Mixed-initiative constraint-based activity planning for Mars Exploration Rovers

John Bresina, Ari K. Jónsson, Paul H. Morris, Kanna Rajan
NASA Ames Research Center
Mailstop 269-2
Moffett Field, CA 94035
{bresina,jonsson,pmorris,kanna}@email.arc.nasa.gov

Abstract

In January, 2004, two NASA rovers, named Spirit and Opportunity, successfully landed on Mars, starting an unprecedented exploration of the Martian surface. Power and thermal concerns constrained the duration of this mission, leading to an aggressive plan for commanding both rovers every day. As part of the process for generating these command loads, the MAPGEN tool provides engineers and scientists an intelligent activity planning tool that allows them to more effectively generate complex plans that maximize the science return each day. The key to the effectiveness of the MAPGEN tool is an underlying artificial intelligence plan and constraint reasoning engine. In this paper we outline the design and functionality of the MAPGEN tool and focus on some of the key capabilities it offers to the MER mission engineers.

Introduction

The Mars Exploration Rovers (MER) Mars 2003 mission is one of NASA?s most ambitious science missions to date. The rovers will be launched in the summer of 2003 and each rover will carry a rich suite of instruments to conduct remote and in-situ observations to elucidate the planet?s past climate, water activity, and habitability. They will arrive in January and February 2004 at two scientifically distinct sites. Each rover will have an operational lifetime of 90 Martian sols or more and will have the capability to traverse an integrated distance of one kilometer or more, although the maximum range from the landing site may be less than one kilometer. Among the scientific objectives of the MER Mission are to: i) determine the aqueous, climatic, and geologic history of a site on Mars where conditions may have been favorable to the preservation of evidence of pre-biotic or biotic processes ii) to identify hydrologic, hydrothermal, and other processes that have operated at the landing site iii) to identify and investigate Martian rocks and soils that have the highest possible chance of preserving evidence of ancient environmental conditions and possible pre-biotic or biotic activity and iv) to respond to other discoveries associated with rover-based exploration. Science is the primary driver of MER and, as a consequence, planning for scientific activities using the suite of instruments onboard the rovers within the restrictive bounds of the resources available is crucial. To address this criticality, the MER project has selected MAPGEN

(Mixed-Initiative Activity Plan GENerator) as an activity planning tool.

MAPGEN is a tool for science activity planning. The primary users of this tool are to be MER mission tactical planners and scientists who will be manipulating the science objectives in concert with specific engineering. The capabilities offered by MAPGEN are to assist the tactical planners in building a complex, yet safe activity plan that achieves as much of the science objectives as possible for each command cycle. Among the high-level capabilities are:

> Active flight rule enforcement during plan editing
>
> Automated plan completion methods that have different scopes
>
> Automatic handling of support activities like CPU and heating
>
> Advanced editing capabilities that automatically reestablish flight rules and constraints

In this paper, we outline the capabilities and design of the MAPGEN system, and discuss some of the issues that have arisen in development, integration, fielding and use. In the next section, we discuss the requirements of the activity planning process for MER and specifically the requirements on the mixed-initiative planning tool MAPGEN. We then discuss the underlying constraint-based planning framework and one of the many ways in which constraint-based planning was adapted to the needs of the users. We describe the rover and the flight rules that are modeled in the planning tool. We then conclude with a discussion of some of the challenges in this project, and remarks on how the system has performed in the mission.

Requirements

The Mixed Initiative Planner is commanded through the normal APGEN GUI interface. There are extra menu items for planner-specific commands.

In addition, many of the regular APGEN commands have effects that are slightly modified by the active constraint enforcement of the Planner.

After a menu item is selected, the normal changes to the APGEN database are first performed. Then a synchronization step occurs where the Planner database is changed to reflect the changes to the

APGEN database. Next, the constraint engine in the Planner undergoes a propagation to enforce consistency and detect unresolvable inconsistencies. If there are unresolvable inconsistencies, the user command is undone, and a warning message is posted to the user.

Otherwise, a resynchronization step occurs where the effects of the propagation on the Planner database are fed back in turn to APGEN.

The MAPGEN command interface provides the following capabilities:

o The ability to place activities in the plan. When it is unable to fit all activities into the plan, MAPGEN rejects activities based on the observation and activity priorities and moves rejected activities into the hopper.

o The ability to determine a range of acceptable start times for activities in the plan and set the APGEN start-time attribute to the earliest start time for every activity in the plan.

o The ability for "constrained" moves of activities in the plan within valid extents determined by the Planner to be consistent with the current plan.

o The ability for the user to directly edit a plan, including adding, deleting, moving activities and modifying activity parameters. After such edits, MAPGEN re-checks the current constraints and flags anyviolations.

o The ability to attempt to correct inconsistencies in imported activities when being read in, as follows:

o Science Activities shall be placed within the permitted range of start times in the constraints read in.

o If an activity has constraints that cannot be satisfied, then the activity shall be made unscheduled and put in the hopper.

o The ability to add required heaters automatically in the plan.

System design

The MAPGEN system consists of two components; a plan display and editing tool called APGEN, and an underlying constraint and plan reasoning engine called EUROPA. The two are linked with an interface component that handles interactions between the two main components and packages the capabilities of the autonomous reasoning system into the functionality items available to the user. A third part of the system is an external constraint editor that provides constraint editing services that could not reasonably be implemented within APGEN.

The front end of the MAPGEN system is a plan editing system called APGEN. This system is well established in the spacecraft operation community. It offers a generic plan editing capability through a user interface. It also provides a set of underlying modeling capabilities that can be used to calculate states and numerical resources for a given activity plan. Finally, the system supports checking flightrules and highlighting any violations. APGEN can be adapted to different missions by specifying the activity types, modeling rules, and flight rules, using external declarative files.

The EUROPA system is a constraint-based planning framework that supports complex domain descriptions, time and resources. In MAPGEN, this system is utilized as an active plan database. The plan in APGEN is mirrored as a constraint-based plan in EUROPA, along with user constraints, planner decisions and other input. As changes are made, the EUROPA system updates its database, using propagation, active domain rule enforcement and other automated reasoning techniques. These updates are then passed back to the APGEN front end.

One of the key challenges of developing MAPGEN into a useful interactive tool was to design an interface between the APGEN front and the EUROPA planner backend that would be efficient, correct and give the user access to suitable functionality and feedback. Simply offering the user access to fully automated planning and then present them with the results is of little use in activity planning for the MER mission. As a consequence, we implemented an interface between APGEN and EUROPA. To name just a few of the functions of the interface:

- Updating EUROPA plan database in response to user changes made via the front end
- Updating the APGEN plan based on results from automated reasoning
- Offering suitable plan completion functions to the user, that still allow the user to target the plan completion
- Various advanced plan editing capabilities such as swapping the order of activities and automatically reestablishing flight rule enforcement

Updating the APGEN plan database in response to automated reasoning turned out to be a key challenge. Here below, we go into details about one aspect; the handling of activity placement in time.

Although it is not the subject of this paper, it is worth mentioning the constraint editor that is part of the MAPGEN system. The APGEN plan editing component is not suitable for constraint editing, which requires a very different interface than is offered by a timeline display. The tool allows users to specify temporal constraints on sets of activities, which then get imported into the MAPGEN tool. Having the constraint editor as a separate tool causes some difficulties, such as the users not getting feedback on the effect of constraints on the current plan, so the hope is that in the future the constraint editing capabilities can be incorporated more seamlessly into the MAPGEN system.

Constraint-based planning

The automated reasoning component of MAPGEN is based on an advanced constraint-based planning system called EUROPA. In constraint-based planning, activities and states are described by predicate statements that hold over temporal intervals. The interval timepoints and the predicate parameters are represented by variables connected by constraints. This approach supports a variety of complex planning constructs, including: activities with temporal durations, states that expire, exogenous events, complex constraints on parameters, temporal constraints linking activities and states, and subgoaling rules with conditions and disjunctions.

A constraint-based planning domain model defines a set of predicates, each of which has a set of parameters with possible values. The model also defines configuration constraints on predicates appearing in a plan. The notion of these configuration constraints is quite general and includes specific temporal and parametric constraints, as well as requirements for other activities and states in the plan. For example, the domain model may define a predicate **takePic** that indicates a picture being taken. The domain might then include rules specifying that during any **takePic** activity, the camera must be available, and that prior to **takePic**, the camera must be on and warmed up.

In constraint-based planning, a partial plan consists of a set of intervals, connected by constraints. The partial plan may be incomplete, in that rules are not satisfied and pending choices have not been made. The planning process then involves modifying a partial plan until it has been turned into a complete and valid plan. Traditional search-based methods accomplish this by trying different options for completing partial plans, and backtracking when constraints or rules are found to be violated. Constraint reasoning methods, such as propagation and consistency checks can be used to help out in that process. This planning approach also allows arbitrary changes to be made to a plan, thus supporting user changes, random exploration and a variety of other methods for building plans.

Preferred time placement

One of the capabilities offered by constraint-based planning is that complete valid plans can retain temporal flexibility. The MAPGEN tool utilizes this capability of constraint-based planning both to quickly respond to changes in the set of plan constraints, and to provide a "user preferred" instance of the flexible plan.

Flexible time means that instead of finding a single solution, the Planner preserves maximum temporal flexibility by maintaining a set of solutions that satisfy the constraints. This is represented internally as a Simple Temporal Network (STN). As a result of propagation in the STN, each activity acquires a refined time window for its start time.

One advantage of preserving a flexible set of solutions is that the Planner may adapt to additional constraints by exploiting the flexibility, rather than completely re-solving the problem. However, this has to be reconciled with APGEN, which expects to see a fixed time schedule. Also, many tools associated with APGEN, such as those that do calculations of resource usage, require a fixed schedule of activities. Apart from these pragmatic considerations, direct presentation of temporal flexibility to a plan GUI in a way that is not confusing poses significant problems: it is difficult to provide a visual representation of flexibility and temporal relations between activities in a way that does not obscure the display.

The approach we take is to present a single solution to the user in the APGEN GUI, while the Planner maintains the flexible set of solutions as a backup. This raises the issue of determining which fixed schedule to present to the user. The solution is to allow the human operator to modify the plan in a way that incorporates his or her implicit preferences.

In this application, there is a variety of constraints and preferences that arise from engineering restrictions and scientific need, many of which may not be recognised until specific circumstances arise in operation.

The explicit temporal constraints fall into three categories: *model* constraints, *daily* constraints, and *expedient* constraints. The model constraints encompass definitional constraints and some flight rules. For example, the decomposition of activities into sub-activities specifies temporal relations between the parent and its children. Some activities might be restricted to the day or the night. The daily constraints comprise ``on the fly" temporal relations between elements of scientific observations, depending on what scientific hypotheses are being investigated. For example, an image may be taken before using a specific instrument in some circumstances, but not in others. The expedient constraints are those imposed by the Europa planner to guarantee compliance with some higher level constraint that cannot be directly expressed in an STN. For example, a flight rule might specify that two activities are mutually exclusive (such as taking a picture while the rover is moving). This is really a disjunctive constraint, but the planner will satisfy it by placing the activities in some arbitrary order. This has important implications for the tweaking process: the operator may wish to reverse the arbitrary order selected by the planner.

There are also preferences that arise from varied sources. Some are based on engineering or scientific considerations such as desiring calibrations to be close to measurements, or wanting separate observations to occur in similar lighting conditions. Perhaps most are derived from the need to solve problems related to resources. In general, the tweaking process is driven by a desire to fit as much ``science" as possible into the plan, while steering it on a course that avoids running aground on competing resource limitations. The planner has a limited ability to automatically tweak a plan to try to resolve a battery energy shortfall, for example by increasing activity overlap (thus reducing cpu time), but most tweaks are performed by the human operator.

These considerations rule out formal modelling of most preferences and dictate the need for a process of informal tweaking by a human operator. The preferences are implicit in the modifications made during this period. However, the modifications interact with the hard constraints discussed above. The automated system must prevent these from being violated. Within this framework, a policy of minimal change provides a reasonable approach for respecting the implicit preferences.

A dramatic illustration of the need for the minimal change occurs when switching from a native APGEN mode, where users are free to modify activities at will, unimpeded by constraints, to the mode where constraints are enforced. To satisfy constraints, some activities must be moved, but arbitrary reorganization of the plan is undesirable.

Assume that a plan has been produced, and no preferences have yet been expressed to modify the solution. Then the initial solution presented is the earliest time one discussed earlier. During the subsequent tweaking phase, MAPGEN provides a GUI feature, called {\em constrained move}, that allows dragging an activity to a new location. When the mouse button is released, other activities are also moved to maintain the integrity of the constraints. For example, the moving activity may ``push" other activities ahead of it because of precedences established by the user or the planner.

This raises an issue with respect to the expedient constraints. Since these arise from disjunctive constraints that could be satisfied by different arbitrary choices, a mode is provided in which the expedient constraints are relaxed. This allows moved activities to pass over intervening activities that would otherwise be pushed ahead because of expedient constraints. When this relaxed mode is exited, there is a need to re-establish constraints in a way that minimizes the disturbance to the existing plan. A similar need arises when passing from the native APGEN mode to the constraint-maintenance mode. Also, the input files presented to MAPGEN are implicitly in the APGEN mode, and require a similar assimilation to the constraint-maintenance mode.

In this section, we describe the algorithm that is used to modify the solution presented to APGEN by the Europa system. In this interactive application, efficiency considerations seem to rule out the seeking of true. Instead, we have adopted a greedy algorithm that locally minimizes the amount of change from the existing positions of activities.

It is convenient to use a special set of unary singleton constraints to store the current positions of the start and end times of activities. Then the algorithm for updating after a constrained move can be outlined as follows:

```
1. Save all the current positions
in a temporary list.
2. Remove all the current position
constraints and repropagate.
3. For each saved position t of
timepoint x do:
 if t is within the STN bounds
 for x then:
  add a position constraint
  setting x to t
 else if t < the lower bound for
 x then:
  add a position constraint
  setting x to the lower bound
 else if t > the upper bound for
 x then:
  add a position constraint
  setting x to the upper bound
 Propagate the effect of the new
 constraint
```

We see that each step that reinstalls a position constraint tries to minimize the departure from the previous position while maintaining consistency. However, the greedy nature of the algorithm means that the order in which activities are considered may affect the outcome. For example, suppose that activity A is constrained to end before activity B starts. If an APGEN file is loaded where activity A is initially simultaneous with B, then one of A or B must be moved. Which of these occurs will depend upon the order in which A and B are considered for the position update in step 3.

The algorithm for updates when exiting a relaxed mode is similar, except that the relaxed constraints are reimposed after step 2. In the case of expedient constraints, the arbitrary planner choices for resolving the disjunctions are subject to change to reflect the saved positions of the timepoints as much as possible.

There are certain situations in which the user needs to ensure that a particular activity prevails in the update lottery. For example, after a constrained move, clearly the activity that is moved should be held to its new position. This is easily done by considering it first. (The new position is guaranteed to be within the STN bounds because a visual indication of these bounds is given during the move, and attempts to move the activity outside that range are ineffective.)

For more general situations, a {\em pinning} mechanism is provided that allows the user to lock specified activities at their current positions. This is achieved by applying additional constraints. There is a visual indication of which activities are pinned, and they can be unpinned on request. (Certain engineering activities, such as generally immutable communication windows, are pinned by default.)

Rover and model

The MER rovers incorporate the ``Athena" suite of instruments developed by the scientific investigators. This includes the Panoramic Camera (Pancam), Navigation Camera (Navcam), and Miniature Thermal Emission Spectrometer (MiniTes or MTES), which are associated with the mast (known as the Pancam Mast Assembly or PMA) that towers above the rovers. It also includes the microscopic imager (MI), Mossbauer spectrometer (MB), and Alpha Particle X-Ray Spectrometer (APXS), and Rock Abrasion Tool (RAT), which are integrated with the robot arm (known as the Instrument Deployment Device or IDD) on the underside of the rover. There are also Hazard Cameras (Hazcams) deployed around the rover.

The rovers are equipped with extensive communication facilities, including a High Gain Antenna (HGA) and Low Gain Antenna (LGA) for Direct-To-Earth (DTE) transmission and reception, as well as a UHF antenna for communicating with various Mars orbiting spaceflight, primarily the Mars Odyssey (ODY) and Mars Global Surveyor (MGS).

The rovers are of course mobile and ride on six wheels that can be moved independently and turned in various directions. The wheels can also be viewed as an additional scientific instrument in that they can be used to do ``trenching," where the rotation of a single wheel causes a hole to be dug in the ground, which can then be examined by the other instruments.

The actions of the rover are controlled and coordinated by an onboard computer (CPU). Some of the instruments, such as the APXS, have internal control hardware, and may require the CPU only for switching on and off.

In the daily planning cycle, the scientists request ``observations," which consist of coordinated activities involving the instruments. These have to be integrated with required engineering activities, such as communication sessions.

From a planning perspective, the main task is to schedule all the activities, including support activities where required, and to do so in a manner that does not violate transient or permanent restrictions on resource availability. An important class of restrictions involves mutual exclusion constraints on which activities can be performed simultaneously. These typically arise because of physical constraints on how the instruments can be used. For example, a RAT and an MI require different configurations of the robot arm, so they cannot be performed simultaneously.

Integration and fielding

From Kanna

Concluding Remarks

At the time of this writing, the MAPGEN tool is being used daily, for both rovers, as a critical part of the uplink process. The tool has performed very well and has without doubt increased the science return of the mission.

Observing the tool in operation, it is clear that one of the primary advantages is the active constraint enforcement. This allows the engineers to easily make changes to the plan, and yet be secure in that the changes will get propagated throughout the plan, so that new conflicts, constraint violations, or flight rule violations are not introduced. The ability to easily make changes in turn allows engineers to gradually build up plans, by adding more and more science observation activities into the plan. In the end, this makes it possible to fit in more activities than if decisions had to be made without feedback and conflicts had to be fixed manually.

The MAPGEN tool is designed to be adaptable to multiple missions. As a result, there is a great deal of future work to be done. A key goal is to make the tool more easily adaptable to new missions and to better integrate it into future mission control capabilities In terms of technical capabilities, there is work to be done in terms of resource reasoning, in particular when it comes to retaining flexibility while also satisfying resource constraints.

References

Mitchell Ai-Chang, John Bresina, Len Charest, Adam Chase, Jennifer Cheng-jung Hsu, Ari Jonsson, Bob Kanefsky, Paul Morris, Kanna Rajan, Jeffrey Yglesias, Brian G. Chafin, William C. Dias, and Pierre F. Maldague, "MAPGEN: Mixed-Initiative Planning and Scheduling for the Mars Exploration Rover Mission", IEEE Intelligent Systems, 2004.

R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. Artificial Intelligence, 49:61–95, May 1991.

J. Frank, and A. Jonsson, "Constraint-based attribute interval planning", Journal of Constraints, Constraints: An International Journal, Vol. 8, No. 4, 2003.

Ari K. Jonsson, Paul H. Morris, Nicola Muscettola, Kanna Rajan, and Benjamin D. Smith. Planning in interplanetary space: Theory and practice. In Artificial Intelligence Planning Systems, 2000.

L. Khatib, P. Morris, R. Morris, and B. Venable. Tractable pareto optimization of temporal preferences. In Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03), Acupulco, Mexico, 2003. Morgan Kaufmann, San Francisco, CA.

Maldague, Ko, Page, and Starbird. Apgen: A multi-mission semi-automated planning tool. In Proceedings of the 1st International Workshop on Planning and Scheduling for Space, Oxnard, California, 1997.

N. Muscettola, P.P. Nayak, B. Pell, and B.C. Williams. Remote agent: to boldly go where no AI system has gone before. Artificial Intelligence, 103(1-2):5–48, August 1998.