An Integrated Planning and Scheduling Prototype for Automated Mars Rover Command Generation

Rob Sherwood, Andrew Mishkin, Steve Chien, Tara Estlin, Paul Backes, Brian Cooper, Gregg Rabideau, Barbara Engelhardt

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109
818-393-5378
818-393-4100 (fax)
firstname.lastname@jpl.nasa.gov

Abstract

With the arrival of the Pathfinder spacecraft in 1997, NASA began a series of missions to explore the surface of Mars with robotic vehicles. The Pathfinder mission included Sojourner, a six-wheeled rover with cameras and a spectrometer for determining the composition of rocks. The mission was a success in terms of delivering a rover to the surface, but illustrated the need for greater autonomy on future surface missions. The operations process for Sojourner involved scientists submitting to rover operations engineers an image taken by the rover or its companion lander, with interesting rocks circled on the images. The rover engineers would then manually construct a one-day sequence of events and commands for the rover to collect data of the rocks of interest. The commands would be uplinked to the rover for execution the following day. This labor-intensive process was not sustainable on a daily basis for even the simple Sojourner rover for the two-month mission. Future rovers will travel longer distances, visit multiple sites each day, contain several instruments, and have mission duration of a year or more. Manual planning with so many operational constraints and goals will be unmanageable. This paper discusses a proof-of-concept prototype for ground-based automatic generation of validated rover command sequences from high-level goals using AI-based planning software.

Keywords

demonstration, planning, scheduling, robotic applications, rovers, aerospace applications

1 Demonstration

We will demonstrate a ground based automated planning prototype for a multi-instrument Mars rover using the Web Interface for Telescience (WITS) front end to generate science goals that will be automatically planned with the ASPEN planner. Using WITS, new goals can be added to the existing plan, resulting in conflicts that will be solved by ASPEN using an iterative repair algorithm. Reasoning about complex resource timelines utilizing the generalized timeline approach highlighted in this paper will be demonstrated. The end result will be a valid sequence of commands for execution on a rover. This software will be demonstrated on a laptop computer or a Sun workstation.

2 Introduction

Since our first close-up picture of Mars in 1965, spacecraft voyages to the Red Planet have revealed a world strangely familiar, yet different enough to challenge our perceptions of what makes a planet work. Every time we feel close to understanding Mars, new discoveries send us straight back to the drawing board to revise existing theories. Over the past three decades, spacecraft have shown us that Mars is rocky, cold, and sterile beneath its hazy, pink sky. We've discovered that today's Martian wasteland hints at a formerly volatile world where volcanoes once raged, meteors plowed deep craters, and flash floods rushed over the land. And Mars continues to throw out new enticements with each landing or orbital pass made by our spacecraft.

Among our discoveries about Mars, one stands out above all others: the possible presence of liquid water on Mars, either in its ancient past or preserved in the subsurface today. Water is key

because almost everywhere we find water on Earth, we find life. If Mars once had liquid water, or still does today, it's compelling to ask whether any microscopic life forms could have developed on its surface. To discover the possibilities for life on Mars—past, present or our own in the future—the Mars Program has developed an exploration strategy known as "Follow the Water." Following the water begins with an understanding of the current environment on Mars. We want to explore observed features like dry riverbeds, ice in the polar caps and rock types that only form when water is present. We want to look for hot springs, hydrothermal vents or subsurface water reserves. We want to understand if ancient Mars once held a vast ocean in the northern hemisphere as some scientists believe, and how Mars may have transitioned from a more wet environment to the dry and dusty climate it has today. Searching for these answers means delving into the planet's geologic and climate history to find out how, when, and why Mars underwent dramatic changes to become the forbidding, yet promising, planet we observe today.

To pursue these goals, all of our future missions will be driven by rigorous scientific questions that will continuously evolve as we make new discoveries. Brand new technologies will enable us to explore Mars in ways we never have before, resulting in higher-resolution images, precision landings, longer-ranging surface mobility and even the return of Martian soil and rock samples for studies in laboratories here on Earth. (Birchak-Birkman et al., Apr. 2001)

The Mars Pathfinder mission sent the first mobile robot, the Sojourner rover, to the surface of Mars in 1997. NASA plans to send 2 more rovers to Mars in 2003. Unlike orbiting spacecraft, 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 command sequence generation is manual (Mishkin, et al., 1998), with limited ability to automatically generate valid rover activity sequences from more general activities/goals input by science and engineering team members.

The motion-planning tool Rover Control Workstation (RCW) and the science-planning tool Web Interface for Telescience (WITS) provided mechanisms for human operators to manually generate plans and command sequences on the Mars Pathfinder mission. (Backes, et. al, 1998) These tools even estimated some types of resource usage and identified 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 allows shorter than once a day planning cycles. In 2000, the Modified Antarctic Mapping Mission (MAMM) on the Canadian Space Agency's RadarSAT used the ASPEN planner for mixed-initiative mission operations design, planning, and replanning, to collect complete inferometric coverage of the Antarctic continent. Specifically, ASPEN was used to create downlink schedules given a set of imaging activities, validate compliance with mission and operation constraints, perform "what-if" studies during mission design, and available for fast turn-around replanning in response to failures during the mission. (Smith, et al., 2001)

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 goals specified by the mission science and engineering team. This system will automatically generate a command sequence to accomplish the given goals that will execute within resource constraints and satisfy flight rules. This prototype is based on ASPEN, the Automated

Scheduling and Planning Environment. (Chien, et al., 2000) 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 an engineering prototype rover, Rocky-7, at JPL. (Backes, et al., 1999) The Rocky-7 rover was created to demonstrate new technology concepts for use in a long-range (50km) traversal across Mars, and is similar in size to Sojourner.

The planning system 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 planning times while maximizing science return.

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 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. This mixed-initiative planning architecture lends itself well to the rover-planning problem.

3 ASPEN Planning System

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

ASPEN (Chien, et al., 2000) is an object-oriented planning and scheduling system that provides a reusable set of software components that can be tailored to specific domains. These components include:

- An expressive constraint modeling language to allow the user to define naturally the application domain
- A constraint management system for representing and maintaining spacecraft and rover operability and resource constraints, as well as activity requirements
- A set of search strategies for plan generation and repair to satisfy hard constraints
- A language for representing plan preferences and optimizing these preferences
- A soft real-time replanning capability
- An activity representation with activity decompositions
- Functions for defining dependencies between user parameters
- A temporal reasoning system for expressing and maintaining temporal constraints
- A graphical interface for visualizing plans/schedules (for use in mixed-initiative systems in which the problem solving process is interactive).

The job of a planner/scheduler, whether manual or automated, is to accept high-level goals and generate a set of low-level activities that satisfy the goals and do not violate any of the rover operational rules or constraints. Goal-based rover planning 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. ASPEN provides a Graphical User Interface (GUI) for manual generation and/or manipulation of activity sequences. (See Figure 1.)

In ASPEN, the main algorithm for automated planning and scheduling is based on a technique called *iterative repair* (Rabideau, et al., 1999, Zweben et al., 1994). During iterative repair, the conflicts in the schedule are addressed one at a time until conflicts no longer exist, or a user-defined time limit has been exceeded. A conflict occurs when a resource requirement, parameter dependency

or temporal constraint is not satisfied. 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, decomposing an activity into subgoals, abstracting an activity, making a resource reservation on 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 and the activities, states, and resources involved in the conflict.

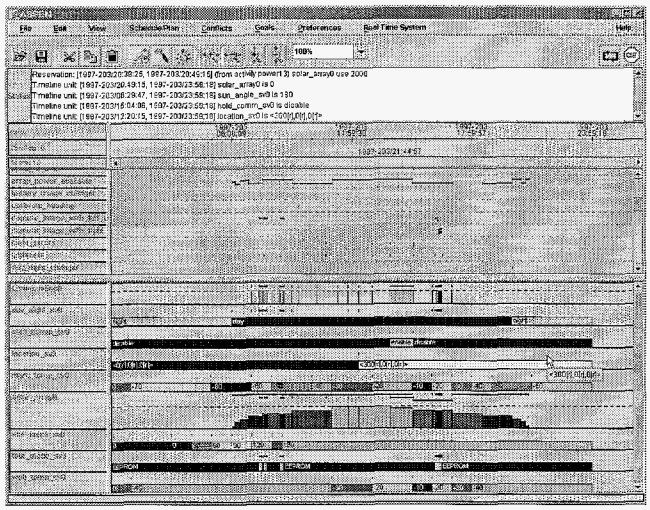

Figure 1 - ASPEN GUI

```
Activity move_rover {
 constraints =
   starts after end_of
   rate_sensor_heat_up by [2m,3m];
 reservations =
   solar_array_power use 35,
   rate_sensor_state change_to "on",
   target_state must_be "ready";
};
```

Figure 2 - ASPEN Modeling Language Example

Rover knowledge is encoded in ASPEN under seven core model classes: activities, parameters, parameter dependencies, temporal constraints, reservations, resources and state variables. (Sherwood, et al., 2001) 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 requested by the user, to a low-level event or command. Activities are the central structures in ASPEN, and also the most complicated. In groups, 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, as in object-oriented programming. Multiple activity instances created from the same type might have different parameter values, including their start time. Many camera-imaging activities, for example, can be created from the same type but with different image targets and different start times. The sequence of activity instances is what defines the plan.

Rover mission flight rules and constraints are defined within the activities. The flight rules can be defined as temporal constraints, resource constraints, or system state constraints. Figure 2 contains an example of an ASPEN activity definition including a temporal constraint and resource reservations. Temporal constraints are defined between activities. Figure 2 contains a temporal constraint definition for a rate sensor. This is a device that measures the speed of the rover while it's moving. In this example, the rate sensor must warm up for two to three minutes before moving the rover. In ASPEN, this might be modeled within a "move rover" activity as shown in Figure 2. A constraint within the "move rover" activity requires that it start after the end of the rate_sensor_heat_up activity. Presumably the rate_sensor_heat_up activity turns the rate sensor on. This constraint acts as a temporal dependency between the two activities.

Constraints can also be state or resource related. State constraints can either require a particular state or change a state resource to a particular state. Resource constraints can use a particular amount of a resource. Resources with a capacity of one are called atomic resources (i.e., camera resource) ASPEN also uses non-depletable and depletable resources. Non-depletable resources do not need to be replenished. An example would be the rover solar array power. Depletable resources are similar to non-depletable except that their capacity is diminished after use. In some cases their capacity can be replenished by the activities (memory buffer capacity) and in other cases it cannot (spacecraft

propellant). Resource and state constraints are defined within activities using the keyword "reservations." See Figure 2 for an example.

4 Difficulties in Modeling Rover 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. (Knight, et al., 2001) 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.

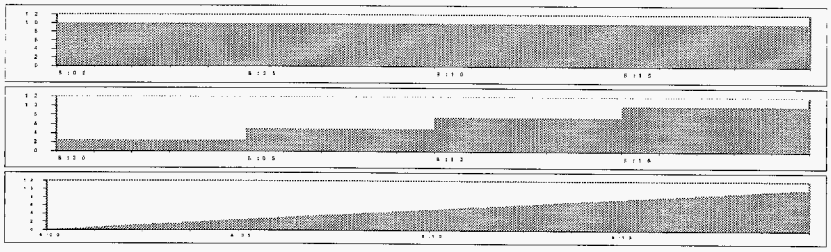

Figure 3 – Examples of (1) Simple Resource Modeling in ASPEN; (2) Using 4 Subactivities to Split-up Resource Usage; (3) Using a Generalized Timeline to Model Resource Usage

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. (See example 1 in Figure 3.) 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. (See example 2 in Figure 3.) 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. (See example 3 in Figure 3.) 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.

5 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 (Cooper, 1998). 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 (Cooper, 1998). 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.

6 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.

7 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. Both the WITS science-planning tool and the RCW motion-planning tool require human interaction. These tools allow the user to select rover destinations and science targets in three dimensions using surface imagery. 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. ASPEN will then locally repair any new conflicts using the iterative repair algorithm. Additional iterations can be performed using WITS and RCW if necessary. 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.

Operating a rover on the surface of Mars introduces a new environment each time the rover changes location. Planning the rover activities for new environments requires either sophisticated onboard autonomy or mixed-initiative planning. Current rover missions being built for Mars do not have the computing power necessary for full onboard autonomy. The constantly changing environment lends itself well to mixed-initiative planning.

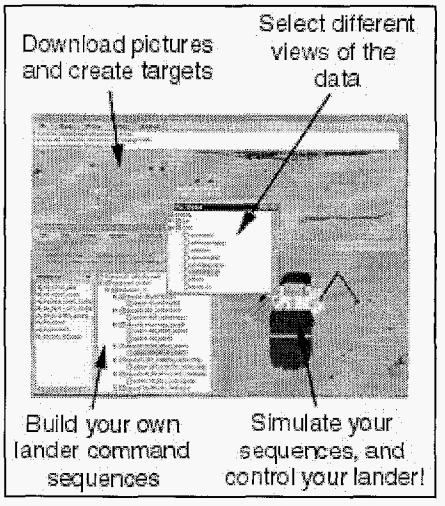

Figure 4 - WITS GUI

8 Status

Initial work in 1998 consisted of a preliminary proof of concept demonstration in which we used automated planning and scheduling technology integrated with WITS to demonstrate automated commanding for the Rocky-7 rover from the WITS interface. (Backes, et al., 1999) (See Figure 4.) The Rocky7 research rover has been developed at JPL by the Long Range Science Rover task of the NASA Telerobotics program. The rover was tested using

WITS and ASPEN in the JPL Mars Yard, a simulated Mars landscape. The 1.5-meter tall deployable mast camera on the rover took a set of panorama images. Using these images, the WITS user selected a dig target locations, science imaging targets, spectrometer imaging targets, and their associated parameters and priorities. The WITS tool was used to visualize the terrain around the rover, generate the initial science 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 final sequence was then executed on the rover in the JPL Mars Yard.

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 (Mishkin, et al., 1998). The engineering model of the Sojourner rover, Marie Curie, exists at JPL and can be used for field-testing of the generated sequences. The Marie Curie rover was scheduled to fly onboard the Mars 2001 Lander mission before being cancelled in early 2000. The majority of this work done so far focused on creating a rover model using the ASPEN planning system for use on this 2001 Lander mission. The rover-planning model was built at a level for which all flight rules and constraints could be implemented. The resources include the three cameras, Alpha Proton X-Ray Spectrometer (APXS), APXS deploy motor, drive motors, solar array, battery, RAM usage, and non-volatile memory usage. There are 27 different state variables used to track the status of various devices, modes, and parameters. Some of these parameters map directly onto rover internal parameters and others are related to the ASPEN specific model. We have defined 162 activities of which 63 decompose directly into low-level rover commands.

There are several constraints that affect overall operations of the Marie Curie rover. These include:

- Earth-Mars one-way communications time delay (5-20 minutes)
- Limited communications bandwidth (generally < 10 Mbits downlink per sol[1] available to rover)
- Limited communications opportunities (1 command uplink, 2 telemetry downlinks per sol)

The power system is the single most important resource for the Marie Curie Rover. This system consists of a .22 square meter solar array and 9 LiSOCL batteries. The batteries on Marie Curie are primarily used during the night for APXS data collection. They are non-rechargable batteries and therefore modeled as non-renewable depletable resources. The solar array is the primary power source used during the day. The predicted available solar power profile throughout the Mars day must be input before planning begins. Using a daily model is required due to changing solar array power available as a result of degradation from dust accumulation and seasonal solar irradiation variability. The angle of the solar array, which depends on the terrain, will also affect the availability of solar energy. Solar array angle estimates are generated by the RCW for input into ASPEN.

A typical Mars day for the Marie Curie rover might involve a subset of the following activities:

- Complete an APXS data collection that was carried out during the prior night
- Capture a rear image of the APXS site
- Traverse to an appropriate site and perform a series of soil mechanics experiments
- Traverse to a designated rock or soil location
- Place the APXS sensor head
- Capture end-of-day operations images with its forward cameras
- Begin APXS data collection (usually occurs overnight while the rover is shutdown)
- Shut down for the night

The exact position of the rover after a traverse activity is subject to dead-reckoning error. This error occurs when estimates of the rover's position are based solely on the wheel odometry (e.g. dead-reckoned estimates). These estimates accrue a significant error as the rover traverses across the terrain. The timing of traverse activities is also non-deterministic. Because of the inherent problems of coordinating activities between the event-based rover and time-based lander, wait commands are used to synchronize activities. When the lander is imaging the rover after a traverse, a wait command is used to ensure the rover will remain stationary at its destination until the lander completes imaging.

---

[1] A Sol is a Martian day, equivalent to about 24 hours and 39 minutes

Because the rover executes commands serially, this ensures that another command will not start execution before the previous command has completed. All rover traverse goals are generated using the RCW. RCW outputs position information to ASPEN to set the rover end position state.

Rover data storage is a scarce resource that must be tracked within the ASPEN model. The largest consumer of data storage is the camera image activity. This activity can fill the on-board data storage if a telemetry session with the lander is not available during the data collection. ASPEN will track the data storage resource to ensure that all data is downlinked before the buffer is completely full.

Initial testing on the Marie Curie 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.

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

Table 1 - Test Results

The next level of testing involved generating plans for two typical rover days on Mars. These plans were compared with the manually generated sequences that were run during the Sojourner mission. The command sequences were very similar, and the details of each plan are summarized in Table 1. Both days produced results very quickly, in seconds rather than the several hours it took to construct the plans manually. 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 essentially 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.

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. (Rabideau, et al., 2000)

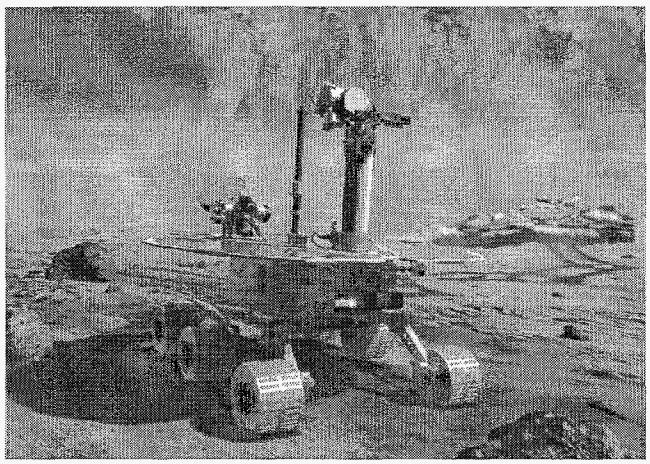

Figure 5 - Mars Exploration Rover

9 Future Work: Mars Exploration Rover and Beyond

The goal of this automated planning work is a deployment on a future planetary rover such as the Mars Exploration Rover (MER) mission (G. Birchak-Birkman, et al., May 2001). (See Figure 5.) Two rovers are planned for launch from Cape Canaveral, Florida, during June 2003 for an early 2004

arrival. The rovers will be identical to each other, but will land at different regions of Mars. Each rover will carry a sophisticated set of instruments, the Athena payload, that will allow it to search for evidence of liquid water that may have been present in the planet's past. Each rover has a mass of nearly 150 kilograms (about 300 pounds) and has a range of up to 100 meters (about 110 yards) per sol, or Martian day.

The Athena payload (G. Birchak-Birkman, et al., Apr. 2001) consists of the Pancam Mast Assembly (PMA), which includes a high-resolution stereo panoramic multi-spectral imaging system (Pancam), and Mini-TES, an emission spectrometer operating in the 5 to 29 micrometer spectral window. Mini-TES is also designed to be a point spectrometer that gathers thermal data as individual spectra or as arrays for key targets identified using Pancam data.

Three additional instruments are to be placed on the end of the Instrument Deployment Device (IDD). The IDD is a deployable arm/instrument package that will perform in-situ analyses of rocks and soils. Instruments on the IDD are the Alpha Particle X-ray Spectrometer (APXS), the Mössbauer Spectrometer, and the Microscopic Imager. Use of all three instruments provides detailed elemental, mineralogical, and textural characterization of rock and soil targets.

The final capability included in the Athena payload is rock abrasion tool, or "RAT," which will be used to expose fresh rock surfaces for study. Additional instruments on the MER rovers will include Navcam stereo imaging systems on the PMA for path planning, and body-mounted Hazcams that image the near terrain to the front and rear of the rover for hazard detection and arm deployment planning.

Each MER rover is designed to conduct traverse science, mast-based remote sensing and in-situ analyses, over a distance of approximately 600 meters during the nominal operational period of 90 sols, but could continue longer, depending on the health of the vehicles. Due to the communication time delays between Earth and Mars, the rover must perform its traverses autonomously, with human operator input generally limited to designation of traverse waypoints and high level commands specifying experiment execution once per sol.

The landed portion of the Mars Exploration Rover mission features a design dramatically different from Mars Pathfinder's. Where Pathfinder had scientific instruments on both the lander and the Sojourner rover, these larger rovers will carry their instruments with them. In addition, these exploration rovers will be able to travel almost as far in one Martian day as the Sojourner rover did over its entire lifetime.

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 in the presence of resource constraints such as power and bandwidth.

In 2001, we are providing an in-depth validation of the automated command-generation concept using the MER mission. The ASPEN planning and scheduling system will be integrated with the current versions of RCW and WITS. ASPEN will receive extensible markup language (XML) 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 XML formatted sequences to RCW. The ASPEN Java-based GUI 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.

The end-to-end data flow for this system is shown in Figure 6. 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.

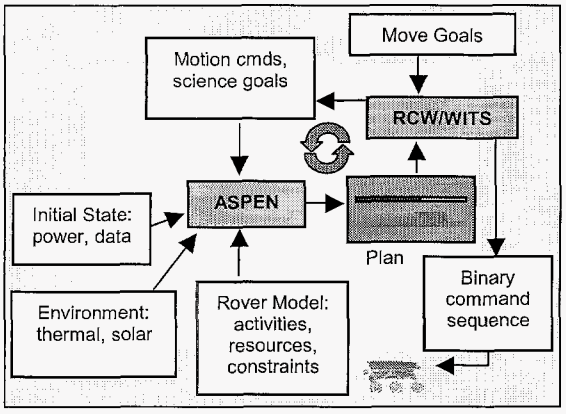

Figure 6 - End-to-End Commanding System

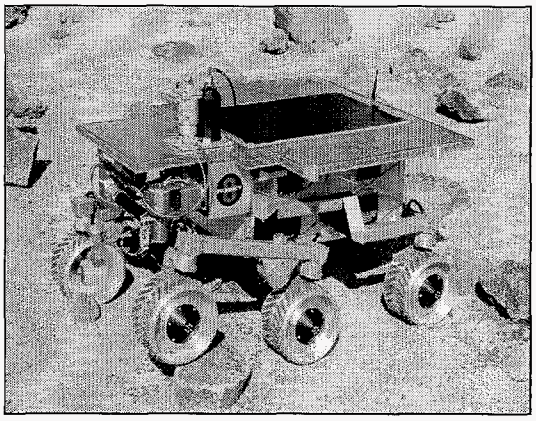

Figure 7 - Rocky 8 Rover

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 7.) 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 that NASA plans to 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.

A summary of the ground-based planning work is contained in Table 2. This summary includes past and future rover missions, as well as engineering field prototypes.

Rover/Mission	Status of Automated Commanding
Rocky-7	Field tested in 1998 with limited set of goals using WITS interface
Sojourner/Marie Curie	Fully developed model of rover, flight rules, constraints. Compared with Sojourner surface operations for 2 sample days of operations
MER	Model being built for possible shadow mode testing during field tests and Mars operations using WITS, RCW
Rocky-8	Model will be built Fall 2001 for field testing in early 2002 using Rocky-8 & WITS
2007 Rover	Ground-based automated planning used for operations

Table 2 – Summary of Automated Rover Planning Work

10 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 (Chien et al., 1999; Chien et al., 2000), 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 terrain images from a Mars orbiter spacecraft camera or Mars Lander descent camera are 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 (Volpe et al., 2001, Volpe et al., 2000) and is currently being integrated with the Rocky 8 control software. Onboard planning for both rovers is currently being tested in the JPL Mars Yard.

11 Conclusions

Current approaches to rover-sequence generation and validation are largely manual, resulting in a labor and knowledge intensive process. This is an inefficient use of scarce science-investigator 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 provide 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. Traditionally, there has been little coordination between these teams because science and motion planning use different tools and there wasn't enough time.

Enabling goal-driven commanding of planetary rovers 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 during operations, and thus reducing overall planning times.

12 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.

13 Bibliography

P. Backes, K. Tso, and G. Tharp. "Mars Pathfinder mission Internet-based operations using WITS. Proceedings IEEE International Conference on Robotics and Automation," pages 284-291, Leuven, Belgium, May 1998.

P. Backes, G. Rabideau, K. Tso, S. Chien, "Automated Planning and Scheduling for Planetary Rover Distributed Operations," Proceedings of the IEEE Conference on Robotics and Automation (ICRA), Detroit, Michigan, May 1999.

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

G. Birchak-Birkman, et al., "Mars Exploration Program Overview," Jet Propulsion Laboratory, Available from: http://mars.jpl.nasa.gov/overview/index.html [Accessed 17 Apr 2001].

G. Birchak-Birkman, et al., "Mars Exploration Program Overview," Jet Propulsion Laboratory, Available from: http://mars.jpl.nasa.gov/missions/future/2003.html [Accessed 4 May 2001].

J. Bresina, K. Golden, D. Smith, and R. Washington, Increasaed Flexibility and Robustness of Mars Rovers, Proceedings of the 5th International Symposium on AI, Robotics, and Automation in Space, Noordwijk, The Netherlands, June 1999.

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.

S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, "Using Iterative Repair to Improve Responsiveness of Planning and Scheduling," Proceedings of the 5th Intl. Conference on Artificial Intelligence Planning and Scheduling, Breckenridge, CO, April 2000.

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.

R. Knight, G. Rabideau, S. Chien. "Extending the Representational Power of Model-Based Systems using Generalized Timelines." The 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space: A New Odyssey. Montreal, Canada. June 2001.

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.

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.

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.

G. Rabideau, B. Engelhardt, S. Chien, "Using Generic Preferences to Incrementally Improve Plan Quality," in Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling, Breckenridge, CO, April 2000.

R. Sherwood, et al., "ASPEN User's Guide," Jet Propulsion Laboratory, Available from: http://aspen.jpl.nasa.gov/usersguide.pdf [Accessed 1 May 2001].

B. Smith, B. Engelhardt, and D. Mutz, "The RadarSAT MAMM Automated Mission Planner" in Proceedings of International Conference on Applications of Artificial Intelligence (IAAI)., August, 2001.

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.

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.

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.