Responding to Environmental Change

Adele E. Howe, Paul R. Cohen

COINS Technical Report 90-23

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

Abstract

Responding to environmental change is perhaps the most difficult aspect of designing agents to plan and act in complex dynamic environments. In this paper, we analyze the environmental response requirements of such an environment, provided by the Phoenix forest fire fighting simulator, and describe three mechanisms that together address the demands of that environment. The limitations on response imposed by the environment are described in terms of "windows of response opportunity". This framework matches the demands of the different types of environmental change occuring in the environment to the abilities and limitations of the mechanisms intended to address change. As applied in the Phoenix environment, it motivates the design of three different mechanisms that address three different types of change. The three mechanisms, reflexes, lazy skeletal expansion, and responsive adaptation, are described in detail, and their interaction with the environment is illustrated in an example.

This research was supported by DARPA-AFOSR contract F49620-89-C-00113 and the Office of Naval Research, under a University Research Initiative grant, ONR N00014-86-K-0764. We wish to thank members of the Phoenix group, especially Michael Greenberg, David Hart, David Westbrook, Paul Silvey, Scott Anderson, and Evan Smith, for their contributions to the design and implementation of the Phoenix system. We also wish to thank Carol Broverman and Ross Beveridge for their comments on several drafts of the paper.

1 Introduction

Environmental conditions change. In the most advantageous cases, the change is caused by the efforts of agents working to control aspects of the environment. In the least, the changes are caused by environmental forces, which may or may not be predictable. Whether the environmental change is detrimental to an agent depends upon whether the agent's actions, both thinking and effecting, are *responsive* to those changes.

Responsiveness requires timely appropriate action in response to environmental change. When a truck is driving toward you in your lane, *timely* is immediate and *appropriate* is a simple evasive action. When a hurricane is due to arrive tomorrow, *timely* is within the next day and *appropriate* is a complex combination of actions intended to protect home and family. Different types of environmental changes require different types of responses. An agent must take actions appropriate to and within the time frame of the types of events in its environment. In this paper, we describe how agents in the Phoenix system are responsive to changes in their environment.

2 Responding to Change in Phoenix

Agents in Phoenix work to contain simulated forest fires in Yellowstone National Park[4]. Fires are contained by removing fuel from their paths, causing them to burn out. This process, called building fireline, requires the coordination of several field agents to surround the fire with fireline. One agent, the fireboss, coordinates the activities of semi-autonomous field agents, bulldozers, to build fireline at many points around the fire. Fire spread is influenced by many environmental factors: wind speed, wind direction, terrain cover, elevation gradient, and moisture content. The fire's overall shape and spread is determined by these factors, but so many minute factors are involved in the precise spread of the fire that it is not possible to predict the exact time at which a particular point on the map will catch fire.

2.1 Response requirements for the Phoenix environment

The primary constraint on responsiveness is time. For any environmental change, there is a "window of response opportunity", the time during which the agent can respond. In the example of the truck in the wrong lane, the window is very narrow and once the window has passed, it is simply too late to act.

The window starts when an environmental change is perceived (T_{wb}). It ends when the effects of that change occur (T_{we}). These two points define the window, the time delay between perception of a change, real or impending, and its effect (shown in Figure 1). For certain classes of environmental change, the delay will be short (e.g., the truck); for some, it will be longer (e.g., the hurricane). However, environmental forces are not the only influence on window size. Actions often require some start up time or overhead between their initiation and their effects; for example, the time delay between deciding to put on the car brakes and the car's stop is significant enough to require a safe following distance.

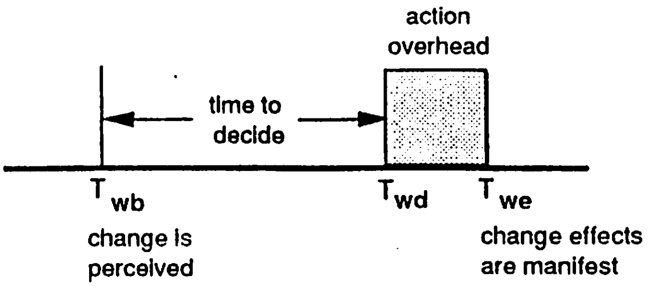


Figure 1: Environmental Change Window of Response

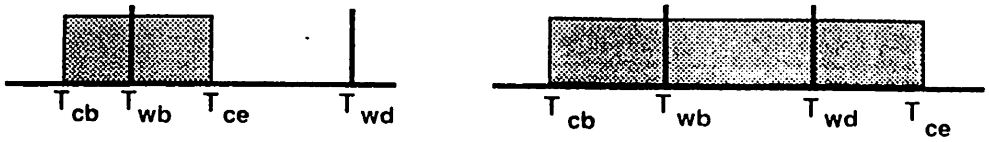
So, response overhead (T_{wd}) can narrow the window. The window that remains is the amount of time left to decide how to act.

The Phoenix environment includes two qualitatively different kinds of environmental change distinguished by their response windows. Narrow response windows are common when environmental change is unpredictable. For example, when agents work close to the fire, rapid response is necessary if the fire suddenly threatens to engulf them. Wider response windows are common when environmental change is gradual. For example, it takes simulated hours or days to surround even small fires which allows at least hours to make strategic decisions about containment.

Because agents must respond to both types of change, they need mechanisms to address both. A narrow window suggests rapid decision making (to minimize the decision time) and simple action (to minimize the response overhead). If the window is too narrow, it precludes deliberation and complex responses, but encourages reactivity. A wide window affords time to deliberate over the best response. Reactive and deliberative approaches have different fundamental limitations: reactive approaches guarantee response within fixed time bounds and so cannot use additional time even if it is available; deliberative approaches cannot guarantee response within short fixed time, but can exploit additional decision time. The fundamental differences in time usage for the two approaches conflict, which precludes incorporating them both in the same mechanism. Thus, Phoenix agents need two separate mechanisms: a reactive mechanism, which we call ***reflexes*** and a deliberative mechanism, which we call ***lazy skeletal expansion.***

Together, reflexes and lazy skeletal expansion form a two-layer response system. As in Brooks' subsumption architecture [3], each layer provides a particular level of competence. Reflexes address change that occurs faster than lazy skeletal expansion can respond to it; lazy skeletal expansion coordinates actions and avoids detrimental plan interactions.

As discussed, the response window is defined by the time delay between when a change is perceived and when its effects are felt. The response window assumes that the agent ***notices*** the change at the moment that the window opens and immediately starts formulating a response. However, because agents may have already committed to other actions (e.g., attending to different fires), additional time may pass between when the environmental response window opens and the agent notices it. Lazy skeletal expansion attempts to minimize the additional time by deferring commitment, as much as possible, to a precise course of action. When necessary commitments have a duration less than the response



a: Commitment is flexible within demands of response window

b: No response flexibility due to commitment

Figure 2: Environmental Change Window Compared to Response Flexibility

window, environmental change can be addressed by lazy skeletal expansion. Figure 2, part a, displays this relationship, with T_{cb} indicating the commitment beginning and T_{ce} indicating commitment end; note that the commitment ends before the response window does. When the window has shorter duration than the commitment, then the planner may commit to a course of action that may be rendered impossible by the environment (as shown in Figure 2, part b), thus, resulting in plan failure. Because it isn't always possible to defer commitment, another mechanism is required to adapt the commitment structure (i.e., plans) in response to the detrimental environmental change. We call this mechanism ***responsive adaptation*** because it adapts plans in progress in response to detrimental environmental change.

All agents, fireboss and field agents, share a common agent architecture that includes these response mechanisms. That architecture consists of four basic components: sensors, effectors, reflexes and a cognitive component. Sensors perceive the state of the environment. Effectors take physical action for the agent in the environment. Reflexes change the settings of effectors to respond within a narrow window. The cognitive component is responsible for tasks related to deliberative response, action coordination and resource management. The cognitive component includes both lazy skeletal expansion and responsive adaptation. In response to environmental conditions, it selects plans from the plan library and adds them to a partially ordered agenda structure, called the *timeline*, for later execution (this process is part of lazy skeletal expansion and will be explained in more detail in Section 2.3). The information and control relationships among these components are displayed in Figure 3.

The three mechanisms: reflexes, lazy skeletal expansion and responsive adaptation, account for the range of response that are demanded by the Phoenix environment. Reflexes change what the agent is currently doing. Lazy skeletal expansion changes what the agent is planning to do. Responsive adaptation changes how the agent was planning to do something. The remainder of this section will describe, in detail, the three mechanisms that together ensure responsiveness in Phoenix.

2.2 Reflexes

Reflexes provide time-bounded responses to critical situations. They compensate for the time delay between when a situation occurs and when the cognitive component can notice and deal with it. As such, they constitute an architectural component specialized

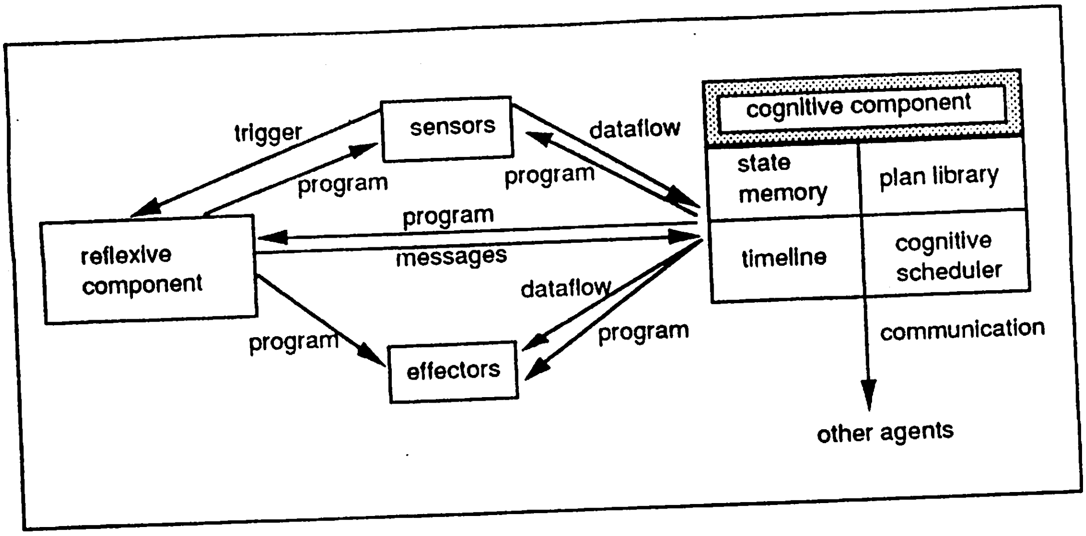


Figure 3: Phoenix Agent Architecture

for rapid, simple response.

Reflexes are associations between sensors and effectors mediated by two simple functions, one for triggering reflex execution based on sensor readings (the trigger function) and one for changing effector settings based on those sensor readings (the response function). Trigger functions are simple functions of the sensor readings, such as whether the value exceeds a threshold or equals some value. Response functions make simple changes to effectors, turning them off or on or making minor parameter adjustments. These functions rely only on currently available sensor readings; reflexes retain no persistent state information other than the values of the operating parameters. These parameters define reflex sensitivity and are set by the cognitive component.

Reflexes are activated in tandem with the sensors and effectors. After a sensor executes, trigger functions, which link the sensor to particular reflexes, are executed to determine whether the associated reflexes should be activated. The trigger function may rely on values from more than one sensor; in which case, it simply checks the most recent readings for the critical sensors. When activated, the reflex executes the response function to change effector settings. For example, when the *sense-road-heading* sensor has a value different than the *sense-agent-heading* sensor, the *follow-road* reflex changes the *heading* parameter of the movement effector to maintain the road heading.

The time required for response by the reflexes is bounded by the activation rate of the sensors. Reflexes are executed immediately after the sensors and require little time to execute. Thus, the agent can respond as quickly as its sensors can notice the environmental change, which reduces the response time from the cycle time of five simulation minutes required by the cognitive component, to seconds of simulation time. This approach can be contrasted with that adopted in PRS[7]. PRS includes all responses (called KAs) in the same framework and relies on the most crucial of them being quickly selected and executed. That is, it relies on fast KAs for reacting to crucial situations. This produces a guaranteed "reactivity delay" of $s+t$ where s is the maximum time to determine the KAs applicability and t is the cycle time (as dictated by the maximum time required to execute a primitive action).

Reflexes respond to environmental changes that have narrow response windows. Consequently, imminent disasters are prevented, but sometimes, at a cost of temporarily stopping progress in a plan. As a result, when reflexes change the settings of an effector, superseding those of the cognitive component, the reflexive component sends a message to the cognitive component warning it of the change. This interaction is described in further detail in Section 2.4.

2.3 Lazy Skeletal Expansion

Lazy skeletal expansion responds to environmental change characterized by wide response windows. Deferring commitment to a precise course of action maximizes the opportunity for change to influence decisions. Skeletal plans provide a structure in which to base action decisions, which expedites action coordination and minimizes interactions.

Deferred commitment is accomplished by interleaving three basic activities: find, expand, and execute. *Find plan* actions are placed on the timeline as part of plans (to defer commitment to a particular plan or action) or in response to exceptional conditions (as noted by messages from the reflexive component or from other agents). These actions use their context within the timeline to search the plan library for skeletal plans appropriate for the context and the current state of the world. For example, if the find plan action is to get an agent to the fire, the context includes information about the type of fire fighting plan it is part of, the techniques being used to fight the fire, and the coordination needed with other agents, in addition to the location of the fire. *Expand plan* actions instantiate the plan's network of actions on the timeline. *Execute* actions calculate variable values. manage resources, and control the agent's interactions with the world. As the timeline actions are executed, plans and actions become incrementally added, sensors and effectors are activated, and the agent pursues the plan. Actions become eligible for execution only when the siblings that precede them have already executed; even then, execution may be deferred until information about the state of the world is available. Because plans are combinations of primitive actions and plan expansion actions, this leads to interleaving of action and planning.

Skeletal plans have four parts: applicability conditions, resource requirements, execution methods, and an action network. Applicability conditions describe the conditions under which the action is appropriate. Resource requirements describe the expected time, information (as represented by variables), and physical resource needs of the action. Execution methods are the procedures for executing the action, i.e., what gets called when the action is chosen for execution. An action network is a network of problem solving actions associated by temporal and data dependencies. Figure 4 shows a "generic" Phoenix plan as it might become expanded on the timeline. Vertical lines indicate that the higher action placed the lower action on the timeline. Horizontal lines are temporal relations between actions. The boxes under the action further describe some of its characteristics.

Skeletal planning has previously been applied in domains in which planning and acting are completely separate, cancer therapy advice [13] and experiment design [6]. While the

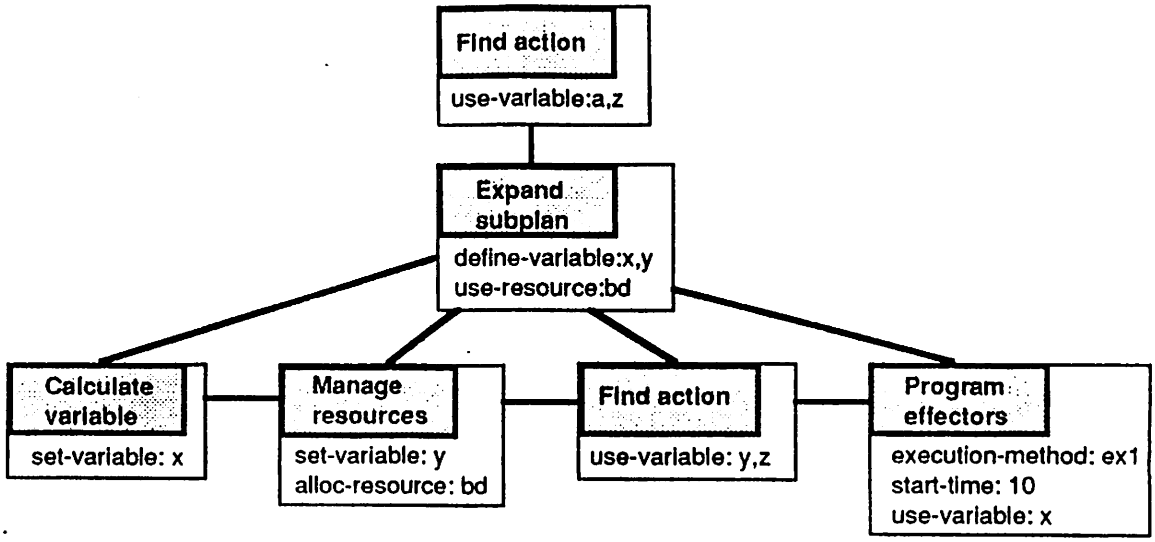


Figure 4: "Generic" Phoenix plan as represented on the timeline with information about inter-dependencies

goals and implementation of these other skeletal planning projects are rather different, the structure of the planning is much the same. Lazy skeletal expansion is similar to the planning method employed in PRS[7]. KAs, the representation for procedural knowledge, include an invocation condition, which specifies when the KA is useful, and a body, which describes the sequence of subgoals which constitute the procedure. Thus, the structure of PRS is similar, but the control is distinct. At each execution cycle, PRS checks all KAs for applicability, selecting one for execution. In Phoenix, only actions on the timeline are considered for execution. Thus, PRS provides for more reactive planning, but at the expense of being unable to allocate time and schedule actions beyond an execution cycle. The timeline structure in Phoenix was designed to support real-time scheduling of actions[11], but does complicate plan modification.

2.4 Responsive Adaptation

When lazy skeletal expansion overcommits to a course of action and crosses over an environmental response window, plan failures can occur. Failures occur when the plan either cannot continue or cannot succeed if it does continue. In effect, the environmental change response window conflicts with plan commitment. In Phoenix, this mismatch is a result of: non-local interactions, uncertain or obsolete information, unpredicted changes, and novel situations.

Non-local interactions occur when the agent attempts to respond, in parallel, to different environmental changes and so overcommits its resources. For example, the fireboss treats each fire in the environment as a separate situation, making decisions about containment largely independently; yet, the resources for controlling the fires are fixed and must be shared between the situations. Consequently, decisions about changing resource allocation to a particular fire impact and may thwart resource expectations for plans in progress on a different fire.

Uncertain information causes failure when the agent is forced to commit resources without being certain of the magnitude of the need. For example, fires that appear at the periphery of view may look small to a watchtower, but may actually be conflagrations. Without better information, the fireboss must take an "educated guess" at the real situation and commit field agents to contain it, accepting the possibility that more or fewer agents may actually be needed.

Unpredicted changes naturally produce failure. Fire fighting involves working in constrained situations that are vulnerable to unexpected changes. If the wind changes unexpectedly, a previously safe area in which to build fireline may become dangerous.

Finally, novel situations require commitment to plans that may not actually be best for the situation. When environmental change results in a novel situation, the agent may not know to respond.

The responsive adaptation mechanism responds to environmental changes by adapting plans in progress. Plans provide the structure for controlling action coordination, undesirable plan interactions and resource use. Thus, the expectations included in the plan structure should be preserved, while still addressing the changes in the environment; so, responsive adaptation should change the intended plan by the minimum required for the agent to resume acting. It should make the changes as quickly as possible because computation time is itself a resource and because the response window may have been closed before the response is computed.

Because the process of adapting plans is an action within the context of the plan, it should be accessible to other problem solving mechanisms that direct and constrain the agent's actions, such as resource allocation. Because responsive adaptation is activated when exceptional conditions occur, it should provide broad coverage of possible situations; it is the mechanism of last resort. Georgeff etal in [7] describe examples of bizarre behavior in PRS that results from "mis-applying" actions to novel situations. Without a general replanning capability, an agent repeatedly performs the same inappropriate behavior.

2.4.1 Detecting Failures

The agent detects failures when it cannot successfully conclude an action or plan. Three mechanisms signal failures: execution errors, reflexes, and envelopes. Execution errors occur when an action cannot execute to completion because the state of the world does not match the assumptions, information is not yet available, or, for some problem solving actions, no solution exists. When the agent encounters a dangerous environmental condition, reflexes change effector programming to remove or at least reduce the danger. If this change of programming conflicts with the programming previously set as part of a plan action, then the reflex signals a failure.

Both execution errors and reflexes signal obvious failure in that the plan is actually prevented from executing. Determining whether an on-going action can *ever* succeed requires active monitoring of the progress of the action. *Envelopes* detect impending failures [8,12]. 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 signaled.

2.4.2 Responding to failure

The detection mechanisms signal failures by adding actions that find recovery plans to the timeline, placed in parallel with the action that initiated them. These actions include readily available information about the failure trigger, the agents involved, and the error type. Recovery actions are structured like other planning actions in that they have a context within other plans; they reference variables and information available in that context; they are scheduled like other actions; and they employ the same planning method–lazy skeletal expansion–for deciding on response. As a type of planning action, adaptation can be smoothly integrated into the planning process. As a timeline action, adaptation has access to the same memory structures and is subject to the same resource management techniques as are other timeline actions.

Responsive adaptation in Phoenix searches the plan library of general recovery plans for one appropriate to the failure. These plans are represented in the same action description language as the domain plans and so are interpreted by the standard execution methods. The plan structure as it is represented on the timeline (and displayed in Figure 4) defines a context or locality for action, indicates dependencies between actions and distinguishes decision points. These structural characteristics provide backtracking points for adapting the plan. Decision points are actions in the plan structure that rely on environmental context to direct their execution. Any action that binds variables or calculates variable values based on context is a decision point. Actions that select other plans for execution use environmental context to determine applicability, and so are certainly decision points. The following recovery plans use the plan structure to identify decision points and dependencies between the decision points and the failure point to support recovery:

- Wait and then re-execute the failed action.
- Reinstantiate the action, updating values for its variables.
- Select an alternative execution method for the failed action. Some execution methods sacrifice accuracy for computation speed; sometimes, the accuracy is necessary.
- Re-calculate values for variables used in the action. Some variables are calculated by actions that were executed earlier in the plan; this action involves re-executing a previous action that set a variable used in the failed action.
- Substitute other variables of the same type as those used in the action.
- Substitute a similar plan step for the failed action that produces approximately the same effect in the environment.
- Allocate additional resources to the plan.
- Re-execute the parent plan selection action, i.e., re-plan.

The recovery plans make mostly simple repairs to the structure of the evolving plan. As such, they can be used in different situations, do not require expensive explanation, and ensure full coverage of all possible failures–the agent must be able to do something. SIPE

[14] and IPEM [2] rely on a similar strategy of replanning by general plan repair methods. This strategy sacrifices efficiency for generality and so results in a planner capable of responding to any failure, but perhaps in a less than optimal manner.

Like any plan in Phoenix, recovery plans have applicability conditions to guide their selection. When a failure is recognized, the find-plan method searches the recovery plans for the most appropriate and adds it to the timeline at the current location for execution. Because these responses are general, they may not always work. Failure of a recovery plan works much the same as any failure, except that rather than repair the recovery plan, the system will select another method to repair the original failure. Thus, responsive adaptation should always have some response to environmental change that has eluded the other mechanisms.

3 An Example from Phoenix

Fires are fought by building barriers to prevent further spread. Building fireline close to the edge of the fire minimizes the loss of forest, but maximizes the agent's vulnerability to the fire. A more conservative strategy is to fight fires with indirect attack, which involves predicting the likely spread of the fire and building line around it so that the field agents can completely surround the fire before it reaches the fireline.

The indirect attack plan determines where to build fireline to contain the fire at some designated point in the future, allocates the field agents needed to do so, and sends the field agents out to build their first segments of fireline. Relying then on lazy expansion, the plan is expanded further in response to messages from the field agents on their status, e.g., if they've encountered problems, need fuel, or have finished their last assignments. Only when information about field agents' status is known does the planner commit the agents to a course of action and so further expand the plan. In this way, the planner can be responsive to environmental changes that impact what the agents should be doing and exactly how they should be doing it.

Lazy skeletal expansion may keep agents from being assigned to work in a currently dangerous area, but it will not necessarily keep them from danger if the situation changes. If the wind has shifted or the fire spread more rapidly than predicted, the fire may cross the assigned fireline segment. Figure 5 shows a bulldozer (on the left) building a fireline segment (shown as a dotted line) which has been crossed by fire (shaded grey area south of the bulldozers). When sensors detect fire in the bulldozer's path, they trigger an *emergency-stop* reflex. This reflex programs the *movement* effector to stop, thus re-setting the programming installed as part of the plan. The reflex sends a message to the cognitive component which causes a failure signaling action to be added to the timeline, registering the emergency stop.

When the failure action is noted and executed, the planner searches for an appropriate response. Several recovery responses are possible: the plan variable for the fireline segment could be re-calculated, the build line action could be replaced by another type of build line action, or the parent plan could be replaced by another. In this case, the agent chooses to re-calculate the fireline segment variable because it is the cheapest action that

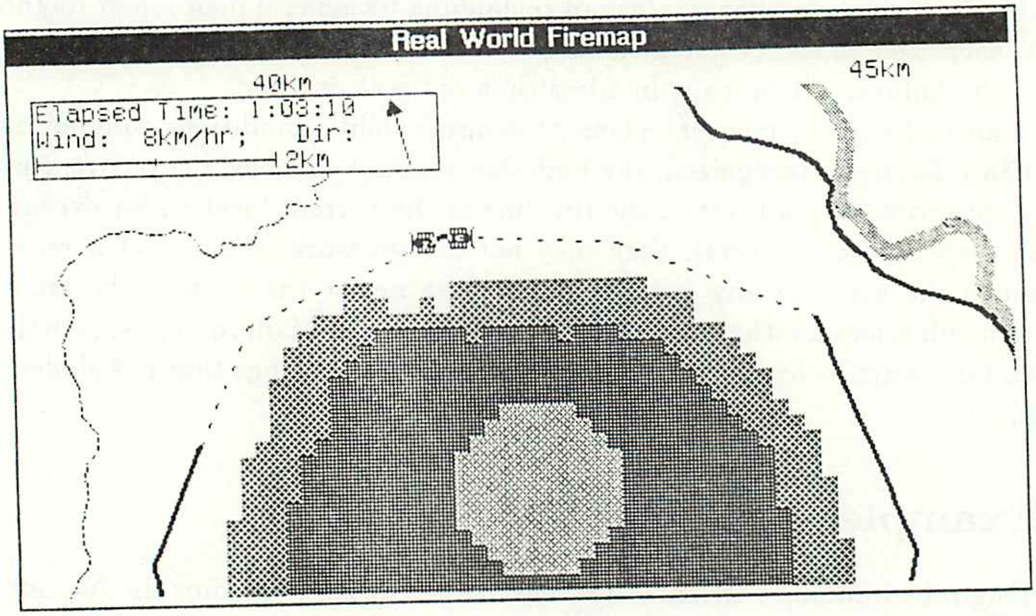


Figure 5: Bulldozer building fireline in indirect attack plan

is applicable. In Figure 6, the bulldozer has executed this recovery plan and completed *building* the fireline while still avoiding the fire. If this action had not worked because the fire had engulfed the endpoint of the fireline segment, the bulldozer could replace the original *build line on path* action with an alternative action, *build line parallel to fire* action that would change its behavior to direct attack, or it could have selected a new plan entirely.

4 Understanding Responsiveness in Phoenix

The three responsive mechanisms in Phoenix cover the range of environmental change that an agent will encounter in this environment. These mechanisms work because they are designed to address the demands and exploit the facilitating characteristics of the environment. Reflexes can rely on simple triggers and make simple responses because rapid catastrophic change is easily recognized and can be averted by simple evasive response. The reactive strategy supported by reflexes has been successful in other systems operating in domains with these demands [1,5]. Moreover, catastrophic change is relatively rare in the Phoenix environment. Generally, Phoenix provides a forgiving environment. Agents may pursue plans that are not the best and still manage to contain the fire, but at a higher cost. This forgiving nature allows the agent to use skeletal plans in different situations and to rely on general methods for responsive adaptation without risking disaster.

Action in the environment is characterized by stereotypical plans. Most of the basic strategies for fighting forest fires in Phoenix can be represented by a relatively small number of skeletal plans. Additionally, because most actions take place in different geographic locations, interactions are essentially limited to resource contention, which can be

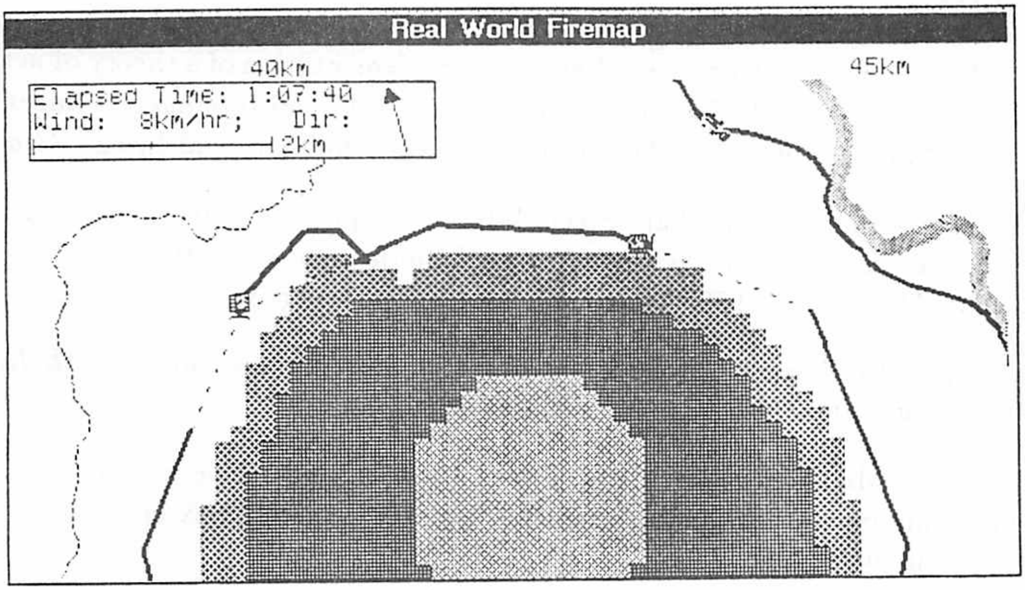


Figure 6: Bulldozer completes building line, albeit not in the originally planned location

accommodated in the plan structure. Thus, we reduced the need for reasoning about plan interaction by using skeletal structures that have "compiled out" that reasoning.

With their responsive mechanisms, agents can respond to any environmental change. Unfortunately, in the current state of development, they cannot always recognize change, and their responses are not always successful. Change is difficult to recognize when its detection depends on something *not* happening or depends on interactions of action and environmental forces. For example, the fireboss must conjecture that a bulldozer agent has become incapacitated when it fails to make contact. The fireboss must predict that a plan will not succeed when the fire starts to expand more quickly than it can be surrounded. Fortunately, research on envelopes [8,12] will address recognizing detrimental environmental change, as early as possible, in these difficult to detect situations.

Responses are not always successful because the plan library is incomplete. The scope of environmental factors that define situations and the difficulty of anticipating precisely the width of response windows for change in different situations makes it difficult to build a plan library that offers "best" plans for any situations. The agent should learn appropriate plans for itself by experimenting in its environment. A project currently in progress seeks to have the agents do just that [9,10].

Phoenix provides a rich environment in which to explore notions of responsiveness. It forces agents to confront qualitatively different types of environmental change and respond to them. Understanding the nature of the response demands of the environment is crucial to the design of mechanisms for addressing those demands. The success or failure of agents depends upon the ability of those mechanisms to respond appropriately to their environment.

References

[1] Philip E. Agre and David Chapman. Pengi: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 268–272, Seattle, Washington, 1987. American Association for Artificial Intelligence.

[2] Jose A. Ambros-Ingerson and Sam Steel. Integrating planning, execution and monitoring. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, Minneapolis, Minnesota, 1988. American Association for Artificial Intelligence.

[3] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1), March 1986.

[4] Paul R. Cohen, Michael L. Greenberg, David M. Hart, and Adele E. Howe. Trial by fire: Understanding the design requirements for agents in complex envronments. *AI Magazine*, Fall 1989.

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

[6] Peter E. Friedland and Yumi Iwasaki. The concept and implementation of skeletal plans. *Journal of Automated Reasoning*, 1:161–208, 1985.

[7] Michael P. Georgeff, Amy Lansky, and Marcel J. Schoppers. Reasoning and planning in dynamic domains: An experiment with a mobile robot. Technical Report Technical Note 380, SRI International, Menlo Park, CA, April 1987.

[8] David M. Hart, Scott D. Anderson and Paul R. Cohen. Envelopes as a vehicle for improving the efficiency of plan execution. Submitted to *the Eighth National Conference on Artificial Intelligence*, 1990.

[9] Adele E. Howe. Adapting planning to complex environments. unpublished PhD Dissertation Proposal, COINS Dept., University of Massachusetts, September 1989.

[10] Adele E. Howe. Integrating adaptation with planning to improve behavior in unpredictable environments. In *Proceedings of AAAI Spring Symposium on Planning in Uncertain, Unpredictable or Changing Environments*, Palo Alto, CA, March 1990.

[11] Adele E. Howe, David M. Hart, and Paul R. Cohen. Addressing real-time constraints in the design of autonomous agents. To appear in *Real-Time Systems*, 1990.

[12] Gerald M. Powell and Paul R. Cohen. Operational planning and monitoring with envelopes. In *Proceedings of the IEEE Fifth AI Systems in Government Conference*, Washington, DC, 1990.

[13] Samson W. Tu, Michael G. Kahn, Mark A. Musen, Jay C. Ferguson, Edward H. Shortliffe, and Lawrence W. Fagan. Episodic skeletal-plan refinement based on temporal data. *Communications of the ACM*, 32(12):1439–1455, December 1989.

[14] David E. Wilkins. *Practical Planning: Extending the Classical AI Planning Paradigm.* Morgan Kaufmann Publishers, Inc., Palo Alto, Ca., 1988.