A structure for systems that plan abstractly

by WALTER JACOBS

The American University
Washington, D.C.

INTRODUCTION

The advantages of introducing planning procedures in systems that deal with complex tasks are generally recognized. Here the special benefits of using abstraction in the planning process will be considered, and a structure particularly well suited for abstract planning, the purposive system, will be described. An example will be presented to show how abstract planning operates under the purposive system; this is the program PERCY, which simulates the behavior of a nest-building insect.

There are superficial points of likeness between PERCY and L. Friedman's ADROIT (Friedman 1967, 1969). ADROIT is also a simulation of a nest-building insect, but PERCY does much more, using a considerably shorter and simpler program. The value of the purposive system structure is clearly shown by this example.

The concept of abstraction in planning is explained in the second section as the making of decisions without exploring the course of actions involved in executing these decisions. Instead of such exploration, a set of goals intermediate to the principal task are evaluated. The example of PERCY is described in the third section. It shows how the selection of intermediate goals can be separated from the translation into action of the corresponding decisions.

The fourth section discusses the hierarchical structure that makes the separation possible. The process of abstraction takes place in the course of communication between adjacent levels of the hierarchy. The operations of the purposive system are explained in the fifth section; they are few in number, simple, and independent of the task. The system's information about the task is contained in its knowledge, organized in a data structure of standard format. A summary describing the results achieved in the PERCY application is given in the concluding section.

DECISIONS, PLANNING AND ABSTRACTION

Abstraction in the sense used here is applicable to a system whose tasks require continual interaction with its environment. For such a system, the term action refers to an external commitment—an element of the system's observable behavior. A decision, on the other hand, is an internal operation. The term implies that a number of alternatives are evaluated and a selection is made. The decision then determines, directly or through some intervening process, what action or set of actions will follow.

It is useful to consider some examples. The nest-building insect ADROIT mentioned above provides an example where decisions as just defined are not made. Sequences of actions that accomplish desired objectives are controlled by a hierarchy of routines called Release Mechanisms and Selector Mechanisms. But these sequences are simply triggered by conditions in the environment; no evaluation of alternatives takes place.

In A. Samuel's checker-playing program (Samuel, 1959), and in game-playing programs generally, each move made by the program is an action. There is one decision for each action, and evaluation of the alternatives considered in reaching the decision is done by analyzing possible sequences of moves to follow the one being considered.

The SRI robot "Shakey" follows a pattern quite different from either of these (Nilsson, 1969). It uses a problem-solving approach, in which the entire sequence of actions to handle the task is worked out on an internally represented model of the environment before the first action is taken. Thus, there is only one decision per instance of a task, except where a mismatch between the environment and its model causes the planned solution to break down during execution. Again, as in the game-playing programs, analysis involves exploring possible sequences of actions.

The kind of decision making to be described does not
concern itself with the specific actions that may result from the decision that is selected. Rather, it considers intermediate goals, or way stations on the road to completion of the task. In abstract planning, a decision selects a goal among those that are available at the time the decision is faced, and this selection sets in motion a course of action that will arrive at some outcome—usually one of several that are possible. At this point a new decision must be made. Thus, there are a number of decisions taken in completing a task, and a very much larger number of actions.

Evaluation of goals is based on prior experience with similar decisions rather than on analysis of a course of action. The likelihood of each outcome is estimated, as well as the satisfaction to be gained in arriving at the outcome. Thus, an estimate of the expected satisfaction from selection of a goal is obtained, and these expectations are used in reaching a decision.

Past experience can be used because a decision remains in force till an outcome occurs. Which outcome takes place depends on what is met in the environment. In abstract planning, there is an assumption that the probabilities are stable.

Thus, abstract planning frees the system from the burden of analyzing the actual process of trying to realize a desired outcome. It must be used when the system is not able to explore in an internal model the explicit consequences of a course of action.

It is worth emphasizing that in most man-machine systems, abstract planning is the man's role in the process. The introduction of it into machine systems is a step—though far from the final one—toward having these systems handle their tasks in a way that deserves to be called "intelligent."

PERCY—AN EXAMPLE OF AN ABSTRACT PLANNING SYSTEM

The environment in which PERCY operates is pictured in Figure 1. There is a nest location, with eight individual sites at which material is to be placed in order to complete the nest-building task. There are a number of clumps of material (indicated by x's), a part of the upper area where food (small triangles) is hunted, and some obstacles to be avoided—in particular, the wall that separates the upper and lower areas. In addition, there are landmarks (small squares) which are visible at a distance, and are used by PERCY in reaching spots from which it can see its main targets.

PERCY’s task (there is only one in the present version) is to build its nest. To do this calls for eight trips. On each trip, one piece of material is located, picked up, carried back to the nest, and placed at an empty site. On some of the trips, PERCY will also get food.

PERCY uses abstract planning for its trips, and makes decisions that try to maintain a high level of satisfaction. But its satisfaction will drop if it devotes too small a part of its time to the search for food, and gets hungry as a result. Satisfaction goes down also when it spends too much time getting food, and progress on the nest lags. Thus, in order to have good performance, the decisions it faces at various stages of the task must bring about a proper allocation of its time between getting food and building its nest. PERCY must rely on abstract planning because it has no internal model of the environment on which to explore in detail the consequences of a course of action.

Its observable behavior consists of a series of elementary actions. Each of these is produced in a cycle of behavior, which takes up a brief moment of PERCY's lifetime. A cycle begins with the receipt of a stimulus from the environment, continues by determining the desired response, and ends with the action that forms the response.

In the nest-building task, PERCY uses only five

Figure 1
elementary actions. They are:

(i) a short movement in the direction of perceived target,
(ii) a short side-step to avoid an obstacle,
(iii) a turn through a limited angle,
(iv) a movement partly extending its arm,
(v) a movement partly retracting its arm.

To accomplish even a limited objective, a number of these elementary actions must be strung together. To reach an outcome of a selected goal may require a large number of such actions in proper sequence, all initiated by a single decision. An example of such a sequence from a run of PERCY’s task will be given shortly.

Decisions are called for only at certain stages of PERCY’s task. One such stage occurs when PERCY has just completed a trip by adding material to the nest, and is about to start another trip. At this point, it must decide whether food or material should be sought first. A principal factor in this decision is the amount of time that has elapsed since it last had food.

Assigning priority to food accounts for one of the goals open at this stage. There are three others, for if PERCY wants to look for material first, there are three sections of the environment it may head for: the upper area shown in Figure 1, and the lower and upper sections of the other area. (PERCY is unable to distinguish between the two landmarks in the same section; it simply heads for the first landmark it sees in the chosen section.)

To specify the goal, it is sufficient to name the classes of targets that are of interest during the time that goal is pursued, and to identify which of them is the principal target. For example, among the four goals just described, the one that gives priority to food is specified as follows:

Food is the principal target, the door landmark and material are alternate targets.

The “door landmark” is the one at the entrance to the upper area; it must be used because PERCY cannot see any target when the wall is in between.

Each goal has several outcomes. The one just described has three outcomes possible: Food will be obtained, or material will be spotted on the way, or something “illegal” may take place. This last outcome occurs when the environment fails to behave as called for in PERCY’s program. For example, if the door landmark cannot be located, PERCY will go on searching till it reaches a point where it gives up the task. If no way out of the impasse were available, PERCY would continue to search until it “died.”

PERCY uses prior experience to estimate, for each goal considered, the probability that each outcome will occur and the length of time that will be needed to reach that outcome if it does occur. Taking account of its current state of hunger and progress on the nest, together with the estimated time that will elapse, it further estimates what its satisfaction will be should that outcome happen. Applying its probability estimates, it then arrives at an estimate of expected satisfaction for the goal being evaluated. Finally, it decides on the goal with the highest expectation.

Nowhere in this process of evaluation does it give consideration to the actions that will follow its decision. Its evaluations deal only with measures that are quite detached from its environmental interactions. In spite of this, its decisions give rise to coherent and purposeful series of actions that can go on for many cycles of behavior before a new decision is required.

Figure 2 (showing the output of the program as plotted on an SC 4060 printer) depicts the greater part of one of PERCY’S trips. PERCY’s movements during this trip are marked by a trail of dots, which show the successive positions of PERCY’s turning center. The interval between two dots when PERCY is headed toward a target indicates how far it moves in a single action. The direction in which it is facing is shown by an
arrow, every fifth cycle of behavior. When arrows are shown without four dots intervening, PERCY is turning, or, in one case, standing and eating food.

Only three decisions were involved in this series, involving more than 300 individual actions by the time the trip was finished. About half of this total derived from the decision to seek food after material was picked up at the location shown in the figure.

This single food-hunting sequence thus took nearly 150 cycles. It was made up of four parts, in each of which PERCY was executing a limited objective, or subtask, contributing to the desired goal. The parts, and their corresponding objectives, were as follows:

(i) A series of turns, searching for a target of interest, and ending when the door landmark was seen.
(ii) A series of moves in the direction of that landmark, aiming to get near it, and ending when the objective was attained.
(iii) Another series of turns, searching for food, and ending when food was spotted.
(iv) A series of moves in pursuit of the food, that brought PERCY within arm’s length of it; a sidestep to avoid the wall occurred at one point. When PERCY was near the food, it extended its arm in a series of movements, seized and ate the food, and returned its arm to the rest position.

This behavior was generated in face of an environment that is known only in the most limited way, and sensed very crudely. For example, PERCY has only the simplest ability to discriminate distances; it uses only the three categories “far,” “near” and “at arm’s length.” The only information it has about the location of key features of its environment is contained in the way it associates landmarks with its goals.

It clearly takes a program of some complexity to translate a single goal into a cogent course of action in the face of an environment that is so poorly known. The next two sections describe how the purposive system makes it possible to reuse the major part of that program for other goals, and thus to handle a complex task successfully with a program of moderate size. For further information about PERCY, see Jacobs, 1971.

THE HIERARCHY OF ABSTRACT PLANNING

The ability to realize an economical system for abstract planning with a variety of goals depends critically on the way programs for individual goals are combined. The purposive system uses a hierarchical organization to achieve this economy. Further, the structure of the system is independent of the nature of the tasks handled. This is why the greater part of the program is common not only to different goals, but also to distinct tasks, as long as they use the same set of perceptions and actions.

Thus, it would take a relatively modest enlargement of the present PERCY program to add an exploration task. This would precede the nest-building, and would find a suitable place for a nest, and also pick the landmarks to be used in the later task. If other bugs were added to PERCY’s environment, courting and fighting tasks could be readily introduced.

The hierarchy in the purposive system has four levels, corresponding to four components in a physical realization of the system. These are called the body, task, method and strategy components.

The body component forms the system’s interface with the environment. It contains the sensory and motor apparatus; thus it receives the stimuli that originate in the environment, and carries out the actions that respond to these stimuli.

The task component controls the operation of the body component. It coordinates sequences of actions into subtasks, such as those described in Section 3. It frees the higher components from any concern with the details of environmental interaction.

The method component guides the task component through the succession of subtasks appropriate for a given goal. Since conditions in the environment determine how a subtask will wind up, management at this level requires more than a linear sequencing of subtasks. The conversion of a decision into its subtasks is expressed as a submethod, which resembles a subtask in structure.

The strategy component evaluates and selects goals. Since the other components take complete responsibility for execution of its decisions, this component is concerned only with the outcomes of its decisions.

There is an obvious analogy in this hierarchy with the organization of a large enterprise. The strategy component corresponds to the top executive level, while the body component acts like the mass of employees who deal with the suppliers and customers. The other two components correspond to the intervening levels of management.

Figure 3 names the communications that are passed in support of this hierarchy. (In general, the received form of the message is distinguished from the issued form.) The way these communications are handled is important in allowing the system to operate simply and economically. The only communications that occur
Internally are the following:

(i) Once in every cycle of behavior, the body component reports a perception to the task component. In return it receives an intention. This specifies the action to be performed, and also specifies the kinds of targets that are of interest in forming the perception for the next cycle.

(ii) When the subtask reaches a conclusion, the task component reports the situation that has been reached to the method component. That component issues a plan that specifies the next subtask to be performed, and identifies the kinds of targets involved.

(iii) When the submethod finally arrives at an outcome, the plan in effect is communicated to the strategy component. That component sends back its goal, which sets the next submethod in motion.

Under this scheme of operation, the body component is continually active in receiving stimuli from the environment and issuing responses. At successively higher levels, each component is involved less frequently than its predecessor, and is more detached from what is going on in the environment. The term "abstract" planning is therefore an apt description of the way the system controls its behavior.

The upward and downward flows in Figure 3 may be found suggestive of the afferent and efferent flows in the Central Nervous System of a living creature. There, too, a decision can be turned over to lower levels of the CNS for execution without demanding the conscious attention of the higher brain centers. This analogy will be pursued elsewhere, for it can be shown that the structure and operations of the purposive system have implications for a theory of the functioning of the Central Nervous System in cognitive activity.

The operation of the purposive system

The components of the purposive system function as finite automata. In this way the structure takes maximum advantage of the use of abstraction. The inputs and outputs of each component are simply the communications that have been described, together with external stimuli and responses of the body component. The state variable of the component provides the task context, appropriately abstracted, for processing the communications most simply and efficiently.

Except in the body component, each state variable acts as a pointer to an entry in a list structure. This structure is called the knowledge of the system. It contains, in a general format, all information specific to the task. The knowledge, as acted on by the general operations of the system, simply expresses the two functions—the state transition function and the output
function—that describe the component as a finite automation.

This is a very efficient arrangement. Its efficiency is enhanced by introducing, as parameters of the knowledge of subtasks and submethods, the specific objectives of a class of goals. Then a single entry in knowledge can apply to a whole group of state variables that differ only in their parameters. This makes it possible to execute a variety of subgoals with a relatively small number of subtasks and submethods. Here again, the structure is integrally bound up with the process of abstraction.

An example from PERCY’s program may help to illustrate these points. The following stage of the task is assumed to have been reached: PERCY is hunting food, has not yet picked up material on the current trip, and has the door landmark in sight. The system’s state (not including the knowledge of the task) is given by four variables:

(i) The goal (state of the strategy component) indicates that food is the principal target; the alternate targets are the door landmark and any item of material.

(ii) The plan (state of the method component) indicates which subtask is in progress: i.e., going to a sighted landmark. Its parameters are the identities of the landmark and of the alternate targets (food and material) whose sighting would also bring the subtask to an end.

(iii) The situation (state of the task component) specifies the stage reached in the subtask: The landmark is visible but not near, and neither alternate target nor obstacle is in sight.

(iv) The perception (state of the body component) records that the principal target is in sight but still “far away.” No other item of interest is in sight. The arm is in the rest position, and nothing is held.

It has been mentioned that the first three of these states act as pointers to entries in knowledge. For example, the entry that is pointed to by the situation (iii) just described contains three items of information:

(i) The intention, or output of the task component. The intention in this case specifies the action of moving toward the principal target, and also identifies the types of the principal and alternate targets.

(ii) The criteria by means of which the set of inputs—i.e., the possible perceptions—is partitioned into events. The events are the distinct meanings that the next reported perception can have in this situation. They include the following: “no change,” “principal target near,” “alternate target sighted,” “obstacle in way,” and “target not seen.”

(iii) For each event, the new situation that points to the knowledge entry needed should that event be recognized. For example, if the event is “no change,” the new situation is the same as its predecessor; if the event is “alternate target sighted,” the new situation is the one that marks the end of the subtask, with the particular target sighted as parameter.

With this format, only the simplest and most general operations are needed in order to make the task component function as a finite automaton and, in so doing, carry out its role in the production of behavior. These operations include receiving the perception from the task component, applying the criteria in knowledge to recognize the event, replacing the situation by the new one for that event, and issuing the output located in the knowledge entry for the new situation. Except when the subtask is done, this output is the intention and it goes to the body component.

If the subtask is done, the output of the new situation is the situation descriptor itself, and it goes to the method component. In that case, the method component will return a plan. The plan is recognized as a choice, and this input to the task component will cause a second change of situation in the same cycle. This second situation will issue an intention to begin the new subtask. The process has already been shown schematically in Figure 3.

The same operations, using the same format of knowledge, take place in the method component when a subtask ends. In this case the input from the task component is the situation, and its recognized form is the feature. After the new plan replaces its predecessor, the output to the task component is that plan. And when the submethod ends, the plan is reported to the strategy component. The goal that is received in return is recognized as the decision, which then causes another change of state to the initial plan of the next submethod.

When a submethod comes to an end and an input to the strategy component takes place, it is recognized as an outcome of the goal that has been in effect. However, a new operation—evaluation—takes place at this level in arriving at the new state of the component, and an appropriate modification of the structure of knowledge is necessary.

A block diagram summarizing the operation of the purposive system is shown in Figure 4. The RECOGNIZE operation carries out the task-independent steps just described: It applies the criteria listed in
the knowledge entry to the other input, selects the listed alternative that best matches that input, and issues the corresponding output. This points to the knowledge entry that replaces the one just used, and activates the communication needed for the next step of the process.

RESULTS OF THE APPROACH

A summary of the results accomplished on PERCY's task is given here as a basis for assessing the potential of abstract planning. That task was chosen as one requiring an appreciable number of nonroutine decisions, producing activity of long duration without human intervention. (Subject to this, the example was constructed with an eye to minimizing the labor of simulating an environment and the necessary interactions with it.)

In the process of completing its nest, PERCY used its five types of subtasks about 80 times. They required more than 1900 individual actions. The course taken by each subtask or action depended generally on the particular targets of the current goal as well as on the external conditions that were encountered.

<table>
<thead>
<tr>
<th>Number of Trip</th>
<th>Outcome</th>
<th>Description of Outcome</th>
<th>Number of Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Material taken at (205, 15)</td>
<td>33</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Food eaten at (228, 410)</td>
<td>147</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Material placed at (246, 120)</td>
<td>124</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Material taken at (115, 335)</td>
<td>68</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Material placed at (250, 130)</td>
<td>68</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Material taken at (205, 320)</td>
<td>56</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Food eaten at (189, 370)</td>
<td>75</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Material placed at (260, 134)</td>
<td>117</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Material taken at (115, 340)</td>
<td>78</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Material placed at (270, 139)</td>
<td>81</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>Material taken at (110, 340)</td>
<td>87</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Food eaten at (217, 390)</td>
<td>60</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Material placed at (274, 129)</td>
<td>136</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>Material taken at (205, 320)</td>
<td>69</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Material placed at (270, 110)</td>
<td>78</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>Material taken at (110, 335)</td>
<td>75</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Food eaten at (234, 370)</td>
<td>64</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Material placed at (260, 106)</td>
<td>152</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>Material taken at (280, 320)</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Food eaten at (270, 390)</td>
<td>65</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Material placed at (250, 110)</td>
<td>171</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>Material taken at (200, 320)</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Food eaten at (276, 390)</td>
<td>98</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Material placed at (250, 110)</td>
<td>1902</td>
</tr>
</tbody>
</table>

Figure 5—Sequence of outcomes in a single run of PERCY's nest-building task.

PERCY had to make successful decisions in order to show good performance. No constraints were imposed on these decisions to keep PERCY from choosing unsatisfactory courses of action. Further, it operated with only the most summary memory of what happened in past decisions, and with almost no information about environmental conditions on which the quality of its decisions depended. It had no way of exploring the future beyond the point at which the next decision would be made.

In spite of these handicaps, PERCY was able to perform in impressively successful fashion. Figure 5 shows the actual outcomes of its series of decisions. The coordinates of the places where material and food were taken, and the nest locations where material was placed, are used to indicate the outcomes. The behavior in arriving at these outcomes was in every case as direct, purposeful and non-tentative as in the example of Figure 2. On the other hand, there was no mechanical or trivial pattern to this behavior.

PERCY did not spend an excessive amount of time going after food, which was hunted in only five of the eight trips. It also avoided the opposite error of subjecting itself to severe hunger. To an observer, every one of its many actions would appear meaningful in relation to its task.

The structure of the purposive system made it possible to realize these results with a comparatively simple program. Written in Fortran (not the most

---

Figure 4—Block diagram of purposive system
efficient language for the purpose), the program con-
tained about 600 instructions, although no special effort
was made to hold down this total. The complete run on
a third-generation computer took about 3 seconds of
CPU time.

The problem space in which PERCY operated was
larger by many orders of magnitude than any that have
been successfully tackled by a general problem-solving
system. PERCY’s accomplishments are thus a clear
demonstration of the power of the abstract planning
approach. Clearly, the way people handle most of their
tasks is much closer to this pattern than it is to other
existing general-purpose systems for problem solving,
which use extensive exploration of a detailed internal
representation of the problem.

In pointing this out, there is no intention to claim
that the present approach deserves to be described as
“intelligent.” It seems safe to assert that in order to
merit that characterization, a system must possess both
problem-solving and abstract planning capabilities, and
other capabilities as well. Nevertheless, it is believed
that for many kinds of tasks, the abstract planning
approach will turn out to be an efficient, economical way
to deal with them.

BIBLIOGRAPHY

L FRIEDMAN
Instinctive behavior and its computer synthesis
Behavioral Science Vol 12 No 2

Robot control strategy
Proceedings of the International Joint Conference on
Artificial Intelligence.
Bedford Mass The Mitre Corp 1969

W JACOBS
Help stamp out programming
Theoretical Approaches to Non-Numerical Problem
Solving
Banerji and Mesarovic ed Berlin Springer-Verlag 1970

How a bug’s mind works
Proceedings American Society for Cybernetics Fourth
Symposium
To appear 1971

N NILSSON
A mobile automation
Proceedings of the International Joint Conference on
Artificial Intelligence
Bedford Mass The Mitre Corp 1969

A SAMUEL
Some studies in machine learning using the game of checkers
IBM Journal of Research and Development
Vol 3 No 3 1959