheme for distributed databases, and his dissertation on distributed shared memory consistency models, and he received a graduate teacher certificate from Boulder for his training and work as an instructor. He now lives in Pasadena with his wife Mollie and daughter Adelaide.

Justin Wick is a Junior at Cornell University in Ithaca, NY, majoring in Applied and Engineering Physics. Justin collaborates with Russian scientists by developing highly parallel versions of their astrophysical simulations. He has also worked on many projects relating to the Athena project at Cornell. A resident of western Pennsylvania, Justin has always been interested in working with science and computers. When he's not busy studying, he enjoys hanging out with friends, broadway musicals, science fiction, and archaic board games. Justin plans to eventually pursue a career in computational physics.