ABSTRACT
This paper presents two models of goal generation which enable a motivated autonomous agent to generate goals in response to changes in its underlying drives or motivations, while it is both planning and executing. A Mars rover domain is used to illustrate the two models: the first model involves goals being generated explicitly in response to changes to the agent’s motivations where such goals are then provided to a planner, while the second model implicitly generates goals and the goals that they may generate as part of the planner’s domain model. Results from experiments on integrating the models with different planners suggest that while they may bring the benefits of autonomy that we seek, they also introduce more complexity into the planning problem.

1. INTRODUCTION
AI planning traditionally assumes that all goals are known in advance and externally imposed, and that the planning process is finished once a plan has been generated in which all goals are achieved. This is typically not directly connected to executives — an external user supplies problems to be solved, where a problem consists of a model of the agent’s current state, the goals to be achieved, and a representation of the activities the agent is capable of carrying out. Once a plan has been generated, its execution is no longer the concern of the planner. Exceptions to this include the work of Knight et al. [5], which relaxes these assumptions by introducing continuous planning in Casper, an architecture for autonomous planning and control, intended for application in space missions involving onboard autonomy.

This paper aims to investigate the role that an agent’s context may play when it generates goals and plans, where context is defined as the agent’s model of its current and predicted future states of the environment, and its desires and preferences, represented by modelling the agent’s motivations. While several models of motivation have been developed, they have typically been abstract or implemented in simple scenarios with arbitrary architectures. Our work is concerned with the integration of a motivational component with the planning and execution of actions in an autonomous agent, at a general level, for many planners and architectures. More specifically, the key contribution of this paper is in the development of a mechanism for autonomous goal generation that has been integrated with several different planners.

Motivations have been defined by Kunda [6] as “any wish, desire, or preference that concerns the outcome of a given reasoning task”. Many researchers in the fields of psychology, philosophy and AI have been interested in the way motivations affect the reasoning process [1, 4] and have attempted to develop taxonomies of motivations (e.g. [7, 4]). In [3], motivations were modelled as a set of tuples: motivation == (name, value, importance) where name is a unique identifier associated with the motivation, value is a measure of the current value (or strength) associated with that motivation (its motivational value), and importance is a measure of how important that motivation is to the agent. This simple but effective representation of motivations is also used in the work described here: an agent has a set of different motivations that affect its reasoning process in various ways, and the agent acts to keep the value of each of its motivations within certain limits. Associated with each motivation is a goal that is triggered or generated when the motivational value crosses a predefined threshold. Achieving this goal restores the motivational value to an acceptable level.

Here, when an agent invokes an AI planner, its motivations may influence the planning process through the insertion of newly generated goals. Two models of goal generation are presented. In the first model, the motivation and goal generation component are decoupled from the AI planner. Motivations explicitly generate goals in response to changes in motivational values, and such goals are then posed to a planner. In the second model, motivations are encoded within the planner’s domain model, and the planner must implicitly generate goals in order to ensure the motivational values remain within certain bounds. Both models are described in the next section in the context of the MADbot architecture and a Mars rover domain.

2. THE MADBOT AND MARS ROVERS
The MADbot system (Motivated And Goal Directed Robot) developed by Long [2] is a continuous, motivated, autonomous planning and execution system which is capable of generating its own goals in response to changes in its low-level drives or motivations.

The system contains several motivations, each of which monitors one or more associated state variables which model the agent’s internal state. As the agent executes its plan, the value of each state variable changes which in turn triggers changes to the values of the motivations. When certain constraints are met, changes to the value of a motivation cause one or more goals to be generated. The architecture contains a Motivation component which is responsible for updating the value of each motivation in response to changes to the agent’s state variables, and for generating new
goals whenever the values associated with the motivations satisfy certain constraints. Goals may also be passed to the system by an external source. Once generated, goals are passed to a Planner, which generates a plan to achieve the new goals. Each action of the resulting plan is sent in turn to be executed by the agent. Execution results in changes to the values of the state variables which model the agent’s internal state, such as the location of the agent and its internal battery charge. Such changes will in turn cause changes to the motivations causing further goals to be generated.

Figure 1 shows a simple Mars rover domain which was inspired by recent Mars Exploration Rover missions and consists of a rover, or madbot, initially located at waypoint 1, a rectangular area containing a number of obstacles (in the figure these are depicted as hatched shapes), a lander, two recharge points at waypoint 1 and waypoint 5 respectively, and a relocation point at waypoint 3 at which the rover can relocalise. The idea is that the rover is able to explore this environment by following a series of connected waypoints. The domain map may contain a few known locations that may be of interest to an external source such as a science team, and so the rover’s initial goal (externally posed) may be to explore this environment by following a series of connected waypoints. Each action of the plan shows the time at which execution should commence as well as the name and parameters of the action. The parameters w1...w8 represent waypoint 1, waypoint 8 respectively, o2 and o3 represent objective 2 and objective 3, while c1 represents the camera available onboard the rover. The numbers in square brackets indicate the degree to which the action affects the underlying state variables associated with the motivations: the first indicates the amount of battery charge consumed by the action which affects the motivation conserve-energy, (the action navigate madbot w1 w2) consumes 8 units; the second indicates the duration associated with the action which affects the motivation acquire-data; the third indicates the amount of storage space required which affects the motivation communicate-data; the fourth indicates the distance travelled as a consequence of executing the action.

In our example, the first goal to be generated involves obtaining a panoramic image, (have-panoramic-image madbot). As shown in Table 1, the motivational value of acquire-data is determined by value of the state variable which records the amount of time which has passed since the rover last obtained a panoramic image. This value is initially set to 0, while the motivation conserve-energy has a threshold of 100, so that after the rover has finished executing the action the motivational value (have-panoramic-image madbot w4 o2 c1) of the plan above, the value (have-panoramic-image madbot w5 w3) will exceed this threshold causing a goal to obtain a panoramic image to be generated, (have-panoramic-image madbot).

The newly generated goal (have-panoramic-image madbot) as well as the remaining high-level goal (have-image madbot o3) are then sent to the Planner together with a representation of the current state (i.e. the state that results after the rover has executed the first 6 actions of the plan shown above in which the rover is now at waypoint 4). The Planner generates the following new plan which achieves the two goals.

### Table 1: Mars rover motivations, their associated state variables, and the goals they trigger

<table>
<thead>
<tr>
<th>Motivation</th>
<th>State variable</th>
<th>Replenishment goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conserve-energy</td>
<td>battery charge</td>
<td>recharge battery</td>
</tr>
<tr>
<td>Acquire-data</td>
<td>transmit image data</td>
<td>(take-image madbot w4 o2 c1)</td>
</tr>
<tr>
<td>Communicate-data</td>
<td>transmits image data</td>
<td>(communicate-image-data madbot)</td>
</tr>
<tr>
<td>Conserve-memory</td>
<td>memory usage</td>
<td>(acquire-memory madbot)</td>
</tr>
</tbody>
</table>

## 3. MOTIVATED GOAL GENERATION

In this section, we describe two alternative mechanisms for goal generation, the first in which the mechanism is explicit and external to the planner, and the second in which the mechanism is implicit and handled by the planner itself. In the first model, goals are generated explicitly in response to changes that occur to the agent’s motivations. Initially, two goals are presented to the system by an external source, which involve obtaining images of objective 2 and objective 3, e.g. (have-image madbot o2) and (have-image madbot o3). These goals are sent to the Planner, which generates the following plan using a PDDL2.1 level 3 domain model.

Each line of the plan above, the motivational value of acquire-data is determined by value of the state variable which records the amount of time which has passed since the rover last obtained a panoramic image. This value is initially set to 0, while the motivation conserve-energy has a threshold of 100, so that after the rover has finished executing the action (have-panoramic-image madbot w4 o2 c1) of the plan above, the motivational value (have-panoramic-image madbot w5 w3) will exceed this threshold causing a goal to obtain a panoramic image to be generated, (have-panoramic-image madbot).

The newly generated goal (have-panoramic-image madbot) as well as the remaining high-level goal (have-image madbot o3) are then sent to the Planner together with a representation of the current state (i.e. the state that results after the rover has executed the first 6 actions of the plan shown above in which the rover is now at waypoint 4). The Planner generates the following new plan which achieves the two goals.

Again, each action of this new plan is sent in turn to be executed by the rover and once the rover has finished executing the first action of the new plan, (have-panoramic-image madbot), the goal triggered by the motivation acquire-data is satisfied. As a consequence, the value of the state variable which records the amount of time that has passed since the rover last acquired new data is reset.

The rover continues executing the remaining actions of second plan, and after it has finished executing the action (navigate madbot w5 w3), the motivational value of conserve-energy (which is...
directly related to the level of battery charge) will have fallen below its associated threshold, the value 40. This causes a goal to recharge to be triggered, (recharged madbot), which is presented to the Planner. A new plan is then to achieve this goal, which is passed to the rover to execute.

This process continues indefinitely because even when both high-level goals have been achieved, (have-image madbot o2) and (have-image madbot o3), the MADbot architecture reverts to motivated goal generation, so periodically, whenever their associated thresholds are crossed, the motivations will generate goals to obtain panoramic images (have-panoramic-image madbot), to recharge (recharged madbot), to relocalise (relocalised madbot) or to communicate image data, (communicated-image-data madbot lander). When deciding which goals to address, and how to incorporate these into the plan, the top level goals of the original plan under execution remain paramount – the need to respond to a motivated goal temporarily suppresses the plan to allow an originally unplanned for task to take priority.

This approach addresses the key problem of being able to direct behaviour by explicitly generating goals in response to changes to motivations in the absence of externally imposed high-level goals. An autonomous system such as a Mars rover can thus continue to perform useful tasks during periods when it is impossible for ground control to send further instructions. However, the approach suffers from a number of limitations. First, by decoupling resource consumption from planning, it is possible for the system to generate plans that violate resource constraints (e.g. it is possible for the rover to run out of battery charge, as the Planner has no model of battery charge consumption) as an indeterminate time period occurs between generating a goal and achieving that goal, and during this period the relevant resource might run out.

To address these problems, as an alternative to the explicit model of goal generation, we can also integrate the motivation model implicitly into the planning machinery. Here, we model the motivation machinery internally within the domain description used by the planner, so that the planner ensures the motivational values remain within certain bounds. As shown in Table 1 the motivations conserve-energy, acquire-data, communicate-data and relocalise are directly related to the value of their associated state variables (the level of battery charge, the amount of time which has passed since the rover last obtained a panoramic image, the amount of free data storage, and the distance travelled since the rover last recalibrated), which change as the rover acts within its environment. Because it is possible to predict the effect of each action on the values of these state variables, it is possible to model these motivations implicitly through their associated state variables and associated thresholds in terms of resource consumption in the domain model used by the planner. The thresholds associated with each motivation are modelled in the domain model as resource preconditions. Thus when the Planner generates a plan, it must ensure the motivations remain within their bounds. This method ensures the relevant resources never run out but with added costs as reasoning about resources adds to the complexity of the planning problem. In this model goals may still be imposed on the system by an external source, although the external source is the only means by which goals are inserted to the system. In the absence of externally imposed goals, the system has no means for operating autonomously.

4. CONCLUSIONS
In this paper we have described how changes to an agent’s motivations might cause new goals to be generated. Motivations provide a solution to the problem of where goals come from and thereby improve the effectiveness of remote autonomous systems such as a Mars rover which is only able to operate by communicating with ground control. We presented two approaches to motivated goal generation. The first explicitly models the agent’s motivations which change in response to changes to the agent’s state variables, and cause goals to be generated whenever such changes satisfy certain constraints. The second involves modelling motivations implicitly as resource constraints in the planner’s domain model – here it is the Planner’s responsibility to ensure that the implicit motivational values remain within predefined bounds. Experiments have shown both approaches have a number of advantages and disadvantages so our aim is to construct a hybrid approach to the model of motivations. The paper thus offers some initial solutions to the problem of incorporating autonomous control into existing agent and planning architectures. While the solutions are effective, they are still limited, and much further work remains.

5. REFERENCES

<table>
<thead>
<tr>
<th>Action</th>
<th>Supports</th>
<th>Undermines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigate</td>
<td>conserve-energy, acquire-data, relocalise</td>
<td></td>
</tr>
<tr>
<td>Calibrate</td>
<td>conserve-energy, acquire-data</td>
<td></td>
</tr>
<tr>
<td>Take-image</td>
<td>communicate-image-data, acquire-data</td>
<td></td>
</tr>
<tr>
<td>Take-panoramic-image</td>
<td>acquire-data, communicate-image-data</td>
<td></td>
</tr>
<tr>
<td>Recharge</td>
<td>conserve-energy, acquire-data</td>
<td></td>
</tr>
<tr>
<td>Communicate-data</td>
<td>communicate-image-data, acquire-data</td>
<td></td>
</tr>
<tr>
<td>Localise</td>
<td>relocalise, acquire-data</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: The Mars rover actions and the motivations they support or undermine.