Integrating Adaptation with Planning to Improve Behavior in Unpredictable Environments

Adele E. Howe

COINS Technical Report 90-29

Experimental Knowledge Systems Laboratory
Department of Computer and Information Science
University of Massachusetts, Amherst, MA 01003

Integrating Adaptation with Planning to Improve Behavior in Unpredictable Environments[1]

COINS Technical Report 90-29

Adele E. Howe

Experimental Knowledge Systems Laboratory
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

Abstract

As research in planning has progressed from static well-defined domains to dynamic unpredictable domains, planning and acting have become more closely coupled, and adaptability has become increasingly important. Because a planner cannot predict with certainty the outcome of actions in these domains, planners must adapt on-going plans in response to failures. Constructing plan knowledge bases for these domains is costly and difficult to perfect; so, a planner should adapt its own model of how to act, how far ahead to plan, and how often to assess its progress in response to changing conditions.

---

[1]This paper appeared, without the appendix, in the Working Notes of the 1990 AAAI Spring Symposium on Planning in Uncertain, Unpredictable or Changing Environments.

Why is "good" planning not always enough?

Good planning is enough when plans can be prevented from failing. Plans do not fail when they do not commit to action in the future or when their operating environments are predictable and static. They may not fail in domains in which it is possible to enumerate contingencies. Even the best laid plans are likely to fail in domains with multiple agents, resource limitations (particularly time pressure), unpredictability, and/or complex dynamics. These domains usually cannot be characterized by tractable domain models and are fraught with subtle interactions between the agents and their environment.

Crisis management domains, such as forest fire fighting and oil spill containment, exhibit most of these characteristics. The goal of planners in these domains is to contain an environmental process that occupies an increasingly large geographic area and behaves somewhat unpredictably. The actions available to agents operating in these domains produce localized, limited effects, thus requiring the coordination of many agents over a large area and a large span of time. The Phoenix project provides just such a domain in the form of a simulation of forest fires in Yellowstone National Park (Cohen et al. 1989). The simulation and basic agent architecture supported by Phoenix constitute the basis for these explorations into adaptable planning.

What is adaptable planning?

Because of the complexity and unpredictability of the environment, plans may not progress as intended and cannot be constructed to include contingencies for all plan-damaging events.The agent needs to adapt to changing conditions and to adapt its model of how to act through experience with its environment. Thus, we can distinguish two senses of adaptation: responsive and impressionable. A *responsive* planner responds to environmental changes by modifying plans in progress. An *impressionable* planner remembers the results of plan modifications and so adapts behaviors over extended periods of time.

The two kinds of adaptation increase flexibility and reduce brittleness in the underlying planner. The Phoenix planner is a lazy skeletal expansion planner, designed to delay commitment to precise actions as long as possible. It generates plans by searching a plan library of skeletal plans for one appropriate to the situation. These plans are expanded into a network of actions, which are represented along with their resource requirements and execution priority on an internal agenda mechanism, called the timeline. A scheduler selects actions from the timeline for execution and allocates resources (usually processing time) to them. When executed, these actions may initiate sensor or effector actions, perform problem solving actions, or search for subplans to accomplish the action's objective. Integrating responsive and impressionable adaptation with this style of planning should produce an agent able to act in a changing environment and able to exploit this experience by modifying its model of how to act.

Responsive Adaptation

Selecting the best action in a complex domain often requires up-to-date information. The two most common approaches to this requirement are situated action and error recovery. In the situated action approach, planning and acting are tightly coupled so that decisions about how to act are made at the time of action (Agre & Chapman 1986). Rapid response to changing situations is a natural result of this tight loop. However, this approach does not naturally support reasoning about resource allocation and coordination of disparate activities or agents (Georgeff & Lansky 1987). In error recovery, situations that indicate failure of plans in progress trigger mechanisms that change those plans. Unlike the situated action approach, in error recovery, adaptation is viewed as an adjunct to the standard planning process usually requiring additional mechanisms like monitoring and replanning (Wilkins 1987).

Many complex domains are characterized by scarce resources, both physical and temporal. Each approach, situated action and error recovery, possesses capabilities desirable for these domains. Situated action offers immediate response that is smoothly integrated with normal agent actions. Error recovery offers a plan structure that facilitates coordination and control in support of resource reasoning. Responsive adaptation should offer both capabilities.

Responsive adaptation should change the intended plan by the minimum required for the agent to continue acting successfully. Plans provide the structure for controlling action coordination, undesirable plan interactions and resource use. Responsive adaptation should preserve the expectations of these constraints while still addressing the changes in the environment. It should make the changes as quickly as possible because computation time is itself a resource and because the environment may have changed before the response can be determined. Finally, because responsive adaptation is activated when exceptional conditions occur, it should provide broad coverage of possible situations; it is, in effect, the action of last resort.

An agent detects failure conditions in several ways. Actions may be unable to execute to completion, thus signalling a failure. In systems with multiple control layers (such as Brooks' subsumption architecture (Brooks 1986)), settings or commands from different layers may conflict. Potential failures can be detected by monitoring the progress of actions and plans toward their objectives. Each of these occurrences provides a context for the error along with information about its nature and perhaps its cause.

Responsive adaptation in Phoenix proceeds by using the context of the failure to search a skeletal plan library of general recovery plans. These plans make mostly simple repairs to the structure of the evolving plan. These repairs are based on structural manipulations of the plan representation and can be used in different situations. This strategy for recovery plans is similar to that of SIPE (Wilkins 88).The recovery plans vary from simply re-scheduling the failed action to aborting the plan in progress and selecting another. Other plans re-instantiate plan variables or update information from the environment. These plans are represented in the same action description language as the domain specific plans and so are interpreted by the normal planning mechanisms. Furthermore, these plans serve a dual purpose: they are intended to salvage the failed plan and to guide improvements to the plan knowledge base.

Impressionable Adaptation

Failures occur when the environment changes unexpectedly and detrimentally. At times, this occurs because the environment is wildly unpredictable; most often, it is because the plan was not quite right. Relying on the contents of a static plan library biases the agent toward repeating the same plans regardless of whether these actions are the best for the situation.

Impressionable adaptation changes the agent's internal model of how to act in response to experience with failure. Plan failure provides an ideal opportunity for model refinement for two reasons: plan failures tell the planner where its knowledge is inadequate or brittle, and the agent has less to lose in trying something new. Several systems have exploited the opportunity of plan failure to refine the knowledge base. For example, Kristian Hammond's CHEF system modifies plans to anticipate and avoid failures by explaining the cause of the failure (Hammond 1986). SOAR learns new rules by chunking the results of searches triggered by impasses (Laird 1987). Each relies on a domain model. For CHEF, that model is a model of relationships between actions and effects; for SOAR, it is a model of a search space of operators and their applicabilities.

When the domain model is incomplete or approximate, impressionable adaptation must rely on other characteristics of the environment. In on-going environments in which the potential cost of acting is not catastrophic, the environment can be viewed as a laboratory for discovering how best to act. Changes to the plan library can be treated as hypotheses about how to act, which are confirmed or refuted by their later execution. Successful planning reinforces known behaviors; unsuccessful planning suggests alternative behaviors. Thus, inadequacies in the domain model can be counterbalanced by experience.

Because impressionable adaptation must share computational and physical resources with responsive adaptation and other activities in support of planning, it must opportunistically gather information or use what was gathered in the service of other activities. Adaptation mechanisms must operate from limited knowledge of the domain and make use of available temporal and other resources to minimize competition with other activities of the system.

Impressionable adaptation in Phoenix is initiated in response to information about the success of a plan, usually the lack of success. When initiated, the adaptation mechanisms search for a skeletal plan that gather evidence and modify the failed plan. As in responsive adaptation, these plans rely as much as possible on information already available. The modifications to suspect plans are mostly simple changes to monitor particular conditions or actions for potential failure, and avoid or recover more easily from failure. These plans include adding choice points to prevent premature commitment to a course of action, changing the conditions of plan application, restricting action selection at choice points, and adding monitoring. The modified plans are added to the plan library causing the original plan to be either replaced or restricted in its applicability. By augmenting the plan library, the agent has the option of alternative actions when later it encounters the situations that previously led to plan failure.

Integrating Adaptation into Phoenix

Like planning, adaptation is just one of the many problem solving activities that an agent performs. Consequently, it should be subject to the constraints of any problem solving activity. For systems that operate under time pressure, the primary constraint on problem solving is computation time. As described earlier, all problem solving activities in Phoenix are represented uniformly on the timeline. The cognitive scheduler accesses this structure in determining processing order and allocating processing time. Because responsive and impressionable adaptation are represented as problem solving actions, they can be included in existing resource reasoning and activity monitoring.

Adaptation actions are added to the timeline when failures occur. Failures are recognized through three mechanisms in Phoenix: execution errors, reflexes, and envelopes. Execution errors occur when an action cannot execute to completion because the state of the world was not right, required information was not yet available, or, for some problem solving actions, no solution exists. Reflexes are a low level control mechanism that compensate for time delays in planner response to keep the agent out of catastrophe. When they are triggered in response to dangerous environmental conditions, they program effectors to remove or at least reduce the danger. Envelopes predict impending failure (Hart, Cohen & Anderson 1990, Powell & Cohen 1990). They perform sophisticated monitoring of the plan's progress in the world, integrating the efforts of many agents, to determine whether the plan can complete within its environmental and resource limitations. If a plan will be unable to complete successfully under the present conditions, the performance envelope is violated and an impending failure is signalled. These mechanisms signal failures by adding adaptation actions to the timeline.

These actions include information readily available about the circumstances of the failure.

Adaptation actions on the timeline are treated as a type of planning action. They rely on the same planning method--lazy skeletal expansion--and when executed, search a taxonomy of skeletal adaptation plans in the plan library to find the appropriate response to the situation. As a type of planning action, adaptation may access and use the same methods as other planning methods and can be smoothly integrated into the planning process. As a timeline action, adaptation actions have access to the same memory structures and are subject to the same resource management techniques as are other timeline actions.

Improving Behavior through Adaptation

Adaptation compensates for inadequacies in planning. When the environment changes so that on-going plans will fail, responsive adaptation searches the plan library for an appropriate response and changes the on-going plan to allow the agent to continue from the failure. When the model of how to act in the environment fails, impressionable adaptation changes the plan library to avoid or anticipate the problem in the future. Thus, impressionable adaptation reflects the experience gained from responsive adaptation into the plan library to improve planning behavior. Figure 1 shows how the two types of adaptation influence the plan library and the timeline. Responsive adaptation modifies the timeline based on plans found in the plan library. Impressionable adaptation modifies the plan library based on information gleaned from the timeline.

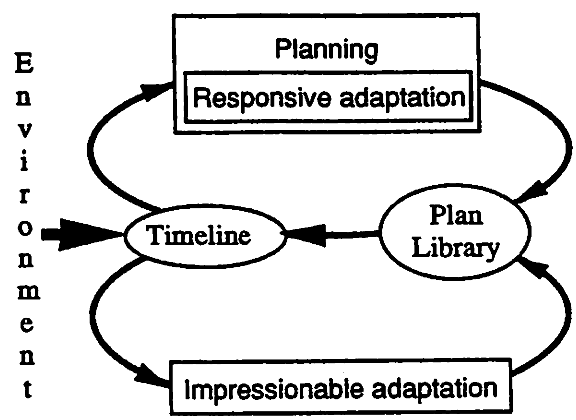
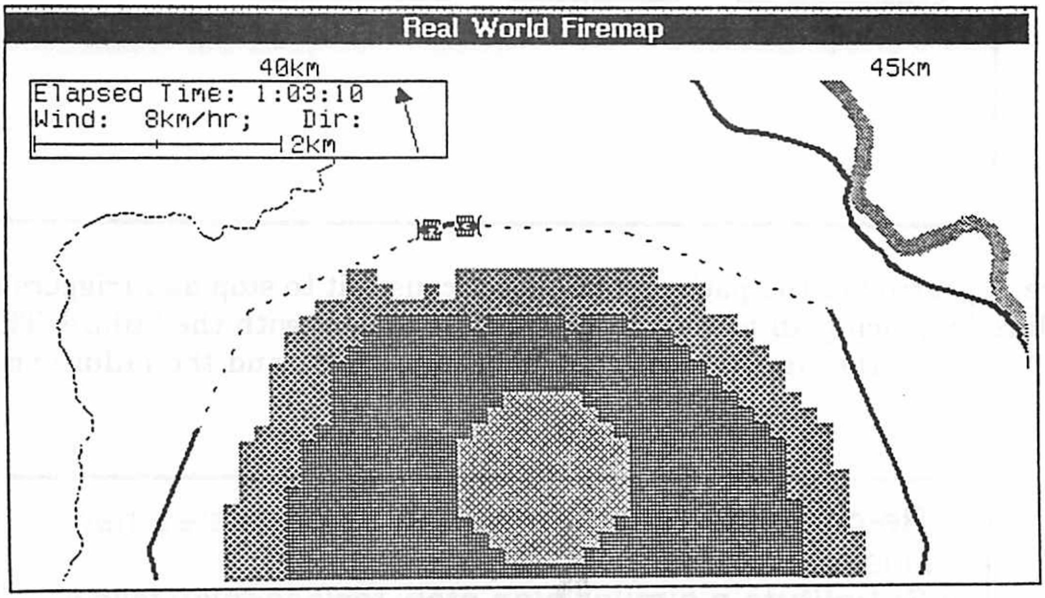


Figure 1: Responsive and impressionable adaptation interact through the timeline and the plan library.

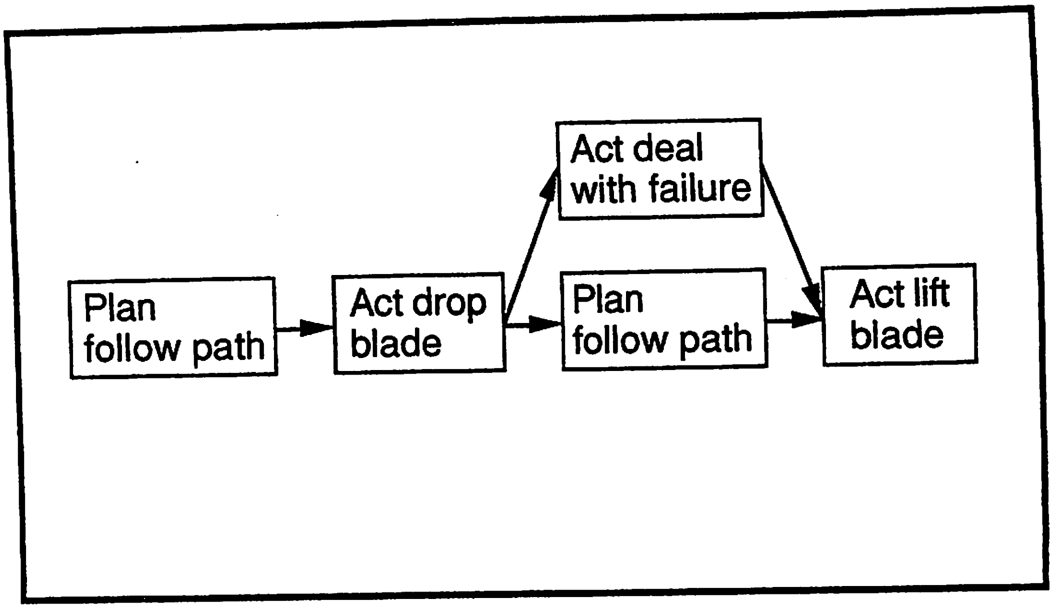
Integrating responsive and impressionable adaptation into the Phoenix system forms the core of my dissertation, as proposed in (Howe 1989). My hypothesis is that when the agent adapts its own behavior to its environment, its behavior will be better suited to that environment, and the agent's internal model will more accurately reflect the constraints of that environment. Consequently, the goal of the project is to understand the relationship between adaptation and the characteristics of the environment. The internal structures resulting from adaptation should evidence the impact of environment; for example, plans should reflect the variability and rate of change of the environment in the number and nature of the decision points in them. The benefits of additional plan knowledge should more than compensate for the computational overhead; in other words, adaptation must have demonstrable utility.

Appendix: Example of Adaptation in Phoenix

The indirect attack plan fights fires by predicting the likely spread of the fire and building fireline at those points so that field agents can surround the fire before it reaches the build points. In the wind has shifted or the fire has spread more rapidly than predicted the fire may cross fireline segments under construction.



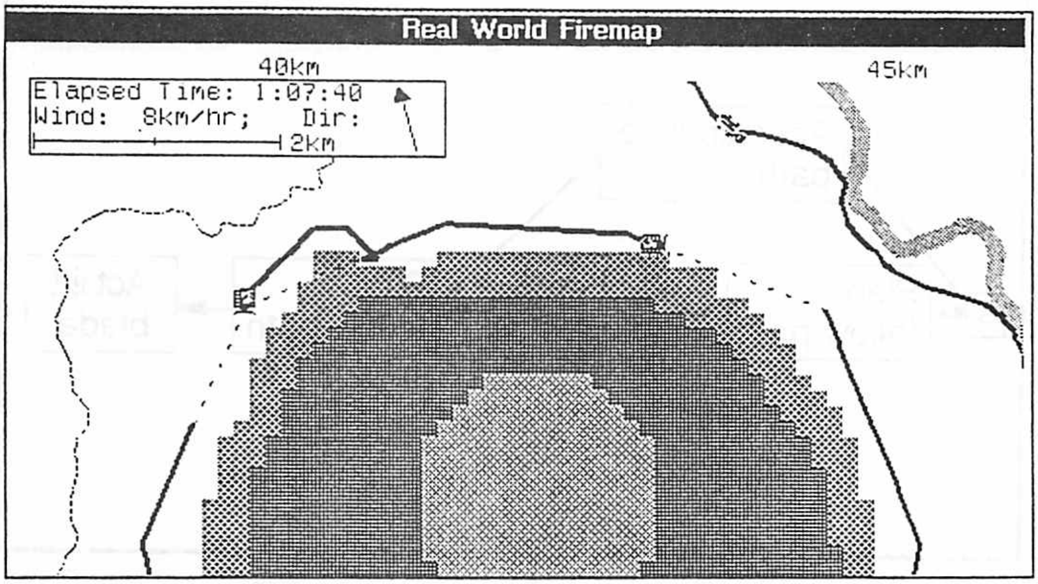
Two bulldozers are shown working to build fireline around the fire. Already constructed fireline is shown as a solid black line; fireline to be constructed is shown as a dotted line. The fire is the shaded grey area south of the bulldozers.



Fire appearing in the path of a bulldozer causes it to stop and triggers a failure by placing an action on the timeline to deal with the failure. This figure shows the timeline with the plan in progress and the failure action.

- **Re-calculate values for variables:** calculate a new value for variable "path" (fireline segment)
- **Substitute a similar plan step:** replace "plan follow path" with "plan follow parallel to fire" or "plan move on heading"
- **Abort the current plan and re-plan:** replace indirect attack plan with a direct attack plan

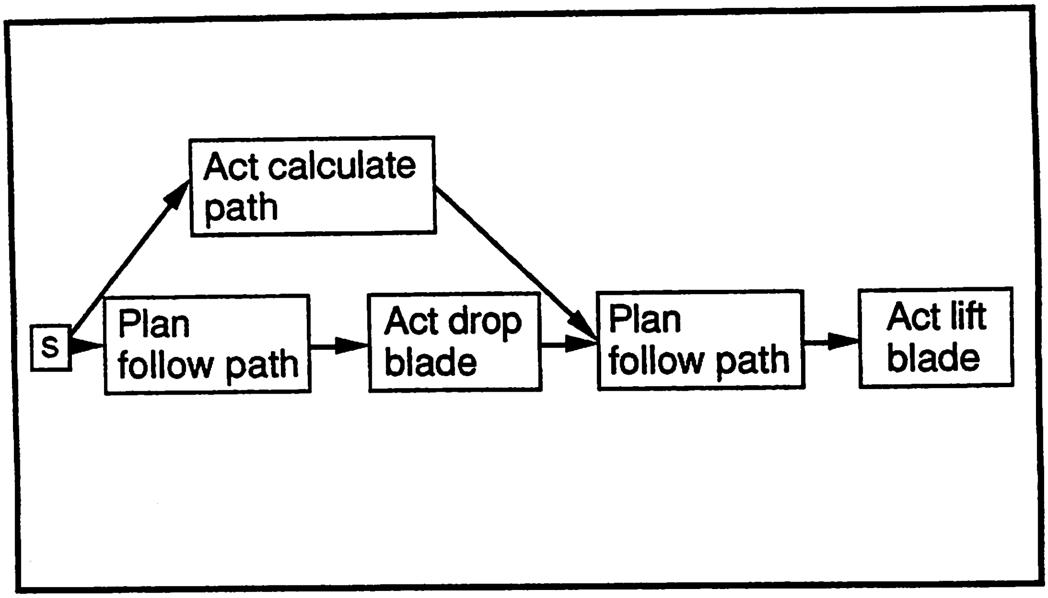
Several Responsive Adaptation methods are applicable in this situation.



In this case, the agent chooses to re-calculate the path variable because it is the cheapest applicable action. As shown, the bulldozer has executed this adaptation plan and completed building the fireline while still avoiding the fire.

- Add choice points to the plan to delay commitment.
- Add actions to the plan.
- Divide single plan into two smaller plans.
- Restrict the applicability of the plan.

Given the experience from responsive adaptation, impressionable adaptation can modify the original plan that failed. Several of the impressionable adaptation methods are applicable to this situation.



Based on the context, impressionable adaptation adds the new action, as in the failure recovery, to calculate the variable "path", creating a new plan that calculates a path based on the situation that exists just before miving. This new plan is added into the existing plan library.

Acknowledgments

This research was supported ONR University Research Initiative grant N00014-86-K-0764 and by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Office of Scientific Research under Contract No. F49620-89-C-00113. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

This research exploits the richness of the environment and the capabilities of the architecture in the Phoenix system. As such, I wish to thank members of the Phoenix group for many helpful suggestions on the research described here and the tremendous design and programming effort that resulted in the Phoenix system: Paul Cohen, Mike Greenberg, Dave Hart, Paul Silvey, Scott Anderson, and Dave Westbrook.

References

Philip E. Agre and David Chapman. 1986. Pengi: An Implementation of a Theory of Activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence*. Seattle, WA.

Rodney A. Brooks. 1986. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, RA-2(1).

Paul R. Cohen, Michael Greenberg, David M. Hart, and Adele E. Howe. 1989. Trial by Fire: Understanding the Design Requirements for Agents in Complex Environments. *AI Magazine*, 10(3).

Michael P. Georgeff and Amy L. Lansky. 1986. Reactive Reasoning and Planning. In *Proceedings of the Sixth National Conference on Artificial Intelligence*. Seattle, WA.

Kristian J. Hammond. 1986. Learning to Anticipate and Avoid Planning Problems through the Explanation of Failure. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA.

David M. Hart, Scott D. Anderson and Paul R. Cohen. 1990. Envelopes as a Vehicle for Improving the Efficiency of Plan Execution. submitted to the Eighth National Conference on Artificial Intelligence.

Adele E. Howe. 1989. Adapting Planning to Complex Environments. PhD Dissertation Proposal, Dept. of Computer and Information Science, University of Massachusetts.

John E. Laird, Allen Newell, and Paul S. Rosenbloom. 1987. SOAR: An Architecture for General Intelligence. *Artificial Intelligence Journal*, 33:1-64.

Gerald M. Powell and Paul R. Cohen.1990. Operational Planning and Monitoring with Envelopes, in *Proceedings of the IEEE Fifth AI Systems in Government Conference*, Wash., DC.

David E. Wilkins. 1988. *Practical Planning: Extending the Classical AI Planning Paradigm.* Morgan-Kaufmann Publishers, Palo Alto, CA.