# Reference features as guides to reasoning about opportunities

**Louise Pryor and Gregg Collins***
Northwestern University
The Institute for the Learning Sciences
1890 Maple Avenue, Evanston IL 60201
email: pryor@ils.nwu.edu, collins@ils.nwu.edu

## Abstract

An intelligent agent acting in a complex and unpredictable world must be able to both plan ahead and react quickly to changes in its surroundings. In particular, such an agent must be able to react quickly when faced with unexpected opportunities to fulfill its goals. We consider the issue of how an agent should respond to perceived opportunities, and we describe a method for determining quickly whether it is rational to seize an opportunity or whether a more detailed analysis is required. Our system uses a set of heuristics based on *reference features* to identify situations and objects that characteristically involve problematic patterns of interaction. We discuss the recognition of reference features, and their use in focusing the system's reasoning onto potentially adverse interactions between its ongoing plans and the current opportunity.

## 1. Introduction

An intelligent agent acting in a complex and unpredictable world must be able to both plan ahead and react quickly to changes in its surroundings. In AI, agent models have generally exhibited one or the other, but not both, of these capabilities. In particular, two opposing schools of thought have arisen: classical planning, in which a sequence of actions that the agent intends to execute is produced ahead of time (*e.g.* Newell & Simon 1963, Fikes & Nilsson 1971, Sacerdoti 1977, Tate 1977, Wilkins 1988), and reactive planning, in which the agent simply responds to its surroundings at any given moment, instead of following an explicit plan[1] (*e.g.* Brooks 1986, Agre & Chapman 1987, Beer *et al* 1990, Kaelbling & Rosenschein 1990). It seems clear that a competent agent model must combine elements of both approaches (*c.f.* Georgeff & Lansky 1987, Firby 1989, Hammond *et al* 1990, Simmons 1990, McDermott 1991).

Classical planning has principally concerned itself with the construction of models that are complete and sound (*e.g.* Chapman 1987, McAllester & Rosenblitt 1991), but proofs of these formal properties depend on unrealistic assumptions. For example, it must be assumed that the agent has full knowledge of the conditions in which the plan will be executed, that all actions have perfectly predictable results, and that no unpredictable changes will occur through causes other than the agent's actions. Such an approach leads naturally to models in which issues of plan execution are ignored, since without unpredictability nothing can happen that has not been foreseen, and plan execution will simply consist of performing the preordained steps. The assumption of perfect foresight severely reduces the practicality of classical planners.

Reactive systems tend to the opposite extreme, performing no lookahead, and concomitantly constructing no plans. A reactive system is instead directed by a set of rules that specify how to react in any given situation, and its competence thus depends entirely upon the extent to which its rules are able to specify the precise action to take in the particular situation in which it finds itself. This approach leads to models in which projection is ignored, as a reactive system is incapable of making use of a predictive model of its world; it does not use projection to determine whether a contemplated action is in fact a good one to take.

A competent agent should fall somewhere between the extremes of classical and reactive planning, making use of projection where possible, yet being able to react with minimal forethought when necessary. In particular, such an agent must be able to react quickly in the face of unexpected opportunities to fulfill its goals[2], even in situations in which it lacks the time or the information necessary to construct a detailed plan before proceeding (*e.g.* Birnbaum 1986, Hammond *et al* 1988, Brand & Birnbaum 1990). The issue of engineering a compromise between classical and reactive planning thus comes down to the problem of responding to opportunities: when an opportunity arises, the

*This work was supported in part by the AFOSR under grant number AFOSR-91-0341-DEF, and by DARPA, monitored by the ONR under contract N00014-91-J-4092. The Institute for the Learning Sciences was established in 1989 with the support of Andersen Consulting, part of The Arthur Andersen Worldwide Organization. The Institute receives additional support from Ameritech, an Institute Partner, and from IBM.

[1] There are obvious similarities to behaviorism (Skinner 1974).

[2] Or to unexpected threats against its goals, but for our purposes these can be regarded as being the same thing.

agent should put only as much effort as is rationally justified into projecting the consequences of pursuing that opportunity. In this paper, we shall concentrate on the issue of how an agent should respond to perceived opportunities, and we shall introduce a method for determining quickly whether it is rational to seize an opportunity without first acquiring more information.

## 2. Effective independence

We are building a system, PARETO[3], that operates a simulated robot delivery truck in the TRUCKWORLD domain[4] of Firby and Hanks (1987). PARETO can, for example, recognize that a sack of cement mix sitting by the side of the road presents an opportunity to achieve the goal of satisfying a customer who has asked for cement, and can reason about whether this particular opportunity should be pursued. PARETO might choose not to take advantage of an opportunity if doing so would be detrimental to other goals it is currently pursuing. For example, it might not want to pick up a sack of cement that it has come across if it is low on fuel, or if it is late in making another delivery.

The decision to pursue an opportunity depends on an analysis of the costs and benefits of doing so. PARETO must therefore have a way of determining what the costs and benefits are, and how they compare. For opportunities, the benefits can be measured in terms of goal achievement and beneficial side effects on other goals, and costs in terms of forgoing other opportunities and harmful side effects on other goals. A theory of expected utility can be used to compare the results of taking the different courses of action that are available to the agent (Von Neumann & Morgenstern 1944, Feldman & Sproull 1977). There is a well-defined theory of how to arrive at the expected utilities, given certain information; in particular, the agent requires the prior and conditional probabilities from which it can calculate the probabilities of the various outcomes, and the utility values of the outcomes. Unfortunately, precise values are often not available, and indeed in most real-world situations the planner only has access to at most crude approximations of the necessary probabilities and utilities (Haddawy & Hanks 1990).

Even if the necessary information were available, the calculations of the expected utilities for all possible courses of action and all possible outcomes would in general be extremely complex and time-consuming (Hanks 1990). For decisions about whether to take advantage of an opportunity, aspects that might be relevant include anything that might bear on how the pursuit of an opportunity will interact with ongoing plans that are intended to achieve other goals. This covers a great deal of territory. For example, the decision of whether to pick up a sack of cement might depend on whether there is a gas station nearby, in the case where the truck is low on gas; it might depend on whether there is a bridge with a low weight limit on the route that the truck plans to take, if the load is a heavy one; it might depend on whether cement thieves have been reported in the vicinity recently; and so on. If a system considers all the information that could potentially be relevant to a decision, it will be unlikely to complete the reasoning in time for it to be of any use: there are simply too many ways in which plans can interact with each other. The calculation of expected utility is thus not by itself an adequate theory of how an intelligent agent should react to opportunities. Intelligent agents must have a quick and easy way to decide whether the detailed reasoning will be worthwhile.

What is needed is *focus*: in addition to determining whether the pursuit of an opportunity is likely to interact significantly with ongoing plans, the system must identify the areas in which such interactions are likely to occur. These decisions must often be made rapidly if an opportunity is to be seized in a timely fashion. Because of this it is impractical to attempt to make such a determination analytically; instead, the system must reason heuristically. A major simplification that would significantly reduce the complexity of the reasoning required would be to assume that the agent's various goals are *independent*, *i.e.* the pursuit of one goal does not in any way interact with the pursuit of any other goals. Unfortunately, this assumption would deny the possibility of recognizing those circumstances in which the likelihood of adverse interactions should suggest that the opportunity not be pursued.

PARETO therefore uses a weaker version of the independence assumption. It assumes that its various goals are *effectively independent* of each other, *i.e.* that there are no *significant* interactions between them, unless it can infer otherwise (Pryor & Collins 1991). So, in our example, in the absence of evidence to the contrary the system would assume that picking up the cement would have no adverse effects on any other deliveries the truck might be making. If it is valid to assume effective independence of the agent's goals, the decision about whether to pursue an opportunity becomes much simpler, since all insignificant interactions with other goals can simply be ignored. However, if this assumption is to be used we need to be able to recognize potential violations of effective independence quickly and easily. PARETO uses heuristics that indicate potential violations to focus its attention on those aspects of the decision that are likely to repay more detailed analysis.

PARETO pursues plans[5] in order to achieve its delivery goals, and while pursuing them may notice oppor-

[3]Planning and Acting in Realistic Environments by Thinking about Opportunities.

[4]TRUCKWORLD simulates a world in which items can react in a rich variety of ways and can change state with the passing of time. The actions performed by the truck can fail for a variety of reasons, including chance, and other random events can occur.

[5]PARETO is based on Firby's (1990) RAPs system, and thus uses a hierarchy of sketchy plans. At any time one of these plans is active, and others are dormant, awaiting execution.

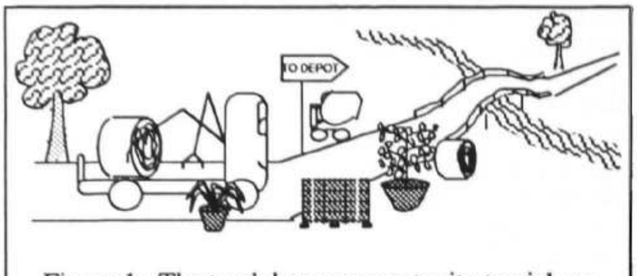

Figure 1: The truck has an opportunity to pick up some cinder blocks

tunities to achieve currently dormant goals. When an opportunity is recognized, a two-stage process is employed to determine whether or not that opportunity should be pursued. The first stage of the process involves the use of heuristics which flag potential violations of effective independence. If there are potential violations, PARETO moves on to the second stage and performs a more detailed analysis of the adverse interactions indicated by the heuristics.

## 2.1 An example

The types of decisions that PARETO faces when deciding whether to take advantage of an opportunity are illustrated in the following example. Suppose that the robot delivery truck is in the process of delivering some rolls of insulating material, the plan for which requires the truck to cross a bridge. Suppose further that there are some cinder blocks near the truck (figure 1), and that the truck has a currently inactive goal to deliver cinder blocks to a customer. The presence of the cinder blocks thus represents an opportunity to pursue an existing goal, and the agent must determine whether to pursue that opportunity.

In deciding whether or not to pursue the opportunity, the agent might in principle consider any number of possible interactions between its existing plans and the pursuit of the opportunity. The truck would have to stop by the side of the road to pick the cinder blocks up which might involve problems with passing traffic; the other objects by the side of the road might obstruct the truck during the loading process; cinder blocks are heavy, and the load might exceed the truck's weight capacity; they take up space in the truck, which might not be available; they are both hard and abrasive, and might damage other objects in the truck's load; other objects such as acid or iron beams might damage the cinder blocks; if they are not securely fastened to a pallet, they will be difficult to handle; loose cinder blocks might get damaged in transit; the time taken to pick them up might cause other delivery deadlines to be missed; picking them up might use more fuel than the truck has available; their weight might affect the fuel consumption and speed of the truck, and hence the truck's ability to make other deliveries; and so on. The number of possible interactions is enormous.

Unfortunately, any one of these interactions could actually constitute a serious threat. For example, let us suppose that in the situation depicted above the combined weight of the truck and the cinder blocks is greater than the bridge can bear. The problem confronting the agent is how to spot this problematic interaction without considering all the possible interactions in detail.

## 3. Reference features

An agent operating on the assumption that its various goals are effectively independent must be able to recognize when this assumption is inappropriate. This involves the ability to pick out the few genuinely problematic interactions from the potentially enormous number of harmless interactions in a given situation, and doing so quickly enough that the opportunity is not lost before the computation is complete. As the example above makes clear, this is by no means a trivial task. One approach is to tag elements of situations that are frequently involved in problematic interactions, and then to concentrate resources on detecting interactions involving the tagged elements.

This strategy can be seen as a simple application of common sense. For example, we might expect a nursery school teacher to take note of a child who is often involved in fights and disagreements, and to mark this child as a potential troublemaker to be watched closely in the future. In a similar way, our agent can tag potentially problematic elements in its planning environments, and use these labels to help it spot potential problems. By using different labels to designate different types of potential problems, the agent can in addition use these tags to focus subsequent analysis aimed at determining whether the problem will actually arise in the current situation. For instance, if objects made of a certain substance frequently break when they are involved in impacts with other objects, the agent can take note of that fact and mark such objects as fragile. When handling fragile objects, the agent should recognize that breakages are likely. Similarly, heavy objects often cause supporting structures to collapse, and bulky objects fill large volumes of space.

Objects are not the only elements of situations that can lead to unwanted interactions. For instance, when it is important that a goal be achieved within a short time period, there are often time conflicts with other tasks. By marking such goals as urgent, the agent can use that knowledge to avoid undertaking tasks that will interfere with their timely achievement.

We use the term *reference features* to denote tags such as disruptive, fragile, and urgent that help to direct an agent's attention to interesting functional aspects of the situation. In this paper we are primarily concerned with their use in indicating problematic interactions,

| Object | Description | Reference Features |
|---|---|---|
| insmat-2 | insulating-material | bulky |
| cblocks-6 | cinder-block | heavy<br>rough |
| bridge-23 | | rickety |
| customer-A | | impatient |
| road-57 | | bumpy |

Figure 2: Some reference features

but they also facilitate detecting a specified object, and can be used in planning to achieve goals.

PARETO uses heuristics based on reference features to indicate potentially problematic interactions involved in the pursuit of opportunities. There are several requirements that must be met in order for this strategy to be effective:

- The reference features must be easily recognizable.
- The agent must be able to determine quickly which elements of the situation may have reference features that indicate potentially adverse interactions.
- The agent must be able to use the reference features to indicate the *type* of the potentially problematic interaction.

There are many ways of meeting these requirements. In the current implementation of PARETO we are experimenting with a simple algorithm that looks for reference features indicating similar interaction types.

## 3.1 Availability

Reference features are useful only insofar as they provide cheap heuristics that indicate the desirability of more detailed reasoning. Reference features must therefore be easily inferable in most situations in which they are applicable, and must be inferable in few of the situations in which they are not. PARETO can link reference features to individual objects (it may know, for example, that a specific bridge is rickety), to descriptions that may apply to objects (the description cinder-block has the reference feature heavy attached to it), to sketchy plans and actions (which may be, for example, lengthy), and to goals (*e.g.* urgent). In TRUCKWORLD it is easy[6] to observe, for example, that an object is a stack of cinder blocks: the fact that the object has the reference feature heavy can then be inferred. Figure 2 shows some of the reference features in our example.

## 3.2 Situation elements

Since reference features are associated with elements of the situation in which PARETO finds itself, PARETO must be able to determine which elements of the situation are relevant. PARETO uses sketchy plans (Firby 1989) that comprise, among other things, a list of actions to be executed and the goal that the plan serves. Action descriptions consist of an action predicate applied to a set of objects (see figure 3). The set of situational elements associated with a given plan thus consists of all the objects that play a role in any primitive action, the actions themselves, and the goal that the plan serves.

| Goal | (deliver ?item ?dest) | (travel-to ?dest) |
|---|---|---|
| Plan steps | (travel-to ?item-loc)<br>(load ?item)<br>(travel-to ?dest)<br>(unload ?item) | (traverse ?road1)<br>(traverse ?road2)<br>(traverse ?road3) |
| Current plan Variables | (?item insmat-2)<br>(?destination cust-A) | (?destination cust-A)<br>(?road1 road-42)<br>(?road2 bridge-23)<br>(?road3 road-31) |
| Opportunity Variables | (?item cblocks-6)<br>(?destination cust-B) | (?destination cust-B)<br>(?road1 road-45)<br>(?road2 road-57)<br>(?road3 road-76) |

Figure 3: Sketchy plans

When PARETO is considering whether to pursue an opportunity, it examines both the plan it is currently executing and the plan that would be used to pursue the opportunity, collecting the relevant situational elements from each. It then checks to see whether any of these elements is associated with a reference feature, and, if so, it flags that element.

In our example, PARETO's current plan is to deliver some insulating material to customer-A. An outline of the sketchy plan for this task is shown in figure 3. The sketchy plan (travel-to cust-A), for example, consists of the three steps: (traverse road-42), (traverse bridge-23) and (traverse road-31). Similarly, the plan for pursuing the opportunity , involving the goal (deliver cblocks-6 cust-B), is a different instantiation of the same sketchy plan. The reference features of bridge-23 (rickety) and cblocks-6 (heavy, rough) are therefore among those that are relevant to the decision of whether to pursue the opportunity to deliver the cinder blocks.

## 3.3 Focusing reasoning

In addition to flagging potential violations of effective independence, reference features play a role in focusing the agent's reasoning onto the particular aspects of the situation that should be considered in determining whether the violation will actually occur. In the example described above, for instance, there is a potentially problematic interaction involving the rickety bridge and the heavy cinder blocks. PARETO's analysis should concentrate on the question of whether the bridge is likely to collapse under the weight of the blocks.

The knowledge that PARETO needs in order to guide the analysis process is associated with reference fea-

[6] Perception in TRUCKWORLD grounds out at the level of object descriptions, and thus ignores the many important problems of object recognition.

| REFERENCE FEATURES OF TASKS | INTERACTION TYPE | ROLE OF TASK |
|---|---|---|
| lengthy | time<br>fuel | consumer<br>consumer |
| urgent | time | requirer |

| REFERENCE FEATURES OF TERRAINS | INTERACTION TYPE | ROLE OF TERRAIN |
|---|---|---|
| bumpy | impact | cause |
| narrow | volume-capacity | limiter |
| rickety | load-bearing | base |

| REFERENCE FEATURES OF OBJECTS | INTERACTION TYPE | ROLE OF OBJECT |
|---|---|---|
| bulky | volume-capacity | consumer |
| explosive | fire | igniter |
| flammable | fire | burner |
| fragile | impact<br>time | hit<br>consumer |
| hard | impact | hitter |
| heavy | fuel<br>load-bearing | consumer<br>load |
| rough | surface-damage | abrader |

Figure 4: Reference features and interactions

tures themselves. Each reference feature predicts a particular type of problematic interaction (see figure 4), which is represented by an *interaction description* consisting of three items: the configuration that is required in order for the interaction associated with the feature to occur, the potentially problematic outcome of the interaction, and a list of variables designating the elements that play a role in the interaction (see figure 5). When a situational element is flagged with a reference feature, PARETO must determine whether the interaction description associated with that feature applies to that element in the current situation.

In order to determine whether an interaction description applies, PARETO must first determine which other elements of the situation could be involved in such an interaction with the flagged element. For instance, the pile of cinder blocks in the example creates the potential for an interaction in which an object supporting the blocks collapses, but this does not tell us which, if any, specific objects are in danger of collapsing. One approach might be to examine every element of the situation to determine if any are likely to be involved in this interaction; in effect, this would be like asking, for every element, whether it is ever likely to support the cinder blocks, and, if so, whether it can bear their weight. While this is possible, it involves a somewhat unfocused search.

A more focused solution is based on the observation that different reference features may be used to designate objects that play different roles in the same type of interaction. For example, heavy means that an object is likely to make the object that supports it collapse, while rickety means that an object is likely to collapse under an object it is supporting. The features heavy and rickety thus form a natural pair. This knowledge can be used to focus the analysis by considering interactions only when at least one object has been

| Interaction type: | load-bearing | surface-damage |
|---|---|---|
| Roles: | ?base ?load | ?abrader ?hurt |
| Configuration: | (bears ?base ?load) | (rubs ?abrader ?hurt) |
| Outcome: | (collapse ?base) | (scratched ?hurt) |

Figure 5: Interactions

flagged for each role in the interaction. In our example, PARETO considers the possibility that something will collapse only when it recognizes that both a heavy object—the blocks—and a rickety object—the bridge—have been flagged.

Once PARETO knows the type of problematic interaction it is looking for, and the elements that are involved in that interaction, it must determine whether the interaction will actually occur. This involves two steps: determining whether the *configuration* described in the interaction description will arise, and determining whether the problematic *outcome* will result if it does. PARETO must therefore be able to perform inference over the causal theory of the planning environment, which must include a theory of action projection in addition to the causal relationships among the objects in the world. We are assuming that this inference is performed by a general purpose query system since we have not yet addressed the problem of a a more specialized system for plan projection.

In our example, then, PARETO should query whether, if the opportunity is pursued, the cinder blocks will at some stage be supported by the bridge. In this case it will discover that this configuration condition will indeed occur. PARETO must therefore decide whether the pursuit of the opportunity would result in the problematic outcome of the interaction, or if that outcome can be avoided.

In order to answer this query, PARETO applies inference rules describing the causality of the domain, for example that the bridge will collapse if the total load on the bridge is greater than its weight limit. In order to use this particular rule, PARETO will need to know the weight limit on the bridge, and the weight of the total load on the bridge at the time when the configuration conditions are met. Further inference rules will enable it to recognize that the total load on the bridge will consist of the truck, its current contents, and the cinder blocks, whose weights must be added together. These individual weights are therefore pieces of information that will be useful to PARETO in making the decision. However, there may be costs involved in acquiring this information (for further discussion of this point see Pryor & Collins 1992). PARETO must therefore consider the value and acquisition costs of each piece of information. For example, if the weight limit on the bridge is very high compared to the weight of any load the truck would carry, information about the weight of the truck's load would not change the decision as to whether to pick up the cinder blocks, and so is not worth acquiring.

## 4. Discussion

We have described the design of a system, PARETO, that reasons efficiently about whether to pursue an opportunity. The keys to PARETO's approach are, first, that it assumes that its various goals are effectively independent of each other, and, second, that it uses reference features to flag situations in which this assumption of effective independence is likely to be violated. By using reference features to indicate potentially adverse interactions between an opportunity and its current plans, PARETO is able to ignore the many insignificant interactions that may be present.

It is important to note that reference features are not infallible: clearly there may be a problematic interaction that is not indicated by any reference features. As a result, PARETO will occasionally produce incorrect plans. We believe that such an outcome is unavoidable for any planner that is intended to operate in a complex, and unpredictable environment. the upshot of this is that such a system must be able to recover from errors and unforeseen failures, and to learn from its mistakes. One way in which such a system might be expected to learn from mistakes is by positing new reference features indicating problematic interactions that the system has observed. Reference features thus form a natural basis for a theory of learning to plan. We will pursue this issue further in future work.

**Acknowledgments**: Thanks to Larry Birnbaum, Matt Brand, Will Fitzgerald, Mike Freed, and Bruce Krulwich for many useful discussions.

### References

Agre, P. E., & Chapman, D. (1987). Pengi: An implementation of a theory of activity. Proceedings of AAAI 87.

Allen, J., Hendler, J., & Tate, A. (Eds) (1990). Readings in Planning. Morgan Kaufman.

Beer, R. D., Chiel, H. J., & Sterling, L. S. (1990). A Biological Perspective on Autonomous Agent Design. Robotics and Autonomous Systems 6, 169-186.

Birnbaum, L. (1986). Integrated processing in planning and understanding. Technical Report YALEU/CSD/RR #489, Dept. of Computer Science, Yale University, New Haven CT.

Brand, M., & Birnbaum, L. (1990). Noticing Opportunities in a Rich Environment. Proceedings of the Twelfth Annual Conference of the Cognitive Science Society, Cambridge MA.

Brooks, R. (1986). A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, vol. 2, no 1

Chapman, D. (1987). Planning for Conjunctive Goals. Artificial Intelligence, 32, 333-337. Also in (Allen *et al* 1990).

Feldman, J. A., & Sproull, R. F. (1977). Decision Theory and Artificial Intelligence II: The Hungry Monkey. Cognitive Science 1, 158-192, and in (Allen *et al.* 1990).

Fikes, R. E., & Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence 2, 189-208, and in (Allen *et al.* 1990).

Firby, R. J. (1989). Adaptive execution in complex dynamic worlds. Technical Report YALEU/CSD/RR #672, Dept. of Computer Science, Yale University, New Haven CT.

Firby, R. J., & Hanks, S. (1987). The simulator manual. Technical Report YALEU/CSD/RR #563, Dept. of Computer Science, Yale University, New Haven CT.

Georgeff, M., & Lansky, A. (1987). Reactive Reasoning and Planning. Proceedings of AAAI 1987, and in (Allen *et al* 1990).

Haddawy, P., & Hanks, S. (1990). Issues in Decision-Theoretic Planning: Symbolic Goals and Numeric Utilities. Innovative Approaches to Planning, Scheduling and Control: Proceedings of a Workshop. DARPA, San Diego.

Hammond, K. J., Converse, T., & Marks, M. (1988). Learning from opportunities: Storing and reusing execution-time optimizations. Proceedings of AAAI 1988.

Hammond, K., Converse, T., & Martin, C. (1990). Integrating Planning and Acting in a Case-Based Framework. Proceedings of the Eighth National Conference on Artificial Intelligence, Boston,

Hanks, S. J. (1990). Projecting plans for uncertain worlds. YALEU/CSD/RR #756, Yale University, New Haven CT.

Kaelbling, L. P., & Rosenschein, S. J. (1990). Action and Planning in Embedded Agents. Robotics and Autonomous Systems 6 35-48.

McAllester, D., & Rosenblitt, D. (1991). Systematic Nonlinear Planning. Proceedings of the Ninth National Conference on Artificial Intelligence, Anaheim, CA. pp. 634-639.

McDermott, D. (1991). Robot planning. Technical Report YALEU/CSD/RR#861, Dept. of Computer Science, Yale University, New Haven CT.

Newell, A., & Simon, H. A. (1963). GPS, a program that simulates human thought. In E.A. Feigenbaum & J. Feldman (Eds), Computers and Thought, R. Oldenbourg K.G., and in (Allen *et al* 1990).

Pryor, L., & Collins, G. (1991). Information gathering as a planning task. In Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society. Chicago, IL.

Pryor, L., & Collins, G. (1992). Planning to perceive: A utilitarian approach. In Working Notes of the AAAI Spring symposium on Selective Perception.

Sacerdoti, E. (1977). A Structure for Plans and Behavior. American Elsevier, New York.

Simmons, R. (1990). An architecture for coordinating planning, sensing, and action. DARPA Planning Workshop, San Diego, CA.

Skinner, B. F. (1974). About Behaviorism. Random House, New York.

Tate, A. (1977). Generating project networks. IJCAI 1977, and in (Allen *et al* 1990).

Von Neumann, J., & Morgenstern, O. (1944). The Theory of Games and Economic Behavior. Princeton University Press.

Wilkins, D. E. (1988). Practical Planning: Extending the Classical AI Planning Paradigm. Morgan Kaufman, CA.