

PLAN RECOGNITION
USING A HYPOTHEZIZE AND REVISE PARADIGM:
An Example

Charles F. Schmidt and N.S. Sridharan

Departments of Psychology and Computer Science
Rutgers University, New Brunswick, NJ 08903

A great deal of research has focused on the problem of how to generate a plan that satisfies a given goal [2]. The problem of plan recognition is to take as input the sequence of actions performed by an actor and to identify the goal pursued by the actor and also to organize the action sequence in terms of a plan structure. This plan structure explicitly describes the goal-subgoal relations among its component units. Our concern with plan recognition has arisen from our research [4,5,6,7] in the development of a theory of how persons understand the observed actions of others. A variety of experiments [7] have demonstrated that the summary as well as the recall protocols of an action sequence can be predicted if it is assumed that the observer is actively reconstructing from the observations the plan or plans that the actor is executing.

The psychological theory of plan recognition, BELIEVER, consists of a set of knowledge structures and processes defined within the AIMDS system [11]. The specification of the knowledge sources for the BELIEVER model arose from considerations of the properties of the problem domain and do not rest on any particular psychological assumptions about properties of the human information processing system. For example, the observed sequence of actions that serve as input to a plan recognition system do not provide information that identifies which of the set of outcomes associated with an action sequence corresponds to the goal or goals of the actor. In fact, the actor's plan may have failed or been abandoned or interrupted in which case the goal outcome would not be represented at all in the action sequence. If cases such as these are to be recognized, then a plan recognition process must be capable of counterfactual reasoning that shows how a hypothesized goal could have been achieved had some aspect of the physical world been otherwise. Further, actions are

linearly ordered and unbounded in time. Plans are bounded, hierarchical, and nonlinearly ordered based on the logical connection between an enduring outcome of one action and the precondition of some subsequent action. Further, a plan is well-formed with reference to the planner's beliefs about the world rather than with respect to the actual state of the world. For these and similar reasons, we have argued that any process of plan recognition must use meta-knowledge about plans and the psychological constraints which define a well-formed plan. Knowledge of only the physical constraints on action does not provide sufficient information to recognize plans [8].

In this paper we present, by means of an example, those aspects of a plan recognition process that have been implemented in BELIEVER. The properties of this process have been motivated primarily by our desire to incorporate within this process certain psychological assumptions about characteristics of the human plan recognition process. The two major assumptions are that the human process is: (1) a general non-specialized process; and (2) is based on a hypothesize and revise strategy. The remainder of this paper is concerned with conveying the meaning of these labels. However, it will be useful to first explicate these ideas in an informal way.

Because of interest in application or for methodological reasons, much of the recent work in AI has been concerned with the representation of expert knowledge within a relatively narrow domain. Chess, restaurants, flush toilets and particular electronic circuits are well-known examples of this approach. An expert's knowledge is, by definition, highly specialized and customized to a particular domain. This has led researchers to encode this knowledge in similarly specialized and customized forms which are often generically referred to as scripts [3]. The use of script-like knowledge is an important aspect of the human information processing capability.

However, very few of us are expert in everything and none of us are born that way. Specialized, customized knowledge must be acquired using quite general and schematic knowledge [9]. Accordingly, the system we have designed and implemented uses multiple communicating knowledge sources to customize and specialize the general knowledge about acts and plans to specific observational input.

The knowledge needed by a process of plan recognition has been decomposed into three major knowledge sources. These knowledge sources together with some of the major lines of communication between them are shown in Figure 1. In this figure the knowledge sources are referred to as the Person, Plan, and World Domains. Each knowledge domain defines the concepts, relations, and logical dependencies between concepts that govern the representation of particular instances of objects within that domain. The boxes labeled Person Model (PM), Expectation Structure (ES), and World Model (WM) graphically represent the different contexts within which particular assertions are maintained. The WM consists of the assertions that the system believes to be true of the current physical situation. This set of assertions is constantly changing and being updated as a new description of an observed action is input to the system. In Figure 1 the beginning of such an input stream is shown and referred to as Input Action Sequence. The PM contains assertions about what the system believes the actor believes, wants, and likes, etc.

The ES represents the plan structure that the system currently believes the actor is executing. This attribution of a plan to the actor requires that: (1) the incoming act sequence match the partial order of the plan structure; (2) the actor have certain beliefs and wants (the Person Model Requirements); and (3) that certain assertions about the physical world be true (the World Model Requirements). We will show that by defining the ES appropriately and using this knowledge decomposition, the ES can be used by the plan recognition process to compute the PM and WM requirements. Further, by matching the plan units of the ES to the incoming actions the schema variables of the ES can be bound and used to further customize and specialize the ES to the observational context. In this way the basic process of plan recognition is

reduced to various types of matching processes.

The requirements derived from the current ES are logical requirements that must be met if the actions are to be interpreted using that particular ES. Consequently, a mismatch signals the violation of an expectation and indicates that the ES must be revised. After presenting the operation of this basic matching process we will show how the results of this mismatch can be used to trigger and guide a strategy of ES revision. This hypothesize and revise strategy is congruent with our goal of using knowledge stated at a general level in plan recognition. The revision rules are anchored on classes of mismatches between ES requirements and the observations. Consequently, we have been able to avoid anchoring these revision rules to particular ES structures.

This hypothesize and revise strategy is further motivated by our assumption that human observers do not use a strategy of heuristic search to explore a large space of possible interpretations of a sequence of actions. Rather the observer is assumed to explore only a few, usually only one, hypotheses at a time. This hypothesis is adapted to the realities of the observations by a process of refinement and revision. This strategy does not insure correctness nor completeness. However, this is not our goal. Rather, we hope that this strategy will allow us to predict the conditions under which the human observer will succeed or fail in the task of plan recognition.

The Process of Interpretation

The process we define below takes a sequence of act descriptions and by re-representing them as plan units, produces a plan structure that is supported by and is consistent with the world model and the person model.

Hypothesize and Revise

The three main steps of the process are: (1) [SETUPES] An appropriate ES is set up and this ES is customized for use in the current context. The ES may be simply summoned from memory, constructed by a plan generator from knowing the goal of the planner, or may be postdictively assembled unit by unit from the incoming observations. (2) [ACTINPUT] The input act description is used to

update the world model and is then matched against an expected plan unit in the ES. A proper match produces bindings for the variables in the ES thus making the further expectations more specific. The details of this expectation guided matching process are reported in [10]. (3) [REVISE] Conflicts that arise between the expectations and observations trigger rules for hypothesis revision.

The structure of ES schemata

The ES consists of a number of plan units with specified "inorderto" relations among them which constitutes a partial order. The starting plan units and the final unit are identified. The goal of the final unit is also the goal assigned to the ES. The set of triples [P1 S P2] called the "inordertoprops" is formed from those plans P1 and P2 such that the goal proposition S of P1 is among the preconditions of P2. For this it should also be the case that no other plan unit occurring in between P1 and P2 undoes the proposition S. The "modelrequirements" are pairs [S P] with propositions S, which are opportunities for a plan unit P in the ES, such that there is no other plan Q set up to accomplish S. Each such proposition is required to be satisfied in the initial world model.

Monitoring these propositions aids the interpretation process in developing further expectations and in the recognition of conflicts and failed plans. As input actions are matched against expected plan units, they are shifted into the Hypothesis set H, to designate that they are supported by observations and are no longer "expected" to occur. This distinction between H and E is represented graphically in Figure 1 where those units in H are linked to input actions that have been taken up by the system. As a plan unit is shifted from the set E to the set H, other members of E are considered to be "next expected" plan units. A next expected member has its modelrequirements checked against the world model. If such a modelrequirement proposition is true in the person model and false in the world model, then this plan unit is flagged as "expected to fail". If a modelrequirement proposition is false in both the person model and world model, then a plan to accomplish the proposition is generated and inserted into the ES. If the modelrequirements

are satisfied in both these models, the inordertoprops are checked. If the proposition mentioned in an inordertoprops was in fact accomplished by an earlier plan but is no longer true then the assumption of "inorderto" between the two plans is not supportable.

The diagnosis and analysis of the currently attributed plan is performed by a decision tree [8] using information maintained in a structure called INTERPRETATION. Each terminal node of this decision tree calls for a corresponding revision rule. The revision rules perform actions that could range from accommodating the next observed action as being optional, or including it as part of the present plan, or generating and inserting plans to accomplish failed modelrequirements, or claiming that the planner failed in successfully carrying out his plan, or as drastic as abandoning the current assumptions and developing a different plan structure using whatever information can be meaningfully carried over from the present plan structure. We illustrate below the actions of the three components of the hypothesize-revise procedure.

Example of Basic Match Process and Conflict Resolution

An example expectation structure is presented below using KRL-like [1] conventions. The notation [refer: V] introduces a variable name and all subsequent uses of the variable designate the same entity. This ES is constructed by SETUPES from eight plan units that use variables to make references to objects and locations in the world model. The eight plan units form a connected partially ordered structure shown in Figure 1.

```
(a EXPECTATION STRUCTURE
Assemble/and/Eat with
[Variables
  (LIST A L1 L2 L3 O1 O2 O3 O4)]
[agent (a PERSON [refer: A])])
```

The actions embedded within the ES are listed below.

```
DESCRIPTION of Assemble/and/Eat
Expectation Structure:
  (a WALK with
    [toloc (a LOCATION [refer: L1])])
  (a TAKE with
    [object (a OBJECT [refer: O1]
              with [loc L1]
              which IsEdible)]
    [inordertoof (the last WALK)])
  (a CARRY with
```

```

[object O1]
[toloc (a LOCATION [refer: L3])]
[inordertoof (the last TAKE))]
(a WALK with
[toloc (a LOCATION [refer: L2])])
(a TAKE with
[object (a OBJECT [refer: O2]
with [loc L2]
which IsEdible
[ASSEMBLYPARTS O1 O2])])
[inordertoof (the last WALK))]
(a CARRY with
[object O2]
[toloc L3]
[inordertoof (the last TAKE)])
(a MAKE with
[components O1]
[components O2]
[instrument (a OBJECT [refer: O3]
with (loc L3))]
[assembly (a Object [refer: O4]
which IsEdible
(ASSEMBLYOF
O1 and O2 using O3))]
[inordertoof (the last CARRY)]
[inordertoof (the prev CARRY)])
(a EAT with
[object O4]
[inordertoof (the last MAKE)])

```

The Assemble/and/Eat assembles two objects O1 and O2 to produce another object O4 which is then eaten. The instrument used in making O4 is O3. The two sequences of Walk->Take->Carry satisfy the subgoal of the Make that require the components to be at the location of the Make.

The Design of an Expectation Structure

The types of information that go into defining the ES have been carefully chosen and it is worthwhile to consider here the reasons for the choices made. (1) All references to objects in the world model are through variables, so that the ES is not specific to the particular situation but is generally applicable to a variety of situations. For example, O4 is not bound, and could be a peanut butter sandwich, an ice cream cone, or anything else. Similarly, there is no identification of the location L3 to which O1 will be carried. If the person goes over to the cabinet and picks up an inedible object, there is sufficient information in the ES to know that this ES is no longer supportable in its present form. This information is reflected in the predicates [Assemblyparts ..] [IsEdible] and [Assemblyof ..] (2) The ES reflects no explicit world knowledge and thus no assumptions about the edibility or desirability of eating

ice-cream are made in setting up the ES. (3) Knowledge that ice-cream will melt if left on the counter, that liquids need containers to carry, and that a cabinet door must be open to access its contents are factored out and do not even implicitly influence the structure of the plan in the ES. Thus, although a door might have to be opened to Take O2, and a container might be needed to Carry O2, these context dependent aspects of the plan are left out of the ES and the goal-subgoal structure is described in a context independent fashion. (4) There are only implicit alternatives and optional plans included in the ES. None of these alternatives and options are explicitly given in the structure. A dish might substitute for cone in eating ice-cream. If the scoop is not already on the counter the person might have to fetch one. The person might optionally sit down before eating. Such alternatives and options, including ways to take care of interruptions and accidents are factored out into the revision rules. This allows a single sketchy ES to be used and adapted to a number of action sequences. (5) Similarly, no explicit assumptions about the planner, his belief state, knowledge state or motivations are introduced into the ES. The statement of goals and inorderto relations certainly have implications about what the planner wants and believes but these will be derived and accounted for as the interpretation proceeds by matching actions against the plan units in the ES.

The ES is set up to reflect only the causal and intentional structure of the plan. The objects entering the plan are defined by their role in the plan. The ES leaves out as much world knowledge, person knowledge, and listing of alternatives and options as possible. This factoring is done to achieve generality in the structure of the plans used and also because it is impossible to anticipate all revisions, options, and world knowledge contexts in which a general kind of goal may be pursued.

Setting up and customizing the ES

The SETUPES component customizes the ES to the context of a given initial World Model, WM, and an initial Person Model, PM, which it compares with ES. This gives rise to an initial set of bindings and the calculation of the next expected action.

Initial World Model WM:

(Tom loc Kitchen) (Cabinet in Kitchen)
(Cone loc Cabinet)
(PeanutButter loc Cabinet)
(Refrigerator in Kitchen)
(Bread loc Refrigerator)
(Refrigerator in Kitchen)
Not-(IceCream loc Refrigerator)
(Counter in Kitchen)
(Scoop loc Counter)
accompanied with the general world knowledge:
(Assemblyparts IceCream Cone)
(Assemblyparts Bread PeanutButter)
(Assemblyof IceCream Cone Scoop ICC)
(Assemblyof Bread PeanutButter Knife PBS)
(IceCream IsEdible) (Cone IsEdible)
Not-(Knife IsEdible) etc.

Initial Person Model PM:

(Tom believes (Scoop loc Counter))
(Tom believes (IceCream loc Refrigerator))
(Tom believes (Cone loc Cabinet))
and so on, conforming to the model WM.
(Tom likes (Tom ate ICC))
(Tom likes (Tom ate PBS))

In setting up the ES the modelrequirement propositions are matched against the Person Model PM and the World Model WM. Each proposition falls into one of three classes with respect to each model: (1) It is known to be true in the model; the requirement is satisfied. (2) It is unknown in the model; it will be presumed true until explicitly denied. (3) It is known to be false in the model. If the proposition is false in the person model, that is, if the planner has a belief that contradicts one of the requirements then the attribution of the plan structure to the planner cannot be made and the ES will have to be revoked. On the other hand, if it is the world model that contradicts the proposition while the person model supports it, the divergence between the two models allows the plan but requires that the proposition be flagged "expected to fail during execution".

Modeling and Matching Actions

We shall follow the basic matching process, especially the actions of ACTINPUT, for an episode whose first four acts are:

1. Tom walks to the cabinet. (Mwalk1)
2. Tom takes out a cone. (Mtakel)
3. Tom carries the cone to the counter. (Mcarryl)
4. Tom walks to the Refrigerator. (Mwalk2)

The first act, Mwalk1, is taken in

and the updated WM now shows the proposition (John loc Cabinet). Mwalk1 is then matched to Walk1 from the ES yielding the following bindings:
Walk1->(Mwalk1), A->(Tom),
L1->(Cabinet),
O1->(PeanutButter, Cone, ..),
L2->(Cabinet, Counter, Refrigerator, ..)
and so on.

The matching process pursues unique bindings such as L1->(Cabinet) deeply to make sure that what is required about L1 by the ES is in fact matched by the properties and relations of Cabinet. Variables such as O3 and L2 have a set of candidate bindings and the procedure does not generate and analyze the various combinations of assignments that are possible. However, as soon as the actor takes out the cone in the second act, Mtakel, the binding of O1->(Cone) causes an investigation of what objects O2 can be assembled with the binding for O1. The specifications in the world knowledge constrains O2 to have the unique binding O2->(IceCream), which then yields the unique binding L2->(Refrigerator) and also O3->(Scoop) and O4->(Counter) since the scoop is on the counter. Thus after the second act all the variables are bound uniquely:
A->(Tom), L1->(Cabinet),
L2->(Refrigerator), L3->(Counter),
O1->(Cone), O2->(IceCream), O3->(Scoop) and O4->(ICC).

There is a modelrequirement (O2 loc L2) associated with the plan unit Take2 and this proposition is now tested since O2 is uniquely bound. This process of verification recognizes that it is false in the world model and thus flags Take2 as being "expected to fail". Thus later on at the time of revision either the goal of the ES is considered abandoned, or Take2 is replaced with another plan to accomplish its goal, or a plan is inserted to accomplish the proposition (O2 loc L2) or O2 is bound to some other object which will still allow the final goal to be pursued. At this point in the processing none of these alternatives are enumerated. Instead a "wait and see" strategy is followed and the next input act is taken up.

Generation and explanation of conflict

After the four acts shown above are input and matched the matching generates the following bindings:
Walk1->(Mwalk1), Takel->(Mtakel),
Carry1->(Mcarryl), Walk2->(Mwalk2),
A->(Tom), O1->(Cone), O2->(IceCream),
O3->(Scoop), O4->(ICC), L1->(Cabinet),

L2->(Refrigerator), L3->(Counter).
The next expected unit is Take2, which translated using the current bindings is "Tom takes out the ice cream."

The next input act, call it Xwalk, is "Tom walked to the counter," and this observation requires a revision of the current interpretation. Since Xwalk does not correspond to the expected next action, Take2, the system recognizes a violation of expectation. The report of the mismatch has two parts: (1) Xwalk does not match the next expected unit of ES; and (2) In the inordertoprop [Walk2 (A loc L2) Take2], the Walk2 is already matched to an action, (A loc L2) is now false, thus signalling a violation of the alleged [Walk2 inorderto Take2] relation.

The observed action is to be explained either as an attempt to satisfy the goal of Take2 by a different sequence of actions (Tom is doing something to get ice cream) or to treat it as part of a sequence reversing the consequences after failing in the plan (Tom is putting away the cone). There is also the option that the actor may be attempting to reach his original goal (to eat some object, not necessarily ice cream cone). The maintenance of such explanations at different levels of generality is aided by the generality with which the ES is designed. When alternative revisions are forthcoming, the revisions will have to await further confirmation from subsequent observations. Again, as in the matching of input acts, a "wait and see" strategy that delays the analysis of alternatives is used in the case of ES revision since subsequent inputs will provide guidance in selecting an alternative.

Conclusion

The principal tenet of the hypothesize and revise paradigm is that when the hypothesis revision rules are robust enough to set the process back on the right track and to fill missing details whenever appropriate, there is no pressure to set up the "correct" or the most detailed expectation at the start. It is characteristic of most recognition problems, including vision, speech and in our case plan recognition, that there is little information at the start to set up strong expectations. Yet, generating hypotheses and conducting "best first" or backtracking search demands strong

expectations in the beginning and lays heavy penalties of effort on the process when the expectations are either partially inappropriate or when they lack adequate detail. Further investigation of the robustness of the hypothesize and revise procedure and further characterizations of revision rules will be the focus of our continuing research.

Status of implementations

The representational structures described above, the consistency checking, action modeling have been in operation for about a year now. The basic match process, conflict analysis and reporting was working in February 1977. The case analysis and interpretation revision is being programmed and is expected to be working in the summer and will be demonstrated at the Third Annual Workshop on AI in Medicine and the Sciences, to be held at Rutgers University in July 1977. BELIEVER is developed in the AIMDS system, which is a smaller version of MDS customized to the task of reasoning about actions. AIMDS has taken from the MDS formalism the concepts of Templates, Consistency Conditions and Residues and components of the MDS architecture [12] providing a framework for representing knowledge in terms of frames and transition frames.

This research was supported by Grant RR-643 to the Rutgers Research Resource on Computers in Biomedicine from the BRP, Division of Research Resources, NIH.

References

1. Bobrow, D. and Winograd, T. An Overview of KRL, A Knowledge Representation Language. Cognitive Science, vol 1, January 1977, p3-46.
2. Sacerdoti, E.A. A Structure for Plans and Behavior. Ph.D. Thesis, Computer Science Department, Stanford University, 1975.
3. Schank, R.C. The structure of episodes in memory. In D.G. Bobrow and A. Collins (Eds.), Representation and Understanding. New York: Academic Press, 1975. p237-272.
4. Schmidt, C.F. and D'Addamio, J. A model of the common-sense theory of intention and personal causation.

5. Schmidt, C.F. Understanding human action: Recognizing the plans and motives of other persons. In J. Carroll & J. Payne (Eds.), Cognition and Social Behavior. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1976. p47-67.
6. Schmidt, C.F., Sridharan, N.S., & Goodson, J.L. Recognizing plans and summarizing actions. Proceedings of the Conference on Artificial Intelligence and the Simulation of Behavior. Edinburgh, Scotland: University of Edinburgh, 1976. p291-306.
7. Schmidt, C.F. & Goodson, J.L. Recalling and Summarizing Action Sequences: Some Empirical Tests of the BELIEVER Thoery. Technical Report CBM-TR-81, Department of Computer Science, Rutgers University, June, 1977.
8. Schmidt, C.F. and Sridharan, N.S. The Plan Recognition Problem: A Hypothesize and Revise Paradigm. Report CBM-TR-77, Department of Computer Science, Rutgers University, March 1977.

9. Soloway, E.M. and Riseman, E.M. Knowledge-directed Learning Using Multiple Levels of Pattern Description. In Fifth International Joint Conference on Artificial Intelligence. Cambridge, Mass., 1977.
10. Sridharan, N.S. and Schmidt, C.F. Knowledge directed inference in BELIEVER. In D. Waterman and F. Hayes-Roth (Eds.), Pattern-directed Inference Systems. New York: Academic Press, May 1977 (to appear). Available as report CBM-TR-75, Department of Computer Science, Rutgers University.
11. Sridharan, N.S. A Frame-based system for reasoning about actions. Technical Report CBM-TM-56, Department of Computer Science, Rutgers University, July 1976.
12. Srinivasan, C.V. The Architecture of Coherent Information System: A General Problem Solving System, IEEE Transactions on Computers, April 1976, p390-402. Originally appeared in 3rd IJCAI, Stanford, 1973.

