

# A Step Toward Autonomous Planetary Rovers

Gregg Rabideau<sup>1</sup>, Steve Chien<sup>1</sup>, Paul Backes<sup>1</sup>, Gene Chalfant<sup>1</sup>, Kam Tso<sup>2</sup>

<sup>1</sup>*Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109*

<sup>2</sup>*IA Tech, Inc., 10501 Kinnard Ave., Los Angeles, CA 90024*

**Abstract.** Recently, significant advances have been made in enabling autonomous rovers and robotic vehicles. Current robotic vehicles employ significant elements of autonomy in their low-level operations and control. A further important step is the integration of high-level intelligent systems with autonomous rover control architectures. In this paper, we describe the integration of two software systems: ASPEN (Automated Scheduling and Planning ENvironment) and WITS (Web Interface for TeleScience). WITS provides a high-level graphical interface to Rocky7, an experimental rover developed by the Jet Propulsion Laboratory (JPL). Using WITS, a user can naturally specify science activities and locations using actual images of the rover's environment. ASPEN accepts the science goals input using WITS, reasons about the low-level activities and resources required to achieve these goals, and generates the executable sequence to enable achievement of the requested goals. Future steps would include a migration of this software to the rover itself, allowing the rover to schedule its own activities, and thus behave more autonomously.

## INTRODUCTION

In the current arena of more frequent, less expensive space missions, NASA is being pushed to perform space missions with smaller mission operations teams. As a result, NASA is examining advanced technologies to automate a wide range of mission operations functions such as: command generation and validation, fault detection, isolation and recovery, and science data processing and analysis.

Traditionally, NASA has employed large operations teams (upwards of one hundred personnel for the Galileo mission) to perform the complex and demanding task of determining spacecraft operations. This task involves combining and arbitrating among science goals to find a set which can be satisfied, and constructing a low-level set of commands to achieve these science goals while respecting engineering operations constraints (such as resource limitations, hardware constraints, etc.).

In this model, a science team first constructs a high-level plan of the science activities to be performed. This plan has only limited insight into engineering activities or constraints. This plan is then given to the engineering team, which has the responsibility of generating the low-level command sequence that: conforms to the science plan in that it achieves the underlying science objectives, incorporates engineering goals and activities as required, and respects operations constraints, resource constraints, and hardware constraints for the spacecraft. After the engineering team produces a sequence, the science team reviews it in order to verify that acceptable science activities have been included in the final sequence. If the science team is unsatisfied with the current sequence the process iterates until the science and engineering teams can agree on a satisfactory plan.

This mode of operations has enabled highly successful missions exploring the outer reaches of the solar system. The derivation and review of plans by large numbers of very knowledgeable personnel have allowed optimum use of spacecraft resources. Additionally, because of the review processes, and detailed prediction, simulation, and verification of spacecraft state, this mode of operations also has enabled an extremely high degree of mission reliability.

However, this mode of operations could be improved in several ways. First, employing large numbers of highly trained and knowledgeable operations personnel is expensive. Second, in cases where fast turnaround is required for

command generation, the manual process can be unwieldy and thus limit feedback from the science team into the engineering sequence generation. This can result in missed science opportunities or decreased science return. Third, this mode of operations is highly stressful to the human operators and while sustainable for short periods of time (planetary encounters, flybys, or other short duration missions) it is not sustainable for extended flybys or longer duration missions of perhaps a year or longer.

A number of these difficulties occur in operation of planetary rovers. During the Mars Sojourner mission operations in 1997, a science team determined daily science activities for the Sojourner rover. These science activities were then given to an engineering team that generated the actual uplinked sequence. The engineering team held the responsibility of generating an executable low-level rover activity sequence that: achieved the science goals specified by the science team, incorporated any necessary engineering activities, and respected the operations constraints of the rover. During the approximately 90 day Sojourner mission, daily sequences were generated in this fashion. While outstanding efforts of both the science and engineering teams enabled the incredible success of the Sojourner mission, longer future rover missions with more complex rovers will require higher degrees of automation to achieve the high standards of mission success and science return set by the Sojourner mission. While the Sojourner mission duration was relatively short, the science team could be collocated at JPL and the science and engineering team could sustain working extremely long hours per day. Future rover missions will have a 1 year (or longer) lifespan. For missions of this length, it is not practical to collocate science team and the science and engineering teams cannot reasonably be expected to sustain such high levels of effort without burnout. Additionally, future rovers will have more complex science instruments, more complex experiments and science goals. All of these conditions combine to present significant challenges to rover missions of the future.

This paper describes work to leverage automated planning and scheduling technology to enable automated generation of rover command sequences from science goals. These generated command sequences would respect operations constraints encoded in rover activity models. Such a capability would allow a science team to assess the viability of a wide range of science options with respect to their viability in light of engineering constraints. This would allow the science team to construct a science plan with a high degree of confidence that it would be implementable. The engineering team would then accept this science plan with generated sequence and perform additional validation on it to ensure that it would satisfy all engineering constraints and the resultant sequence could then be uplinked.

## **DISTRIBUTED ROVER SCIENCE**

The Web Interface for Telescience (WITS) (Backes et al., 1998) is an Internet-based tool that enables scientists to collaboratively participate in planetary lander and rover missions from their home institutions. WITS enables the viewing of downlinked images and results in various ways, terrain feature measurement and annotation, and planning of daily mission activities. WITS is written in the Java language and is accessible by mission scientists and the general public via a commercial web browser. The public can use WITS to plan and simulate their own rover missions. WITS was used in the 1997 Mars Pathfinder mission for public outreach and limited science operations and will be a nominal science operations tool in the 1998 lander and 2001 and 2003 lander-rover missions to Mars. Information on WITS activities can be found on the internet (<http://robotics.jpl.nasa.gov/tasks/wits/>).

Various view windows in WITS permit looking at the images and data in different ways. Planning and measurement can be done in all views, with all inputs displayed in all views. The Descent View has an orbital image or an image taken by the spacecraft during descent to the surface. The Panorama View shows an overhead view of the area around the lander or rover in various ways, e.g., color coded elevation map or texture map. The Mosaic View is a mosaic of the actual panorama images. For the Pathfinder mission, the panoramic images were taken by a camera on the lander, as will be the case for the 1998 lander mission. For the rover missions, the panoramic images will also be taken from a mast on the rover. The Close-up View provides viewing of images taken by the navigation cameras on the rover. The VRML View provides 3D solid-model visualization and simulation, and viewing from different directions. The user can specify science targets and waypoints in any view. Activities to be performed at a target or waypoint are specified by opening a Task window. Access to the mission system is controlled through the use of user logins and passwords.

## **AUTOMATED PLANNING AND SCHEDULING**

ASPEN (Fukunaga et al., 1997) is an automated planning and scheduling tool developed using Artificial Intelligence (AI) techniques and algorithms. First, ASPEN requires a "model" describing the possible set of activities and the constraints between those activities. Once the model is specified, the user can make initial requests, manually edit schedules, and automatically generate and repair schedules. ASPEN uses a client/server design where the server contains the schedule data and algorithms and clients, such as the Graphical User Interface (GUI), can send schedule modifications or request information. The GUI displays the activities, their location in time, and the resources and states affected by the activities. Conflicts are displayed when any of the constraints are violated. The GUI also provides an interface for manually editing the schedule. Several algorithms have been implemented for generating initial schedules from the requests, and repairing existing schedules with conflicts. These algorithms are generic and can be used on any given model. More information on ASPEN can be found on the internet (<http://www-aig.jpl.nasa.gov/planning/aspenn/>).

The most commonly used algorithm in ASPEN is based on a technique called "iterative repair" (Fukunaga et al., 1997, Zweben et al., 1994, Minton et al. 1988). Basically, after given the initial requests, ASPEN loops through all of the conflicts, trying to resolve each by performing a schedule modification. A conflict in the schedule is simply a violation of one of the resource, state, or temporal constraints defined in the model. The modifications it can perform include moving, adding, and deleting activities. Because the specified requests from WITS may be incomplete (e.g., missing activities, too many activities, no designated start times) ASPEN will initially have a sequence that contains conflicts. For example, the initial sequence in Figure 1 has only one "go\_to\_location" which changes the location state variable to a value that represents the coordinates of the rover. Because the science requests have constraints for different values of the location state variable, conflicts are immediately detected. To resolve these conflict, new "go\_to\_location" activities are created with the appropriate parameter values. This repair algorithm is implemented at a level independent of the model. ASPEN computes the set of conflicts, and based on the type of conflict chosen for repair, it computes the set of possible modifications that might resolve the conflict. In the example above, the conflict is on a state variable (location), so ASPEN looks for an activity type ("go\_to\_location") that could change the value of the state variable. This algorithm also works well for interactive scheduling. The user could manually change the schedule using the ASPEN GUI. Any conflicts created by the user could be automatically resolved by pressing the "Repair" button, which starts the algorithm.

## **ROCKY7 APPLICATION**

In a prototype demonstration (<http://bisc.jpl.nasa.gov/pages/aspennWits/external/aspennWits.html>), WITS and ASPEN were integrated and applied to Rocky7, an experimental rover in operation at JPL. First, a model describing Rocky7 was developed for both WITS and ASPEN. Then, during operation, WITS was used to visualize the terrain around the rover and generate the initial requests. The user selected science targets in WITS and specified one of the science tasks for that location. Science tasks include digging, spectrometer reads, and stereo images. Tasks were given different priorities with a threshold indicating whether or not that task is optional. The request sequence was sent electronically to ASPEN in the Spacecraft Activity Sequence Format (SASF) (Lombard, 1997). ASPEN was used to take the initial sequence from WITS and automatically generate a more complete and valid sequence to return to WITS. ASPEN performed planning, resource analysis, and constraints checking to generate a new sequence that achieves the science goals within the operations constraints. This sequence is sent back to WITS where it was displayed to the operator. The new sequence contained new activities, had some optional activities removed, and had specific times assigned to each activity. The command sequence was then sent for execution on the Rocky7 rover in the JPL marsyard.

### **Science Planning Using WITS**

The WITS user selected science target location (L2) to the West and used the Task Window to specify a dig activity at that location and set the priority for the dig to priority level 1 (lowest priority). Then the user selected a spectrometer read at a new target location (L3) to the South and left the priority at the default level 3. Then the user specified another dig at a new target location (L4) to the East and again using the default priority level 3. Finally, the

operator selected a science image with the default priority at a new target location (L5) on a rock to the North-East. Using the activity pop-up window, the operator specified the requirement for full-illumination of the target when taking the image and the local time for the operation was automatically computed for the command.

```

go_to_location(rover, anytime, 1)
dig(L2, anytime, 13)
spectrometer_read(L3, anytime, 14)
dig(L4, anytime, 15)
image_nav_time(L5, 1998-092/10:15:00.000, 16, full, now, 60)

```

FIGURE 1. Sequence sent from WITS to ASPEN.

The sequence that was generated by WITS is shown in Figure 1 in a terse format (the SASF format is too verbose to include here). The initial “go\_to\_location” is included to move the rover to the assumed start location labeled “rover”. Figure 2a shows the Panorama and Mosaic views in WITS after the operator specified the spectrometer read, two digs, and science image activities. The user then sent the sequence to ASPEN by pressing the “Scheduler” button in the WITS Sequence window, which sent the sequence to ASPEN in SASF format.

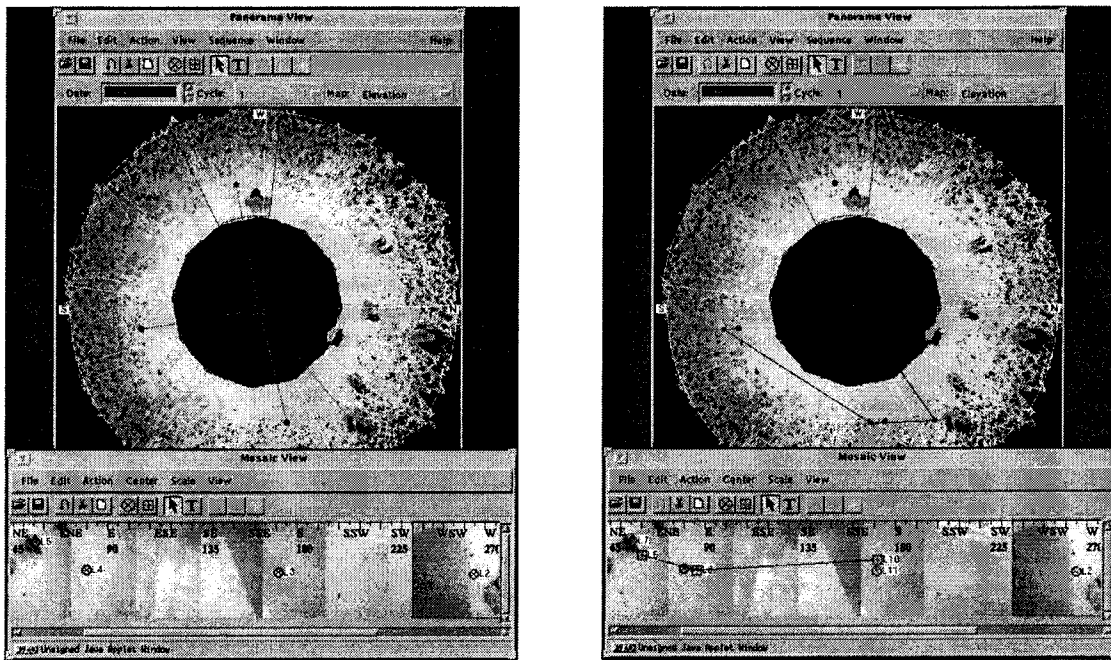


FIGURE 2. WITS views of a) initial and b) final sequences.

### Automated Sequence Generation Using ASPEN

ASPEN received the sequence from WITS and performed planning, resource analysis, and constraints checking to generate a new sequence to achieve the science goals within operations constraints. The new sequence has the science image task scheduled first due to the constraints on the angle of illumination of the target. Traverse and orient activities were added to move the rover into the desired position for the stereo image. The “go\_to\_location” command is given a new location (L6), which was computed based on parameters for the requested heading to and distance from the target. The “face\_location” command is given the same location (L5) as the requested image. In addition, one of the digs has been removed due to a limited overall execution time. The dig selected for removal was the one at location L2, which was the only science request given the lowest priority. To satisfy mission rules, a dump has been inserted after the dig to empty any material collected by the dig. Finally, the spectrometer read was scheduled last. No timing constraints were given on either the dig or spectrometer read, and so their order was

chosen arbitrarily. Again, for each science target, “go\_to\_location” and “face\_location” activities were inserted to position the rover at the desired location and orientation. After automatically making all of the required modifications, ASPEN sent the new sequence back to WITS in SASF format. Figure 2b shows the Panorama and Mosaic views in WITS visualizing the updated sequence after receiving it from ASPEN. The actual sequence that was returned from ASPEN to WITS is shown in a terse format in Figure 3.

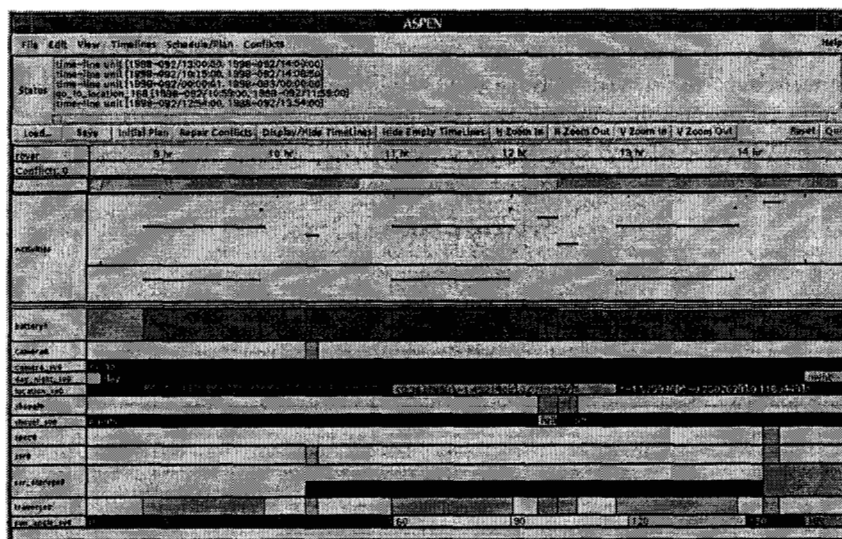
```

go_to_location(rover, 1998-092/08:30:00, 1)
go_to_location(L6, 1998-092/08:51:30, 6)
face_location(L5, 1998-092/09:51:30, 7)
image_nav_time(L5, 1998-092/10:15:00, 5, full, now, 60)
go_to_location(L8, 1998-092/10:59:00, 8)
face_location(L4, 1998-092/11:59:00, 9)
dig(L4, 1998-092/12:13:45, 4)
dump(L4, 1998-092/12:23:45, 10)
go_to_location(L10, 1998-092/12:54:00, 11)
face_location(L3, 1998-092/13:54:00, 12)
spectrometer_read(L3, 1998-092/14:08:50, 3)

```

**FIGURE 3.** Sequence sent from ASPEN to WITS.

For this demonstration, ASPEN was used in batch mode to automatically generate an updated valid sequence. Using the GUI, an operator could have used ASPEN to do further interactive scheduling before providing the updated sequence to WITS. Also, ASPEN could have returned resource analysis results to the WITS operator. The ASPEN GUI with the updated sequence is shown in Figure 4. The top section under the buttons shows a time-oriented view of the activities. Each horizontal bar represents the scheduled start and end time for an activity. The lower section displays the resources and states monitored by ASPEN.



**FIGURE 4.** ASPEN view of final sequence.

### Sequence Execution on Rocky7

The sequence that was returned from ASPEN to WITS was expanded into the Rocky7 command format and transmitted to the rover for execution. The Rocky7 rover system was provided for this demonstration by the Long Range Science Rover (Hayati et al., 1997) task of the NASA Telerobotics Program. The executed sequence of commands is the same as in Figure 3. Pictures of the rover executing the command sequence are in Figures 5a-c, showing the science image, dig, and spectrometer read respectively.



FIGURE 5. Rocky7 executing activities a) science image b) dig and c) spectrometer read.

## DISCUSSION

Automated planning and scheduling technologies have great promise in reducing operations cost and enabling autonomous rovers. By automating the command sequence generation process and by encapsulating the operation specific knowledge, ASPEN will enable commanding by a small operations team without subsystem experts. In addition, it will provide scientists with immediate and more informed feedback on the feasibility of their science requests. ASPEN and WITS were combined to provide a more intelligent tool for rover ground operations.

Ultimately, automated planning and scheduling technologies could be migrated on-board rovers to provide more autonomous behaviors. Moving ASPEN to the on-board rover computer would allow the rover to schedule its own activities. Only the high-level goals would need to be transmitted to the rover. This would mean that less overall uplink would be required to operate the rover. In addition, with an on-board scheduler, the rover could reschedule in reaction to failures, changes in the environment, or detected opportunities. For example, using on-board software for collision avoidance, the rover may take longer to execute a traverse than originally planned. Given this loss of time, the rover may need to reschedule future activities, or even terminate the current traverse activity. Having this capability on-board would reduce the need to have a human in the loop. Without the round trip communication times, the rover could react and reschedule much faster, keeping it busy with the endless supply of science requests. Automated planning and scheduling is one of the key components to an autonomous rover.

## ACKNOWLEDGMENTS

This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract to the National Aeronautics and Space Administration. Work described in this paper was supported by the Autonomy Technology Program, NASA Code SM and by the Exploration Technology Program, Telecommunications and Mission Operations Directorate (TMOD).

## REFERENCES

- Backes, P., Tso, K., and Tharp, G., "Mars Pathfinder Mission Internet-Based Operations Using WITS," in *Proceedings IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998.
- Fukunaga, A., Rabideau, G., Chien, S., and Yan, D., "Towards an Application Framework for Automated Planning and Scheduling," *Proceedings of the IEEE Aerospace Conference*, Snowmass, CO, 1997, pp. 375-386.
- Hayati, S., Volpe, R., Backes, P., Balaram, J., Welch, R., Ivlev, R., Tharp, G., Peters, S., Ohm, T., and Petras, R., "The Rocky7 Rover: A Mars Sciencecraft Prototype," *Proceedings IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, 1997, pp. 2458-2464.
- Lombard, W., Spacecraft Activity Sequence File, JPL Software Interface Specification Module (SFOC-1-SEQ-ANY-SASF), Revision 06, 1997.
- Minton, S., Johnston, M., Philips, A., and Laird, P., "Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems," *Artificial Intelligence* **58**, 161-205 (1988).
- Zweben, M., Daun, B., Davis, E., and Deale, M., "Scheduling and Rescheduling with Iterative Repair," *Intelligent Scheduling*, edited by M. Zweben and M. Fox, Morgan Kaufmann, 1994, pp. 241-255.