Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

# Mars Exploration Rover Operations with the Science Activity Planner*

Jeffrey S. Norris, Mark W. Powell, Marsette A. Vona,
Paul G. Backes, Justin V. Wick

*Jet Propulsion Laboratory*
*California Institute of Technology*
*Pasadena, California, USA*

maestro@list.jpl.nasa.gov

*Abstract* – **The Science Activity Planner (SAP) is the primary science operations tool for the Mars Exploration Rover mission and NASA's Software of the Year for 2004. SAP utilizes a variety of visualization and planning capabilities to enable the mission operations team to direct the activities of the Spirit and Opportunity rovers. This paper outlines some of the challenging requirements that drove the design of SAP and discusses lessons learned from the development and use of SAP in mission operations.**

***Index Terms – teleoperation, visualization, planning, simulation, space***

## I. Introduction

The Spirit and Opportunity rovers completed their journey to Mars in January of 2004. The considerable expense and effort required to deliver these rover to Mars and the harsh environment that threatens their continued survival demand that the rovers be operated as efficiently as possible on every sol (Martian day) of their mission. Powerful and intuitive operations tools are a critical part of efficient mission operations.

The primary tool used by the science team on the Mars Exploration Rover Mission is the Science Activity Planner (SAP), shown in Fig. 1. On each sol of operations, SAP is used to analyze the data arriving from the Spirit and Opportunity rovers and construct a plan of activities for the rovers to execute on the next sol. This plan is refined by additional tools before it is transmitted to the spacecraft [1][2]. SAP is also used as the operations interface for research rovers in development at the Jet Propulsion Laboratory and was released as a public engagement tool for the Mars Exploration Rover mission. SAP's successes in mission operations, technology development, and public engagement earned it NASA's Software of the Year Award in 2004.

This paper begins with a description of some of the unique challenges involved in the operation of complex robots on the surface of another planet. Sections III and IV discuss how SAP's capabilities in downlink analysis and uplink planning address those challenges, respectively. Throughout this discussion, lessons learned from the use of SAP in mission operations are presented as recommendations for future robot operations tools.

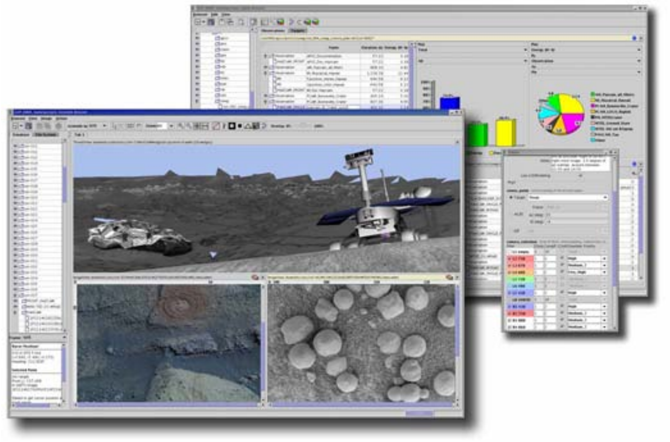

Fig. 1 The Science Activity Planner

## II. Mars Surface Mission Operations

### A. Challenges

The operation of rovers like Spirit and Opportunity on Mars is complicated by a number of constraints. Mission operations tools must be designed to address the following challenges:

*1) Limited operating resources*: Operating power for the rovers is generated entirely by solar cells on the top of the vehicles and stored in batteries. This stored energy is used not only to operate the vehicles' motors, computers, and instruments, but to power heaters that ensure the rovers' survival overnight. The total amount of data that can be stored onboard the rovers and transmitted to Earth is also strictly limited. All of these resources must be carefully modeled and managed within mission operations tools to maximize mission success without placing the spacecraft at risk.

*2) Few, irregular downlink opportunities*: Though the rovers can transmit data directly to Earth, this requires a large amount of power. Instead, the vast majority of the data acquired by the rovers are transmitted to the Mars Odyssey and Mars Global Surveyor spacecraft in orbit around Mars which then relay the data to Earth. Each rover performs only two communication sessions with an orbiter each day on average. To further complicate matters, these communication sessions don't occur at a consistent time each day because they must take into account the availability of the orbiting spacecraft and the Deep Space Network antennas on Earth. Therefore, mission operations

* This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration.

tools must be designed to accept bursts of new data at any time in the operations process.

*3) Full day plans*: It takes approximately 5 to 20 minutes for a signal to travel from Earth to Mars (depending on the relative positions of the planets in their orbits). This constraint, coupled with the scarcity of communication opportunities mentioned above makes real-time operation of the rovers infeasible. Instead, the rovers receive an entire sol's plan each morning and operate autonomously throughout the day. These plans must be carefully simulated within mission operations tools to ensure that they achieve all of the intended goals without endangering the spacecraft.

*4) Hard deadlines for uplink*: Spirit and Opportunity listen for transmissions from Earth at specific times each day. Thus, plans must be ready for transmission to the spacecraft at a specific time or an entire day of rover operations will be lost. However, the construction of these plans usually cannot begin until information about the final state of the vehicle on the previous day has been received. This tightly constrains the amount of time available for constructing plans and demands highly reliable and efficient mission operations tools.

The next two sections of this paper describe some of the features in SAP that address the challenges enumerated above.

## III. Downlink Analysis

Spirit and Opportunity have gathered a wealth of scientific data that deserves years of careful study. Scientists use a variety of tools for the methodical analysis of scientific data, but these tools are not appropriate for the extremely tight timeline of tactical mission operations. When data arrives from a rover on Mars, it must be quickly processed and analyzed by the mission operations team so that informed planning decisions can be made and uplink deadlines can be met. Thus, the development of SAP's downlink analysis features emphasized those capabilities that enable rapid downlink analysis in support of tactical operations. These capabilities fall into two categories: those used for the analysis of imagery and those used for the analysis of hyper-spectral data.

### A. *Image Analysis Capabilities*

Spirit and Opportunity each have nine cameras onboard. SAP allows its users to view the images produced by these cameras in multiple ways in order to quickly increase their awareness of the rovers' surroundings and identify locations of scientific interest. Any image acquired by the rover can be opened for individual study, but SAP also automatically collects images acquired by the rovers' mast cameras into panoramic collections as they arrive from Mars. SAP can render these collections in a cylindrical projection which shows the rover's surroundings from the perspective of the camera. This is a natural and familiar way to visualize these images, but users often become disoriented when viewing panoramic images rendered as a flat rectangle. To address this problem, SAP also provides a polar-azimuthal projection that simulates an overhead perspective. This projection makes it easier for a user to understand the relative locations of objects in the vicinity of the rover and

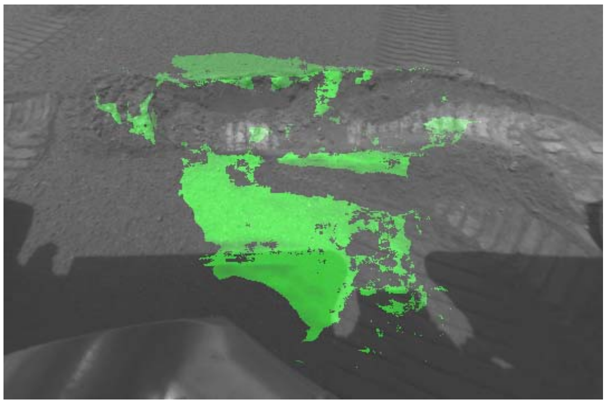

Fig. 2 Overlay showing positions reachable by the rover arm

is often used as a companion to a cylindrical projection view. SAP renders both of these projections from the necessary source images entirely on-the-fly, which enables the user to view these extremely large panoramas at any desired resolution without forcing the computer to load and hold a monolithic image in memory.

Eight of the nine cameras on Spirit and Opportunity are arranged as stereo pairs. These stereo images are used to produce range maps and 3D terrain meshes. Range and elevation data can also accessed for any point in any image by clicking on it and these datasets can be displayed as semi-transparent overlays on top of any image. As shown in Fig. 2, one of these overlays shows only those points that have been verified to be reachable by the rover's arm. SAP users can also visualize terrain meshes in an immersive 3D view [3]. The 3D view also displays a simulated 3D rover that enables users to see the vehicle in the context of its environment and evaluate the state of its articulated elements. Memory usage of the 3D view is reduced by sharing the images it displays with any 2D views that are displaying the same image. Fig. 3 shows the same collection of images rendered in cylindrical, polar-azimuthal, and 3D views.

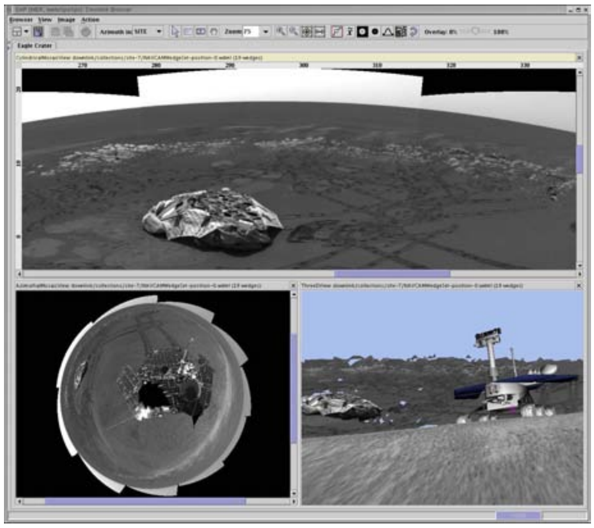

Fig. 3 Cylindrical (top), polar-azimuthal (bottom-left), and 3D (bottom-right) renderings of a mosaic of images acquired by Opportunity.

All of the views used to visualize images in SAP share a consistent set of analysis features. All images are indexed to their associated range map so that clicking in an image will display the 3D coordinates of that location and display the corresponding location in all other views. Distances can be measured using a virtual ruler and points of interest can be stored as targets for use in activity planning. Images can also be enhanced with a variety of filters and image arithmetic functions.

As discussed in the "Challenges" section at the beginning of the paper, data can arrive from the rovers at any time during the operations process on Earth. These data may be essential for ongoing planning discussions and decisions, so SAP is designed to automatically refresh all of its views when new data arrive. This ensures that mission operators will always be working with the latest data. This capability has proven valuable during mission operations, but could have been implemented more effectively if the mission had developed a data product announcement service that operations tools could subscribe to in order to be notified of the arrival of new data products.

### B. *Hyper-spectral Analysis Capabilities*

In addition to their cameras, Spirit and Opportunity each carry a powerful hyper-spectral instrument called the Miniature Thermal Emission Spectrometer (Mini-TES) [4]. This instrument measures the infrared radiation emitted by rocks and soil in the environment of the rover and can also be used to characterize the Martian atmosphere. SAP provides a novel visualization tool for this data called the Image Cube View, shown in Fig. 4. This view plots an array of spectral curves acquired by the instrument as a hyper-spectral cube that can be "sliced" with a plane through any of the cube's three major axes. The cube can also be "skewered" to plot the full spectral curve for any point. These features enable scientists to analyze the abundance of important minerals within the rocks and soil near the rover. All of the image enhancement features described in the last section are also available within the Image Cube View.

One of the chief purposes of the Mini-TES instrument is to provide mineralogical information on the soil and rocks in the vicinity of the rover. This information can be difficult to interpret unless it can be referenced to images of the same scene. SAP provides a feature called Data Fusion that

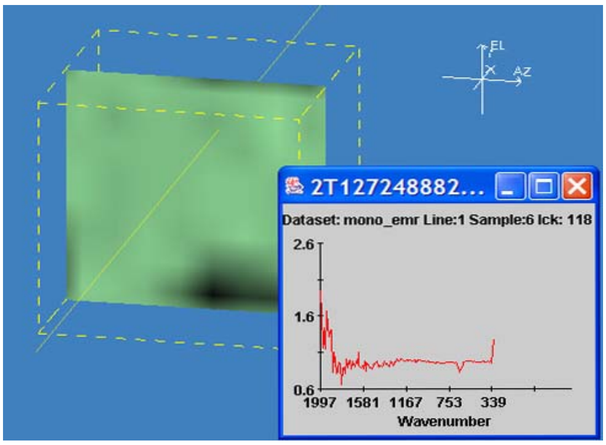


Fig. 4 The Image Cube View with a skewer plot (lower right)

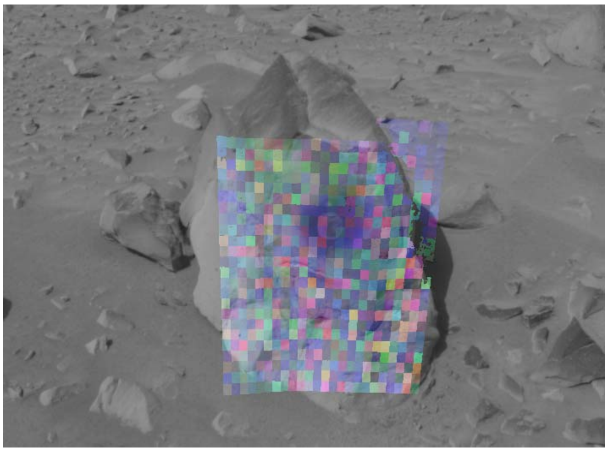

Fig. 5 Mini-TES data overlaid on a Navcam image using Data

enables users to display data from the Image Cube View as a semi-transparent overlay on top of images acquired by the rover [5]. An example of Data Fusion is shown in Fig. 5.

## IV. UPLINK PLANNING

Once the mission operations team has used SAP to analyze the data received from the rovers, they must turn their attention to the construction of a plan for the next sol of operations. SAP eases this transition by integrating uplink planning features into the downlink analysis environment.

### A. *Targeting*

When analyzing the data received from the rovers, mission operators often find points of interest that need to be labeled for later reference and use in planning. The Mars Exploration Rover used a slightly different approach than previous missions by defining two types of stored locations, described below.

*1) Features*: Objects in the environment such as rocks, patches of soils, craters, and mountain ranges need to be given consistent names for the purposes of discussion. These names refer to the object in its totality instead of a specific location within the object and typically describe the appearance of the object. These larger-scale elements of the environment are stored in SAP as Features.

*2) Targets*: Tactical Activities (image acquisition, for instance) are often directed at a specific location, such as a crack in a rock or a dark pebble in a patch of soil. It's often the case that multiple Activities need to be directed at the same exact location. These locations are stored within SAP as Targets. All Targets are associated with a parent Feature, creating a part/whole relationship between them.

In practice, this two-level hierarchy for naming locations produced mixed results. The distinction between names for large-scale objects and specific locations is a logical one, but users found the requirement that all Targets be associated with a parent Feature to be inconvenient and unnecessary. In addition, SAP represented both Features and Targets as points. This representation was found to be lacking for Features, which by their nature represent a region instead of a point.

Perhaps the most critical lesson learned from SAP's implementation of Features and Targets is that these constructs must be designed from the beginning to persist across multiple sols of operations. This is not a straightforward task because the rover's knowledge of its own position is not perfect after it drives. There is currently no reliable way to automatically adjust the locations of Features and Targets to account for this loss in positional knowledge. Thus, SAP Features and Targets must be respecified whenever the rover moves. This is a laborious process that could be made far easier if a Feature or Target were allowed to have multiple versions corresponding to different sols in which they were used. Then a user would merely need to indicate a new location for a Feature or Target (thus defining a new version) rather than completely recreating it.

To meet the hard uplink deadlines discussed in the "Challenges" section at the beginning of this paper, SAP had to be designed to support multiple users working with separate instances of the software in parallel throughout the mission operations center. Features and Targets created by one user needed to be instantly available to all other users. This required a centralized database. The SAP team chose to use an open-source database (MySQL) rather than develop a custom database from scratch. This decision reduced costs and increased the reliability of the overall system. When one SAP user creates a Feature or Target, it is instantly displayed in all running instances of SAP. This reduces the chance of two scientists assigning different names to the same object.

### *B. Activity Plan Creation*

Fundamentally, the task of activity planning for a rover is to encode the intent of the mission operators and store it for refinement by other operators and eventually, transmission to the spacecraft. In a manner similar to the Feature and Target constructs described in the "Targeting" section above, SAP uses two constructs to capture operator intent in activity plans.

*1) Activities*: This is the primitive unit from which SAP plans are constructed. Most Activities represent a single action by the rover, such as acquiring an image or driving the rover to a new location. Some Activities represent slightly more complicated actions such as acquiring a mosaic of images. These Activities can be automatically expanded into a set of more simple Activities. Each Activity type has a defined set of arguments that provide details on how the Activity should be accomplished. For instance, a Drive Activity has arguments that specify the destination.

*2) Observations*: Scientists often construct several related Activities to accomplish a particular goal. For instance, a scientist may build an Activity to acquire a Navcam image of a boulder and then follow that with a Pancam image to see one part of the boulder in detail. Related Activities are collected within an Observation. Observations are higher-level planning constructs that are used to capture the rationale behind a set of Activities. Once a user has added a set of Activities to an Observation, he/she describes the purpose of the Activities within text fields attached to the Observation.

In practice, the distinction between Observations and Activities was well received by the operations team. However, Observations were not as effective at capturing the rationale behind a set of Activities as was desired. The free-form text fields used for this purpose allowed great freedom of expression but could not be interpreted by other planning tools. A more rigorous, parameter-based scheme for capturing this information would have been more effective.

Activity plans for Spirit and Opportunity are constructed in a series of meetings. On each sol, SAP is initially used by individual users who construct fragments of plans as Activities and Observations. These users meet in small Science Theme Groups that are responsible for integrating these fragments into a plan that reflects their particular scientific interests. Next, representatives of each science theme group meet in a large meeting called the Science Operations Working Group. This meeting is responsible for integrating the plans from each theme group into a single, authoritative plan for the next sol. This plan is refined further before it is transmitted to the spacecraft. At each step in this process, less suitable Observations and Activities are discarded and the surviving elements are prioritized to reflect their relative importance to the operations team. SAP is designed to support this process of iterative integration and prioritization.

### *C. Simulation*

As discussed in the "Challenges" section at the beginning of this paper, a full day's set of commands are sent to a rover as a single plan. Since the rover is not in contact with Earth while executing the plan it is not possible to monitor its progress. These limitations make the simulation of activity plans before they are sent to the spacecraft extremely important. SAP employs a novel approach to plan simulation, described below, that provides orders of magnitude improvement in performance over the prior state of the art.

Activity plan simulation requires the development of models that accurately represent the impact that an Activity will have on the state of the rover if it is executed. These models can be developed "bottom-up" by studying the properties of the mechanical elements of the rover and the software that controls them onboard, or "top-down" through empirical measurements made as the flight system (or more often, a spare) executes the Activity in a laboratory setting. The Mars Exploration Rover Mission used a combination of these approaches to develop its initial models. After the rovers landed, careful attention was paid to the actual resources consumed by particular activities. This information was used to update the models used in simulation. Unfortunately, the process for updating these models was quite arduous because it required the same level of review and regression testing as a change to the code within one of the operations tools. Future missions should consider establishing a separate, more lenient change control procedure for simulation models so that changes can be more easily introduced.

Once these models are available, activity plan simulation is traditionally accomplished by beginning with the last known state of the rover and executing the models for every Activity in the plan in order. In effect, a simulated rover is stepped through all of the Activities in the plan to create a complete vector of simulated rover states. Depending on the complexity of the models, the simulation of an entire plan may require several minutes to complete, even on powerful mission operations workstations. Unfortunately, this approach re-simulates the entire plan whenever any modifications have been made because it can't predict the impact of an particular change. There is no distinction made between modifications that will have little or no effect on the state of the rover and those that will have a dramatic impact. These limitations greatly reduce the utility of plan simulation and degrade the performance of mission operators.

SAP is the first mission operations tool to use dependency-based plan simulation. As the user constructs a plan, SAP builds and maintains a directed acyclic graph that represents dependencies between Activities, their arguments, simulated rover states, and the visual representation of those states for the user. When a user makes a modification to the plan (by changing the value of an activity argument, for instance), SAP only recomputes the values of those elements that directly or indirectly depend on the element that was modified. The performance benefits of this approach are so great that SAP can update the simulation of the activity plan in real-time as modifications are made to the plan. This provides instant feedback to the user as they construct the activity plan. For the Mars Exploration Rover mission, this has dramatically improved the performance of the mission operations team.

SAP communicates the results of plan simulation to the user in several ways:

*1) Resource Charts and Graphs*: One of the challenges discussed at the beginning of this paper is the fact that rovers have very limited operating resources. In particular, the amount of power, time, and data volume that a plan will consume must be carefully tracked. SAP displays this information for the user in a tabular, spreadsheet-like view and via user-configurable bar and pie charts. All of these displays are updated in real-time as the user modifies the plan. This allows the user to try many "what-if" scenarios in search of a plan that fits within resource constraints.

*2) Image Footprints*: When an Activity that instructs the rover to acquire an image is added to the plan, SAP's simulation computes the position and dimensions of the area that will be included in the acquired image. This information is communicated to the user as an image "footprint" drawn on top of any existing image of that area that is currently being viewed by the user. As the user modifies the imaging Activity, the image footprint is updated in real-time. Image footprints allow the user to be certain that the Activity they are constructing will result in an image of the area that they are interested in. Footprints are also available for the Mini-TES instrument, as shown in Fig. 6. Image footprints have been a mainstay of the SAP application since its earliest versions [6].

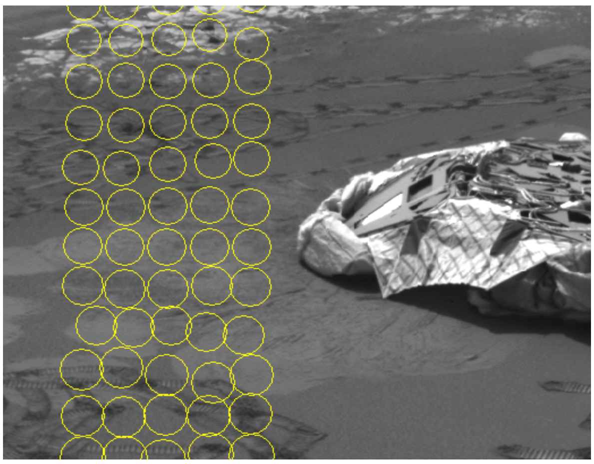

Fig. 6 Circular footprints representing a Mini-TES mosaic activity

*3) 3D Rover Simulation*: SAP's simulation computes the expected position and orientation of the rover along with the articulated state of its arm and mast. These states are rendered as a 3D model of the rover in any 3D image views that are currently open. This provides a similar benefit to the image footprints described above – the user can modify the Activities in the plan until the state of the simulated rover reflects their intent. 3D simulation of the rover is particularly helpful when planning Activities that use the rover arm, as shown in Fig. 7.

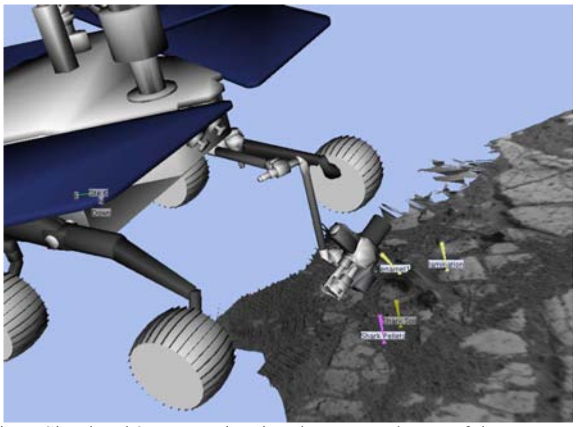

Fig. 7 Simulated 3D rover showing the expected state of the rover arm

## V. Conclusions

Lessons learned from the implementation and use of SAP have been included throughout this paper in the relevant sections above. At the time this paper was written, SAP had successfully planned over 450 sols of operations without a single critical failure. The features described in this paper have also enabled the mission science team to drastically reduce the amount of time needed to complete their work on each sol, enabling the mission to transition from a grueling Mars-time schedule to a more standard workday. Meanwhile, a public version of SAP called Maestro has allowed over 300,000 members of the general public to follow along with the progress of Spirit and Opportunity with the tool the science team uses.

Work has already begun on enhancements to SAP for future Mars missions and continued support of the goals of the technology program. Among the chief areas of for future development are distributed operations and the integration of data from orbital assets into tactical mission operations.

## Acknowledgment

This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology. The authors would like to thank Frank Hartman of JPL for providing the 3D rover model used in the SAP application.

## REFERENCES

[1] M. Ai-Chang et al, “MAPGEN: mixed-initiative planning and scheduling for the mars exploration rover mission,” *IEEE Intelligent Systems*, vol. 19, pp. 8-12, Jan.-Feb. 2004.
[2] S. Maxwell et al, “The design and architecture of the rover sequencing and visualization program (RSVP),” *SpaceOps*, May 2004.
[3] M. Vona et al, “Challenges in 3D visualization for mars exploration rover mission science planning,” *Proceedings, IEEE Aerospace Conference*, March 2003.
[4] P. R. Christensen et al, “Initial results from the mini-tes experiment in gusev crater from the spirt rover,” *Science*, vol. 305, no. 5685, pp. 837-842, Aug. 6, 2004.
[5] M. Powell, P. Backes, M. Vona, and J. Norris, “Visualization of coregistered imagery for remote surface operations,” *Proceedings, IEEE Aerospace Conference, Big Sky, MT,* March 2003.
[6] P. Backes et al, “Mars polar lander mission distributed operations,” *Proceedings, IEEE Aerospace*, March 2000.