Evidence-Based Plan Recognition*

Norman Carver

COINS Technical Report 88-13
February 1988

Abstract

Plan recognition is a complex and uncertain task. Because of this, general purpose plan recognition systems must be equiped to deal with uncertainty. In this paper, we first examine relevant issues in control and evidential reasoning and then survey previous plan reconition research. Previous approaches to plan recognition are inadequate for use as general-purpose plan recognition systems and we identify those properties which such a system must possess. Finally, we present a new approach to plan recognition which addresses the critical issues. Plan recognition is viewed as a process of *gathering evidence* to *manage uncertainty*. This provides a framework within which to evaluate alternative interpretations and apply expert-level heuristic control knowledge. Key to the system are the control component which develops control plans under the supervision of heuristic focusing knowledge and the evidential representation component which includes information about the sources of uncertainty in the evidence. Throughout the work, heavy emphasis is placed on having the ability to explicitly consider control decisions. These decisions not only determine which interpretations to pursue, but also the best methods for gathering evidence to resolve the uncertainty.

*This work was supported by the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York 13441-5700, and the Air Force Office of Scientific Research, Bolling AFB, DC 20332 under Contract No. F30602-85-C-0008. This contract supports the Northeast Artificial Intelligence Consortium (NAIC).

Contents

Chapter 1

Introduction

Problems which require the application of artificial intelligence techniques are distinguished by their reliance on incomplete and uncertain knowledge. Plan recognition, the interpretation of data in terms of instances of particular plans, is just such a problem. Combinatorial considerations generally make it impossible to enumerate all of the possible alternative interpretations of the data. Of those interpretations which are considered, evaluation of the relative likelihood of the alternatives is complex and uncertain. In many domains the data may itself be incomplete and/or incorrect.

Previous approaches to plan recognition fail to address many of the issues necessary to produce practical systems. In particular, they provide very limited methods for reasoning about control decisions and for evaluating the appropriateness of alternative interpretations. Because of these concerns, we developed a focus-of-attention scheme for the POISE project [6] which used heuristic knowledge to make the decisions about which hypotheses to pursue. A major advantage of this system was the fact that it provided an explicit representation of the (uncertain) assumptions upon which the control decisions were based. This meant that when interpretation errors were discovered, the system could backtrack, examine earlier decisions, reason about why they were made, and decide how to revise them. While the explicit representation of this control information improved the control process, the approach still suffers from a number of shortcomings. The most important problems include: lack of flexibility for specifying heuristic control knowledge, confusion of belief in an alternative with the decision to pursue the alternative, and little guidance during the revision process.

This paper will develop a new approach to plan recognition which will address the deficiencies of existing systems. The key to this approach is to view plan recognition as a process of *gathering evidence* to *manage uncertainty*. Viewing plan recognition in this way provides a framework within which to apply expert-level heuristic control knowledge and evaluate alternative interpretations. Data is considered as a source of evidence for the plan hypotheses: when data can be interpreted as part of a hypothesis it provides evidence for that hypothesis. Evidential links are maintained between data and hypotheses the data supports. This provides the system with an explicit representation of the reasons to believe the interpretation hypotheses. The use of an explicit, symbolic representation of evidence is important because it makes it possible to explicitly reason about control decisions. Knowing what evidence supports hypotheses, we can understand the *sources of uncertainty* in the evidence and decide how best to resolve them. When evidence is summarized in numeric degrees of belief, access to this sort of knowledge is lost.

We take a broad view of the class of plan recognition problems. A range of interpretation and situation assessment problems can be viewed as plan recognition problems. The prototypical example is the interpretation of a series of actions as part of some overall task. This ability is relevant to natural language understanding and computerized intelligent assistants. Vehicle monitoring and related situation assessment problems may also be treated as plan recognition problems. Here, the "plans" represent vehicle movements or missions and are composed of characteristic sequences of sensor data rather than "actions." In all of these interpretation problems the goal is to form a higher-level, more abstract view of the data. In other words, to provide an appropriate context within which to understand the data.

Plans specify the hierarchical relations between the data and more abstract views of this data. Although the form and specification of plans varies with the application domain, plans are composed of sets or sequences of subplans. For example, a plan for processing a form is composed of steps for filling the form out and then sending it on to the appropriate office. In the case of vehicle monitoring, a vehicle plan might be composed of subplans which represent sets or sequences of radio and radar emissions which identify the vehicle and its purpose. Plan recognition then, involves the interpretation of sets of subplan instances as (perhaps partial) instances of more abstract plans.

Plan recognition is a complex and uncertain process:

- In general, there are multiple, ambiguous interpretations for each subplan or sequence of subplans.
- Ambiguity is compounded in domains which admit multiple, concurrent plans since the subplans may be interleaved.
- For some applications, interpretation must be done in real-time, relying on preliminary and partial plan data to make "best guesses" about the complete plans.
- Since computational considerations generally preclude constructing all the possible interpretations, there is actually uncertainty whether any of the system's interpretation hypotheses are correct.

- The volume of data may be so massive as to preclude complete examination.
- Data may also be missing, uncertain, and/or incorrect.

As a result of these factors, plan recognition systems must be designed to deal with many uncertainties. We feel that intelligent plan recognition systems must be able to:

- Evaluate the level of belief and uncertainty in alternative interpretations.
- Understand the reasons for beliefs.
- Encode and apply heuristic control knowledge.
- Revise interpretation hypotheses as information accumulates.
- Handle uncertain and incorrect data.
- Integrate data from multiple sources.
- Actively control data accumulation.
- Reflect system goals in control decisions.

Since there will generally be a number of alternative interpretations of any set of data, it is crucial to have some method for evaluating the relative merits of the alternatives. Evaluation has often been accomplished as an implicit part of the control scheme. This is undesirable because it limits potential control strategies to pursuing only the most believed alternatives. Combinatorial and real-time considerations make focus-of-attention strategies crucial. Expert-level heuristic control knowledge can be used if a proper framework is available for encoding it and applying it. However, since control knowledge is fallible and since data may be missing or in error, interpretations should be able to be revised as data is incrementally accumulated.

The final three requirements represent extensions to the normal notion of plan recognition, but have broad applicability nonetheless. In most domains, there are several sources of knowledge which a human expert would use to support his interpretations. Plan recognition systems have typically failed to make use of multiple sources of evidence despite its advantages in dealing with uncertain, incomplete, or incorrect data. For example, in aircraft monitoring applications there would be data from several types of sensors as well as information about terrain and air defenses. Active control could be used to greatly reduce processing effort and system uncertainty when possible. In an aircraft monitoring system some sensors may automatically produce data while others may be controlled by the interpretation system. In an intelligent assistant, the system may prefer to query the user rather than waiting for additional data to resolve uncertainties. The goals the interpretation process in a domain need not always be the same. An aircraft monitoring system may be trying to protect a sensitive installation or may simply be trying to monitor all air traffic. It may be under time constraints or it may not. The system should adjust its operation to best meet the specific goals–monitoring all aircraft or only the potentially hostile ones, for instance.

We believe that the plan recognition requirements outlined above can be met by viewing plan recognition as a process of *gathering evidence* to *manage uncertainty*. The key characteristics of the approach are:

- Plan, subplan relations are treated as uncertain, evidential relations.
- Evidence and sources of uncertainty are explicitly represented.
- Heuristic control decisions are based on the sources of uncertainty in the hypotheses and the need to manage uncertainty.

Treating plan, subplan relations as evidential relations rather than as absolute goal, subgoal relations means viewing these relations as uncertain inferences. This approach helps to address several of the limitations of existing plan recognition systems. An evidential reasoning system can now be used to provide a representation of the evidence for the alternative hypotheses. This allows their relative likelihoods to be evaluated independent of the control decisions. Reasoning about control decisions can be easily extended to all stages and levels of the interpretation process because the abstraction of any hypothesis is simply an inference. In particular, we need not wait for the construction of top-level plan hypotheses before applying focusing knowledge. Incomplete, uncertain, and incorrect data are naturally accommodated as they simply result in additional sources of uncertainty which can be resolved by gathering sufficient additional evidence. Different types of data can also be easily accommodated because they simply represent different sources of evidence for the interpretations. Revision is a natural part of the accumulation of evidence as conflicting data produces uncertainties which the system can represent and resolve.

By explicit, symbolic representations for evidence, we simply mean that we maintain explicit links between hypotheses and the reasons we believe the hypotheses. Sources of evidence include subplan hypotheses and knowledge such as terrain and weather information. Access to detailed information about the evidence makes it possible for us to make use of an important body of expert-level knowledge about the task: the *sources of uncertainty* in the evidence. In plan recognition, evidence is rarely conclusive. The sources of uncertainty represent the reasons why evidence may fail to support a particular conclusion. For example, acoustic sensor data may fail to support a vehicle because it is actually the result of a sensor malfunction or sensor ghosting. The control component can now reason about the best course of action for the interpretation system to take because it understands the purpose of its actions: to try to resolve the sources of uncertainty in the hypotheses. An independent, explicit representation of the evidence also makes it possible to represent the relations between the hypotheses. Thus, though direct evidence for a hypothesis may not be available, there may be sources of evidence for related hypotheses–like alternatives.

Active control of data accumulation is possible since the control process explicitly considers the sources of uncertainty in an interpretation and can direct the action of data sources to produce evidence to resolve the uncertainty. Of course, the amount of control the interpretation system can exercise over the evidence it has available will depend on the domain. For example, in vehicle monitoring applications, the operation of a radar sensor can be tuned in order best resolve uncertainty in a particular aircraft ID or location. As system goals change, the importance of different uncertainties changes. When evidential support is summarized numerically it is impossible to consider the context in making control decisions. Maintaining an explicit representation of evidence makes it possible to accommodate varying system goals because beliefs and decisions can be made sensitive to the goal context. The criticality of the uncertainties can be judged in relation to the current state and purpose of the system.

Chapters 2 through 4 motivate this work by examining existing approaches to control, evidence, and plan recognition. Chapter 2 is an introduction to control issues as they relate to plan recognition. An overview of existing approaches to representing and using evidence is contained in chapter 3. Chapter 4 presents the plan recognition and related interpretation problems which motivate this research. POISE, the Distributed Vehicle Monitoring Testbed, and several other plan recognition systems are examined. In chapter 5 we present our preliminary view of how evidence and uncertainty should be used in plan recognition systems. Finally, chapter 6 summarizes the goals of this research.

Chapter 2

Control

One of the major problems facing any AI system is the control problem: what should the system do next? An AI system can be viewed as proceeding through a sequence of states as it runs and, in general, there will be several possible actions which could be taken in each of these states. This results in a sequence of choice points for the control component. For example, the control problem for several AI paradigms includes:

Search —which path to pursue.

Plan Recognition —which interpretations to pursue (focus-of-attention).

Production Systems —which satisfied production to "fire" (conflict resolution).

Problem-Solving —which subgoal to work on next.

An important characteristic of AI problem domains is the uncertain and incomplete nature of the knowledge available for problem solving. Thus, control decisions in AI systems are uncertain because the systems will not have complete, correct information which would allow them to choose the right action to take at each decision point. The control component will have to decide the "best" action to take at each decision point based on inexact, heuristic knowledge. Even if more "exact" decision procedures are available for AI problems, the data and knowledge upon which these decisions would be based will be uncertain, incomplete, and/or incorrect: models of the state of the world, models of the state of the problem solving, applicability of operators, etc.

Because of the uncertainty inherent in AI control decisions, problem solving systems must be designed to cope with uncertainty and the resulting incorrect decisions. A number of different approaches to managing this uncertainty have been used in AI systems: using domain-dependent heuristic knowledge to guide the decision process, pursuing multiple alternatives (delaying decisions), backtracking to revise decisions when inconsistencies develop, and opportunistic control with likelihood measures, etc. The appropriate approach depends on the characteristics of the problem-solver and of the application domain.

Regardless of the approach taken to managing uncertainty, as problem solving proceeds, additional knowledge is accumulated which can be used to reduce the uncertainty in the control decisions. Unfortunately, traditional AI systems have suffered from what Doyle has termed the "fatal flaw" of "inaccessibility of control information" [18] due to the implicit nature of most reasoning. This leads to problems which Doyle labels the "inexpressibility of control information" and the "inexplicability of actions and attitudes." Because programs are unable to reason explicitly about why they should or should not take particular actions, it is impossible to encode and maintain sophisticated heuristic control knowledge. Likewise, it is impossible for programs to reason about how to revise their decisions in the face of new evidence since they cannot understand why they did or did not take particular actions.

In the next two sections we will discuss control in AI problem-solvers in terms of two distinct processes: the process of making (the initial) control decisions and the process of revising control decisions. The use of meta-rules and dependency-directed backtracking will be presented as examples of intelligent approaches to control and revision. These techniques have had a strong influence on this work so it is instructive to examine their limitations.

2.1 Control Decisions

Control decisions in AI systems have typically been made in a two stage process. Davis [13] terms these two stages the "retrieval" stage and the "refinement" stage. The retrieval stage determines the actions which may plausibly be taken given the current state of problem-solving. Typically, this stage is implemented using some kind of knowledge indexing scheme appropriate to the characteristics of the problem. A number of different retrieval/indexing paradigms have been used in AI problem-solvers: data-directed, goal-directed, difference-directed, etc. In general, however, such retrieval strategies do not produce a single permissible action at each choice point. That is, indexing schemes alone cannot be used to select the single, correct action to take due to the uncertain and incomplete nature of knowledge in AI systems. Some additional decision procedure must be applied to select the single action to be taken. This is the purpose of the refinement stage–also known as the conflict resolution stage in a production system.

AI systems have often used simple, static refinement procedures. For example, conflict resolution schemes in production systems have selected the "first" rule satisfied, the rule using data most recently added to the database, or the most "specific" rule. A more sophisticated approach to refinement involves the use of numeric ratings or certainty/likelihood factors. In these schemes, ratings or weights are computed for each of the alternatives from the retrieval stage and the most highly rated alternative is chosen. Ratings can be based on various attributes of the relevant alternatives and have included subjective strength of belief, utility, salience, and a priori probabilities. Numeric approaches provide a more dynamic control than is possible with static approaches since a number of characteristics of the particular relevant data can be considered.

Neither static nor numeric approaches are suitable for use with a truly intelligent control system, however, because they lack explicit knowledge about their decisions. In particular, the intelligence that can be applied to the revision of decisions is severely limited by the implicit nature of the decision process. Static refinement approaches do not explicitly consider any characteristics of the alternatives–their choices are based on fixed, implicit selection criteria. In this situation, it is difficult to see how any revision process could perform better than a blind search. For example, suppose a decision based on the "most recent data" criteria proves to be incorrect. Presumably this decision criteria has been used due to certain assumptions about the nature of the problem solver and/or domain. Since these assumptions are not explicitly considered in the refinement process, they cannot be considered during the revision process. It is impossible now to reason about how to proceed. Should the alternative using the next most recently created data be pursued–is the basis of the decision still sound and applicable? Or, does the failure of the decision indicate that the criteria is invalid? Perhaps it is then appropriate to pursue the alternative based on the oldest data or, perhaps, the age of the data has absolutely no relevance to the correctness of the decision. Without more explicit knowledge about the decision criteria it is impossible to say.

Numeric refinement approaches are not substantially better than static approaches with respect to intelligent control and revision. Their "reasoning" is implicit in their rating calculations and so is unavailable for introspection. Suppose, for example, that ratings are based on the "quality" of the data and the "quality" of the inference rule to which the data is applied. It is impossible to distinguish between an alternative rated highly due to good data and a mediocre inference rule and one which is rated highly due to mediocre data and a good inference rule. All a revision process has to work with is the final numeric rating which condenses the relevant factors–it cannot consider them separately. This makes it impossible to assign blame for failures and take this knowledge into consideration by examining how alternatives are related to the failure. Should inferences based on the same data be tried next because the inference rule is likely the cause of the failure or should this same data be avoided because it was likely the cause of the failure?

Davis [13] and others have advocated viewing the process of making control decisions as a problem solving task in itself. A body of meta-knowledge including heuristic information about the domain and the (object-level) actions would be used to guide the refinement process. Such a refinement process can "reason" about the control decisions by considering a number of different control criteria in the decision process. For a production system, the addition of meta-level knowledge might take the form of a set of meta-rules with a meta-rule interpreter. The meta-rules would be applied during the conflict-resolution stage of the production system to select a single object-level rule to be fired. For example, a meta-rule from [13] for the domain of stock analysis is:

```
IF the LEI index has been climbing steadily for the past several months
   AND there are rules which mention recession in their premise
THEN it is likely (.7) that these rules will not be useful.
```

While meta-level knowledge embodied in the form of meta-rules allows a system to explicitly consider various characteristics of potential actions, it is clear that many of the control problems we have discussed are still present. In particular, when several meta-rules may be applicable, their advice is typically combined by resorting to numeric ratings. Now, however, we are right back where we started since the object-level factors explicitly considered by the meta-rules are lost in the conversion to numeric ratings. About all that has been gained is a more modular representation for the numeric ratings calculations. In particular, this approach does not provide enough depth or structure to the meta-knowledge and the way it is applied to allow the integration of distinct, but relevant information. Davis [13] recognizes these problems and states that there may be "situations where the rationale behind an argument is as important a factor as its strength." For example, the meta-rule given above limits the use of certain (object-level) stock analysis rules in a particular context. Implicit in this meta-rule is the assumption that a recession is unlikely under the specified conditions and the knowledge that a great deal of processing will be required to recognize this fact. Meta-rules which give conflicting orderings for decisions may be doing so based upon conflicting assumptions about the occurrence of a recession. If the recession assumptions were made explicit, it would be possible to use additional evidence about the likelihood of a recession to resolve the conflicting rankings. Such knowledge could be available either from "external" sources (the user, a backtracking process, or meta-meta-rules) or from other meta-rules which assume that a recession is or is not likely. Even if it were not available, the conflicting rankings could be recognized as alternatives to be resolved as additional evidence is accumulated by the system. This information could then be used to focus and control the system.

2.2 Revising Control Decisions

The uncertain nature of control knowledge means that systems will make errors in their control decisions. When a contradiction occurs or a dead end is reached during problem solving, it is necessary to revise some control decision(s). In general, this requires determining which decisions were responsible for the difficulty, retracting these decisions and their consequences, and making new decisions. However, retraction can be handled in different ways depending on the characteristics of the domain and the characteristics of the control scheme.

Actual retraction of an action may not be required because the application of an incorrect action does not preclude a successful subsequent search for the solution. Systems that exhibit this characteristic are known as commutative. For example, the application of a "wrong" theorem in a theorem-proving system simply results in the derivation of facts which are useless with respect to the goal of proving the desired fact. The derived facts are still true statements and will not prevent the derivation of the desired result by the application of the "correct" theorems. Nonetheless, a "revision" process is still required to determine which results are useless in order to "prune" the search space and avoid combinatorial explosion in the search, e.g., uncontrolled antecedent deductions in a theorem-proving system. This points up the fact that though answer generation is commutative in such systems, control of problem solving is not due to resource limitations [37]. In other cases, retraction of actions might not be required because multiple paths were pursued by the problem solver, e.g., exhaustive, breadth-first, or beam searches. Again, a revision procedure is required in order to assign blame for the failure and prune the search space in an appropriate manner. These approaches have limited applicability, however. Exhaustive searches are seldom practical for AI problems due to combinatorial explosion in search paths or cost of searching incorrect paths. Partial searches can only avoid retraction if they can guarantee that they cover the correct solution–which can be difficult.

With *chronological backtracking* the state/context of the system is switched to that existing prior to the application of the decision being retracted. This retracts the decision and its consequences. Chronological backtracking is normally implemented so that a contradiction causes the most recent control decision to be retracted. If the temporal order of decisions is not of primary importance, however, the result is a blind, depth-first search for the relevant decision(s). This is exceedingly inefficient because decisions which are not responsible for the inconsistency are unnecessarily withdrawn and reconsidered. In addition to problems with exponential search complexity, much valuable information is discarded in the context switches. This includes information about the inconsistency which led to the retraction and about the paths which have already been explored. In certain domains, the best that can be done is to retract all actions back to the point of the incorrect action–i.e., each action depends on the previous action. Even in these cases, information about the reason for retraction and the paths which had been explored can be useful for determining which paths to explore next and how explore them.

Nonchronological backtracking is one response to the inefficiencies of chronological backtracking. In its purest form, nonchronological backtracking involves examining all decisions, identifying the source of the inconsistency, and choosing an alternative. Thus, only those decisions which could be responsible for the problem need be withdrawn and reconsidered and only the affected aspects of the state/context need to be retracted. This greatly reduces the search space, although complexity is still exponential in the number of *relevant* decisions.

The best understood implementations of nonchronological backtracking involve a technique known as *dependency-directed backtracking*. In order to make use of this technique, dependency records are used to link conclusions with their antecedents. This results in a dependency network which is stored and maintained by a *reason maintenance system* or *RMS* [17,42]. An RMS is a domain independent, syntactic database subsystem for representing propositional deductions and maintaining their consistency. Statement justifications, in the form of dependency links, can be traced back from the inconsistent statements to locate the set of statements upon which the inconsistent statements depend. Certain statements are usually considered to be *premises* or *assumptions*. The RMS is able to change its belief in these statements in order to affect belief in deduced statements and eliminate the inconsistency.

Dependency-directed backtracking is an implementation of the concept of nonchronological backtracking, but the two terms are often confused and used interchangeably [44,52]. This results in nonchronological backtracking appearing to be better understood than it is and dependency-directed backtracking appearing more general than it is. Winston [52] even supplies an "algorithm" for nonchronological backtracking which is similar in style to that provided for chronological backtracking. Unfortunately, unlike the chronological backtracking algorithm, the nonchronological backtracking "algorithm" cannot be implemented. In particular, it does not explain how relevant decisions are to be located, how to go about revising the relevant decisions once located, nor how to proceed with problem solving once the inconsistency is eliminated.

The class of problems to which dependency-directed backtracking can be applied is limited. It is best suited to problems where an explicit constraint network can be constructed such as constraint satisfaction problems and theorem-proving problems. From the point of view of control, the major conceptual failing of dependency-directed backtracking is that it separates control into two disjoint subsystems. The conventional system control component, or problem solver, is here only responsible for the initial control decisions. Revision is handled by the database through the dependency-directed backtracking routines. Dividing the problem solving responsibilities makes it difficult to coordinate control. This leads to a number of problems which will be discussed below.

Because reason maintenance systems only represent *propositional* deduction, instantiation of facts becomes a major control issue. If dependency-directed backtracking is to automatically and independently revise assumptions, the normal control process must effectively pursue all possible alternatives in order to be able to explicitly instantiate them in the dependency network. Doyle [17] recognizes that this is impossible in domains which involve many alternatives (or domains where alternatives are expensive to compute–see chapter 4), but his proposed solution involves an external process which must somehow be coordinated with the backtracking process. This coordination is difficult to achieve because separating responsibilities for control separates decisions from decision points and decision processes. It is unclear, then, how to re-examine a "decision point" and restart the decision process because the decision point is not explicitly represented in the dependency network in connection with the decision alternatives. This is the reason for what deKleer [16] calls the unouting problem–it is difficult to pursue previously abandoned problem solving paths because the states of the problem solver are not represented in the RMS.

Because dependency-directed backtracking is part of a domain independent subsystem, it is a purely syntactic process, unable to reason about the semantics of a situation when revising decisions. The normal control component of the problem solver must precompute all control information which might be required by the backtracking process and store it within the dependency network. Nonmonotonic dependencies are used to encode sets or sequences of alternative assumptions [17]. These predetermined sequences are then used to revise assumptions without regard to the semantics of the inconsistency. Recording control information in the dependency network means that dependency network dependencies not only record logical deductive relationships, i.e., reasons for believing some facts based on belief in certain other facts, but also control choices. This confuses the role that nonmonotonic justifications play in the nonmonotonic logic notion of default reasoning. Also, the use of little or no intelligence in the revision process severely limits the value of dependency-directed backtracking in real world problem domains. deKleer [16] reports that for qualitative reasoning, RMSs are very inefficient. The reasoner spends most of its processing time backtracking because it must still perform an exhaustive search on the relevant assumptions and revision of the database for each possibility takes a substantial amount of time.

RMS dependencies are typically used to represent only logical deductions, but logical deductions are not the only relations between beliefs and facts which need to be represented. Non-logical inferences, evidential relations, causal connections, etc., will be needed in order to locate relevant assumptions and to reason about their consequences during the revision process. Of course, relations other than logical deduction can be recorded in an RMS by simply including a node to represent the relation among the justifications of a conclusion. However, the semantics of such relations are not understood by the backtracking process and so cannot be used to guide revision. Only one form of non-logical inference, nonmonotonic or default inference [17], can be represented as an integral part of the RMS/dependency-directed backtracking formalism.

Even when explicit dependencies link decisions it is not always obvious which decisions are "relevant" to eliminating the contradiction. In the planning example in [52], decisions are not independent because of interacting constraints on system resources. When constraints are violated, it is easy to follow dependencies to locate those decisions which are (directly) relevant to the contradiction because of their use of the violated resource. However, these "relevant" decisions are not independent of other decisions with which they share resources. To develop a completely consistent plan may require withdrawing certain of these "irrelevant," but related decisions (a problem which is ignored in the text). How is this to be done without resorting to exhaustive, exponential search?

Finally, it must be noted that the recognition of a "contradiction" or a "dead-end" might involve major problem solving activity of its own [30]. The constraint satisfaction problems to which dependency-directed backtracking has been applied have tended to gloss over the difficulties involved in detecting contradictions and attributing reasons for the contradictions. A measure of developing uncertainty could play an important role in forewarning of these "contradictions."

Chapter 3

Evidential Reasoning

One approach to problem solving in uncertain domains is to apply evidential reasoning techniques. In this chapter we will survey techniques which have been used to represent evidence and belief. The purpose of the chapter is not to examine these approaches in detail, but rather to provide a review of their strengths and weaknesses as they relate to plan recognition. The presentation is divided between those techniques which use numeric representations of evidence and those which use symbolic representations.

Evidential reasoning has typically been applied to AI problems using the parallel certainty inference approach. Inferences are made using two fairly independent processes: conclusions are first derived as if they result from deductive (certain) inferences and then the degree of belief in the conclusion is computed. Computing the degree of belief in the conclusion requires the use of two *combining functions. Propagation* combining functions adjust the belief in a conclusion to reflect the belief in the premise and the characteristics of the deduction. *Pooling* combining functions determine the belief in a conclusion which is deduced by multiple independent inferences.

AI systems using evidential reasoning techniques have most commonly involved classification or diagnosis problems. In these problems, evidence is being accumulated to select the most likely hypothesis out of a *fixed set of alternatives*. The fact that all three of the numeric techniques discussed below rely on the existence of a fixed set of alternatives makes it clear why their application has been limited. Plan recognition, for instance, does not fit into the classification framework. Hypotheses are created dynamically as part of the evidence gathering process and can be modified by the very evidence gathered to support the hypotheses. Hypotheses are also frequently interrelated. Because of this, it is questionable whether existing numeric approaches to evidential reasoning are applicable to plan recognition. Symbolic representations of evidence while not inherently limited to diagnosis have primarily been applied in this area and so offer little guidance for the use of symbolic evidence in plan recognition.

3.1 Numeric Evidence

By numeric systems of evidence, we mean those systems which represent their belief, evidence, and/or uncertainty in terms of one or more numbers. AI systems have used a variety of ad hoc numeric rating schemes. However, we focus here on three formal or semi-formal evidential reasoning systems: Bayesian probability, the Dempster-Shafer Theory of Evidence, and the MYCIN certainty-factor model.

Bayes' theorem provides one approach to pooling evidence. In a system based on Bayes' theorem, a single degree of belief would be attached to each hypothesis. These degrees of belief would then be treated as probabilities and Bayes' theorem used to compute the conditional probability of a conclusion given the set of evidence hypotheses. Bayes' theorem has a formal basis in probability theory, but suffers from many deficiencies in relation to its use in an evidential reasoning system: large amounts of data about a priori conditional and joint probabilities are required, but rarely available for domains of interest, the complete set of hypotheses must be known in advance, and these hypotheses must be independent.

The Bayesian approach is unable to distinguish between uncertainty and ignorance because it forces probability to be assigned to singleton sets of the possible conclusions. The Dempster-Shafer theory [25] rectifies this problem by allowing belief to be assigned to any subset of the possible conclusions. Thus if we have evidence which results in a degree of belief x in one conclusion, but no other evidence is available, we can represent our ignorance in the belief in each of the other conclusions by assigning the remaining belief to the subset consisting of these conclusions. This representation also allows us to represent the amount of uncertainty of a hypothesis. Since the evidence not supporting a conclusion need not support the negation of the conclusion, a belief interval is produced which is bounded by the belief in the conclusion and its plausibility–the extent to which the evidence allows one to fail to doubt the conclusion. Despite these advantages, the Dempster-Shafer theory still requires that the set of hypotheses under consideration be mutually exclusive and exhaustive. In addition, the computations required by the Dempster-Shafer theory can become intractable under certain conditions.

One of the major problems with both the Bayesian and Dempster-Shafer approaches is that subjective degrees of belief data used in AI systems does do not represent true probabilities and so these approaches are not applicable. The MYCIN certainty-factor approach [47] was developed as a model of how the kind of non-probabilistic and unformalized reasoning typically carried out by experts could be captured in a computerized reasoning system. In the case of expert knowledge, the data acquired is highly subjective and uncertain and it's impossible to explore the all of the conditional probabilities and interrelationships of the hypotheses. The certainty-factor model recognizes these facts and avoids the problems through use of a simplified, approximate application of Bayes' theorem.

The intelligence in numeric systems is in the specification of the combining functions. That is, in the process for computing the numbers. Control schemes based on numeric evidence schemes simply select the best rated alternative. As was discussed in section 2.1, a numeric approach to representing evidence means that the control scheme cannot be flexible and dynamic since the reasoning behind the numeric ratings is unavailable. Typically numbers representing expert judgements are a combination of many different factors. For example, in MYCIN, rule qualifications included information about a priori probabilities, causal connections, and utility. This brings up the question of representational adequacy. Numbers do not provide adequate knowledge about situations to allow intelligent control decisions to be made because they only implicitly represent the many different factors relevant to the situations. Doyle [21] has suggested that the inference qualifications represented by the numeric factors be explicitly represented as part of the specification of the inference rule.

3.2 Symbolic Evidence

In response to the deficiencies of numeric representations of evidence, symbolic representations of evidence have been developed. This approach has received far less research attention than have numeric systems. Therefore, the work presented here does not constitute the kind of formalized methods developed for numeric approaches to representing evidence. Instead, it has served to define the complex problems which much be solved in order to use symbolic evidence effectively. The discussion of endorsements summarizes the main arguments for symbolic representations of evidence and the problems which must be solved to use such systems effectively. MUM, a system under development which uses symbolic evidence for the control of medical diagnosis, represents a recent approach to heuristic reasoning about uncertainty.

Numeric degrees of belief implicitly represent summaries of the reasons for believing and disbelieving the conclusions and the inference rules to which they are connected. As we have discussed in sections 2.1 and 3.1, the fact that the numbers hide the reasoning which has produced them severely limits the reasoning that can be done with them. Systems are unable to treat different kinds of evidence differently or to treat evidence differently in different contexts since the only characteristic which is accessible is how much it is believed. Numbers simply do not provide a rich enough representation to support the kind of reasoning that people use to function effectively in the face of uncertain knowledge.

The theory of *endorsements* [9,10] is one response to the limitations of numeric evidence through the use of symbolic representations of evidence. Endorsements are explicit representations of the factors which affect one's certainty in a conclusion. Cohen stresses the distinction between reasoning *under* uncertainty in numeric systems as opposed to the potential for reasoning *about* uncertainty using a system of endorsements. Since so much more can be known about the evidence, heuristic knowledge can be brought to bear to discount the effect of the uncertainty in a particular context. Reasoning about uncertainty is a knowledge intensive process in which domain specific heuristic knowledge is applied to make the best control decisions given the evidence and the situation. For example, a set of endorsements may be sufficient for one goal, but not for another–in which case the decision could be made to attempt to gather more evidence.

A system which represents evidence symbolically isn't straightforward to develop. If more sophisticated reasoning about evidence is to be done then much more knowledge is required. A system of symbolic evidence doesn't make this any easier–what it does is make it possible to represent and apply such knowledge. Each domain will have a characteristic set of endorsements and a set of methods for reasoning with the endorsements. These methods must include rules for ranking sets of endorsements, rules for combining endorsements (to replace the pooling and propagation combining functions), and rules for resolving and discounting uncertainty. This involves a great deal of information because instead of the uniform, global approaches for dealing with evidence in numeric systems, heuristic reasoning about uncertainty must take into account the characteristics of the context and the particular evidence involved.

MUM (Management of Uncertainty in Medicine) [11] is a medical diagnosis and consultation system which is designed to manage the uncertainty inherent in medical diagnosis. MUM generates workups for chest and abdominal pain. This involves taking histories, asking for physical findings, ordering tests, and prescribing trial therapies. Control decisions are made by reasoning about features of evidence and sources of uncertainty in order to minimize uncertainty or its consequences. The architecture is "based on the idea that managing uncertainty and controlling a complex knowledge system are manifestations of a single task, namely, acquiring evidence and using it to solve problems."

MUM uses a number of different kinds of knowledge. The most basic type of knowledge is *data*, which includes such things as personal and family history and test results. Data must be abstracted through *interpretation functions* to become *evidence*. Interpretation functions are essentially *belief curves* that relate data attributes to evidence or belief in evidence. For example, data about the number of cigarettes smoked per day is abstracted to evidence about the smoking category of the patient: non-smoker, light-smoker, moderate-smoker, or heavy-smoker. In other cases, data is related to belief in a single form of evidence, as duration of chest pain is abstracted to belief in classic-anginal-pain.

Evidence can be characterized by a number of features such as cost to obtain, reliability, and roles. *Roles* represent the relations evidence can play with respect to evaluating belief in hypotheses. MUM recognizes seven roles: confirming, disconfirming, supporting, detracting, triggering, and modifying. The roles which relate directly to belief in a hypothesis are realized in the non-numeric degrees of belief used in MUM: confirmed, strongly-supported, supported, unknown, detracted, strongly-detracted, and disconfirmed. Evidence plays a triggering role when it focuses attention on a particular hypothesis. Modifying evidence does not affect belief in a hypothesis so much as it alters the way diagnosis of the hypothesis proceeds. Evidence can play multiple roles with respect to hypotheses. For example, most triggering evidence is also supporting individually or in combination with other evidence. Collections of evidence which occur regularly and play particular roles with respect to hypotheses are grouped into *clusters*. Systems which use representations of belief need to be able to combine evidence and propagate belief. MUM uses *combining functions* which are local to each evidence and disease cluster to accomplish these functions. By doing this instead of using some global, general function, an expert can precisely specify how belief in evidence affects the belief in a cluster.

Strategic knowledge consists of heuristic knowledge for focus-of-attention and actions for gathering pertinent evidence. Strategies are represented as rules and include the following components: conditions for selection of the strategy, focus policies, and planning criteria. Focus policies guide the choice of disease hypotheses to focus on based on plausibility, criticality, or ability to provide alternate/differential explanations for symptoms of the hypotheses. Planning criteria use cost, roles, and diagnosticity (ability to differentiate alternatives) of the potential evidence to control actions to gather evidence.

More recent work on MUM takes advantage of the fact that for medical diagnosis the inference net from data to diagnosis is static and predetermined. As was discussed above, this is not the case for plan recognition. Medical diagnosis is much more a matter of template matching than is plan recognition. This is not, of course, to say that medical diagnosis is any easier than plan recognition since there are still many uncertainties in the domain. It simply suggests that techniques for control in MUM are unlikely to be directly applicable to plan recognition tasks.

Chapter 4

Plan Recognition

We include a broad range of interpretation and situation assessment applications within the class of plan recognition problems we are studying. The classic example of plan recognition is the interpretation of a series of user actions as particular steps in a task instance. This capability is important for natural language understanding systems which must interpret descriptions of user activities or as part of an intelligent assistant such as POISE (see section 4.1). Situation assessment applications such as vehicle monitoring may also be treated as plan recognition problems. Here, for example, the plans describe vehicle movements or missions and the plan "steps" specify characteristic sensor data and other evidence. What is common to these various applications is the goal of producing a higher-level, more abstract view of the data. These interpretations then provide an appropriate context within which to understand the data.

Plan specifications define the hierarchical relations between the data and more abstract views of the data. Although the exact form and specification of plans varies with the application domain, all plans are composed of subplans. These subplans are then further decomposed into subplans. The decomposition continues until the subplans represent available data. For example, an office domain plan for processing a form may be composed of steps for filling out the form and then sending it to the appropriate office for verification. These steps are further decomposed until they correspond to the actions that a user may take using an automated office system. An aircraft monitoring plan might represent a particular kind of mission in terms of vehicle movements and states. The vehicle movements and states would be expressed in terms of of radio and radar emissions necessary to identifying them. The exact form for specifying plans depends on the domain. The subplans of a plan may be explicitly specified in a hierarchical script-like framework or may be more loosely related through goal and subgoal relations depending upon the application. Each plan also has an associated set of parameters. Plan constraints then define the legal subplan instantiations of a given plan instantiation based on various attributes of the subplan parameters. Thus, different forms may be sent to different offices for verification during processing and different vehicles will have different emissions characteristics.

Plan recognition is the kind of complex, uncertain process which requires the application of AI techniques. The major factors which complicate plan recognition are:

- In general, there are multiple, ambiguous interpretations for each subplan or sequence of subplans.
- Ambiguity is compounded in domains which admit multiple, concurrent plans since the subplans may be interleaved.
- For some applications, interpretation must be done in real-time, relying on preliminary and partial plan data to make "best guesses" about complete plans.
- Since computational considerations generally preclude constructing all the possible interpretations, there is actually uncertainty whether any of the system's interpretation hypotheses are correct.
- The volume of data may be so massive as to preclude complete examination.
- Data may also be missing, uncertain, and/or incorrect.

There is typically insufficient constraint information associated with a plan instantiation to be able to eliminate all but the one correct interpretation from consideration. Of course, the situation improves rapidly if all subplan instances are associated with a single plan since constraint information would accumulate rapidly. However, in many domains multiple, concurrent plan instantiations may occur. For example, users may temporarily suspend a task to start another while waiting for a form to be verified in an intelligent assistant application or multiple vehicles may need to be simultaneously monitored in a vehicle monitoring system. Many applications also require that interpretation be done in real-time–i.e., as the data is being received. An intelligent assistant must try to understand user actions as they are taken if it is to provide maximum assistance and vehicle monitoring is typically required to provide immediate feedback such as in air traffic control. This greatly increases the possible interpretations for a sequence of actions since plan instantiations must be recognized from fragments of the plan and no potential partial plan instantiation may be ruled out as it may be continued later. Becuse these characteristics can lead to a combinatorial explosion of the number of potential ambiguous interpretations for the data, it is often infeasible to construct and evaluate every possible interpretation. Control schemes must instead select and pursue only the "most likely" interpretations. Since these control decisions are uncertain and may be incorrect, there is even the possibility that the correct interpretations are not represented among the constructed hypotheses. Thus, the control process must not only choose the most likely hypotheses, but also decide if the hypotheses cover the correct answer. In some domains, control must also be exercised over the selection of what data to interpret. For example in vehicle monitoring, many sensors will provide continuous output resulting in huge amounts of data for interpretation. Control of data interpretation is even more critical when the data may be in error. The potential for missing or incorrect data greatly increases the number of potential interpretations since we may be uncertain about ruling out plans just because subplans are missing or because constraints are violated.

In this chapter we will examine the POISE and DVMT applications which have motivated this research as well as several other plan recognition systems. The final section of the chapter discusses the characteristics that we feel plan recognition systems must have to address the deficiencies of existing systems.

4.1 The POISE Intelligent Assistant Project

The POISE project [31] involved the development of an intelligent assistant for users of computerized systems. The project encompasses a number of different components. The purpose of the plan recognition component is the development of a model of user activities based on information supplied by another component which monitors the interactions between the user and the computer. In this section, plan recognition in POISE will be presented along with a discussion of the current control scheme and its deficiencies.

In POISE, possible user tasks are represented as hierarchies of script-like plans. Each plan specifies its substeps using a shuffle grammar to denote the relative temporal ordering of the substeps. Additional constraints specify valid values for the parameters of these substeps. The POISE plan recognition component uses these plan specifications to form an interpretation of user actions. "Primitive" plan instantiations representing the actions are passed to the recognition component by the monitor component. Choice points occur following each newly monitored user action as the system must find an interpretation which covers the latest action. Recognition must be done in real-time in order to provide timely assistance to the user. Thus only partial plan instantiation data will be available and no potential interpretations can be absolutely ruled out due to the possibility of their being continued later. Typically, a user will be engaged in a number of tasks at the same time since most tasks require many steps which take place over a period of time. This means that the plan recognition component must also deal with multiple, concurrent partial plans whose steps are interleaved. Errors made by the user must be considered in the interpretation process. One of the important functions of an intelligent assistant is the detection and correction of user errors. An unexpected action could be due to a user error or it might be due to previous incorrect system interpretations. The recognition component must assign blame for errors and correct its interpretations when possible. If the error originates with the user, the interface would be notified and a dialog might be carried out to inform the user and correct the error.

Because of these complexities, constraint information contained in the plan specifications is seldom sufficient to fully disambiguate potential action interpretations. Nevertheless, the role of the plan recognition component in modeling user activities for an intelligent assistant makes it crucial that the correct interpretations be rapidly and reliably formed. In order to accomplish this objective, POISE uses heuristic knowledge to focus on the most likely interpretations to be pursued. While the plans contain object-level knowledge about how it is possible to accomplish various tasks, the focusing heuristics contain meta-level knowledge about how people tend to carry out tasks. This heuristic knowledge in effect supplements the knowledge in the plan specifications in order to disambiguate the alternative interpretations. When faced with ambiguous interpretations for the data, the heuristics are used to make assumptions about the most likely interpretations. Of course, since these assumptions are based on heuristic knowledge, the system must include methods for revising the interpretations as additional data is accumulated.

The original focusing algorithm developed for POISE examined all of the possible interpretations for a new action, ordered them, and selected enough of the more likely ones to cover all actions. The heuristic meta-knowledge was implicit in the part of the focusing algorithm which ordered the alternatives. Procedural embedding of the heuristics means that it's not obvious which heuristics have been applied. This caused major problems when actions occurred which were inconsistent with the existing interpretations. This means that the system had incorrectly interpreted some earlier action(s) and needs to backtrack: identify its interpretation error, retract it, and correct it. However, the only information available to the backtracking system was an (ordered) list of preferred interpretations–the *result* of the focusing *process*. There was no information about how this ranking was achieved, nor was there any information about which interpretations were alternatives based on the control assumptions implicit in the focusing algorithm. Thus, there was no way to reason about which interpretations were likely to be in error and the alternatives to pursue instead. Consider for example, the plans I = a, b and J = b, c and the sequence of actions, a, b, c. The system was unable to reason that the interpretations of action b as occurring in plan I or in plan J (Iab vs. Jbc) were alternatives based on an assumption of plans steps being unshared. Thus at this point in the interpretation process, the system selected {Iab, Jbc} as the best interpretations rather than {Ia, Jbc} (which is more correct based on the heuristics). Because it had no explicit record of the interpretation assumptions it had made and the consequences of those assumptions, it would have been necessary for such a system to resort to chronological backtracking in order to reach the correct conclusion.

The current approach to focusing [6] uses meta-level knowledge in the form of heuristic rules similar to [13] (see section 2.1). These heuristic rules result in pairwise orderings of interpretation alternatives which are explicitly recorded and serve as the basis of the interpretation decisions. A list of meta-rules which justify the orderings are also recorded using an RMS. Backtracking occurs when an action cannot be interpreted within the existing task interpretations. Decisions relevant to the error are located and the heuristic reasons for these decisions examined. If a meta-level heuristic rule results in what is deemed an incorrect interpretation decision, then the rule is made inapplicable at the decision point. The RMS then uses the heuristic justifications to make different assumptions about the relative likelihoods of possible interpretations which results in an updated interpretation decision.

While this focusing scheme has a number of advantages, some of the control difficulties discussed in chapter 2 remain. Although the focusing mechanism makes use of an RMS to record and enforce focusing assumptions, the RMS is not used for automatic dependency-directed backtracking. Focusing in POISE is an example of a problem which is not amenable to dependency-directed backtracking, but can be approached through some form of nonchronological backtracking (as was discussed in section 2.2). Dependency-directed backtracking cannot be used because of its requirement that all relevant knowledge be instantiated in the dependency network so that inconsistencies can be detected and eliminated. This would essentially require that all possible interpretations be pursued and recorded–including those deemed unlikely. Avoiding this work, though, is exactly the point of focusing because it is prohibitively expensive to pursue all interpretations and because it dilutes the value of the system as an intelligent assistant. Thus, the nonchronological backtracking mechanism is external to the RMS. Relevant assumptions are located by determining which control assumptions resulted in the elimination of task interpretations which could explain the current action and the heuristic focusing rules applied to the revised view of the situation.

The focusing heuristics have been structured in a form like Doyle's reasoned assumptions [21]: A UNLESS B ASSUME C. Encoding control knowledge in this way has a number of drawbacks. We must explicitly state when and only when to make an assumption or else inconsistent assumptions may be suggested. Having to precisely specify the exact conditions makes the heuristics complex and difficult to accurately specify. In addition, such a representation is not modular since the addition of new heuristics may require changes to the existing heuristics. Using numeric ratings to resolve conflicting heuristics as has been done for meta-rules (see section 2.1) is not acceptable since it eliminates the kind of explicit representation of control knowledge which is required for an intelligent revision process.

The heuristic control knowledge in POISE is only applied in a limited way to control the construction of interpretation hypotheses. For each new action, the system constructs all possible interpretations for that action (given the existing interpretation assumptions) in terms of top-level task descriptions before it applies any heuristic focusing knowledge. While this approach seems satisfactory for an intelligent assistant for office automation, it may not work in domains such as software engineering where many tasks are accomplished with few primitive actions. This leads to a large branching factor and hence a very large number of potential alternative interpretations. In this case it may be necessary to apply heuristic knowledge to all plan levels during the construction of interpretation hypotheses. It may even be desirable to limit the abstraction level at which data is interpreted until sufficient data is accumulated to provide some level of certainty.

The current system confuses belief in hypotheses with the control decisions about how to develop the hypotheses. Belief in an interpretation hypothesis is implicit in its membership in the set of currently "in-focus" hypotheses. This often forces the system to prematurely commit to one of a set of alternative interpretations regardless of how tenuous the evidence is. However, when there is a great deal of uncertainty over the proper interpretations it might be better to pursue interpretations other than those that are currently most supported. This would give the system control over using a depth-first vs. a breadth-first approach to pursuing hypotheses. An independent representation of belief in hypotheses would also make it possible to provide more information to a user about the relative level of belief and uncertainty in the alternatives. This same knowledge could be used to guide and limit the revision process. Uncertain assumptions could be examined first and revisions can be limited to those assumptions which are not strongly believed. This is important since it is computationally infeasible to reconsider all interpretation decisions when faced with a contradiction. It may in fact, be possible to recognize incorrect decisions before a "hard" error is caused by evaluating the uncertainty in the system.

4.2 The Distributed Vehicle Monitoring Testbed

The Distributed Vehicle Monitoring Testbed (DVMT) [12] is a research environment for the evaluation of alternative designs for distributed problem solving networks. The vehicle monitoring task involves the generation of maps of vehicle movements through some geographical area. Input data is provided by a set of acoustic sensors distributed over the area to be monitored. Because of the distributed nature of the acoustic sensors, there can be advantages to distributing the computational resources. This requires a problem solving architecture which makes it possible for each node to work with only partial information by communicating with other nodes and by coordinating its problem solving activities with these nodes. The testbed simulates a network of nodes, each of which is a complete, goal-directed blackboard system capable of functioning as a centralized vehicle monitoring system.

The vehicle monitoring task can be formulated as an interpretation task very similar in character to plan recognition in POISE. The interpretation of acoustic sensor data involves the use of a simple, four-level grammar (plan) representing vehicle tracks in terms of characteristic acoustic sensor data. The data blackboard is divided along these four levels of abstraction. At the lowest level of abstraction, the *signal* level, hypotheses correspond to signal data received from low-level analysis of acoustic sensor data. The *group* level involves collections of harmonically related signals–signals emanating from a common source vehicle. Vehicles are represented as collections of groups associated with particular vehicle types at the *vehicle* level of the blackboard. Finally, the *pattern* level involves collections of vehicles with specific spatial relationships as well as single vehicles. At each level, hypotheses may represent single locations or tracks covering a sequence of locations. The grammar specifies relations between classes of hypotheses from one level to the next as well as constraints such as vehicle velocity and acceleration. The goal of the DVMT is to create pattern-level track hypotheses representing the vehicle movements being monitored by the acoustic sensors.

Vehicle monitoring is an inherently uncertain task. The number of vehicles being monitored is unknown. Constraints in the track grammar are fairly weak. Sensors can fail to detect signals, malfunction and "detect" non-existent signals, or incorrectly determine the location and frequency of signals. These factors result in large numbers of alternative interpretations for a set of signal data–ambiguity and uncertainty which much be resolved by the control process. The DVMT deals with this uncertainty through the use of an opportunistic, goal-directed blackboard architecture. A goal-directed blackboard system involves an extension of the typical HEARSAY II blackboard architecture to include a goal blackboard. Goals are used to focus problem solving through subgoaling while maintaining the advantages of opportunistic data-directed control common to blackboard systems.

Goals are created on the goal blackboard by the blackboard monitor in response to the changes on the data blackboard–e.g., the creation of hypotheses. Goals explicitly represent the system's intention to create hypotheses with particular attributes. The insertion of goals on the goal blackboard results in the planner instantiating Knowledge Sources (KSs) which might achieve the goals. The planner executes a KS's precondition procedure to estimate whether the KS is likely to generate hypotheses to satisfy the desired goal. Hypotheses are created by executing the Knowledge Source Instantiations (KSIs). Knowledge Sources (KSs) are provided to abstract hypotheses from one blackboard level to the next, create tracks from location hypotheses, extend tracks, and merge overlapping tracks. KSs are also provided for internode communication of hypotheses and goals as part of distributed problem solving.

Goals, KSIs, and hypotheses are assigned numeric ratings as they are created. Goal ratings are based on the ratings of the hypotheses which stimulated the creation of the goal and on the ratings of supergoals (goals which have the goal being rated as a subgoal). KSI ratings reflect both data-directed and the goal-directed components. A KSI rating is a weighted sum of the rating of the goal which the KSI is to accomplish and the estimated rating of the hypothesis the KSI will create. The scheduler uses this formula to rate KSIs on the agenda and selects the most highly rated KSI for execution at each system cycle. Hypotheses are rated as they are created by the executing KSs. The knowledge for producing these ratings is one of the major engineering aspects of the testbed. Though KSs may be independent in principle, the ratings functions associated with the KSs must be consistent with each other if effective control is to result. Thus the reasoning about control decisions is really being done during the engineering of the system rather than during the running of the system.

As we've discussed earlier, systems which use numeric ratings are unable to explicitly consider the evidence implicit in their numeric ratings. They cannot reason about which actions are best for resolving their uncertainty since essentially all they know is the *amount* of their uncertainty–not the *source* of the uncertainty. Much of the research on control for the DVMT has dealt with focusing to avoid distraction from noisy and incorrect data (e.g., data due to ghost tracks and sensor failures). Since the likelihood of potential sources of uncertainty in particular situations cannot be explicitly considered, these focusing mechanisms involve relatively unsophisticated, uniform methods. The DVMT is also unable to reason about the relationships between actions and so may waste processing resources successively pursuing actions which have the same purpose. For example, there are typically a number of sequences of actions which can be used to extend a track hypothesis. Failure of one approach (e.g., due to missing or garbled data) suggests that the other approaches will also fail if they are seeking the same sorts of evidence. What is needed is an action which will seek different sources of evidence to resolve the uncertainty. Processing may also be wasted accumulating less critical evidence. A group-level hypothesis may be pursued by interpreting additional signal data prior to examining more crucial evidence of the group's possible inclusion in a vehicle track. Control should be able consider the purpose of actions in relation to the goal–i.e., producing evidence of complete tracks.

A good deal of effort has been expended developing systems for understanding and explaining DVMT activities because numeric representations of evidence hide much of the problem solving activity. It is difficult to determine why data was or was not used and why hypotheses are or are not believed (beyond meaningless restatements of the ratings). This sort of knowledge is important for users to have confidence in the system's answers and to detect problem solving errors or situations which call for additional problem solving knowledge. Since the ratings play such an important role in control it is not surprising that they do not even really represent belief in the hypotheses, but include focusing information as well. For example, (potential) track extensions are always rated more highly than their base tracks when they may, in fact, be less certain than a well-supported base track. Finally, the simplistic numeric scheme being used is incapable of representing more sophisticated evidential concepts such as uncertainty and conflicts. This leads to the goal satisfaction problem: how to determine when the system is done–that is, when it has found all the answer hypotheses.

A distributed approach to vehicle monitoring increases the need for intelligent control which reasons about evidence and the best ways to obtain it. In a distributed problem solving environment, no node has access to all of the signal data necessary to be able to interpret vehicle movements. This requires communication between the nodes to request and receive necessary data and evidence. However, communication involves cost, so it is important for such systems to request and transmit only the most appropriate data for reducing the overall interpretation uncertainty.

4.3 Other Plan Recognition Systems

A review of the AI literature reveals that other plan recognition work suffers from the same limitations that our research addresses. In the all of the examples systems discussed in this section, plan recognition is intended to provide an understanding of human goals and intentions based on descriptions of actions such as would be available from a natural language comprehension system.

Schmidt, Sridharan, and Goodson [45,46] present their plan recognition process as a model of how humans understand others actions. Their process involves what they call a *hypothesize and revise strategy.* This approach is motivated by their belief that humans "do not use a strategy of heuristic search to explore a large space of possible interpretations of a sequence of actions," but "explore only a few, usually only one, hypotheses at a time" and are able to adapt the hypothesis to the observations "by a process of refinement and revision." While this sounds very similar to our emphasis on revision, their use of revision is very different. In their system, plans are very general structures which can account for a large number of activities. As actions are interpreted, the revision process uses rules indexed to classes of constraint violations to customize the instantiations of general plans to the particular context. For example, a plan template for making and eating food will not contain any specification of the particular implements to be used nor the possible sequences of actions to obtain and prepare the food. Action observations are used to bind object variables to the particulars of the situation and to insert action hypotheses as appropriate to the goals, subgoals, and prerequisites of the plan instantiations. This makes it possible to deal with alternatives and errors in a way that has not been possible in POISE. However, this generality comes at a cost. More general plans provide far fewer expectations about future actions. Fewer expectations mean less constraints and thus greater uncertainty. In the extreme, we could imagine this approach being taken with a single completely general DO-ACTION plan consisting of an indeterminate number of substep DO-ACTIONs. The result would be what [46] terms "postdictive" plan recognition. Such an approach is inappropriate for plan recognition applications which require expectation information–such as an intelligent assistant. If more specific plan templates are to be used, however, there are several problems which must be solved, but which are not addressed in [45,46]. In particular, this system relies on the initial plan instantiation selected being able to be modified to account for later actions. In general, though, there will be multiple relevant alternative instantiations which must be considered. This work has no method for selecting the correct initial plan instantiation (focusing), no method for evaluating belief in alternative instantiations, and no method for shifting between plan instantiations should later action observations completely invalidate an alternative.

In work by Wong [53], plan recognition is used to establish the context within which actions described by English sentences are taking place. Contexts are hierarchies of plan/script instantiation frames which can be used to fill in information not made explicit in the English text. This work does address the problem of selecting the appropriate contexts to some degree. Unfortunately, no explicit representation of evidence and uncertainty is used so the "best validated" context instantiation is chosen based on an ad-hoc heuristic measure. Contexts are deemed more likely when they involve the fewest number of new plan instances and the fewest number of statements to establish context (links from action descriptions to context instantiations). This heuristic is somewhat similar to the "continued vs. started" heuristic in the existing POISE system, however this system has no facility for reconsidering and revising its context interpretations. It can only perform what Wong terms "first-impression" recognition as opposed to "contemplative" recognition. Wong recognizes that this is a serious problem when initial context clues are weak or when the initial context suggested is wrong and must be revised as additional data is accumulated. Finally, there is little control in this system over the instantiation of scripts. The recognition process is bottom-up from an action to all possible contexts–existing and newly created. This approach is infeasible when a large number of potential contexts are suggested as was discussed in connection with POISE.

Work by Allen [2] once again deals with the interpretation of natural language in terms of the goals and intentions of human agents. Allen's system views utterances as speech acts: actions as part of a plan. Rules about likely inferences are then used to drive inferences from the utterance toward the goals and intentions of the speaker. The search is considered to be through a space of partial plans with potential actions in each state being represented by the applicable plan inference rules. Control is accomplished by rating the alternative partial plans based on a number of heuristics. These heuristics refer to domain-independent relations between plans, their subplans, preconditions, and effects. Ratings are produced from the heuristics using an ad-hoc system of weights. The system is not a general plan recognition system as it is designed to work from a single utterance rather than a sequence of utterances (although work to extend the framework has been done since [38]). Because of this, there is never any need to reconsider interpretations. This is fortunate since the system cannot support revision due to the ad-hoc and implicit nature of the control rating scheme. Finally, the system relies on there being a very small number of plans or goals which the user might be attempting to accomplish–the problems and uncertainties of indexing into a large set of plans is ignored.

Work by Kautz [33,34] is concerned with the development of a general plan recognition system. Kautz uses a logic framework to specify plans in terms of a decomposition hierarchy, a specialization hierarchy, and temporal and parameter constraints. Closed-world assumptions are applied to the action hierarchy to produce the action taxonomy: a complete description of the ways actions can be performed. Plan recognition is then viewed as deductive inference based on the axioms representing the observed actions and the action taxonomy. The framework handles incomplete and missing data in the sense that permissible interpretations can still be determined. It cannot, however, reason about the data in order to resolve uncertainty over partial or conflicting interpretations and so cannot deal with data which is actually incorrect. This work is complementary to our own for it provides a precise, formal semantics for plan recognition in terms of permissible deductions. It does not provide the basis for a practical plan recognition system, however, since it lacks a framework for including the control knowledge necessary for making only likely deductions and interpretations. The only focusing knowledge which the system can apply is the so-called "simplicity constraint." This heuristic closely corresponds to the POISE "continued vs. started" heuristic in its minimization of potential hypotheses although here it is given a formal semantics in terms of circumsription. Kautz makes much of his system's basis in logic and freedom from "probabilistic inference." However the simplicity constraint represents heuristic knowledge which is applied without any explicit representation of its application or its effect on the system's conclusions. Furthermore, since no general purpose theorem proving techniques are capable of handling the inferences in this system, the plan recognition system which is proposed for implementation has a very different flavor from the formal work. In fact, the approach which is proposed is very similar to early POISE control schemes: all of the potential top-level plans which might be supported by the observed actions are created, ranked, and pruned to cover the actions. All observed actions are automatically driven up to all top-level actions through all disjuncts without any application of control knowledge. Focusing decisions are made implicitly through the ranking and covering operations and must consider all possible interpretations of the data at every cycle.

4.4 Plan Recognition Requirements

In this chapter a number of plan recognition systems have been examined and their deficiencies discussed. This final section recaps the major problems that must be solved in order to develop intelligent plan recognition systems. The next chapter will introduce our approach for solving these problems. We feel that an intelligent plan recognition system must be able to:

- Evaluate the level of belief and uncertainty in alternative interpretations.
- Understand the reasons for beliefs.
- Encode and apply heuristic control knowledge.
- Revise interpretation hypotheses as information accumulates.
- Handle uncertain and incorrect data.
- Integrate data from multiple sources.
- Actively control data accumulation.
- Reflect system goals in control decisions.

Of the systems examined, only the DVMT has any ability to evaluate its level of belief in interpretation hypotheses. A single number representation of belief cannot differentiate between belief and uncertainty, however, and does not provide access to any of the reasons for the beliefs. The POISE focusing system does contain a representation of its reasons for making focusing decisions, but does not provide any measure of the belief or uncertainty of the alternative hypotheses. Use of an independent evidential reasoning system has many advantages. Knowledge of belief and uncertainty can be used by the focusing and revision processes to develop efficient and sophisticated control schemes. Control need no longer be limited to pursuing only the most highly rated/believed hypotheses, but may reason about the best actions to take given the existing interpretations and data. Reasons for beliefs can be used to justify system interpretations to users and to facilitate analyses of system performance. Existing plan recognition systems cannot explain why they believe the current hypotheses, why they are still uncertain about them, and why they chose to perform particular actions.

One of our major concerns in earlier work was the development of a framework for representing and applying heuristic knowledge for focus-of-attention. This is an important issue because of the ambiguity inherent in many plan recognition applications. It is computationally infeasible to use a brute force approach and pursue all of the potential interpretations. Decisions must be made about which interpretations and data to devote the system's limited processing resources to. The sort of commonsense and expert knowledge that can provide this focusing is available if there is an appropriate framework within which to encode and apply it. The existing systems that make use of heuristic focusing knowledge have inadequate mechanisms for dealing with the amount of knowledge we envision using.

It should be noted here that we intend heuristic control to be extended to all aspects of the interpretation process. In existing systems, little attention has been paid to controlling the actual steps in building potential interpretations. In POISE and the Kautz' system for example, data is abstracted to top-level plan instantiations before any sort of focusing intelligence is applied. While this may be acceptable for some domains, it is not in general. In the software engineering domain for POISE, it is easy to recognize lower-level plan instances such as editing a file, but difficult to determine what top-level plan instance these actions may be part of. This is because of the weakness of the constraints and the large number of disjunctions in the plan library (editing a file can be a part of nearly every plan). In such situations, constructing all possible interpretations for each action and then eliminating those deemed less likely is impractical. Instead, the system needs to reason about whether it is appropriate to abstract the current interpretations further based on the degree of uncertainty over what they represent.

Any focusing scheme excludes interpretation alternatives based upon uncertain, heuristic knowledge. Since additional evidence may show that incorrect focusing decisions were made, such a scheme must include provision for revising its decisions and reconsidering abandoned alternatives. Both POISE and the Kautz' system provide some revision capability. In the case of Kautz, since focusing consists of applying a single heuristic there is little reasoning that can be done during the revision process. In POISE the problem is that it is difficult to integrate the new knowledge with existing focusing knowledge to control the revision process. Thus revision suffers from a lack of control knowledge for focusing the revision process.

None of the example systems handle uncertain data in a satisfactory way. Data may be uncertain because of noise or errors or because data is missing. Handling uncertain data means more than simply making those inferences which are possible given missing data or ignoring data which seems to be noise since it cannot be satisfactorily interpreted. An intelligent plan recognition system should be able to reason about the nature of the uncertain data–e.g., what data is missing or how that data has been garbled. In order to fully comprehend a system's interpretations, users must have access to knowledge about assumptions that underlie the interpretations.

One approach to handling uncertain data is to be able to integrate multiple forms of evidence to reach interpretations. Existing plan recognition systems have limited their evidence to that obtained from a single main source of input data. True vehicle monitoring situations will require the integration of data from different types of sensors and other intelligence sources. Just as there is knowledge beyond that contained in the plans which can be used for control and focusing, there typically is additional knowledge which can used as evidence for or against interpretation hypotheses. This sort of domain knowledge is available to expert users to help resolve interpretation uncertainties. Such domain knowledge, termed "first-principles" knowledge, has been investigated for its role in human understanding of software engineering tasks.

In addition to the problems discussed above, there are two more areas of concern which are best considered extensions to the view of plan recognition used by the existing research systems. In any actual plan recognition application, the overall system will have some purpose. Rather than use completely different systems tailored to the application it should be possible to make the operation of the system sensitive to different goals. For example, an aircraft monitoring system might be used for air traffic control and it might also be used for military monitoring. The purpose of military monitoring applications may be for the protection of certain installations. In this case it would be less important to form complete models of the environment than it would be to focus on aircraft which could pose a threat. Thus the goal of protecting the installation should be able to influence the interpretation process of the aircraft monitoring system.

Another extension involves active control over the gathering of data. In vehicle monitoring, for example, sensors might be under the control of the interpretation system. The system could then adjust sensor characteristics to gather the kind of data that would best resolve its uncertainties. Active control might also be used in an intelligent assistant such as POISE by allowing the system to interact directly with the user to gather information. Active control of data gathering could greatly enhance the efficiency of an interpretation by allowing the system to gather the most useful data. The degree to which this is possible will depend upon the domain. In an intelligent assistant there would be only limited opportunities for the system to actively pursue information.

Chapter 5

Evidence-Based Plan Recognition

In this chapter we introduce our view of evidence-based plan recognition. The first section outlines the basic elements of the approach and explains how they allow us to address the plan recognition requirements of section 4.4. In the Architecture section, the various kinds of knowledge necessary for such a system are examined along with instances from the POISE and DVMT domains. Finally, an example of evidence-based plan recognition using the vehicle monitoring domain is presented.

5.1 The Approach

By evidence-based plan recognition, we mean that plan recognition should be treated as a process of *gathering evidence* to *manage uncertainty.* Uncertain interpretation hypotheses must be "proved" by collecting appropriate additional evidence. This is a unique view of the plan recognition process. Typically, plan recognition systems have used a "constraint satisfaction" approach, with the input data providing the interpretation constraints. Of the systems that were reviewed in chapter 4, only the DVMT could be described as accumulating evidence. The DVMT uses an extremely limited concept of evidence, however, and does not make explicit decisions about the uncertainties its actions are intended to resolve. Extending the representation of evidence and using this representation as the basis for control decisions makes it possible to develop a framework within which the *purpose* of actions can be understood. Interpretation actions are taken in order to resolve particular sources of uncertainty. Thus, the most appropriate action to take depends upon the most "desirable" uncertainty to try to resolve and the "best" action to resolve it. A system which reasons about its control decisions in this way can truly *gather* evidence to *manage* uncertainty rather than just *accumulate* evidence to *resolve* uncertainty. The distinction results from the existence of multiple system goals and the need to consider the tradeoffs during the control process. The immediate effect of an action must be weighed against the long term role of the action as well as other concerns such as execution time and safety.

Taking this view of plan recognition, it is possible to develop a system which meets the requirements outlined in section 4.4. The key characteristics of the approach are:

- Plan, subplan relations are treated as uncertain, evidential relations.
- Evidence and sources of uncertainty are explicitly represented.
- Heuristic control decisions are based on the sources of uncertainty in the hypotheses and the need to manage uncertainty.

Treating plan, subplan relations as evidential relations means that we treat the interpretation inferences that are based on these relations as uncertain inferences. By way of comparison, the constraint satisfaction approach to plan recognition treats these relations as absolute grammatical relations. Thus, instead of saying that a (grammatically correct) subplan instantiation *satisfies* a subgoal of a plan instantiation, we say that it is (uncertain) *evidence* for that plan instantiation. This gives us a better framework for dealing with the inherent uncertainties of plan recognition. Interpretation hypotheses can now be represented with an evidential scheme that makes it possible to evaluate the levels of belief and uncertainty in the alternative interpretations. Uncertain or incorrect data can easily be accommodated since this possibility simply results in additional uncertainty for hypotheses relying on such data. Multiple sources of evidence can be integrated because all evidence is treated in a uniform fashion–i.e., as uncertain, evidential inferences. Hypothesis revision is naturally accommodated because an evidential representation system is inherently nonmonotonic–the belief in the alternatives changes as evidence is accumulated. "Inconsistencies" are represented as contradictory evidence and as such are just another source of uncertainty to be resolved.

By an explicit representation of evidence, we mean that evidential links are explicitly maintained between symbolic representations of evidence and the representations of the hypotheses that the evidence supports. This approach makes it possible to reason about more aspects of the evidence than simply its "strength." Knowledge of the kinds of evidence underlying beliefs makes it possible to understand the *sources of uncertainty* in the beliefs. Evidence provides uncertain support for a hypothesis because there are conditions under which the evidence may fail to support the hypothesis: the sources of uncertainty in the evidence. Numeric rating functions gathered from experts typically summarize just such knowledge–along with a priori likelihood judgements. Explicit information about the sources of uncertainty in evidence makes it possible to evaluate belief dynamically–in specific contexts–rather than having to rely on general, a priori judgements. For example, correlated signals more strongly support a track when the signals are the same frequency than when they differ (but provide support for the same vehicle type). This evaluation would be impossible to perform in a system which merely recorded levels of belief since the levels of belief of the lower-level hypotheses (group and individual vehicle positions) cannot take this more global distinction into account.

The major advantage of sources of uncertainty knowledge is that it provides the perfect basis for making control decisions. Since the goal of plan recognition is to "prove" hypotheses by resolving uncertainty, a sophisticated control process must be able to understand the sources of uncertainty in the hypotheses and reason about the best actions to take to resolve them. Thus sources of uncertainty are used to elucidate control choices by a process that determines which goals remain unsatisfied due to excessive uncertainty, what the sources of that uncertainty are, and what actions could provide evidence to resolve the uncertainty. Note that the "actions" we refer to here involve the way the interpretation system pursues evidence for its hypotheses as opposed to the "domain actions" that the system is trying to interpret. The most frequent type of interpretation action is the interpretation of data and hypotheses to infer evidence for higher-level hypotheses. *Managing* uncertainty requires that the system be sensitive to the various goals of the application. Since the purpose of the different actions is now understood, a variety of factors can be weighed during the decision process. For example, the facility protection goal in a vehicle monitoring application results in increased importance being placed in resolving any uncertainty in the *identity* of potentially hostile vehicles. Active control of data accumulation is also easily accommodated in a system which reasons explicitly about uncertainty. Understanding what information would be most effective at reducing its uncertainty, such a system might choose to actively gather the evidence rather than waiting passively for data to constrain its interpretations.

Sources of uncertainty information also provides an ideal framework for specifying and applying heuristic control knowledge. Heuristic focusing and control knowledge simply represents expert knowledge about methods for dealing with uncertainty. POISE focusing heuristics, in effect, specify what alternatives to gather additional evidence for based on implicit decisions about evidence and uncertainty represented in the rule antecedents. For example the "continued vs. started" heuristic is really saying that it is best to look for additional evidence for the in-progress hypothesis since, implicitly, it has accumulated more evidence. Using a system for representing evidence and uncertainty, such a heuristic can be generalized in two stages. First, we can imagine a heuristic which says to "pursue the most believed hypothesis." This form is advantageous because there no longer needs to be a large collection of heuristic focusing rules which might offer conflicting advice. The same level of reasoning must be done, i.e., figuring out which alternative has the most evidence for it, but this reasoning can be done in a more logical way as part of the evidential reasoning system. The problem with this form of the heuristic is that it still requires qualifications specifying exactly when it's *not* best to pursue the most believed hypothesis. This is because pursuing the most highly believed alternative–while usually the best way to gather evidence–may not always be the most effective way to resolve particular interpretation uncertainties. Selecting the correct decision depends upon an understanding of the purpose of the action as well as the characteristics of the situation. Thus an alternative method for capturing this focusing knowledge is a scheme for deciding what actions can potentially provide the best evidence based on resolving the relevant *uncertainties.* Again, it's not that any less knowledge is needed to be able to reason effectively in this way, it's simply that a framework keyed to interpretation uncertainties provides a logical, modular framework for the specification and application of expert focusing knowledge. Actions are taken for the purpose of resolving particular uncertainties so it makes sense to identify which actions are best for which uncertainties.

5.2 Architecture

In this section we discuss the architecture of an evidence-based plan recognition system. We first describe the components of such a system and how they interact. The sections that follow then contain more detailed information about these components and the types of knowledge they contain.

Figure 5.1 illustrates the major components of an evidence-based plan recognition system and the relations between them. The controller is responsible for executing the basic control loop:

1. Expand control plans until primitive plans are selected using a multi-step process of expanding plans and focusing.
2. Execute the interpretation action represented by the primitive plan.
3. Update plans by checking for satisfaction of subgoals based on action outcome.
4. Repeat.

The control plan instances created by the planning process are maintained on the Control Plans Blackboard. Expansion of the control plans is accomplished using the Planning KSs. Interepretation hypotheses are developed by executing interpretation actions using the Evidential Inference KSs and the Data Gathering KSs. The hypotheses are maintained on the Interpretation and Data Hypotheses Blackboard. This blackboard also contains the information about the evidence supporting the hypotheses, the uncertainty in that evidence, and the relations between the hypotheses.

5.2.1 Hypotheses

Hypotheses represent plan instantiations for which evidence has been gathered–i.e., for which we have some level of belief or disbelief. In order to control the creation of hy-

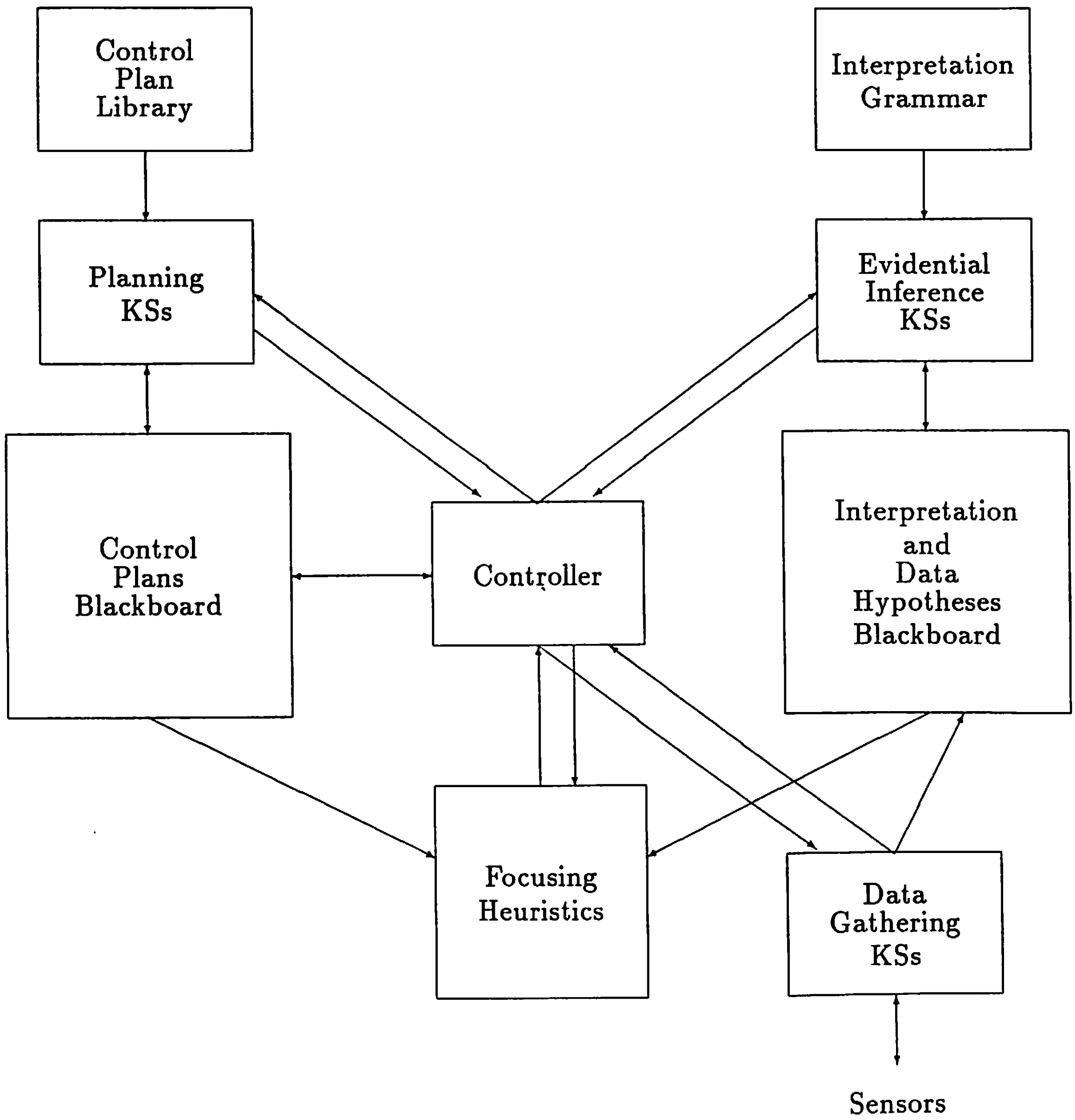

Figure 5.1: Architecture of an Evidence-Based Plan Recognition System

potheses we must be able to represent just the level of plan hypothesis that the evidence actually supports. This means that hypotheses at any level of abstraction must be treated in a uniform fashion. Most existing plan recognition systems force hypotheses to be abstracted to top-level plans–through multiple levels of uncertain disjunctions–regardless of whether there is sufficient evidence to guide the process. This is because of the special role that top-level plans play in these systems. Hypothesis uniformity is also important for the integration of evidence from a variety of sources since this evidence might support hypotheses at any level of plan abstraction.

Hypotheses are complex structures. They consist of multiple versions or *extensions* and include parameter values and links to their supporting evidence. One of the key characteristics that distinguishes plan recognition from diagnosis problems is that evidence not only *justifies* the plan recognition hypotheses, it also *refines* them (see chapter 3). That is, we don't simply gather evidence to decide our belief, we gather it to define exactly what it is we believe as well. Plan definitions not only specify what constitutes valid evidence for the plans, but also how the plan parameter values are related to the characteristics of the evidence. For example, in vehicle monitoring, group-level hypotheses not only provide evidence for vehicles, but also define the vehicle type and position. As a consequence, multiple versions of each hypothesis, called extensions, must be used to represent the various plan refinements supported by the (uncertain) evidence. Uncertainty is no longer simply a question of whether or not evidence supports a hypothesis, it is also now a question of exactly what extension is correct.

Hypotheses maintain explicit links with the evidence which supports or detracts from them. The evidential links include information about the type of evidential inference, the role of the inference in the hypothesis, and the sources of uncertainty in the inference. The sources of uncertainty in each evidential inference are represented as symbolic tags on the evidential links. Sources of uncertainty are not propagated, but are accessible by tracing the evidence hierarchy. See sections 5.2.2 and 5.2.4 for further information on sources of uncertainty.

5.2.2 Evidence

By evidence, we mean the reasons to believe or disbelieve hypotheses. As discussed earlier, we view subplan instances as (uncertain) evidence for plan instances. Rather than summarize the "quality" of this evidence numerically, however, we maintain explicit links between the evidence and the supported hypotheses. This makes it possible to understand why the evidence fails to conclusively prove the hypotheses by giving us access to the sources of uncertainty in the evidence–that is, the reasons the evidence may fail (see section 5.2.4.

Viewing plan, subplan relations as evidential relations means that we view plan definitions as specifications of the valid evidential inferences, $\{A_i\} \Rightarrow B$, where the A_i are sources of evidence and B is a plan. Plan inferences are uncertain, evidential inferences rather than deductive inferences because the existence of the evidence, $\{A_i\}$, is not sufficient to guarantee the conclusion, B. These inferences are in fact a form of *abductive* inference [7]: if x causes y and y is true then hypothesize x. Plan definitions can be viewed as statements that if plan B occurs then the data $\{A_i\}$ will (be caused to) occur. Thus B is an explanation for $\{A_i\}$. Of course, abductive inferences are merely plausible inferences as there may be other explanations for the same data. In addition to this basic uncertainty, plan recognition inferences are uncertain for other reasons as well. For instance, plans must typically be inferred based on incomplete, partial evidence. That is, plan B may be hypothesized based on evidence $\{A_j\}$ where $\{A_j\} \subset \{A_i\}$. A complete discussion of uncertainty may be found in section 5.2.4.

There are two ways to go about resolving the uncertainty in plan hypotheses resulting from these evidential inferences:

1. Gather additional evidence to directly resolve the uncertainty in the inferences.

2. Gather independent evidence for the supported hypotheses.

Explicit representation of the evidential links makes it possible to understand what the remaining *sources of uncertainty* are in an inference and gather evidence to resolve these sources of uncertainty. For example, a hypothesis B, based on the evidence $\{A_j\}$ (where $\{A_j\} \subset \{A_i\}$ above) is uncertain because only part of the conditional evidence is being used to make the inference. Thus, uncertainty in B could be resolved by accumulating the rest of the evidence $\{A_i\}$. There are typically a number of sources of uncertainty for each source of evidence. Each of these sources of uncertainty will then have a characteristic set of sources of evidence which can be used to resolve the uncertainty.

Independent evidence for this same hypothesis B, requires that there are additional ways to infer the plan, e.g., $\{D_i\} \Rightarrow B$. Then the evidence $\{D_i\}$ can be used to lend further support to B and so resolve uncertainty in the hypothesis. It should be noted that evidence is not necessarily independent just because it is based on a separate inference axiom. Instead, what must be true is that the independent evidence must not include the same sources of uncertainty as the old evidence. For example, if evidence from radar scanning and radio emissions detection can both be affected by the same kind of weather conditions (and affect the interpretation in the same way) then one could not be used to resolve uncertainty based on the other.

In POISE and the DVMT, plan inferences are based on partial conditional evidence. A plan for Purchasing Items may be inferred from the occurrence of a Receive-Purchase-Request plan in POISE. Likewise, a vehicle may be inferred from a single group hypothesis. In each case, the recognition systems pursue these hypotheses by trying to complete the set of subplan instantiations for the hypotheses. Neither POISE nor the DVMT use multiple sources of evidence which could be used to develop independent evidence for the hypotheses. For example, POISE could interrogate the user as to the correctness of the plan for Purchasing Items while the DVMT could make use of data from other types of sensors to support vehicle hypotheses.

As well as the supportive evidential inferences described above, the system must also handle negative evidential inferences–i.e., inferences which detract from belief in hypotheses. Some negative evidence may result from what we term direct negative inferences–that is, from inference axioms of the form $\{A_i\} \Rightarrow \neg B$. These inferences are not based on axioms which result from the standard plan definitions, but on additional evidential axioms which may be specified. Most negative evidence, however, results from inferences which are based on the standard plan definitions. Viewing the plan definitions as statements that if plan B occurs then the data $\{A_i\}$ will (be caused to) occur, it is clear that this is equivalent to inferences of the form: $\neg\{A_i\} \Rightarrow \neg B$. Thus evidence gathered for an alternative interpretation of evidence A_k in an inference $\{A_j\} \Rightarrow B$ acts as negative evidence for the plan hypothesis B.

One issue for systems which use explicit, symbolic representations of evidence is how to go about evaluating the level of support provided by the evidence. The level of support provided by evidential inferences depends on the likelihood of the sources of uncertainty holding true. In conventional numeric evidential reasoning systems, this evaluation is made using a priori knowledge and is fixed in a degree of belief rating. While the exact same information that is used in the numeric combining functions could be used to evaluate the explicit evidential representation, it is possible to do much better if additional knowledge can used to examine the particular situation. The likelihood of inferences can in part be evaluated based on the strength of the constraints which validate the inference. The strength of constraints varies with the particular inference axiom and with the premise evidence, i.e., $\{A_j\} \subset \{A_i\}$ as above. For example, each successive vehicle track extension provides additional support for a vehicle track hypothesis. That support is not uniformly additive, however, because little or no support is provided until the number of correlated signals reaches some sort of minimum, support provided by a single extension is much smaller than the corresponding fraction of belief in a fairly long track, and the length of complete tracks varies. Belief is not just some more complex, fixed function of track length either as it would be in a numeric reasoning system. This discussion highlights the differences between the sources of uncertainty and the factors which can used to evaluate likelihood. Sources of uncertainty are the reasons that evidence may fail while uncertainty factors can be used to judge the likelihood of failure.

5.2.3 Data

We use the term data to refer to externally generated information which can serve as a source of evidence for interpretations. The other sources of evidence are the plan hypotheses themselves since they provide evidence for higher-level hypotheses. All sources of evidence must be processed, or "interpreted," before they become evidence. This involves evaluating the plan constraints and deciding how to represent any uncertainties. For example, radar data may support an existing vehicle hypothesis or it may support a new vehicle hypothesis. The exact representation of this uncertainty is a heuristic control decision which must be made during the interpretation process.

Both POISE and the DVMT use only single sources of interpretation data. In POISE, data consists of primitive plan hypotheses representing abstracted views of the actions taken by the user. The recognition of tasks then requires the interpretation of these primitives through several levels of plan hypotheses. Data in the DVMT consists of signal hypotheses which simulate the result of low-level processing of acoustic sensor output. Each signal hypothesis represents a perceived environmental source and includes the following information: frequency, position, time of detection, and a numeric belief rating. Data must be interpreted through several levels in the plan grammar before it provides evidence for vehicle track hypotheses.

While the DVMT has used small amounts of simulated data, a real-world vehicle monitoring system would generate such large amounts of data that it would be infeasible to consider all of the data. However, since most of this data results from noise or irrelevant signal sources, failing to examine and interpret all of the data can have little effect on the uncertainty of the system. This is in marked contrast to POISE where all data must be interpreted even if it turns out to be an erroneous action on the user's part. This suggests the need for some kind of system parametrization to account for the role that data plays in driving the interpretation process. In the DVMT, the examination of data must be strictly controlled based on its potential for providing evidence for the developing interpretations. User actions in an intelligent assistant, on the other hand, drive the interpretation process. The different roles that data plays in different applications can be accounted for through the use of system goals (see section 5.2.7).

5.2.4 Uncertainty

Most evidence is inconclusive. That is, it does not confirm or disconfirm a hypothesis, but merely supports or detracts from the hypothesis. The reason for this is that the evidence is uncertain–there are conditions under which the evidence may fail to support the conclusion. These conditions–the reasons that evidence can fail–are what we term the *sources of uncertainty* in the evidence. One of the important advantages of our approach is that we make it possible to understand exactly what the sources of uncertainty are in the evidence for a hypothesis. This allows the control process to reason about the best ways to resolve the system's uncertainty by considering the reasons for that uncertainty.

We view plan definitions as specifications of the valid evidential inference axioms for the plan recognition application. The form of these axioms is, in effect, $\{A_i\} \Rightarrow B$, where the A_i are sources of evidence and B is the supported plan. Inferences based on the axioms are uncertain, evidential inferences rather than deductive inferences because the existence of the conditional evidence, $\{A_i\}$, is not sufficient to guarantee the conclusion, B. Hypotheses are further compromised by the fact that inferences must often be based on partial evidence. That is, a plan of type B will typically be hypothesized based on evidence $\{A_j\}$ where $\{A_j\} \subset \{A_i\}$. Partial evidence may be used because the application requires incremental plan recognition for real-time recognition or because of computational considerations such as the complexity of evaluating the constraints to recognize valid inferences.

Given the hypothesis B based on the inference $\{A_j\} \Rightarrow B$, where the axiom is $\{A_i\} \Rightarrow B$ and $\{A_j\} \subset \{A_i\}$, there are the following potential classes of uncertainty sources in the hypothesis:

- The premise evidence may be uncertain–i.e., some A_k is uncertain because it is also based on uncertain evidence.
- There may be alternative interpretations for the evidence–i.e., for some $A_k in \{A_j\}$ the correct inference is $A_k \Rightarrow C$.
- The inference may be based on partial premise evidence–i.e., $\{A_j\} \neq \{A_i\}$.
- It may be uncertain whether the evidence satisfies the inference axiom (meets the constraints)—that is, it is uncertain whether $\{A_j\} \subset \{A_i\}$.
- The inference axioms may themselves be uncertain–that is, the correctness of $\{A_i\} \Rightarrow B$ is uncertain.

The actual instances of sources of uncertainty for each source of evidence in a domain fall into these classes. For instance, acoustic sensor data in the DVMT provides uncertain support for an environmental signal hypothesis because it may result from sensor malfunction. That is, a source of uncertainty in the evidential link between acoustic sensor data and a signal hypothesis is the potential alternative interpretation of the data as resulting from a sensor malfunction. Likewise, a signal hypothesis may support a group hypothesis (and so eventually a vehicle hypothesis), but it may also fail to support it because the signal actually is noise–the correct (alternative) interpretation of the signal is as noise rather than as part of a group. A group hypothesis may support a vehicle hypothesis, but until the complete set of groups for the vehicle support the vehicle hypothesis, it is uncertain.

Because of sensor resolution, the exact values of acoustic data parameters such as position and frequency are uncertain. This can result in uncertainty over whether a particular evidential inference is valid. Track hypotheses may represent actual vehicle tracks, but might also result from correlated noise or ghost data. While with likelihood of the track increases as the length is increased, there is always some degree of uncertainty from the fact that there is no set definition of a "complete" track. POISE never considered data errors, but a real-world intelligent assistant would have to be able to deal with user errors. User errors can be handled by making them a source of uncertainty in the data.

In a sophisticated plan recognition system with access to many sources of evidence, sources of uncertainty are used to drive the control process by selecting actions which result in evidence to resolve the important sources of uncertainty in the interpretations. For example, uncertainty due the possibility of sensor malfunction may be resolved (or at least partially resolved) by ordering diagnostics to be run on the sensor. Noise due to weather or terrain factors might be ruled out by checking weather conditions and the terrain. Of course, uncertainty due to partial premise evidence is one of the most prevalent sources of interpretation uncertainty and one which is not terribly amenable to sophisticated evidence gathering. However, because this source of uncertainty can be explicitly considered it may still be possible to improve evidence gathering: some incompleteness may not be very significant in the overall support of a hypothesis, the support provided by various (still incomplete) extensions may be stronger than others, and alternative, independent evidence may provide a less expensive resolution.

The sources of uncertainty in an evidential link may be represented in different ways depending on the type of uncertainty and what is most appropriate for future action. One way is to do it in a manner similar to the parallel-certainty inference approach (see chapter 3). Attached to the evidential links are symbolic statements which qualify the links with information about why they are uncertain. When uncertainty results from alternative interpretations, it may be more desirable to represent the alternative interpretations explicitly as hypotheses. Alternative interpretations are then connected with the links which specify that they are alternatives. This makes it easier to reason about pursuing evidence for the alternatives.

As was mentioned in section 5.2.1, hypothesis parameter values can also be uncertain. These uncertainties result from parameter uncertainties in the evidence, e.g., resolution uncertainty of acoustic sensors for frequency and position, and from incomplete evidence with which to define the parameters. Parameter uncertainty is represented by representing the potential range or set of supported values for the parameters. Copying of hypotheses when parameters are refined eliminates the need to understand the exact relationship between the evidence and parameter values since the values need never be revised. However, some understanding of the connections are required to evaluate the relative likelihood of the various values and to choose appropriate evidence to resolve the parameter uncertainties.

5.2.5 Actions

The control process eventually involves a decision about the next action for the interpretation system to take in order to generate additional evidence. These interpretation actions must be distinguished from the domain "actions" that the system is trying to interpret. Typically, actions involve the interpretation of existing data and hypotheses as evidence for higher-level hypotheses. For example, using acoustic sensor data to support a signal hypothesis or a using a vehicle hypothesis to support a track hypothesis. Such interpretation actions do not involve any interaction with the environment. However, in some domains, evidence-gathering actions may be extended to involve active processes to accumulate data. For example, a vehicle monitoring system might be able to make specific requests for data from particular sensors. It depends on the nature of the domain as to how active the interpretation system may be in gathering evidence. In some applications such as intelligent assistants, the ony "action" that the system may be able to take is to wait for more data to be generated and provided to it.

Evidence-gathering actions involve the generation of evidence for the interpretations. The system also takes other actions–control actions–as part of the process of making a decision about which domain action to take. These actions involve the expansion of control plans. For further information on control actions see section 5.2.8.

5.2.6 Relations

The plan recognition process involves the creation and consideration of interpretation hypotheses. These hypotheses are not always independent, but are related in that belief/disbelief in one affects the belief/disbelief in another. For example, a hypothesis can be an extension of another hypothesis. Evidence against an extension may or may not be evidence against the original hypothesis depending on whether it is really evidence against the plan instantiation or just against this particular extension of the instantiation. However, evidence against a hypothesis is always evidence against any extension hypotheses. Hypotheses may also be related as alternatives either through the interpretation of the same data or because they are based on inconsistent problem solving situations. In this case, evidence for a hypothesis is evidence against its alternative and vice versa. The existence of alternatives increases the level of uncertainty in a hypothesis.

POISE plan instantiation hypotheses may be related by being alternatives or by one being an extension of the other. Hypotheses are alternatives when they cover overlapping steps and they are not assumed to share the steps. Alternatives may also be recognized based on domain knowledge such as that in the "first-principles" knowledge mentioned above or through dependence on conflicting assumptions. For example, there may be knowledge that makes it unlikely that two plan instantiations could occur simultaneously or that one plan would be carried out soon after another. These relations must be explicitly recorded since they affect belief and uncertainty in the hypotheses and since evidence must be propagated over these relations. When one hypothesis represents an extension of another hypothesis with additional step data they are not alternatives in the sense given above since belief and evidence propagates differently between such hypotheses and since control strategies proceed differently. In particular, extensions enhance belief in a hypothesis, but control will generally proceed to explore the extensions.

Relations between hypotheses play an important role in the interpretation process in a vehicle monitoring system and yet a representation of these relations is absent from the current DVMT. In particular, the existence of a number of alternative extensions for a track hypothesis results in a good deal of uncertainty in the true track position which can be eliminated by exploration of the alternatives. For example, given a fairly good length track hypothesis with several possible extensions, our belief in the correctness of the track might be high without high belief in any one of the extensions due to the number of alternatives. The best control approach would be to try to reduce the uncertainty in the least expensive manner. This might suggest a breadth-first development of the alternatives rather than the depth-first approach which the DVMT would pursue. The DVMT would proceed depth-first because it has no knowledge of the alternatives and the uncertainty they cause. Instead, any single point extension which succeeds would be rated more highly than the unextended track despite the fact that a single point extension would do little to reduce the uncertainty.

5.2.7 System Goals

The primary responsibility of the control component is to try to satisfy the system goals. For example, the overall system goal may be to determine all and only those answers that can be supported by available data. This goal is what we might call an "uncertain goal" since its satisfaction will always be uncertain due to the inconclusive nature of evidence. Instead of saying these goals are *satisfied*, we will say that they are *sufficiently satisfied* or not as evidence is accumulated. What constitutes an acceptable answer is specified by answer and evidence constraints which express the desired system goals. The answer constraints specify the acceptable plan instances including plan types and parameter values while the evidence constraints specify acceptable hypothesis and parameter uncertainties. Thus the top-level control process must evaluate the suitability of the existing evidence in terms of the remaining uncertainties in order to determine whether the system constraints are met. If the goals are not yet satisfied, the control component must decide which uncertainties to resolve and how to resolve them.

Since the number of answers supported by the data is uncertain, an integral part of satisfying this system goal is the determination that all acceptable answers have been found. The explicit inclusion of this control goal is important because it affords a natural way of including certain types of evidence and uncertainties which are awkward to apply in the process of constructing answers. It also affords a metric for judging when processing is complete. This is crucial for situation assessment tasks like vehicle monitoring where processing of all data is impossible due to the huge volume of data being generated by many sensors.

Other applications may be handled by changing the overall system goals. An intelligent assistant application such as POISE would require that all data (from the user) be covered by an interpretation. Monitoring aircraft to prevent an attack would mean specifying that answers consist only of attack plans. In addition the system goals may also include resource constraints. For example on interpretation costs such as elapsed time, processing time, and safety concerns. These goals affect the choice of actions to take by influencing the desirability of goals and actions.

5.2.8 Control Plans

As was discussed in section 2.1, control decisions in AI systems are typically made in a two stage process which first identifies actions relevant to satisfying the goal and then chooses the best action to take next. In our approach, relevant actions are identified through a planning process based on the evidence gathering view of plan recognition. The selection of the *best* action is accomplished using a focusing scheme in parallel with the plan refinement process (see section 5.2.9). The result of the planning process is a control plan structure which represents the current state of the problem solving strategies and the potential alternative approaches to pursuing the system goals. Note that because we are *planning* how to recognize *plans*, the reader must be aware of the two different types of "plans" we refer to: *control plans* represent the strategies the system is using in order to interpret the data as *domain plan* instances.

Planning is a recursive process in which plans are expanded into a set of subgoals to be satisfied next, (lower-level) plans relevant to satisfying these subgoals are identified, and the new plans are then expanded to continue the refinement process. Plan refinement terminates when subgoals can be satisfied by "primitive plans" which correspond to evidence-generating actions (see section 5.2.5). Actions may fail or may succeed and return results. The outcome of an action is propagated up to the plan whose subgoal the action satisfies or fails to satisfy and the plan is updated appropriately. This might involve further propagation if the result means that the plan succeeds or fails or it might involve elaborating the plan over time to identify the next sub-goals which must be solved. Thus, planning and execution are interleaved: control plans are only elaborated to the point of choosing the next evidence-gathering action because the outcome of these actions is uncertain.

Planning makes use of explicit knowledge of the problem solving situation during the expansion of plans. For example, when determining what sources of data to use, the system actually determines what sources of data are available. Another key aspect of the problem solving situation is the *sources of uncertainty* in the evidence for the hypotheses. This knowledge is used when expanding plans to resolve the uncertainty in a hypothesis. Sources of uncertainty information is important because it is just these uncertainties which must be resolved by the plan recognition process. That is, the *purpose* of the actions taken by the plan recognition system is to resolve these uncertainties: each type of evidence has its characteristic uncertainties and each type of evidence can be used to resolve particular uncertainties.

Conventional planning is a top-down process which reflects the *desirability* of the problem solving methods. In domains where the outcome of actions is uncertain, the control process must also take into account the *feasibility* of an approach. In other words, a control strategy must integrate both data-directed and goal-directed components. We accomplish this because plan elaboration may be driven by the particulars of the current problem solving situation–e.g., the available data or the sources of uncertainty in the evidence. Note, however, that our control is not "opportunistic" in the sense of [26] as it does not key on data to suggest appropriate goals (independent of a particular problem solving plan). Typically, opportunistic control has relied on simplistic measures of data "goodness" to select appropriate actions. We believe that the "goodness" of data can only be judged relative to a particular goal which requires top-down development of context. However, changes in the problem solving situation may be used to signal the need for reconsidering focusing decisions (see section 5.2.9).

5.2.9 Focusing Heuristics

Since planning typically results in a number of ways in which the system goals can be pursued, an additional *focus-of-attention* process is used to select the single action to take next. This focusing process relies on the encoding of heuristic (meta) knowledge about the best ways to pursue goals and gather evidence–i.e., strategies for selecting the plan recognition strategies. Control plans are well-defined while the focusing heuristics embody the uncertain control knowledge of the system. Focusing is applied at each step in the refinement and elaboration of the control plans: following plan expansion or updating to select the goal(s) to pursue and following the identification of sub-plans relevant to satisfying each goal to select the plan to pursue.

We believe that hierarchical plan refinement based on sources of uncertainty knowledge provides the perfect structure for applying focusing heuristics. Conventional approaches to heuristic focusing tend to suffer from the problem of conflicting heuristics (see section 2.1). The specification of the heuristics is so general that multiple heuristics with conflicting suggestions apply at the same time. However, by indexing the heuristics according to the class of situations to which they apply, the conflicts can be eliminated. This is possible here because of the well-defined control plan structure. For example, in a DVMT-like vehicle monitoring system we could imagine the following two focusing heuristics regarding the selection of acoustic sensor data: prefer well-sensed (loud) signal data and prefer data at times with small numbers of clusters (potential sources). While these heuristics may seem to offer conflicting advice in many instances, once we understand the reasoning behind them we see that they actually apply in different situations based on the purpose of the data interpretation (see section 5.3).

Heuristic knowledge is indexed via a tree which is a static abstraction of the potential control plan graphs. The nodes in this tree then represent the control options and data characteristics about which the heuristics make ordering suggestions. The heuristics mentioned above do not conflict because though they both refer to acoustic sensor data characteristics, they do so on independent paths of the heuristics tree. The well-sensed data heuristic, for instance, is associated with the purpose of resolving hypothesis parameter uncertainties like position and frequency–that is, it is associated with acoustic sensor data in the path below the goal of resolving uncertainty in parameters. The cluster heuristic is associated with acoustic sensor data also, but on a different path in the tree associated with resolving hypothesis existence uncertainty.

The hierarchical definition and application of heuristic knowledge has a couple of additional important advantages over other systems which use a control plan approach–e.g., [26]. First, since focusing knowledge is applied *during* the refinement process rather than afterwards, it is possible to drastically limit the number of (evidence-gathering) actions which must examined. Contrast this with the Control BlackBoard scheme which requires that *all* potential actions be examine at rated on each control loop. Second, by limiting the scope of the focusing decisions, the use of straightforward, explicit focusing heuristics is facilitated. Again, constrast this with the Control Blackboard scheme which requires that all factors be taken into account at once in rating actions.

Of course in reality focusing is more complicated than the basic scenario outlined above. For one thing, focusing heuristics may not always be able to select a single path to pursue at each decision point. We handle this by allowing multiple paths to be expanded and including heuristic information in the skeletal plan structure that can focus between these multiple paths. Such information is represented by special relations between particular plans and/or goals where comparative decisions can be made. In addition, focusing decisions are preliminary and often based on certain assumptions about such things as the characteristics of the available data. As the plans are refined, it may become apparent that basis of one of the focusing decisions is not met and the system should backtrack, refocus and replan. Backtracking capability is provided by allowing focusing decisions to attach assumptions to the control path. Violation of an assumption causes the control process to reconsider the relevant focusing decision.

We view focusing as a task for expert-level heuristic reasoning both in selecting the uncertainty to pursue and in determining how to pursue it. Expert knowledge is based on a broad understanding of the domain. It involves knowledge of what evidence can be most easily gathered and how evidence affects the uncertainties in the various goals. Note, though, that control decisions must take into account the long-term effects of the actions rather than just the immediate effects. This is what makes the process so difficult and forces the system to rely on expert-level heuristic control knowledge. It is essentially hopeless to try to make an "objective" cost/benefit analysis because it's impossible to objectively assess the long-term effects of actions. For example, in a vehicle monitoring system we might reason that it makes little sense to accumulate evidence for a precise track position before accumulating evidence to resolve whether the track represents a vehicle of interest (a potential answer) or not. It may also be that much of the evidence for resolving uncertainty in a vehicle hypothesis will also be useful for resolving vehicle ID and position. From these considerations we could conclude that evidence to resolve uncertainty in whether a hypothesis represents an actual source should be gathered next– even if actions to resolve uncertainty in position might look better in some immediate sense.

5.3 An Example

In this section we will present a simple example which illustrates the basic operation of the sort of system we envision. A problem which is often used with the DVMT [22] will be examined. The acoustic sensor data for this example is shown in figure 5.2. Two potential, intersecting vehicle tracks are contained in the data. Data points 1a through 6a, track a, represent the "correct" track while those of track b represent "ghost" data. The plotted data points consist of clusters of signals where the point diameter corresponds to how well sensed the data in the cluster is. Clustering is common practice for facilitating the handling and analysis of acoustic sensor data [24] and this approach has been taken with the DVMT [22]. In fact, there will typically be various types of automatic, low-level processing which can be done on raw data to assist the interpretation process. For example, clustering tends to group signal data from a single source and hence can guide the selection of data for interpretation when the task is to confirm the existence of a given potential source. Additional information such as the number of clusters (and thus likely sources) at a given time and the potential connecting clusters might also be computed automatically for use

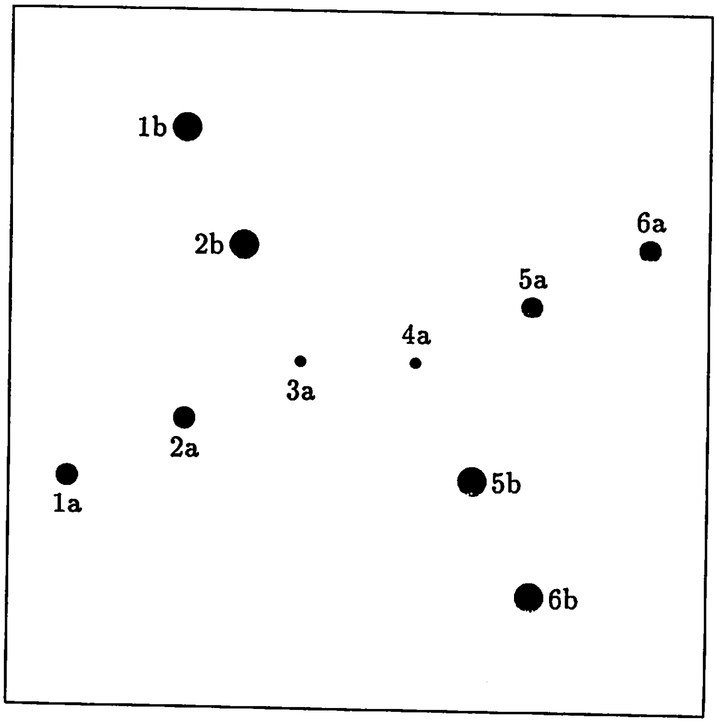

Figure 5.2: Acoustic Sensor Data Clusters

by the interpretation process.

The example will be examined from batch mode, i.e., the system has access to all of the data and must decide what data to focus on and interpret. Real-time interpretation would proceed in a similar fashion except that the system would have less flexibility in choosing from existing data and skipping around in time, but more flexibility in actively directing the data gathering process. The example will only make use of the acoustic sensor data in forming its interpretations, however, additional types of data will also be discussed.

The general goal for this problem is to find all (and only those actual) answers. This goal can be viewed as two subgoals: making sure that all potential answer have been found and then making sure that each potential answer is a correct answer. These subgoals are made specific in terms of a set of answer and evidence contraints. The answer contraints specify the connection between domain hypotheses and goal answers–i.e., they define which domain hypotheses represent answers. In this example, track hypotheses represent answer evidence. In other situations, answers may be required to be higher-level plan instantiations which describe the purpose of vehicle movements or they may be limited to particular subsets of tracks such as those for vehicles which pose a threat. Evidence contraints specify acceptable evidence for goal satisfaction. Here they state that for all space-time of interest, we want to be sufficiently sure that the space is covered with an answer or with a non-answer (to be sure *all* answers are found). Just what is meant by "sufficiently sure" is specified in terms of acceptable residual sources of uncertainty in the evidence.

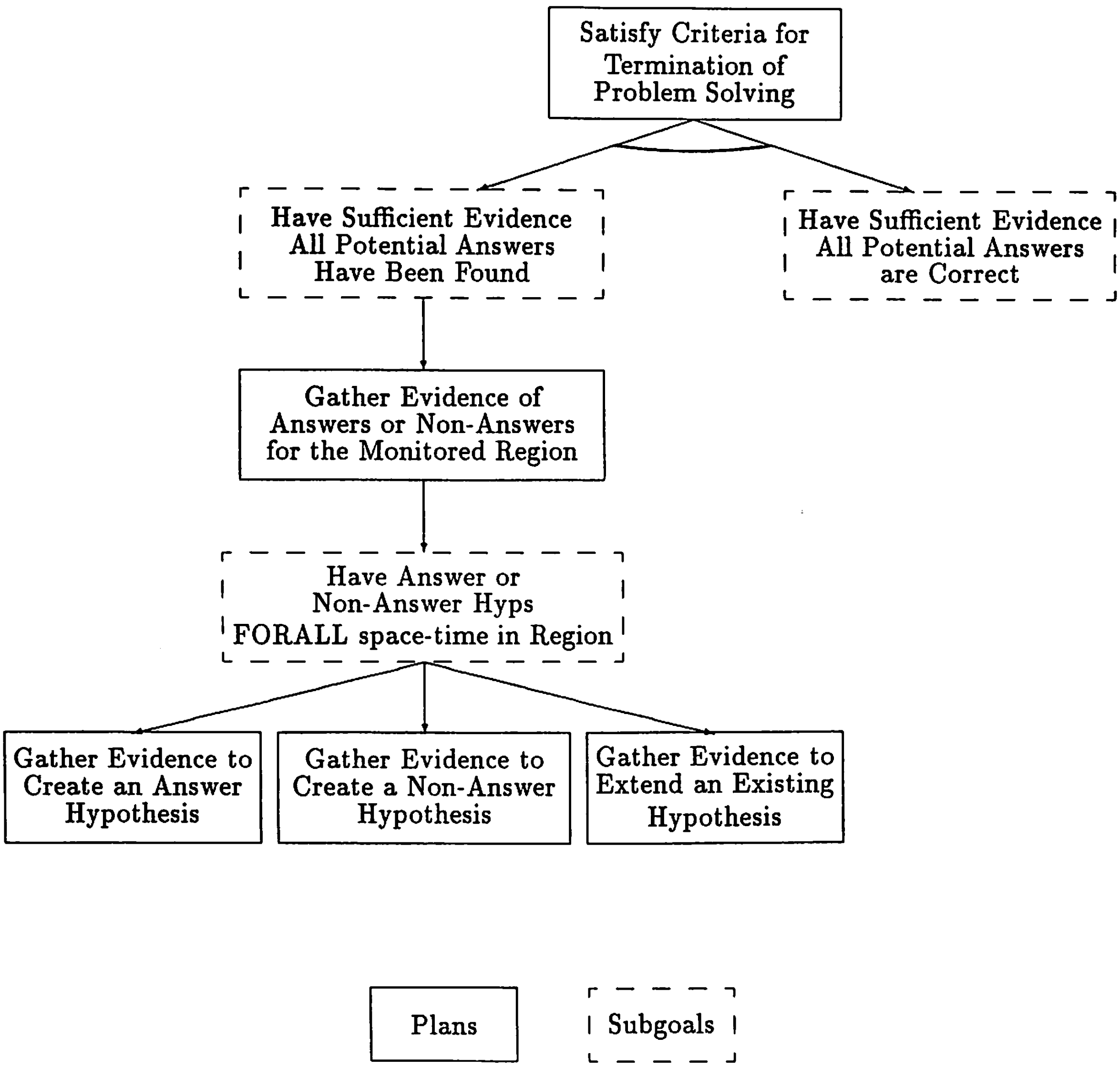

Figure 5.3: Initial refinement of the control plan.

The initial control plan refinement is represented in figure 5.3. Control plans are displayed as AND/OR trees composed of plans and subgoals which are distinguished by solid vs. dashed boxes. Each plan consists of one or more subgoals which must be satisfied to complete the plan. For each subgoal, there may be one or more plans which can be used to satisfy the subgoal. Focusing decisions are made in parallel with plan refinement whenever there are multiple active subgoals for a selected plan or when there are multiple plans applicable to a selected subgoal. In general, only active subgoals–those subgoals currently needing to be satisfied–are displayed in the figures. However, in some cases we have included subgoals which are not yet active in order to better present the flow of steps in a plan. These subgoals are marked by arrows on the plan links rather than the normal solid line AND node notation. The arrows indicate the sequencing of the subgoals.

Control planning begins with the expansion of the general plan for meeting the problem solving criteria. This results in the creation of two subgoals which comprise the overall problem solving goal for the system. At this point, there is no evidence as to whether all potential answers have been found so this goal is unstaisfied. On the other hand, since no potential answer hypotheses have yet been created, the subgoal of having evidence that potential answers are correct *is* currently satisfied. This subgoal may alternate between being satisfied and unsatisfied several times during the problem solving process, however, the top-level control plan is only complete when both of its subgoals are simultaneously satisfied.

Focusing on the unsatisfied subgoal, the system locates plans that are applicable to satisfying that subgoal. In this case, there is just one such plan. This plan develops evidence to determine whether an answer or a non-answer covers each point in the space-time region being monitored. Expanding this plan results in a "compound" subgoal which represents the goal of having answer or non-answer evidence for the entire region. By a compound subgoal, we mean that this goal represents a condition which may be satisfied in multiple steps where the number of steps is unknown. As evidence is created in various regions, this subgoal is *partially satisfied* and is updated to reflect the region in which the conditions remain to be met.

Proceeding with the planning process, there are three plans which can be used to generate answer or non-answer evidence: Create a (new) Answer Hypothesis, Create a (new) Non-Answer Hypothesis, or Extend an Existing Answer Hypothesis. Focusing will determine which of these plans will be further refined. The focusing heuristics may select just one plan for further expansion or they may select more than one. At this level in the control plans it seems likely that more specific information about the characteristics of the available data would be needed to select the best choice. In this case, more than one of the options would be expanded to a level at which the focusing heuristics could make a decision based upon the additional information which had been generated. Note also, that

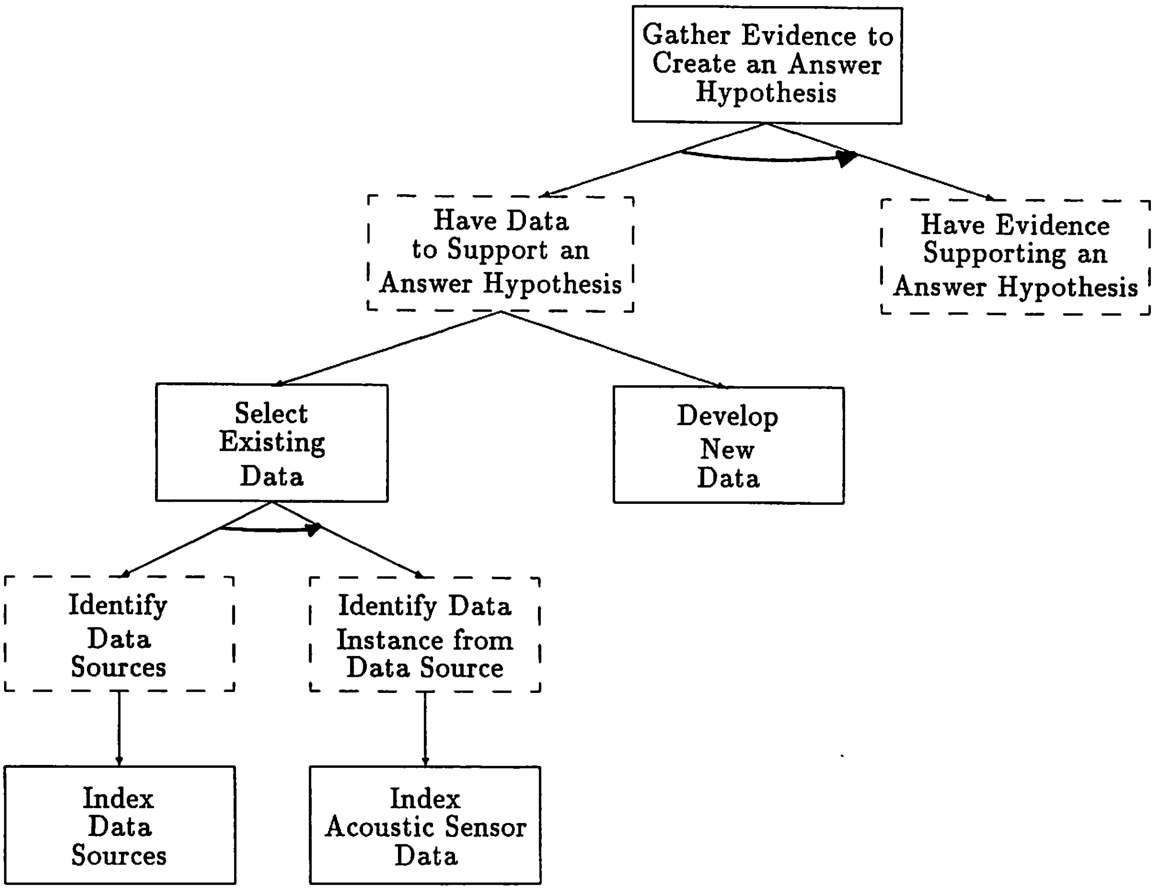

Figure 5.4: Partial plan refinement to create answer evidence.

the plan for extending an existing answer would fail if it were chosen for further expansion because there are no existing answers. If this were the only plan focused on, the failure would cause backtracking and additional focusing.

For the purpose of illustrating the control process, we will concentrate on developing answer evidence at this point in the example. Figure 5.4 represents the further refinement of this plan. The plan to create an anwer hypothesis is a two step plan. Both steps in the sequence are represented in the figure even though only the first of the subgoals is currently active. The first step in the plan is to Have Data to Support an Answer Hypothesis. There are two methods for obtaining data: selecting from among the existing data that has been collected by the sensors or developing new data by directing a sensor to collect additional data. We focus here on the Select Existing Data plan because we are not assuming control of the sensors and because at this point in the processing there is no information which would help us to direct data acquisition.

The Select Existing Data plan is also a two step plan which has both of its subgoal steps displayed. The first step is to select the data source to be used. There is exactly one

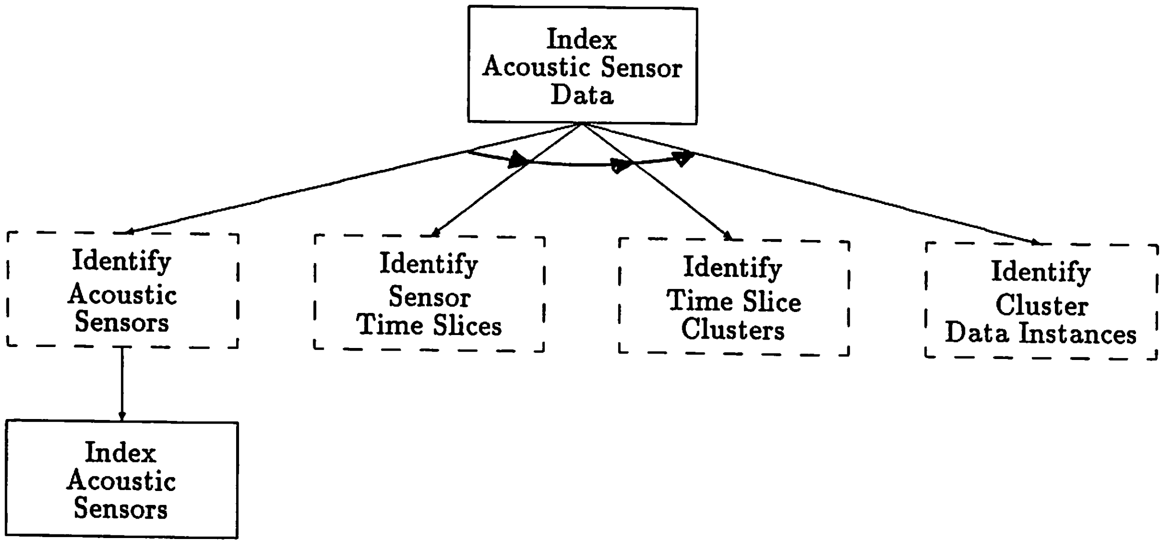

Figure 5.5: Partial plan refinement to select data.

plan for satisfying this subgoal and it is a primitive plan. Being a prmitive plan means that the plan is not going to be exanded, but that it represents an action which can be taken to generate information. In this case, the primitive plan Index Data Sources returns a list of the data sources that are actually available for use–i.e., that the system actually has data from. Focusing is applied to the result of the plan action prior to binding of the variables in the succeeding step. This may result in multiple active subgoals. For example, we might choose to pursue further both acoustic data and radar data. Thus there would be subgoals for the plan of identifying both an acoustic data instance and a radar instance.

Focusing on the only data we are assuming is available, the acoustic sensor data, refinement to select data involves indexing into the data through a number of stages–see figure 5.5. For acoustic sensor data, indexing first involves the selection of a particular sensor, follwed by the time slice of data from that sensor, follwed by the cluster in that time slice, and finally the data instance in that cluster. For each subgoal step there is a corresponding primitive plan whose execution actually determines the avaiable data. Focusing is applied after each primitive plan execution and so limits the portion of the data which is actually examined.

Focusing here provides a good example of using context to disambiguate meta-level heuristic information. In the POISE focusing framework the applicable heuristic knowledge would have to be represented as two rules: prefer data at times with fewer clusters and prefer better sensed (louder) signals. However, these heuristic rules conflict in this case since there are only single clusters at times 3 and 4, but this data is more poorly sensed than data at the other times. By indexing our knowledge to the purpose of the actions, this conflict is avoided in our framework: the two rules are associated with different control plan

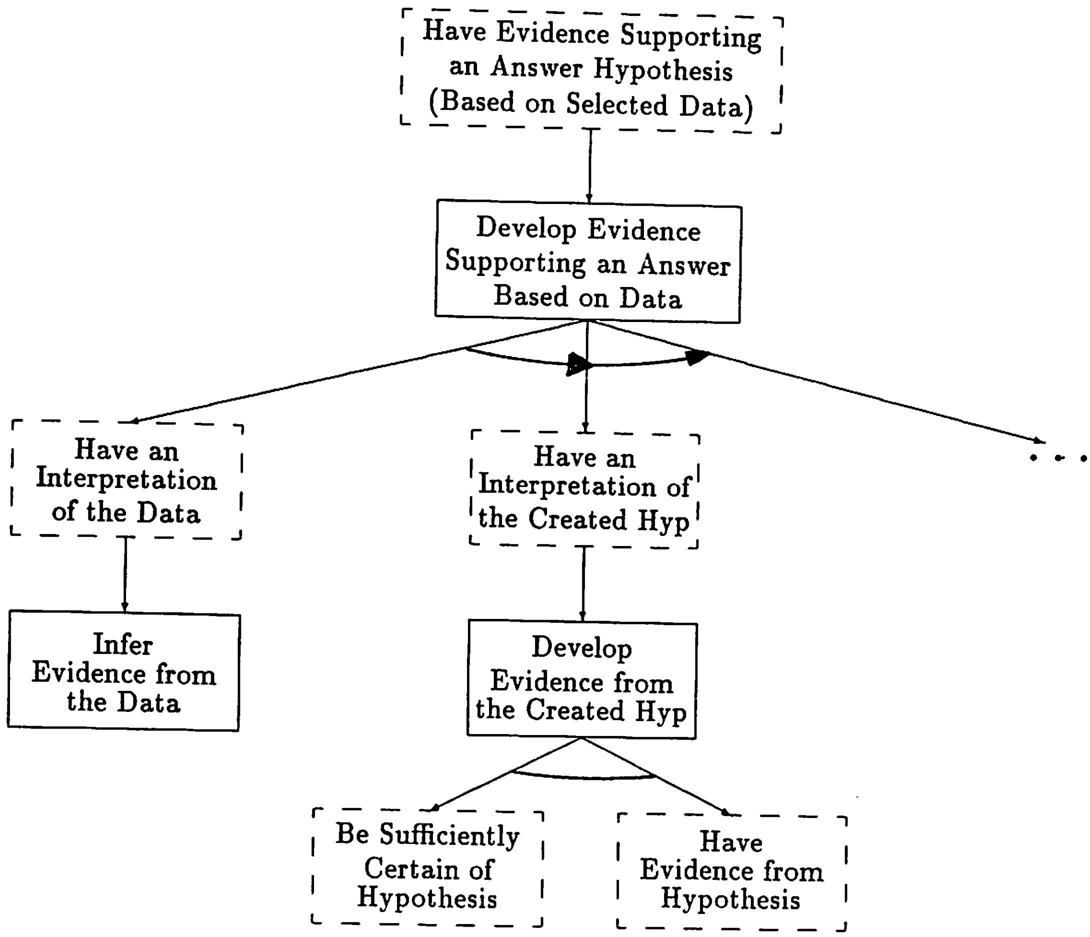

Figure 5.6: Control plan refinement following data selection.

contexts. The cluster density may be specified as the primary metric when trying to develop evidence of potential answers while the signal strength is the primary metric for resolving vehicle ID and position uncertainties. This reflects the purpose of the original rules since data sets which contain the fewest numbers of clusters should contain less potential sources to be resolved. On the other hand, if the goal were to resolve uncertainty in a vehicle position or type, then data sets containing clusters with well sensed (loud) signals might be preferred. The reasoning here being that louder signals tend to be more accurately sensed, but are not inherently more likely to represent sources of interest–consider, for instance, high-flying aircraft or battlefield conditions. In any case, the focusing heuristics result in the selection of clusters at either 3a or 4a. The choice of 3a over 4a and the choice of a signal in the cluster may be made randomly if there are no distinguishing characteristics.

Once data has been selected, the Select Existing Data plan is complete and thus the Have Data to Support an Answer Hypothesis subgoal is satisfied. This causes the Create Answer Hypothesis plan to be updated so that the next subgoal, Have Evidence Supporting

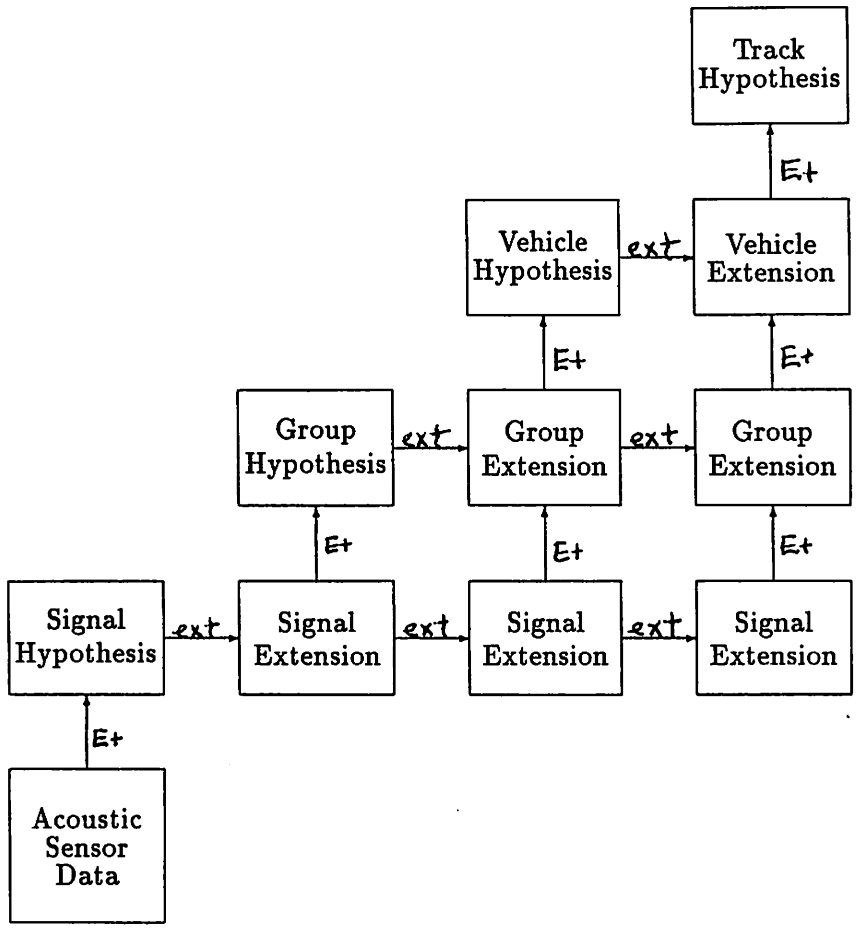

Figure 5.7: Evidential support for track hypothesis.

an Answer Hypothesis Based on the Data, is now active. There is one plan applicable to satisfying this subgoal as shown in figure 5.6. The Develop Evidence Supporting an Answer Based on Data plan is a multi-step plan where the number of steps is indeterminate–depending on the number of evidential inferences which must be made to create an answer-level hypothesis from the data. Acoustic signal data, for instance, must be abstracted through the signal, group, and vehicle levels before being used to support a track hypothesis while radar data immediately supports a vehicle level hypothesis. Again, the first couple of steps of the plan are displayed in the figure.

The result of the Infer Evidence from the Data primitive plan is the creation of a signal-level hypothesis connected by an evidence link to the supporting data. This is the lowest level evidential link in figure 5.7 The hypothesis is uncertain–that is, it is uncertain whether the signal hypothesis represents an actual environmental signal–because the data may be the result of sensor malfunction. Because of this uncertain connection between data and what it represents, we choose to add an additional abstraction level to the DVMT hierarchy: the acoustic sensor data level. This level represents the data as received from the sensor and as such contains no uncertainties–the data is whatever it is. What is uncertain is just what the data represents about the environment. This uncertainty is represented by attaching appropriate symbolic tags to the evidence link which stand for the sources of uncertainty in the evidence (these are not shown in figure 5.7). Acoustic sensor data support for a signal hypothesis includes the sources of uncertainty: sensor-noise and sensor-ghost. Uncertainty in the parameter values of the signal hypothesis is represented through the use of range values. For example, the sensor resolution characteristics tell us that the actual frequency of the environmental signal is within one frequency class of the sensor data so the signal hypothesis frequency is $f \pm 1$ where f is the acoustic data signal frequency class.

The creation of a signal hypothesis satisfies the first step of the plan for developing an answer from the data. This causes the plan to be updated so that the newly created hypothesis can be used to create further evidence–see figure 5.6. Of course, in a real-time vehicle monitoring system other sources of evidence may become available in the intervening period and would be evaluated relative to pursuing the new hypothesis. Thus while the example shows the control plans functioning in a completely top-down manner, it is important to remember that there is also a bottom-up component to the overall control process (see section 5.2.9). Data-directed control is accomplished through the use of conditional focusing decisions. That is, when focusing decisions are made, the system may attach conditions to the decisions which would require the decisions to be reconsidered. For example, while the system may choose to pursue existing acoustic sensor data rather than waiting for radar data, once the radar data is avaiable this decision may be reconsidered.

Expanding the Develop Evidence from the Hypothesis plan results in two subgoals: one to create further evidence from the newly created hypothesis and one to resolve uncertainty in that hypothesis. This plan is interesting because it is considered complete when the Have Evidence from Hypothesis subgoal is satisfied even if the Be Sufficiently Certain of Hypothesis subgoal isn't. This sort of plan allows the focusing system to reason about whether it should immediately pursue the hypothesis to create an answer or whether the hypothesis is so uncertain that further evidence may need to be gathered for it first. This decision would depend on the sources of uncertainty in the hypothesis and other factors such as the available processing time. Because of the explicit recording of evidence and its associated sources of uncertainty, the control process can consider exactly why hypotheses are uncertain and what actions are appropriate to resolve the uncertainty. This is one sense in which the sources of uncertainty information is used to drive the control process. For example, we could imagine an action of running diagnostics on the sensor to determine the possibility that the sensor is malfunctioning to resolve the sensor-noise source of uncertainty. On the other hand, the focusing heuristics may suggest that it is more appropriate to pursue this hypothesis further to determine if it ca even support an answer before accumulating additional evidence for it.

Eventually, a group hypothesis will be created from the signal hypothesis, then a vehicle hypothesis from the group, and finally a track hypothesis. The resulting levels of evidential support are represented in figure 5.7. Explicit links are maintained between representations of data or hypotheses and the hypotheses this evidence supports. Each hypothesis in fact consists of a set of *extensions* which represent different versions of the hypothesis as additional evidential inferences are made which refine the hypothesis parameters. Thus each extension represents the version of the hypothesis under a different set of evidential inferences. Note, also, that the figure assumes direct abstraction of the data to support a track without any intermediate uncertainty resolution as discussed above.

Each evidential link has associated with it information about the type of the inference (i.e., the role the evidence plays in support of the hypothesis) and the sources of uncertainty in the inference. Since uncertainty in the track hypothesis results from uncertainty in the evidence supporting the track, the sources of uncertainty information associated with the evidential links explicitly represents the sources of uncertainty in the track hypothesis. Thus, "proving" the track hypothesis correct means planning how to resolve the uncertainty in the track hypothesis by identifying actions which can resolve these sources of uncertainty. The sources of uncertainty in the track evidence result from the incomplete set of vehicle hypotheses supporting the track and from the uncertainty in the vehicle evidence itself.

The track hypothesis satisfies part of the problem solving goal in that it represents the creation of a potential answer. This changes the status of the Have Sufficient Evidence All Potential Answers are Correct subgoal from satisfied to unsatisfied. Additional evidence must still be developed to satisfy the evidence constraints by sufficiently "proving" the hypothesis and by sufficiently refining the hypothesis' parameters such as position and vehicle type. The high-level portion of the control plan as it exists following the creation of a potential answer hypothesis is shown in figure 5.8. Once again, the sources of uncertainty information is used in the plan refinement process to eleaborate the control choices. The track hypothesis is uncertain because of the incomplete or partial evidence that has been collected and because of the uncertainties in that evidence–these are the sources of uncertainty in the track hypothesis. The partial vehicle evidence uncertainty is resolved by generating additional vehicle hypotheses to support the track–i.e., tracking the vehicle. Resolution of the uncertainty in the evidence would involve examination of the sources of uncertainty in that evidence.

In tracking, constraints can be used to limit the data to be examined based on frequency class and position. The planning process makes use of the plan constraints to refine the plans in order to limit the actions and data which are deemed relevant. At this point, the

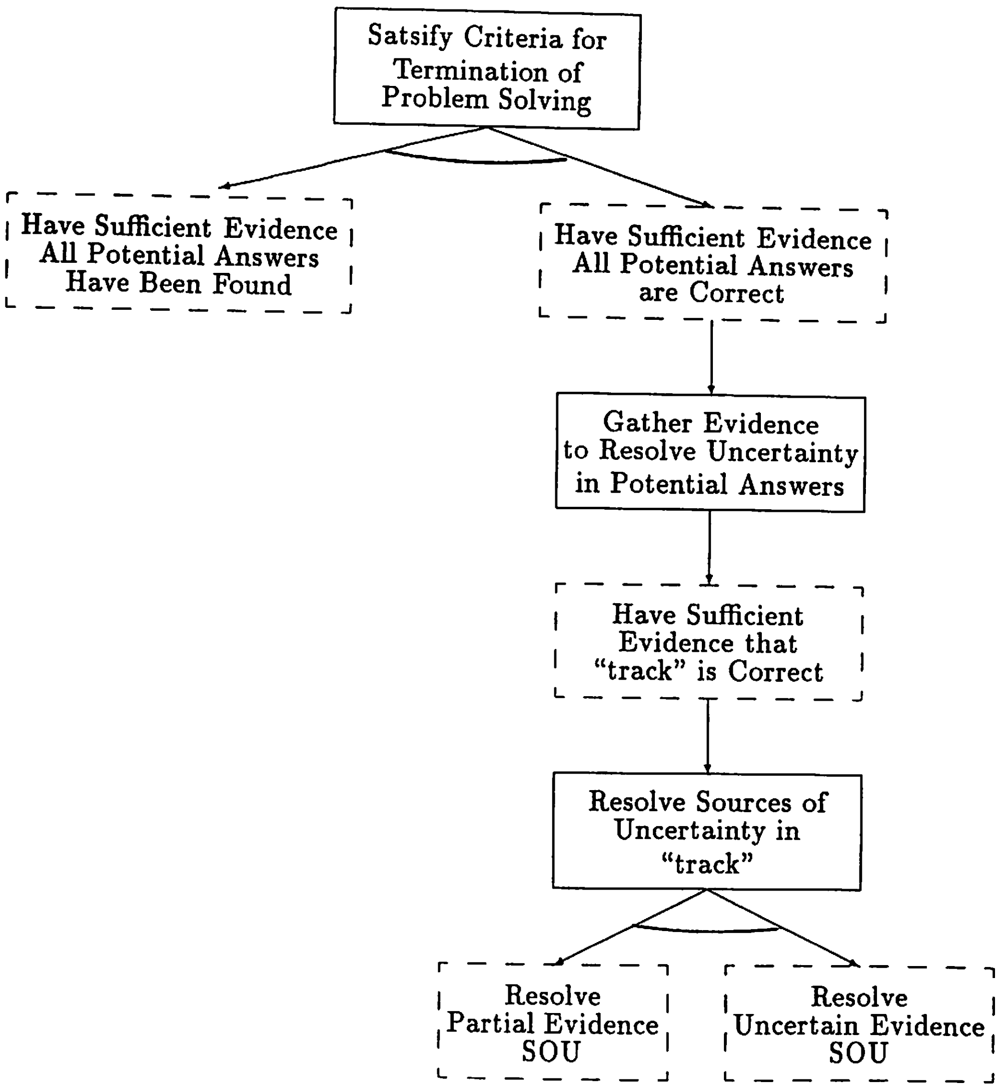

Figure 5.8: Partial plan refinement following answer hypothesis creation.

track could be extended using time 2 data or time 4 data. Since only a single cluster is potentially applicable at time 4 the focusing heuristics would probably select this as the data to use for the next extension of the track. This process would continue until the track is deemed to have sufficient supporting evidence to meet the evidential constraints. This may or may not require the construction of a "complete" track. If it does not and a complete track is required of valid answers then tracking would continue–driven by the valid answer problem solving goal. In this example, tracking is quite straightforward because the focusing process directed the crucial time 3 and time 4 data to be used to construct the basis of the track. This segment is incompatible with the "b" track extensions because of kinematic constraints. Thus, there is never the possibility of multiple, alternative extensions here. In general, however, there will be alternative extensions for hypotheses. Alternatives are identified by "alternative links" set up between the extension hypotheses. Alterntive relations represent a general type of negative evidential relationship which can exist between any two hypotheses. That is, evidence for one of the alternatives is considered as evidence against the alternative and vice versa. The addition of this negative evidence causes conflict uncertainty which must be resolved by resolving uncertainty in the alternative extensions.

Even when only acoustic sensor data is used to form the interpretations, it need not always be used in the same way. For example, as additional evidence is gathered for the track, confidence in the correctness of the track and the vehicle type increases. At some point, it may become reasonable to generate less certain, but also less expensive evidence by abstracting directly from a data cluster to a vehicle hypothesis (using a different evidence-generating KS). This evidence would contain some rather significant sources of uncertainty were it to be considered as a solitary inference because the consistency of the frequency information would not have been established. However, in the overall track hypothesis these sources of uncertainty may be of little consequence and are offset by the advantages of faster processing. This same sort of option occurs with other sources of evidence such as radar. Resolution can be controlled by varying the power and scan speed with the tradeoffs being the risk of missing certain targets and limited coverage area.

To fully meet the system goal of creating all and only the correct hypotheses, the system must also investigate space-time not covered by the "a" track in order to develop evidence that there are no additional answers. Unlike real acoustic sensor data, there is little "noise" in this simple example. In general, acoustic sensor data would probably not be the best possible source of negative answer evidence because it tends to be relatively noisy. To produce negative evidence, acoustic sensor data is interpreted differently than when producing potential answer evidence. An entire time slice is interpreted at once with the absence of any data clusters producing direct negative evidence for vehicles and clusters producing direct (but very uncertain) evidence of vehicles. Any vehicle hypotheses

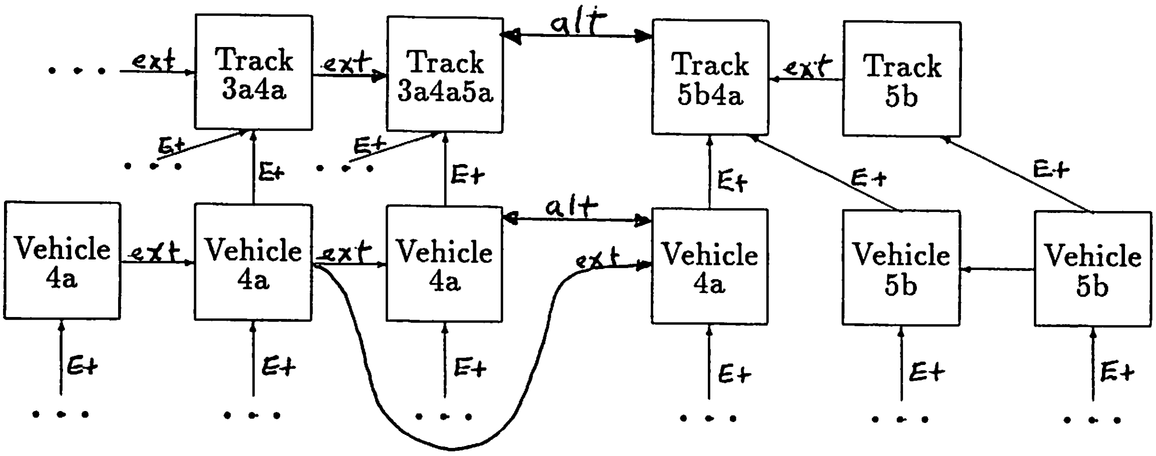

Figure 5.9: Alternative tracks due to shared vehicle hypothesis.

produced would be a source of uncertainty in the system goals since they would fail to meet the evidential constraints. The system must resolve these uncertainties by gathering additional evidence to determine whether this evidence represents an answer or not.

In the example, this means that the system would have to examine the "b" track data to produce sufficient evidence that this data does not support an answer track. The exact negative evidence generated depends upon the order in which the system pursues the alternative interpretations of this data. What is clear is that any tracks involving "b" data would have to include "a" data from times 3 and 4 in order to be extended and completed. However, attempting to extend a "b" track into these areas causes the system to recognize the "b" and "a" tracks as alternatives because they share vehicle evidence. This results in an alternatives evidential relationship being set up between the "a" and "b" track hypotheses. Figure 5.9 shows such a conflict resulting from one potential track construction scenario. The addition of this conflicting negative evidence to the tracks results in an additional source of uncertainty which must be resolved by resolving the other sources of uncertainty in the hypotheses. Since the "a" track is well supported, this conflict causes little additional uncertainty and so it is probably best to resolve the uncertainty in the "b" tracks by trying to extend them. However, there are no possible track extensions for the "b" tracks here because the constraints on the vehicle kinematics prohibit these tracks from containing both 3a and 4a data. This extension failure is an additional source of strong negative evidence for the "b" tracks. Of course, this negative evidence is still somewhat uncertain since the possibility of missing data at times 3 and 4 is a source of uncertainty for extension failure evidence. Should the combination of negative evidence from the "a" track and from the extension failure not be deemed sufficient proof against the "b" tracks then additional actions would be needed to try to resolve the uncertainty. These actions would involve pursuing the sources of uncertainty in both the positive and negative evidence for the "b" segments. For example, by gathering additional evidence for the "a" track, resolving whether the "b" segments fit the criteria for sensor ghosts of the "a" track, and postulating missing data.

The control reasoning in this example can be more involved than in the standard DVMT since it can be made to rely heavily on domain knowledge about evidence and uncertainty. This is exactly the point of this work: basing control on a process of accumulating explicit, symbolic evidence to manage and resolve uncertainties makes it possible to reason in more detail about control decisions. In this example, heuristic control knowledge greatly limits the amount of work that is done by focusing the problem solving process on the data which is the most promising for meeting the goals. The DVMT spends a great deal of effort building and re-building track segments for the "b" data without recognizing the crucial role of data at times 3 and 4 and without recognizing the redundancy of its actions (this problem has been addressed in a different manner in recent work [22]). Our approach also provides a better basis for understanding the solution. This is a particularly problematic example for the DVMT since the "solution" track data is weaker than the "ghost" track data. In the DVMT, constraints on vehicle kinematics are used to eliminate the ghost track from consideration. Since we wish to consider the possibility of mis-sensed and missing data the problem becomes more difficult. Postulating missing data at time 3 and/or 4, it is possible to complete the "ghost" track. While this alternative could presumably be eliminated from consideration by "tuning" the procedure for weighing evidence, in real problems there would be other sources of evidence to resolve this uncertainty. For example, it seems extremely unlikely for there to be *no* sign of the actual vehicle at these times and complete signal data for the incorrect track. These would be considered as strong sources of evidence in resolving the missing data track extension alternative. In any case, by maintaining an explicit record of evidence, uncertainties, and assumptions, our system provides a basis for understanding the remaining sources of uncertainty in the interpretation.

Chapter 6

Conclusion

This document describes a plan recognition framework which addresses the limitations of existing plan recognition systems. A sophisticated, general-purpose plan recognition system must be able to meet the requirements laid out in section 4.4. Recognizing that an evidence-gathering approach to plan recognition allows these problems to be solved in a natural and coherent manner is one important result of this work. The key contribution of general interest, however, is the development of a framework within which control decisions can be made by *explicit* reasoning. Conventional AI systems have tended to rely on the use of globally-valid numeric ratings to make decisions. The problem with this approach is that much of the reasoning the system is doing is implicit in the ratings and so unavailable for further consideration. This makes it difficult for the decisions to be dynamic and context dependent.

In order to make the most appropriate control decisions possible, a system must be able to reason explicitly about its alternatives and its goals. The two key aspects of our framework which make this possible are: the explicit recording of sources of uncertainty information and the use of control plans with integrated focusing. Since the purpose of plan recognition is to gather evidence to resolve interpretation uncertainty, explicit information about the *sources of uncertainty* in the evidence allows the system to understand why it is taking actions. Control plans are elaborated based on the evidence uncertainties and the methods for resolving these uncertainties. The plans then represent the potential approaches for meeting the problem solving goals. Because focusing decisions are made in parallel with the hierarchical refinement of the control plans, the number of factors which must be considered in each decision is limited. In addition, the control plan framework provides context information so the system understands the purpose of each decision. These developments facilitate explicit reasoning about control since the alternatives can now be directly compared and selected.

Bibliography

[1] Adams, J. Barclay, "Probabilistic Reasoning and Certainty Factors," in Bruce Buchanan and Edward Shortliffe, editors, *Rule-Based Expert Systems*, Addison-Wesley, Reading, 1984, 263–271.

[2] Allen, James, "Recognzing Intentions From Natural Language Utterances," from *Computational Models of Discourse*, Michael Brady and Robert C. Berwick, editors, The MIT Press, 1983, 107–164.

[3] Azarewicz, Jerome, Glenn Fala, Ralph Fink, and Christof Heithecker, "Plan Recognition for Airborne Tactical Decision Making," submitted to AAAI-86, 1986.

[4] Barton, George, Jr., *A Multiple-Context Equality-Based Reasoning System*, Technical Report AI-TR-715, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1983.

[5] Buchanan, Bruce and Edward Shortliffe, editors, *Rule-Based Expert Systems*, Addison-Wesley, 1984.

[6] Carver, Norman, Victor Lesser, and Daniel McCue, "Focusing in Plan Recognition," *Proceedings of AAAI-84*, 1984, 42–48.

[7] Charniak, Eugene and Drew McDermott, *Introduction to Artificial Intelligence*, Addison-Wesley, 1985.

[8] Clancey, Willian, "Extensions to Rules for Explanation and Tutoring," in Bruce Buchanan and Edward Shortliffe, editors, *Rule-Based Expert Systems*, Addison-Wesley, Reading, 1984, 531–568.

[9] Cohen, Paul and Milton Grinberg, "A Theory of Heuristic Reasoning About Uncertainty," *The AI Magazine*, Summer 1983, 17–24.

[10] Cohen, Paul, *Heuristic Reasoning About Uncertainty: An Artificial Intelligence Approach*, Pitman, 1985.

[11] Cohen, Paul, David Day, Jeff Delisio, Mike Greenberg, Rick Kjeldson, Dan Suthers, and Paul Berman, "Management of Uncertainty in Medicine," *International Journal of Approximate Reasoning*, Spring 1987.

[12] Corkill, Daniel, *A Framework for Organizational Self-Design in Distributed Problem Solving Networks*, Ph.D. Thesis and Technical Report 82-33, Department of Computer and Information Science, University of Massachusetts, 1982.

[13] Davis, Randall, "Meta-Rules: Reasoning about Control," *Artificial Intelligence,* 15 (1980), 179–222.

[14] Davis, Randall and Douglas Lenat, *Knowledge-Based Systems in Artificial Intelligence,* McGraw-Hill, New York, 1982.

[15] Davis, Randall and B. Buchanan, "Meta-Level Knowledge: Overview and Applications," in Bruce Buchanan and Edward Shortliffe, editors, *Rule-Based Expert Systems*, Addison-Wesley, Reading, 1984, 507–530.

[16] deKleer, Johann, "Choices Without Backtracking," *Proceedings of AAAI-84,* 1984, 79–85.

[17] Doyle, Jon, "A Truth Maintenance System," *Artificial Intelligence,* 12 (1979), 231–272.

[18] Doyle, Jon, *A Model for Deliberation, Action, and Introspection,* Technical Report AI-TR-581, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1980.

[19] Doyle, Jon and Philip London, "A Selected Descriptor-Indexed Bibliography to the Literature on Belief Revision," *SIGART Newsletter,* No. 71, 1980, 7–23.

[20] Doyle, Jon, *Some Theories of Reasoned Assumptions,* Technical Report CMU-CS-83-125, Department of Computer Science, Carnegie-Mellon University, 1983.

[21] Doyle, Jon, "Methodological Simplicity in Expert System Construction: The Case of Judgements and Reasoned Assumptions," *The AI Magazine,* Summer 1983, 39–43.

[22] Durfee, Edmund and Victor Lesser, "Incremental Planning to Control a Blackboard-Based Problem Solver," submitted to *AAAI-86*, 1986.

[23] Garvey, Alan, Craig Cornelius, and Barbara Hayes-Roth, "Computational Costs versus Benefits of Control Reasoning," *Proceedings of AAAI-87*, 1987, 110–115.

[24] Green, Peter, "Distributed Acoutic Surveillance and Tracking," *Proceedings of Workshop on Distributed Sensor Networks*, Lincoln Laboratory, Massachusetts Institute of Technology, 1982.

[25] Gordon, Jean and Edward Shortliffe, "The Dempster-Shafer Theory of EVidence," in Bruce Buchanan and Edward Shortliffe, editors, *Rule-Based Expert Systems*, Addison-Wesley, Reading, 1984, 272–292.

[26] Hayes-Roth, Barbara, "A Blackboard Architecture for Control," *Artificial Intelligence*, 26, 1985, 251–321.

[27] Hayes-Roth, Barbara, et. al., "PROTEAN: Deriving Protein Structure from Constraints," *Proceedings of AAAI-86*, 1986, 904–909.

[28] Hayes-Roth, Barbara, et. al., *A Layered Environment for Reasoning about Action*, Technical Report KSL 86-38, Knowledge Systems Laboratory, Computer Science Department, Stanford University, 1987.

[29] Hayes-Roth, F. and R. Wesson, "Distributed Intelligence for Situation Assessment," *Proceedings of Workshop on Distributed Sensor Nets*, Carnegie-Mellon University, 1978.

[30] Hudlicka, Eva and Victor Lesser, "Meta-Level Control Through Fault Detection and Diagnosis," *Proceedings of AAAI-84*, 1984, 153–161.

[31] Huff, Karen, and Victor Lesser, *Knowledge-Based Command Understanding*, Technical Report 82-6, Department of Computer and Information Science, University of Massachusetts, 1982.

[32] Johnson, M. Vaughn, Jr. and Barbara Hayes-Roth, "Integrating Diverse Reasoning Methods in the BB1 Blackboard Control Architecture," *Proceedings of AAAI-87*, 1987, 30–35.

[33] Kautz, Henry and James Allen, "Generalized Plan Recognition," *Proceedings of AAAI-86*, 1986, 32–37.

[34] Kautz, Henry, *A Formal Theory of Plan Recognition*, Technical Report TR 215, Department of Computer Science, University of Rochester, 1987.

[35] Lacoss, R., "Sensor Tutorial," *Proceedings of Workshop on Distributed Sensor Nets*, Carnegie-Mellon University, 1978.

[36] Lenat, Douglas, Randall Davis, Jon Doyle, Michael Genesereth, Ira Goldstein, and Howard Schrobe, "Reasoning about Reasoning," in F. Hayes-Roth, D. Waterman, and D. Lenat, editors, *Building Expert Systems*, Addison-Wesley, Reading, MA, 1983.

[37] Lesser, Victor and Daniel Corkill, "The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks," *The AI Magazine*, Fall, 1983, 15–33.

[38] Litman, Diane and James Allen, *A Plan Recognition Model for Subdialogues in Conversations*, Technical Report TR-141, Department of Computer Science, University of Rochester, 1984.

[39] London, Philip, *Dependency Networks as a Representation for Modeling in General Problem Solvers*, Technical Report TR-698, Department of Computer Science, University of Maryland, 1978.

[40] Lowrance, John and Thomas Garvey, *Evidential Reasoning: An Implementation for Multisensor Integration*, Technical Note 307, SRI International, 1983.

[41] McCallester, David, *The Use of Equality in Deduction and Knowledge Representation,* Technical Report AI-TR-550, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1980.

[42] McCallester, David, *An Outlook on Truth Maintenance,* AI Memo No. 551, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1980.

[43] Quine, W.V. and J.S. Ullian, *The Web of Belief,* Random House: New York, 1970.

[44] Rich, Elaine, *Artificial Intelligence,* McGraw-Hill, New York, 1983.

[45] Schmidt, Charles, and N.S. Sridharan, "Plan Recognition using a Hypothesize and Revise Paradigm," *Proceedings of IJCAI-77*, 1977, 480–486.

[46] Schmidt, C.F., N.S. Sridharan, and J.L. Goodson, "The Plan Recognition Problem: An Intersection of Psychology and Artificial Intelligence," *Artificial Intelligence*, 11 (1978), 45–83.

[47] Shortliffe, Edward and Bruce Buchanan, "A Model of Inexact Reasoning in Medicine," in Bruce Buchanan and Edward Shortliffe, editors, *Rule-Based Expert Systems*, Addison-Wesley, Reading, 1984, 233–262.

[48] Smith, Reid, Howard Winston, Tom Mitchell, and Bruce Buchanan, *Representation and Use of Explicit Justifications,* Knowledge Systems Laboratory Working Paper No. KSL 85-3, Computer Science Department, Stanford University, 1985.

[49] "Planning and Meta-Planning (MOLGEN: Part 2)," *Artificial Intelligence*, 16, 1981, 141-170.

[50] Sullivan, Michael and Paul Cohen, "An Endorsement-Based Plan Recognition Program," *Proceedings of IJCAI-85*, 1985, 475–479.

[51] Thompson, Alan, "Network Truth-Maintenance for Deduction and Modelling," *Proceedings of IJCAI-79*, 1979, 877–879.

[52] Winston, Patrick, *Artificial Intelligence,* second edition, Addison-Wesley, Reading, MA, 1984.

[53] Wong, Douglas, "Language Comprehension in a Problem Solver," *Proceedings of IJCAI-81*, 1981, 7–12.

[54] Wilensky, Robert, *Planning and Understanding* Addison-Wesley, Reading, MA, 1983.