Mixed-Initiative Planning in MAPGEN: Capabilities and Shortcomings

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

MAPGEN (Mixed-initiative Activity Plan GENerator) is a mixed-initiative system that employs automated constraint-based planning, scheduling, and temporal reasoning to assist the Mars Exploration Rover mission operations staff in generating the daily activity plans. This paper describes the mixed-initiative capabilities of MAPGEN, identifies shortcomings with the deployed system, and discusses ongoing work to address some of these shortcomings.

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.

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.

In this paper, we report on the mixed-initiative capabilities of the MAPGEN system, outline some of the shortcomings that we observed during the deployment effort or during mission operations, and then briefly describe more recent research that is attempting to address some of these shortcomings. We first present some background material on the MER mission and then summarize the characteristics of the MAPGEN system.

* Research Institute for Advanced Computer Science - USRA

Background

The MER rovers (see Figure1), Spirit and Opportunity, are solar-powered (with a storage battery) and incorporate a capable sensor and instrument 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.

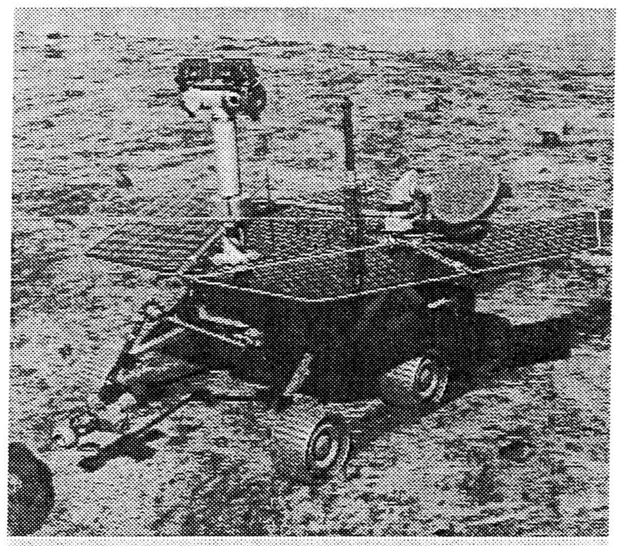

Figure 1: MER Rover

An onboard computer 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 used judiciously.

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 and the Deep Space Network schedule or orbital schedules for the satellites.

For this mission, the communication cycle was designed so that both rovers could be commanded every sol (i.e., 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 nevertheless finish in time to get the command load to the rover. During the nominal mission, this left 19.5 hours for ground operations. In this process, 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. At this stage only rough resource guidance is available. Hence, the scientists are encouraged to oversubscribe to ensure that the rover's resources will be fully utilized in the final plan.

In the next step in the commanding process, the science observation requests are merged with the engineering requirements (e.g., testing the thermal profile of a particular actuator heater) 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 the Tactical Activity Planner (TAP) employs MAPGEN.

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.

MAPGEN System Summary

Traditionally, spacecraft operations' planning is done manually; utilizing software tools 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 ground-operations problem.

As an integral part of a large mission operations system, MAPGEN's capabilities have 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 subject to the changing landscape of the overall operations system. We can summarize the primary features as follows:

- **Plan editing:** Both activities and constraints can be modified, via direct manipulation, form editing, or menu items.

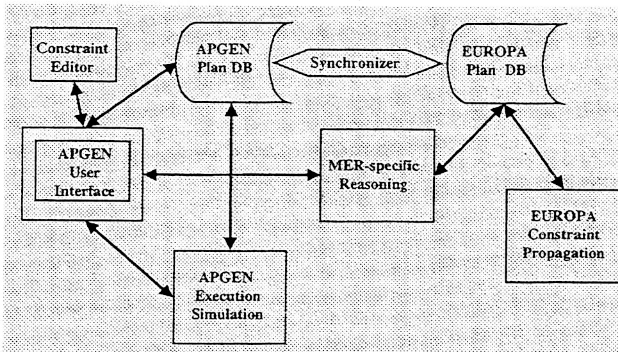

Figure 2: MAPGEN Architecture

- **Plan completion:** The selected subset of activities can be completed, in the sense that all subgoals are achieved and any necessary support activities are added to the plan.
- **Active constraints:** During plan editing, the formal 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.

The MAPGEN system has five primary components, some of which were pre-existing software modules (see Figure 2). 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.

We next further describe the EUROPA, APGEN, and Constraint Editor components.

EUROPA

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`

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 decision choices 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, 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 "on the fly" relations between specific activities in a planning problem. 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**: A method for calculating, using forward simulation, resource states that range from simple Boolean 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 useful 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.

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 APGEN interface. In this tool, users can view activities and existing temporal constraints, and then add, delete, or edit constraints.

Mixed-Initiative Planning in MAPGEN

In this section, we first motivate the need for a mixed-initiative approach to activity planning and then describe the capabilities in MAPGEN that supported this approach.

In traditional automatic planning, the operator loads in the goals and initial conditions, pushes a button, and waits for a complete plan. Due to the need to bring human expertise in mission planning and science operations to bear on solving this complex operational problem, this approach was deemed unacceptable; consequently, we adopted a mixed-initiative approach for this application.

There were many aspects of 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 by the Tactical Activity Planner (TAP),

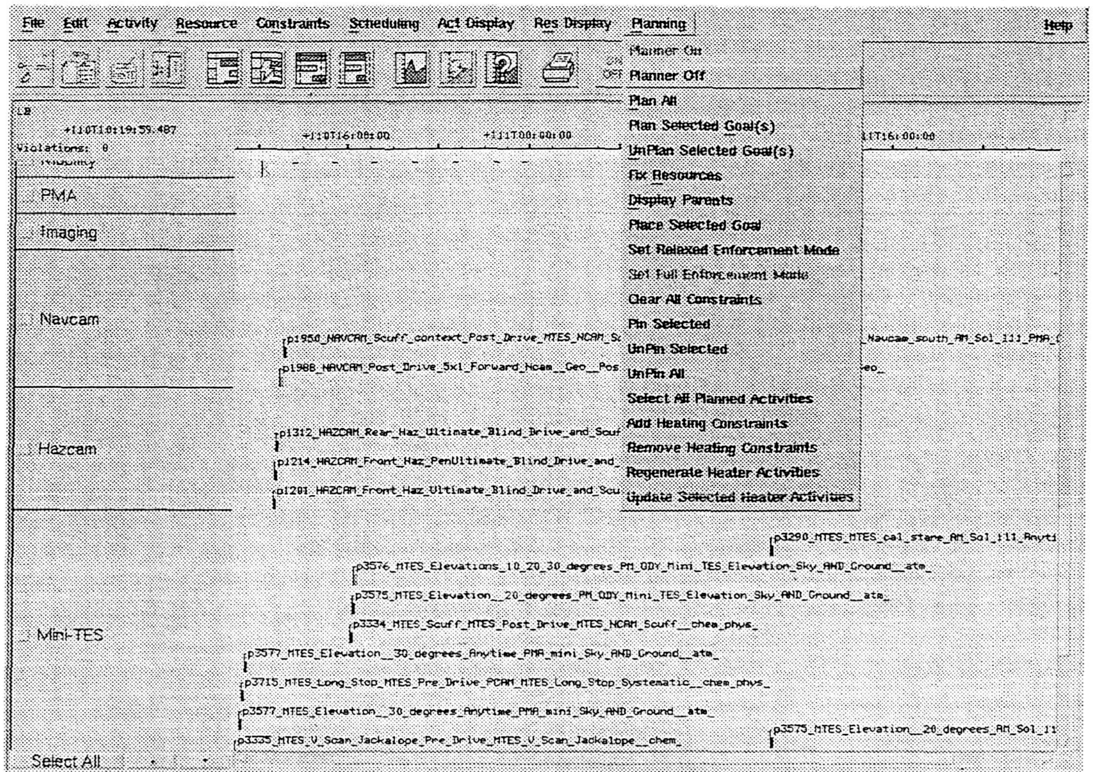

Figure 3: MAPGEN with planner menu

critiqued by both scientists and mission specialists, and, hopefully, accepted, possibly with minor modifications. As a result, the TAPs had to be able to understand, defend, and sign-off on the validity of the plan. 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 major 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.

Interactive plan modification

One of the core issues in mixed-initiative planning is the introduction of external decision-making and plan editing into a carefully designed automated search engine. The intrusion of user choices complicates commonly used approaches such as backtracking search and propagation-based checking of consistency. The EUROPA planning framework used in MAPGEN supports non-chronological backtracking, but it cannot propagate information in plans that have constraint violations. To support arbitrary changes by users, MAPGEN included a plan modification strategy that would adjust plans to eliminate inconsistencies.

Mixed-initiative planning systems must respond and return control quickly to the user. For an automated planning operation, which involves a cascading decision process, MAPGEN relaxes completeness in favor of responsiveness. This has to be done carefully to maximize chances of finding near-optimal solutions within limited time. We developed a backtracking algorithm that noted the difficulty of planning activities, and when the effort to plan an activity exceeded an allowance determined by its priority, the activity was rejected from the plan.

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 support a common way for users to modify plans, namely to change the placement of activities in 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.

Adjustable automation

MAPGEN users wanted an adjustable spectrum of automated planning services (see Figure 3). The system offers a fully automated "plan everything" operation, a selective "plan this and everything related to it" operation, and a fine-grained "plan this and try to put it here" operation. Users can also un-plan activities and store them in a "hopper," which holds requested activities that are not yet in the plan.

The *plan all* operation leaves 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 more incremental operation, called *plan selected goals*. With this operation, 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.

The user could exercise even more control over the planning process via the *place selected goals* operation, which was applicable only to individual activities. This operation 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* a part of the plan editing capabilities, they are, in fact, a small re-planning 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.

Minimizing perturbation

The key to making the automated services feel natural and unobtrusive is for them to respect the existing plan as much as possible. This is accomplished by combining an effective form of temporal placement preference with a heuristic bias. For changes in the temporal placement of activities, the system exploits the underlying temporal flexibility of EUROPA plans. As each plan represents a family, the system chooses an instance to display that is as close as possible to what the user had prior to the changes being made.

The method we developed is based on minimizing the departure from a *reference* schedule, which need not 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 user to reflect preferences or eliminate problems are respected and maintained unless they violate constraints or are revised by the user.

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. For more details on this method, see (Bresina, et al., 2003).

Addressing MAPGEN's Shortcomings

During the multi-year deployment effort, there were a number of capabilities on our task agenda that never made it to the top of the stack; we also encountered issues that require significant research before being ready for mission deployment. During mission operations, we observed a number of shortcomings, and often we were not able to address them at that time due to the restrictions of the change control process or due to the complexity of the issue. In this section, we focus on the shortcomings in MAPGEN's mixed-initiative approach and describe some of the new research we are carrying out to address them.

Explanations

The clearest lesson we have learned from our observations is the need for the automated reasoning component to provide better explanations of its behavior. Especially important are 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 useful when editing constraints, only the developers used this facility in the context of constructing and modifying plans, and this was done 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. For example, during MER, nogoods encountered during planning could involve hundreds of constraints.

There are several contexts in which inconsistencies can arise during planning. First, when an activity is considered for insertion, it may be inconsistent with the current plan even before any location is examined. Second, it may be inconsistent with the *specific* location chosen in a Place Selected operation. Third, it may be inconsistent with *each one* of the possible locations identified during a Plan Selected operation. The first context gives rise to a nogood directly. In the second context, a nogood can be extracted by temporarily placing the activity in the infeasible location. In the third context, it may be possible to resolve the individual nogoods arising from each location to form a compound nogood. Note that these cases may arise before or during the search. We have focused our efforts thus far on the first context; we expect similar considerations to apply in the other contexts.

The lengthy nogoods are partly an artifact of the mixed-initiative planning process. When MAPGEN attempts to insert an additional activity into the evolving plan, it first brings in (i.e., starts enforcing) the constraints associated with that activity. Since the existing plan was formulated without those constraints, it is often the case that they are inconsistent with previous ordering decisions made to prevent forbidden overlaps (due to mutual exclusion restrictions). Furthermore, the ordering decisions may involve mutual exclusions between low-level activities that are part of activity expansions. Because of this, the constraint engine must keep track of interactions between activity expansion constraints and planner decision constraints, as well as daily constraints. The duration of a high-level activity is also determined by its activity expansion constraints, so if this is a factor in an inconsistency, the raw nogood will include the entire expansion of the high-level activity. Thus, the raw nogoods during planning can be very large. Note that this is not an issue for the nogoods occurring in the constraint editor, because only top-level activities are considered there and orderings resulting from planner decisions are not involved.

It is obviously impractical to expect a time-pressured TAP to read, let alone grasp the significance of, a nogood involving hundreds of constraints. However, we believe that the essential content of the nogood can be summarized in a concise form. To this end, we have been investigating methods of *compressing* nogoods.

The first compression step rolls up expansions that are only needed because they determine a higher-level duration that is involved in the inconsistency. While this step helps, the explanations can still be quite long, often involving chains of duration and daily-constraint pairs. We can distinguish between these constraints, which should be known to the TAP, and the "hidden" constraints that come from planner ordering decisions. The second compression step rolls up the duration/daily sequences into a single chunk. Based on MER examples, these two steps typically compress the nogood by a factor of about ten.

A remaining issue is that sub-chains of the nogood that pass through planner ordering decisions can wander somewhat randomly through large portions of the plan. The intermediate wandering is not very meaningful in terms of understanding the inconsistency, so a further step could involve rolling such a segment into a single statement about planner placement of the bookend activities in the segment.

These compression steps carry the risk that one of the components of the compressed summary will itself be mystifying. To counter this, it would also be useful to allow components of the summary to be re-expanded on demand. Thus, the nogood would be organized into a hierarchical structure that is more easily grasped.

In general, an inconsistent network may involve more than one inconsistency. The approach used in the constraint editor is to first present one (the first one found by the temporal reasoning algorithm), have the user resolve that, then present another one if the network is still inconsistent, and so on. This may not be the best approach within the planning context.

Considering the entire set of nogoods, it may be possible to select the one nogood that yields the "best explanation", i.e., an explanation that is easiest to understand and leads to the easiest resolution of the associated inconsistency. Another approach is to focus on constraints common to multiple nogoods, such that the user could resolve more than one inconsistency with one constraint retraction. A prerequisite for either of these approaches is a suitable algorithm for enumerating all the temporal nogoods. At this point, it is not clear how practical it is to compute such an enumeration, since theoretically the number of nogoods may be exponential in the size of the network.

Temporal preferences

A second important issue is that the user does not have sufficient means to control the planning process and to influence the types of solutions generated. In MAPGEN, the user's only language for specifying their desires is to create a set of absolute (hard) temporal constraints, which represent what is necessary for the observation requests to be scientifically useful. These constraints can specify ordering among the activities and observations (along with temporal distances required) and can specify that an activity or observation has to be scheduled within a particular time window. For example, the scientist can specify that three atmospheric imaging activities have to be a minimum of thirty minutes apart and a maximum of six hours apart. However, the scientist cannot specify that they *prefer* the largest possible spacing between the three activities. Likewise, they cannot specify that a particular spectrometer reading must occur between 10:00 and 15:00 but it is preferred to be as near to 12:00 as possible. It is clear that both absolute constraints and temporal preferences are needed to generate a high-quality science activity plan.

MAPGEN did have a limited capability for expressing start time preferences via the reference schedule of the minimal-perturbation approach. The operator could also establish more complex preferences by an iterative process of relaxing or tightening hard constraints, but this is too time-consuming and too primitive of an approach.

We are currently investigating a number of alternative, automated approaches to incorporating temporal preferences into MAPGEN. We have extended the Constraint Editor to allow specification of temporal preferences on an activity's start or end time, as well as on distances between start/end time points of two activities.

There are three key issues involved in utilizing temporal preferences in mixed-initiative planning. The first is the usual problem of how to combine local preferences into global evaluation functions. The second problem is a generalization of the question of how to instantiate a flexible plan so that perturbation is minimized, i.e., given a flexible plan and a set of temporal preferences, which instantiation should be chosen. Finally, the third problem is how to guide the search for flexible plans so as to generate a plan that can be instantiated into a solution that is globally preferred.

Let us first look at the second problem. To effectively solve constraint problems that have local temporal preferences, it is necessary to be able to order the space of assignments to times based on some notion of global preference, and to have a mechanism to guide the search for solutions that are globally preferred. Globally optimal solutions can be produced via operations that compose and order partial solutions. Different concepts of composition and comparison result in different characterizations of global optimality. Past work (Khatib, et al., 2001; Khatib, et al., 2003, Morris, et al., 2004) has presented tractable solution methods (under certain assumptions about the preference functions) for four notions of global preference: weakest link, Pareto, utilitarian, and stratified egalitarian. These four notions are examples of general solutions to the first problem, namely, how to combine local preferences into an overall comparison of solutions.

We are incorporating these preference-optimization methods into MAPGEN and plan to employ them for a number of purposes. One use is to apply the optimization, as a post-process, to the family of solutions represented by a flexible MAPGEN plan in order to display the most-preferred solution to the user. These methods can also be employed, as a pre-process, to compute the reference schedule as a globally optimal solution to the specified temporal preferences. The minimal-perturbation method would then try to stay close to this globally optimal reference. We also intend to investigate other heuristic methods that include consideration of the preferences when making search decisions.

Other shortcomings

The need for explanations and handling of temporal preferences were the most obvious shortcomings that needed to be addressed. Consequently, work is already underway to address those. However, a number of other issues have been identified.

In addition to temporal preferences, users may have preferences regarding the global characteristics of the solution, such as plan structure preferences or resource usage preferences. Many constraints can have absolute validity limits and a preference on the legal values. For example, the limits on the energy usage may be determined by minimum battery levels, but it is preferred that the battery be left charged above a certain level at the end of the plan. As with temporal preferences, the main issues are how to combine local preferences into global evaluations functions and how to then control the search towards preferred plans.

In MAPGEN, the underlying plan is always kept consistent. This allows propagation to take place at any time, which in turn enables active constraint enforcement, constrained moves, and other propagation-based capabilities. However, the users sometimes desire to "temporarily" work with plans that violate rules or constraints. One possible approach for allowing violations is to isolate the inconsistent parts of the plan; a second approach is to allow constraints and rules to be disabled and re-enabled. The latter approach was in fact designed for the MAPGEN tool, but we never got a chance to implement it. Future work will explore possible approaches and techniques for this.

The users also want to advise the planner on how it makes decisions at a high level and on how the planner's search is done. Users have noted that they would like to specify limits on what the automated reasoning process can change in order to enforce constraints and rules. For example, users may want a portion of the plan to remain unchanged, either in terms of a timeframe or a set of activities.

It would also be useful for the system to answer questions from the user regarding trade-offs, for example, by answering the following types of queries:

- 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?

Another technique for supporting trade-off analyses is to help the user better understand the space of possible solutions by presenting qualitatively different solutions. We are extending some previous work on advisable planners (Myers, 1996; Myers, et al., 2003) to apply within the context of our constraint-based planning technology in order to help address these issues.

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, Adans Ko, Pierre Maldague, Richard Springer, and Jeffrey Yglesias. We would also like to acknowledge our research collaborators on mixed-initiative planning, Karen Myers and Michael Wolverton from SRI.

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.

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.

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

Khatib, L., Morris, P., Morris, R., Rossi, F. Temporal reasoning about preferences. *Seventeenth International Joint Conference on AI*, Seattle, WA, 2001.

Khatib, L., Morris, P., Morris, R., Venable, B. Tractable pareto optimization of temporal preferences. *Eighteenth International Joint Conference on AI*, Acapulco, Mexico, 2003.

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.

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.

Myers, K. L., Jarvis, P. Tyson, W. M., and Wolverton, M. J. 2003. "A Mixed-initiative Framework for Robust Plan Sketching". In *Proceedings of the 13th International Conference on Automated Planning and Scheduling*, Trento, Italy.