

# Towards Motivation-based Plan Evaluation

**Alexandra M. Coddington**

Department of Computer Science,  
University of Durham,  
Durham, DH1 3LE, U.K.  
A.M.Coddington@durham.ac.uk

**Michael Luck**

Department of Electronics of Computer Science,  
University of Southampton,  
Southampton SO17 1BJ, UK.  
mml@ecl.soton.ac.uk

## Abstract

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.

## 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 (Fox & Long 2001), a planning domain description 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 how aspects of context may be encapsulated in a plan evaluation metric to direct the search behaviour of a situated planning agent.

## The Use of Context in Planning

Human, real-world problem-solving involves a degree of subjectivity that has led researchers such as (Picard 1997) 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 (Kunda 1990).

## 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 are due to the *motivations* of each agent. Work on motivation in computing has been limited, but dates back to Simon (1979), who explored the relation of motivation to information-processing behaviour from a cognitive perspective. Sloman (1987) elaborated on Simon's work, showing how motivations are relevant to emotions and the development of a computational theory of mind. 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 eg. (de Lioncourt & Luck 2002).

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. 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 below.

## 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 (d'Inverno & Luck 2001). 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.

Norman (1997) 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 system. 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).

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 in 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.

## 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 (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.

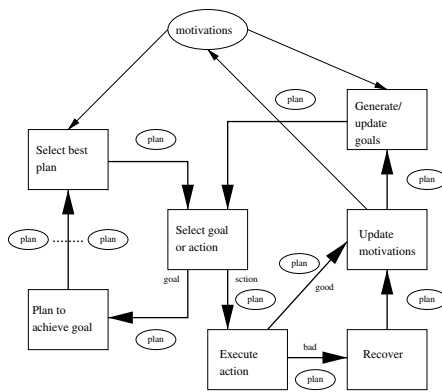


Figure 1: The Continuous Planning/Execution Framework

## A Continuous Planning Framework

### The Basic Architecture

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

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 should be achieved or whether one of the actions (within a plan) 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 pro-

vide 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 (Luck & d'Inverno 1998; Norman 1997).

### 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, during the process of planning, an agent selects a new or existing 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 differ with respect to each plan, so that each plan representation must include a corresponding set of motivations, as discussed further in (Coddington 2001). 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.

### 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. 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

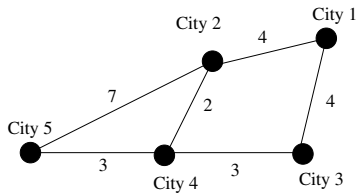


Figure 2: The Driver Log Domain

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 2 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 framework illustrated in Figure 1.

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 from 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 3 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 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)))
```

Figure 3: A *drive-truck* action.

action instance is that specific domain knowledge is required to provide the correct instantiation. For example, 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 framework has a look-up table containing instantiations of the fields *pros* and *cons* for each action instance.

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.

$$FM = \sum_{i=1}^n \left( value_{m_i} - \sum_{j=1}^p strength_{pros_{a_j}} + \sum_{j=1}^p strength_{cons_{a_j}} \right)$$

Here, *FM* is the sum of the values indicating the predicted future strength associated with each of the agent's motivations  $m_i$ ;  $n$  is the total number of motivations;  $value_{m_i}$  is a measure of the current strength associated with each motivation  $m_i$ ;  $p$  is the number of actions in the plan;  $strength_{pros_{a_j}}$  is the degree to which each action  $a_j$  in the plan the supports motivation  $m_i$  ( $m_i$  belongs to a tuple in the set *pros* associated with  $a_j$ );  $strength_{cons_{a_j}}$  is the degree to which each action  $a_j$  undermines  $m_i$ . 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 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

action	<i>pros</i>	<i>cons</i>
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)

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

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).

## 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* 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.

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 planning agent. In practice, the 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 (e.g. Luck (1993) describes how motivations can be used to bias machine discovery to suit the current circumstances, and Norman (1997) 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

- Coddington, A. M. 2001. *Self-motivated Planning in Autonomous Agents*. Ph.D. Dissertation, University of London.
- de Lioncourt, S. W., and Luck, M. 2002. *Motivated Agent Behaviour and Requirements Applied to Virtual Emergencies*. Springer. 43–60. LNAI 2322.
- d’Inverno, M., and Luck, M. 2001. *Understanding Agent Systems*. Springer.
- Fox, M., and Long, D. 2001. PDDL2.1 : An extension to PDDL for expressing temporal planning domains. Technical report, Department of Computer Science, University of Durham, UK.
- Kunda, Z. 1990. The case for motivated reasoning. *Psychological Bulletin* 108(3):480–498.
- Luck, M., and d’Inverno, M. 1998. Motivated behaviour for goal adoption. In Zhang, and Lukose., eds., *Multi-Agent Systems: Theories, Languages and Applications: Proceedings of the 4th Australian Workshop on Distributed AI*, 58–73. Springer. LNAI 1544.
- Luck, M. M. 1993. *Motivated Inductive Discovery*. Ph.D. Dissertation, University of London.
- Norman, T. J. F. 1997. *Motivation-based Direction of Planning Attention in Agents with Goal Autonomy*. Ph.D. Dissertation, University of London.
- Picard, R. 1997. *Affective Computing*. MIT Press.
- Simon, H. A. 1979. Motivational and emotional controls of cognition. In *Models of Thought*. Yale University Press.
- Sloman, A. 1987. Motives, mechanisms, and emotions. *Cognition and Emotion* 1(3).