Open-ended planning
Extended abstract

Charles L. Ortiz, Jr.
Artificial Intelligence Center
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
ortiz@ai.sri.com

Abstract

Most real-world planning problems arise in incompletely specified situations. One rather computationally expensive way to cope with such incompleteness is to formulate comprehensive contingency plans. A more attractive alternative is to instead design more flexible planning systems: for example, it should be possible for a planner to incorporate reasonable default information when available; it should also be possible for a planner to postpone planning parts of a plan lacking sufficient information and begin execution, if it is more likely to later be in a position to complete those parts of the plan. Conventional planners, however, are not sufficiently flexible in these respects; indeed, most plan representations still assume a completely specified world. This paper presents an extension to HTN planning — called *open-ended planning* — in which preferences to certain plan-execution behaviors can be expressed in order to achieve such flexibility.

Keywords: Planning under incomplete information, plan execution and monitoring, belief revision, HTN planning.

This paper describes ongoing work on a new approach to planning with incomplete information called *open-ended planning.* It extends hierarchical task network (HTN) planning to problems in which worlds and goals are only partially described. One way in which such incompleteness can manifest itself is in the form of statements in the planner's representation language that include disjunction. To manage the complexity that such expressiveness inevitably gives rise to, open-ended planning supplies a framework in which preferences to certain plan-execution sequences — called *behaviors* in this paper — can be stipulated by the user. These preferences specify strategies for goal expansion when requisite information is missing: if a planner cannot ascertain whether a precondition is satisfied it might substitute a default value for that precondition or perhaps postpone expansion of that goal in favor of execution of preliminary parts of an existing (partial) plan (presumably so that the agent would later find itself in a situation in which the value could be discovered).

This paper begins by discussing some of the problems that come about when attempting to formulate plans in the face of incomplete information. A sketch of a formalization which extends HTN planning is then described; this extension can handle incomplete information, context-dependent effects, and defaults; the focus is on incompleteness of beliefs and not uncertainty of beliefs. A number of strategies for coping with incompleteness are then examined and an algorithm implementing these ideas is briefly described.

Sources of incompleteness

Imagine a robot assistant that could follow commands to locate objects and that could also navigate to retrieve and deliver those objects. Suppose that such a robot were situated within the area shown in Figure 1. Now, consider the following hypothetical exchange.

```
Human> Please get me a pen.
Robot> Where are they?
Human> In the stationary cabinet.
Robot> Where is that?
Human> In the printer room.
Robot> Where is that?
Human> Across from the conference room.
Robot> Ok.
```

Suppose that we wish to build a robot that could execute such a task. Consider the sorts of information about the world that such a robot could reasonably have available to it and the degree of partiality of such information. Assuming that the robot represented its beliefs in some logical language, the above statements might be captured in the following way. The robot's goal could be expressed as:

$$\exists s.have(s) \land pen(s)$$

That is, the robot wants to achieve a state in which it has possession of some object, s, that happens to be a pen; any pen that the robot should come across along the way would also satisfy its goal.[1] The other

[1] Assuming, of course, that the robot is equipped to respect prevailing deontic conditions.

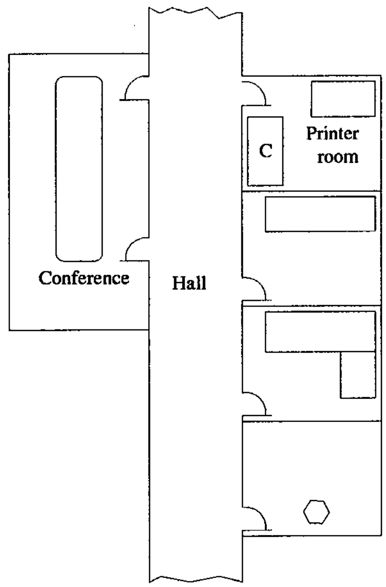

Figure 1: Map of the robot's world. The hallway in the center accesses a conference room on the left and a set of offices on the right. The robot assistant is shown as a hexagon and the stationary cabinet (C) is in the printer room.

information acquired as a result of the exchange could be represented as:

$$\exists s \exists c \exists l \exists o. pen(s) \wedge cabinet(c) \wedge in(s,c) \quad (1)$$
$$\wedge\, in(c,o) \wedge office(o) \wedge at(o,l) \wedge across(l,R)$$

The first line states that the robot knows that there is some pen that is in a cabinet somewhere. The second line provides information on the cabinet: it is located in an office across from the conference room, R. The robot can also be assumed to have some additional knowledge: a map of the offices (with office numbers). For this example, however, we assume that the robot does not know the identity of the printer room.

The existential statement in (1) can be equivalently expressed, assuming a finite universe of objects, as a disjunctive statement. That is, if a domain contains exactly two objects, a and b, then the statement $\exists\phi(x)$ can be expressed as $\phi(a) \vee \phi(b)$. The goal statement can be similarly translated into a disjunctive statement. The problem in formulating a plan in a scenario such as the one described is then that standard HTN and state-based planners require restricted representations that do not permit disjunction in either the world description or in the goal statement.[2]

[2]A representation that is equivalent to one that captures uncertainty through disjunctive statements is one that makes use of a multi-dimensional logic to translate incompleteness at the model level to indifference with respect to some set of (complete) worlds. Such a representation is adopted in the full paper.

Notice that one would not want a robot to manage such incompleteness by considering plans for every possible office, cabinet or pen. Instead, in this case, a reasonable way to proceed would be to navigate towards the conference room in the hope of finding the printer room (perhaps by asking someone or looking for signs) and then the cabinet and then a pen. Notice also that one cannot expect that all of the steps involved in planning to get the pen once near the conference room are independent of getting there: for example, the robot may have to reason that it will need a key for the cabinet before it sets out towards the printer room.

Sketch of formalization

The full paper formalizes these ideas through extensions to Erol's (1995) work on HTN planning, within a logic which has the following properties.

- A representation and a semantics for primitive actions, methods for achieving higher level actions, and knowledge-producing actions, as well as a representation of incompleteness in terms of possible worlds. Knowledge is interpreted at the meta-level in terms of an S5 modal logic: that is, in terms of truth in all possible worlds (Chellas 1980).
- Behaviors correspond to sequences from a set of *mental actions* (McCarthy 1995; Ortiz 1999). The set of mental actions include the actions: *elaborate*, *update*, *revise*, *extend execute*, *look*, and *monitor*. The elaborate action takes a partial plan and expands it, the update action simulates the performance of an action, the revise action corrects the agent's beliefs (after, for example, a sensing action), and the extend action takes a set of possible worlds corresponding to the agent's belief state and extends it by making some default assumptions. The execute action takes a partial plan and executes a preliminary part of the plan and the look action is a sensing action used to check the value of some fluent. The monitor action is used to check whether or not some previous assumption was, in fact, correct. In the case of these last three action types, only the mental aspects of execution are addressed in this paper: that is, the corresponding changes to the current plan and belief state. In an actual system these processes would be augmented with additional processes that controlled actual motor activities.
- The use of deductive operators, as in SIPE, for inferring side-effects of actions and default rules for drawing inferences when information is incomplete. The set of predicates is partitioned into several types: *closed, open, assumable*, and *postponable*; these correspond, respectively, to predicates which are subject to the closed world assumption, ones that are not, ones that can be substituted for by some default, and ones for which ascertaining their truth can be postponed. This permits the behaviors to be conditioned on the type of information available.

- A semantics for plan satisfaction and a proof theory for plan derivability together with soundness and completeness results. Since the intent of open-ended planning is to be able to deal with partial world descriptions, plan satisfaction is expressed relative to a set of worlds, U, that reflect an agent's incomplete beliefs. A partial plan, π, is said to be satisfied relative to the set of worlds, U, just in case π satisfies goal g for each $u \in U$. Unfortunately, this introduces a great deal of complexity (Baral, Kreinovich, & Trejo 1999). For example, if in the initial state the formula $p \vee q$ is true, and the goal is g then the *same* plan must be found that will take one from the three initial states: $\{p \wedge q, p \wedge \neg q, \neg p \wedge q\}$ to g. The point behind open-ended planning is to free an agent from having to complete a plan before execution begins: if an agent has good reason to assume, for example, $p \wedge \neg q$, then that agent could do so and follow this with execution of part of the plan. If this resulted in the agent achieving g, then there was no need in the first place for contingency planning. Of course, if the agent's assumptions turn out to be incorrect, then the agent may have to physically backtrack or, even worse, get to a situation in which g might not be achievable at all.

Agent types

Within the open-ended planning framework, an agent type corresponds to a set of possible behaviors: a particular type of agent will prefer certain behaviors over others depending on the situation in which the agent is embedded. Open-ended planning supplies the machinery with which a designer can specify such preferences.

One way to classify an agent is in terms of the degree to which the agent is willing to deliberate; two extremes are possible. A *cautious* agent might be one that considers every possible contingency before acting. In contrast, a *bold* agent might be one that is willing to make assumptions whenever its beliefs are incomplete. In the first case, the agent may deliberate too much whereas in the latter case the agent might be too optimistic.

The full paper describes a number of behavioral preferences which are glossed in the remainder of this section.

Preference 1 (Contingency planning) *Contingency planning is captured through conditional plans which represent alternatives for all possible states of the world. Contingency planning represents the default behavior when other, more preferred behaviors of the sort described below are not available.*

Less cautious strategies appear more reasonable. One way in which an agent can eliminate incompleteness is by enacting some information gathering plan. If some ϕ is unknown and the agent does not believe that ϕ will be irrelevant to its future planning it can focus on planning for ϕ. (Computing irrelevance is discussed below.) Once such a plan is devised, it may be preferable for the agent to enact it — in order to ascertain whether or not ϕ holds — rather than to plan for subsequent contingencies. Note that subscribing to such behavior depends on an agent's other goals; such preferences can be specified in a variety of ways using the language described in the full paper, either with generality or great specificity.

Preference 2 (Exploratory behavior) *Prefer exploratory behavior to ascertain the knowledge preconditions (ϕ) for some action in a partial plan, π. That is, when ϕ represents a knowledge precondition of some act in π, elaboration of the sub-plan, τ of π followed by execution and an observation to ascertain the truth of ϕ is preferred over elaborating the entire π task. Knowledge preconditions are any preconditions whose value is not known.*

Preference 3 (Assumptive behavior) *If it is not known whether ϕ, and ϕ is assumable and not irrelevant, then apply defaults and set up monitor to verify assumption at a later time.*

For example, suppose one doesn't know whether one has gas in one's car and one wants to plan to go to the grocery store. Instead of involved contingency planning, one can assume that one does have gas in the car and then check that assumption later. Again, such a preference is not meant to be completely general.

Preference 4 (Eliminating irrelevant conditions) *If ϕ is irrelevant to task net τ then restrict contingency planning to non-ϕ states.*

Irrelevance reasoning is important in general cases of incompleteness as a pre-processing step: if p is unknown and is determined to be provably irrelevant to the agent's goals then the agent can proceed as if its beliefs were complete (Levy & Sagiv 1993). For example, in the pen example, let the language include a constant, Mary, such that it is not known where Mary is located. Irrelevance reasoning would be used to determine whether that additional fact was needed for forming a plan for the goal. Since a plan is represented as a conditional action, the plan corresponding to such a situation would not include a case for that fact.

Preference 5 (Anticipatory behavior) *For tasks which require certain resources, always check that one has possession of those resources before execution or further planning.*

For example, in the pen example from the beginning of this paper, it makes sense to check first whether one will need a key (and whether one has one) to open the stationary cabinet. We observed this same sort of preference in another domain involving air campaign planning: for certain problems which required planning for the task sequence `defense` and then `offense`, it made sense to plan the offense first as it might uncover the need for certain resources that could impact the defensive planning. This sort of anticipatory behavior can be handled in two ways: by employing a sufficiently expressive representation in which such resources can be made explicit (SIPE does this, for example) or through some sort of hypothetical reasoning in which one imagines a likely future situation in order to anticipate what one would need in such a situation.

Preference 6 (Domain dependent examples) *An example of a strictly domain-dependent preference is one in which a pessimistic agent might always compute a backup plan.*

There are a number of other types of incompleteness which can impact the choice of behavior. For example, various forms of incompleteness can be analyzed from the perspective of representation under disjunction (Borgida & Etherington 1989). Uniform predicate disjunctions, for example, are formulas of the form $p(a) \vee p(b)$ and might be simplified through a process of irrelevance reasoning: one of the disjuncts might be provably irrelevant to the problem at hand. Uniform argument disjunctions, on the other hand, are of the form $p(a) \vee q(a)$; Etherington *et al* (1989) suggest that such instances can be eliminated through some subsumption mechanism. Indeed, certain planners such as SIPE employ a class hierarchy for drawing such inferences. Similarly, for objects whose identities are not known, the most one can do is to supply constraints on those objects. Finally, the preferences discussed have not addressed control issues: for example, if one has the goal $p \wedge q$ and q has fewer instances that can bring it about than p, it may be preferable to order the expansion of q first. Once again, an example comes from the pen example: there are fewer rooms across from the conference room then there are offices, hence it seems preferable to plan to get close to the conference room first.

The HOP algorithm

Open-ended planning can be implemented by way of what is referred to in the full paper as a hierarchical open-planning (HOP) algorithm. The algorithm essentially preprocesses the initial world description (applying irrelevance reasoning and partitioning into sets of worlds) and then computes the most preferred behavior given the behaviors stipulated by the designer. HOP then calls on a standard HTN planning algorithm to either elaborate an initial portion of the task statement or does one of the following: extends the initial situation description with default assumptions, executes an initial portion of the plan, or performs an information-gathering action.

We have begun implementation of the HOP algorithm through additions to the SIPE planning system.

Related work

As discussed earlier, one very cautious approach to planning with incomplete information is to plan for contingencies. The Cassandra system is a partial-order contingency planner which can also represent information gathering actions (Pryor & Collins 1996). Open-ended planning tries to instead limit the amount of contingency planning in favor of the more optimistic strategies discussed in this paper. An approach similar to that of Cassandra is taken by the Cypress system (Wilkins *et al.* 1994) which combines the SIPE planner and the Gister system for reasoning under uncertainty (Lowrance 1994). In Cypress, partial plans are sent to Gister for evaluation: in cases in which the initial world description is incomplete, the system finds equivalence classes of states in which the same plan will satisfy the problem goal. Neither of these systems interleaves planning and execution. The Mahinur system is more selective in limiting the number of contingencies which are planned for: the focus is on building sufficient robustness into a planner so that it can recover from plan-execution failures (Onder & Pollack 1997). The SAGE system provides an integrated framework for planning, execution, sensing and replanning in the domain of information access (Knoblock 1995). However, a designer cannot stipulate desired situation-dependent behaviors as in the open-ended planning approach.

The CWA is one very basic approach to dealing with incompleteness. Recently, more flexible notions of the CWA have been suggested: the *localized closed world assumption* (LCWA) limits the CWA to only certain predicates and has been applied to planners such as SNLP to reduce redundant information gathering (Golden, Etzioni, & Weld 1994; Etzioni *et al.* 1992). Methods to represent an agent's incomplete knowledge of the world dates back to Moore's seminal work in reasoning about knowledge and action (Moore 1985). More recently, database representations based on this work have been put forward: a database can be partitioned into factual, unknown, and assumed knowledge (the latter through the LCWA) (Bacchus & Petrick 1998). The default mechanism proposed in this paper is more general than either the CWA or the LCWA.

The general problem of planning under incomplete information has been shown to belong to the next level of complexity just beyond NP completeness (Baral, Kreinovich, & Trejo 1999). It is hoped that the sort of open-ended planning argued for in this paper involving assumptive reasoning and interleaved planning and execution can reduce this complexity; this task is left for future work. The value of defaults to guide search in planning has been proposed by Ginsberg (1991); in open-ended planning, default assumptions are used to instead reduce the initial search space.

An alternative approach to some of the problems discussed in this paper is a decision theoretic one in which the utility of information and the cost of acquiring that information could be traded off. One major challenge for such approaches is that of eliciting complete utility models for reasonably sized domains. Such approaches have also only begun to consider issues related to hierarchical planning (Dearden & Boutilier 1997). Instead, this paper adopts a more qualitative approach. In addition, the focus in decision theory is on goal preferences whereas here the focus is on preference to certain plan-execution behaviors.

The general question of disjunctive reasoning and its complexity has been explored by Borgida and Etherington (1989). Finally, although systems such Graphplan are often referred to as disjunctive planners, they do not address the problem of managing disjunction (through assumptive reasoning, information gathering actions, etc).

Contributions and future work

The framework described in the full paper includes the following elements.

- An extension of HTN planning to include methods for the representation and reasoning of incompleteness, contingencies, defaults, and deductive operators. Although there are systems such as SIPE which make use of the latter for deriving side-effects, these ideas have not appeared in formal HTN descriptions (Erol, Hendler, & Nau 1995).
- A language for expressing preferences to particular plan-execution behaviors. The language reifies mental actions; behaviors can be used to focus planning and execution in profitable ways when an agent's knowledge is incomplete.
- A semantics for mental actions based on a well-understood theory of information change that has been applied to the design of rational agent architectures (Ortiz 1999).
- A preliminary set of useful open-ended planning behaviors.
- A provably correct, hierarchical open-planning (HOP) algorithm that implements these ideas.

The work described in this paper also represents an extension of previous work on agent architectures (Ortiz 1999) in which mental actions were reified as a means of, *inter alia*, grounding the idea of commitment that underlies the notion of intention. That work made use of a richer representation in which time was also reified.

Open-ended planning can be viewed as a member of a class of planners called advisable planners (Myers 1996). Advisable planners have, to date, focussed on processing *plan-content advice* through which a user can constrain the set of acceptable plans for satisfying some goal. Open-ended planning extends this to *process advice* for restricting the manner in which a plan is discovered and executed. For example, suppose an agent is planning a vacation, the agent might specify plan-content advice of the form, "I prefer taking the shuttle to the airport", while process advice might take the form of "I prefer to check that I have my ticket before I board the shuttle." The first form of advice constrains the set of plans produced while the second constrains the manner in which plans are produced (in this case, the agent will not assume possession of the ticket, but rather include an explicit look action during execution to reduce uncertainty).

Future work will include further implementation and testing as well as combining plan-content and process advice. We have chosen the air-campaign planning domain for implementation. We plan to also develop a strategy library of process advice together with provable properties of those strategies relative to characteristics of the domain.

The representation of non-deterministic actions and their use in planning deserves attention. It is well-known that non-deterministic actions can be represented by translating non-determinism into incompleteness of state information (Moore 1985): this observation should be investigated in the context of open-ended planning.

Acknowledgments

I would like to thank Karen Myers for helpful discussions and for suggesting the term "open-ended planning." This work was partially funded under the DARPA JFACC program.

References

Bacchus, F., and Petrick, R. 1998. Modeling an agent's incomplete knowledge during planning and execution. In *Proceedings of Principles of Knowledge Representation and Reasoning*, 432–443.

Baral, C.; Kreinovich, V.; and Trejo, R. 1999. Computational complexity of planning and approximate planning in presence of incompleteness. In *Proceedings of the International Joint Conference in AI*, 948–953.

Borgida, A., and Etherington, D. W. 1989. Hierarchical knowledge bases and efficient disjunctive reasoning. In *Proceedings of the First Conference on Principles of Knoweldge Representation and Reasoning*, 33–43.

Chellas, B. F. 1980. *Modal Logic: An Introduction.* Cambridge University Press.

Dearden, R., and Boutilier, C. 1997. Abstraction and approximate decision theoretic planning. *Artificial Intelligence* 89(1):219–283.

Erol, K.; Hendler, J.; and Nau, D. S. 1995. Complexity results for HTN planning. *Annals of Mathematics and Artificial Intelligence.*

Etzioni, O.; Hanks, S.; Weld, D.; Draper, D.; Lesh, N.; and Williamson, M. 1992. An approach to planning with incomplete information. In *Proceedings of KR-92.*

Ginsberg, M. L. 1991. The computational value of nonmonotonic reasoning. In *Proceedings of the Second Conference on Principles of Knowledge Representation and Reasoning*, 262–268.

Golden, K.; Etzioni, O.; and Weld, D. 1994. Omnipotence without omniscience: Efficient sensor management for planning. In *Proceedings of AAAI-94*, 1048–1054.

Knoblock, C. A. 1995. Planning, executing, sensing, and replanning for information gathering. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence.

Levy, A. Y., and Sagiv, Y. 1993. Exploiting irrelevance reasoning to guide problem solving. In *IJCAI-93*.

Lowrance, J. D. 1994. Evidential reasoning with Gister-CL: A manual. Technical report, SRI International.

McCarthy, J. 1995. Making robots conscious of their mental states. Proceedings of the 1995 workshop on Machine Intelligence.

Moore, R. C. 1985. A formal theory of knowledge and action. In *Formal Theories of the Commonsense World*. Ablex Publishing Corporation.

Myers, K. L. 1996. Strategic advice for hierarchical planners. In *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning*, 112–123.

Onder, N., and Pollack, M. E. 1997. Contingency selection in plan generation. In Proceedings of the Fourth European Conference on Planning.

Ortiz, C. L. 1999. Introspective and elaborative processes in rational agents. *Annals of Mathematics and Artificial Intelligence* 25(1–2):1–34.

Pryor, L., and Collins, G. 1996. Planning for contingencies: a decision-based approach. *Journal of AI research* 4:287–339.

Wilkins, D. E.; Myers, K. L.; Lowrance, J. D.; and Wesley, L. P. 1994. Planning and reacting in uncertain and dynamic environments. *Journal of Experimental and Theoretical AI* 6:197–227.