Decisions, decisions: Knowledge goals in planning

Louise Pryor
Department of Artificial Intelligence, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN
louisep@aisb.ed.ac.uk

Abstract. Goals to acquire information, or *knowledge goals*, are common in the real world but have been largely ignored in the field of classical planning. This paper argues that they arise out of the need to make decisions in an unpredictable world, and describes a planner, Cassandra, that embodies this approach. Previous work on knowledge goals has concentrated on logics that can be used to represent them; the use of these logics appears to be consistent with the crucial rôle played by decisions in the construction of plans containing knowledge goals.

1 Introduction

Goals to find things out are ubiquitous in everyday life, and we have many ways of achieving them. We listen to the radio for the weather forecast and traffic news, look people up in address books and the telephone directory, and consult road maps and train timetables. We do all these things in order to acquire the information we need to perform our everyday tasks of getting from place to place and communicating with other people as efficiently as possible. Goals to acquire information, which are commonly termed *knowledge goals*, were first discussed about twenty-five years ago by McCarthy and Hayes [12]. However, in spite of their long history in AI they have generally been ignored by researchers into plan construction, especially in the area of classical planning,[1] which has long been the dominant theory. This paper starts by addressing the question of how they arise: what is it about a planning problem that leads to the presence of knowledge goals?

Knowledge goals have been ignored in the area of classical planning because they are rendered irrelevant by the assumptions on which classical planning systems are based. These systems are designed to construct a plan, or sequence of actions, that will achieve a specified goal from given initial conditions. The plans are constructed in advance and specify fully all the actions that are to be performed by the agent that executes them (which may not be the agent that constructs them). One of the chief concerns of recent research in this area has been to find algorithms that are provably complete and correct [2, 11]. The desire for these properties has led to the development of systems that rely on three crucial assumptions about the worlds in which they operate:

Simplicity: it is possible to know everything about the world that might affect the

[1] The term is due to Wilkins [24].

agent's actions;

Stasis: there will be no changes in the world except those caused by the agent's actions;

Certainty: the agent's actions have deterministic results.

Were these assumptions valid, knowledge goals would never arise: the planner would always already know everything that it needed to know. Of course, there are few natural worlds in which the classical assumptions hold up. Most natural environments display the following characteristics:

Complexity: it is impossible to know everything;

Dynamism: changes occur as a result of the actions of other agents or of natural phenomena;

Uncertainty: the agent cannot be sure what the results of its actions will be.

Environments with these characteristics are in many respects unpredictable. An agent constructing plans in such an environment may lack information at the time of plan construction that will be needed at the time of plan execution. Information may be lacking either because it is in principle unavailable, for example if it depends on the actions of another agent, or simply because the agent constructing the plan doesn't happen to have the relevant knowledge. Knowledge goals are therefore common in such environments.

In this paper I shall argue that, in an unpredictable environment, knowledge goals arise out of the need to make decisions during plan execution, and that this view provides a framework that accounts for knowledge goals resulting from both types of missing information. The next section describes the rôle of decisions and knowledge goals in planning. Section 3 decribes a plan construction system, Cassandra, whose design is based on these ideas. Section 4 discusses related work on knowledge goals.

2 Decisions and knowledge goals

An agent needs all sorts of information about the world in which it operates. It needs information both to perform mundane actions and to execute complex plans. For example, to make a bowl of cereal in the morning you need to know whether the cereal packet is in the cupboard or on the counter top; whether there is a bowl in the draining rack or whether to take one out of the cupboard; where to find the milk in the fridge; when you pour the cereal and milk into the bowl you must watch the level to stop it overflowing. Similarly, to travel from Chicago to London you need to know where the airport bus stops; what it looks like; when it arrives at the airport; which terminal your flight leaves from; which desk to check in at; which departure gate to go to; where your seat is; and so on.

2.1 Planning and decisions

This need for information arises because no action that an agent intends to perform can be completely specified without knowledge of the state of the world in which it will be carried out. How far you stretch your arm to reach the cereal packet depends on where the packet is; the steps you take between the check-in desk and the departure gate depend on the layout of the airport and the whereabouts of the other people moving around it. The effects of actions depend on the circumstances in which they are performed. For example, the amount of cereal in the packet determines how much cereal comes out when you tip it; whether a vehicle that you flag down is a taxi or a private car determines whether it will take you to the airport; whether the immigration official lets you into the country depends on whether the document you present is a valid passport.

An agent that uses a plan to achieve a goal is deciding in advance which of its goals it should pursue, what actions will achieve those goals, and when the actions should be performed. As Collins has pointed out, it is effectively deciding on a course of action that will achieve the chosen goal [5]. The agent makes these decisions on the basis of information that it has at the time that it chooses a plan. For example, you might decide to make a bowl of cereal on the basis of a number of beliefs, including the belief that you have a bowl, the belief that there is cereal in a packet in the cupboard, and the belief that there is milk in the fridge. To make a plan, an agent must therefore have the necessary information on which to base these decisions. You cannot plan to make a bowl of cereal if you have no idea whether you have a bowl or not, or if you are totally ignorant of the location of any milk. You could, of course, take an incremental approach: find out whether you have a bowl, and if you do take it to the table, find out whether you have cereal, and if so pour some into the bowl, and so on. In this case you cannot plan ahead to make the bowl of cereal because you lack the necessary information, but you can acquire the necessary information as you perform the necessary actions. The important point is that when planning for a goal you can only use the information available at the time of planning.

One of the difficulties of using plans in an unpredictable world is that it may be impossible to make all the necessary decisions in advance. The information needed to make some of these decisions may simply not be available at the time a plan is chosen. For example, you cannot predict the movements of all the other people who will be in the airport at the same time as you, so you cannot decide in advance on the exact steps you will take on your route from the check-in desk to the departure gate. When constructing its plans, the agent must therefore be able to recognise those decisions that must be deferred owing to the lack of information. When executing its plans, it must be able to make those decisions on the basis of the information that it has acquired since the planning process took place. In effect, the agent must interleave plan construction and plan execution.

Information is therefore required during plan execution in order to make the decisions that could not be made when the plan was constructed.[2] If all decisions could be made in advance, no further information requirements would arise during plan execution and there would be no need for knowledge goals. We can thus see why knowledge goals are not treated by classical planners: the very premise of classical planning is that all decisions can be made in advance.

A planner that lacks information and that therefore cannot construct a detailed plan for its goal need not abandon its plan altogether. Instead, it can construct a plan that includes provision for making the decisions that cannot be made when the plan is constructed: in other words, it can *defer* some of the decisions. Such plans must also provide for the acquisition of the information that will be needed to make the deferred decisions. The presence of knowledge goals in these plans is thus due to the presence of deferred decisions.

[2]It is also required in order to recognise when the decisions that were made during plan construction were wrong, and when decisions arise the need for which was not foreseen during plan construction. These issues are discussed further in [19].

2.2 Knowledge preconditions

The traditional view of knowledge goals is that they are subgoals of direct actions. McCarthy and Hayes were the first to discuss them; their classic example is that of needing to know the combination in order to open a safe [12]. They formalise this notion in the situation calculus by introducing the function *idea-of-combination* which represents a person's idea of a given safe's combination. The representation of the action of opening a safe must include a representation of the combination to be used; McCarthy and Hayes argue that the combination to be used should be one's idea of the combination, and that the action will succeed if one's idea is in fact correct, in other words if *idea-of-combination(person, safe, situation) = combination(safe)*. This formalisation, if used in a planner, would then lead to a knowledge goal to have an accurate idea of the safe's combination.

Moore argues that the knowledge preconditions for an action, such as opening a safe, arise because of the need to know what action to take [14]. An agent could, for example, know that dialling the combination of the safe would have the desired effect, but still not know what to do if the combination is unknown. Moore's approach to this problem is to use the concept of a *rigid designator* from possible worlds theory; he equates an agent's knowledge of what action to perform with the possession of rigid designators for the parameters of that action. Haas claims that Moore's approach is inadequate [10]: he says "there is no apparent reason why having a name that denotes the same object in all possible worlds should help you to perform an action." Instead, Haas proposes an approach using a syntactic theory of belief, in which the achievement of knowledge goals enables an agent to perform the relevant action. Morgenstern presents a formalism of action and planning in which the essential argument is again that an agent must in some sense know the parameters of the actions to be performed [15]. She, like Moore, takes a semantic approach using designators. All three concentrate on the representational issues involved in the problem of how an agent can determine that it can execute an action, or indeed a complete plan, and ignore the problem of how such plans can be constructed. Both Haas and Morgenstern, although using very different approaches, overcome many of the inadequacies of Moore's treatment.

It is notable that although Haas, Moore and Morgenstern all formalise the the notion of knowing the combination of a safe, say, as having effective parameters for the action of opening a safe, they all also appear to accept the intuition that the need to know the combination is simply a precondition of the action. Although their descriptions of knowledge goals as preconditions for actions ignore decisions, their formalisms are thus not inconsistent with the view presented in this paper that knowledge goals arise out of the need to decide exactly what actions to perform. This view provides a viable basis for the construction of plans containing knowledge goals, as we shall see in Section 3.

The unpredictability of the world may manifest itself as an inability to decide in advance exactly how to perform an action — for example, at exactly what angle to hold the milk over the cereal bowl — or as an inability to decide in advance what actions to perform — for example, whether to have cereal or toast for breakfast. However, there is no essential difference between the two forms of unpredictability. The dividing line between specifying the parameters of a single primitive action and specifying what actions should be included in a particular course of action is not hard and fast, but depends on the granularity of the representation. For example, the reasons for not knowing at what angle to hold the milk — because you don't know how much there is left in the carton — are similar to those for not knowing whether to have cereal or toast — because you don't know how much food there is in the kitchen. At one level you are simply finding the value for the *angle* parameter in the *pour-milk(angle, time)* action, while at another level you are simply finding the value for the *food* parameter in the *prepare-breakfast(food)* action.

The explicit consideration of decisions and the resulting knowledge goals allows this flexibility. To return to McCarthy and Hayes's safe example, the decision of what actions to perform in order to open a safe should be treated in the same way as the decision of what safe to open, or even whether to open a safe or go to the bank. In an unpredictable world it may well be impossible to tell in advance what the best method of obtaining cash will be; the decision of how to achieve that particular goal must be postponed until the circumstances under which the agent will be operating are known. In principle, this decision is exactly the same as the decision of whether to turn the dial on the safe to the left or the right; until the agent knows the safe combination, it cannot decide what actions it should perform.

The use of decisions also accounts for the example of the burglar who plans to break into a house and then open the safe. There are at least two methods that the burglar might use to open the safe:

- Search the desk for a piece of paper on which is written the combination, and then dial the figures indicated on the piece of paper;
- Use a stethoscope to listen to the tumblers and thus discover the combination.

This example shows the difficulty of viewing knowledge goals as direct preconditions of actions, as it appears not to be necessary to know the combination of the safe in order to open it. In the second case the burglar opens the safe by using the combination but without knowing it in advance. Instead of deciding what actions to perform in turning the dial on the basis of knowledge of what the combination is, the decision procedure used in the first case, she decides what actions to perform on the basis of the information she acquires by listening to the tumblers. However, the actions actually performed, expressed as a sequence of clockwise and anti-clockwise turns of the dial, are the same in both cases. The difference between the cases is not in the actual actions performed, and certainly not in their preconditions, but is in the procedures used to decide on the actions, and in the resulting knowledge goals.

3 Planning for knowledge goals

Straightforward classical planners have no knowledge goals, for reasons explained earlier. Recently, however, several attempts have been made to extend the use of classical planning techniques by relaxing some of the basic assumptions. In particular, the field of *contingency planning*[3] has experienced something of a renaissance. The issues of contingency planning were first addressed in the seventies by Warren [23], but were more or less ignored from then until recently [8, 18, 22, 7].

Contingency planning is necessary when the planner's knowledge of the world is incomplete, as is the case in most real world domains. A contingency plan allows for more than one possible course of action: in other words, contingency plans *branch*. The branch that should be followed depends on the circumstances that obtain when the plan is executed — it is because these circumstances cannot be predicted that a contingency

[3]The use of this term instead of the more common *conditional planning* avoids confusion with planners that use conditional effects (a.k.a. context-dependent effects).

plan is necessary. By their very nature, contingency plans must include provision for gathering information during plan execution; in other words, if knowledge goals appear at all in the process of plan construction, they will appear in the construction of contingency plans.

This section describes Cassandra,[4] a contingency planner that uses explicit deferred decisions in its plans and in which knowledge goals arise as a result of these decisions. This use of explicit decisions provides several significant advantages over other contingency planners, not least of which is the natural way in which knowledge goals are integrated into its plans.

3.1 Cassandra

Cassandra is a member of the family of partial order planners derived from McAllister and Rosenblitt's SNLP [11]. It is based on Penberthy and Weld's UCPOP which extends the basic SNLP algorithm to handle context-dependent effects[5] and a limited form of universal quantification [17]. Cassandra, like UCPOP, represents actions using modified STRIPS operators, each defined by the *preconditions* for executing an action and the *effects* that may become true as a result of executing it [9]. For each possible effect there is in addition an associated set of *secondary preconditions*, which specify the conditions under which the action will have that effect [16].

The use of secondary preconditions is critical to Cassandra's ability to represent uncertain effects. It does so by assigning them *unknowable* preconditions, constructed using the pseudo-predicate `:unknown`. Unknowable preconditions carry two parameters, one designating a particular source of uncertainty (*e.g.*, an instance of a coin toss), the other a particular possible outcome of that uncertainty ("heads"). The following propositions are assumed:

1. Each outcome of a given source of uncertainty has a unique name;
2. Different outcomes of a given uncertainty are mutually exclusive;
3. The set of named outcomes is exhaustive.

These assumptions license two critical judgments that Cassandra must be able to make: (1) that two actions or effects may not co-occur because they depend upon different outcomes of the same uncertainty, and (2) that a goal will necessarily be achieved by a plan, because it will be achieved for every possible outcome of every relevant uncertainty.

Planning under uncertainty, unlike conventional planning, requires the planner to build plan segments that may or may not be executed, depending on the contingencies that are encountered. Keeping track of which elements of the plan are relevant under various contingencies requires effective bookkeeping, which Cassandra accomplishes through a system of labelling. Labels are propagated over actions, effects, and goals in the plan to indicate both the contingencies in which these elements play a rôle, and the contingencies with which they are compatible. Note that these are two different issues: a plan element may be compatible with a contingency, in that it does not *interfere* with the achievement of any goals in that contingency, without actually helping to establish them.

Cassandra does not construct a contingency plan until it encounters an uncertainty. Until this point, it proceeds in much the same manner as other planners in the SNLP

[4] Cassandra was a Trojan prophet who was fated not to be believed when she accurately predicted future disasters.

[5] Following [3].

family, using a means-ends analysis involving the alternation of two processes: *planning for open subgoals* and *avoiding bad interactions*.[6] An uncertainty is introduced into a plan when a subgoal in the plan is achieved by an uncertain effect, *i.e.*, an effect with an unknowable precondition. Cassandra thus notices an uncertainty when its current plan becomes dependent upon a particular outcome of that uncertainty. In effect, the plan that it has built so far becomes a contingency plan for that uncertain outcome. To build a plan that is guaranteed to succeed, Cassandra must also construct contingency plans for all other possible outcomes of the uncertainty as well. This means splitting the plan into a set of *branches*, one for each possible outcome of the uncertainty. Cassandra therefore labels the components in the existing plan to show that they depend on a particular outcome of the uncertainty, and introduces a new set of goals for each other possible outcome. The plan is thus not complete until all the goals are achieved in every possible outcome. This procedure is repeated whenever an uncertainty is encountered.

Once an uncertainty has been encountered, and plan branches have been introduced, every new component (action or causal link) that is introduced into the plan is relevant only in some branches. Each component is therefore labelled with the contingencies in which it facilitates goal achievement (these labels propagate backwards from the goals) and the contingencies in which it hinders goal achievement (these labels propagate forwards towards the goals). The latter type of labels are introduced when an *unsafe* link is detected: a causal link that is threatened by another effect (the *clobberer*) in the plan. There are two standard methods for resolving an unsafe link in planners in the SNLP family: *separate* the clobberer and the unsafe link by adding codesignation constraints ensuring that they don't unify; or *reorder* the actions in the plan to ensure that the clobberer occurs outside the range of the link. The first of these methods is extended in Cassandra: another method of separating the clobberer from the unsafe link is to ensure that they only occur in different contingencies, resulting in plan components being labelled with contingencies in which they may not occur.

3.2 Decisions in Cassandra

When an agent executes a branching plan, it must at some point decide which branch to take. Rather than treating this decision as an explicit part of the plan, previous work has in effect simply assumed that the agent will execute those steps that are consistent with the contingency that actually obtains [23, 18]. However, in order to know which contingency holds during execution, an arbitrarily large amount of work may be necessary in order to gather the information on which the decision is based [19]. For the plan to be viable the planner must first be able to ensure that it will be in a position to make the decision; and second guarantee that the steps required to gather information do not conflict with those required to carry out the rest of the plan. The most straightforward approach to this problem is to treat decisions as explicit plan steps, along with the actions to acquire and evaluate information that support the decision-making procedure. By doing this it can make use of its existing mechanisms for achieving preconditions and avoiding conflicts.

Cassandra, in contrast to previous planners, adds an explicit decision step each time it introduces a new branch into a plan. It is added to the plan along with ordering constraints ensuring that it occurs after the step with which the uncertainty is associated and before any step with a subgoal the achievement of which depends upon a particular outcome of the uncertainty.

[6] Cassandra's algorithm is described in [22].

For a decision step to be operational, there must be an effective procedure by which the agent executing the plan can determine which decision to make. In Cassandra, the action of deciding which contingency to execute is modelled as the evaluation of a set of condition–action rules. Each decision step has associated with it a set of rules, derived from an analysis of the uncertainty that led to the introduction of the new branch, representing the decision procedure that will be used to determine whether that branch should be performed when the plan is executed. The action of deciding which contingency to execute is thus modelled in Cassandra as the evaluation of a set of condition–action rules of the form:

If *condition1* then *contingency1*
If *condition2* then *contingency2*
If *condition3* then *contingency3*
...

The executing agent can then make the decision as to which branch to pursue by evaluating these rules when it comes to the decision step in the course of executing the plan. Since the intended effect of evaluating the decision-rules is to choose the appropriate contingency given the outcome of a particular uncertainty, the conditions are effectively meant to be diagnostic of particular outcomes of the uncertainty. The agent cannot, of course, directly determine the outcome of an uncertainty, so it must infer it from the presence or absence of effects that depend upon that outcome.

The most straightforward approach to constructing the antecedent conditions of a decision-rule would be to analyse the plan operators to identify all the effects that could be expected to result from a given outcome of the uncertainty, and make the antecedent be the conjunction of these effects. However, this turns out to be overkill. In fact, it is necessary only to verify that the contingency plan can actually succeed.[7] It is thus only necessary to check for those effects of a given outcome of an uncertainty *that are actually used to establish preconditions in the contingency associated with that outcome.* The antecedent condition of the decision-rule constructed by Cassandra is thus simply a conjunction of all the direct effects of a particular outcome that are used to establish preconditions in the contingency plan for that outcome. Decision-rules are constructed incrementally as the plan is elaborated and effects are used to establish preconditions, as described in [22].

In order to evaluate a decision-rule, the executing agent must be able to determine whether the rule's antecedent holds. The preconditions for the decision step must thus include goals to know the current status of each condition that appears as an antecedent of a rule in this condition. The preconditions of a decision step become open subgoals in the plan in the same way as do the preconditions of any other step. In other words, Cassandra generates a set of goals to acquire the knowledge necessary to evaluate the decision-rules. These knowledge goals then become preconditions of the decision step, and are planned for in the same manner as any other subgoals in the plan. The knowledge goals in Cassandra's plans thus arise directly from the decisions that must be made in the course of their execution.

The actual representation of knowledge goals in Cassandra is a variant of the syntactic approach proposed by Haas [10]. They are all of the form *know-if(fact)*, where *fact* is one of the conjuncts in the antecedent of a decision-rule. This simple representation has

[7] This has the interesting consequence that the executing agent might, in principle, end up selecting a contingency plan even though the outcome of the uncertainty were not the one with which that plan was associated. Notice that this would not cause a problem in the execution of the plan, since it would only occur if all the conditions for the plan's success were met.

the advantage that if is very easy to recognise when another action in the plan potentially interferes with the achievement of a knowledge goal. Cassandra can therefore use its existing mechanisms to avoid conflicts. This representation is adequate for the current version of Cassandra, in which all possible outcomes of any given uncertainty are known. In general, Cassandra's representation, based as it is on STRIPS-like operators, is less powerful than the logics proposed by either Haas or Morgenstern [10, 15].

4 Related work

This section compares the treatment of knowledge goals in Cassandra to that in other recent contingency planners: SENSp [8], CNLP [18] and C-BURIDAN [6, 7].

4.1 How knowledge goals arise

Cassandra is the first planner in which deferred decisions are represented as *explicit* actions in the plans that it constructs. Cassandra's knowledge goals arise specifically from the need to decide between alternative courses of action, as preconditions of the decision actions.

In SENSp, information-gathering actions are included in a plan in order to bind *run-time variables*. Run-time variables are treated as constants whose values are not yet known: in Cassandra's terms, they can be seen as variables whose values depend on uncertain outcomes. In a sense, every run-time variable that appears in the plan represents a knowledge goal. In other words, knowledge goals in SENSp arise only in response to the decision of what the parameters for an action should be, and not in response to the decision of what method to use.

CNLP's information-gathering actions are included in a plan whenever an action with uncertain effects occurs. This is necessary because the uncertainty is represented in the information-gathering action rather than in the action that actually produces the uncertainty. Knowledge goals are thus not represented explicitly in CNLP. There is no way in CNLP to distinguish the need to know whether a particular plan branch will work from the need to know the actual outcome of an uncertainty, as there is in Cassandra.

C-BURIDAN's information-gathering actions are introduced into a plan as part of the branching process. It appears they are not considered as achieving specific knowledge goals: in the plans Draper *et al.* present there are no causal links from information-gathering actions to other subgoals in the plan [6, 7]. It is not clear whether C-BURIDAN can construct plans in which several actions are required to distinguish between possible outcomes.

Knowledge goals are thus not represented as arising *explicitly* from the need to decide between alternative branches of the plan in SENSp, CNLP, or C-BURIDAN. As a result, none of the three distinguishes effectively between sensing or information-gathering actions on the one hand, and decision making on the other. An important reason for making this distinction is that a decision may depend on more than one piece of information, each available through performing different actions.

In addition, separating information-gathering from decision-making provides a basis for introducing alternative methods for making decisions. For example, an extension of Cassandra introduces a type of decision that directs the executing agent to perform all branches resulting from a given source of uncertainty [4], which allows the construction of plans that can succeed in situations in which there is no way of telling what the actual outcome is — for example, in Moore's "bomb in the toilet" problem (cited by McDermott [13]). The explicit representation of different methods for making decisions is an important direction for future research.

4.2 Achieving knowledge goals

Information acquisition during plan execution is a goal-driven activity; the goals arise as preconditions of the decisions that the agent must make. In fact, in these situations information acquisition is a *planning task* [20, 21, 19]. The acquisition of information should not be given special treatment in planning, but should be treated as just another task to be accomplished. Cassandra's plans include steps to acquire information in support of deferred decisions.

All decisions have information preconditions that must be planned for and achieved in the same way as the preconditions of any other actions. The explicit treatment of decisions thus demands the explicit treatment of the acquisition of information. The plans constructed by Cassandra include explicit decision actions, and also allow arbitrarily complex subplans to acquire information in support of those decisions.

In Cassandra, planning for knowledge goals is thus fully general: this is a significant difference from other contingency planners. Actions that achieve knowledge goals cannot have preconditions in SENSP: this restriction is required in order to maintain completeness [8]. This is a serious limitation. There is no reason to assume that knowledge goals can always be achieved by a single action that has no preconditions. The confusion between the source of uncertainty and the observation of uncertain results limits the ways in which knowledge goals can be achieved in CNLP: they must be achieved through special observation actions that specify the uncertain outcomes [18]. Neither of these two systems allows fully general methods of information gathering in response to knowledge goals.

C-BURIDAN allows information-gathering actions to have fully general preconditions [7]. Moreover, these actions may be *noisy*: they are not guaranteed to produce accurate results. For example, if a coin has landed heads up, an *observe* action might produce the result "heads" with a probability of 90% and the result "tails" with a probability of 10%. In principle, Cassandra's information-gathering actions may have uncertain effects in just the same way as other actions: however, because Cassandra is not a probabilistic planner it cannot reason about the reliability of evidence as C-BURIDAN can. Both C-BURIDAN and Cassandra allow a single action both to have causal effects and to result in the gathering of information.

5 Conclusions

This paper has argued that knowledge goals arise out of the need to make decisions in an unpredictable world, and has presented a planner, Cassandra, that demonstrates this approach. Cassandra explicitly plans to make deferred decisions: decisions the need for which can be foreseen but that cannot be made in advance. Cassandra also plans to acquire the information that will be required to make these decisions: in Cassandra's plans, knowledge goals arise as subgoals of decision steps.

There is a major issue that has not been addressed in this paper. Not all decisions can be foreseen — the essence of an unpredictable world is that unexpected things will happen. Often, a decision must be made whose necessity has not been predicted. A mechanism for recognising such decisions is presented in [19]. However, unexpected decisions require information in just the same way as do deferred decisions; when these decisions are encountered, they too result in knowledge goals. These knowledge goals are no different from those arising from deferred decisions, and, once the need for them has been recognised, they can be planned for in exactly the same way.

The approach described in this paper is based on two observations: first, knowledge goals are necessary because the world is unpredictable; and second, plans that contain knowledge goals must actually be constructed. The work described in Section 2.2 does not specifically address either of these points. This paper has shown that taking the notion of unpredictability seriously enables a deeper understanding of how knowledge goals fit into a general theory of planning. Knowledge goals will inevitably arise in an unpredictable environment such as the real world, and no theory of planning is complete if it does not account for the process of plan construction. The design of Cassandra illustrates how an understanding of the reasons that knowledge goals arise supports a method of constructing plans that contain them. However, Cassandra itself does not by any means constitute a complete theory of planning, not least because of the inadequacies of the STRIPS-based representation. Future research is to combine the explicit representation of decisions that is the chief contribution of this work with a more powerful and flexible scheme of representation.

Acknowledgments

Much of this work was performed while the author was a student at the Institute for the Learning Sciences, Northwestern University. Thanks to Gregg Collins for many helpful ideas and useful discussions.

References

[1] Allen, J., J. Hendler, and A. Tate, (Eds.) 1990. *Readings in Planning*. San Mateo, CA: Morgan Kaufmann.

[2] Chapman, D. 1987. *Planning for Conjunctive Goals*. Artificial Intelligence 32: 333–337. Also in [1].

[3] Collins, G. and L. Pryor. 1992. "Achieving the functionality of filter conditions in a partial order planner." *Proceedings of the Tenth National Conference on Artificial Intelligence*, San Jose, CA, AAAI, 375–380.

[4] Collins, G. and L. Pryor. 1995. *Planning under uncertainty: Key issues.* Submitted.

[5] Collins, G. C. 1987. *Plan creation: Using strategies as blueprints*. Department of Computer Science, Yale University. Technical Report YALEU/CSD/RR #599.

[6] Draper, D., S. Hanks, and D. Weld. 1994. "A probabilistic model of action for least-commitment planning with information gathering." *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Seattle, WA, Morgan Kaufmann.

[7] Draper, D., S. Hanks, and D. Weld. 1994. "Probabilistic Planning with Information Gathering and Contingent Execution." *Proceedings of the Second International Conference on Artificial Intelligence Planning Systems*, Chicago, IL, AAAI, 31–36.

[8] Etzioni, O., S. Hanks, D. Weld, D. Draper, N. Lesh, and M. Williamson. 1992. "An approach to planning with incomplete information." *Proceedings of the Third International Conference on Knowledge Representation and Reasoning*, Boston, MA, Morgan Kaufmann.

[9] Fikes, R. E. and N. J. Nilsson. 1971. *STRIPS: A new approach to the application of theorem proving to problem solving*. Artificial Intelligence 2: 189–208. Also in [1].

[10] Haas, A. R. 1986. *A Syntactic theory of Belief and Action*. Artificial Intelligence 28: 245–292.

[11] McAllester, D. and D. Rosenblitt. 1991. "Systematic nonlinear planning." *Proceedings of the*

Ninth National Conference on Artificial Intelligence, Anaheim, CA, AAAI, 634–639.

[12] McCarthy, J. and P. J. Hayes. 1969. “Some philosophical problems from the standpoint of artificial intelligence.” In *Machine Intelligence 4*, ed. B Meltzer and D. Michie, Edinburgh University Press, 463–502. Also in [1].

[13] McDermott, D. V. 1987. *A Critique of Pure Reason*. Computational Intelligence 3: 151–160.

[14] Moore, R. C. 1985. “A Formal Theory of Knowledge and Action.” In *Formal Theories of the Commonsense World*, ed. J. R. Hobbs and R. C. Moore. Norwood, NJ: Ablex. Also in [1].

[15] Morgenstern, L. 1987. “Knowledge Preconditions for Actions and Plans.” *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, IJCAI, 867–874.

[16] Pednault, E. P. D. 1988. *Synthesizing plans that contain actions with context-dependent effects*. Computational Intelligence 4: 356–372.

[17] Penberthy, J. S. and D. S. Weld. 1992. “UCPOP: A sound, complete, partial order planner for ADL.” *Proceedings of the Third International Conference on Knowledge Representation and Reasoning*, Boston, MA, Morgan Kaufmann.

[18] Peot, M. A. and D. E. Smith. 1992. “Conditional Nonlinear Planning.” *Proceedings of the First International Conference on Artificial Intelligence Planning Systems*, College Park, Maryland, Morgan Kaufmann, 189–197.

[19] Pryor, L. 1994. *Opportunities and Planning in an Unpredictable World*. The Institute for the Learning Sciences, Northwestern University. Technical Report #53.

[20] Pryor, L. and G. Collins. 1991. “Information gathering as a planning task.” *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, Chicago, IL, Lawrence Erlbaum Associates, 862–866.

[21] Pryor, L. and G. Collins. 1992. “Planning to perceive: A utilitarian approach.” *Workshop notes of the AAAI Spring Symposium: Control of Selective Perception*, Stanford, CA, AAAI.

[22] Pryor, L. and G. Collins. 1993. *Cassandra: Planning with contingencies*. The Institute for the Learning Sciences, Northwestern University. Technical Report #41.

[23] Warren, D. 1976. “Generating conditional plans and programs.” *Proceedings of the AISB Summer Conference, University of Edinburgh*.

[24] Wilkins, D. E. 1988. *Practical Planning: Extending the Classical AI Planning Paradigm*. San Mateo, CA: Morgan Kaufmann.