Activity Planning for the Mars Exploration Rovers

John L. Bresina, Ari K. Jónsson*, Paul H. Morris, Kanna Rajan

NASA Ames Research Center
Mail Stop 269-2
Moffett Field, CA 94035
{bresina,jonsson,pmorris,kanna}@email.arc.nasa.gov

Abstract

Operating the Mars Exploration Rovers is a challenging, time-pressured task. Each day, the operations team must generate a new plan describing the rover activities for the next day. These plans must abide by resource limitations, safety rules, and temporal constraints. The objective is to achieve as much science as possible, choosing from a set of observation requests that oversubscribe rover resources. In order to accomplish this objective, given the short amount of planning time available, the MAPGEN (Mixed-initiative Activity Plan GENerator) system was made a mission-critical part of the ground operations system.

MAPGEN is a mixed-initiative system that employs automated constraint-based planning, scheduling, and temporal reasoning to assist operations staff in generating the daily activity plans. This paper describes the adaptation of constraint-based planning and temporal reasoning to a mixed-initiative setting and the key technical solutions developed for the mission deployment of MAPGEN.

Introduction

In January 2004, NASA landed rovers on the surface of Mars at two widely separated sites. Their mission: to explore the geology of Mars, especially looking for evidence of past water. At the time of writing, signs of past water presence have been discovered at both sites, and although well past their design lifetime, both rovers are still healthy, and the mission is continuing.

Mars Exploration Rover (MER) operations are an interesting application from an automated planning perspective. While a number of ongoing planning research efforts are aimed at automating rover operations, operations are still done manually in actual missions; engineers develop command sequences on the ground and send them to the spacecraft, which executes them. MER, due to the complexity of the mission and the aggressive operations plan, challenged this traditional approach. So, although onboard decision-making capabilities were not considered, an opportunity did open up for ground-based automation, including automated planning. This led to the development of a mixed-initiative, constraint-based planning system called MAPGEN (Mixed-initiative Activity Plan GENerator), which has played a critical role in generating the daily activity plans throughout the nominal and extended mission.

Traditional AI Planning systems are given an initial state and a goal state and are expected to automatically produce a plan of actions that will achieve the goal state starting from the initial state. While this approach is suitable for fully automated operations, MER operations involved significant human participation in plan evaluation and construction. Consequently, the paradigm of *mixed-initiative planning* (Burstein and McDermott, 1996; Ferguson, et al., 1996; Myers, 1996; Veloso, 1996) was found to be more suitable.

Another characteristic of MER operations is the involvement of metric time and other numerical quantities. AI Planning has traditionally focused on symbolic systems, generally avoiding the use of numerical quantities. In recent years, the community has recognized that practical problems often involve quantitative reasoning (Smith, 2003). Among the approaches that address these issues is *constraint-based planning* (Smith, et al., 2000).

As yet, there are few practical examples of mixed-initiative constraint-based systems. Consequently, the

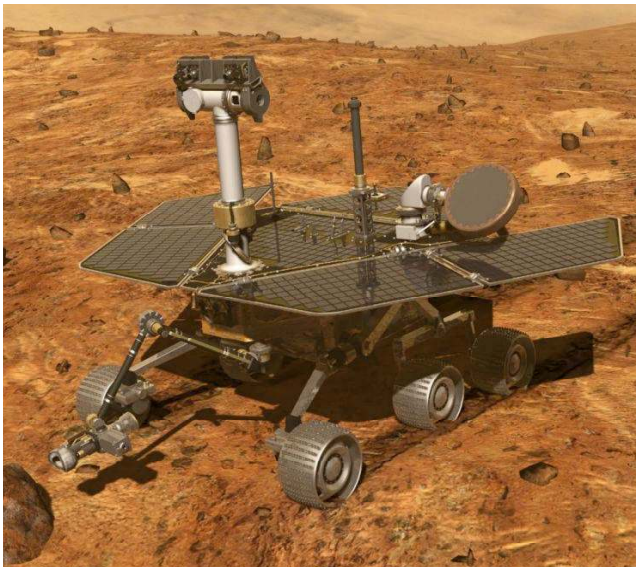

Figure 1: MER Rover

* Research Institute for Advanced Computer Science - USRA

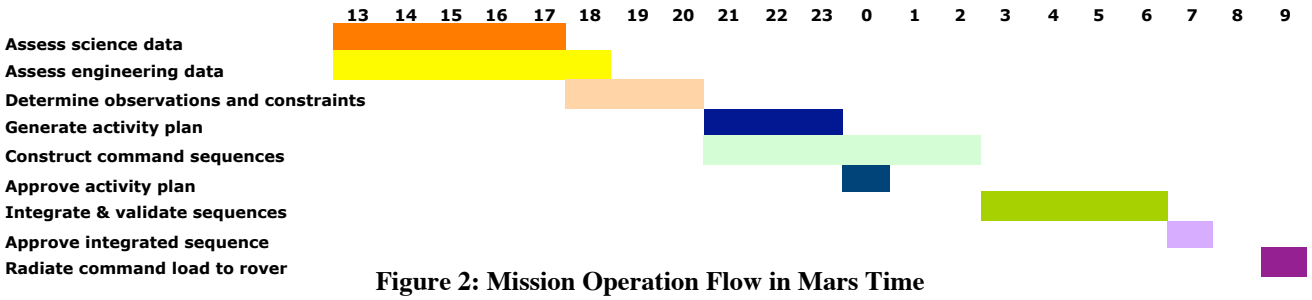

Figure 2: Mission Operation Flow in Mars Time

features and capabilities needed in such systems is an open research topic. In this paper, we report on the application of a mixed-initiative, constraint-based planning system to the task of tactical planning for the Mars Exploration Rover (MER) mission, focusing on the architecture and the novel reasoning techniques developed. We believe this work constitutes an important case study that will be valuable to the community in elucidating requirements for such systems and identifying future research directions for mixed-initiative planning.

MER Rovers

The MER rovers (see Figure1), Spirit and Opportunity, are solar-powered (with a storage battery) and incorporate a capable sensor and insturment payload. Panoramic cameras (Pancam), navigation cameras (Navcam), and a miniature thermal emissions spectrometer (MiniTES), are mounted on the mast that rises above the chassis. Hazard cameras (Hazcams) are mounted on the front and rear of the rover. A microscopic imager (MI), a Mössbauer spectrometer (MB), an alpha particle X-ray spectrometer (APXS), and a rock abrasion tool (RAT), are mounted on the robotic arm.

The rovers are equipped with extensive communication facilities, including a High Gain Antenna and Low Gain Antenna for Direct-To-Earth transmission and reception, as well as an UHF antenna for communicating with satellites orbiting Mars. Communication opportunities are determined by each rover's landing site, the Deep Space Network schedule, and orbital schedules for the satellites.

An onboard computer (CPU) governs the operation of subsystems and provides data handling, system state tracking, limited obstacle avoidance, and so forth. Because of its large power draw and the rover's limited energy supply, the computer is only turned on when it is needed.

Mission Operations

For this mission, the communication cycle was designed so that both rovers could be commanded every sol. (A sol is a Mars mean solar day, which is 24 hours, 39 minutes, and 35.2 seconds). The time for ground-based mission operations is severely limited by the desire to wait until up-to-date information is available but still finish in time to get the command load to the rover. During the nominal mission, this left at most 19.5 hours for ground operations.

In this process (shown in Figure 2), the engineering and science data from the previous sol are analyzed to determine the status of the rover and its surroundings. Based on this, and on a strategic longer-term plan, the scientists determine a set of scientific objectives for the next sol. Since detailed planning is not done at this stage, and only rough resource estimates are available, the scientists' requests will oversubscribe time and resource estimates, to ensure that the rover can be fully utilized in the final plan.

In the next step in the commanding process, the science observation requests are merged with engineering activities (e.g., communications, onboard data management, and subsystem checks) and a detailed plan and schedule of activities is constructed for the upcoming sol. The plan must obey all applicable *flight rules*, which specify how to safely operate the rover and its instrument suite and remain within specified resource limitations. It is in this step that MAPGEN is used.

Once approved, the activity plan is used as the basis to create sequences of low-level commands, which coordinate onboard execution. This sequence structure is then validated, packaged, and communicated to the rover. This completes the commanding cycle.

MER activity planning

Generating activity plans for a rover situated on a different planet is a unique problem. The remote location and expense of getting there require stringent rules for operations in order to minimize risk of vehicle impairment or loss. At the same time, these very factors drive a desire to maximize the amount of science done, as the expected mission lifetime is limited. Compared to spacecraft, rover surface operations are constrained by a much larger and more complex set of safety flight rules and policies.

Whereas the observation requests and the command sequences are constructed by teams of scientists and engineers, a single person, called the Tactical Activity Planner (TAP), has primary responsibility for producing the activity plan. Analyzing the previous sol's data and deciding what science observations to do next sol can take a long time; likewise, constructing the sequence structure is very time-consuming. This leaves little time for activity planning and puts the TAP under pressure to, on the one hand, construct a high-quality plan that maximizes science return and, on the other hand, not hold up the sequencing team. Figure 2 shows the limited time given to the activity planning process within a commanding cycle.

As a planning problem, the MER activity plan generation process has the following key characteristics:

- Optimization: The objective is to achieve the maximum number of highest-priority science goals, selected from a request set that oversubscribes rover resources.
- Temporal Constraints: Specified activities are constrained in continuous time, both absolutely and relatively. Furthermore, flight rules impose mutual exclusion requirements.
- Resources: Complex continuous quantity resources, such as energy, play a major role.
- Hierarchical Activities: High-level activities are decomposed into specific temporal configurations of lower-level activities.
- Size: The number of low-level activities in an activity plan is typically in the high hundreds to low thousands.
- Incomplete problem definition: Preferences and other subjective solution criteria cannot be specified up front.

Outline

The rest of this paper describes how the MAPGEN system tackles this planning problem in a mixed-initiative fashion. The main focus is on the overall system architecture and key methods and techniques that were developed for MAPGEN. We start by describing the overall characteristics of the MAPGEN system. Next, we lay out the system architecture. Subsequent sections focus on the key technical aspects of time handling and planning search methods. We conclude with remarks on results, lessons learned, future research directions, and related work.

MAPGEN: System Characteristics

Traditionally, spacecraft operations planning is done manually; software tools are primarily for simulating plan executions and identifying flight rule violations. The time criticality and complexity of MER operations, combined with advances in planning and scheduling technology, provided an opportunity for deploying automated planning and scheduling techniques to the Mars rover activity plan generation problem.

Mixed-initiative planning

In traditional automatic planning, the operator loads in the goals and initial conditions, pushes a button, and waits for a complete plan. For generating MER activity plans, a fully automated approach was deemed unacceptable, due to the need to bring human expertise in mission planning and science operations to bear. Consequently, we adopted a mixed-initiative approach for this application.

There were many aspects to the need for human involvement. Mission operations rely on a number of checkpoints and acceptance gates to ensure safety. For activity plans, the critical gate was the activity plan approval meeting where the fully constructed plan would be presented, critiqued, and, hopefully, accepted, possibly with minor modifications. As a result, the TAPs had to be able to understand and defend the validity of the plan; however, initial user tests indicated that a plan constructed automatically in its entirety was too difficult to analyze by the human operator, especially given the inherent time pressures. The TAPs, therefore, prefer to incrementally construct a plan in small, understandable chunks.

Another concern was the infeasibility of formally encoding and effectively utilizing all the knowledge that characterizes plan quality. One aspect of plan quality involves a rich set of science preferences, including everything from preferences on absolute and relative scheduling of activities to preferences on which combinations of science observation cuts and changes are least painful in the face of strict resource limitations. A second, and more complex, aspect of quality is concerned with global characteristics of a plan, such as acceptable profiles of resource usage, and the estimated complexity of turning a plan into a command sequence structure.

The role of mixed-initiative planning in MAPGEN is very much in the spirit of the original notion of such planning (Burstein and McDermott, 1996); the purpose is to support collaboration between a human user and an automated system to build a high quality activity plan. However, it is worth noting that, unlike some variations of mixed-initiative planning, MAPGEN does not actively solicit user assistance during planning. The primary role of the operator is to direct and focus the plan construction process and to provide qualitative evaluation of plans. The system makes automated planning capabilities available to the user and performs potentially tedious tasks, such as expanding activities and maintaining constraints. The intended interaction between user and system is that the system handles expansion and constraint enforcement constantly in the background, while automated plan construction is user invoked.

Key Features

As an integral part of a large mission operations system, MAPGEN's capabilities evolved over time with the rest of the ground data system. The current user features are the end result of a journey through the design space, guided by feedback from the users in the course of many tests and impacted by frequent changes in the overall mission operations system. We can summarize the primary features as follows:

Plan editing: Activities and constraints can be added, deleted and modified, using direct manipulation (e.g., dragging activity to a new time), form-based editing, and menu item selection.

Plan completion: Selected subsets of activities can be completed, in the sense that all subgoals are achieved and all necessary support activities are added to the plan.

Active constraints: During plan editing, constraints and rules are *actively* enforced. Thus, when one activity is moved or modified, other activities are modified as needed to ensure the constraints are still satisfied.

In the course of developing these capabilities, it became clear that the following additional features were needed.

Hopper: In order for the user to incrementally plan, a staging area, called the hopper, is needed for activities that have not yet been made part of the plan.

Goal rejection: If a planning request cannot be completed, MAPGEN can reject lower-priority activities to make room for higher-priority activities in the plan. There is also a timeout mechanism to terminate any excessively long search, after which incompletely planned activities are rejected. Rejected activities are placed in the hopper and, thus, continue to be available for planning.

Constrained-move: The user has the capability of moving activities to preferred locations using a drag-and-drop mechanism. The system supports the active maintenance of constraints by automatically adjusting other activities as needed to ensure plan validity. There are two variants, depending on whether mutual exclusion orderings can be revised or not. In both cases, visual feedback about the possible range of movement is provided and the user is prevented from moving outside this range.

Minimal perturbation: Actively enforcing constraints gives rise to choices, as the impact of modifications can be handled in different ways. For example, following a constrained-move, there are many ways to move other activities that would ensure that the resulting plan satisfies all temporal constraints. For continuity and to facilitate user understanding, the system attempts to minimize change by keeping activities as close as it can to their previous positions, while satisfying the constraints.

MAPGEN Architecture

The MAPGEN system architecture, which is shown in Figure 3, combines pre-existing (APGEN and EUROPA), and MAPGEN-specific software components.

One of the requirements for infusing this technology into the mission was the use of an existing interactive plan editor from JPL, called APGEN (Maldague, et al., 1998), as the front end of MAPGEN. The core of the plan representation and reasoning capabilities in MAPGEN is a constraint-based planning framework called EUROPA (Extendable Uniform Remote Operations Planning Architecture), developed at NASA Ames Research Center (Jónsson, et al., 1999; Frank and Jónsson, 2003).

The new functionality in the MAPGEN system involves the interface between these two subsystems, support for extensions to the APGEN graphical user interface to provide the mixed-initiative capabilities, and more sophisticated plan search mechanisms that support goal rejection, priorities, and timeouts. The APGEN and EUROPA databases, which remain separate, are kept synchronized; changes may be initiated by either database.

Finally, we considered it expedient to develop an external tool, called the Constraint Editor, to enter and edit daily science constraints, since this is not conveniently supported by the current APGEN graphical user interface.

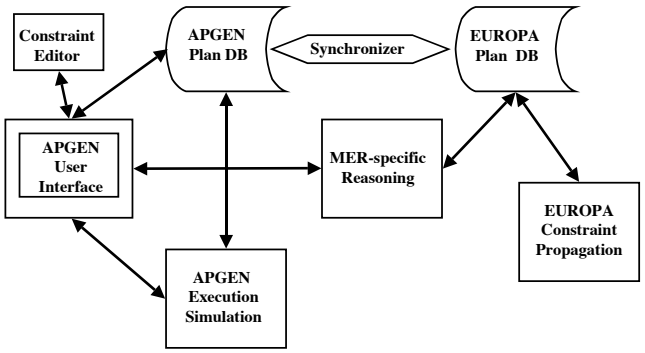

Figure 3: MAPGEN Architecture

EUROPA: Constraint-based planning framework

In constraint-based planning (Frank and Jónsson, 2003), actions and states are described as holding over intervals of time. Each state is defined by a predicate and a set of parameters, as in traditional planning paradigms. Actions, which are durative, are also represented by parameterized predicates. The temporal extent of an action or state is specified in terms of start and end times. For example, specifying that the panorama camera heater needs to be on for 25 minutes, starting at 8:00, could be written as:

```
holds(8:00,8:25,pan_cam_htr(on,0:25))
```

However, in constraint-based plans, each time and parameter value is represented by variables, connected by constraints. Consequently, the statement would be:

```
holds(s,e,pan_cam_htr(state,dur))
s=8:00, e=8:25, state=on, dur=0:25
```

This approach allows specifying, for example, that the heater must be on for 30 minutes, can happen within a certain time range, and must happen before, by at most 5 minutes, a given use of the camera:

```
holds(s1,e1,pan_cam_htr(state,dur1))
s1 ∈ [8:00,8:30], state=on, dur1=0:30
e1=s1+dur1
holds(s2,e2,pan_cam(tgt,#pics,dur2))
s2 ∈ [9:20,9:40], tgt=rock, #pics=8
e2=s2+dur2
s2-e1 ∈ [0:00,0:05]
```

Constraint reasoning plays a major role in the constraint-based planning paradigm. Any partial plan, which is a set of activities connected by constraints, gives rise to a constraint network. Constraint-based inference can provide additional information about plans, reduce the number of choices to make, and identify dead-end plans early. Achieving arc consistency is one commonly used example of applicable constraint reasoning methods.

Typically, the temporal variables and associated constraints give rise to a simple temporal network (STN), or can be reduced to one by making decisions that enforce the mutual exclusion constraints. For STNs, it is possible to make the network arc consistent and to determine consistency in low-order polynomial time, using the Bellman-Ford algorithm (Dechter, Meiri, and Pearl, 1991; Cormen, Leiserson, and Rivest, 1990).

In constraint-based planning, domain rules are specified in terms of activity/state patterns and constraint schemas. A given constraint schema is applied to any instance matching the associated pattern. For example, a rule might specify that a camera heater is needed between 0 and 5 minutes prior to any camera use that occurs before noon. In other words, this rules states that for any occurrence of:

`holds(s2,e2,pan_cam(tgt,#pics,dur2))`

where `s2` is necessarily less than 12:00, there must exist a

`holds(s1,e1,pan_cam_htr(state,dur1))`

such that:

`s2-e1 ∈ [0:00,0:05]`

Constraint schemas can also apply to single interval statements, e.g., for any occurrence

`holds(s1,e1,pan_cam_htr(state,dur1))`

we ensure the heater is never on for more than two hours:

`dur1 ∈ [0:00,2:00]`

The EUROPA framework performs sound constraint reasoning and provides a mechanism for firing applicable domain rules. Search methods and other techniques for manipulating partial plans then build on this framework.

In constraint-based planning, explicit temporal constraints fall into three categories: *model* constraints, *problem-specific* constraints, and *expedient* constraints. The model constraints encompass definitional constraints and mutual-exclusion flight rules. In MER, for example, the expansion of activities into sub-activities gives rise to temporal relations between the parent and its children.

The problem-specific constraints comprise relations between specific activities in a planning problem instance. In MER, these constraints, often called "daily constraints", related elements of scientific observations in order to capture the scientists' intent. As an example, several measurements of atmospheric opacity may be required to be at least 30 minutes apart. These constraints are entered using the Constraint Editor tool, described below.

The expedient constraints are those resulting from arbitrary decisions made to guarantee compliance with higher-level constraints that cannot be directly expressed in an STN. For example, a flight rule might specify that two activities are mutually exclusive (such as moving the arm while the rover is moving). This is really a disjunctive constraint, but satisfying it will involve placing the activities in some arbitrary order. Expedient constraints are typically added during search in automated planning.

APGEN

APGEN (Activity Plan GENerator) is an institutional tool at JPL and has been used in a number of spacecraft missions. It has a large number of features, but the core capabilities can be summarized with three components:

- **Activity plan database**: A set of activities, each at a specific time. This database has no notion of constraints between activities, but does support context-free activity expansion.
- **Resource calculations**: Methods for calculating, using forward simulation, resource states that range from discrete states to complex numerical resources.
- **Graphical user interface**: An interface for viewing and editing plans and activities.

To deploy APGEN for a particular mission, the mission-specific information is stored in an adaptation, which can be viewed as a procedural domain model. It defines a set of activity and state types and then defines a way to calculate resource states from a given set of activities. In addition, it defines a set of "constraints" on legal combinations of resources. The constraints and resource calculations are only used for passively identifying problems with a plan; APGEN does not have the capability to reason with this information in order to help fix the identified problems.

Synchronization

The plan databases in APGEN and EUROPA remain separate. Consequently, a component is needed for translating information between the two plan databases. The translation is straightforward for the most part. However, it is worth noting that a constraint-based plan invariably has temporal flexibility, in the sense that start and end times need not necessarily be grounded. At the same time, APGEN can only represent grounded times; i.e., each activity has a single grounded start and end time. The challenging issues involved in mapping between grounded and flexible temporal representations are addressed in the "Time Handling" section.

Automated planning and reasoning

The functionality provided by the MER-specific planning and reasoning component can be split into two categories. One consists of extensions of existing functionality in APGEN. An example of this is the extension of the activity parameter editing capability to reject any changes that violate constraints, with a warning to the user. The other category consists of new functionality, typically accessible via the user interface. These include automatically planning a selected subset of the activities in the hopper. Aspects of the functionality provided are discussed further below in the "Planning Methods" section.

Constraint Editor

The APGEN plan-editing interface has no notion of variables and constraints in the traditional AI sense. This raised the issue of how to get the daily constraints into the reasoning component of MAPGEN. These daily constraints were needed to coordinate the activities in scientific observations, and these could vary in unforeseen ways. For example, it might be specified that two specific measurements should be taken within 10 minutes of each other. This required an ability to enter and modify temporal constraints dynamically.

To resolve this, an external, temporal-constraint editing tool, called the Constraint Editor, was developed as an augmentation to the MAPGEN interface. In this tool, users can view activities and existing temporal constraints, and then add, delete, or edit constraints.

Two issues arose with respect to the Constraint Editor. First, the sheer number of constraints posed problems for manual entry. However, by grouping activities into sets and then specifying relations between the sets, it was possible to enter large numbers of similar constraints simultaneously. Second, the TAP would occasionally enter constraints that were inconsistent. Often it was difficult for the TAP to know what caused this inconsistency. Thus, when an inconsistency was detected, a description of a minimal nogood, which corresponds to a negative cycle in the temporal network, was extracted to present to the user and allow the user to delete one or more of the inconsistent constraints.

Time Handling

As mentioned earlier, MAPGEN synchronizes a front-end database that uses a fixed-time schedule with a back-end planner that maintains flexible plans. This raises the question of which instantiation of the flexible plan to present to the front-end. An obvious candidate is the earliest-time solution; however, this would miss out on an opportunity to *exploit* the flexibility. In MAPGEN we take advantage of this feature to provide a novel method for incorporating temporal preferences in the plan.

Temporal Constraints and Preferences

Constraints and preferences for temporal placement arise from a variety of sources. For example, flight rules may require heaters or the CPU to be on in support of activities. Additionally, directly specified constraints may require that activities be done in a particular order and at a particular time in order to satisfy the science intent. Other external factors also impact scheduling; chief among those are energy limitations and communication opportunities.

Activities use different amounts of energy, depending on when they occur, due to heaters being needed only during cold times of the day. In addition, some schedules will cause solar power to be discarded (shunted) to avoid overcharging the battery. Modifying when activities are scheduled could make use of this wasted energy.

Activity scheduling also interacts with communication opportunities. Suppose that one of the activities in the plan is a drive to approach an interesting rock, with the intent of deploying arm instruments the following sol. This requires operators to have a good picture of the workspace around the arm. Thus, an activity that takes a picture of the workspace after the drive must be done prior to a suitable communication opportunity, so that the picture will be available when needed to plan the arm movements.

Some of these requirements can be captured as hard constraints that are enforced by the planner. Others have a more ephemeral or malleable nature, and can best be described as preferences. The primary onus is on the operator to take these into account. However, MAPGEN plays a supporting role in facilitating their entry and maintenance, as we will see.

Flexible Plans

In constraint-based planning, partial plans have an underlying simple temporal constraint network (Dechter, Meiri, and Pearl, 1991). The consistency of STNs can be determined by checking for arc consistency. Furthermore, each value in an arc-consistent temporal variable domain appears in at least one legal solution for the temporal network. The set of such values defines a temporal interval that can be represented by its bounds.

Consider a plan where all decisions have been made, except for grounding temporal variables appearing only in simple temporal constraints. Finding a fixed solution is then an easy matter of choosing a value for any variable within its legal bounds, re-enforcing arc consistency, choosing a value for another variable, and so on.

It is not necessary to immediately ground the variables; plans with temporal variables left ungrounded are called *flexible plans*. In MAPGEN, we utilize the fact that the underlying plans are flexible to provide scope for expressing and maintaining temporal preferences.

Reference Schedule

The MAPGEN user interface can only display a single instantiation of a flexible plan. This raises the issue of which instantiation should be chosen. The method we developed is based on minimizing the departure from a *reference* schedule, which need not necessarily be consistent. The reference schedule provides a general method for expressing unary temporal preferences. Its primary use in MAPGEN is to support a minimum perturbation framework where changes to the previous plan are minimized when a planner-supported operation is invoked. This is accomplished by continually updating the reference schedule to reflect the evolving plan. This means that changes made by the operator to reflect preferences or eliminate problems are respected and maintained unless they violate constraints or are revised by the operator. The reference schedule also plays a role in the heuristic search, as discussed in a later section.

Consider the example in Figure 4 where the user has just added an ordering constraint between activities T1 and T2 that overlap in the reference schedule. The constraint is inconsistent with respect to the preferred positions, but (as indicated by the bounds) the constraint addition does not make the underlying flexible network inconsistent. The objective is then to find a new instantiation that is consistent with the constraint while still being close to the preferred positions.

Efficiently finding a preferred instantiation is a subject of ongoing research (Morris, et al., 2004), but an approximate technique was found to work well in MAPGEN. This technique employs a greedy heuristic, where variables are instantiated in some order; for each variable, the legal value that is closest to the preferred placement value is chosen. The algorithm, described in more detail in (Bresina, et al., 2003), is outlined here:

1. Remove all the current position constraints and re-propagate.
2. For each timepoint x with preferred position t 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 (lb) for x
     then add a position constraint setting x to lb
   else if t > the upper bound (ub) for x
     then add a position constraint setting x to ub
   Propagate the effect of the new constraint
3. Update the preferred positions to the current ones.

Figure 4 shows how this applies to the example. Not only is this guaranteed to yield a consistent instantiation whenever the underlying flexible network is consistent, but the result was found to be very intuitive to users.

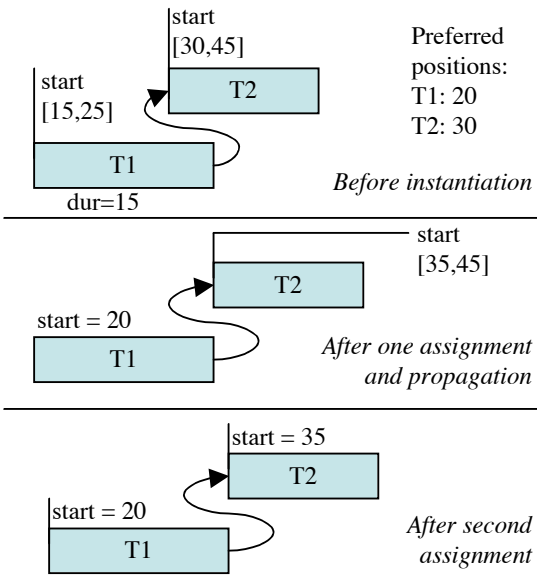

Figure 4: Preferred time placement example

Constrained moves

One very commonly used plan modification is to move an activity to a new time. As long as the activity is moved only within the flexibility range defined by the domain in the underlying arc-consistent flexible plan, the result is necessarily another consistent instantiation. This observation gave rise to the notion of a *constrained move*.

During a constrained move, the system actively restricts the movements of an activity to stay within the permitted range. Then, once the user places the activity, the minimal perturbation update is applied to all affected activities, yielding a new valid plan instance.

Note, however, that the consistency enforcement takes into account *all* the constraints that determine the flexible plan. This includes expedient constraints resulting from decisions about how to order mutually exclusive activities. Since these decisions are maintained, the ordinary constrained move has the effect of "pushing" the excluded activities ahead of it. However, sometimes the TAP wants to *reorder* mutually excluded activities. To support this, we provided a variation, called a *super-move*, that temporarily relaxes expedient constraints until the move is completed. This amounts to a simple form of re-planning; this topic is further discussed in the next section.

Planning Methods

In MAPGEN, the use of planning, in the sense of using search to find complete and valid plans, is in some ways different from more traditional applications. This stems from both the mixed-initiative nature of the tool and the specifics of the domain.

In the MER domain, the notion of traditional subgoals was almost entirely absent. Instead, activities had fixed, HTN-like expansions, defined in the mission activity dictionary. Context-dependent activities, such as those needed to establish preconditions, were rare, and were typically added manually. The chief exceptions were the management of the CPU, which was handled by the automated planning system, and heater activities, which the user could request the system to automatically create, based on the current thermal policies.

Another key factor in the design of MAPGEN planning methods was that the set of science observation requests oversubscribes available resources and, thus, each observation request had a given priority that had to be taken into account.

Finally, the planning methods had to be accessible to the user and had to ensure timely responses to user requests. The user-accessibility requirement led to a number of different planning methods, and the response-time requirements led to an anytime-like search technique.

Core planning algorithm

The main decisions made by the MAPGEN planning algorithm are whether to include an activity in the plan, along with all its sub-activities, and for each included activity and sub-activity, how to order it in relation to mutually exclusive activities. For each ordering decision, the planner inserts an expedient temporal constraint to enforce it. The planner maintains a stack of all decisions made, backtracking if necessary in order to achieve a consistent plan.

There are several complicating factors in the search. As noted above, the problem is oversubscribed and science observations are prioritized. A simplistic approach is to use the standard backtracking search that prefers to work on higher priority activities and prefers not to reject activities. This will easily handle direct conflicts between higher and lower priority activities, but becomes intractable when search is required to prove that there is a conflict between a lower priority activity and a collection of higher

priority activities. To avoid the potentially exponential search, the basic backtracking method was modified to provide an incomplete variation of dynamic backtracking (Ginsberg, 1993). However, instead of collecting and combining nogoods, the search "charges" backtracks to associated top-level activities. When a top-level activity exceeds its allotment, which varies depending on its priority, it is deemed "troublesome" and rejected. Due to the possible interleaving of decisions stemming from different top-level activities, this requires a form of dependency-directed blame assignment and cleanup.

Response time guarantees

The non-systematic rejection of "troublesome" activities was essential for balancing the amount of search effort done and the quality of the solution. However, the time taken to complete the search can still vary greatly, both depending on the size of the problem and on the level of interaction between activities.

To provide a guaranteed response time, the search includes a global timeout mechanism. The key element in this mechanism is the cleanup of activities that are only partially planned when the search is halted. This again requires dependency-directed methods.

The timeout rejection methods were infrequently triggered in practice, because the MER users typically planned incrementally in small chunks.

Heuristic search guidance

As noted above, the search was biased to work on high priority activities before low priority ones. Virtually no other heuristics were used for determining the order in which open decisions were tackled.

When it came to making activity placement choices, i.e., expedient ordering decisions, the heuristic guidance used was based on minimizing deviation from the reference schedule. The motivation behind this was twofold. One was that it would be intuitive to the user, as this approach would attempt to preserve the temporal placement of activities. The other motivation was that it would allow users to "sketch out" a plan in the hopper and then ask the system to complete the plan.

An important issue of *foresight* arose with respect to incremental goal achievement. If earlier goals are achieved completely without regard to the needs of later goals, then arbitrary choices may be made that preclude the later goals. For example, two early goals A and B may have no direct ordering, but a later goal C may need to occur after A and before B. Without foresight, the planner may choose to order B before A and, thus, prevent the later achievement of C. This can be remedied by enforcing constraints arising from items in the hopper. In our example, this would ensure that A comes before B. However, this can have the undesirable consequence of eliminating valid plans from consideration. For example, if other factors prevent A from coming before B, then the constraint involving C will eliminate the option of placing B before A and rejecting C. The approach we took was to initially obtain a consistent schedule for all activities, including those in the hopper, based only on the daily constraints. The result was used as the initial reference schedule. Thus, the solution of early goals was biased by the constraints on activities still in the hopper, but not dictated by them.

Variations on a theme

As noted above, the automated planning capability was presented to the user in terms of a few different options.

One variation was to plan everything, thus leaving it entirely up to the automated search to find a plan that achieved as much science as possible. This functionality is most like what traditional automated planning methods do. This capability functioned well and yielded near-optimal plans in terms of the number of science observations in the plan. However, the plans tended not to have an intuitive structure and, therefore, did not allow the TAP to explain the plan structure during the approval meeting. Additionally, they were often sub-optimal with respect to preferences and other solution quality criteria that were not encoded in the domain model or the priorities. Consequently, it was rarely used.

Instead, the users often applied a variation where the user could select a set of observation requests not in the plan and request that these be inserted into the partial plan already in place, such that all rules were satisfied. While repeated application of this led to a result similar to the full planning variation, users found this more intuitive, in part because it allowed them to fine-tune and understand the incremental plans as they were built. Furthermore, this made it possible for the users to have a complete plan ready at just about any time.

Another variation, applicable only to individual activities, allowed the user to select an activity in the hopper and then choose an approximate temporal placement for it in the plan. The planning algorithm would then treat the user-chosen time as heuristic guidance and search for a plan where the selected activity was as close to the desired time as possible.

While users considered the super moves as plan editing capabilities, a super move is, in fact, a small replanning operation. During a super move, the activity being dragged, along with all its sub-activities and subgoals, is removed from the plan. Then, when the user drops the activity at the end of the move, to place it, the planner is asked to find a plan where the moved activity is placed as close as possible to the chosen time. In the event that the placement fails, the plan is left unchanged.

Concluding Remarks

The MAPGEN deployment constitutes a major advance in ground support tools for NASA missions. It has demonstrated that automated reasoning techniques can be combined with human knowledge and insight in a way that greatly benefits mission operations. MAPGEN has been

used throughout the nominal and extended Mars Exploration Rover mission as an essential part of the ground operations process. Furthermore, it has had a significant impact on the science return from the mission. Subjective estimates, from both science investigators and mission managers, indicate a 20-40% increase in science return over the manual approach used in Mars Pathfinder.

The system has also changed the way TAPs approach the planning process. With the added efficiency resulting from the tool, they have had enough time to explore alternative "what-if" scenarios and to perform solution fine-tuning, thus achieving a higher-quality plan. Moreover, they are more willing to incorporate late-breaking information, given their new confidence in being able to rebuild the plan within the available time. This became critical once the mission was no longer operated on Mars time, because planning often had to start before necessary information from the rovers was fully processed. In fact, there were sols when the entire plan had to drastically change at the last minute due to revised information, and without MAPGEN, the TAP would not have had time to generate a new plan.

Discussions with mission operators suggest that MAPGEN has raised the bar on what will be expected from ground tools in future missions.

Planning technology lessons

Infusing technology into an actual mission requires a certain degree of conservatism. Much of the novelty in MAPGEN lies in the architecture of the system rather than in developing radically new methods and techniques. However, this work leads to insights into what kinds of techniques and capabilities are needed in the future.

It became clear that a mixed-initiative system was the right choice for reasons beyond those that led to its adoption. The human component provided for adaptability and flexibility in the use of the tool that allowed us to cope with evolving and changing requirements. Moreover, the ground operations process is not perfect, and the mixed-initiative framework provided scope for workarounds to deal with shortcomings, perhaps temporary, in other areas.

Another lesson we learned is that planning in the traditional sense is not necessarily the most prized feature from the user's perspective. At least as important (perhaps more so) are features like unplanning, replanning, active constraint enforcement, and constrained moves. Moreover, some relatively mundane planning, such as determining when the CPU must be on, may be more valuable to the user than complex, context-sensitive goal achievement.

One thing that is clear is a need for the automated reasoning component to provide better explanations of its behavior, particularly explanations of why the planner could not achieve something, such as inserting an activity in the plan at a particular time, or moving an activity beyond the enforced limit. Such a facility would have greatly helped during training, in addition to increasing the TAPs effectiveness during operations. The system did have a form of explanation of inconsistency by presenting a minimal nogood. While the TAPs found it to be very useful when editing constraints, only the developers used the facility in the context of constructing and modifying plans, for the purpose of debugging the system. The reason is that in this context, the explanation typically involved complex chains of activities and constraints that could not easily be grasped.

Another need is for a way to express temporal preferences. MAPGEN did have a limited capability in this regard in terms of the reference schedule. The operator could also establish more complex preferences by an iterative process of relaxing or tightening constraints, but this was time-consuming and one could envisage an automatic facility to achieve this.

An attempt to automatically resolve certain resource violations had only limited success. One difficulty was obtaining sufficient information about resource usage from the complex legacy resource computation modules. We also found that the more important resources, such as battery energy, had complex properties such as continuous variation, saturation effects, and thermal dependencies. As a result we could only check consistency in the context of a fixed schedule. The current state of the art in resource reasoning for flexible plans, e.g., (Muscettola, 2002), is limited to handling addable resource transactions occurring at discrete time-points. Until this can be extended to handle the complexity seen in MER, a more fruitful approach may be to provide the user with a suite of heuristic techniques that could be judiciously employed to improve the resource usage profile, such as a technique for adjusting the schedule to make better use of shunted energy.

It would also be useful for the system to be able to answer trade-off queries such as the following examples:

- What needs to be unplanned (in priority order) to enable additional time for arm instrument use, or to allow for driving further?
- For a given panorama that does not fit as a whole, which parts of it can be fit into the current plan?
- In order to fit in another imaging activity, what needs to be unplanned or shortened?

Finally, the work on MAPGEN has underlined the fact that when it comes to building model-based systems for complex domains, the issues of knowledge acquisition and domain knowledge validation remain open. Model-based systems represent and reason about knowledge that is traditionally kept by mission operators, making in-depth validation of models and reasoning methods a key requirement for mission-critical applications.

Related Work

Besides the mixed-initiative literature cited earlier, there are few comparable mission-planning systems that employ general planning techniques. In Deviser (Vere, 1983), planning was done at a very high level of abstraction. Nonetheless, even with only a few hundred activities in the plan for Voyager's encounter with Uranus, the planner took 40 CPU hours on a LISP machine, making real-time planning infeasible. In 1999, PlanIT-2 (Eggemeyer et al.

1998) was used for sequencing activities on the Mars Pathfinder mission. AI search was deemed unnecessary, and the system provided only basic constraint maintenance (Grenander, 2004). Other work related to planning and scheduling in the space domain includes the seminal Remote Agent system (Muscettola, et al., 1998), and the recent Autonomous Science Experiment (Cichy, et al., 2004). However, these systems addressed onboard rather than ground-based planning.

Acknowledgements:

The authors would like to acknowledge the entire MAPGEN team, which includes: Mitch Ai-Chang, Len Charest, Brian Chafin, Adam Chase, Kim Farrell, Jennifer Hsu, Bob Kanefsky (who implemented the constraint editor), Adans Ko, Pierre Maldague, Richard Springer, and Jeffrey Yglesias. The authors would also like to thank the MER TAP's, Bill Dias, Jason Fox, Jeff Norris and Tom Starbird, for their valuable feedback. Others instrumental in making MAPGEN a success are Jim Erickson, MER Project and Mission Manager at JPL, and Nicola Muscettola and Dave Korsmeyer at NASA Ames.

References

Bresina, J., Jónsson, A., Morris, P., and Rajan, K.. Constraint Maintenance with Preferences and Underlying Flexible Solution. *CP-2003 Workshop on Change and Uncertainty*, Kinsale, Ireland, 2003.

Burstein, M., and McDermott, D., Issues in the development of human-computer mixed-initiative planning. In *Cognitive Technology*, B. Gorayska, and J. L. Mey, editors, pages 285-303. Elsevier, 1996.

Cichy, B., Chien, S., Schaffer, D., Tran, G., Rabideau, G., Sherwood, R., Validating the Autonomous EO-1 Science Agent, International Workshop on Planning and Scheduling for Space (IWPSS 2004). Darmstadt, Germany, 2004

Cormen, T., Leiserson, C., and Rivest, R.. *Introduction to Algorithms*. MIT press, Cambridge, MA, 1990.

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

Eggemeyer, W., Grenander, S., Peters, S., and Amador, A. Long Term Evolution of a Planning and Scheduling Capability for Real Planetary Applications. *First International NASA Workshop on Planning and Scheduling*, Oxnard, CA, 1998.

Ferguson, G., Allen, J., and Miller, B. TRAINS-95: Toward a Mixed Initiative Planning Assistant. *Third International Conference on Artificial Intelligence Planning Systems*, 70-77. 1996.

Frank, J., and Jonsson, A., Constraint-Based Interval and Attribute Planning, *Journal of Constraints* Special Issue on Constraints and Planning, 2003.

Ginsberg, M.L, Dynamic Backtracking, Journal of Artificial Intelligence Research (JAIR) 1 : 25-46, 1993.

Grenander, S., Personal Communication, 2004.

Jónsson, A. K.; Morris, P. H.; Muscettola, N.; and Rajan, K. 1999. Next generation Remote Agent planner. In Proceedings of the Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS99).

Jónsson, A., Morris, P., Muscettola, N., Rajan, K., and Smith, B., Planning in interplanetary space: Theory and practice. *Fifth International Conference on AI Planning Systems*, Breckenridge, CO, 2000.

Maldague, P., Ko, A., Page, D., and Starbird, T., APGEN: A multi-mission semi-automated planning tool. *First International NASA Workshop on Planning and Scheduling*, Oxnard, CA, 1998.

Morris, P., Morris, R., Khatib, Ramakrishnan, and Bachmann. Strategies for Global Optimization of Temporal Preferences. *Tenth International Conference on Principles and Practices of Constraint Programming* (CP-2004), Toronto, Canada, 2004.

Muscettola, N., Computing the Envelope for Stepwise-Constant Resource Allocations. *Eighth International Conf. on Principles and Practices of Constraint Programming*, Ithaca, NY, 2002.

Muscettola, N., Nayak, P., Pell, B., Williams, B., Remote Agent: To Boldly Go Where No AI System Has Gone Before. *Artificial Intelligence*, 103(1/2), August 1998.

Myers, K.L. Advisable Planning Systems. In *Advanced Planning Technology: Technological Achievements of the ARPA/Rome Laboratory Planning Initiative*, ed. A. Tate, 206-209. AAAI Press, 1996.

Smith, D.E., Frank, J., and Jónsson, A.K., Bridging the Gap between Planning and Scheduling, Knowledge Engineering Review, 15(1), 2000.

Smith, D.E. (ed), Journal of Artificial Intelligence Research, Vol 20, Special Issue on the 3rd International Planning Competition), 2003.

Veloso, M.M. Toward Mixed-Initiative Rationale Supported Planning. In *Advanced Planning Technology: Technological Achievements of the ARPA/Rome Laboratory Planning Initiative*, ed. A. Tate, 277-282. Menlo Park, Calif.: AAAI Press, 1996.

Vere, S., Planning in time: Windows and durations for activities and goals, IEEE Transactions on PAMI 5 (1983) 246-267.