**Proceedings of the 2001 IEEE/RSJ
International Conference on Intelligent Robots and Systems
Maui, Hawaii, USA, Oct. 29 – Nov. 03, 2001**

Autonomously Generating Operations Sequences For a Mars Rover using AI-based Planning

**Rob Sherwood, Andrew Mishkin, Tara Estlin, Steve Chien,
Paul Backes, Jeff Norris, Brian Cooper, Scott Maxwell, Gregg Rabideau**

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr. • Pasadena, CA 91109
firstname.lastname@jpl.nasa.gov

Abstract

This paper discusses a proof-of-concept prototype for ground-based automatic generation of validated rover command sequences from high-level science and engineering activities. This prototype is based on ASPEN, the Automated Scheduling and Planning Environment. This Artificial Intelligence (AI) based planning and scheduling system will automatically generate a command sequence that will execute within resource constraints and satisfy flight rules. An automated planning and scheduling system encodes rover design knowledge and uses search and reasoning techniques to automatically generate low-level command sequences while respecting rover operability constraints, science and engineering preferences, environmental predictions, and also adhering to hard temporal constraints. This prototype planning system has been field-tested using the Rocky-7 rover at JPL, and will be field-tested on more complex rovers to prove its effectiveness before transferring the technology to flight operations for an upcoming NASA mission. Enabling goal-driven commanding of planetary rovers greatly reduces the requirements for highly skilled rover engineering personnel. This in turn greatly reduces mission operations costs. In addition, goal-driven commanding permits a faster response to changes in rover state (e.g., faults) or science discoveries by removing the time consuming manual sequence validation process, allowing rapid "what-if" analyses, and thus reducing overall cycle times.

1 Introduction

Unlike more traditional deep space missions, surface roving missions must be operated in a reactive mode, with mission planners waiting for an end of day telemetry downlink--including critical image data--in order to plan the next day's worth of activities. Communication time delays over interplanetary distances preclude simple 'joysticking' of the rover. A consequence of this approach to operations is that the full cycle of telemetry receipt, science and engineering analysis, science plan generation, command sequence generation and validation, and uplink of the sequence, must typically be performed in twelve hours or less. Yet current rover sequence generation [7,8] is manual, with limited ability to automatically generate valid rover activity sequences from more general activities/goals input by science and engineering team members. Tools such as the Rover Control Workstation (RCW) and the Web Interface for Telescience (WITS) provide mechanisms for human operators to manually generate plans and command sequences [1,2]. These tools even estimate some types of resource usage and identify certain flight rule violations. However, they do not provide any means to modify the plan in response to the constraints imposed by available resources or flight rules, except by continued manual editing of sequences. This current situation has two drawbacks. First, the operator-intensive construction and validation of sequences puts a tremendous workload on the rover engineering team. The manual process is error-prone, and can lead to operator fatigue over the many months of mission operations. Second, the hours that must be reserved for sequence generation and validation reduces the time available to the science team to identify science targets and formulate a plan for submission to the engineering team. This results in reduced science return. An automated planning tool would allow the science team and sequence team to work together to optimize the plan. Many different plan options could be explored. The faster turnaround of automated planning also permits shorter than once a day planning cycles.

The RCW software, used to operate the Sojourner rover during the Pathfinder mission, provides visualization for vehicle traverse (movement) planning, a command interface, constraint checking for individual commands, and some resource estimation (for sequence execution time and telemetry volume). However, this tool was never intended for automated goal-based planning of rover activities. To deal with these issues, there is a need for a new tool that is specifically geared toward automated planning.

We are using AI planning/scheduling technology to automatically generate valid rover command sequences from activity sequences specified by the mission science and engineering team. This system will automatically generate a command sequence that will execute within resource constraints and satisfy flight rules. Commanding the rover to achieve mission goals requires significant knowledge of the rover design, access to the low-level rover command set, and an understanding of the performance metrics rating the desirability of alternative sequences. It also requires coordination with external events such as orbiter passes and day/night cycles. An automated planning and scheduling system encodes this knowledge and uses search and reasoning techniques to automatically generate low-level command sequences while respecting rover operability constraints, science and engineering preferences, and also adhering to hard

0-7803-6612-3/01/$10.00©2001 IEEE

temporal constraints. A ground-based interactive planner combines the power of automated reasoning and conflict resolution techniques with the insights of the Science Team or Principal Investigator (PI) to prioritize and re-prioritize mission goals.

2 ASPEN Planning System

Planning and scheduling technology offers considerable promise in automating rover operations. Planning and scheduling rover operations involves generating a sequence of low-level commands from a set of high-level science and engineering goals.

In ASPEN [6], the main algorithm for automated planning and scheduling is based on a technique called *iterative repair* [12,9]. During iterative repair, the conflicts in the schedule are detected and addressed one at a time until conflicts no longer exist, or a user-defined time limit has been exceeded. A conflict is a violation of a resource limitation, parameter dependency or temporal constraint. Conflicts can be repaired by means of several predefined methods. The repair methods are: moving an activity, adding a new instance of an activity, deleting an activity, detailing an activity, abstracting an activity, making a resource reservation of an activity, canceling a reservation, connecting a temporal constraint, disconnecting a constraint, and changing a parameter value. The repair algorithm may use any of these methods in an attempt to resolve a conflict. How the algorithm performs is largely dependent on the type of conflict being resolved.

Rover knowledge is encoded in ASPEN under seven core model classes: activities, parameters, parameter dependencies, temporal constraints, reservations, resources and state variables. An activity is an occurrence over a time interval that in some way affects the rover. It can represent anything from a high-level goal or request to a low-level event or command. Activities are the central structures in ASPEN, and also the most complicated. Together, these constructs can be used to define rover procedures, rules and constraints in order to allow manual or automatic generation of valid sequences of activities, also called plans or schedules.

Once the types of activities are defined, specific instances can be created from the types. Multiple activity instances created from the same type might have different parameter values, including the start time. Many camera-imaging activities, for example, can be created from the same type but with different image targets and at different start times. The sequence of activity instances is what defines the plan.

3 Difficulties in Modeling Constraints

There are several aspects of modeling the Mars rover domain has proven to be very difficult. The power system is a good example. The rovers planned for 2003 contain solar arrays and rechargeable batteries. During the daytime, the power for rover operations is produced using the solar arrays. If the total power drain from operating the rover exceeds the available power from the solar arrays, the batteries must be drawn upon. Because the battery drain is context dependent, the planner needs to understand all the influences and be able to repair conflicts using this knowledge. Additionally, computing the energy taken from a battery is a function of the battery parameters such as temperature, current, voltage, etc. Representing this in a planning model is very difficult.

To solve the power-modeling problem, we initially used a parameter dependency function to calculate the amount of solar power and battery power as a function of the activity duration, available solar array power, available battery power, and power required by the activity. This technique will only work if there are no overlapping power activities because the calculated solar array and battery usage are based on the amount available at the beginning of the activity. In the ASPEN representation, resource use is assumed to be constant over the duration of the activity. In the same manner, we can only request the existing value of a resource at the start of the activity and we must assume that the existing resource profile remains constant until the end of the activity. In the case of overlapping activities that consume power, the first of the two activities would calculate the required power based on the available power at the start time of the first activity. The power available would change during the activity due to the overlap of the second activity.

To address the limitations of the simple timeline representations available to most planner/schedulers (including ASPEN), we have developed a new representation called Generalized Timelines (GTL). GTLs provide a framework for describing unique states and resources and their constraints within an existing planner. We utilize a generic scheduler to reason about these timelines. Combining this with the ASPEN system gives us considerable representational capability. Not only are we able to represent the previously mentioned battery succinctly and accurately, but we can also extend this representation to such states as quaternions (for orientation) or two-dimensional manifolds (for temperature control). In fact, if the validity of state or resource at any time relies only on previous values and the current requirements, then GTL can represent and reason about it.

Another difficulty with modeling the depletable resources in planning systems is the usage profile. Some examples in the spacecraft and rover domains include the memory buffer resource, battery, and fuel. If an activity that uses the memory buffer resource has duration of several minutes, ASPEN will change the value of the resource timeline at the beginning of the activity. In this case, the entire amount of memory buffer resource used by the activity is unavailable for the entire activity. In the example, the memory resource is set to them maximum value at the start of the timeline. This is the equivalent of consuming an entire tank of gas in a car at the beginning of a trip rather than using the gas gradually over the course of the trip. Likely the actual resource usage is linear over the duration of the activity. For long activities, the depletable resource value near the beginning of the activity can be very inaccurate. One workaround for this problem is to split the activity up into several subactivities, each using an equal fraction of the resource. This solution has several problems. First, it increases scheduling complexity by adding multiple activities into the activity database. Second, it creates the problem of trying to determine how many subactivities is enough to accurately model the resource usage. Third, it's non-intuitive for the user to see multiple subactivities that don't represent actual events. The ideal method for modeling resource usage is to use a generalized timeline. Generalized timelines allow modelers to provide a set of functions to describe the depletable resource timeline and its constraints. The generic scheduler can then accurately reason about the described timelines. The example given contains a linear depletable timeline, but any other function could have been modeled as well.

Many rover activities cannot be modeled in planning systems without using external functions. ASPEN has the ability to call external C functions to calculate resource and state usage. An example of this used in the rover model is the telecommunications activity. This activity involves transmitting the data from the rover to Earth during prescribed windows when the Earth is in view. The amount of data to transmit is calculated using a function:

```
transmit amount = minimum[(rate *
duration), (amount in buffer)]
```

The transmit amount is based on the communications rate, duration of the communications activity, and the amount in the storage buffer resource. Specifically, this function transmits the maximum data possible during the communications activity, unless that value is higher than what is in the buffer resource. The external functions are important for accurately modeling many resources in the rover domain. Other examples include calculating camera activity duration and picture size, calculating rough traverse durations and geometry for rover motion activities, adding an activity to turn off a rate sensor after the last motion-related activity, and calculating the earliest start time for an activity that must be the first activity in the schedule.

There are other constraints related to the telecommunications activity that are difficult to model in planning software. There is some uncertainty in the time a communications link will be established due to weather, ground station equipment problems, and ground station operator errors. Because of this uncertainty, the transmitter has to be turned on several minutes before the start of a contact. This constraint leads to an overhead for every communications activity. Because of the power used by the transmitter during this overhead period, it is beneficial to have a fewer number of longer communications activities rather than many short communications activities. The ability to reason about these constraints is important in rover planning.

Another activity constraint is the communications data rate and one-way communications delay. The distance between Earth and Mars varies considerably as the two planets orbit the sun. The time it takes for a signal to reach Earth from Mars or vice-versa varies from about 7 to 20 minutes. The data rate also varies depending on distance, but can be easily calculated for the entire rover mission. The data can be placed in a lookup table within the planning model that is accessed using an external dependency function.

4 Rover Motion Planning

ASPEN is able to reason about simple resource and state constraints. As previously described, it also has the ability to use simple external functions to calculate parameters for resource usage. Many rover constraints are too complex to reason about in a generalized planning system, or use simple parameter functions to solve. For these, an external program must be used to reason about these constraints. ASPEN can interface with other domain-specific programs (or special purpose algorithms) using input files, library calls, a socket interface, or software interfaces.

Motion planning for rovers is a very difficult problem that requires dedicated tools. JPL uses a tool called Rover Control Workstation (RCW) for the motion-planning problem [3]. RCW provides a unique interface consisting of a mosaic of stereo windows displaying the panorama of Mars using camera images from both a lander and a rover. The operator uses liquid crystal shuttered goggles to perceive stereo depth and a special six-degree of freedom input device to move a stereo rover cursor on the screen. RCW displays this rover "CAD" model cursor in real time over the stereo image background, correctly simulating rover perspective, size, and appearance. The operations team uses the RCW to make decisions about where to safely send the rover and what to do when reaching the goal. The RCW also provides a "virtual reality" type flying camera view of the surface using computer generated terrain models [3]. RCW calculates the maximum safe tilt angles for the rover traverse goals input by the user. RCW also calculates the parameters for the rover motion commands. These commands are then output to ASPEN as required activities.

RCW also interfaces with existing surface dynamics simulation software. The uncertainty in the dynamics associated with the quasi-static slip/traction/stability of the soil/machine interface introduces significant uncertainty into the operations of a rover. To address this uncertainty, a linear programming approach represents the inequality-based description of the friction cones together with an equality constraint defining the allowable manifold of forces/torques resisting the impressed forces. Metrics for the slip, traction, and stability, as well as constraint violation information can be obtained by a suitable linear program solver. To deal with the terrain uncertainties, a robust iterative solution to the rover/terrain kinematics solver has been developed. The RCW and associated dynamics simulation software are well suited to solving the rover motion-planning problem.

5 Environment Planning

Another area in which external solvers are used to input state and resource data into ASPEN are environmental conditions. These include orbiter view periods, earth view periods, thermal predictions, and solar array power predictions. The orbiter and Earth view periods are calculated using orbital dynamics analysis software packages such as Satellite Toolkit (STk) or the Satellite Orbit Analysis Program (SOAP). These tools are able to calculate the relative positions of the rover, Mars, Earth, and any orbiting spacecraft. The view periods output by these tools are used to specify when the rover can communicate with the Earth or a Mars-orbiting spacecraft. The view periods are input into ASPEN as fixed activities that change the values of a resource required by the communications, science, or power activities. The solar array and power predictions are calculated using analysis programs that take as input the daily conditions on Mars.

Rover internal environment changes such as instrument or sensor failures are captured within the planning model. When telemetry indicates an equipment failure on the rover, the operators must modify the planning model so the planning software is aware of any new constraints imposed upon the rover planning process.

Risk levels are also incorporated in the planning model. Activity parameters may have different ranges depending on the current risk level of the rover.

6 Mixed-Initiative Rover Planning

While the goal of this work is an integrated fully automated planning system for generating a rover sequence of commands, the human operator is required to be part of the planning process. There is not enough CPU capability onboard current flight rovers to run autonomy software such as path planning or generalized planning. The WITS science-planning tool and the RCW motion-planning tool each require human interaction. These tools allow the user to select rover destinations and science targets in three dimensions using surface imagery. The WITS tool does not actually enforce an order of the goals, but instead relies on ASPEN to build the plan, schedule, and check the resource usage.

Combining these tools with ASPEN creates a "mixed-initiative" end-to-end planning system. The ASPEN operator starts with a set of goals from WITS and RCW, but can then modify the schedule within ASPEN by inserting new goals, changing existing activities, or deleting activities. The schedule is then generated using a forward dispatch algorithm followed by an iterative repair algorithm to fix any conflicts. The repair actions available for each activity are defined within the model for that activity. If the rover resources are over-constrained or under utilized, the user may decide to modify the schedule to optimize the rover resource usage, then re-run the iterative repair algorithm. Several iterations can be performed using ASPEN, WITS, or RCW to modify the goals. This capability allows the rover operations team to try several different scenarios before deciding on the best course of action. The result of this mixed-initiative optimization strategy is a plan with increased science opportunities. Because ASPEN is autonomously checking flight rules and resource constraints, the plan should also be safer than a manually generated plan.

We are also investigating how the user should be interacting with each of the tools involved in building a schedule. The science and engineering users are used to interacting with WITS and RCW, but not with ASPEN. Yet WITS and RCW do not show resource information and activity ordering. Currently the system requires the user to utilize the ASPEN GUI for resource and activity information. In the future, this information could be added directly to the WITS GUI.

7 Status

The focus of our recent work has been to compare the automated ground-based commanding tool to the manual commanding process of the Mars Pathfinder Sojourner rover. The engineering model of the Sojourner rover, Marie Curie, exists at JPL and can be used for field-testing of the generated sequences. The majority of this work done so far focused on creating a rover model using the ASPEN planning system. The Sojourner planning model was built to a level at which all flight rules and constraints could be implemented. We have defined 162 activities of which 63 decompose directly into low-level rover commands.

Initial testing on the Sojourner ASPEN model with a representative set of 136 activities produced a conflict free plan in about 9 seconds. This testing was completed on a Sun Ultra-2 workstation. These relatively quick plan cycles would allow a rover operations team to perform "what-if" analysis on different daily plans. Our goal is that this quick planning capability will be used to generate commands more frequently than once-per-day, if communications opportunities permit.

The next level of testing involved generating plans for two typical Sojourner rover days on Mars. These plans were compared with the manually generated sequences that were run during the Sojourner mission. The automatically and manually generated command sequences were almost identical. The results are summarized in Table 1. Both days produced results very quickly. However, it was a lengthy process (about 10 work weeks) to produce a model that contained constraints and flight rules from a mission not designed for automated planning. Many of the commands were built into macros, which were basically mini-sequences. There was not enough flexibility to utilize all the capabilities of ASPEN in building these plans. If the operations of a mission are designed with an automated planning system in mind, the model building time could be reduced significantly. Once the model is built, valid sequences can be produced very quickly.

	Number of Activities	Planning Time
Sol 18	197	41 seconds
Sol 28	110	6 seconds

Table 1 - Test Results

Eventually we would like to add performance metrics to the planner model to optimize the generated plans. This will enable automated "what-if" analysis to generate plans that maximize science and engineering value. Example performance metrics include maximizing data return, minimizing primary battery usage, and minimizing the difference between data acquired and data returned.

8 Future Work: Mars Exploration Rover

In 2001, we are providing an in-depth validation of the automated command-generation concept using the Mars Exploration Rover (MER) mission. The ASPEN planning and scheduling system will be integrated with the current versions of RCW and WITS. ASPEN will receive RML formatted high-level engineering requests from RCW, and high-level science requests through WITS. ASPEN will then automatically generate validated rover-command sequences that satisfy these requests and provide those RML formatted sequences to RCW. The ASPEN Java-based interface will enable the user to access planned activities and to observe resource and state constraints. The computation intensive aspects of the commanding capability (such as the planner/scheduler, path planner, uncertainty estimation software, vision and image processing software, etc.) will reside on one or more rover workstations based in a central location.

MER has similar operations constraints as previous JPL rovers. Power is the most limited resource, followed by communications bandwidth. The bandwidth is further constrained because there will be two rovers operating simultaneously. Each rover has the ability to communicate directly with Earth through the Deep Space Network, or through the orbiting Mars Odyssey or Mars Global Surveyor using UHF communications. ASPEN is particularly well suited to building schedules that optimize science based on resource constraints such as power and bandwidth.

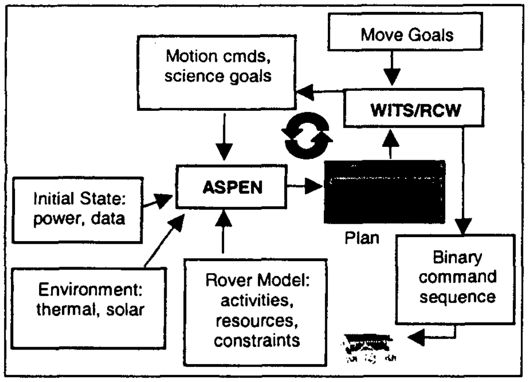

Figure 1 - End-to-End Commanding System

The end-to-end data flow for this system is shown in Figure 1. The interaction between ASPEN and RCW/WITS is an iterative process. RCW will receive high-level motion goals from the user through a 3-dimensional interface utilizing Martian surface imagery. RCW will output detailed traverse commands to ASPEN for inclusion into the schedule. ASPEN will merge these motion commands with high-level science goals from WITS to produce an intermediate level plan. The plan will be output to RCW to update motion commands as necessary. Science goals can be updated through the ASPEN interface or additional high-level science goals can be input through WITS. This process will continue until an acceptable plan is generated. Finally a time ordered list of commands is output for sequence generation.

Work is continuing on creating a high-fidelity MER planning model. The automated planning system may be used for goal-based operations during field-testing of MER prior to launch in 2003. The goal of this work is to perform shadow testing in parallel with MER operations to evaluate the effectiveness of automated planning. In addition, we are formulating plans for using this architecture in field-testing of the Rocky-8 rover starting in Fall 2001. (See Figure 2.) These tests would likely be performed initially in the JPL Mars Yard, followed by demonstrations in desert sites in California. The Rocky-8 rover is similar to the rover NASA plans for launch in 2007. The experiences learned from field-testing an automated planner with Rocky-8 will lead to a more robust planning system for the 2007 mission.

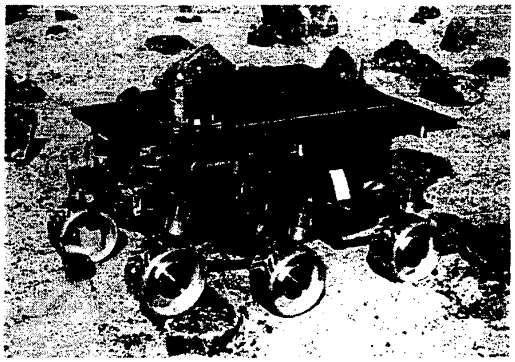

Figure 2 - Rocky 8 Rover

9 Onboard Rover Planning

In addition to the ground-based planning previously described, we are developing a dynamic, onboard planning system for rover sequence generation. The CASPER (Continuous Activity Scheduling, Planning, Execution and Re-planning) system [4,5], is a dynamic extension to ASPEN, which can not only generate rover command sequences but can also dynamically modify those sequences in response to changing operating context. If orbital or descent imagery is available, CASPER interacts with a path planner to estimate traversal lengths and to determine intermediate waypoints that are needed to navigate around known obstacles.

Once a plan has been generated it is continuously updated during plan execution to correlate with sensor and other feedback from the environment. In this way, the planner is highly responsive to unexpected changes, such as a fortuitous event or equipment failure, and can quickly modify the plan as needed. For example, if the rover wheel slippage has caused the position estimate uncertainty to grow too large, the planner can immediately command the rover to stop and perform localization earlier than originally scheduled. Or, if a particular traversal has used more battery power than expected, the planner may need to discard one of the remaining science goals. CASPER has been integrated with control software from the JPL Rocky 7 rover [10,11] and is currently being tested on Rocky 7 in the JPL Mars Yard.

10 Conclusions

Current approaches to rover-sequence generation and validation are largely manual, resulting in an expensive, labor, and knowledge intensive process. This is an inefficient use of scarce science-PI and key engineering staff resources. Automation as targeted by this tool will automatically generate a constraint and flight rule checked time ordered list of commands and provides resource analysis options to enable users to perform more informative and fast trade-off analyses. Initial tests have shown planning times on the order of seconds rather than hours. Additionally, this technology will coordinate sequence development between science and engineering teams and would thus speed up the consensus process.

Enabling goal-driven commanding of planetary rovers by engineering and science personnel greatly reduces the workforce requirements for highly skilled rover engineering personnel. The reduction in team size in turn reduces mission operations costs. In addition, goal-driven commanding permits a faster response to changes in rover state (e.g., faults) or science discoveries by removing the time consuming manual sequence validation process, allowing "what-if" analyses, and thus reducing overall cycle times.

11 Acknowledgement

The TMOD Technology Program partly funded the research described in this paper. The Mars Surveyor Operations Office, the Athena Rover Office, and the Athena Precursor Experiment funded the RCW development work. All work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

12 Bibliography

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

[2] P. Backes, K. Tso, and G. Tharp, "The Web Interface for Telescience," Journal of Presence, vol. 9, number 5, pages 529-537, October 1999.

[3] B. Cooper, "Driving On The Surface Of Mars Using The Rover Control Workstation," SpaceOps 98, Japan, 1998.

[4] S. Chien, R. Knight, A. Stechert, R. Sherwood, G. Rabideau, "Integrated Planning and Execution for Autonomous Spacecraft", Proceedings of the IEEE Aerospace Conference (IAC), Aspen, CO, March 1999.

[5] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, "Using Iterative Repair to Improve Re-sponsiveness of Planning and Scheduling," Proceedings of the 5th Intl. Conference on Artificial Intell-igence Planning and Scheduling, Breckenridge, CO, April 2000.

[6] S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, D. Tran , "ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling," SpaceOps, Toulouse, France, June 2000.

[7] A. Mishkin, "Field Testing on Mars: Experience Operating the Pathfinder Microrover at Ares Vallis," presentation at Field Robotics: Theory and Practice workshop, May 16 1998, at the 1998 IEEE International Conference on Robotics and Automation, Leuven, Belgium.

[8] A. Mishkin, J. Morrison, T. Nguyen, H. Stone, B. Cooper, B. Wilcox, "Experiences with Operations and Autonomy of the Mars Pathfinder Microrover," proceedings of the 1998 IEEE Aerospace Conference, March 21-28 1998, Snowmass at Aspen, Colorado.

[9] G. Rabideau, R. Knight, S. Chien, A. Fukunaga, A. Govindjee, "Iterative Repair Planning for Spacecraft Operations in the ASPEN System," International Symposium on Artificial Intelligence Robotics and Automation in Space (ISAIRAS), Noordwijk, The Netherlands, June 1999.

[10] R. Volpe, T. Estlin, S. Laubach, C. Olson, and J. Balaram, "Enhanced Mars Rover Navigation Techniques" To appear in the Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, April 2000.

[11] R. Volpe, I. Nesnas, T. Estlin, D. Mutz, R. Petras, H. Das, "The CLARAty Architecture for Robotic Autonomy." Proceedings of the 2001 IEEE Aerospace Conference, Big Sky, Montana, March 10-17, 2001.

[12] Zweben, M., Daun, B., Davis, E., and Deale, M., "Scheduling and Rescheduling with Iterative Repair," Intelligent Scheduling, Morgan Kaufmann, San Francisco, 1994, pp. 241-256.