PLAN GENERATION AND EXECUTION FOR ROBOTICS

Technical Note 209

April 15, 1980

by: Earl D. Sacerdoti, Associate Director

Artificial Intelligence Center
Computer Science and Technology Division

SRI Project 1349 and 8871

This paper was presented at the Workshop on Robotics Research, sponsored by the National Science Foundation, at Newport, Road Island, April 15-17, 1980.

The work reported herein was supported by the Air Force Office of Scientific Research under Contract F49620-79-C-0188, and by the Office of Naval Research under Contract N00014-80-C-0300.

ABSTRACT

Applicability of existing industrial robot systems is limited; truly flexible automation must make use of significant sensory feedback to respond appropriately to each new stimulus. This requires fundamental research in problem solving and the monitoring of plan execution. A number of problems in this area requiring further research are discussed, including dealing with time, planning for parallel execution, planning for information gathering, planning for planning, learning, interactive planning, dynamic plan repair, and distributed robotics.

This paper was prepared for the NSF-sponsored workshop on the research needed to advance the state of robotics, to be held April 15-17, 1980.

PLAN GENERATION AND EXECUTION FOR ROBOTICS*

Earl D. Sacerdoti
Artificial Intelligence Center
SRI International
Menlo Park, California

I INTRODUCTION

A robot device operates by performing a sequence of actions, drawn from a reasonably small repertoire of action types. Industrial robots commercially available today are typically configured to perfom a fixed sequence of actions, cued by some visual or contact stimulus. The action sequence is not subject to any significant alteration; it is performed identically each time the stimulus is presented. Such robots can be called reflex robot systems, since they make a rather inflexible response to a fixed stimulus.

The applicability of such systems is limited; truly flexible automation must make use of significant sensory feedback to respond appropriately to each new stimulus. Thus, a worthwhile goal for robotics research is to develop the capabilities needed to create what we may call instrumental robot systems. The term "instrumental" is used to suggest, by analogy with classical psychology, the mediation of explicit goals and deliberately initiated actions (goal-oriented behavior, in the psychologists' terms) in the performance of the robot system.

* Preparation of this paper was supported by the Air Force Office of Scientific Research under contract F49620-79-C-0188, and by the Office of Naval Research under contract N00014-80-C-0300. Many of the concepts discussed in this paper were developed during discussions with Nils Nilsson over many years.

The development of such instrumental robots will require extensions of current capabilities in many areas, including control systems, mechanics, and sensors. At the core of the capabilities to be developed lies fundamental research in problem solving and the monitoring of plan execution.

II THE RESEARCH BASE

Before we delineate the necessary work in plan generation and execution, it will be helpful to characterize briefly the current state of the art*. Experimental systems have been developed that display the following features (note that no integrated system design yet exists that incorporates all of these capabilities):

- Plans can be generated at multiple levels of detail.
- Plans can be viewed as partially ordered sequences of actions with respect to time.
- Each action is expected to produce a single state change characterized by a single primary effect.
- A plan is not generated at all unless the planner determines that it will be totally successful in meeting all specified goals.
- Simple plans requiring information gathering can be generated.
- Unsophisticated techniques for dynamic repair of unsuccessful plans during execution have been developed.
- Plans can be used to control robot devices with simple use of sensory feedback and simple replanning.

The integration of these existing capabilities into a single plan generation and execution system would in itself constitute a formidable (and worthwhile) research goal.

There are fundamental inadequacies of knowledge in the field, however, that must be filled before instrumental robots can become feasible. The following sections will delineate the areas that, in the

* The reader who is unfamiliar with these subjects can find an introduction to them in [1] and a survey of recent work in [2].

author's opinion, are most critically in need of further research in support of advanced robotics*. They describe differing functional behaviors that a robot system should possess. The plan formation and execution techniques required for one function are often also needed for others; such interrelationships will be noted where they arise.

III DEALING WITH TIME

Existing systems typically treat each action as if it occurred instantaneously. This is clearly inadequate in planning for such activities as locomotion, control of external processes, and interaction with such other active agents as humans, manufacturing equipment, or other robots.

Time must be reasoned about in different ways. First, noninstantaneous actions (such as moving a robot arm from one position to another) must be modeled so that a description of the state of the world at arbitrary instants can be computed. Hendrix [4] performed early work in this area but much remains to be done. A related problem is that of planning not just to _achieve_ a goal state, but to _maintain_ a desired state over time in a dynamic environment.

A second way time must be dealt with has to do with the scheduling of actions, particularly when the actions may be executed in parallel. This requires integrating two rather different styles of reasoning about actions. The first is artificial-intelligence-style structural planning, the determination of strict before-and-after relationships among the actions in a plan. The second is operations-research-style scheduling, the precise determination of starting and ending times for each action. Tate and Daniels [5] began work on a planner that would integrate these styles of reasoning, but many significant problems remain. In particular, typical PERT algorithms compute a set of output values, but provide no rationale for the answers they generate. Thus,

* The need for research in many of these areas has been noted for some time. See for example, [3].

if the structural aspects of a plan change (i.e. the graph structure of the PERT chart is modified) in any way, the complete PERT algorithm must then be rerun. The development of PERT-type algorithms that can reason explicitly about the semantics of plan structures will constitute an important step forward in reasoning about time.

IV PLANNING FOR PARALLEL EXECUTION

Development of robot systems with multiple effectors will permit the performance of multiple actions simultaneously (each by a different effector). Existing plan generation systems have focused on planning for a single action to be taken at a time. Even the nonlinear (i.e. partially ordered, as opposed to strictly ordered, with respect to time) plan structures introduced the author [6] were composed with the intention of executing the actions serially.

Nonlinear plans allow for decomposition of a problem into quasi-independent subproblems, so that the planner can work on each subproblem relatively independently of the others. Plans intended for execution by parallel effectors are indeed nonlinear, but the solutions to the subproblems for each effector require additional kinds of coordination with the other solutions than for the single-effector case. In particular, the equivalent of semaphores or signal-wait pairs may need to be incorporated into the plan structures in order to permit coordination of the parallel effectors.

V PLANNING FOR INFORMATION GATHERING

Plans for robot execution mediated by sensory input must provide for dynamic acquisition during execution of information that was not known or could not be known during planning. This problem has received considerable attention from workers in automated program construction, where the incorporation of conditionals into programs represents exactly this kind of planning. (See, for example, Manna and Waldinger [7]). The problem has received little attention from workers on robot problem solvers, however. The only system known to the author that planned for information acquisition was developed by Weissman [8], and this operated only in a simulated environment. Given the strong base developed for program generation applications, it is likely to be relatively easy to incorporate information gathering into robot planning and execution systems.

VI PLANNING FOR PLANNING

As the number of activities engaged in by a planning system grows, the planner's control structure must become more flexible. At a certain point it becomes economical to use the planner itself, operating on a set of potential actions that the planner can perform, to determine what to do next at each stage while the plan is being developed. Such a scheme has a number of highly attractive features, including the ability to integrate gracefully planning, execution, and information gathering. Such a metaplanner must have very well-integrated capabilities to generate plan steps and to initiate and monitor their execution, since the planner cannot usually plan very far ahead about the planning process itself.

Stefik [9] has just completed a Ph.D. thesis that describes a solid first attempt at such a multilevel planner. It is the author's personal opinion that development of instrumental robots absolutely requires such integrated systems for planning and monitoring the problem-solving process as they plan and monitor the solutions to the system's real-world problems.

VII LEARNING; USING A PLAN DATA BASE

When a problem solver generates a plan from primitive actions, it is essentially reasoning from first principles about the interactions among the actions in the domain. Without a mechanism for learning action sequences and using them appropriately in subsequent planning, a problem solver will be doomed to reasoning from first principles forever--which is an inefficient, not to mention boring, prospect.

Several functions are involved in learning and reusing plans. First, a successful plan for achieving some particular goal in some particular world model must be parameterized to be applicable for dealing with objects other than those present in the original problem. Second, the applicability tests for the learned plan must be relaxed so that it can be employed in as wide a range of initial conditions as possible. Third, the collection of learned plans must be appropriately indexed so that just those plans that are potentially relevant may be selected efficiently from a large plan data base. Fourth, the plan data base must be managed; appropriate criteria must be applied to determine when a plan is worth saving in the data base, and when a plan should be deleted for lack of use or because it is subsumed by another.

The first function was reasonably dealt with in the STRIPS problem solver [10], where a new operator, called a macrop, was created upon completion of a plan. The macrop contained a parameterized version of the original plan, and was encoded in a data structure called a triangle table that permitted any subsequence of the original plan to be selected as a relevant operator for achieving any given goal. Unfortunately, a macrop with more than a few steps has a set of preconditions so large that it is both very expensive to test if the macrop is relevant in a given situation and rather unlikely that the macrop will indeed be relevant. After a number of macrops were added to the system, it typically operated less efficiently than when only the basic operators were available. Thus, the last three functions cited above were not fulfilled. The author's work on hierarchical planning [11] was originally undertaken to deal with these functions. However, expansion of an abstracted macrop into full detail encounters problems of plan repair (discussed below) and has never yet been accomplished.

VIII INTERACTIVE PLANNING

Research in automatic problem solving has resulted in planning systems that perform fully automatically but have limited capability. The plan generation ability of such systems could be significantly extended by permitting human participation. Two key functions of automated planning systems can be greatly improved by incorporating a human planner's inputs. First, a human can control the search to focus on what he perceives to be reasonable lines of attack, integrating the overall search picture in a way automated systems cannot do today. Second, he can exercise judgment based on knowledge extrinsic to the automatic system in those situations in which the system would normally make an arbitrary guess. Today's planning systems either guess or avoid making any commitment when they encounter a situation in which, all things being equal, there is no reason to prefer one alternative over another. The human is potentially aware of many additional factors beyond the scope of the automatic planner's knowledge base, and so can provide substantial guidance.

Development of an interactive planning system requires research on flexible control structures amenable to human mediation, on techiques for presentation of plan information to humans, and on acquisition of plan information from humans. Research on such a system is just beginning at SRI International.

IX DYNAMIC PLAN REPAIR

The most self-evident robot application for automatic problem solving is in automatic recovery from unexpected minor deviations in a planned sequence of actions. This is best done not by replanning when an error is discovered but rather by patching up the abortive plan.

At least three approaches have been explored to assess where a plan went wrong and repair it appropriately. The author [6] used an optimistic execution-monitoring scheme that performed no error checking until some action failed, and then developed a new subplan to establish the preconditions of the unsuccessful action. Davis [12] developed a very cautious system that included many checks for errors as a plan was executed, and invoked specialized procedures upon encountering each type of error. Fikes [13] employed a scheme whereby the world model was checked after each action was executed; the latest remaining executable step was executed next (thus taking advantage of fortuitous actions by others). If no step was executable, the system tried to achieve a state in which any step was executable, and thus return to the solution path. The first of these systems is general, but of limited effectiveness; the second is effective, but requires much special-purpose knowledge for each potential interaction among (possibly unsuccessful) actions. The third is expensive since extensive checking of the state of the world is carried out after each action is taken.

More work is required to define new techniques both for diagnosing errors efficiently and for repairing plans effectively. For applications in which a plan is developed once and then executed many times, it is possible that a postplanning, preexecution phase could be used to analyze potential interactions and errors and automatically develop specific error recovery functions similar to those used in Davis' approach.

X DISTRIBUTED ROBOTICS

The initial robot systems placed in factories and elsewhere are either stand-alone systems or consist of multiple processors, effectors, and sensors contolled by a central computer. It is likely that as more robots are emplaced, these systems will need to communicate and cooperate. A worthwhile research goal for the immediate future is thus the development of what are in essence distributed robots (abbreviated here as "disbots"). Disbots can be characterized as consisting of a set of independent problem-solving systems, located aboard spatially distributed specialists (with specialized sensors, effectors, and, possibly, computational capabilities), organized so as to operate with even limited intermodule communication.

Several workers are currently engaged in research on aspects of distributed problem solving; among these are Lesser and Erman [14], Smith [15], and groups at the University of Massachusetts [16], RAND [17] and SRI [18]. The development of disbots entails most of the novel planning features cited in the preceding sections, plus several additional ones. Thus, it provides a good potential vehicle for experimenting with solutions to most of the research problems arising in plan generation and execution. Following is a brief discussion of some of the issues that must be resolved in the development of techniques for distributed robotics.

A. Planning Issues

Specialist robots must each be capable of hierarchical planning in a dynamic world. They must be capable of maintaining as well as achieving goals. They must be able to plan for information acquisition, either from local sensors or by communication with other specialists. (Each of these points has been discussed in previous sections.)

B. Control Issues

The group of robots must be capable of dynamic self-organization to carry out particular tasks in particular environments. The coordination of asynchronous processors at the level of goals and actions (as opposed to the more standard level of machine primitives and function calls) requires the integration of techniques from artificial intelligence, more traditional computer science, and, perhaps, from other disciplines such as management science and sociology as well.

C. Basic Representation Issues

Traditional problem solvers operated on a world model that was always believed to be correct. In a disbot environment there may be several independent problem solvers, each believing a different world model to be the correct one. To interact with one another, each probem solver will need the ability to reason about the knowledge, beliefs, goals, and plans of the others. Some initial work in this area has been performed at an abstract level by Moore [19] and Weyhrauch [20]; such techniques have not yet been integrated into a distributed problem solving system.

D. Communication Issues

For disbots to communicate within reasonable bandwidth constraints, communication must be seen from a novel perspective. It must be viewed not at the level of protocols and message formats, but rather as a knowledge-sharing activity with significant similarities to human interactions in natural language. Communicating processes must share, update, and maintain common background information of several types, including

* The subject domain -- the actors, objects, relationships, and ongoing events.

* The context of each current interaction -- knowledge of the current environment, the history of previous states of the world, the structure of previous dialogue, and special characteristics of the entities participating in the dialogue.

- The "rules of the game" of communicating -- In effect, very-high-level protocols enabling the coding and decoding of the goals behind the message, rather than just the content of the message. This includes means of specifying and describing physical interactions, changes in the knowledge state of of the hearer (in particular, what the hearer believes the current topic of discussion to be), the structure of the discourse (i.e., maintaining coherence), and control issues (such as who can interrupt whom under what circumstances).

E. Plan Execution Issues

Because disbots will operate in a very dynamic environment with numerous active agents, many new execution-monitoring problems arise. Each specialist processor must be able to monitor the effects of its own actions, both for information gathering and to verify that requests and information sent to other processors have been understood and acted upon. Each processor must also monitor the effects of its counterparts' actions, and perform a plan recognition task to interpret those actions. Each processor must plan for potential interactions among its sensor, effector, and communications capabilities.

Mechanisms must be developed for modification of distributed plans when things go awry. In addition, provision must be made for decisions during execution to shift control of particular activities from one processor to another.

XI CONCLUSION

We have cited a large number of research issues that must be dealt with in developing practical robot systems that are flexible and adaptable enough to be called "instrumental" robots. If such robot systems are to be developed in the next decade, a significantly increased effort appears warranted in basic research devoted to automated common-sense reasoning, plan generation and execution. A suggested focus for such a program of research is the development of distributed robot systems. Because the concept of a disbot provides a novel perspective on a set of difficult problems, a distributed robot project would likely make significant contributions both in conceptual progress and in ultimate practical utility.

REFERENCES

1. N.J. Nilsson, *Priciples of Artificial Intelligence* (Tioga Publishing Co., Palo Alto, California, 1980).

2. E.D. Sacerdoti, "Problem Solving Tactics," *Proc. Sixth International Joint Conference on Artificial Intelligence*, pp. 1077-1085, Tokyo, Japan (August 1979).

3. R.E. Fikes, P.E. Hart, and N.J. Nilsson, "Some New Directions in Robot Problem Solving," *Machine Intelligence 7*, pp. 405-430 (Edinburgh University Press, Edinburgh, Great Britain, 1972).

4. G.G. Hendrix, "Modeling Simultaneous Actions and Continuous Processes," *Artificial Intelligence*, Vol. 4, pp. 145-180 (1973).

5. A. Tate, "Generating Project Networks," *Proc. Fifth International Joint Conference on Artificial Intelligence*, pp. 888-893, Cambridge, Massachusetts (August 1977).

6. E.D. Sacerdoti, *A Structure for Plans and Behavior* (Elsevier North-Holland, New York, 1977).

7. Z. Manna and R. Waldinger, "Synthesis: Dreams ==> Programs," *Proc. IEEE Transactions on Software Engineering*, Vol. SE-5, No. 4, pp. 294-328 (July 1979).

8. S.J. Weissman, "On a Computing System for Planning and Execution in Incompletely Specified Environments," Coordinated Science Laboratory Report R-741, University of Illinois, Urbana, Illinois (September 1976).

9. M.J. Stefik, "Planning with Constraints," Stanford University Computer Science Department Report No. STAN-CS-80-784, Stanford, California (January 1980).

10. R.E. Fikes, P.E. Hart, and N.J. Nilsson, "Learning and Executing Generalized Robot Plans," *Artificial Intelligence*, Vol. 3, No. 4, pp. 251-288 (Winter 1972).

11. E.D. Sacerdoti, "Planning in a Hierarchy of Abstraction Spaces," *Artificial Intelligence*, Vol. 5 No. 2, pp. 115-135 (Summer 1974).

12. P.R. Davis, "Using and Re-using Partial Plans," Coordinated Science Laboratory Report R-772, University of Illinois, Urbana, Illinois (June 1977).

13. R.E. Fikes, "Monitored Execution of Robot Plans Produced by STRIPS," Proc. IFIP Congress '71, Ljubljana, Yugoslavia (1971).

14. V.R. Lesser and L.D. Erman, "An Experiment in Distributed Interpretation," Information Sciences Institute, University of Southern California, Marina del Rey, California (April 1979).

15. R.G. Smith, "A Framework for Distributed Problem Solving," Proc. Sixth International Joint Conference on Artificial Intelligence, Tokyo, Japan (August 1979).

16. V.R. Lesser, ed., "Working Papers in Distributed Computation I; Cooperative Distributed Problem Solving," Distributed Computation Group, Computer and Information Science, University of Massachusetts (July 1979).

17. F. Hayes-Roth and R. Wesson, "Distributed Intelligence for Situation Assessment," Report N-1447-ARPA, RAND, Santa Monica, California (January 1980).

18. E.D. Sacerdoti, "What Language Understanding Research Suggests About Distributed Artificial Intelligence," Proc. Distributed Sensor Nets Workshop, pp. 8-11, Carnegie-Mellon University (December 1978).

19. R.C. Moore, "Reasoning about Knowledge and Actions," Ph.D. Dissertation, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts (1979).

20. R. Weyhrauch, "Prolegomena to a Theory of Mechanized Formal Reasoning," Memo AIM-315, Stanford Artificial Intelligence Laboratory, Stanford, California (December 1979).