Automated Planning and Scheduling for Planetary Rover Distributed Operations

Paul G. Backes*, Gregg Rabideau*, Kam S. Tso†, Steve Chien*
*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California
†IA Tech Inc., Los Angeles, California

Abstract

Automated planning and scheduling, including automated path planning, has been integrated with an Internet-based distributed operations system for planetary rover operations. The resulting prototype system enables faster generation of valid rover command sequences by a distributed planetary rover operations team. The Web Interface for Telescience (WITS) provides Internet-based distributed collaboration, the Automated Scheduling and Planning Environment (ASPEN) provides automated planning and scheduling, and an automated path planner provides path plannning. The system was demonstrated on the Rocky7 research rover at JPL.

1 Introduction

Planetary mission operations consists of receiving downlink data, processing the data, generating the next command sequence, and then uplinking the sequence to the spacecraft. In past missions, this process for generating each command sequence could take days or weeks and require numerous operations personnel. The highly successful 1997 Mars Pathfinder mission was able to reduce the time for generating a command sequence to the Sojourner rover to about 12 hours. New distributed operations, automated planning and scheduling, and automated path planning technologies described in this paper are being developed to reduce this operations planning time further for future Mars rover missions.

For the Pathfinder Sojourner rover daily mission operations, the engineering team determined the state of the rover and provided it to the science team. The science team, colocated at JPL with the engineering team, then colaborated to specify a high-level science plan consisting of a list of science activities which they wanted to have included in the next uplink rover command sequence. This list of science activities included science activity parameters and locations, but the trade-offs for including the activities versus engineering constraints were not known. The high-level science plan was given to the engineering team, which had the responsibility for generating the low-level command sequence that satisfied the science plan as well as possible while considering operations constraints, resource constraints, and hardware constraints for the rover. The process of generating the command sequence was largely manual relying on the significant skills of the operations personnel. If there was time after the engineering team produced a sequence, then the science team reviewed it in order to verify that acceptable science activities were included in the final sequence, and iterated with the engineering team until a sequence was generated which satisfied both science and engineering constraints. The engineering team was usually able to accomodate most of the science objectives, but usually there was not time for iterations on the sequence with the science team.

Since the Pathfinder mission was relatively short, a few months long, the science team could be colocated with the engineering team at JPL. Also, the operations team members were able to work very long hours coincident with Mars daily cycles - meaning that they might work 12, and often more, hours in a day starting in the middle of the Earth night.

Future Mars rover missions, i.e., in 2001, 2003, and 2005, will be more challenging by being longer with missions lasting up to a year or longer, with more sophisticated rovers and increased numbers of science instruments, and with more constrained communications oportunities. New technologies need to be provided for mission operations to maximize mission science return while minimizing opertions costs and providing a reasonable working environment for the operations personnel for the long duration of the missions.

This paper describes the results of recent work to integrate Internet-based distributed operations with automated resource analysis and scheduling of activities to enable increased science return at lower cost for future planetary rover missions. Automated path planning has also been added to the system so that the automated scheduler can determine traverse costs between science targets when performing the scheduling. The system takes science activity requests specified by the distributed operations team and generates a valid command sequence which can be sent to a rover. The integrated system was used to command the Rocky7 research rover at JPL.

The integrated system of this paper, while being an initial prototype, has many of the basic features needed in future planetary rover mission operations. Distributed operations enables full participation by scientists from their home institutions, as well as collaborative planning by the science and engineering teams. Automated resource analysis and scheduling enables the science team to perform trade-offs on science objectives and generate a sequence which they have confidence is likely to be achievable within mission resource constraints, before handing it off to the engineering team. Automated path planning is needed by the resource analysis system to determine traverse costs between science targets. Also, it enables the science team to generate initial approximate paths without relying on a planning iteration with the engineering team who have the skill to do it manually. Mission planning time will be reduced by having automated, fast, constraints checking and valid sequence generation. The number of necessary iterations between the science and engineering teams will decrease by providing more operations constraints checking capability to the science team via the automated scheduler and path planner. The expertise of individuals in the operations team will still be required to generate and validate a final safe command sequence, but the sequence that they get from the science team will be much more complete and viable.

2 System Architecture

The system architecture is shown in Figure 1. The architecture is a simplified diagram of planetary rover mission operations which describes the interaction of the technologies of this paper. The Downlink system receives downlink data from the rover and the Data Processing system processes the data and puts both the raw data and data products, such as images and range maps, into the Database. Distributed Sequence

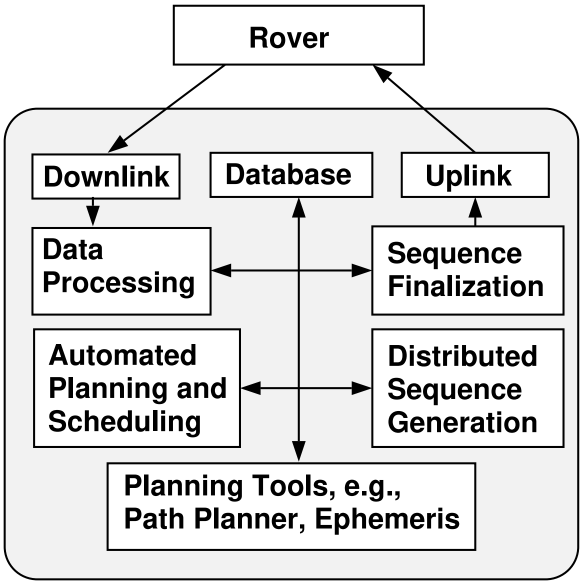

Figure 1: Operations System Architecture

Generation provides collaborative sequence generation by the geographically distributed science and engineering teams and is performed here by the Web Interface for Telescience (WITS), described below in Section 3. WITS provides distributed visualization and activity specification and is accessed by a user via a web browser, e.g., Netscape Navigator. The Internet-based WITS client communicates with a server system at JPL which includes other elements of the architecture. The Automated Planning and Scheduling system receives sequences and performs resource analysis on them and generates output sequences which are valid within modeled operations constraints. The Automated Scheduling and Planning ENvironment (ASPEN), described in Section 4, was used as the Automated Planning and Scheduling system here. Various other planning tools might be used to support distributed sequence generation or automated planning and scheduling such as automated path planning or automated computation of planetary ephemeris data. A new automated path planning system, described in Section 5, was used for path planning. The SPICE planetary ephemeris system [1] was used to compute angles to the sun from the rover position.

The Sequence Finalization system would be responsible for taking the sequence generated by the distributed operations team and outputting the final, valid, command sequence which is uplinked to the rover for execution. This role was performed in the Pathfinder Sojourner rover mission by the engineering team using the Rover Control Workstation. In the system described in this paper, the sequence generated via the WITS-ASPEN system was valid enough to send directly to the rover for execution.

The various software modules could run on different computers. In the examples of this paper, three computers were used. The ASPEN system ran on one computer, the rest of the server systems ran on one computer, and the WITS client ran on a separate computer. The Database was a Unix file system. The ASPEN system, path planner, and SPICE systems all ran as active server systems. The WITS client accessed Database data as URL calls through an httpd web server on a Sun Unix computer or via CGI programs which read or wrote data in the Database. The WITS client communicated with the ASPEN, path planner, and SPICE server systems via CGI programs which acted as clients to the server programs. An Uplink server program took the final command sequence from WITS via a CGI program and sent it to the rover for execution.

3 Distributed Operations

The Web Interface for Telescience (WITS) [2] 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 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 3D 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.

Distributed collaboration is achieved in WITS via sharing a common server-located database. A user can save inputs such as science targets and science activities to the common database by pressing a save icon. All users can see the current shared data by pressing the load icon. All users share a common sequence which is collaboratively generated.

4 Automated Planning and Scheduling

The Automated Scheduling and Planning Environment (ASPEN) [3] is an automated planning and scheduling tool developed using Artificial Intelligence (AI) techniques and algorithms. ASPEN generates a sequence which is an ordered list of activities. A model of the system to be scheduled is first generated which describes the possible set of activities, the constraints between those activities, and the resources required for each activity. ASPEN can be used in both an interactive mode where a user inputs and modifies activity requests directly, or in a batch mode where a sequence is automatically input, e.g., from WITS as used in this paper, and the sequence is automatically modified to meet modeled resource constraints. 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 when used in an interactive mode. More information on ASPEN can be found on the Internet (http://www-aig.jpl.nasa.gov/planning/aspen/).

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. The most commonly used algorithm in ASPEN is based on a technique called iterative repair [3, 4, 5]. With iterative repair, the scheduler determines the conflicts in an input schedule of requests and 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.

The initial sequence which ASPEN receives from WITS usually contains conflicts since the specified requests from WITS may be incomplete e.g., missing activities, too many activities, or no designated start times. For example, the initial sequence in Figure 4 has only one go_to_location activity 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 conflicts, 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 can 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.

5 Automated Path Planning

The automated path planner system consists of an obstacle map generation system and a path planner [6]. The obstacle map was generated when the panorama data was provided. The obstacle map was then subsequently used by the path planner. Obstacles were determined via segmentation of objects in the scene and morphological image operations to correct for errors in obstacle outlines and remove noise originating from the position data. This will be ex-

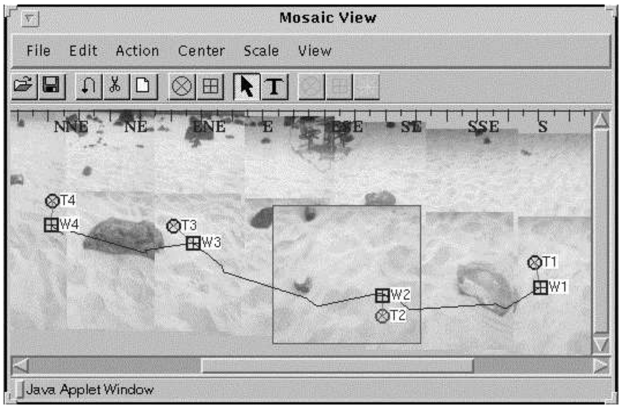

Figure 2: Path Before Path Planning

tended in the next year to include the use of range maps in generating the obstacle map. The path planner utilizes a modified A^* search algorithm to search through the obstacle map from the initial to final point. The following refinement stage eliminates extraneous via-points, resulting in very similar paths to the more computationally expensive A^* algorithm. The resulting path is a piecewise linear path of via-points. The end points are called waypoints. Waypoints are the inputs to the path planner and are points which the rover must traverse through. The waypoints might be specified by a user or by the automated scheduler.

Figures 2 and 3 show results, displayed in the WITS Mosaic View, of the automated path planner using panorama images from the JPL marsyard. The circle icons are science targets where science activities will be performed. The square icons are waypoints which the user has specified were the rover must traverse to before beginning the science activities. The paths are not smooth due to inaccuracies in the camera models at the boundaries of neighboring images (the camera models are used to project the 3D path points into the image plane for display in the Mosaic View). The crosses along the path in Figure 3 are the via-points which the path planner automatically generated to specify a collision-free path.

6 Automated Scheduling Demonstration

Results from a demonstration of the automated scheduling with distributed operations is shown in Figures 4, 5, and 6. WITS was used to visualize the terrain around the rover, generate the initial science

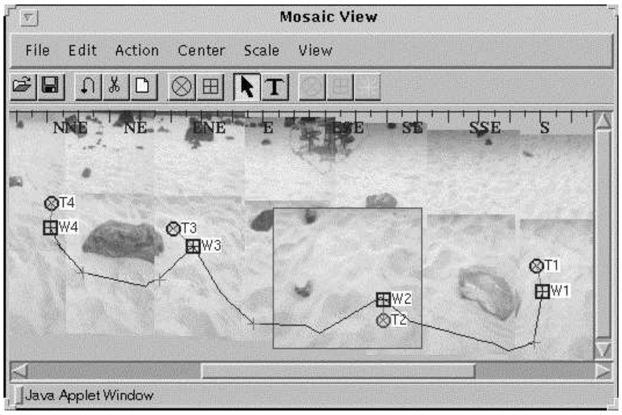

Figure 3: Path After Path Planning

targets and activities, and to send the final sequence to the Rocky7 rover. ASPEN utilized automated resource analysis, planning, and scheduling to take the initial sequence from WITS and generate a more complete and valid final sequence which was returned to WITS. The Rocky7 research rover has been developed at JPL by the Long Range Science Rover task [7] of the NASA Telerobotoics program. The rover was placed in the JPL Marsyard and a set of panorama images was taken by the 1.5 meter tall deployable mast on the rover. These images are shown in the Mosaic Views of Figures 4 and 5.

The WITS user selected a dig target location (L2) to the West and used the Task Window to specify a dig activity there and set the priority for the dig to priority level 1. Then the user selected a spectrometer read target (L3) to the South and used the Task Window to specify the spectrometer read activity and left the priority level at the default level 3. Then the user specified another dig target (L4) to the East and used the Target Task Window to specify the dig activity there with the default priority level 3. Finally, the operator selected a science image target (L5) on a rock to the North-East and used the Task Window to specify a science image task (image_nav_time) to be performed there, and left the priority for the activity at the default level 3. Using the activity pop-up window, the operator specified that full-illumination of the target was desired when taking the image and the desired local time for taking the image was automatically computed and inserted in the command. The WITS client called a CGI program at the server computer to access the SPICE system to compute the ephemeris data to determine the angle to the sun from the target location for various possible times. Figure 4 shows the Panorama and Mosaic views in WITS after the operator specified the spectrometer read, two digs, and science image activities.

The sequence that was generated by WITS is shown below in Terse format.

```
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)
```

The operator then sent the sequence to ASPEN by pressing the Scheduler button in the WITS Sequence window. WITS sent the sequence to ASPEN in spacecraft activity sequence (SASF) format [8]. SASF is a standard spacecraft sequencing format, with more details than the Terse format which is normally viewed by the operator. 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 had the science image task first due to its time of execution constraint, one of the digs was removed due to overall execution time constraint, and a dump was inserted after the dig due to mission rules. Also, traverse (go_to_location) commands and face location (face_location) activities were inserted to position the rover at the activity locations. Straight line paths were used due to the lack of obstacles in the demo area. This new sequence was sent back to WITS in SASF format, and is shown below in Terse format.

```
go_to_location(rover, 1998-092/08:30:00, 1)
go_to_location(L6, 1998-092/08:51:30, 6)
face_location(L7, 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(L9, 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(L11, 1998-092/13:54:00, 12)
spectrometer_read(L3, 1998-092/14:08:50, 3)
```

Figure 5 shows the Panorama and Mosaic views in

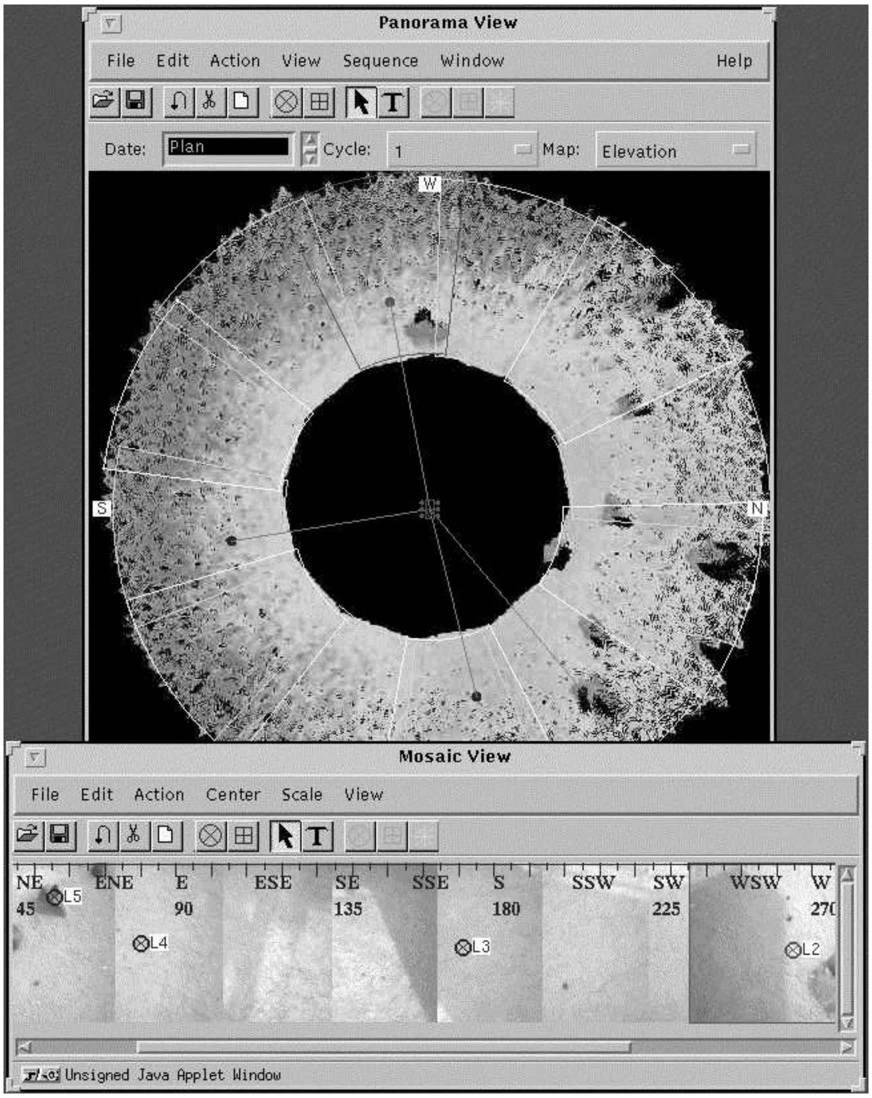

Figure 4: Initial Sequence In WITS

WITS visualizing the updated sequence after receiving it from ASPEN.

The sequence that was returned from ASPEN to WITS was expanded into Command format and sent for execution on the Rocky7 rover. The Rocky7 rover performing the dig activity is shown in Figure 6.

The ASPEN system was used in this demonstration in batch mode where it worked autonomously to generate an updated valid sequence. The ASPEN tool could also have been used interactively by an operator to do further interactive scheduling before providing the updated sequence to WITS. Also, ASPEN could have returned resource analysis results to the WITS

Figure 6: Rocky7 rover performing dig activity

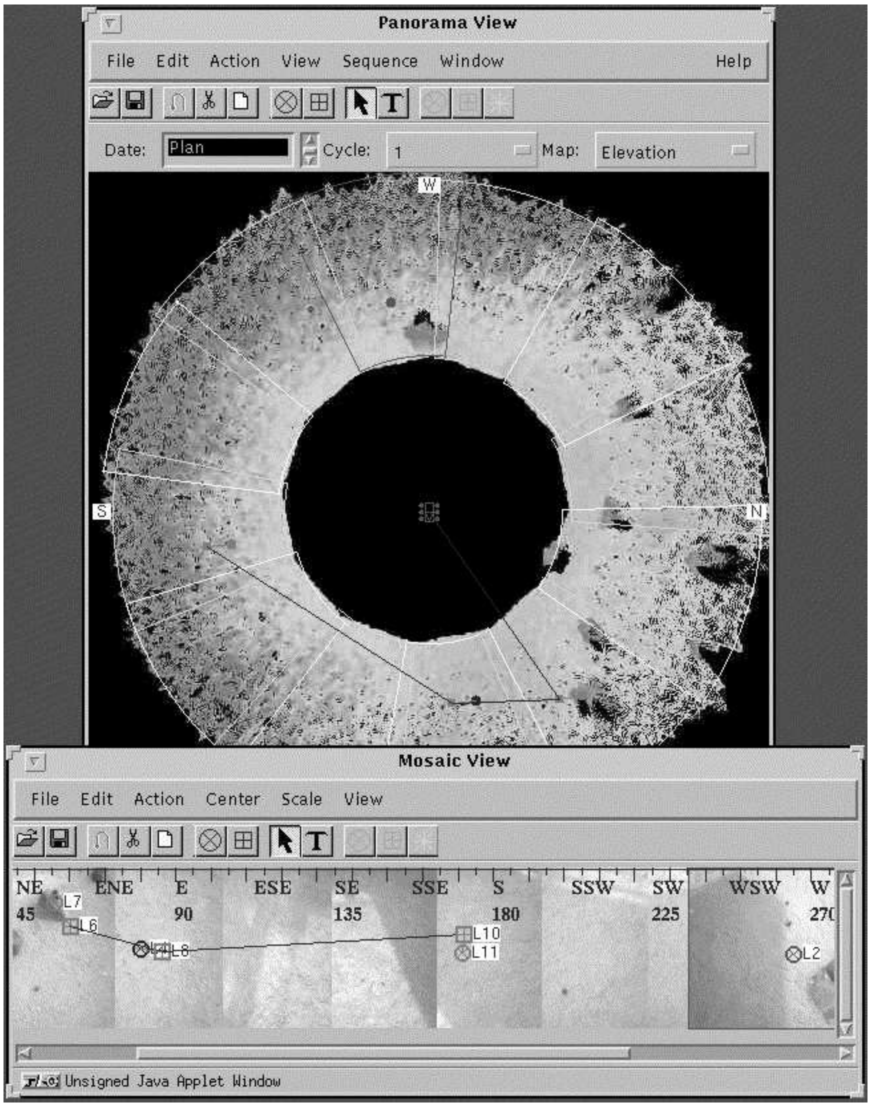

Figure 5: Sequence Returned from ASPEN

operator.

7 Conclusions

The integration of automated planning and scheduling and automated path planning with a distributed operations system, as represented by the integrated system described in this paper, has the potential to greatly increase the capabilities of planetary rover mission operations. Internet-based distributed opertations will enable a large science team to fully participate in daily mission operations from their home institutions. Automated planning and scheduling will enable the distributed science team to rapidly perform trade-offs while generating their mission activity sequences and allow them to have confidence that the sequence that they provide to the uplink engineering team is acheivable. Automated path planning enables the automated scheduling system to determine traverse costs between science target locations.

The system described in this paper is the first distributed operations system for planetary rover operations utilizing automated planning and scheduling and automated path planning. While the system represents the functionality desired for future planetary rover operations, each component of the system will still need to evolve to provide the complete capabilities, with the necessary robustness, to enable the system to be used in the next Mars rover missions. Specific enhancements which are currently being developed include the following. The WITS system will integrate more Internet security features as well as enhance its distributed collaboration capabilities. The ASPEN system will utilize more complete activity models. The obstacle map that the path planner uses will be generated utilizing range information as well as image intensity information, and the path planner will generate paths for different risk levels, e.g., low, medium, and high risk paths. The operations team will then be able to also do trade-offs which include risk levels.

Acknowledgements

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] Jr Charles H. Acton. Ancillary data services of nasa's navigation and ancillary information facility. *Planetary and Space Science*, 44(1):65–70, 1996.

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

[3] A. Fukunaga, G. Rabideau, S. Chien, and D. Yan. Towards an application framework for automated planning and scheduling. In *Proceedings IEEE Aerospace Conference*, pages 375–386, Snowmass, Colorado, 1997.

[4] M. Zweben, B. Daun, E. Davis, and M. Deale. *Intelligent Scheduling*, chapter Scheduling and Rescheduling with Iterative Repair, pages 241–255. Morgan Kaufmann, 1994.

[5] S. Minton, M. Johnston, A. Philips, and P. Laird. Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*, 58:161–205, 1998.

[6] Magnus Snorrason, Jeff Norris, and Paul Backes. Vision based obstacle detection and path planning for planetary rovers. In *SPIE International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, Orlando, Florida, April 1999.

[7] S. Hayati, R. Volpe, P. Backes, J. Balaram, R. Welch, R. Ivlev, G. Tharp, S. Peters, T. Ohm, and R. Petras. The rocky7 rover: A mars sciencecraft prototype. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 2458–2464, Albuquerque, New Mexico, April 1997.

[8] W. Lombard. Spacecraft activity sequence file. Technical Report JPL Software Interface Specification Module (SFOC-1-SEQ-ANY-SASF), Revision 06, Jet Propulsion Laboratory, 1997.