Mixed-Initiative Constraint-Based Activity Planning for Mars Exploration Rovers

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

Abstract

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

1 Introduction

The Mars Exploration Rovers (MER) mission is one of NASA's most ambitious science missions to date. The rovers were launched in the summer of 2003 with each rover carrying a rich suite of instruments to conduct remote and in-situ observations to elucidate the planet's past climate, water activity, and habitability. Among the scientific objectives of the MER Mission are to: i) determine the aqueous, climatic, and geologic history of a site on Mars where conditions may have been favorable to the preservation of evidence of pre-biotic or biotic processes ii) to identify hydrologic, hydrothermal, and other processes that have operated at the landing site iii) to identify and investigate Martian rocks and soils that have the highest possible chance of preserving evidence of ancient environmental conditions and possible pre-biotic or biotic activity and iv) to respond to other discoveries associated with rover-based exploration. Science is the primary driver of MER and, as a consequence, making best use of the suite of scientific instruments onboard the rovers within the restrictive bounds of the resources available is essential.

On MER, the tactical commanding process has been designed to command the two rovers every day, requiring that new activity plans be generated daily within a very narrow time window. This, combined with the complexity of the MER rovers and the demand for high science return, places a great burden on the Tactical Activity Planners or TAPs, who are responsible for generating the daily activity plans. In order to enable these human planners to effectively perform their job under these circumstances, and to optimize the quantity and quality of science, the MER project chose MAPGEN as a critical part of the mission operations software.

MAPGEN is a tool for science activity planning. The primary users are the MER mission tactical planners who manipulate plans to best achieve science objectives in concert with specific engineering requirements. Thus, MAPGEN assists the tactical planners in building a complex yet safe activity plan that achieves as much as possible of the science objectives for each command cycle. Among the high-level capabilities are:

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

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

2 Rover and model

Picture of rover courtesy JPL

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

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

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

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

As part of daily planning, the scientists may request “observations,” which consist of coordinated activities involving the instruments. These have to be integrated with required engineering activities, such as communication sessions.

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

3 Requirements

The requirements were initially presented in broad general terms. The detailed requirements were subsequently developed and refined during an evolutionary process that involved extensive feedback from the users. Consequently, the following overview of requirements does not reflect the effort that went into development, prototyping and requirements analysis, but only highlights some of the requirements that contributed to the final version of the system.

One of the primary requirements was that the planning and constraint reasoning capabilities had to be grafted onto an existing well-established multi-mission manual planning tool called APGEN (Maldague et al. 1997), using its graphic user interface as a front end for user interactions, plan display, plan editing and so forth. This meant that the underlying planning component had to handle all normal APGEN functions. In addition, MAPGEN would add capabilities, in the form of additional menu items, and added interactive functionality, as needed to give users access to the automated reasoning capabilities.

The basic planning and constraint reasoning capabilities were to be provided by another existing system called Europa (Jonsson et al. 2000).

Both of these systems already had databases for representing actions and plans. Given the existence of these legacy systems, and the limited time available for development, it was decided that it would be impractical to fully integrate the databases. Instead, MAPGEN was required to provide a mapping between the two databases, and to continually maintain the two databases so that they were in synchronization. This requirement was made more difficult by a mismatch in representational capabilities between the databases: APGEN utilizes a fixed schedule for activities, whereas Europa maintains a flexible one.

MAPGEN needed to provide the user with access to automated planning, but in such a fashion that the users could control the level of automation and focus it on specific parts of the plan. This led to capabilities ranging from users specifying the desired placement for activities to fully automatic plan completion.

MAPGEN was asked to provide a variety of ways in which the user could modify plans. At one end of the spectrum, there was the desire to move a single activity and have related activities in the plan move in concert. At the other end was the need to be able to temporarily “turn off” automatic flight rule enforcement so that any activity could be moved anywhere, and that subsequently the enforcement could be turned on again and the plan automatically readjusted to satisfy flight rules.

The MAPGEN tool also had to automatically handle the creation of certain activities that were needed to satisfy flight rules. Among these were the CPU on/off commands, so that the computer would be turned on only when needed.

MAPGEN was asked to provide a variety of other capabilities to assist the users in small ways, taking advantage of the automated reasoning capabilities.

3 System design

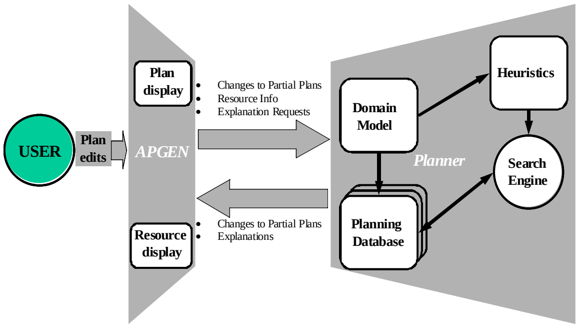

Figure 1: System architecture.

The MAPGEN system includes two modules based on pre-existing systems: a plan display and editing tool called APGEN, and an underlying constraint and plan-reasoning engine called EUROPA. The two are linked by a third module that handles the interface between the two main components, and presents the capabilities of the autonomous reasoning system in a packaged form available from menu items. Figure 1 shows the overall structure of the combination. A fourth standalone part of the system is an external constraint editor, which provides constraint manipulation services that could not reasonably be added to the APGEN user interface.

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

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

Designing the interface was one of the key challenges in making MAPGEN a useful interactive tool. Simply offering the user access to fully automated planning and then presenting them with the results is of little use in activity planning for the MER mission. As a consequence, the interface was constructed with a more collaborative style in mind. The following is a sampling of the interface functionality:

- Updating EUROPA plan database in response to user changes made via the front end
- Updating the APGEN plan based on results from automated reasoning
- Supporting targeted plan completion, where the user selects the goals to be achieved
- Various advanced plan editing capabilities such as swapping the order of activities and automatically reestablishing flight rule enforcement

Updating the APGEN plan database in response to automated reasoning turned out to be a key challenge.

Later in this paper, we cover one aspect in detail: the handling of activity placement in time.

Special considerations applied to the constraint editor, which is part of the MAPGEN package. The APGEN graphic user interface is based on a timeline display. This is not suitable for constraint entry and editing, which require a more specialized interface. This led us to implement the constraint editor as a separate standalone tool. The constraint editor allows users to specify temporal constraints on sets of activities. The constraints are written into a file that is then imported into MAPGEN. Having the constraint editor as a separate tool causes some difficulties, such as users not getting feedback on the effect of constraints on the current plan. Consequently, a future goal is to incorporate the constraint editing capabilities directly into MAPGEN.

4 Constraint-based planning

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

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

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

5 Preferred time placement

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

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

One advantage of preserving a flexible set of solutions is that the Planner may adapt to additional constraints by exploiting the flexibility, rather than completely re-solving the problem. However, this has to be reconciled with APGEN, which expects to see a fixed time schedule. Also, many tools associated with APGEN, such as those that do calculations of resource usage, require a fixed schedule of activities. Apart from these pragmatic considerations, direct presentation of temporal flexibility to a plan GUI in a way that is not confusing poses significant problems: it is difficult to provide a visual representation of flexibility and temporal relations between activities without obscuring the display.

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

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

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

There are also preferences that arise from varied sources. Some are based on engineering or scientific considerations such as desiring calibrations to be close to measurements, or wanting separate observations to occur in similar lighting conditions. Many are derived from the need to solve problems related to resources. In general, the tweaking process is driven by a desire to fit as much "science" as possible into the plan, while steering it on a course that avoids running aground on competing resource limitations. While the planner takes care of scheduling restrictions, the responsibility for making sure that energy and data resources are respected falls on the human operator. For numerical resource issues, the operator has a range of possible corrective actions, many of which require human-level scientific judgment.

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

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

Assume that a plan has been produced, and no preferences have yet been expressed to modify the solution. The initial solution places each activity as close as possible to its original time while satisfying the constraints. During subsequent tweaking, MAPGEN provides a GUI feature called a *constrained move* that allows dragging an activity to a new location. When the mouse button is released, other activities are also moved to maintain the integrity of the constraints. For example, the moved activity may "push" other activities ahead of it because of precedence constraints established by the user or planner.

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

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

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

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

```
  add a position constraint setting x
  to the upper bound
Propagate the effect of the new
constraint
```

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

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

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

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

6 Selective Planning

The needs of the MER mission preclude a fully automatic planner. Every day, the planetary scientists meet to consider the previous downlink data and discuss what observations should be planned for the upcoming day. Because resource availability and need is only known to a rough approximation at this stage, the scientists are encouraged to oversubscribe *vis a vis* the resource profiles. However, this means that lower-priority activities may need to be dropped during detailed planning. The task of squeezing in as many meaningful observations as possible is the province of the human operator of the MAPGEN tool. Because human knowledge and judgment is interleaved with autonomous reasoning, a more fine-grained approach to planning is needed than the usual batch-style fully automatic planner.

The approach we took was to provide a “holding area,” called the *hopper* for observations that have not yet been made part of the plan. Constraints on activities in the hopper are held in abeyance. When activities in the hopper are selected for planning, their constraints are activated, and the planner attempts to fit them into the plan. Any activities that the planner is unable to fit into the plan are returned to the hopper and their constraints are once again deactivated. For activities that did not fit, the operator has the option of trying again after taking some steps that may ameliorate the situation. For example, it is possible to modify parameters that affect the duration, or move around activities that are already in the plan, to help make room. The operator may also move some less-desired activity that is already in the plan to the hopper.

For this style of interactive planning, performance is crucial. In general, each planning episode should not take more than a few minutes. To ensure this, the planning is subject to a timeout, after which the planner does a cleanup and moves items that are not fully planned back to the hopper. A second mechanism that limits the planning time is a process that “charges” the backtracking cost to top-level goals according to a certain policy. Thus, at some point, the most troublesome goals can be eliminated and returned to the hopper.

In order to make sure that the most important items are planned, the operator and the planner can use priorities that are assigned by the scientists to influence the order in which goals are selected and planned.

7 Challenges

The principal challenge to the acceptance of generative planners such as MAPGEN, in the operations environment, has been the perception that planning algorithms have non-deterministic outcomes. As a result, the planning component of the DS1 Remote Agent (Muscettola et al. 1998) was viewed with some suspicion. However, with the success of the RA onboard, and the urgent need in the MER mission for a ground-based decision-support system, the principal challenge for MAPGEN related to software engineering and quantifiable metrics for demonstrating the utility of using an automated planner in the critical uplink process. Both these challenges turned out to have a sizable impact on the infusion and deployment of the MAPGEN system.

The MER mission's requirement to have the entire tactical command cycle take place within 19 hours brought a unique set of challenges. The foremost of these was to ensure that the tools used to build the daily command load for the rovers were not only inter-operable but were also robust enough to generate products within strict deadlines in a serial pipeline. Planner performance was initially a cause for concern; this was alleviated by changes in the uplink process and the level of detail in the plan, as well as improved hardware platforms. Furthermore, over the course of the development cycle, the serial process was streamlined with a substantial impact to the design of MAPGEN's interfaces. One of those changes resulted in the inclusion of a separate tool called the Constraint Editor (CE), the purpose of which was to allow scientific intent to be crystallized as constraints in a machine-readable form so they can be enforced by MAPGEN.

The second major challenge was to convince operations personnel of the utility of doing activity planning in a substantially new way using an automated planner. It had been previously believed that automation, while helpful, could result in undue complexity that would require a large learning curve for MAPGEN operators and substantial investment in infrastructure to support the use of the CE and MAPGEN during actual surface operations. This issue was brought sharply into focus in October 2003, when the manual approach to doing Activity Planning was compared side by side with a process using the automated MAPGEN approach. The resulting 20% increase in activities planned and the efficiency with which MAPGEN enabled the human operators to accomplish it, was a clear-cut result that the MER operations community was able to embrace. Subsequently MAPGEN was deemed mission-critical and baselined for the uplink commanding process for the two rovers.

At this time, the operators of MAPGEN and CE have commanded the two rovers on the surface of Mars for about 100 Martian Sols without turning off the generative planning capability. Furthermore, current estimates of MER operations personnel suggest that upwards of 25% added science has resulted from the use of MAPGEN.

8 Concluding Remarks

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

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

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

References

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

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

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

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

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

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

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