# A PLANNER FOR REASONING ABOUT KNOWLEDGE AND ACTION

*Douglas E. Appelt*

*Stanford University, Stanford, California*
*SRI International, Menlo Park, California*

## ABSTRACT

This paper reports recent results of research on planning systems that have the ability to deal with multiple agents and to reason about their knowledge and the actions they perform. The planner uses a knowledge representation based on the possible worlds semantics axiomatization of knowledge, belief and action advocated by Moore [5]. This work has been motivated by the need for such capabilities in natural language processing systems that will plan speech acts and natural language utterances [1, 2]. The sophisticated use of natural language requires reasoning about other agents, what they might do and what they believe, and therefore provides a suitable domain for planning to achieve goals involving belief. This paper does not directly address issues of language per se, but focuses on the problem-solving requirements of a language-using system, and describes a working system, KAMP (Knowledge And Modalities Planner), that embodies the ideas reported herein.

## I. WHAT A KNOWLEDGE PLANNER MUST DO

Consider the following problem: A robot named Rob and a man named John are in a room that is adjacent to a hallway containing a clock. Both Rob and John are capable of moving, reading clocks, and talking to each other, and they each know that the other is capable of performing these actions. They both know that they are in the room, and they both know where the hallway is. Neither Rob nor John knows what time it is. Suppose that Rob knows that the clock is in the hall, but John does not. Suppose further that John wants to know what time it is, and Rob knows he does. Furthermore, Rob is helpful, and wants to do what he can to insure that John achieves his goal. Rob's planning system must come up with a plan, perhaps involving actions by both Rob and John, that will result in John knowing what time it is.

We would like to see Rob devise a plan that consists of a choice between two alternatives. First, if John could find out where the clock was, he could go to the clock and read it, and in the resulting state would know the time. So, Rob might tell John where the clock was, reasoning that this information is sufficient for John to form and execute a plan that would achieve his goal. The second alternative is for Rob to move into the hall and read the clock himself, move back into the room, and tell John the time.

*This research was supported by the Defense Advanced Research Projects Agency under contract N00039-79-C-0118 with the Naval Electronic Systems Command. The views and conclusions contained in this document are those of the author and should not be interpreted as representative of the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, or the U. S. Government.*

Existing planning mechanisms such as NOAH [6] or STRIPS [3] are incapable of dealing with this sort of problem. First, to solve this problem a system must reason effectively about propositional attitudes such as *know, believe,* and *want*. Existing planning systems are based on knowledge representations that are not adequate for that purpose. Moreover, they are not equipped to handle the integration of the actions of multiple agents into a single plan. In the solution to the above problem, the first choice consists of actions by both John and Rob. Rob does an informing act, and John moves into the hall and reads the clock. This means that Rob has planned for events to occur that are beyond its immediate control and which involve knowledge about the capabilities of another agent.

The KAMP system solves problems such as the example above. It adopts a knowledge representation based on possible worlds semantics, which is capable of representing the knowledge needed for the task. By reasoning about the knowledge and wants of other agents, KAMP determines what courses of action other agents can be expected to take in the future, and incorporates those actions into its own plans.

## II. REPRESENTING KNOWLEDGE ABOUT BELIEF

It is important that a planner be based on a knowledge representation that is adequate for representing different kinds of facts about knowing, wanting, believing, etc. For example, it may be necessary to represent that someone knows the value of a term, without the system itself knowing what that value is, or the system may need to represent that a person knows ~P as opposed to not knowing whether P. A variety of strategies have been suggested for representing such knowledge, but just representing the knowledge is not sufficient. It is also necessary that the system be able to reason with the knowledge efficiently. Many of the alternatives that have been proposed for representing knowledge are fundamentally lacking in either representational adequacy or efficiency. Moore [5] discusses some of the specific proposals and their shortcomings.

The representation which has been selected for the KAMP system is based on Moore's axiomatization of possible worlds semantics. This approach has a great deal of power to represent and reason efficiently with modal operators, and it is particularly elegant in describing the relation between action and knowledge. Because the design of the planner is largely motivated by the design of the knowledge representation, I will briefly outline Moore's strategy for representing knowledge about belief and how it relates to action. For comparison with a system that uses a different knowledge representation for planning to influence belief, see Konolige and Nilsson [4].

The representation consists of a modal object language that has operators such as *believe* and *know*. This object langauge is translated into a meta-language that is based on a first order axiomitization of the possible worlds semantics of the modal logic. All the planning and deduction takes place at the level of the meta language. In this paper I will adopt the convention of writing the meta language translations of object language terms and predicates in **boldface.**

For example, to represent the fact that John knows P, one asserts that P is true in every possible world that is compatible with John's knowledge. If K(A, w1, w2) is a predicate that means that w2 is a possible world which is compatable with what A knows in w1, and T(w, P) means that P is true in the possible world w, and W0 is the actual world, then the statement that John knows P is represented by the formula:

$$(1)\quad \forall w\, K(\mathbf{John}, W_0, w) \supset T(w, P).$$

This states that in any world which is compatable with what John knows in the actual world, P is true.

Just as knowledge defines a relation on possible worlds, actions have a similar effect. The predicate

$$(2)\quad R(\mathbf{Do}(A, P), w_1, w_2)$$

represents the fact that world w2 is related to world w1 by agent A performing action P in w1. Thus possible worlds can be used in a manner similar to state variables in a state calculus.

Using a combination of the K and R predicates, it is possible to develop simple, elegant axiom schemata which clearly state the relationship between an action and how it effects knowledge. For example, it would be possible to axiomatize an informing action with two simple axiom schemata as follows:

$$(3)\quad \forall w_1 \forall w_2 \forall a \forall b \forall P\, R(\mathbf{Do}(a, \mathbf{Inform}(b, P)), w_1, w_2) \supset [\forall w_3\, K(a, w_1, w_3) \supset T(w_3, P)]$$

$$(4)\quad \forall w_1 \forall w_2 \forall a \forall b \forall P\, R(\mathbf{Do}(a, \mathbf{Inform}(b, P), w_1, w_2) \supset \forall w_3\, K(b, w_2, w_3) \supset \exists w_4[K(b, w_1, w_4) \wedge R(\mathbf{Do}(a, \mathbf{Inform}(b, P)), w_4, w_3)]$$

Axiom (3) is a precondition axiom. It says that it is true in all possible worlds (i.e. that it is universally known) that when someone does an informing action, he must know that what he is informing is in fact the case. Axiom (4) says that it is universally known that in the situation resulting from an informing act, the hearer knows that the inform has taken place. If the hearer knows that an inform has taken place, then according to axiom (3) he knows that the speaker knew P was true. From making that deduction, the hearer also knows P.

## III. USING THE POSSIBLE WORLDS KNOWLEDGE REPRESENTATION IN PLANNING

The possible worlds approach to representing facts about knowledge has many advantages, but it presents some problems for the design of a planning system. Axiomatizing knowledge as accessiblity relations between possible worlds makes it possible for a first order logic deduction system to reason about knowledge, but it carries the price of forcing the planner to deal with infinite sets of possible worlds. Planning for someone to know something means making a proposition true in an infinite number of possible worlds. The goal wff, instead of being a ground level proposition, as is the case in other planning systems developed to date, becomes an expression with an implication and a universally quantified variable, similar to (1).

Another problem arises from the way in which actions are axiomatized. Efficient axioms for deduction require the assumption that people can reason with their knowledge. The effects of some actions manifest themselves through this reasoning process. For example, this occurs in the axiomatization of INFORM given in (3) and (4). Speakers usually execute informing actions to get someone to know something. However, the hearer does not know what the speaker is informing directly as a result of the inform, but from realizing that the speaker has performed an informing act. (See Searle [7]) The only effect we know of for an informing act is that the hearer knows that the speaker has performed the act. When KAMP has a goal of someone knowing something, it must be able to determine somehow that INFORM is a reasonable thing to do, even though that is not obvious from the effects of the action.

A related problem is how to allow for the possiblity that people can reason with their knowledge. If the system has the goal Know(A, Q), and this goal is not achieveable directly, and the system knows that A knows that P ⊃ Q, then Know(A, P) should be generated as a subgoal. Examples of this sort of situation occur whenever Q is some proposition that is not directly observable. If P is some observable proposition that entails Q, then the planner can perform some action that will result in knowing P, from which Q can be inferred. This is the basis of planning experiments.

Since there is a tradeoff between being able to take full advantage of the logical power of the formalism and being able to efficiently construct a plan, a strategy has been adopted that attempts to strike a balance. The strategy is to have a module propose actions for the planner to incorporate into a plan to achieve the current goal. This module can be thought of as a "plausible move generator". It proposes actions that are likely to succeed in achieveing the goal. The system then uses its deduction component to verify that the suggested action actually does achieve the goal.

To facilitate the action generator's search for reasonable actions, the preconditions and effects of actions are collected into STRIPS-like *action summaries.* These summaries highlight the direct and indirect effects of actions that are most likely to be needed in forming a plan. For example, the action summary of the INFORM action would include the hearer knowing P as one of the (indirect) effects of INFORM. The effects of actions as they are represented in the action summaries can be any well formed formula. So it is possible to state effects as implications which can match the implications that are the meta language translations of the Know operator. Using action summaries is not equivalent to recasting the knowledge representation as a STRIPS-like system. The action summaries are only used to *suggest* alternatives to be tried.

To allow the possibility of agents reasoning with their knowledge, KAMP follows the following process whenever a goal involving a knowledge or belief state is encountered: The system tries to invoke an operator that will achieve the goal directly. If this fails, then the system's base of consequent rules is examined to find subgoals that could be achieved and that would allow the agent to deduce the desired conclusion. Although this approach has the advantage of simplicity, the number of subgoals that the planner has to consider at each step can grow exponentially. It seems intuitively correct to consider subgoals in the order of the length of inference that the agent has to go through to reach the desired conclusion. The planner needs good criteria to prune or at least postpone consideration of less likely paths of inference. This is an interesting area for further research.

## IV. UNIVERSAL KNOWLEDGE PRECONDITIONS

When planning actions on a strictly physical level, there are few if any preconditions which can be said to apply universally to all actions. However, when dealing with the knowledge required to perform an action, as well as its physical enabling conditions, there are a sufficient number of interesting universal preconditions so that their treatment by the planner as a special case is warranted. Universal knowlege preconditions can be summarized by the statement that an agent has to have an executable description of a procedure in order to do anything. For example, if an agent wishes to perform an INFORM action, it is necessary for him to know what it is that he is informing, how to do informing, and who the intended hearer is. Since these preconditions apply to *all* actions, instead of including them in the axioms for each action, the planner automatically sets them up as subgoals in every case.

## V. THE REPRESENTATION OF THE PLAN WITHIN THE SYSTEM

The KAMP planner uses a system of plan representation similar to that of Sacerdoti's procedural networks [6]. The major difference is that CHOICE nodes (OR-SPLITS) are factored out into a disjunctive normal form. Since a choice may occur within an AND-SPLIT that affects how identical goals and actions are expanded after the choice, all the choices are factored out to the top level, and each choice is treated as a spearate alternative plan to be evaluated independently, (but perhaps sharing some subplans as "subroutines" with other branches.) As in Sacerdoti's system, nodes can specify goals to be achieved, actions to be performed, or they may be "phantoms", goals which coincidentally happen to already be satisfied. Each node of the procedural network is associated with a world. This world represents the real world at that particular stage in the execution of the plan. At the beginning of the plan, the first node is associated with $W_0$. If the first action is that the robot moves from A to B, then the node representing that action would have an associated world $W_1$ and there would be an assertion of the form R(**Do(Robot, Move(A, B))**, $W_0$, $W_1$) added to the data base.

## VI. CONTROL STRUCTURE

The planner's control structure is similar to that of Sacerdoti's NOAH system. A procedural network is created out of the initial goal. The planner then attempts to assign worlds to each node of the procedural network as follows: First, the initial node is assigned $W_0$, the initial actual world. Then interatively, when the planer proposes that a subsequent action is performed in a world to reach a new world, a name is generated for the new world, and an R relation between the original world and the new world is asserted in the deducer's data base. Then all goal nodes that have worlds assigned are evaluated, i.e. the planner attempts to prove that the goal is true using the world assigned to that node as the current state of the actual world. Any goal for which the proof succeeds is marked as a phantom (achieved) goal.

Next, all the unexpanded nodes in the network that have been assigned worlds, and which are not phantoms, are examined. Some of them may be high level actions for which a procedure exists to determine the appropriate expansion. These procedures are invoked if they exist, otherwise the node is an unstaisfied goal node, and the action generator is invoked to find a set of actions which might be performed to achieve the goal. If an action is found, it is inserted into the procedural network along with its preconditions, both the universal ones and those specific to the particular action. After the nodes are expanded to the next level, a set of critic procedures are invoked, which can examine the plan for global interactions and take corrective action if needed. This entire process is repeated until the plan has been expanded to the point where every unexpanded node is either a phantom goal or an executable action.

## VII. CURRENT STATUS OF RESEARCH

The KAMP planning system has been implemented and tested on several examples. It has been used to solve the problem of John, Rob and the clock cited earlier in the paper. All the critics of Sacerdoti's NOAH have either been implemented or are currently undergoing implementation. Further development of the planner will be dictated by the needs of a language planning and generation system currently under development. It is expected that this langauge generation task will make full use of the unique features of this system.

KAMP is a first attempt at developing a planner that is capable of using the possible worlds semantics approach to representing knowledge about belief. Combining a planner with a very powerful knowledge representation will enable problem solving techniques to be applied to a variety of domains such as language generation, and planning to acquire and distribute knowledge, in which they have played a relatively small role in the past.

## ACKNOWLEDGEMENTS

The author is grateful to Barbara Grosz, Gary Hendrix and Terry Winograd for comments on earlier drafts of this paper

## REFERENCES

[1] Appelt, Douglas E., *Problem Solving Applied to Natural Language Generation,* proceedings of the Annual Conference of the Association for Computational Linguistics, 1980.

[2] Cohen, Philip, *On Knowing What to Say: Planning Speech Acts,* University of Toronto Technical Report #118, 1978.

[3] Fikes, Richard E., and N. Nilsson, *STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving,* Artificial Intelligence, No. 2, 1971.

[4] Konolige, Kurt and N. Nilsson, *Planning in a Multiple Agent Environment,* Proceedings of the First Annual Conference of the American Association for Artificial Intelligence, August, 1980.

[5] Moore, Robert C., *Reasoning About Knowledge and Action,* Massachusetts Institute of Technology Artificial Intelligence Laboratory Technical Report TR-???, 1979.

[6] Sacerdoti, Earl, A Structure for Plans and Behavior, Elsevier North-Holland, Inc., Amsterdam, The Netherlands, 1977.

[7] Searle, John, Speech Acts, Cambridge University Press, 1969.