AI Magazine Volume 28 Number 2 (2007) (© AAAI)

Mixed-Initiative Planning in Space Mission Operations

John L. Bresina and Paul H. Morris

■ The MAPGEN system represents a successful mission infusion of mixed-initiative planning technology. MAPGEN was deployed as a mission-critical component of the ground operations system for the Mars Exploration Rover mission. Each day, the ground-planning personnel employ MAPGEN to collaboratively plan the activities of the *Spirit* and *Opportunity* rovers, with the objective of achieving as much science as possible while ensuring rover safety and keeping within the limitations of the rovers' resources. The Mars Exploration Rover mission has now been operating for more than two years, and MAPGEN continues to be employed for activity plan generation for the *Spirit* and *Opportunity* rovers. During the multiyear deployment effort and subsequent mission operations experience, we have learned valuable lessons regarding application of mixed-initiative planning technology to mission operations. These lessons have spawned new research in mixed-initiative planning and have influenced the design of a new ground operations system, called M-SLICE, that is baselined for the Mars Science Laboratory mission. In this article, we discuss the mixed-initiative aspects of the MAPGEN system, focusing on the task, control, and awareness issues.

In this article, we address three of the issues in mixed-initiative systems raised in this special issue: the task issue, the control issue, and the awareness issue. In the context of mixed-initiative assistants, the overall problem is solved through a collaborative effort between the system, or agents, and the users. The task issue involves the division of responsibility between the human and the system for the tasks that need to be performed. The control issue examines when shifts in initiative take place and what control restrictions are placed on the user and the system in order to make the collaboration an effective one. The awareness issue deals with the shared awareness that is needed for an effective collaboration between human and machine and the communications that are needed to achieve it.

The organization of the article is as follows. We first present background material on space mission operations in the Mars Exploration Rover (MER) mission and then present a summary of the mixed-initiative activity plan generator (MAPGEN) system and describe how it is employed in the tactical ground operations for MER. Note that we focus on its use within *nominal mission* operations, which comprise the first ninety sols (Mars solar days) of the mission. Within the ongoing *extended mission* phase that began after the nominal mission, some aspects of the operational process have been streamlined as more experience was gained, primarily for the purpose of reducing the workload and stress on the operations staff. For example, initially operations were carried out every day of the week, essentially around the clock and in synch with the local time on Mars, but now they are mostly accomplished within the hours of a normal workweek.

Next, we outline two post-MAPGEN projects. Within the context of this background material, we then address the three mixed-initiative issues. Note that, though we discuss each issue separately, there are interactions among all three issues. Lastly, we present some concluding remarks.

MER Mission Operations

In this section, we describe the Mars Exploration Rover mission and its commanding process. In January 2004, the U.S. National Aeronautics and Space Administration (NASA) landed rovers on the surface of Mars at two widely separated sites. Their mission is 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 and showing signs of wear, both rovers are still functioning, and the mission is continuing.

The MER rovers (see figure 1), *Spirit* and *Opportunity,* are solar-powered (with a storage battery) and incorporate a capable sensor and instrument payload. Panoramic cameras (pancams), navigation cameras (navcams), 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.

An on-board computer governs the operation of subsystems and provides data handling, system state tracking, limited obstacle avoidance, and other functions. 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 a low-gain antenna for direct-to-earth transmission and reception and a 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 each rover could be commanded every sol (which averages 24 hours, 39 minutes, and 35.2 seconds). The daily commanding cycle in MER's nominal mission proceeds as follows. 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 next step in the commanding process, the science observation requests are merged with the engineering requirements (for example, testing the thermal profile of an actuator heater), and a detailed plan of activities is constructed for the upcoming sol. The plan must obey all applicable flight rules that specify how to operate the rover and its instrument suite safely and remain within specified resource limitations. It is in this step that a human operator, called the tactical activity planner (TAP), employs the MAPGEN tool. Once approved, the activity plan is used as the basis for creating sequences of low-level commands, which drive on-board execution. This sequence structure is then validated, packaged, and communicated to the rover. This completes the commanding cycle. Figure 2 illustrates how this commanding cycle temporally corresponds to the activities of the rovers on Mars. Note that the plan in the figure is oversimplified; real plans have up to 100 top-level activities and 3500 lower-level activities.

The MAPGEN System

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 plan editing, plan completion, and active constraints.

In plan editing, both activities and constraints can be modified, through direct manipulation, form-based editing, or menu operations. In 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. In 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.

An existing interactive plan editor from JPL, called APGEN (activity plan generator) (Maldague et al. 1998), is used as the front end

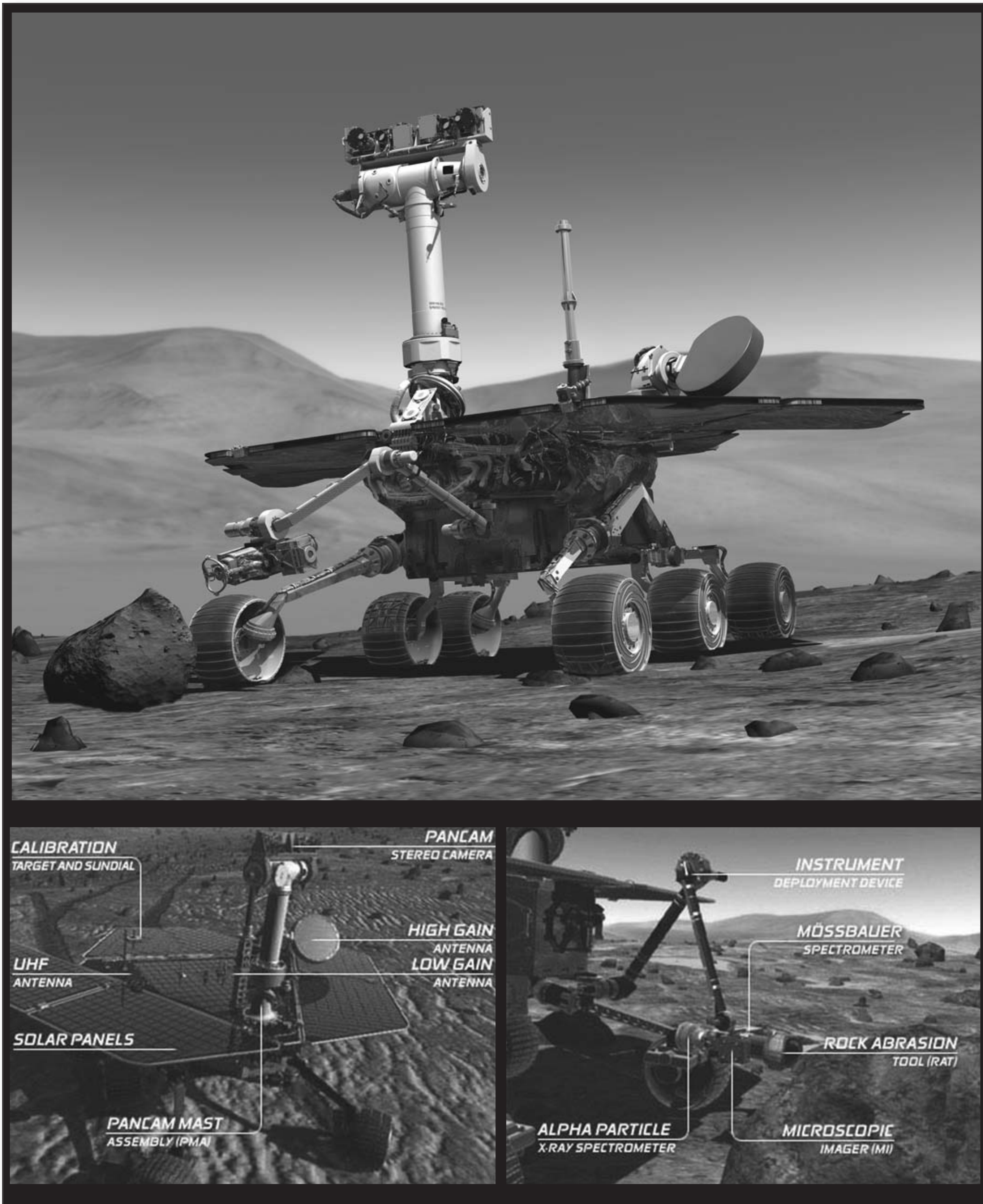

Figure 1. Mars Exploration Rover.

Mars Rover renderings by Dan Maas / Maas Digital LLC for Cornell University and NASA/JPL, © 2002 Cornell University.

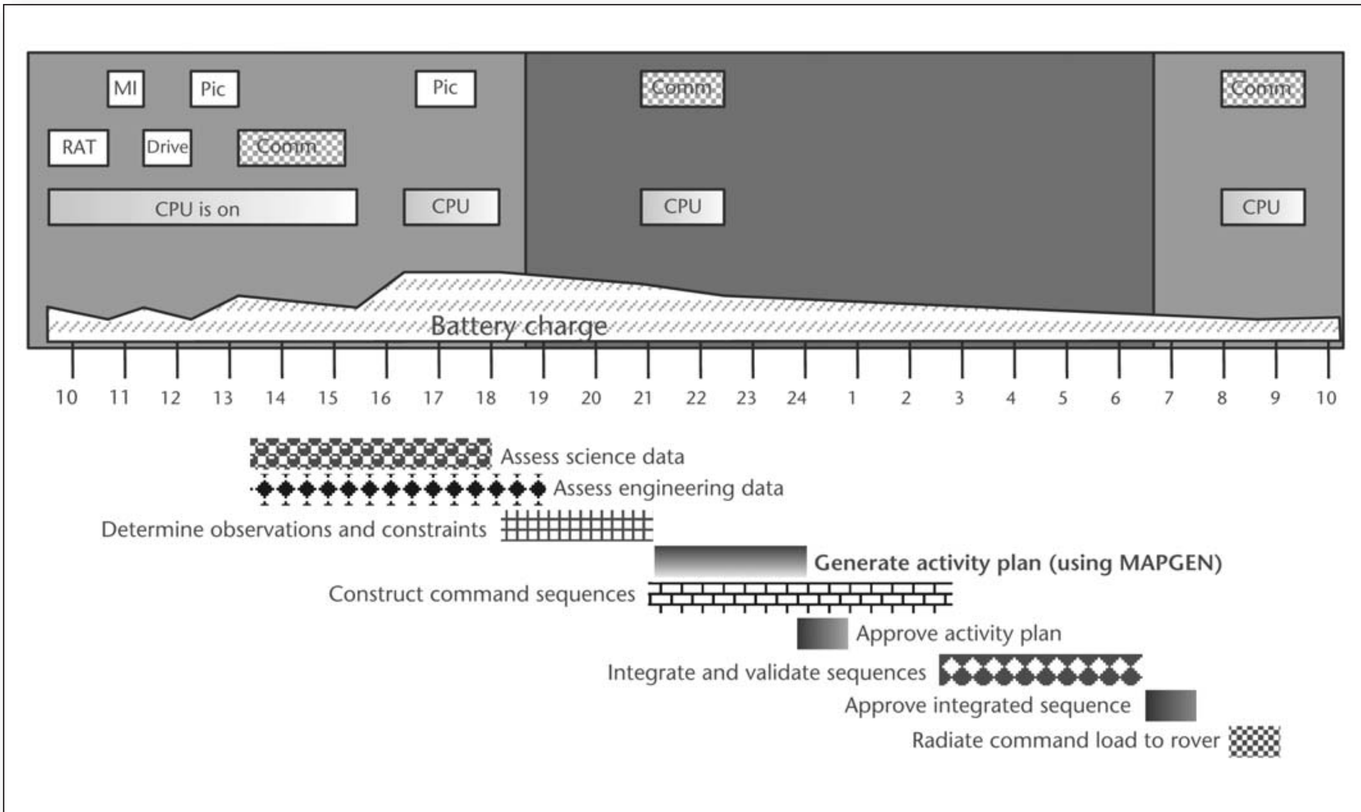

Figure 2. MER Mission Operations Timeline.

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 (Bresina et al. 2005a) in the MAPGEN system involves the interface between these two subsystems, support for extensions to the APGEN graphical user interface (GUI) 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.

Figure 3 illustrates the primary user interfaces of the MAPGEN system. In addition to the main plan display window, the science requests that are not currently in the plan are kept in a separate display window, called the hopper. The top half of the figure illustrates MAPGEN's main plan display window with the hopper window overlaid as an inset. The main plan display window is the standard APGEN GUI with the addition of the planning menu, and the hopper window is new to MAPGEN. The MAPGEN system is supplemented with a separate external tool, called the constraint editor, which is used to enter and edit science constraints. Figure 4 shows the constraint editor's web-based interface; it illustrates part of its main display window as well as two of its constraint-specification forms. The main window shows ordering constraints between observations (the larger boxes) as well as ordering constraints between activities within observations (the small circles within the observation boxes). The figure contains a form for specifying such ordering constraints and a form for specifying time of sol constraints.

We next further describe the EUROPA, APGEN, and constraint-editor components.

EUROPA

A combination of constraint-reasoning technology and planning and scheduling technology provides the foundation for EUROPA. In this approach, pioneered in the Remote Agent Experiment on the Deep Space 1 mission (Muscettola et al. 1998), planning and scheduling are performed at the same time, using an underlying temporal constraint-reasoning sys-

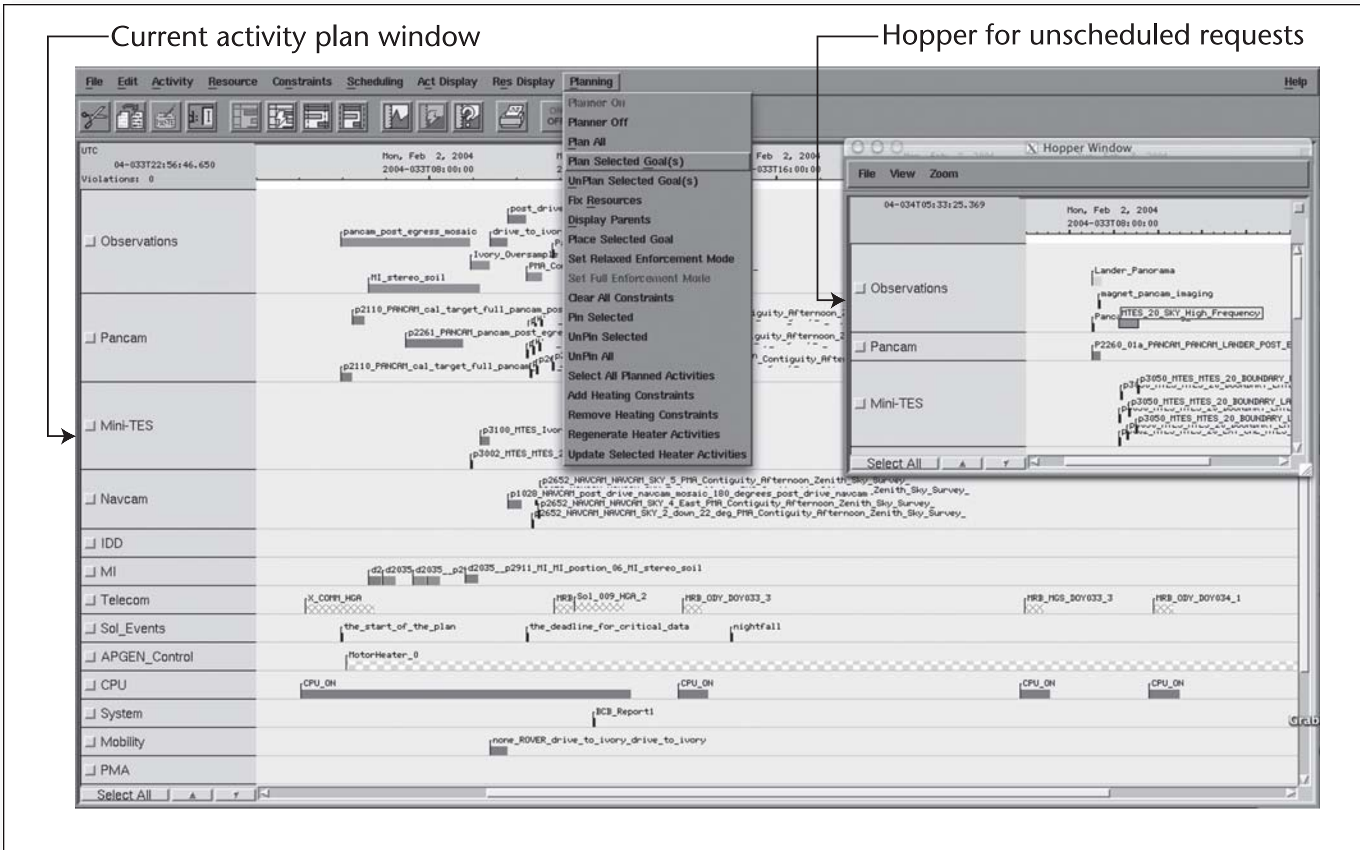

Figure 3. MAPGEN User Interfaces.

tem to maintain a consistent schedule that satisfies applicable rules.

Consistency of the developing plan is maintained using an underlying simple temporal constraint network, or STN (Dechter, Meiri, Pearl 1991). One advantage of STNs is that rather than doing simple consistency checking, they work by eliminating inconsistent values from variable domains. Specifically, they maintain *arc-consistency*, which for STNs is equivalent to full consistency. In effect, they maintain a family of related solutions, called a flexible solution, rather than just a single grounded solution. A flexible solution provides flexibility because it can often merely be refined, that is, further restricted, in response to additional constraints instead of requiring search for a new solution.

The temporal constraints in EUROPA fall into three categories: *model* constraints, *problem-specific* constraints, and *expedient* constraints. Model constraints encompass domain definitions and mutual-exclusion flight rules, for example, *do not move the arm while the rover is moving* or *more than one activity cannot simultaneously point the rover's mast.*

The problem-specific constraints comprise relations between specific activities in a planning problem instance. In MER, these constraints are used to ensure that science objectives are satisfied and that the data collected are scientifically useful; thus, for this domain, we also refer to these constraints as science constraints. The scientists use two types of problem-specific constraints: temporal bounds and temporal ordering relations. The temporal bounds are typically constraints on when an activity can start due to, for example, lighting conditions or temperature. The typical ordering relations are constraints between the end of one activity and the start of another. For example, a hazcam documentation image of an arm placement must be taken at least 2 minutes after the arm is placed (to ensure vibrations have subsided) and before it is moved again. Another more complex example is that a pancam imaging activity must be within 30 minutes of its associated calibration activity, but the activities can occur in either order.

Expedient constraints are typically added during search in automated planning. For example, a model constraint might specify that two activities, *A* and *B,* are mutually exclusive. Thus, either *A* must precede *B,* or *B* must pre-

Figure 4. Constraint Editor.

cede *A*. One of these alternatives is chosen and added as an STN constraint.

APGEN

APGEN is a JPL tool that 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, resource calculations, and graphical user interface.

Activity plan database: a set of activities, each at a specific time. This database does not maintain constraints between activities, but does support activity expansion (without search).

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 manually creating and editing plans and for viewing resource profiles.

To adapt APGEN for a particular mission, the mission-specific information is encoded 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 resource states. The constraints and resource calculations are useful only for passively identifying problems with a plan. APGEN displays a tick mark at the time of each violation; for each tick mark, the user can find out which constraint is violated, but APGEN does not identify the "culprits," that is, the activities that caused the violation. Furthermore, APGEN does not have the capability to reason about the violations in order to help resolve them.

Constraint Editor

The APGEN plan-editing interface does not support entering temporal constraints. This raised the issue of how to get the science constraints into the reasoning component of MAPGEN. These problem-specific constraints were needed to coordinate the activities in scientific observations, and they could vary in arbitrary ways. This required an ability to enter and modify temporal constraints dynamically. To resolve this, an external, temporal constraint-editing tool, called the constraint editor (illustrated in figure 4), 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.

The constraint editor has two categories of constraints: time-of-sol constraints on individual activities (see the bottom of figure 4) and ordering relationships between two activities (see the middle of figure 4). The former are typically used to ensure that science data will be collected at the appropriate time, where appropriateness may depend on lighting conditions, temperature, or timing concerns with respect to data collected on previous sols. The ordering relationships are often used between tightly related activities, for example, to specify that a pancam calibration must be within 30 minutes of the associated science imaging activity. They are also used to constrain the overall plan structure in order to convey the plan's science intent; for example, if the rover is driving, then it is important to indicate which activities must be done before the drive and which must be done after the drive.

Mixed-Initiative Planning in MAPGEN

The tactical activity planner (TAP) employs MAPGEN to collaboratively plan the activities of each rover, with the objective of achieving as much science as possible while ensuring rover safety and keeping within the limitations of the rover's resources. Figure 5 depicts the task context of the TAP, including the interactions with the rest of the operations staff. Within the activity planning process, the role of the TAP is to direct construction of the plan and fine-tune it by bringing to bear expertise that is outside MAPGEN's domain model and beyond its scope of reasoning, such as shunting of the battery and scientist preferences. The intended interaction between user and system is that the system handles constraint enforcement constantly in the background, while automated plan-construction operations are user initiated. The planning process is an incremental one in which the TAP interleaves automatic plan generation and plan-editing phases. Each planning operation is done in the context of the current partial plan and its constraints; thus, previous planning decisions affect what future operations are possible and what additional activities will fit in the plan. This incremental commitment helps the TAP better understand a gradually developed plan. Another advantage is that MAPGEN achieves a fast response time—a satisfactory plan is available at an early time, with additional time devoted to improving the plan quality.

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 the chances of finding solutions within limited time. We developed a backtracking algorithm that noted the difficulty of planning activities, and when the effort to plan an activity exceeds an allowance determined by its priority, the activity is rejected from the plan.

One of the key design characteristics of MAPGEN is *user-adjustable autonomy*: MAPGEN provides a spectrum of automated planning services with different degrees of automation and human guidance. At the full-automation end of the spectrum is the *plan-all* operation—the planner will attempt to fit all the activities into the plan, and the ones that do not make it into the plan get placed back in the hopper. This operation was rarely used during the nominal mission[1] because the TAPs tend to build plans in an incremental fashion, checking the energy resource usage by invoking an external power-thermal detailed modeler every now and then. Due to this incremental approach, the TAPs often apply the *plan-selected* operation. With this operation, the user can select a set of observation requests not in the plan and request that these be inserted anywhere into the current partial plan, such that all constraints are satisfied. The user can exercise even more control over the planning process through the *place-selected* operation, which is applicable only to individual activities. This operation allows the user to select an activity in the hopper and then choose an approximate temporal placement for it in the plan. The planning algorithm then treats the user-chosen time as heuristic guidance and searches for a plan in which the selected activity is as close to the desired time as possible.

The system also supports an activity movement operation, called *constrained-move*, which takes advantage of the flexibility in the STN. As long as an activity is moved only within the flexibility range defined by the underlying arc-consistent flexible plan, the result is necessarily another consistent instantiation. During a constrained move, the system actively restricts the movements of the selected activity to stay within the permitted range and gives a visual indication of the range. Then, once the user places the activity, any dependent activity is automatically updated as necessary to yield a new valid plan instance. Note that the consistency enforcement takes into account all the constraints that determine the flexible plan, including expedient constraints that arbitrarily order activities. In order to allow for the possibility of overriding the expedient constraints, the system also provides an operation called a *super-move*. This temporarily removes the activity being moved from the plan (thus deleting its expedient constraints) and attempts to place it at the new location, as in a *place-selected* oper-

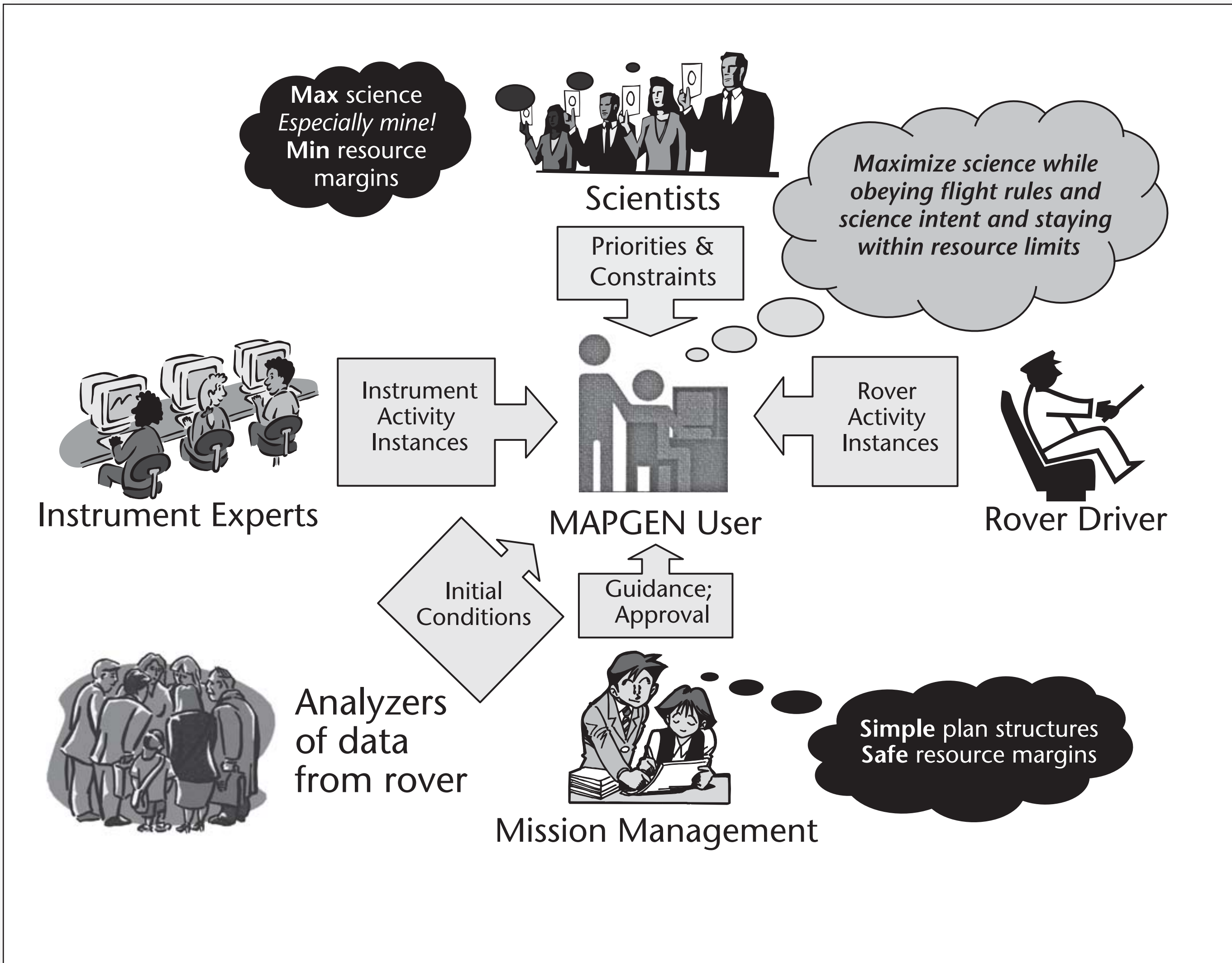

Figure 5. Tactical Activity Planner's Task Context.

ation. If the attempt fails, the activity is returned to its original location.

Although MAPGEN constructs flexible plans, the plan that is displayed to the user is a grounded solution, that is, a specific consistent instantiation of the underlying flexible plan, in accordance with the following *solution grounding algorithm*.

For each timepoint *x* with reference position *t* do the following:

(1) If *t* is within the STN bounds for *x*,
then add a grounding constraint that sets *x* to *t*.
Else if *t* is less than the lower bound (*lb*) for *x*,
then add a grounding constraint that sets *x* to *lb*.
Else if *t* is greater than the upper bound (*ub*) for *x*,
then add a grounding constraint that sets *x* to *ub*.

(2) Propagate the effect of the new constraint.

The grounded solution is thus selected to be "close" to an internally maintained *reference schedule*. The reference schedule is initially based on the science constraints and the initial start times of the activities, which are set by the scientists. This initial reference is computed by first solving a relaxed version of the planning problem composed of only the science constraints (that is, it does not include flight rules); the solution produced is a flexible plan. The reference schedule is determined by grounding this flexible plan, by the above algorithm, to be close to the initial activity start times. Hence, the scientists can bias the initial reference schedule to reflect their preferences. For example, to reflect science preferences, each activity's start time could be set to an ideal time with respect to science quality. Another option is to bias the placement of activities to be when solar power is at a maximum by setting all start times to the time of peak power. During the collaborative planning process, the reference schedule is continually updated to reflect the evolving plan. For example, if the TAP moves some activity, then this new start time becomes incorporated into the reference schedule.

In addition to determining which grounded solution to display, the reference schedule is also used to support a *minimum perturbation* approach. The idea behind this approach is, when adding more activities to a plan, to try not to perturb the existing plan. Users tend to expect that small extensions to a plan will cause only minor, local plan modifications and dislike it when they cause drastic, global modifications. Such drastic change makes it more difficult for the TAP to maintain an understanding of the plan. More importantly, needlessly perturbing the existing plan could undermine previous fine-tuning by the TAP, which could have significant impact on the quality of the solution. This approach is accomplished through a planning heuristic that is used when deciding how to order activities. The *minimal perturbation heuristic* biases the ordering decisions such that the activities remain as close to their reference times as possible. Hence, this tends to have the effect that activities are moved by only small amounts. In order to guarantee that the planner will not move an activity at all, the TAP can *pin* it, which adds a constraint fixing its time.

Beyond MAPGEN

During the multiyear deployment effort and subsequent mission operations experience, we have learned valuable lessons regarding application of mixed-initiative planning technology to mission operations. These lessons have influenced the design of a new ground operations system, currently called M-SLICE, that is baselined for the Mars Science Laboratory mission. A joint JPL/Ames team is developing this new operations system. A second post-MAPGEN project is an ongoing collaboration between Ames and SRI exploring the use of explanations and preferences in mixed-initiative planning (see Bresina et al. 2005b). In particular, we have investigated explanations of temporal inconsistencies and recommendations for resolving such inconsistencies. We have also investigated how to satisfy users' preferences in addition to their constraints.

From the planner viewpoint, M-SLICE has a similar architecture to MAPGEN. With regard to planner capabilities, one of the significant developments has been the inclusion of EUROPA2, a next-generation version of the EUROPA planner used in MAPGEN. This new core system has enabled us to provide many additional capabilities. Our main interest here is that the new planner differs from MAPGEN in a number of key design choices that affect the nature of the mixed-initiative collaboration between system and user. For more details on these post-MAPGEN projects, see Bresina and Morris (2006).

In the next sections, we address three of the issues in mixed-initiative systems that the special issue editors have raised, primarily within the context of the deployed MAPGEN system and occasionally within the context of the two post-MAPGEN projects.

The Task Issue

Mission operations rely on a number of checkpoints and acceptance gates to ensure safety. For activity plans, the critical gate is the activity plan approval meeting at which the fully constructed plan is presented by the TAP, critiqued by both scientists and mission specialists, and, we hope, accepted, possibly with minor modifications. As a result, it is the responsibility of the TAP to defend, and sign-off on, the validity of the plan. This responsibility affects the style of use of MAPGEN; this responsibility also impacts the awareness issue (addressed later). The system helps ensure plan validity with regard to mission flight rules. The APGEN component performs passive violation checking, and the EUROPA component actively enforces mutual exclusion rules (that is, mission rules disallowing concurrency between specified types of activities). However, the TAP is responsible for ensuring that the plan's resource profiles are within allowed margins; the resources of most concern are battery energy, data bandwidth, and temperature.

A related concern is the infeasibility of formally encoding and effectively utilizing all the knowledge that characterizes plan quality. One of the more complex aspects of plan 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. Another aspect of plan quality involves a rich set of science preferences. MAPGEN has no facility for encoding science preferences, but they are often expressed verbally to the TAP. Hence, it is the TAP's responsibility to take all these aspects of plan quality into consideration and improve the plan through manual fine-tuning of the plan, as time allows.

Science preferences and solution quality is also an awareness issue, which we revisit in that section and describe some of the new work addressing it.

Although it is common for users of mixed-initiative planning systems to be tasked with helping make decisions during search, MAPGEN never solicits user assistance while it is planning. Rather, the user is responsible for higher-level planner decisions, such as deciding which activities to plan next or which activities to unplan. Thus, the user can influence which activities get into the plan and which remain in the hopper. Furthermore, the user can influence the order in which activities are planned; if desired, the user can even determine the order by planning each activity separately. The user can also influence the placement of an activity through the *place-selected* operation, but the planner only uses this advice as a heuristic bias. There are also operations available to the user (for example, *supermove*) that can effectively override the planner decisions after the fact, if the user does not like certain planner choices.

The primary responsibility of the EUROPA component is to maintain the consistency of the temporal constraints—both those that capture the intent of the scientists and those that arise due to the mission flight rules. As previously mentioned, MAPGEN helps the user ensure plan validity through active enforcement of constraints and passive violation detection. The constraint-handling mechanisms are important in both the control and awareness issues as well, so we revisit this topic below.

In addition, MAPGEN automates certain routine planning tasks, such as determining when to boot the CPU and how long it needs to be on; it also helps the TAP create and schedule heating activities that are required to warm up actuators or electronics before usage. The constraint-editor component also helps avoid tedium by adding default constraints, such as all activities must start after the plan-start and before the plan-end, and all camera activities must occur before nightfall (though the user can override this default if the intent is to image a celestial object at night). Perhaps more importantly, the constraint editor alerts the user to potential inconsistencies in the science constraints entered.

The Control Issue

The MAPGEN system runs as a single-threaded process in which all operations are triggered by user keyboard and mouse input. From that point of view, all of the initiative is on the side of the user. However, that oversimplifies matters. Indeed, many of the TAPs would maintain that the system did take the initiative, even aggressively at times. Essentially, once the system is activated, it exercises initiative in terms of how it responds to each commanded operation. Thus, the real issue concerns the aspects of the planning process that are *effectively* controlled by the system and the user, respectively, and the (sometimes indirect) ways in which that control is exercised.

The high-level planning operations in MAPGEN are all user-invoked. This suite of operations allows the user to choose how much control to exercise over the planning process. The user cannot explicitly control the timeline ordering decisions that the planner makes in search; however, these decisions can be influenced through the *place-selected* operation and the minimal perturbation heuristic. The user controls the overall nature of the incremental planning process. For instance, the user determines the size and content of each planning phase through the selection of activities to plan: from a single activity to the entire set. Through this repeated selection process, the user affects the order in which activities are planned. The scientists also set activity priorities that govern the order within a group of activities.

Automatic planners typically have ordering heuristics to increase problem-solving effectiveness. MAPGEN employs heuristics to order the activities in the subset selected by the user. However, MAPGEN cannot depend on such heuristics; that is, it must be able to effectively plan when these heuristics are not followed, because the planner does not have total control over the global ordering. In fact, the user can completely determine the order in which activities are planned by always selecting a single activity to plan (through *plan-selected* or *place-selected*).

Because the TAP may change the plan in arbitrary ways between planning operations, including overriding some of the planner ordering decisions, a new planning operation is not allowed to backtrack over decisions from a prior planning operation; it can only further restrict the previous flexible plan or make new planning decisions. As a consequence of this limitation, the impact of the user's influence on planning order is strengthened.

The primary way in which MAPGEN exercises control is in enforcing consistency of the evolving plan's network of constraints. In MAPGEN, consistency is aggressively and constantly maintained in order to ensure plan validity. In addition to the consistency checks that the constraint editor performs, MAPGEN also checks the consistency of the science constraints when they are initially read in and cannot proceed until they are self-consistent. MAPGEN does not allow the user effectively to do anything that would violate the science constraints or violate the mission flight rules that are encoded in the planner domain model. For certain operations, this means the planner completes the operation, if possible, by further restricting the flexible plan to satisfy the flight rules. The *constrained-move* operation only allows movement that is guaranteed to leave the plan in a consistent state, but there are some user operations that could produce an inconsistency if they were allowed to stand; such operations are immediately undone by the system. For example, if the user *super-moves* an activity to a spot in the plan that is invalid, it is put back to its original position in the plan. Similarly, if the user *edits* the start time or duration of a planned activity such that it makes the plan inconsistent, then the edit is undone. A *place-selected* operation may similarly be ineffective. Although the TAP can revise the science constraints if need be, the flight rules cannot be modified or temporarily waived by the TAP.

There were times when the TAPs felt MAPGEN was a little too aggressive about enforcing plan validity, and they wanted to have the option to (at least temporarily) violate a flight rule or science constraint. In response to this feedback, we have designed the constraint-enforcement facility for M-SLICE to be more passive and user-adjustable. The planner constantly performs passive violation checking; however, it applies active enforcement of constraints only when the user requests it through the *fix-violations* operation. Furthermore, the user can adjust the flight rule enforcement facility; specifically, the user can disable, and reenable, a specified flight rule for all activities or all flight rules for a specified activity.

Another way that we increased the planner's flexibility in M-SLICE is by eliminating the backtracking limitation. This is accomplished by requiring the planner, in each incremental phase, to replan the activities from *previous* phases, along with the new activities. This design choice makes the planner's search complete, but additional work is incurred, which could potentially slow the system response. One ameliorating factor to the extra replanning effort is that the minimal perturbation heuristic biases the planner to rebuild the same plan as before, thus respecting the TAP's fine-tuning and also increasing planning efficiency.

The Awareness Issue

In the MER mission deployment of MAPGEN, the user's awareness is an important issue since the TAP has to sign off on the plan's validity and has to be an advocate of it to an approval committee composed of mission managers, engineers, and scientists. In addition to mission safety issues, a key aspect of the MER activity plans produced is that they must capture the intent of the mission science team. In order for MAPGEN to be aware of the science intent, it must be formally encoded in an effective form for planning. In this section, we discuss how MAPGEN is made aware of the intent of the scientists and the TAP, and we discuss how MAPGEN helped the TAP understand the planning process and the final plan. Additionally, we describe current research aimed at addressing both of these awareness issues based on lessons learned from operational experience during the MER mission.

The MER scientists express their intent to the MAPGEN system through the requested activities, the associated priorities, and science constraints. For most of the sols during nominal operations, it was impossible to fit every requested activity into the plan. When the planner had to reject one or more activities, it used the associated priorities in making the choice. By enforcing the specified science constraints, MAPGEN ensured that the data collected satisfied the science intent. However, in addition to these hard constraints, the scientists often have temporal preferences in mind, which could yield higher-quality data. Such temporal preferences cannot be formally encoded in MAPGEN. Some of these preferences are verbally communicated to the TAPs, and if they have time, they try to satisfy them by fine-tuning the plan through *constrained-moves*. There are other more global preferences related to solution quality that were not formally encoded and were left up to the TAPs to satisfy. For example, it is desirable to minimize the number of calibrations in the plan.

In recent basic research (Khatib et al. 2003, Morris et al. 2004), the basic STN model has been extended to incorporate temporal preferences and optimization strategies. We are incorporating these preference-optimization methods into our research version of MAPGEN and plan to employ them for a number of purposes. One simple use is to apply the optimization, as a postprocess, to the family of solutions represented by a flexible MAPGEN plan in order to display the most preferred solution to the user. We have extended the research version of the constraint editor to allow specifying temporal preferences on an activity's start or end time, as well as on distances between start/end time points of two activities. In particular, we have enhanced the constraint-editor tool to allow specification of a *sweet spot* in addition to a base constraint. The sweet spot is an interval of maximum preference, and outside the interval, the preference drops linearly from its maximum value. For example, it may be scientifically valid to perform a MiniTES observation between 10:00 and 15:30, but the sweet spot might be between 12:00 and 14:00. This format can also express the following types of preferences: as close to noon as possible, and as late (or early) as possible, within the hard constraint interval.

The TAP's awareness of the planning process is primarily communicated through what is shown in the plan window and the hopper; the TAP can also view the science constraints in the constraint editor. One major difference between the plan displayed to the TAP and MAPGEN's internal plan is that only a grounded plan with fixed start times can be displayed, but the internal plan is a flexible plan, representing a family of solutions. When the TAP performs a *constrained-move* of an activity, the interval bounds on that activity's start time are displayed, thus shedding a little light on the underlying flexible plan. In addition to being largely unaware of the plan's flexibility, the TAP is also largely unaware of the ordering constraints that the planner has imposed in order to satisfy the mutual-exclusion flight rules. These constraints are not visible; however, the TAP can discover their impact through *constrained-moves*. If the TAP is moving an activity that has been ordered (by the planner) to precede another activity, then once the moving activity bumps into the second activity, it will be pushed ahead of the moved activity. Designing an effective graphical display for complex constraint networks or complex flexible plans remains a challenging problem because of the need to avoid cluttering the display.

As mentioned in the discussion of the task issue, MAPGEN never solicits user assistance while it is planning; hence, the internal state of the planning process is intentionally kept hidden from the user. When failures occur, the planner notifies the user of the failure, but the notification does not include an explanation of the failure. The clearest lesson we have learned from MER mission operations 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. Such a facility would have greatly helped during training, in addition to increasing the TAPs' effectiveness during operations. However, with experience, some of the TAPs developed an impressive facility for intuiting the reasons behind planning failures.

The majority of failures are due to a temporal inconsistency in the planner's constraint network. When a failure of this kind occurs, MAPGEN extracts a temporal *nogood*, or minimally inconsistent set of constraints, which may be regarded as a low-level "explanation" of the failure. However, such nogoods are complicated and often contain hundreds of constraints, making them of little use to a time-pressured TAP. In our ongoing research effort, we have developed algorithms to generate concise, understandable explanations of temporal inconsistencies and to generate recommendations on how the user can resolve the inconsistency. We defer the details to a future article, but the general principles may be of interest here.

The basic issue from the awareness point of view is that the TAPs need to be informed of gaps in their knowledge of the unfolding plan that are relevant to the inconsistency while being protected from myriad unimportant details about the inconsistency. Thus, the system does not need to "overexplain" parts that are more easily grasped at an abstract level, and the system does not need to tell them what they already know. We have focused on the prototypical case of bringing a new activity into the plan where its science constraints are inconsistent with the STN constraints in the existing plan. The original nogood extracted from the STN is large because it includes many low-level elements resulting from the decomposition of higher-level elements. Thus, the first step is to compress the nogood by aggregating lower-level elements into more meaningful units; in the context of an STN nogood, there is a natural quantitative way of doing this. The second step breaks the STN nogood, which is a temporal cycle, into more easily grasped conflicting chains of "new" and "old" constraints. The third step is to segregate the information that is already known to the TAP from the information that is novel, and emphasize the latter. It may also be useful to provide reminders of the most salient known information and make the rest available upon request for more details.

The recommendation procedure translates an explanation-derived remedy into specific operations that are available to the user. This recommendation is not guaranteed to succeed; it may fail due to a new conflict outside the scope of the original nogood. In that case, a new explanation and recommendation will be generated. Moreover, even from a failed recommendation, the user may gain a greater understanding of the underlying issues and make some progress toward a solution.

Concluding Remarks

The articles in this special issue are diverse along many dimensions, including the scope of the mixed-initiative system, the overall style of collaboration, the motivation for employing a mixed-initiative approach, the underlying problem-solving methods, and the intended user communities. In fact, the degree of diversity makes detailed comparisons difficult. The work of George Ferguson and James Allen (2007) is much more motivated by human cognition and dialog than our work. As one of the consequences of this difference, the style of collaboration is much more flexible in their system than in MAPGEN. The scope of their work, as well as the work of Karen Myers et al. (2007), is much broader than our scope.

The underlying method of problem solving is one of the primary differences between our work, that of Michael T. Cox and Chen Zhang (2007), and that of William Cheetham and Kai Goebel (2007). MAPGEN employs an interval-based, constraint-reasoning engine that integrates planning and scheduling. In contrast, the Prodigy system used in the evaluation carried out by Michael Cox and Chen Zhang is a nonlinear, state-space planner, and the STC system described by William Cheetham and Kai Goebel employs case-based reasoning.

In terms of the style of collaboration, our work is closest to that of Karen Myers and her colleagues, which is described as being primarily a delegative model of interaction with some proactive system behavior. In their PExA system, the user allocates tasks to be carried out and can determine the scope of autonomy the system has in carrying them out. This is very similar to the control a MAPGEN user has when invoking the planning services. PExA's proactive behavior includes evaluating scheduling constraints and resource availability and alerting the users to conflicts. This is analogous to MAPGEN's active enforcement of flight rules.

One of the differences in overall interaction style is that there is a tighter collaboration between the tactical activity planner and MAPGEN than there is between the PExA system and its user. PExA and the user jointly solve problems but do so more independently, touching base when necessary. Another is the temporal scope of the collaboration. The PExA system's interaction with its users never really ends; it is meant to become part of the daily fabric of the office. In contrast, the TAP–MAPGEN collaboration takes place within a short, time-pressured, single session, and there is no system memory that persists between sessions.

MAPGEN has demonstrated that automated reasoning techniques can be combined with human knowledge and insight in a way that greatly benefits space mission operations. Discussions with mission operators suggest that MAPGEN has raised the bar on what will be expected from ground tools in future missions. 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.

Over the first months of operations, MAPGEN changed the way TAPs approach the planning process. With the added efficiency resulting from the mixed-initiative approach, they have 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 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.

It seems clear there is no "one size fits all" style of mixed-initiative planning; each application is unique in the blend of approaches that are appropriate. This article has presented a description of the MAPGEN system employed in the MER mission and discussed how it handles three of the mixed-initiative issues raised by the editors: the *task* issue, the *control* issue, and the *awareness* issue. We also expanded the discussion to include post-MAPGEN developments, which were stimulated by lessons learned from the mission experience. We hope this discussion will be helpful in framing the issues that mixed-initiative planning systems need to grapple with in order to enhance their relevance to real-world (or *other*-world!) tasks.

Acknowledgements

We would like to acknowledge the many contributions of the rest of the MAPGEN team: Mitch Ai-Chang (test lead), Len Charest (integration lead), Brian Chafin (APGEN cognizant engineer), Adam Chase (APGEN team), Jimi Crawford (deputy), Will Edgington (development team), Kim Farrell (deputy), Jennifer Hsu (test team), Ari Jónsson (development lead), Bob Kanefsky (CE and test team), Adans Ko (multimission liaison), Pierre Maldague (APGEN team), Kanna Rajan (planner cognizant engineer), Richard Springer (APGEN team), and Jeffrey Yglesias (test team). We would like to give an additional acknowledgement to Kanna Rajan and Ari Jónsson who, as coauthors on previous articles about MAPGEN, helped inform the writing of this article.

Second, we would like to acknowledge the M-SLICE team, and in particular those who influenced the planning facilities: Andrew Bachmann, Kevin Greene, and Michael McCurdy. Thirdly, we would like to thank our research collaborators for helpful discussions: Ari Jónsson, Lina Khatib, Conor McGann, Karen Myers, and Michael Wolverton. Lastly, we'd like to thank NASA's Intelligent Systems Program and the MER Mission Project for their support.

Note

1. However, in the current hyperextended operations phase, now that the planning process is more scripted and restricted, the *plan-all* operation is being applied more frequently.

References

Bresina, J. L., and Morris, P. 2006. Mission Operations Planning: Beyond MAPGEN. In *Proceedings of the Second IEEE International Conference on Space Mission Challenges for Information Technology*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Bresina, J. L.; Jónsson, A.; Morris, P.; and Rajan, K. 2005a. Activity Planning for the Mars Exploration Rovers. In *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling*, 40–49. Menlo Park, CA: AAAI Press.

Bresina, J. L.; Jónsson, A.; Morris, P.; and Rajan, K. 2005b. Mixed-Initiative Planning in MAPGEN: Capabilities and Shortcomings. Paper presented at the ICAPS-05 Workshop on Mixed-Initiative Planning and Scheduling, Monterey, CA, June 5–10.

Cheetham, W., and Goebel, K. 2007. Appliance Call-Center: A Successful Mixed-Initiative Case Study. *AI Magazine* 28(2).

Cormen, T.; Lieserson, C.; and Rivest, R. 1990. *Introduction to Algorithms*. New York: McGraw-Hill.

Cox, M. T., and Zhang, C. 2007. Mixed-Initiative Goal Manipulation. *AI Magazine* 28(2).

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal Constraint Networks. *Artificial Intelligence* 49(1–3): 61– 95.

Ferguson, G., and Allen, J. 2007. Mixed-Initiative Systems for Collaborative Problem Solving. *AI Magazine* 28(2).

Frank, J., and Jónsson, A. 2003. Constraint-Based Attribute and Interval Planning. *Journal of Constraints* 8(4): 339–364.

Jónsson, A.; Morris, P.; Muscettola, N.; and Rajan, K. 1999. Next Generation Remote Agent Planner. *Proceedings of the Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space* (ISAIRAS '99), 363. Paris: European Space Agency.

Khatib, L.; Morris, P.; Morris, R.; and Venable, B. 2003, Tractable Pareto Optimization of Temporal Preferences. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*. Detroit, MI: International Joint Conferences on Artificial Intelligence.

Maldague, P.; Ko, A.; Page, D.; and Starbird, T. 1998. APGEN: A Multi-Mission Semi-Automated Planning Tool. Paper presented at the First International NASA Workshop on Planning and Scheduling, October 28–30, Oxnard, CA.

Morris, P.; Morris, R.; Khatib, L.; Ramakrishnan, S.; and Bachmann, A. 2004. Strategies for Global Optimization of Temporal Preferences. In *Tenth International Conference on Principles and Practices of Constraint Programming (CP-2004)*, Lecture Notes in Computer Science, Volume 3258. Berlin: Springer.

Muscettola, N.; Nayak, P.; Pell, B.; and Williams, B. 1998. Remote Agent: To Boldly Go Where No AI System Has Gone Before. *Artificial Intelligence* 103(1/2): 5–48.

Myers, K.; Berry, P.; Blythe, J.; Conly, K.; Gervasio, M.; McGuinness, D.; Morley, D.; Pfeffer, A.; Pollack, M.; and Tambe, M. 2007. An Intelligent Personal Assistant for Task and Time Management. *AI Magazine* 28(2).

John L. Bresina is a senior research scientist at NASA Ames Research Center with over 25 years of research and development experience in planning, scheduling, and execution and 18 years of NASA experience. Bresina was deputy lead (and is current lead) of the MAPGEN project for the MER mission and was a full-time member of MER operations staff as tactical activity planner. In addition to the MER mission, Bresina is currently contributing to the Mars Science Laboratory (MSL), Phoenix, and Lunar Crater Observation and Sensing Satellite (LCROSS) missions. Bresina earned his Ph.D. in computer science from Rutgers University.

Paul Morris is a senior scientist at NASA Ames Research Center. After undergraduate work at University College, Cork, Ireland, he received Ph.D.s in mathematics and computer science from the University of California, Irvine. At NASA, he joined a team working on the Remote Agent Experiment, where an onboard AI system controlled the Deep Space I spacecraft for several days. He has also worked on ground planning systems for the Mars Exploration Rovers and other missions. His research interests include planning, temporal reasoning, and constraint satisfaction.