International Journal on Artificial Intelligence Tools

A Motivation-based Planning and Execution Framework

ALEXANDRA M. CODDINGTON
Department of Computer & Information Sciences
University of Strathclyde
Glasgow, G1 1XH, United Kingdom
Alex.Coddington@cis.strath.ac.uk

MICHAEL LUCK
School of Electronics & Computer Science
University of Southampton
Southampton, SO17 1BJ, United Kingdom
mml@ecs.soton.ac.uk

Received 18th July 2003
Revised 2nd December 2003

AI planning systems tend to be disembodied and are not situated within the environment for which plans are generated, thus losing information concerning the interaction between the system and its environment. This paper argues that such information may potentially be valuable in constraining plan formulation, and presents both an agent- and domain-independent architecture that extends the classical AI planning framework to take into account *context*, or the interaction between an autonomous situated planning agent and its environment. The paper describes how context constrains the goals an agent might generate, enables those goals to be prioritised, and constrains plan selection.

Keywords: AI planning; agent architectures; motivations; plan evaluation

1. Introduction

Recently, the AI planning community has become increasingly interested in investigating different plan evaluation metrics to guide the search behaviour of various planning systems. This interest has been influenced by the development of PDDL2.1 [7], a planning domain definition language that was used to specify temporal planning problems for the 3rd International Planning Competition held in 2002. PDDL2.1 allows planning domains both to include actions with duration, and to represent the consumption and replenishment of resources associated with action execution using numeric-valued fluents. In previous competitions, planning domains were non-temporal, and resource consumption was not modelled, so that plan evaluation metrics were essentially based on minimising the number of actions and the number of outstanding goals. By contrast, modelling time and resources allows metrics to be developed that examine plan makespan as well as resource consumption profiles.

The three International Planning Competitions (held in 1998, 2000 and 2002) have encouraged AI planning research to focus on generating good quality plans efficiently. However, such planners require goals to be independently posed by an external agent, and there is no information available as to the circumstances that caused those goals to be generated. In addition, these planners are disembodied and not situated within the environment for which plans are generated, thus losing further potential information concerning the interaction between the planning system and the environment for which it is planning. In this paper, we argue that such information may potentially be very valuable in constraining plan formulation, and present an agent-independent and domain-independent architecture that extends the classical AI planning framework to take into account *context*, or the interaction between an autonomous situated planning agent and its environment. Context is important as it constrains the goals a planning agent might generate, enables the agent to prioritise goals, and constrains plan selection. The paper describes both how context enables an agent to prioritise goals and actions (therefore allowing the agent to prefer to achieve goals of high priority in favour of those of lower priority, and to prefer to execute actions of high priority in favour of those of lower priority), and how aspects of context may be encapsulated in a plan evaluation metric to direct the search behaviour of a situated planning agent. In Section 2 we describe context in more detail and present motivations. We demonstrate how motivations enable goals to be generated, how such goals can be prioritised, and how motivations may play a role in plan evaluation. Section 3 presents a continuous planning/execution framework, discusses the impact of being able to assign priorities to goals and actions, and describes the way plans can support or undermine an agent's motivations. Finally, in Section 4 we present conclusions and future work.

2. The Use of Context in Planning

Human, real-world problem-solving involves a degree of subjectivity that has led researchers such as Picard [23] to investigate the impact of emotions on decision-making. A key contribution of this paper is to examine how subjectivity, captured by modelling the context of a planning agent, might affect its plan-generation capabilities. Now, the context of an autonomous planning agent, captured partly by representing and reasoning about the motivations of the agent, is important in three ways: it constrains the goals that the agent might generate; it enables the agent to prioritise those goals by allocating its resources accordingly; and it enables the agent to select plans. We define context to be composed of the following aspects.

- The agent's capabilities which, in AI planning, are represented by action instances reflecting the physical actions the agent is able to perform.
- The environment in which the agent is situated — this includes the current state of the environment (both internal and external to the agent) as perceived by the agent, as well as predicted future states of the environment that arise by executing the actions in the agent's plan.

- The agent's desires or preferences, which are captured by modelling its *motivations*. A *motivation* is any desire or preference that affects the outcome of a given reasoning task [11]. Motivations will be described further in the following Section.

  Of course, what is contextual depends upon an agent's particular sensors, actuators, size and so on. For example, different features within the same environment may be pertinent to different agents; a human in an office environment takes note of objects such as desks and computers, while an insect in the same environment is perhaps more aware of grains of dust on the carpet. With reference to artificial agents, it is very much a responsibility of the designer to take into account such factors, as well as the agent's purpose, in order to determine the appropriate context.

2.1. *Motivations*

When planning to achieve the same goal, two agents may create different plans even though their external environment is the same. The different plans arise as a result of differences in the internal states of those agents and can be said to be due to the *motivations* of each agent. According to Halliday, the word motivation does not refer to a specific set of readily identified processes [8] — it is frequently discussed in terms of drive and incentive. Drives are related to physiological states such as the deprivation of food, hormones, etc, while incentives refer to external stimuli that affect motivation such as the presence of food, as an incentive to eat. Research on motivation is currently being pursued from a variety of perspectives including psychology and ethology.

Some psychological research has recognised the role of motivations in reasoning in a similar way to that suggested here. Kunda [11] informally defines motivation to be, "any wish, desire, or preference that concerns the outcome of a given reasoning task" and suggests that motivation affects reasoning in a variety of ways including the accessing, constructing and evaluating of beliefs and evidence, as well as decision making. Such arguments are supported by a large body of experimental research, but no attempt is made to address the issue of how motivations may be represented or applied in a computational context.

Computational work has also recognised the role of motivations. Simon [24] takes motivation to be "that which controls attention at any given time," and explores the relation of motivation to information-processing behaviour, but from a cognitive perspective. Sloman [26,25] has elaborated on Simon's work, showing how motivations are relevant to emotions and the development of a computational theory of mind.

Problem-solving can be considered to be the task of finding actions that achieve the current goals. Typically, goals are presented to systems without regard to the problem-solving agent so that the reasoning process is divorced from the reality of an agent in the world. Clearly, this is inadequate for research concentrating on modelling autonomous agents and creatures, which requires an understanding of how such goals are generated and selected. Additionally, it is inadequate for research that aims to provide flexibility of reasoning in a variety of contexts, regardless of concerns with modelling artificial agents. Such flexibility can be achieved through the use of motivations which can lead to different results even when goals remain the same [15].

In proposing to develop a 'computational architecture of a mind', Sloman makes explicit mention of the need for a "store of 'springs of action' (motives)" [26]. In the same paper, he tries to explicate his notion of a motive as being a representation used in deciding what to do, including desires, wishes, tastes, preferences and ideals. The key feature of a motive, according to Sloman, is not in the representation itself, but its role in processing. Importantly, Sloman distinguishes between motives on the one hand, and 'mere subgoals' on the other. "Sometimes," he claims, "a mere subgoal comes to be valued as an end," because of a loss of 'reason' information. First-order motives directly specify goals, while second-order motives generate new motives or resolve conflicts between competing motives — they are termed motive generators and motive comparators. "A motive produced by a motive generator may have the status of a desire." This relatively early work presents a broad picture of a two-tiered control of behaviour: motives occupy the top level, providing the drive or urge to produce the lower level goals that specify the behaviour itself. In subsequent work, the terminology changes to distinguish between nonderivative motivators or goals and derivative motivators or goals, rather than between motivators and goals themselves. Nevertheless, the notion of derivative and nonderivative mental attitudes makes one point clear: that there are two levels of attitude, one which is in some sense innate, and which gives rise to the other which is produced as a result of the first.

In a different context, the second of Waltz's 'Eight Principles for Building an Intelligent Robot' requires the inclusion of "innate drive and evaluation systems to provide the robot with moment-to-moment guidance for its actions." [28] In elaborating this principle, Waltz explains that the action of a robot at a particular time should not just be determined by the current sensory inputs, but also the "desires" of the robot, such as minimizing energy expenditure (laziness), and maintaining battery power levels (hunger). This research into robotics, artificial life, and autonomous agents and creatures has provided the impetus for a growth of interest in modelling motivations computationally, and a number of different representations for motivations and mechanisms for manipulating them have been developed at both subsymbolic and symbolic levels (for example [14,9]).

While there has been a steady stream of research over the last twenty years, it is only relatively recently that the incorporation of motivations into effective agent architectures, for functional purposes, has become more prevalent, for example [5]. Neverthless, the varied research into robotics, artificial life, and autonomous agents and creatures has provided an impetus for a growth of interest in modelling motivations computationally, and a number of different representations for motivations and mechanisms for manipulating them have been developed at both subsymbolic and symbolic levels (for example [1,9]).

New agent architectures that draw on emotion-cognition interaction are also now being considered. For example, Oliveira and Sarmento [21] describe an architecture that includes emotion evaluation functions and emotion-based processing to bias several functional aspects of agent behaviour, including plan granularity and execution time. Other aspects that are influenced in this way relate to *mood*-based retrieval of information from memory. Marsella and Gratch have also begun to address the extension of computational models of emotion to show their impact on behaviour [16], but this work is still preliminary.

Perhaps the highest profile work that draws on the psychological literature in emotion, motivation and cognition relates to artificial, life-like or believable agents, in which the focus is on the development of models that are intended to provide accurate and engaging interaction mechanisms. For example, Lester et al. describe a full-body emotive pedagogical agent for multimodal interaction [12], while Ball and Breese consider the impact of emotion and personality on conversational interaction [2]. Such work is also relevant to the kind of model considered here, but largely ignores our core issues of planning and acting without embodiment.

In the prevailing symbolic AI view, an agent may be modelled as having a set of motivations which, in human terms, represent its basic needs and desires. Different kinds of agent have different motivations. In animals, these motivations might represent drives such as *hunger* or *curiosity*, whilst an autonomous truck-driving agent might have motivations concerned with fuel replenishment, *conserve-fuel*, or with preserving the state of the truck's tyres, *conserve-tyres*. Associated with each motivation is a measure of strength, or motivational value, which varies with changing circumstances, and which represents the driving force that directs action to satisfy the motivation. For example, the motivational value associated with *conserve-fuel* might be low just after the truck-driving agent has refuelled but will gradually increase over time as the agent drives between cities, consuming fuel. The truck-driving agent only acts to satisfy the motivation when the strength associated with it is sufficiently high.[a] Feedback of information from an agent's environment causes motivational values to vary — for example, if an agent perceives immediate danger, the motivational value associated with *self-preservation* increases to a high level causing the agent to act.

Context, especially the notion of motivation, is increasingly being seen as a means of influencing or constraining particular aspects of behaviour in autonomous agents. In planning agents, motivations are regarded as important in two particular areas of interest, goal generation and plan evaluation. Each is considered in turn

[a]This example may falsely give the impression that the changing strength associated with motivations is analogous to the consumption and replenishment of resources. There is a subtle difference however — merely modelling the consumption and replenishment of resources does not have an impact upon an agent's behaviour. In contrast, motivations affect the way an agent acts within a domain — for example, when the motivational value associated with *conserve-fuel* is low, the truck-driving agent acts to conserveits fuel by driving more carefully or by generating a goal to replenish the fuel.

below.

2.2. *Goal Generation*

An important feature of an autonomous planning agent is an ability to generate goals in order that it may further its aims, either by taking advantage of opportunities that may arise, or by preventing undesirable situations from occurring.

Motivations directly affect the generation of goals — by achieving a goal, an agent is able to mitigate the motivations that led to the generation of that goal [6]. For example, if the strength associated with *hunger* lies above some threshold, the goal of obtaining food might be generated. A plan to achieve the goal is generated and, once the agent has executed the plan and the goal is satisfied, the motivational value associated with *hunger* is reduced. It is important to distinguish between motivations and goals — whereas goals are states that an agent wishes to bring about, or maintain, motivations are preferences that drive the behaviour of agents [6]. In this sense, and similarly to Sloman's different levels, we could consider motivations to be *meta-goals*.

Norman [20] describes a model of goal generation in which the threshold causing goals to be generated to satisfy one or more motivations is dynamically adjusted in response to feedback from the planning sytem. Thus, if the planner is currently in the process of achieving many (or few) goals, the motivational value threshold causing goal generation is increased (or decreased). Similarly, Moffat & Frijda [18] use a concept which they term 'concerns', which are "dispositions to prefer certain states and/or dislike others". In their model, an agent selects the most relevant information perceived through its sensors. The relevance of an event comes from the agent's concerns. Thus, for example, if an agent detects food in its environment and if this event is relevant to its hunger concern, a goal may be generated to move towards the food and eat it. The most relevant event causes a signal to be emitted which, in turn, causes the relevant goal to be instantiated.

Goal generation is influenced by the current and predicted future states of the environment (encapsulated within an agent's plan), and the current and predicted future strength associated with an agent's motivations. Now, an agent's perception of its immediate environment may directly affect the strength associated with its motivations in such a way as to lead to the generation of goals. For example, the sudden appearance of an oncoming vehicle may cause a sudden increase in an autonomous truck-driving agent's motivation concerned with *self-preservation* which, in turn, may lead to the generation of a goal to avoid a collision. As well as being generated in response to the agent's immediate environment, goals may be generated in response to the agent's future predicted states of the environment (encapsulated its current plan). For example, if an autonomous truck has generated a sequence of actions to achieve the goal of delivering a package to a particular destination, it can predict that it will be at that destination at some time in the future, which may cause it to generate a goal of refuelling at that location. The generation of goals may also be influenced by the predicted future strength of motivations. The truck agent may predict that, as a consequence of executing a sequence of actions involving driving from one location to another, the motivation *conserve-fuel* will increase in strength. This may cause the agent to generate a goal to replenish fuel with a deadline coinciding with the point at which the motivation is predicted to reach the threshold leading to goal generation.

In addition, the relative importance of the various goals generated are directly related to the strength of an agent's motivations. If a motivation is strong (and high in relation to the goal generation threshold), any goal generated to satisfy that motivation will also be important. Changes in motivational values may also cause the priority associated with goals to change. For example, a truck-driving agent may have the goal of delivering a parcel to a client; the priority associated with the goal may change if the agent learns that the client has not paid for previous deliveries.

When executing plans that have been created to achieve one or more goals, it is possible that the context of an agent will change. For example, when a human agent generates a plan to achieve the goal of attending a conference in Florida, executing that plan means that the agent's environment and therefore its context will change. A change in context means that new goals that are appropriate to that particular context may be generated. For example, on a free day during the conference, the agent may decide to visit DisneyWorld. Another way in which a plan results in an agent's context being changed is if the agent chooses to acquire a new skill such as learning to ice skate or, if it is a robotic agent, and has been altered so that its capabilities are extended (such as being fitted with a new gripper). The new skill amounts to a change in the agent's capabilities which, in turn, change the agent's context. One consequence of changing context while executing plans is that an agent can generate plans with the purpose of changing its context — for example, by planning a vacation or by planning to acquire new skills.

To conclude, when goals are generated, the motivations that led to their generation determine their importance or priority. As a consequence of goals having an associated priority, it is possible to assign a value indicating the priority of each action that contributes towards the achievement of such goals. In Section 3.3 we discuss further how an agent is able to prioritise goals and actions on the basis of how important they are to the agent, and how this ability affects the decisions taken by an autonomous planning agent.

2.3. *Plan Evaluation*

Motivations also enable an agent to evaluate the plans generated to achieve its goals. If a human agent executes a plan that involves walking down a dark alley in order to achieve the goal of having some food, when imagining executing the plan, they might experience a small rise in their level of fear. Through being able to predict that walking down a dark alley will cause their fear to increase, they may choose an alternative plan (for example, one that involves driving to their destination). The framework presented in this paper aims to replicate this behaviour — if one sequence of actions (or plan) chosen to achieve some goal conflicts with the motivations of an agent, the agent might choose an alternative sequence of actions. This is decribed in detail in Section 3.4.

3. A Continuous Planning Framework

3.1. *The Basic Architecture*

A continuous planning framework, illustrated in Figure 1, has been designed to be both agent- and domain-independent and allows the agent to reflectively evaluate, taking into account its context, when choosing a course of action.

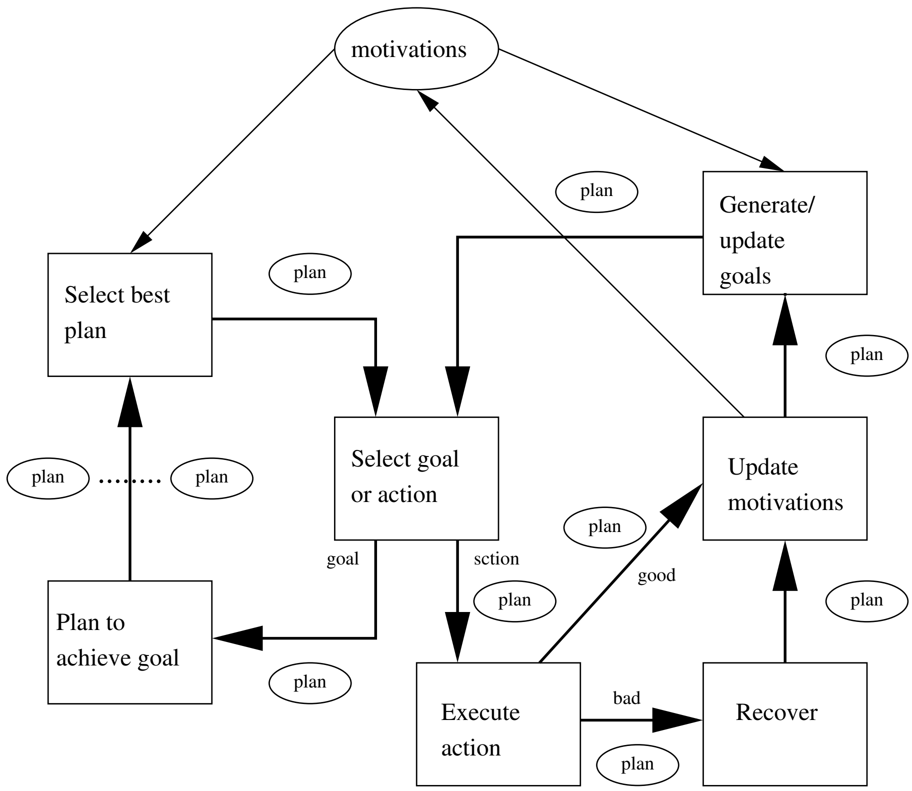

Fig. 1. The Continuous Planning/Execution Framework

Solid rectangular boxes represent the various processes in the framework that are the focus of this research. Oval boxes represent plans (including the initial and goal states), and the agent's motivations, which are represented as a set of tuples: $(name, value)$ where $name$ is the name of the motivation, and $value$ is the motivational value or strength.

The framework can be viewed as a dynamic system in which the agent continually generates goals in response to its perceived current and predicted future states of the environment as well as in response to its motivations. Each newly generated goal has a deadline by which it must be achieved as well as a value indicating its importance or priority. The *Select goal or action* process determines whether one of the goals within a plan should be achieved or whether one of the actions should be executed. If a goal is chosen, it is passed to a planner which plans to achieve that goal. An important part of the planning process involves determining whether or not goals may be achieved by their deadlines as well as assigning deadlines to actions and subgoals. The planner generates a search space of alternatives when planning, which requires a plan evaluation metric to select the most promising plan for further refinement (represented as *Select best plan*).

When a decision is made to execute an action, the *Execute action* component updates the plan and the model of the current state to reflect the changes that have occurred following execution. If the actual outcome differs significantly from the predicted outcome (i.e. enough to undermine the plan in some way), the *Recover* component is responsible for repairing the plan. In addition, as a consequence of changes to the environment and plan following execution, the agent's motivations may change (these are updated by the component *Update motivations*), which in turn may cause new goals to be generated or existing goals to be updated. The aim of this paper is not to provide details of generating or updating goals — others have addressed that issue, and detailed accounts of goal generation in response to motivations and feedback from the current plan can be found in [14,20].

3.2. *Updating Motivations*

The continuous planning framework of Figure 1 assumes that motivations change only in response to physical changes that arise within an agent's environment. Such changes, which are reflected by updating the agent's initial state model, may be caused by both the activities of the agent as well as by the activities of other agents and physical processes occurring within the environment. For example, once our example autonomous truck-driving agent has refuelled, the strength of the *conserve-fuel* motivation drops and, while the agent drives from one city to another, the strength of the *conserve-fuel* motivation increases.

It could be argued, however, that motivations might also change in response to changes in an agent's internal state (i.e. in a planning agent, internal state is encapsulated within the current plan representation which contains the current and predicted future states). For example, if, during the process of planning, an agent selects a new action to achieve a goal, the addition of this new action to the plan can be viewed as constituting a new belief. As a consequence of believing that at some point in the future the agent will be executing the new action (which, when executed, will cause the agent's motivations to change), the current strength associated with motivations might change. For example, the autonomous truck-driving agent may have a motivation associated with keeping busy. Adding a new action to its current plan in order to achieve a goal causes the agent to believe that at some point in the future it will be busier, therefore leading to an immediate decrease in the strength associated with keeping busy. This approach is more complex to model as the strengths of motivations will differ with respect to each plan, so that each plan representation must include a corresponding set of motivations, as discussed further in [4]. Plan evaluation would then be more difficult as, instead of evaluating the degree to which each plan supports the same set of motivations, each plan supports its own, different set of motivations. In order to simplify the modelling of motivations and their relationship to goal and plan generation, we therefore assume that motivations only change in response to physical changes to the environment (either by the agent, or by other agents and physical processes). When the planning agent plans, its motivations remain unaffected.

3.3. *The Impact of Assigning Priorities to Goals*

3.3.1. *Introduction*

When goals are generated in response to an agent's motivations, their priority or importance is determined by the intensity or motivational value associated with the motivations that led to their generation. Each goal is associated with a unique set of motivations, in other words, a unique set of motivations will, once the intensity of each motivation exceeds some threshold, cause a goal to be generated. The intensity of each motivation in the set will determine the importance or priority of the goal when it is generated. There are many ways of determining the importance of each newly generated goal: the importance of a goal may be the same as the highest intensity of its associated motivations; it may be the sum of the intensities of each motivation associated with the goal; it may be the average of the intensities [13]. For the purpose of this paper it is not necessary to choose how the importance of a goal might be calculated, it is only necessary to know that once a goal is generated, a value indicating its importance is calculated. This means it is possible to decide which goals are more important than others.

One of the key features of an autonomous motivated planning agent is this ability to assign priorities to goals as it enables the agent to perform two operations: the first involves deciding which goals to focus upon, deciding whether to achieve a goal or to execute an action, or deciding which action to execute (*Select goal or action*); the second involves choosing which goals to abandon, should there be insufficient time available to achieve all goals. These two operations are described in more detail in the following sections.

3.3.2. *Choosing whether to plan or whether to execute*

Planning is a continual process as new goals may be generated at any time in response to changes in both the agent's motivations and the external environment. The autonomous planning agent must therefore interleave planning with execution which means it must decide periodically whether to plan to achieve a goal or whether to execute an action — this choice is determined by the *Select goal or action* component of Figure 1. If the autonomous planning agent chooses to plan it must choose which goal to achieve, while if it chooses to execute it must decide which action to execute. To facilitate this, all goals and actions are assigned values indicating their priority or importance.

When goals are generated in response to changes in the agent's motivations and external environment, each goal is assigned a value indicating its priority as discussed above. This means that during the planning process, actions which contribute to those goals (i.e. actions that have been selected to achieve the goal) are also assigned a value indicating their priority. If an action contributes to a single goal, the action inherits the value indicating the importance of that goal. If the action contributes to more than one goal, it is assigned the sum of the values indicating the importance of each goal. (There are many ways of assigning priorities to actions; they could be assigned the value of the goal of maximum importance, or they could be assigned the value indicating the average importance.) Preconditions of actions have the same priority as their associated action. For example, Figure 2 shows a partial plan containing three goals, $g1$, $g2$ and $g3$ with actions $a1, ..., a7$. Arrows indicate ordering constraints, so action $a4$ occurs before $a1$ which occurs before $g1$. Actions $a1$ and $a4$ contribute only towards $g1$ and so inherit the value indicating the importance assigned to $g1$, namely the value 6. Actions $a6$ and $a7$, however, contribute towards the goals $g1$, $g2$ and $g3$, and so are assigned the sum of the values indicating the importance of goals $g1$, $g2$ and $g3$, namely the value 23.

In addition to taking into account the importance or priority associated with goals or actions, the component *Select goal or action* also takes into account the deadlines associated with goals and actions when deciding whether to plan or whether to execute. When goals are generated they are assigned a deadline (a value indicating the time by which they must be satisfied) along with a value indicating their priority. The planning process, in addition to assigning values indicating the importance of actions, also calculates the deadline by which the preconditions of an action must be true in order that the goal to which the action contributes may be satisfied by its deadline (this is discussed further in Section 3.3.4). For example, if a goal is to be achieved by 3pm, and an action that takes 30 minutes is selected to achieve that goal, the preconditions of that action must be made true by 2.30pm.

In conclusion, the component *Select goal or action* chooses whether to achieve a goal or whether to execute an action by taking into account the priority and deadline associated which each goal or action. Only actions whose preconditions are true in the current state may be considered for execution. Goals or actions with

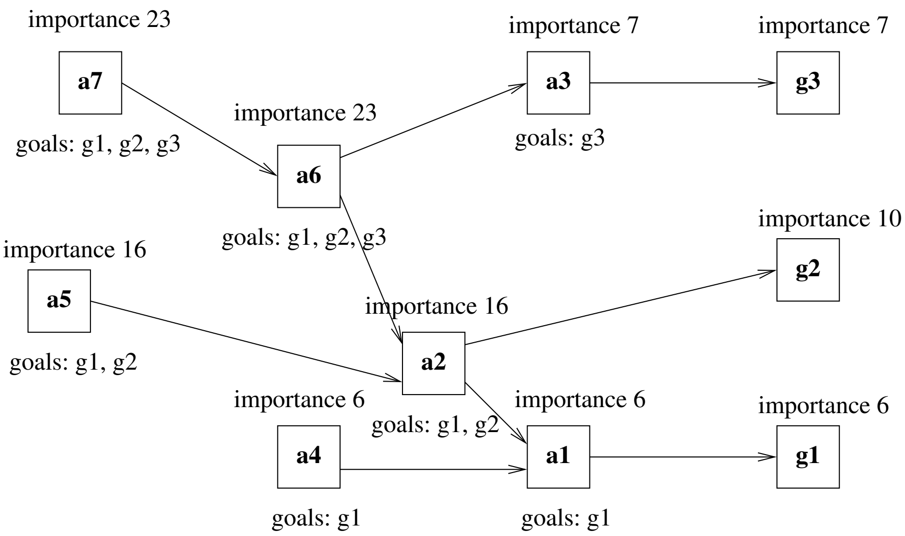

Fig. 2. The importance and goals assigned to actions

high priority and imminent deadlines are preferred over those with lower priority and deadlines that are not imminent. The ability to assign priorities to goals and actions can therefore be seen to have a considerable impact on the decision-making behaviour of the autonomous planning agent. In the following section we describe how the agent can use this ability to abandon goals should there be insufficient time available to achieve all goals by their deadlines.

3.3.3. *Editing Plans*

In order to describe the impact that assigning importance to goals has with regard to abandoning goals should there be insufficient time available to achieve all goals, we present the component *Plan to achieve goals* of Figure 1 in more detail, as shown in Figure 3. Whenever a goal (together with values indicating its importance and deadline) is selected by the *Select action or goal* component, it is passed to an AI planner, *Achieve goal* which generates a plan to achieve that goal. The implementation of this component is based on the partial order planning paradigm (see [17,22,29]) with various extensions which will be described later in Section 3.3.4. As part of the plan generation process, deadlines are estimated and assigned to all actions and their associated preconditions or subgoals — this is the responsibility of the component *Estimate deadlines*. If there is not enough time to achieve all of the goals in the plan by their deadlines, the plan is edited to remove those goals with low priority or low values indicating their importance, (see *Edit the plan*). Once the plan has been edited, new deadlines are estimated and assigned to actions and their subgoals. The process is complete when deadlines have successfully been assigned to all actions and their subgoals.

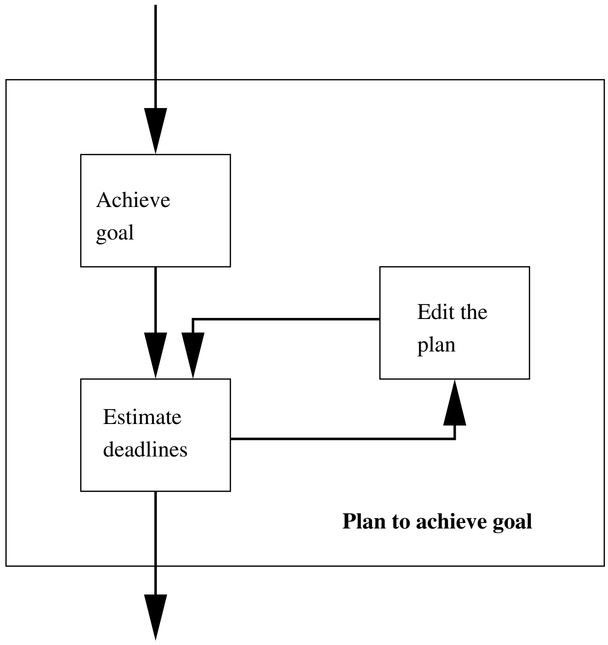

Fig. 3. Planning to achieve a goal

The *Plan to achieve goal* component of the continuous planning framework was implemented using an extended partial order planning paradigm for several reasons. Firstly, partial order planners output plans that offer a higher degree of execution flexibility than those generated by Graphplan style [3] and state search planners [10] and so are more suitable for frameworks in which planning and execution are interleaved [19]. In addition, partial order planners require only minor modifications to deal with the requirement that new goals may be generated continually during the planning process, as the new goals are simply added to the set of outstanding goals without affecting the planning process. Graphplan style planners, in contrast, would have to recommence the plan extraction process to take the new goals into account. Finally, Smith [27] argues that partial order planners offer a more promising approach for handling domains with durative actions, temporal and resource constraints. However, until recently, a significant drawback of partial order planning has been the lack of a good heuristic for selecting plans for further refinement as search control is of fundamental importance for partial order planning — work by [19,29] challenges the prevailing pessimism about the scalability of partial order planning by presenting novel heuristic control techniques.

When humans generate and execute plans to achieve their aims or goals, they often find that there will not be sufficient time available to achieve all of their goals. Instead of rejecting their plans and replanning from scratch, they may choose instead to abandon one or more of their goals, thereby freeing time which can be used to fulfill their other goals. In contrast, classical temporal planners will simply fail if there is insufficient time available to achieve all goals. One of the objectives when designing the continuous planning/execution framework was to give the autonomous planning agent the ability to emulate this aspect of human decision-making — namely the ability to remove one or more goals (together with their associated actions and constraints) from a plan once it has been determined that there is insufficient time available to achieve all of the goals in that plan. By editing a plan it is possible to preserve as much of the original plan as possible in contrast to replanning from scratch. However, this approach has various associated costs — in particular, it is necessary during the planning process to maintain a record of the dependencies between actions and goals to facilitate plan editing. This can be seen in Figure 2 where each action contains a record of the goals to which it contributes (for example $a5$ and $a2$ contribute towards the goals $g1$ and $g2$ while $a4$ and $a1$ contribute solely towards the goal $g1$). In addition, once a plan has been edited, deadlines have to be reassigned to the remaining actions in the plan, and, if there is still insufficient time available, the cycle of editing and reassigning deadlines is repeated. An alternative approach would be to simply replan from scratch once it is established that there is insufficient time available to achieve all goals, by presenting only a subset (selected by taking into account the priorities and deadlines of each goal) of the original set of goals to the planner. In the future it is intended to perform a set of experiments to determine whether or not the decision to edit the plan is more or less efficient than replanning from scratch. If replanning from scratch proves to be less costly, some of the main benefits of partial order planning such as being able to plan to achieve goals using a skeletal partial order plan, will be lost.

3.3.4. *Summary*

In this section we describe how the partial order planning paradigm (the component *Achieve goal* in Figure 3) has been extended in order to: support the autonomous agent in choosing whether to plan or whether to act; enable the autonomous planning agent to abandon one or more goals should there be insufficient time available to achieve all goals; allow the autonomous planning agent to choose the preferred plan for subsequent refinement.

- In order to enable the autonomous planning agent to choose whether to plan or whether to act, all actions and goals are assigned values indicating their priority as described in Section 3.3.2 above.
- If there is insufficient time available to achieve all goals (this is determined by the *Estimate deadlines* component), the plan is edited to remove goals of low importance or priority. In order to edit the plan, a record must be kept of the dependencies that exist between actions and goals. Currently, each action contains a list of the goals to which they contribute as shown in Figure 2. When a

goal is removed from the plan, all actions and constraints that contribute towards that goal (provided they do not contribute towards the achievement of any other goals) are also removed.

- A key requirement of the autonomous planning agent is that it has the ability to reason about whether or not there is sufficient time available to achieve goals by their deadline. In order to support this requirement, when creating a new action or further instantiating an existing action in order to achieve a goal it is necessary to estimate the duration of that action. The duration of an action may depend upon the values assigned to its parameters. For example, the duration assigned to an instance of the operator schema *drive-to(?x, ?y)* will depend upon the values assigned to the variables *?x* and *?y*. In this case, the exact duration can only be determined when both *?x* and *?y* have been instantiated. In the current implementation of the framework, a worst case estimate of the duration of each incomplete action instantiation is provided, which requires a degree of domain knowledge. For example, if the domain contains a network of locations within a town, the worst case estimate of the duration associated with instances of the *drive-to(?x, ?y)* would be the time taken to travel between the two furthest apart locations.
- Actions are assigned values indicating the degree to which they support the agent's motivations (for further details see [4]) — this discussed further in Section 3.4. In the implementation a look-up table containing such values is used.

The above extensions enable us to show how the autonomous planning agent is able to use priorities associated with goals to both decide whether to plan or to execute and to abandon the achievement of one or more goals if there is insufficient time available. In the following section we describe how the last extension enables motivations to be used to evaluate plans thereby allowing the autonomous planning agent to select its favourite plan for further refinement.

3.4. *Using Motivations to Evaluate Plans*

In planning to achieve a goal, a planning agent generates a search space of alternative plans and, at each stage of the planning process, must apply a heuristic to select the most promising plan for further refinement (see *Select best plan* in Figure 1). In this section, an approach based upon examining the degree to which each plan supports or undermines the agent's motivations is discussed.

We start by elaborating the truck-driving agent example we have been using throughout this paper, based on the DriverLog domain (used in the 3rd International Planning Competition) in order to illustrate how an agent's motivations affect the way it both generates goals and plans.

Figure 4 shows the topology of the domain, which consists of five cities connected by roads. The numbers indicate the distance from one city to another. It is the task of a truck-driving agent to transport packages or parcels from one city to another by some deadline. At any point in time, the truck-driving agent may receive an instruction to collect a package from one city, and to transport it to another city by a fixed deadline. In order to achieve its goals in an intelligent timely manner, the truck-driving agent requires the continuous planning/execution framework illustrated in Figure 1.

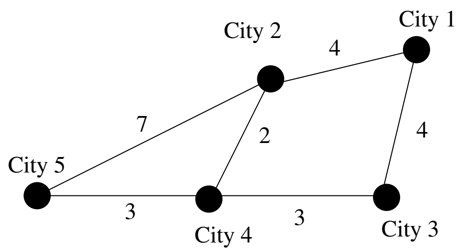

Fig. 4. The Driver Log Domain

Each time an agent executes an action within its environment, its motivations are updated to reflect the fact that it has brought about changes to its environment. For example, as a consequence of having driven between City 1 and City 2, the motivation associated with conserving fuel will increase in strength reflecting the decrease in fuel. This means that there is a difference between the current strength associated with the agent's motivations and the future strength of those motivations once the agent has executed some sequence of actions. The degree to which the actions within a plan support the agent's motivations can be determined by predicting the future motivations of the agent that arise as a consequence of executing those actions. This enables the planning agent to choose a plan containing the sequence of actions that best support the motivations, i.e. a plan containing the sequence of actions favoured by the agent.

In order to represent the degree to which each action may support or undermine motivations, the action representation has been extended to include two fields, *pros* and *cons*, where each contains a set of tuples $(name, strength)$ representing the *name* of each motivation supported/undermined by the action, together with the degree (*strength*) to which executing the action will support or undermine the motivation. Currently, these values are stored in a look-up table, requiring knowledge about the domain in which the planning agent is acting. For example, the action *drive-truck(truck city1 city2)* in Figure 5 has the associated set *pros* consisting of the tuples *((pleasure, 0.1))* and the associated set *cons* consisting of tuples *(conserve-fuel, 1.2), (conserve-tyres 1.0))*, so that when driving from City 1 to City 2, the truck-driving agent supports *pleasure* to a small extent (value 0.1), and undermines *conserve-fuel* and *conserve-tyres* to a greater extent (the road may be busy with traffic and full of pot-holes).

One difficulty with associating *pros* and *cons* with each action instance is that specific domain knowledge is required to provide the correct instantiation. For ex-

```
(:action drive-truck
   :parameters (truck city1 city2)
   :condition (and (connects city1 city2)
                   (at truck city1)
                   (has-fuel truck))
   :effect (and (not (at truck city1))
                     (at truck city2))
   :duration 3
   :pros ((pleasure 0.1))
   :cons ((conserve-fuel 1.2)
          (conserve-tyres 1.0)))
```

Fig. 5. A *drive-truck* action.

ample, an action representing the activity of eating a meal in a restaurant, *dine (?restaurant)* may support the same motivations to a greater or lesser degree depending upon which restaurant is selected during planning to instantiate the variable *?restaurant*. For example, if the chosen restaurant is expensive, dining there will undermine the motivation *save-money* to a greater degree than dining at a cheaper restaurant. Likewise, the *pleasure* motivation may receive greater support at a 3-star Michelin restaurant than a restaurant without a Michelin rating. In order to overcome this, it is assumed that the agent knows the degree to which executing each action instance supports or undermines its motivations. This assumes that the agent has some way of both acquiring and learning this information, which is beyond the scope of this paper. The implementation of the continuous planning/execution framework has a look-up table containing instantiations of the fields *pros* and *cons* for each action instance. The instantiation of the fields *pros* and *cons* of each action instance is performed by the component *Achieve goal* as described in Section 3.3.4 above.

Though many plan evaluation functions to determine the degree to which actions in a plan support an agent's motivations are possible, one simple example, used in the prototype implementation is given below.

$$FM = \sum_{i=1}^{n} \left(value_i - \sum_{j=1}^{p} strength_{pros_j} + \sum_{j=1}^{p} strength_{cons_j} \right) \tag{1}$$

Here, FM is the sum of the values indicating the predicted future strength associated with each of the agent's motivations i; n is the total number of motivations; $value_i$ is a measure of the current strength associated with each motivation i; p is the number of actions in the plan; $strength_{pros_j}$ is the degree to which each action j in the plan the supports motivation i (i belongs to a tuple in the set *pros* associated with action j); $strength_{cons_j}$ is the degree to which each action j undermines motivation i. As indicated above, this is just one particular way of evaluating the degree to which a sequence of actions (or plan) supports an agent's motivations.

There are many other ways of evaluating this support and future work will involve investigating which method is the more effective. Plans with a low value of FM are preferred as they support the agent's motivations to a greater degree than those with high FM.

In the truck-driving domain, the agent that is initially located in City 1 is given the task of delivering a package to a destination in City 5. The three alternative routes generated in three different plans involve: driving from City 1 via City 2 to City 5; driving from City 1 via City 2 and City 4 to City 5; and driving from City 1 via City 3 and City 4 to City 5. Table 1 indicates the degree to which each action instance supports or undermines the *pleasure*, *conserve-fuel* and *conserve-tyres* motivations. Now, a standard plan evaluation function might simply take into account the number of steps in a plan — in this case, the first route (via City 2 to City 5) is preferred as it has the least number of steps. By contrast, a plan evaluation function that minimises the distance travelled in each plan will choose the second plan as the total distance travelled is 9 units. Finally, however, if we assume that the initial motivational value associated with *pleasure*, *conserve-fuel* and *conserve-tyres* is 0.0 then, using the function described above, the preferred plan is the third plan with FM of -3.1 (FM for the first plan is 3.8 and for the second is 0.3).

Table 1. *pros* and *cons* associated with action instances (c1, c2, etc. are abbreviations for City 1, City 2, etc.)

action	*pros*	*cons*
drive(c1 c2)	(pleasure 0.1)	(cons-fuel 1.2) (cons-tyres 1.0)
drive(c1 c3)	(pleasure 1.9)	(cons-fuel 0.3) (cons-tyres 0.1)
drive(c2 c5)	(pleasure 0.2)	(cons-fuel 1.0) (cons-tyres 0.9)
drive(c2 c4)	(pleasure 1.2)	(cons-fuel 0.3) (cons-tyres 0.2)
drive(c4 c5)	(pleasure 1.8)	(cons-fuel 0.3) (cons-tyres 0.4)
drive(c3 c4)	(pleasure 1.2)	(cons-fuel 0.3) (cons-tyres 0.4)

4. Discussion and Conclusions

It is interesting to note the different results obtained depending on the plan evaluation mechanism. It should be clear that the plan evaluation function FM in Eq. (1) is not effective if it is used in isolation to guide the search for solutions to planning problems, as it is a poor measure of progress in achieving the goals in a plan. In experimental trials, FM was found to be useful in combination with an evaluation heuristic that minimised the number of actions and outstanding goals. A complete case-study and full details of our initial experimental results are available in [4].

A key aspect of goals and motivations is their dynamic organisation, which offers much scope for further elaboration of the mechanisms proposed. One limitation of the evaluation function FM described in the previous section is that it treats each motivation as being equal in importance to the autonomous planning agent. In practice, the autonomous planning agent may prefer to support one motivation more than another. For example, if the truck-driving agent has an urgent delivery deadline to meet, it would not be interested in trying to support the *pleasure* motivation by choosing a route it enjoys. Likewise, it may not be so concerned with conserving fuel. The relative importance of each motivation varies with different circumstances, but this issue is not currently implemented and requires further examination. In addition, plan evaluation should take into account the number of high priority goals achieved — a plan that achieves a small number of high priority goals may be preferred over one that achieves a larger number of low priority goals. Again, this has not been implemented and requires further examination.

In this paper, we have described how modelling the motivations of an agent can affect AI planning in terms of generating goals, assigning values indicating the priority of each goal, and choosing the best plan (i.e. the plan that best supports the agent's motivations). While others have recognised the importance of motivation (for example Luck [15] describes how motivations can be used to bias machine discovery to suit the current circumstances, and Norman [20] describes how the number of goals being achieved at any given time can dynamically alter the motivational value threshold affecting goal generation), there has been almost no work on the role of motivation in plan evaluation. Our work addresses that omission, both through the development of an conceptual model for motivated planning, and an implemented system. While the general framework has been set with positive initial results, including experimental trials, more remains to be done, especially in drilling down further into aspects discussed. (One immediate avenue to explore relates to the closer integration of separate models of goal generation and plan evaluation independently developed by the authors, but based on the same underlying motivational principles.) It is clear, nevertheless, that motivations can potentially provide an appropriate means for concentrating attention on the salient aspects of a problem to offer more effective planning behaviour.

References

1. C. Balkenius. The roots of motivation. In H. L. Meyer, J. A. Roitblat and S. W. Wilson, editors, *From Animals to Animats II.* MIT Press, 1993.
2. G. Ball and J. Breese. Emotion and personality in a conversational agent. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied Conversational Agents*, pages 189–219. MIT Press, 2000.
3. A. Blum and M. Furst. Fast planning through plan-graph analysis. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI95)*,

pages 1636–1642, 1995.

4. A. M. Coddington. *Self-motivated Planning in Autonomous Agents*. PhD thesis, University of London, 2001.
5. S. W. de Lioncourt and M. Luck. Motivated agent behaviour and requirements applied to virtual emergencies. In V. Marik, O. Stepankova, H. Krautwurmova, and M. Luck, editors, *Multi-Agent Systems and Application II*, pages 43–60. Springer, 2002. LNAI 2322.
6. M. d'Inverno and M. Luck. *Understanding Agent Systems*. Springer, 2001.
7. M. Fox and D. Long. PDDL2.1 : An extension to PDDL for expressing temporal planning domains. Technical report, Department of Computer Science, University of Durham, UK, 2001.
8. T. Halliday. Motivation. In T. R. Halliday and P. J. B. Slater, editors, *Causes and Effects*. Blackwell Scientific, Oxford, 1983.
9. J. R. P. Halperin. Machine motivation. In J. A. Meyer and S. W. Wilson, editors, *Proceedings of the First International Conference on Simulation of Adaptive Behaviour: From Animals to Animats*, pages 238–246. MIT Press/Bradford Books, 1991.
10. J. Hoffmann and B. Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research (JAIR)*, 14:253–302, 2001.
11. Z. Kunda. The case for motivated reasoning. *Psychological Bulletin*, 108(3):480–498, 1990.
12. J. Lester, S. G. Towns, C. B. Callaway, J. L. Voerman, and P. J. FitzGerald. Deictic and emotive communication in animated pedagogical agents. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied Conversational Agents*, pages 123–154. MIT Press, 2000.
13. F. Lopez Y Lopez. *Social Power and Norms: Impact on Agent Behaviour*. PhD thesis, University of Southampton, 2003.
14. M. Luck and M. d'Inverno. Motivated behaviour for goal adoption. In Zhang and Lukose, editors, *Multi-Agent Systems: Theories, Languages and Applications: Proceedings of the Fourth Australian Workshop on Distributed Artificial Intelligence*, pages 58–73. Springer, 1998. LNAI 1544.
15. M. M. Luck. *Motivated Inductive Discovery*. PhD thesis, University College London, University of London, 1993.
16. S. Marsella and J. Gratch. Emotional advantage for adaptability and autonomy. In *Proceedings of the Second International Joint Conference on Autonomous Agents & Multiagent Systems*, pages 313–320. ACM, 2003.
17. D. McAllester and D. Rosenblitt. Systematic nonlinear planning. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI91)*, pages 634–639, 1991.
18. D. Moffat and N. H. Frijda. Where there's a will there's an agent. In M. Wooldridge and N. R. Jennings, editors, *Intelligent Agents, ECAI-94 Workshop on Agent Theories, Architectures, and Languages*, pages 245–260. Springer, 1995.
19. X. Nguyen and S. Kambhampati. Reviving partial order planning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*, 2001.
20. T. J. F. Norman. *Motivation-based Direction of Planning Attention in Agents with Goal Autonomy*. PhD thesis, University of London, 1997.
21. E. Oliveira and L. Sarmento. Emotional advantage for adaptability and autonomy. In *Proceedings of the Second International Joint Conference on Autonomous Agents & Multiagent Systems*, pages 305–312. ACM, 2003.
22. J. Scott Penberthy and Daniel S. Weld. Temporal planning with continuous change.

In *Proceedings of the Twelfth National Conference of Artificial Intelligence (AAAI94)*, volume 2, pages 1010–1015, 1994.
23. R. Picard. *Affective Computing*. MIT Press, 1997.
24. H. A. Simon. Motivational and emotional controls of cognition. In *Models of Thought*, pages 29–38. Yale University Press, 1979.
25. A. Sloman. Motives, mechanisms, and emotions. *Cognition and Emotion*, 1(3):217–233, 1987.
26. A. Sloman and M. Croucher. Why robots will have emotions. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence (IJCAI81)*, pages 197–202, 1981.
27. D. E. Smith, J. Frank, and A. K. Jonsson. Bridging the gap between planning and scheduling. *Knowledge Engineering Review*, 15(1), 2000.
28. D. L. Waltz. Eight principles for building an intelligent robot. In J. A. Meyer and S. W. Wilson, editors, *Proceedings of the First International Conference on Simulation of Adaptive Behaviour: From Animals to Animats*. MIT Press/Bradford Books, 1991.
29. H. L. S. Younes and R. G. Simmons. On the role of ground actions in refinement planning. In *Proceedings of the Sixth International Conference on Artificial Intelligence Planning and Scheduling (AIPS'02)*, pages 54–61. The AAAI Press, 2002.