Execution-time Response:
Applying plans in a dynamic world

Kristian J. Hammond
Artificial Intelligence Laboratory
Department of Computer Science
The University of Chicago

Phil Agre
Artificial Intelligence Laboratory
Department of Computer Science
The University of Chicago

Richard Alterman
Department of Computer Science
Brandeis University

Reid Simmons
School of Computer Science
Carnegie Mellon University

R. James Firby
NASA: Jet Propulsion Labs

Abstract

This panel is aimed at the issue of how to use and modify plans during the course of execution. The relationship between a plan and the actions that an agent takes has generated a great deal of interest in the past few years. This is, in part, a result of the realization that planning in the abstract is an intractable problem and that much of the complexity of behavior is best understood in terms of the complexity of the environment in which that behavior occurs.

This panel presents five distinct personalities and approaches to this problem:

- Agre looks at replacing "planning" with situated activity. In particular, he has been considering the problems involved with the reference assumptions of classical planning.
- Firby's hierarchical planner has primitive actions that are instantiated at execution-time. The execution of these primitives generates information that can be used to guide selection of later operators.
- In Alterman's model of run-time adaptation, the executive responds to failures by using external cues to move between alternative steps or approaches stored in an existing network of semantic/episodic information.
- Simmons has been exploring techniques to create robust, reactive systems that can handle multiple tasks in spite of the robot's limited sensors and processors. His approach takes full advantage of the resources that the robot does have. This includes using hierarchical coarse-to-fine control strategies, using concurrency whenever feasible, and explicitly focusing attention on the robot's tasks and monitored conditions.
- Hammond suggests a theory of *agency* which casts planning as embedded within a memory-based understanding system connected to the environment. Within this approach, the environment, plan selections, decisions, conflicts and actions are viewed through the single eye of situation recognition and response.

The Role of Plans in Activity

Phil Agre
The University of Chicago
Artificial Intelligence Laboratory

The notion of a plan has long been central to computational research on action. The terminology and characteristic hypotheses of 'planning' research received their most influential early formulation in Miller, Galanter, and Pribram's book *Plans and the Structure of Behavior* (1960), henceforth MG&P. MG&P's central thesis was that the observable structure of an organism's behavior results from its executing a Plan which has that same structure. MG&P demonstrated that a wide variety of phenomena could be assimilated to this model. The first system which operated by constructing and executing plans was Strips, built by Fikes and Nilsson (1971) using the problem-space methodology developed by Newell and Simon (1963). A great deal of research has been conducted within this framework (Chapman, 1987; Georgeff, 1987), recently leading to industrial applications (Wilkins, 1988; 1989).

The argument of MG&P contains a profound ambiguity. It can be viewed as running together two different accounts of plans and execution. On the first account, plans are primarily retrieved from a library, or constructed from scratch, and are executed whole. This account offers an explanation for the structure of behavior, but it portrays the agent as almost entirely inflexible. On the second account, an agent assembles its plans incrementally, so that sequences of actions need not be mapped out ahead of time. This account offers an explanation of how an agent might be capable of dynamically adapting its behavior to circumstances as they arise, but it does not explain why an organism's behavior has its observed structure. MG&P's two accounts of planning might be compatible in some complex combination, but they cannot simultaneously explain both the observable structure and the flexible adaptation of behavior.

This ambiguity within MG&P's argument has been fateful for subsequent research, particularly in the last five years as various groups have worked to build agents which are capable of conducting sensibly organized goal-directed action in environments characterized by unpredictability, uncertainty, and change. Much of this work has been conducted within the vocabulary of plans and their execution, trying to find an acceptable combination of the two approaches that MG&P introduced (Georgeff and Lansky, 1987; Firby, 1987). Much other work, though, has dispensed with the notion of a plan altogether, treating continual interaction with the environment as a central phenomenon (Agre in preparation; Brooks, 1986). For this work, the observable structure of an organism's behavior is an emergent property of these interactions and not the causal product of the execution of a plan. Such is the approach that I developed in the notion of running arguments (Agre, 1985) and that David Chapman and I took in our work on the Pengi system (Agre and Chapman, 1987).

People regularly make and use plans, of course, in the ordinary vernacular sense of the word; the introduction of an alternative to MG&P's account of action reopens the question of what plans actually are. They are not like computer programs, since their use regularly involves a considerable amount of interpretive effort as well as rearrangement, interpolation, and substitution of the actions the plan represents. Plans are moreover not a unified phenomenon; they occur in a wide variety of activities and social contexts, from cooking (Scher, 1984) to office procedures (Suchman, 1983) to navigation (Gladwin, 1970). In each case, though, it is best to view plans as resources for the participants in an activity, and not (as with MG&P) as fully specifying or causally engendering the activity (Suchman, 1987).

In recent work, Chapman and I have begun exploring a view of plans as communication in natural language (Agre and Chapman in press). People who use directions or instructions to find subway stations or play video games interpret them within the cultural background that they share with other participants in the activity. Moreover, they interpret them indexically, in terms of their ongoing situations as they understand them at the moment they turn to the plan for advice. Computational research can be expected to offer insight on these processes by investigating the architectural consequences of various ways of using plans and learning from their use, but such research must be informed by sociologically sophisticated views of the circulation and deployment of representational materials in human societies.

Adaptive Planning[1]

Richard Alterman
Computer Science Department
Brandeis University

There were two key features of early models of planning in artificial intelligence. The first was that the planner did not have a memory of previous planning episodes. This meant that the planner was always planning from scratch, constructing a plan out of a small set of operators. Second, these early models of planning almost entirely separated the planning and acting phases. A robot given some task would construct a plan from scratch to achieve that task. Then it would turn the plan over to an execution monitor that would supervise the robot as it went through the steps of the plan. This model of planning and acting proved undesirable because, in general, it failed to provide for the contingencies that might arise. This critique of early models arose from work on case-based planning (Hammond, 1990 and Kolodner and Simpson, 1989), reactive planning (Firby, 1987) and situated activity (Agre and Chapman, 1987 and Suchman, 1987).

Adaptive planning (Alterman, 1988) was an early effort to deal with the problems of traditional models of planning. An adaptive planner is a common sense planner. It has a memory of previous plans (routines) and retrieves a plan from that memory that seems to match the situation-at-hand. It then adapts that plan (improvises) during the period of engagement. For example, the first time I ride the NYC Subway, I do not plan from scratch; rather, I use my knowledge of riding BART (Bay Area Rapid Transit) as a basis for constructing an interpretation of the actions I should take.

I am currently looking at reasoning about the usage of mechanical devices and the role of instructions. A planner may adapt a known routine to the situation-at-hand, but, if difficulties arise, it has access to instructions. The specific difficulty provides a concrete context for those instructions. Much of this work is informed by my work on semantic memory (Alterman, 1989) and by the lexical semantic theory of Pustejovsky (Pustejovsky, forthcoming) as it impacts spatial and deictic terms.

With Roland Zito-Wolf, I am also looking at extending the adaptive planning model to handle plan learning. We assume that planners have *habitats*—places where they normally plan and act (e.g. home, the office, hotels). Thus plan learning involves a mixture of teasing out descriptions of the planner's habitats while extending plans to cope with new contingencies. Over a history of engagements the planner gradually settles, and re-settles, into customized routines for its habitats. An important advantage of the adaptive planning framework for learning is that learning is fail-safe, since incomplete or incorrect learning is backed-up by normal adaptive functioning.

---

[1] This work was supported in part by the Defense Advanced Research Projects Agency, administered by the U.S. Airforce Office of Scientific Research under contract #F49620-88-C-0058.

Task Directed Adaptive Execution

Robert James Firby
Jet Propulsion Laboratory

A robot plan is usually viewed as a list of primitive robot actions which are assembled in advance and then executed one after another. However, in real domains, a plan must have more structure if it is to cope with the myriad unpredictable details that it will encounter during execution. Adding such structure to a plan involves more than augmenting the primitive plan representation; it requires a complete model of plan interaction with the world. A planner cannot know in advance all of the sensing and control actions that will be required to achieve its goals because it cannot maintain a complete, detailed model of the situations that it will encounter. Most sensing and control decisions must be suspended until execution time. Therefore, the notion of a plan no longer makes sense without a theory of how it will be executed.

The RAP adaptive execution system is a theory of plan representation and execution. The system assumes an incomplete world model and relies on program-like reactive action packages (RAPs) to carry out sketchy plans premised on that model. A plan consists of a list of tasks rather than primitive actions. Each task contains three major components: a satisfaction test, a window of activity, and a set of execution methods that are appropriate in different circumstances. Plan execution proceeds by selecting an unsatisfied task and choosing a method to achieve it based on the current known world state. A task may be executed as many times as necessary to keep it satisfied while it is active. Since decisions on action selection and sensor deployment are made while the task is "situated" in the real world, execution monitoring is an intrinsic part of the execution algorithm, and the need for separate replanning on failure disappears.

The RAP system appears to offer an effective way to cope with the limitations imposed by real sensors, real actuators, and the incomplete understanding of complex domains.

Robust Robots with Limited Resources

Reid G. Simmons
School of Computer Science
Carnegie Mellon University

A prevalent approach to building mobile robot systems is to have the system continually monitor all (relevant) aspects of the environment, and use what are essentially stimulus-response rules to decide what to do next. Getting the robot to perform a new task typically involves adding more sensors and/or processors. While this approach produces very reactive systems, it does not scale well as the tasks become more complex and numerous.

We are exploring techniques to create robust, reactive systems that can handle multiple tasks in spite of the robot's limited sensors and processors. To succeed, our approach tries to take full advantage of the resources that the robot does have. This includes using hierarchical coarse-to-fine control strategies, using concurrency whenever feasible, and explicitly focusing attention on the robot's tasks and monitored conditions.

We have developed the Task Control Architecture (TCA) to support the creation of such systems. TCA is a distributed system with central control that can construct and manipulate hierarchical plans, allocate and manage user-defined resources, monitor selected conditions, and handle exceptions. Robot-specific processes (such as controllers, planners, and vision processes) communicate with one another through TCA and use the TCA mechanisms to schedule and synchronize their activities.

TCA is currently in use on two testbeds — a prototype of the six-legged CMU Planetary Rover, and the Hero, an indoor mobile manipulator, whose tasks include collecting cups from our lab's floor, retrieving printer output, delivering objects, and recharging itself when necessary. In implementing the Hero and Rover systems, several simple, but effective, organizing principles have emerged for taking full advantage of the robot's available resources.

Hierarchy: Hierarchy is effective for planning, monitoring, and handling exceptions. Our system plans only to the level of detail warranted by its current knowledge of the environment, deferring the remaining details to be filled in at execution time. The Hero system uses coarse-to-fine sensing strategies. For example, the system uses its 2D vision system to detect approximately cup-shaped regions, which triggers tasks to approach and map the objects with a wrist-mounted sonar to determine if they in fact match the robot's model of a cup.

Concurrency: The distributed nature of TCA is used to exploit opportunities for concurrency. These include interleaving of planning and execution, and asynchronous pre-processing of visual data. However, our systems do not have sufficient sensors or computational resources to continually monitor all conditions. TCA handles this by enabling processes to specify the temporal intervals during which selected conditions should be monitored, and the frequency at which they should be polled. For optimal performance, these frequencies should be based on the likelihood of the monitored condition occurring, the urgency for response, and the time needed to react.

Focus of Attention: Since it is unreasonable to expect the system to monitor all possible conditions or to plan for all tasks at once, the robot must explicitly maintain a focus of attention. The resource mechanism of TCA is used to maintain the focus of attention on the currently active tasks. Multiple tasks can be handled concurrently as long as they use separate resources; when contention for resources occurs, the associated tasks must be prioritized. While TCA currently uses only a simple, pre-programmed prioritization scheme, we are starting to explore how the robot could make its own prioritization decisions by reasoning about models of its capabilities, limitations, and the relative utility of the tasks. For example, if the Hero's battery monitor warns of a low charge, the robot should decide whether it can afford to complete its current task, or whether it should immediately proceed to the charger.

Memory and Agency[2]

Kristian J. Hammond
The University of Chicago
Artificial Intelligence Laboratory

Increasingly, the study of planning is being recast as the broader study of planning, action, and understanding. The particular approach that we are taking to this study casts planning as embedded within an understanding system connected to the environment. This approach allows us to view plan selection, conflict resolution, and action through the single eye of situation assessment. Together with the use of episodic memory as the vehicle for understanding, this view leads naturally

---

[2] This work was supported in part by the Defense Advanced Research Projects Agency, monitored by the Air Force Office of Scientific Research under contract F49620-88-C-0058, and the Office of Naval Research under contract N0014-85-K-010.

to an ability to learn from both planning and execution. In this paper, we draw an outline of *agency*, our model of the relationship between planning and action.

Our model of the relationship between planning and action is a complete theory of *agency.* Our theory of agency rises out of three pieces of work: Schank's structural model of memory organization (Schank, 1982), our own work in case-based planning and dependency-directed repair (Hammond, 1986), and the work of Martin and Riesbeck in Direct Memory Access Parsing (Martin 1989). Our model has been articulated in two programs, TRUCKER and RUNNER (Hammond, Marks, and Converse, 1988; Hammond, 1989).

Our original objective was to capture the ability of an agent to suspend goals, yet still recognize execution-time opportunities to satisfy them. We used a single set of memory structures both to store suspended goals and to understand the agent's circumstances in the world. In response to a blocked goal, the agent's first step was to do a planning-time analysis of the conditions that would favor the satisfaction of the goal. The agent then *suspended* the goal in memory, indexed by a description of those conditions.

During execution, the agent performed an ongoing "parse" of the world in order to recognize conditions for action execution. Following DMAP (Martin, 1989), this parse took the form of marker passing through episodic memory. Because suspended goals were indexed in the same memory used for understanding the world, the goals were activated when the conditions favoring their execution were recognized. Once active, goals would be reevaluated in terms of the new conditions. If they fit into the current flow of execution, they would be pursued. Otherwise, they would be suspended again.

We called the initial model *opportunistic memory* because the agent's recognition of opportunities depended on the nature of its episodic memory structures. Having turned to the broader issues of integrating planning and action, we now refer to our work as the study of *agency.*

Our theory of agency accounts for the spawning of goals, the selection of plans, and the execution of actions. Like DMAP, our theory relies on a memory organization defined by part/whole and abstraction relationships. Activations from features in the environment are passed up through abstraction links, and predictions are passed down through partially active concepts.

To accommodate action, we have supplemented DMAP with the notion of PERMISSIONS and POLICIES. PERMISSIONS are handed down the parts of plans to the operators they include. The only actions that take place are those that are PERMITTED by the activation of the operators that are associated with them. Following (McDermott, 1978), POLICIES are statements of ongoing goals of the agent. They may take the form of maintenance goals, such as "Glasses should be in the cupboard" or "Always have money on hand." The only way in which goals can be generated is out of the interaction between POLICIES and environmental features.

Most of the processing takes the form of recognizing circumstances in the external world in the context of the policies, goals and plans of the agent. Goals, plans, and actions interact with each other and with the environment as follows:

- Features in the environment interact with POLICIES to spawn goals.
- Goals and environmental features combine to activate plans already in memory.
- Operators are permitted by plans and are associated with the descriptions of the world states appropriate to their performance. Once a set of features has an operator associated with it,

that set of features (in conjunct rather than as individual elements) is now predicted and can be recognized.

- Operators are specialized into actions by features in the environment and by internal states of the system. As with Firby's RAPs (Firby, 1989), particular states of the world determine particular methods for each general operator.
- Conflicts between actions are recognized and mediated by the same mechanism that parses the world.
- Suspended goals are associated with the descriptions of the states of the world that are amenable to their satisfaction.

This theory of agency not only bridges the gap between planning and execution, but approaches a start-to-finish model of behavior in the world. Our goal is a content theory of agency. We see this architecture as simply the vehicle for that content.

References

Philip E. Agre, Routines, AI Memo 828, MIT Artificial Intelligence Laboratory, 1985.

Philip E. Agre and David Chapman, Pengi: An implementation of a theory of activity, *Proceedings of the Sixth National Conference on Artificial Intelligence*, Seattle, 1987, pages 196-201.

Philip E. Agre and David Chapman, What are plans for?, in Pattie Maes, ed., *New Architectures for Autonomous Agents: Task-level Decomposition and Emergent Functionality*, MIT Press, Cambridge MA, in press.

Philip E. Agre, *The Dynamic Structure of Everyday Life*, Cambridge University Press, Cambridge, in preparation.

Richard Alterman. Event Concept Coherence. *Semantic Structures*, Editor David Waltz, LEA, 1989.

L. Birnbaum and G. Collins. Opportunistic planning and Freudian slips. In *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, Boulder, CO, 1984.

Rodney A. Brooks, A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation* **2**(1), 1986, pages 14-23.

David Chapman, Planning for conjunctive goals, *Artificial Intelligence*, **32**(3), 1987, pages 333-377.

Richard Fikes and Nils Nilsson, Strips: A new approach to the application of theorem proving to problem solving, *Artificial Intelligence* **2**(3), 1971, pages 189-208.

R. James Firby, An investigation into reactive planning in complex domains, *Proceedings of the Sixth National Conference on Artificial Intelligence*, Seattle, 1987, pages 202-206.

R. James Firby. Adaptive execution in complex dynamic worlds. Research Report 672, Yale University Computer Science Department, 1989.

Michael P. Georgeff, Planning, in Joseph F. Traub, Barbara J. Grosz, Butler W. Lampson, and Nils J. Nilsson, eds, *Annual Review of Computer Science* **2**, Palo Alto, CA, 1987, pages 359-400.

Michael P. Georgeff and Amy L. Lansky, Reactive reasoning and planning, *Proceedings of the Sixth National Conference on Artificial Intelligence*, Seattle, 1987, pages 677-682.

Thomas Gladwin, *East is a Big Bird*, Harvard University Press, Cambridge MA, 1970.

Kristian Hammond. *Case-Based Planning: Viewing Planning as a Memory Task*, volume 1 of *Perspectives in Artificial Intelligence*. Academic Press, 1989.

Kristian Hammond. Opportunistic memory. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*. IJCAI, 1989.

Kristian Hammond. Explaining and repairing plans that fail. *Artificial Intelligence Journal*, In Press.

Kristian Hammond, Tim Converse, and Mitchell Marks. Learning from opportunities: Storing and reusing execution-time optimizations. In *Proceedings of the Seventh Annual Conference on Artificial Intelligence*, pages 536–40. AAAI, 1988.

Janet Kolodner and Robert Simpson. The MEDIATOR. *Cognitive Science Journal,* 13:507-549, 1989.

Charles E. Martin. *Direct Memory Access Parsing.* PhD thesis, Yale University Department of Computer Science, 1989.

D. McDermott. Planning and acting. *Cognitive Science*, 2, 1978.

George A. Miller, Eugene Galanter, and Karl H. Pribram, *Plans and the Structure of Behavior*, Henry Holt and Company, New York, 1960.

Allen Newell and Herbert A. Simon, GPS: A program that simulates human thought, in Edward A. Feigenbaum and Julian Feldman, eds, *Computers and Thought*, McGraw-Hill, New York, 1963, pages 279-296.

James Pustejovsky. *The Generative Lexicon.* MIT PRESS, forthcoming.

Roger C. Schank. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People.* Cambridge University Press, 1982.

Bob Scher, *The Fear of Cooking*, Houghton-Mifflin, Boston, 1984.

Lucy Suchman, Office procedures as practical action: Models of work and system design, *ACM Transactions on Office Information Systems* 1(4), 1983, pages 320-328.

Lucy Suchman, *Plans and Situated Action*, Cambridge University Press, Cambridge, 1987.

David E. Wilkins, *Practical Planning: Extending the Classical AI Planning Paradigm*, Morgan Kaufmann Publishers, Los Altos CA, 1988.

David E. Wilkins, Can AI planners solve practical problems?, Technical Report 468, SRI International Artificial Intelligence Center, 1989.