The Use of Explicit Goals for Knowledge to Guide Inference and Learning*

Ashwin Ram

College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280
(404) 853-9372
E-mail: ashwin@cc.gatech.edu

Lawrence Hunter

National Library of Medicine
Building 38A, Mail Stop 54
Bethesda, MD 20894
(301) 496-9300
E-mail: hunter@nlm.nih.gov

Abstract

Combinatorial explosion of inferences has always been a central problem in artificial intelligence. Although the inferences that can be drawn from a reasoner's knowledge and from available inputs is very large (potentially infinite), the inferential resources available to any reasoning system are limited. With limited inferential capacity and very many potential inferences, reasoners must somehow control the process of inference.

Not all inferences are equally useful to a given reasoning system. Any reasoning system that has goals (or any form of a utility function) and acts based on its beliefs indirectly assigns utility to its beliefs. Given limits on the process of inference, and variation in the utility of inferences, it is clear that a reasoner ought to draw the inferences that will be most valuable to it.

This paper presents an approach to this problem that makes the utility of a (potential) belief an explicit part of the inference process. The method is to generate explicit desires for knowledge. The question of focus of attention is thereby transformed into two related problems: How can explicit desires for knowledge be used to control inference and facilitate resource-constrained goal pursuit in general? and, Where do these desires for knowledge come from? We present a theory of *knowledge goals*, or desires for knowledge, and their use in the processes of understanding and learning. The theory is illustrated using two case studies, a natural language understanding program that learns by reading novel or unusual newspaper stories, and a differential diagnosis program that improves its accuracy with experience.

Journal of Applied Intelligence, 2(1):47–73, 1992.

1 The focus of attention problem

Combinatorial explosion of inferences has always been a central problem in artificial intelligence. Although the inferences that can be drawn from a reasoner's knowledge and from available inputs is very large (potentially infinite), the inferential resources available to any reasoning system are limited. In general, reasoning systems simply cannot draw all justified inferences. With limited inferential capacity and very many potential inferences, reasoners must somehow control the process of inference.

Not all inferences are equally useful to a given reasoning system. Any reasoning system that has goals (or any form of a utility function), and acts based on its beliefs, indirectly assigns utility to its beliefs. That is, some beliefs have a causal role in actions that, in turn, lead to good outcomes. Beliefs arise, at least in part, by inference; therefore some inferences lead to better outcomes than others. Given limits on the process of inference, and variation in the utility of inferences, it is clear that a reasoner ought to draw the inferences that will be most valuable to it. How can that be done?

Several methods of controlling inference have been proposed. Perhaps the simplest is constrained forward chaining: making as many inferences as possible within the resource constraints. For example, MARGIE [Rieger, 1975] made all the justified inferences that required the chaining of no more than 5 of 17 rules. The amount of inference scales with the number of inference rules to the power of the length of possible chains, so in practical circumstances only a small percentage of justified inferences can be drawn with this method. Empirically, many of the inferences generated this way are useless, and many useful inferences are missed because their derivations are too long. Other systems rely on backward chaining to make only the inferences that might lead to a specified outcome. Unfortunately, many valuable inferences (even quite simple ones) are overlooked by this method, since surprises are impossible; the system can only infer what it's already looking for. Still another method tries to use probability measures to draw the most likely inferences. However, some relatively unlikely inferences can be extremely valuable. Reasoners may be explicitly interested in identifying unlikely events that have significant consequences. Simple counterexamples demonstrate the inadequacy of each of these approaches in the general case.

Our approach to this problem has been to make the utility of a (potential) belief an explicit part of the inference process. The method is to generate explicit desires for knowledge. The question of focus of attention is thereby transformed into two related problems: How can explicit desires for knowledge be used to control inference and facilitate resource-constrained goal pursuit in general? and, Where do these desires for knowledge come from? To address these questions, we must consider the uses of knowledge and inference, and how to assess the value of knowledge in order to control inference. To illustrate our approach, we present two implementations of these ideas: IVY and AQUA. AQUA is a natural language understanding program that learns by reading unusual newspaper stories, and IVY is a medical diagnosis program that improves its accuracy with experience. Both of these programs use knowledge goals to control their processing.

1.1 Inference and desires for knowledge

Inference plays a wide variety of roles in reasoning systems. For example, understanding natural language texts requires more than just identifying the literal meanings of words. Human understanders "read between the lines," making a large number of inferences not directly contained in a text. For example, if told that "John took some aspirin," most people would infer that he was in some pain, that he drank some water with them, that he swallowed them (rather than stole them), and so forth. As early natural language researchers discovered, the number of inferences that can be drawn from even a simple sentence is potentially infinite: There were fewer aspirin in the bottle, there was a bottle that the aspirin came in, the bottle was the size, shape and color of the usual over the counter medication bottle, the aspirin were the size of pills, were round and white, John felt better about 20 minutes later, etc., etc. It is not hard to find a story where being able to infer any of these facts is important to understanding the story, yet it is computationally intractable to

make all the inferences that can be drawn in any given situation. Somehow, people are able to manage the many possible inferences, generally without missing important details or taking a long time to understand.

A similar problem arises in abductive (e.g., diagnostic) systems. Abduction, the construction of causal explanations, is often viewed as inference to the "best" explanation. However, the definition of "best" is, as before, dependent on the goals of the reasoner in forming the explanation and not just on the correctness of the causal chain underlying the explanation [Ram and Leake, 1991]. In situations where there is not just a single correct explanation, the best explanation must address the reason that the explanation was required in the first place. For example, if the purpose of an explanation is to avoid repetition of a failure, the explanation should be generalizable to similar future situations. Often, the operational definition of "best" explanation includes some component of its utility to the reasoner. The point here is that the value of abduced knowledge should play a role in how an abductive process works.

Another instance of the inference control problem arises in the design of machine learning systems in general. It can be formally demonstrated that far more inferences are licensed by induction over a set of experiences than can be distinguished among using those experiences [Dietterich, 1989]. In general, an inductive system must have a method for preferring some inferences over others. Existing machine learning methods have done this by applying inductive biases (e.g., [Utgoff, 1986]), or by a priori limitations on the structure of the inferences they can make, through, for example, the use of decision trees or neural networks. These approaches can be considered *syntactic*, in that they constrain the form of the hypotheses considered, rather than their content.

The problem arises in noninductive learning systems as well. For example, the questions of "when to generalize" and "how far to generalize" are among the central issues in explanation-based learning. Again, most of the approaches to this problem thus far involve syntactic solutions. An exception is Minton's PRODIGY system [Minton, 1988], which evaluated each explanation's effect on the average reasoning process before integrating it into permanent memory. Although Minton did use the utility of an inference directly in his machine learning system, the utility of the inference did not play a role in the generation of the inference, only in deciding whether to store it.

If inference for learning must be constrained, it should be directed towards achieving (or at least facilitating) the goals of the performance system that the learner is part of. When a reasoner encounters difficulties during understanding, planning, or whatever else its task is, it should be able to remember the nature of these difficulties, and learn in order to become better at the tasks that it is trying to perform. This characterization of knowledge that it would be useful to have provides a valuable tool for determining the utility of knowledge and inference later on.

We propose that the method of restricting potentially inferrable hypotheses should be *content*-based. Explicit characterizations of *desirable knowledge* or *required knowledge* provide a principled method for restricting the realm of experience and background knowledge considered in inference, and thereby the size of the hypothesis space that must be considered. Having goals specifying what (kind of) knowledge is desirable provides a significant advantage for systems trying to learn from very complex experience.

In fact, we can carry this idea one step further: Not only can explicit goals about knowledge help control inference, they can be used to direct action intended to accomplish those goals. Rather than passively waiting for useful information to show up, a system can actively pursue the knowledge it desires, using specific learning plans or instantiations of general learning strategies. In order to actively plan to learn, as well as for control of inference in general, a reasoning system needs to represent and reason about its own desires for knowledge, and consider them actively in order to make decisions during the inference process.

This paper presents a theory of inference control for understanding and learning that is based on the notion of *knowledge goals*, a reasoner's specific desires to acquire and organize useful beliefs. This theory is broadly applicable to automated reasoning systems that improve with experience. A knowledge goal represents the need to fill in gaps in the reasoner's knowledge base that are detected when a piece of information required for

a task turns out to be missing, incorrect or otherwise problematic. Our theory addresses the representation of knowledge goals, methods for introspective reasoning about the reasoner's own knowledge to generate goals, heuristics for inference control and hypothesis evaluation using goal-based focus of attention criteria, as well as algorithms for learning through the active pursuit of knowledge goals.

In addition, this theory suggests a method for controlling the timing of inference. Programs don't always know everything they need to know. A program that learns has the additional problem of having what it knows change over time. An inference that may be difficult to make at one time may be much easier to make later, when additional information is available. Explicit knowledge goals make possible opportunistic learning over an extended period of time. A learning program intended to run indefinitely can use knowledge goals to make decisions about when to learn, as well as what to learn. Both of the programs described below manage a list of pending knowledge goals, and notice opportunities to achieve them during other processing, long after the goals have been generated (and after attempting to address them at that time). To our knowledge, this is a unique ability in machine learning programs, and it is mediated by explicit management of knowledge goals.

1.2 Do people have goals for knowledge?

Our theory is based both in a theoretical analysis of the constraints on inference in practical AI systems, and in empirical psychological evidence. People quite clearly have what psychologists often call "goal orientations," which have a significant effect on the inferences that people draw from their experiences. There is a large body of psychological research on goal direction in focus of attention, particularly from social psychology. Zukier's [1986] review concludes: "Experimental studies have clearly demonstrated that a person will structure and process information quite differently, depending on the future use he or she intends to make of it. Information integration clearly is preceded by future-oriented decision-making processes, which guide data selection and the choice of an appropriate strategy or mode from among the several that are available," [p. 495].

Hoffman, et al [1981] demonstrate that different goal orientations (e.g., "form an impression of a person in the following story" or "remember as much as you can from the following story") may influence not only the use of different representations, but also the selection among different kinds of processing. Although the goal orientations tested in that work are quite abstract, they significantly constrain the space of hypotheses consistent with the experimental materials. Srull and Wyer's [1986] results, although divergent in important respects from those of Hoffman, et al, also provide evidence that different goal orientations have a strong effect on learning.

In addition to the empirical psychological findings, consideration of the differences between how existing computer programs parse newspaper stories (e.g., FRUMP [DeJong, 1979]) and how people read them supports this approach. These differences include:

Subjectivity: People are biased. They interpret stories in a manner that suits them. They jump to conclusions. Computer programs, on the other hand, are usually designed to read stories in an objective manner, and to extract the "correct" or "true" interpretation of a story to the extent that they can.

Variable depth parsing: People don't read everything in great detail. They concentrate on details that they find relevant or interesting, and skim over the rest. In contrast, computer programs are designed to attend to every aspect of a story that is within the scope of their knowledge structures. Consequently, they either process the entire story in great depth (e.g., BORIS [Dyer, 1982]), or else they skim everything in the story (e.g., FRUMP [DeJong, 1979]). They can not decide which aspects to process in detail and which ones to ignore.

Learning and change: People change as they read. They never read the same story twice in the same way. They notice different things the second time around, or they simply get bored. After reading a story, they interpret other similar stories differently. Most computer programs, in contrast, are not adaptive; they always read a given story the same way.

What makes people different from computer programs? What is the missing element that our theories don't yet account for? The answer is simple: People read newspaper stories for a reason: to learn more about what they are interested in. Computers, on the other hand, don't. In fact, computers don't even have interests; there is nothing in particular that they are trying to find out when they read. If a computer program is to be a model of story understanding, it should also read for a "purpose."

Of course, people have several goals that do not make sense to attribute to computers. One might read a restaurant guide in order to satisfy hunger or entertainment goals, or to find a good place to go for a business lunch. Computers do not get hungry, and computers do not have business lunches.

However, these physiological and social goals give rise to several intellectual or cognitive goals. A goal to satisfy hunger gives rise to goals to find information: the name of a restaurant which serves the desired type of food, how expensive the restaurant is, the location of the restaurant, etc. These are goals to acquire information or knowledge, what we are calling knowledge goals. These goals can be held by computers too; a computer might "want" to find out the location of a restaurant, and read a guide in order to do so in the same way as a person might. While such a goal would not arise out of hunger in the case of the computer, it might well arise out of the "goal" to learn more about restaurants.

In other words, specific knowledge goals can arise from other, more general, desires to learn, to pursue one's intellectual interests, to improve one's model of the world. (We present a more detailed analysis of the origins of knowledge goals below.) These goals can be viewed as questions about the domain of interest. To be interested in terrorism, for example, is to have a lot of questions about various aspects of terrorism, and to think about these questions in the context of input data about terrorism, such as newspaper stories about terrorist incidents. For someone with these knowledge goals, the point of reading newspaper stories about terrorism is to answer one's questions, as well as to reveal flaws or gaps in one's model so as to improve it. These gaps give rise to new questions which in turn stimulate further interest in terrorism. Both computers and people can be "interested" in terrorism in this sense. These interests arise out of the underlying goal of wanting to learn and improve one's model of the world.

2 Computer programs with knowledge goals: Two case studies

Specific desires for knowledge have a clear role in the focus of attention during natural language processing, and in directing machine learning programs. They are apparent in studies of human cognition, and have strong computational advantages in practical resource-constrained reasoning situations. We believe that AI systems should generate and manipulate explicit knowledge goals. This approach has implications for nearly every kind of automated reasoning system. Here we will discuss the general issues and then concentrate on programs in two broadly representative areas: natural language understanding and medical diagnosis.

In complex knowledge-based systems, it is nearly impossible to create a system that contains all the knowledge it needs in order to accomplish its goals. Instead, such systems should be able to improve their performance with experience. Both natural language texts and medical cases provide a multiplicity of possible inferences that might conceivably be useful in improving the abilities of a performance system. In both areas, the use of explicit knowledge goals helps narrow the vast space of possible inferences to a more manageable set, and helps the program make decisions about when to draw potential inferences.

One of our examples is AQUA, a story understanding program that learns from what it reads [Ram, 1987; Ram, 1989]. In order to understand text, the performance system must integrate the text, which is often

ambiguous, elliptic and vague, with its world knowledge, which is often incomplete and possibly incorrect. In order to learn from what it reads, it must detect perceived anomalies in the text which may identify flaws or gaps in the understander's model of the domain, formulate explanations to resolve those anomalies, confirm or refute potential explanations, and possibly learn new explanations or modify incorrect ones.

These tasks can require a great deal of inference. In formulating an explanation, for example, the understander may need to know more about the situation than is explicitly stated before it can decide which is the best explanation. However, it is impossible to anticipate when a particular piece of knowledge will be available to the understander, since the real world (in the case of a story understanding program, the story) will not always provide exactly that piece of knowledge at exactly the time that the understander requires it. At some later time, a clue to the missing knowledge may become available, and the inferences necessary to acquire the desired knowledge become worth performing, even if they are complex or a priori unlikely. Thus the understander must be able to suspend a request for that piece of knowledge in memory, and reactivate the request at the right time when the information it needs may have become inferrable. In other words, the understander must be able to remember what it needs to know, and why, and those stored desires should have an effect on the inference process.

The process of natural language understanding generates knowledge goals (or questions) representing what the understander needs to know in order to perform an understanding task, be it explanation, learning, or some other cognitive task. These questions constitute the specific knowledge goals of the understander generated during a parsing experience, and are used to focus the reasoning processes on aspects of the input that are actually relevant. These goals are also used to focus the learning process so that the system learns what it needs to know in order to better carry out its tasks. As we shall see, this requires that the system be able to represent and reason about its own reasoning processes, and about the knowledge needed during these processes.

Another, rather different, example is IVY, a program that does differential diagnoses and is intended to improve its accuracy with experience [Hunter, 1989]. The basic idea was to design a program that improves its accuracy by storing information from the cases it diagnoses correctly. The problem is that there is a huge amount of information in the correctly diagnosed cases. Most of that information is not useful for improving diagnostic performance: after all, the cases were handled correctly. On the other hand, there are nuggets of information in that set of cases that are very useful for improving performance. How can a program find the useful bits without drowning in irrelevant information? IVY's approach was to use explanations of failures to identify diagnostic knowledge that is missing or incorrect. Those explanations can be transformed into characterizations of information that would be useful to have to avoid the failures. That characterization is effectively a knowledge goal, and can be used to rapidly scan correctly diagnosed cases for information that would help address previously encountered problems.

In order improve its accuracy, therefore, a diagnosis learner can identify the cause of a failure in terms of knowledge that was missing (or incorrect) and then use that to build a characterization of knowledge that would address the problem. This characterization constitutes a specific goal to acquire the correct knowledge. In order to identify the problematic knowledge and generate the goal, the learner must reason about the diagnostic reasoning process. Once analysis of failures has led to the generation of knowledge goals, the program can plan to acquire the knowledge. For IVY, the plans involved looking for specific kinds of information in one or more cases (either cases already in memory or as they arise for diagnosis), and then to transform and store the information so that it addresses the cause of the motivating failure. IVY's plans were capable of using case information both to supplement its general abilities and to find (and store) exceptions to its general rules.

Before exploring these two programs in more detail, let us pause to consider the commonalities in use and generation of knowledge goals between the programs. Both programs use desires about knowledge to control potentially explosive inferential processes. For AQUA, the number of inferences that can be drawn from a story is very large. AQUA only draws those inferences that are likely to answer questions that it

has. IVY is looking for ways to improve its diagnostic performance. Any given case might be relevant to a problem that has occurred in making a diagnosis. (As will be seen below, sometimes a relevant case may not even involve the same disease that caused the problem in the first place.) Every aspect of every diagnosed case might be relevant to any of the program's diagnostic knowledge. The number of possible interactions between all aspects of all cases and all knowledge is huge. IVY reduces this search space dramatically by characterizing the knowledge it desires as specifically as possible.

It is also apparent that, for both programs, characterizing desirable knowledge requires the ability to reason about internal reasoning processes and the knowledge they use. This ability to dynamically evaluate the knowledge used by internal processing (e.g., to find gaps in that knowledge that would have changed the processing had they been filled) may be a general feature of many kinds of learning systems. This kind of reasoning about internal processing and knowledge is a form of introspection that we conjecture may be a necessary component of human-like learning.

There are certain general questions that arise in any discussion of goal-based systems: Where do the goals come from? Do they conflict? How are conflicts resolved? We discussed the origin of goals for knowledge generally above, and describe in more detail below how the particular implementations handle that problem. AQUA generates knowledge goals when it fails to explain an event in a story; IVY generates knowledge goals when it fails to make a correct diagnosis. In both cases, the set of knowledge goals are dynamic, with new desires for knowledge arising as a result of system performance analysis.

One of the features of goals for knowledge that appears to distinguish them from goals for physical states is that, other than contention for resources like time and storage space, goals for knowledge do not appear to conflict with each other. That is, learning some piece of knowledge does not appear to be able to disable any preconditions for other learning; you cannot "paint the ladder before you paint the ceiling" in the domain of learning. In the programs presented below, we assume that goals for knowledge do not conflict with each other. Even without explicit goal conflict, there can be contention for resources such as time and storage space which may require prioritizing knowledge goals or other methods of resolving the contention. Both systems described here are strictly opportunistic goal pursuers, in that they wait for appropriate knowledge to appear at their inputs. A system that could initiate action in pursuit of a knowledge goal (e.g., issue a database query) would have to prioritize its knowledge goals in ways that IVY and AQUA do not.

2.1 AQUA

The AQUA project explored these ideas in a natural language understanding domain. AQUA is a question-driven story understanding program that learns about terrorism by reading newspaper stories about unusual terrorist incidents in the Middle East. The main point of that research was to create a dynamic story understanding program that is driven by its questions or goals to acquire knowledge (see figure 1). Rather than being "canned," the program is always changing as its questions change; it reads similar stories differently and forms different interpretations as its questions and interests evolve.

The AQUA project explores issues of learning, explanation, and interestingness in an integrated framework. The intent is not to have the program acquire the "right" understanding of terrorism, but rather to be able to wonder about unusual things it reads about and ask questions about them. As it learns more about the domain, it asks better and more detailed questions [Ram, 1991]. This kind of questioning forms the origins of creativity; rather than being satisfied with available explanations, a creative person asks questions and tries to explore the explanations in novel ways.

2.1.1 AQUA's knowledge goals

The questions that AQUA pursues are based on a taxonomy of types of knowledge goals. This taxonomy arises from the understanding tasks that underly AQUA's processing, as well as from a general set of

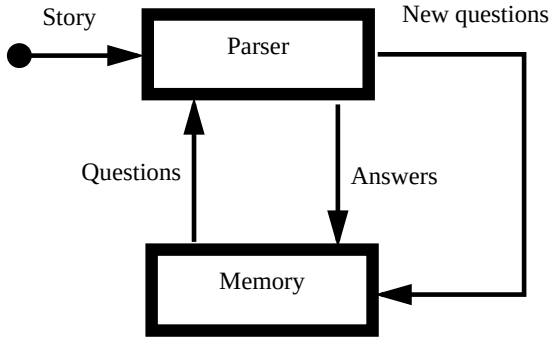

Figure 1: Question-driven understanding: Using knowledge goals to guide story understanding and learning.

"interests" that AQUA begins with.

Text goals: Knowledge goals of a text analysis program, arising from text-level tasks. These are the questions that arise from basic syntactic and semantic analysis that needs to be done on the input text, such as noun group attachment or pronoun reference. An example text goal is to find the referent of a pronoun.

Memory goals: Knowledge goals of a dynamic memory program, arising from memory-level tasks. A dynamic memory must be able to notice similarities, match incoming concepts to stereotypes in memory, form generalizations, and so on. An example memory goal might be to look for an event predicted by stored knowledge of a stereotyped action, such as wondering about what the ransom will be when one hears about a kidnapping.

Explanation goals: Goals of an explainer that arise from explanation-level tasks, including the detection and resolution of anomalies, and the building of motivational and causal explanations for the events in the story in order to understand why the characters acted as they did, or why certain events occurred or did not occur. An example explanation goal might be to figure out the motivation of a suicide truck bomber mentioned in a story.

Relevance goals: Goals of any intelligent system in the real world, concerning the identification of aspects of the current situation that are "interesting" or relevant to its general goals. An example here might involve looking for the name of an airline in a highjacking story if the understander were contemplating travelling by air soon.

The basic process of goal-based understanding involves the generation of knowledge goals seeking information required by various understanding tasks, the transformation of these knowledge goals into subgoals, and the matching of pending knowledge goals to information in the story. One might think of this as a process of question transformation, in which a reasoner generates questions which then trigger a parsing process which can in turn generate more questions. Example explanation goals for a typical suicide car bombing story are shown in figure 2, which represents the questions one might think about while reading such a story.

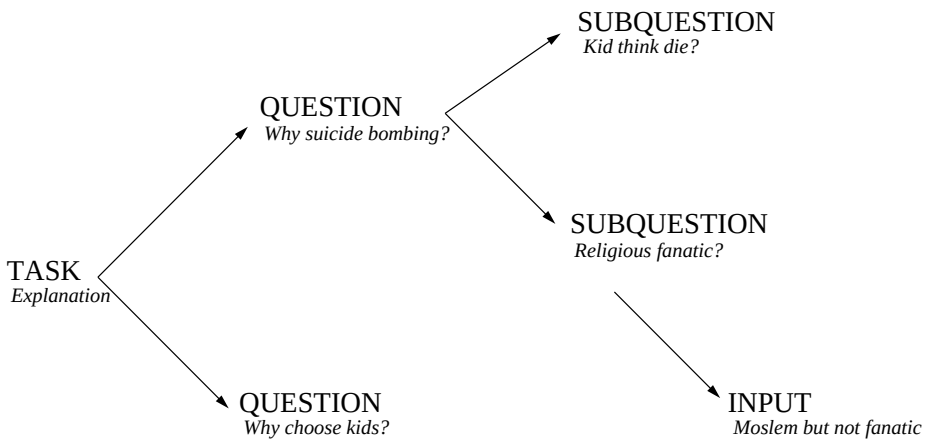

Figure 2: Questions underlying the religious fanatic explanation for suicide car bombing, showing the relationships between tasks, knowledge goals, and input.

2.1.2 AQUA's use of its knowledge goals

Each of AQUA's knowledge goals type is expressed as a question that focuses on a different aspect of the story. For example, explanation questions focus on different types of anomalies, and on explanations for these anomalies. Asking an anomaly detection question is essential to detecting the corresponding anomaly. For example, asking the question "Does the actor want the outcome of this action?" is essential to the detection of a goal violation anomaly, in the sense that the program could not notice the anomaly if it did not focus on the goals of the agent, that is, if it did not think of asking the question.

To put this another way, the questions asked by the understander affect the final understanding that the understander comes to. Thus it is important for the understander to ask the "right" questions in order to achieve a detailed understanding of the situation. For the purpose of understanding stories involving motivations of people, we have developed a taxonomy of motivational questions that focus on those motivational aspects of stories that are needed to build volitional explanations based on the planning/decision model that underlies AQUA's theory of explanation [Ram, 1990a]. A small part of this taxonomy is shown in figure 3, which depicts basic questions the system asks in explaining an agent's actions.

The taxonomy of questions is based on the understanding tasks that AQUA needs to perform when it reads a story. In addition to their theoretical role in our model of inference control and interestingness, knowledge goals have also played an implementational role in our research by providing a uniform mechanism for the integration of various cognitive processes. For example, knowledge goals arising from, say, memory tasks are indexed in memory and used in the same way as knowledge goals arising from explanation tasks. A knowledge goal generated from one task may be suspended, and satisfied opportunistically during the pursuit of some other task at a later stage or even during the processing of a different story. Implementational details of AQUA's *opportunistic memory architecture* may be found in Ram [1989].

The processing cycle in AQUA has three interacting steps: READ, EXPLAIN and GENERALIZE.

The READ step: AQUA reads a piece of text, guided by the questions in memory. It tries to answer these questions using the new piece of information.

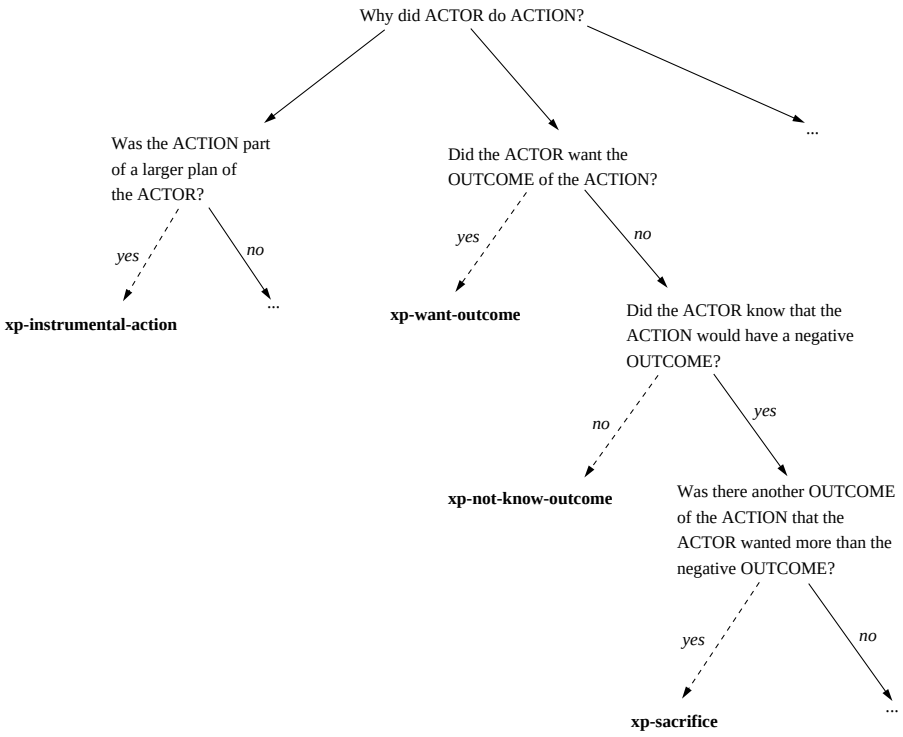

Figure 3: Anomaly detection questions represent knowledge goals arising from the process of detecting anomalies in the input. These goals seek information needed in order to determine whether the input is anomalous. If this information is, or becomes, available, the reasoner can formulate an explanation for the anomalous input.

Read some text, focussing attention on interesting input as determined below. Build minimal representations in memory.

Retrieve extant knowledge goals or questions indexed in memory that might be relevant, i.e., whose concept specifications are satisfied by the new input. Use these questions as an interestingness measure to focus the **read** above.

Answer the questions retrieved in the previous step. Unify the answer with each question, and restart the suspended process represented by the task specification. E.g., if the question is in service of hypothesis verification:

Answer question by either confirming or refuting it.

Propagate back to the hypothesis that the question originated from.

Confirm/refute hypotheses. If the verification questions of a hypothesis are confirmed, confirm the hypothesis and refute its competitors. If any verification question of a hypothesis is refuted, refute the corresponding hypothesis.

Explain the new input if necessary, i.e., if interesting and not already explained.

The EXPLAIN step: The EXPLAIN step implements the basic explanation cycle in AQUA, which is based on Schank's [1986] theory of explanation patterns (XPs). AQUA builds on Schank's theory of explanation patterns in three ways. First, a content theory of volitional explanations for motivational analysis

is proposed. Second, a graph-based representation of the structure of explanation patterns is introduced. Third, the process of case-based explanation, while similar to that used by the SWALE program [Kass *et al.*, 1986], is formulated in a knowledge goal-based framework. Our emphasis is on the knowledge goals that underly the creation, verification and learning of explanations. Further details of the explanation process may be found in Ram [1990a; 1989].

Detect anomalies in input by asking anomaly detection questions

Formulate XP retrieval questions

Retrieve XPs that might help explain the anomaly

Apply XP to input:

If in applying the XP an anomaly is detected, **characterize** the anomaly and **explain** it recursively.

If the XP is applicable to the input:

Construct hypothesis by instantiating the explanation pattern.

Construct verification questions to help verify or refute the new hypothesis.

Index questions in memory to allow them to be found in the next step.

Answer questions by **read**ing further, focussing attention on input concepts that trigger questions in memory.

Confirm/refute hypotheses when their verification questions are answered, as appropriate.

The GENERALIZE step: Since questions represent the knowledge goals of the understander, they provide the focus for learning. As discussed in a later section, AQUA can:

Generalize novel answers to its questions.

Index these answers in memory, so that the task that originally generated the question would now find the information instead of failing.

As currently implemented, AQUA's memory consists of about 700 concepts represented as frames, including about 15-20 abstract XPs, 10 stereotypical XPs, 50 MOPs (most of which deal with the kinds of actions encountered in suicide bombing stories), 250 relations (including causal and volitional relations), and 20 interestingness heuristics (most of which are represented procedurally). The range of stories that AQUA can handle is limited only by the XPs in memory. We have focussed mostly on the domain of newspaper stories about suicide bombing, such as stories about religious fanatics, depressed teenagers, Kamikazes, and so on, although it would be straightforward to extend the program to other domains. As AQUA reads, it asks better and more detailed questions about the stories, formulates knowledge goals to answer these questions, and learns when its knowledge goals are satisfied (perhaps in a later story). AQUA can identify and learn from three types of learning situations:

1. **Missing knowledge:** This occurs when AQUA does not have an XP specific to a novel situation encountered in the story. For example, when AQUA reads a suicide bombing story for the first time without a specific XP that represents religious fanaticism, it can formulate a knowledge goal to learn about religious fanaticism by refining its general knowledge about goal sacrifice.

2. **Misindexed knowledge:** This occurs when AQUA does have the required knowledge, but it is represented in a different context and hence not retrieved in the present story. For example, when AQUA reads a suicide bombing story in which the bomber is blackmailed into going on the bombing mission, it does not immediately think of blackmail as a possible explanation for suicide bombing. After reading this story, however, it learns a new index for blackmail in the suicide bombing context.

3. **Incomplete knowledge:** This occurs when AQUA has the required knowledge and is able to retrieve it in the present situation, but the knowledge structure itself is incomplete. In the blackmail situation, for example, although AQUA ultimately learns to apply blackmail in a suicide bombing context, there are several questions that are still pending: How could someone be blackmailed into suicide? What could the bomber want that was more important than life? These questions constitute knowledge goals that are attached to the "blackmailed into suicide bombing" explanation, and are used to focus attention on, and learn from, relevant facts when AQUA reads future stories about suicide bombers being blackmailed.

Although we have mainly focussed on knowledge goals for the task of learning from explanations and explanation failures, other types of knowledge goals discussed earlier are also formulated by the system and pursued in a uniform manner. AQUA gradually improves its breadth and depth of understanding of the domain through the above types of knowledge goal-driven learning. We are currently extending AQUA to incorporate learning through cross-domain reasoning about knowledge goals as well.

2.2 IVY

Knowledge goals are generated and acted upon somewhat differently in the IVY program. IVY's task is to diagnose structured descriptions of lung tumor pathology images. These descriptions contain information about populations of cells taken from a lung and colored with various stains. There are many levels of description in each image, ranging from characteristics of large groups of cells (such as the shape of a glandular formation) to the presence and character of subcellular organelles. The amount of information available in a typical input is very large (on average, IVY's case descriptions contained 116 slots), and most of it is not relevant to making the ultimate diagnosis. There is also no direct mapping from characteristics to diagnoses in this domain. Many of the diagnoses are imprecise, and do not have definitions in terms of features that are individually necessary and collectively sufficient for their identification.

IVY, like human pathologists, uses the method of differential diagnosis to arrive at a diagnosis. For IVY, the process involves three distinct stages. First, the image description is searched for evidence of the presence of general disease classes; this is the *recognition step*. Any class not explicitly ruled out is included in the first pass hypotheses. The second pass involves specifying these hypotheses as far as possible, to create a final differential; this is the *specification step*. Associated with each child of a disease class is a set of specification rules, which are applied recursively. The last stage in the process is to pick the best hypothesis from the differential; this is the *distinction step*. Some pairs of hypotheses have associated rules that specify evidence that will cause one to be preferred over the other. In other cases, general distinction rules are applied, until a single, best hypothesis remains.

Knowledge goals play a role in IVY only after the diagnosis is complete. A lung tumor pathology expert evaluates IVY's conclusions, specifying the correct diagnosis for each case. If the program's diagnosis was incorrect, the program explains its failure, and generates one or more knowledge goals. If the diagnosis was correct, it examines its pending knowledge goals to see if any of them can be satisfied.

2.2.1 IVY's knowledge goals

Learning in IVY is failure motivated. Knowledge goals are generated when the program makes an incorrect diagnosis. First, IVY explains the cause of a failure (in terms of missing or incorrect knowledge) and then

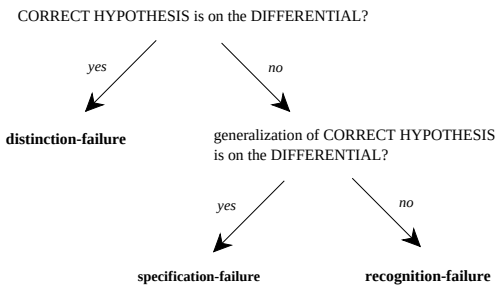

Figure 4: The explanation decision tree for identifying the step in the diagnostic process that led to a failure.

transforms that explanation into a characterization of knowledge that would have prevented the failure, that is, a new knowledge goal.

The basic step in IVY's knowledge goal generation process is explanation of a performance failure. Different kinds of knowledge are used at each stage of the diagnostic process. Therefore, identifying the stage of the diagnostic process at which the failure occurred is helpful in identifying the knowledge that caused the problem. Once the stage of the diagnostic process at which the problem first appeared has been identified, it is possible to further specify the problem by analyzing the specific knowledge used and the evidence gathered at that processing stage. In order to make these assessments, the explanation process must have access to some of the internal states of the diagnostician. In particular, it must have access to the hypothesis list and the evidence gathered at each stage of the diagnostic process. Such information must be stored at diagnosis time, and, along with the (externally supplied) correct diagnosis, forms the input to the failure explanation process.

IVY's explanations of failures have two components: the process that failed, and the kind of knowledge used (or not used) by that process which caused the failure. By working backwards and considering the inputs and outputs of each step, it is possible to determine where the error was made. The decision tree for identifying the step in the diagnostic process that failed is illustrated in figure 4.

Once the diagnostic step that failed has been identified, the explanation process can identify the faulty knowledge. Each step in the diagnostic process uses both general knowledge and rules specific to the current hypotheses to accomplish its tasks. General knowledge can be inappropriate, and specific knowledge can be incorrect, incomplete, or not operational. IVY has a set of explanation rules for each process that use the diagnosis trace to identify the kind of knowledge that caused the problem. For example, if there was no specific knowledge about (what would have been) the correct hypothesis, the problem was that inappropriate general knowledge was used. Alternatively, if there was specific knowledge, but it did not apply given the evidence, then the specific knowledge was incomplete. The failure explanations that IVY produces generally have three parts: the failed process, the kind of knowledge failure, and bindings of specific pieces of knowledge to variables in the description of the kind of failure. An example explanation can be paraphrased "Distinction failed, entities were confused due to missing specific knowledge, the entities were carcinoid tumor cells and intermediate carcinoma tumor cells." Implementational details can be found in Hunter [1989].

The second step in IVY's knowledge goal generation process is to turn these explanations into specific desires. The transformation is generally straightforward. Each general explanation type (i.e., each combination of failed process and type of failed knowledge) has associated with it a knowledge goal skeleton. Continuing the previous example, distinction failures due to missing specific knowledge have an associated knowledge goal skeleton for finding the missing knowledge. Variables in the skeletons are bound to the bindings from the explanation. In the example case, the goal generated is to find a method for distinguishing between carcinoid cells and intermediate carcinoma cells.

2.2.2 IVY's use of its knowledge goals

IVY's knowledge planner takes as input newly generated knowledge goals. Each goal leads to the generation of one or more plans. A plan specifies information that would be useful in addressing a goal, and what to do when that information becomes available. The required information forms the plan's preconditions and the specification of what to do are the plan's actions. When a plan's preconditions are met, the plan is executed, that is, the plan's actions can be taken.

People use a wide variety of plans to achieve their knowledge goals, ranging from looking up an answer in a reference book to designing and running scientific experiments. IVY's plans identify methods for finding desired information in single cases. After each successful diagnosis, IVY compares the unsatisfied preconditions of all of its pending plans to the contents of the just diagnosed case. If the case can be used to satisfy the preconditions, the plan specifies how to store the case in memory so that the diagnostic failure that motivated the plan is addressed.

IVY's planning abilities were limited to selection among and instantiation of eight different plan schemas. Knowledge goal skeletons had from one to three plan schemas associated with them. INVESTIGATOR [Hunter, 1990b; Hunter, 1990a] uses a more flexible knowledge planner. Depending on the specifics of the knowledge goal (i.e., characteristics of the variable bindings in the goal skeletons) one or more of the plan schemata are instantiated. Consider the following example.

One of the plans IVY used to find specific distinction knowledge is to "find the visibility conditions under which the distinction can be made." Failures in pathologic diagnosis often occur because the tissue sample was not viewed under the correct conditions. For example, the magnification may be too low (or too high) or the wrong stain may have been used. It is possible to find out what the appropriate conditions are by looking for a case in which the distinction is made correctly. The preconditions for this plan are therefore a case where (1) both entities appear on the differential, and (2) a correct diagnosis (of one of the entities) is reached. When such a case is found, a new piece of distinction knowledge can be inferred. The left hand side (condition) of the new distinction rule has two conjuncts: (a) the visibility conditions (magnification, stain, etc.) in the success case, and (b) the attribute(s) that were used to make the distinction in the correctly diagnosed case. The right hand side (action) is the diagnosis from the success case.

Consider the example of the carcinoid mistaken for an intermediate cell carcinoma, discussed above. IVY's explanation for its failure was that the diagnoses were confused because of a lack of specific distinction knowledge. This explanation led to the generation of a goal to find knowledge that can be used to distinguish between the two diagnoses. The plan described above applies to this goal. In order for the plan to execute, its preconditions must be met: A case must be found where (1) both carcinoid and intermediate cell carcinoma are on the differential, and (2) the diagnosis reached was either carcinoid or intermediate cell carcinoma and was correct. IVY later encountered a high magnification image of a carcinoid; in that case, both carcinoid and intermediate cell carcinoma were on the differential. The correct diagnosis was reached on the basis of general knowledge (dense core granules, a characteristic of carcinoids but not intermediate cell tumors, were present). After the diagnosis was verified, the plan's preconditions were tested and satisfied. The plan's action component created a distinction method for carcinoids with the following conditions: (a) the visibility conditions under which the distinction was made (high magnification, H&E stain), and (b) the difference between the hypotheses that allowed the distinction to be made (dense core granules).

IVY's knowledge planner was quite simple. A learner situated in a complex world must therefore make decisions about what is worth learning. The results of these decisions are explicit (although not necessarily conscious) goals about the knowledge a learner desires. Learning does not have to be a passive process: people generally act in order to learn. Their goals can be used to direct the selection of the actions taken.

IVY ultimately diagnosed 118 descriptions of lung tumors, selected to be broadly representative. A jackknife test was used to evaluate the power of the learning algorithm. The jackknife test works by removing one case from the training set to use as a test. The learning algorithm is run on the remaining cases, and

evaluated on the test case. This procedure is repeated so that each case in the corpus is used as a test, and the percentage of test cases diagnosed correctly is compared to the performance of the algorithm without learning. IVY was capable of diagnosing 95 of the 118 cases correctly without learning (about 80%). The goals generated by three of those failures could eventually be satisfied by the program, leading to four additional correct diagnoses (about 84% success).

More important than any measure of percentage improvement in performance due to learning is the quality of the material learned. Two of the three "lessons" that IVY learned in response to its knowledge goals were identified by the domain expert, Dr. Yesner, as good teaching cases. One of the images IVY selected had been previously used as an example in one of Dr. Yesner's publications [Yesner and Carter, 1982]. Dr. Yesner considered the third case discovered by IVY "quite useful" for showing how to avoid a subtle diagnostic error. The ability of a program to independently identify cases that a domain expert considers interesting, on the basis of the program's experience and its consequent desire for information, bodes well for the potential of knowledge planning in general. The knowledge it gained through learning involves more than fitting parameters or chunking knowledge it already had. After learning, the program was able to accurately diagnose difficult cases, and demonstrate the basis for its diagnoses by using several previous cases as precedents. The program found a use for case in the expert's library that the expert hadn't thought of before, which, at the very least, impressed him. We suggest that this kind of learning offers a qualitatively significant improvement over traditional machine learning approaches.

3 A theory of knowledge goals

When either AQUA or IVY tries to reason about something, e.g., it is trying to explain something that seems anomalous, and it needs to know something that isn't there in memory, it formulates a knowledge goal that is indexed in memory at the point at which it expected to find the information. These goals consist of two parts:

1. **Concept specification:** the goal object, i.e., the desired information.
2. **Task specification:** what to do with the information once it comes in, which depends on why the goal was generated.

The transformation of a knowledge goal into a plan for achieving the goal is the attempt to operationalize these components.

3.1 Concept specification

The concept specification represents the information that the question is looking for. This is represented using a memory structure that specifies what would be minimally acceptable as an answer to the question. A new piece of knowledge is an answer to a question if it matches the specification completely. The answer could specify more than the question required, of course.

The concept specification looks like any other memory structure, except that it is marked with the label `hypothesized`, `hypothesized-in` or `hypothesized-out`, as appropriate.[1] When the question is answered, the concept becomes `in` or `out`.

[1] The labels `in` and `out` are used to represent belief as in most truth maintenance systems [Doyle, 1979].

3.2 Task specification

The task specification represents what to do with the answer once it comes in, which depends on why the question was asked. Typically this involves indexing the new knowledge in the appropriate place in a structured memory (i.e., in the organization of the program's knowledge) or forming a generalization based on the answer. The task specification may be represented either as a procedure or closure to be run, or as a declarative specification of the suspended task. When the question is answered, either because the program actively pursued it, or opportunistically while it was processing something else, the suspended process that depends on that information is restarted.

Both representations are equivalent for the purposes of restarting suspended understanding tasks. However, if the program needs to reason about the purpose of the question, a declarative representation is necessary because it allows the program to access the internals of the task that produced the question. For example, if the program is trying to decide which of two questions is more interesting or important, it might use a heuristic that preferred explanation questions to, say, text-level questions. In this case, a closure would not suffice as a task specification.

3.3 The origins of knowledge goals

A mechanism for generating knowledge goals must ultimately be judged by the overall utility of the learning that results from those goals. The utility of knowledge learned depends on the goals that the learner is pursuing, the mechanisms that put knowledge to use in pursuing those goals, and the knowledge that the learner already has.

Why would an understander need to find something out in the first place? Ultimately, the point of reading is to learn more about the world. Questions arise when reading a story reveals gaps or inconsistencies in the world model. It is useful to focus attention on such questions because they arise from a "need to learn." For example, questions arising from anomalous facts are more useful than those arising from routine stereotypical facts, since in the former case the understander may learn something new about the world.

We suggest three different approaches to generating knowledge goals. The first approach is to estimate the utility of desired knowledge directly. For certain classes of knowledge, those that are broadly useful to a wide variety of typical goals, this calculation may be possible. The second approach is to generate knowledge goals from other goals of the learner. These goals may be subgoals directly related to a performance goal (e.g., desiring to know the combination to a safe), or may be related through more complex inference, like IVY's goals generated via the explanation of a performance failure. The third method for identifying knowledge goals is to analyze the structural characteristics of the background knowledge of a learner. Let us consider each method in turn.

3.3.1 Knowledge goals of high average utility

Some kinds of knowledge are so generally useful to a goal pursuer that they are always worth pursuing. These goals may be innate, because the analysis of the expected utility of the knowledge does not change as the goals or knowledge of the learner change. These knowledge goals are also likely to be the evolutionarily most primitive. Other methods for generating knowledge goals presuppose some existing knowledge; generating goals based on the expected utility of desired knowledge need not.

What kinds of knowledge have an expected utility so high that they are generally worth pursuing? A specific answer to that question depends on at least a general characterization of the needs and environment of the learner that will acquire that knowledge, but there appear to be several classes of knowledge that many organisms appear to treat as worth learning about generally. One example is the learning of spatial maps of the organism's environment.

Exploration is the general term for behavior based on the goals to learn spatial maps. An animal explores in order to build knowledge of its spatial environment. Knowledge gained by exploration has many uses. One use in particular illustrates the selective advantage of exploration over stimulus-response learning. A creature that avoids predators by running and hiding can acquire a knowledge of good hiding spaces by exploring. A creature that has a stock of hiding places has a clear advantage over one that must find a novel hiding place every time it is pursued. It may be possible for a stimulus-response learner to associate rewards with good hiding places, thereby achieving the same benefit, but the high cost of the repeated trials necessary to make such an association gives the animal that explores a significant advantage.

3.3.2 Knowledge goals derived from other goals

An entity that uses a knowledge base in the pursuit of its goals has a much more specific basis for generating new knowledge goals. By analyzing the relationship between its knowledge and its specific current goals, a learner can devise knowledge goals that directly facilitate the accomplishment of those goals. That is, knowledge goals can be subgoals to other performance goals.

People often generate goals for information that will help them accomplish other tasks. Schank and Abelson [1977] proposes the goal D-KNOW (change knowledge state) as a subgoal for accomplishing some other goal. Their example was the story "Willa was hungry. She took out the Michelin Guide." The actor in this story wanted to know the names and locations of restaurants in order to pick one to eat in.

Other kinds processing can also be used to generate knowledge goals as subgoals to other goals. For example, an animal that associates a particular location both with a good water source and with periodic attacks from predators may be motivated to find a good hiding place or escape route near that location. From a story understander's point of view, a fact that answers a question is worth focussing on since it helps to achieve a knowledge goal of the understander, which in turn allows the understander to continue the reasoning task that was awaiting the answer. Also, an inexplicable or anomalous fact is worth focussing if the questions arise from a gap or inconsistency in the understander's knowledge base, since the understander may be able to improve its knowledge base by learning something new about the world. AQUA uses heuristics of this kind, called *interestingness heuristics*, to decide which knowledge goals to generate and pursue [Ram, 1990c].

A somewhat less direct relationship between the goals that an entity is pursuing and the generation of new knowledge goals can be found in goals generated in response to failures. If a goal pursuer is using knowledge to accomplish a goal and unexpectedly fails to achieve the goal, an analysis of the knowledge used may be useful for generating new knowledge goals. This, of course, is the strategy for generating knowledge goals used in IVY. In addition, it may also be possible for a learner to detect knowledge that it needs to acquire by *simulating* the execution of a proposed plan, rather than waiting for a failure to occur.

In order to generate knowledge goals on the basis of analysis of a goal failure, however, a learner must be able to analyze the processes and knowledge that it was using when the failure occurred. In other words, it must have knowledge about what it was doing, a particularly useful sort of self-knowledge. It is, of course, possible to execute a sequence of actions without knowing the relationship between the actions and the results. On the other hand, an entity that knows the relationship between the actions involved in a sequence and the ultimate result has an advantage over an entity that doesn't. With that knowledge comes the ability to manipulate the process, adapt parts of it to other uses and explain failures. Goals to identify the separable steps in a sequence of actions and to acquire knowledge about their function therefore have an adaptive value. The same is true for finding out about objects that play a role in goal pursuit.

3.3.3 Knowledge goals that arise from analysis of the structure of knowledge

Maintaining a large memory of useful knowledge requires organizing it so that it is computationally feasible to retrieve the right piece of knowledge at the right time. This organization also makes possible the generation of an additional class of knowledge goals: those based on an analysis of the structure of existing knowledge.

Consider the structure of a pathologist's knowledge of disease. For pathologists, a disease consists of an etiology (cause), pathogenesis (loosely speaking, the effects of the disease on the body), and the prognosis (information about the course of the disease useful for making predictions). When a pathologist encounters a novel instance of any of these aspects of a disease, he generates knowledge goals for the other two aspects of that disease. For example, if a pathologist notes an unusual disease course, he wants to know what about the origins or appearance of the disease was also unusual. Of if a pathologist sees a disease with a novel appearance, he then desires information about the origins and expected course of the new disease.

What kinds of characterizations can lead to the formations of new knowledge goals? In the pathology example above, it is an assessment of the completeness of knowledge of an object or process. In order to generate knowledge goals on this basis, there must be some abstract description of what counts as complete knowledge. In the pathology example, this description is expressed in the assertion that every disease has an etiology, pathogenesis and prognosis. Perhaps the most general characterization of the completeness of knowledge is the idea that everything has a cause. A learner that believes this may generate knowledge goals to find the cause of everything it knows. Other general completeness considerations may lead to the formation of knowledge goals about the composition of objects or the goal priorities of agents, and so on.

Completeness is not the only structural characteristic of knowledge that can be used to generated goals. A second important structural characterization of memory can be made on the basis of connectivity. More densely interconnected knowledge is, in general, more useful in generating plans or explanations than knowledge which is less closely related to other knowledge. This observation indicates that there may be knowledge goals to determine the relationships between seemingly independent sets of knowledge. The potential number of relationships to explore is very large, so there must be some additional direction to the search for relationships between classes of knowledge. One source of direction may be similar causal factors. For example, the discovery that a single causal agent plays a role in two disparate domains is a good reason to generate goals for determining other relationships between the domains.

There is a risk with both of these methods of generating knowledge, because the number of knowledge goals that might be generated with each is potentially very large. The process of managing the activation of knowledge goals, described below, may help ameliorate this problem, but the generation and prioritization of knowledge goals is still very much an open research issue.

4 Using knowledge goals to guide processing

A program that uses knowledge goals to guide understanding is an improvement over one that processes everything in equal detail, that is, one that is completely data-driven. For example, an understander that is completely text-driven would process everything in detail in the hope that it might turn out to be relevant. To avoid this, the understander should draw only those inferences which would help it find out what it needs to know. In other words, the understander should use its knowledge goals to focus its attention on the interesting aspects of the story, where "interesting" can be defined as "relating to something the understander wants to find out about."

It is useful to focus on knowledge goals because they arise from a "need to learn." There are two basic ways in which a fact can turn out to be worth processing in this sense:

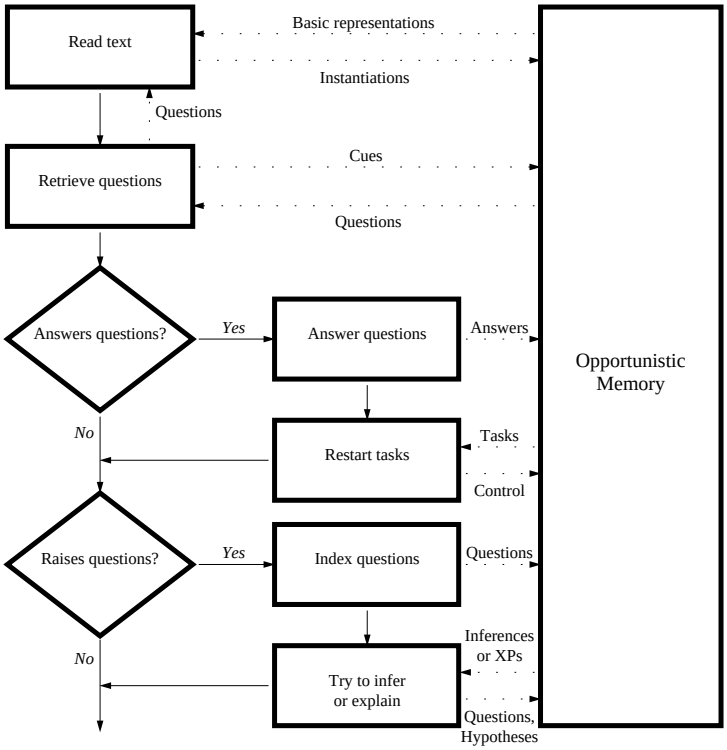

Figure 5: Control structure: The understanding cycle in AQUA. A fact is interesting if it satisfies a knowledge goal pending in memory, or if it gives rise to new knowledge goals. Uninteresting facts pass vertically down with minimal processing; interesting facts cause suspended understanding tasks to be restarted, or new tasks to be created. New tasks can give rise to new knowledge goals, which are suspended along with the tasks if answers are not yet known and cannot be inferred.

Top-down: A fact that helps achieve a knowledge goal, or answers a pending question, is worth focussing on since it allows the reasoning system to continue the reasoning task that required the knowledge in the first place.

Bottom-up: A fact that gives rise to new knowledge goals, or raises new questions, is worth focussing on if the knowledge goals arise from a gap or inconsistency in the reasoning system's knowledge base, since the system may be able to improve its knowledge base by learning something new about the world.

These correspond to the two diamonds in figure 5.

4.1 Theory of inference control

Ideally, only those inferences should be drawn that lead to conclusions that the program needs to know. But this is not always possible in practice. Given that the basic task of a knowledge goal-based understanding program is to try to answer questions in its memory by reading stories, there is an obvious choice to be made in the design of the program as characterized by the following extremes:

Text-driven: A text-driven program would read the text, build representations for it, and then process it to see if it addressed any knowledge goals in memory.

Goal-driven: A program that was totally goal-driven would pick the most interesting or urgent knowledge goal in memory and try to answer it via inference, reading text, or indeed any other method it had available.

A similar dichotomy arises in the design of problem-solving programs as well. Each of these approaches has its disadvantages. The text- or data-driven method is completely bottom-up. It tries to process in detail everything in the hope that it might turn out to be relevant. Instead, we would like the program to concentrate on those aspects that were of interest to it. In other words, we would like the process to be driven by the interests or goals of the reasoner.

The goal- or question-driven method, when taken to the extreme, is too top-down. A program built using this method would only see what it was looking for already. Furthermore, rather than expending resources in trying to pursue a knowledge goal immediately, it might be advantageous to index the knowledge goal in memory, to be satisfied later when the opportunity arises. Finally, so as not to overlook obvious information and be sufficiently sensitive to the exigencies of the input, the process should be data-driven as well. The interaction between these requirements is non-trivial.

AQUA is designed as a compromise between these two approaches. The basic understanding cycle that it uses is as follows. The parser reads the story word by word, trying to build a basic conceptual structure to represent the input. As quickly as possible, this structure is related to pending knowledge goals in memory. If the new structure satisfies a knowledge goal, the suspended understanding task that was awaiting that piece of knowledge is restarted. Thus the program only draws those inferences that are required to match the new structure to its knowledge goals, and, after the suspended task is restarted, those that are demanded by the task that generated the knowledge goal in the first place.

This method is called "graded parsing" or "variable-depth parsing". The process is data-driven to the extent that pieces of the input for which there are no explicit expectations but which are likely to be relevant are processed to the extent necessary to determine their relevance. In practice, this means that there is a set of bottom-up processes that the program always runs on incoming text. Further processing of the input is done only if these processes generate knowledge goals, or if the input turns out to satisfy a knowledge goal already in memory.

The real issue here, of course, is how much inference should be done at the time the knowledge goals are generated, and how much should be done when the input comes in. The answer depends on six factors:

1. **Certainty of inference:** The probability of the inference rules used to find or infer answers to knowledge goals, or the likelihood that the conclusions will be true. In a logic system where an inference rule represents a deduction, this probability is 1.

2. **Cost of inference:** The cost of making inferences or of matching and applying inference rules. The cheaper the inference, the more it is worth the system's while to make it.

3. **Usefulness of knowledge goal:** The usefulness of the conclusion that the knowledge goal is seeking. Since knowledge goals are generated in service of reasoning tasks, this is the same as the importance of performing that task. If the task is very important, it is worth making the inference even if it is very expensive to do so.

4. **Likelihood of knowledge goal being useful:** The likelihood that the knowledge goal will be useful, i.e., the likelihood that the knowledge will actually turn out to be useful in performing the reasoning task. If knowledge goals are only generated from tasks that absolutely require that knowledge (as opposed to those that may be facilitated by that knowledge if it were present), this likelihood is 1.

5. **Indexing cost:** The cost of keeping indexed questions in memory and matching to them. If there are too many questions in memory, it might be too expensive to find them or to match input to potentially

relevant questions. This cost depends on the scheme used to maintain questions in memory, and is discussed below.

6. **Likelihood of knowledge goal being satisfied:** The above factors are "content-free" heuristics in the sense that the reasoning system does not rely on knowledge of the *content* or *types* of knowledge goals that it is likely to generate, or on the content or types of inferences that the system is likely to make when given new input. In addition, one would like the system to generate the types of knowledge goals that are likely to match the inferences normally made by the bottom-up processing that is always performed on incoming facts. The last criterion for inference control, therefore, is the likelihood of a knowledge goal being satisfied, which depends on knowledge about the inferences that are likely to be made by the system's own inference processes.

The above heuristics are not represented formally in the AQUA program. In other words, there are no explicit functions to compute each of these metrics and to make a decision based on them. However, the heuristics to determine the utility or interestingness of knowledge goals, and to index knowledge goals in memory, have been designed keeping these metrics in mind so that the process is efficient. More research is required to develop a theory of inference control based on the above heuristics that can be used by the reasoning system itself (as opposed to by the programmer) in making inference control decisions. The main concern in AQUA has been the formulation and indexing of knowledge goals, and their use in focussing AQUA's understanding and learning processes.

IVY's knowledge goals control its inference less directly. If all of IVY's knowledge plans were converted to forward chaining inference rules, they would produce a great deal of irrelevant, but true knowledge. For example, consider one of the knowledge plans described in the carcinoid/intermediate cell carcinoma example above: finding the visibility conditions under which the distinction can be made. The effect of this plan is to store the visibility conditions (magnification, stain, etc.) under which a desired distinction can be made. Storing this information for every distinction that the system can make would lead to the kind of combinatorial explosion described in the introduction. Instead, IVY only investigates the visibility conditions of methods that could have made a distinction that is needed to avoid an error.

4.2 Mechanisms for knowledge goal management

Maintaining a collection of explicit knowledge goals introduces new issues into the design of AI programs. The goals themselves must be organized, applied, and disposed of when no longer useful. These management tasks were addressed by the following general mechanisms which were required in IVY and AQUA:

- **Knowledge goal retrieval:** finding suspended knowledge goals that a new piece of knowledge might satisfy.
- **Knowledge goal indexing:** storing knowledge goals in memory so that they are found almost only when they are relevant.
- **Process scheduling:** restarting suspended tasks that depend on knowledge goals when the knowledge goals are satisfied.
- **Hypothesis management:** deleting alternative knowledge goals and hypotheses when a knowledge goal is satisfied, because their likelihood of being useful decreases since an alternative has been found.

4.2.1 Indexing knowledge goals

Where should a knowledge goal be placed in memory? Since a potential answer to a knowledge goal may arrive at any time, particularly when the knowledge goal may not even be "active," the knowledge goal must

be indexed in memory exactly where the answer would be placed when it does come in. This ensures that the knowledge goal will be found without extensive searching through lists of questions. The issue of the amount of inference that should be done at this point was addressed in an earlier section.

Thus knowledge goals are indexed in memory on the basis of their concept specifications. In AQUA, these knowledge goals are used to generate expectations that guide the parser when the concepts to which they are attached are active.

4.3 Retrieving knowledge goals

When a new fact becomes known, either because it is part of the input (e.g., it is read in the story), or because it is inferred for some other reason, the reasoner needs to retrieve knowledge goals in memory that the fact could be relevant to. The knowledge goals retrieved in turn determine how useful that fact is. AQUA's knowledge goal retrieval strategies take advantage of the fact that knowledge goals are indexed on the basis of their concept specifications in an inheritance hierarchy. AQUA uses three question retrieval strategies:

Type retrieval: When a new memory structure is activated, knowledge goals indexed off the types of the concept are retrieved. The new structure is matched against the concept specification of the knowledge goal to see whether it provides the desired information. For example, if AQUA reads about a car, it retrieves questions off the "car" concept to see if the car it read about could answer any of these questions.

Relation retrieval: AQUA uses a frame-based representational scheme in which slots and slot fillers specify relations between concepts. For example, the `results` slot specifies a causal relation of a particular kind between an `action` and a `state`. Similarly, the `actor` slot in an `action` frame specifies a participatory relation between the `action` and a `volitional-agent`. Relations are themselves represented as frames in memory (e.g., see [Wilensky, 1986]), allowing AQUA to reason about the relations themselves. Knowledge goals seeking relations between concepts are indexed in the appropriate slots in the frames representing these concepts. This allows AQUA to retrieve knowledge goals that seek relations between memory structures (e.g., the connection between a given terrorist attack and the destruction of some building).

Specialization retrieval: Finally, knowledge goals may be retrieved, given an input cue, by checking whether some specialization or refinement of that input might address a knowledge goal. This allows the understanding process to be sensitive to the questions that the system is currently seeking answers to. Implementational details may be found in Ram [1989].

5 Knowledge goals as a theory of interestingness

One interesting outcome of this work is the formulation of a functional theory of *interestingness* [Ram, 1990c]. The decision to focus attention corresponds closely with the notion of "interestingness." When an understander focuses on a particular fact and processes it in greater detail, it can be said to be "interested" in that fact.[2] For this reason, focus of attention heuristics can also be thought of as *interestingness heuristics*. These heuristics provide a functional definition of "interestingness" as a criterion for focussing attention:

[2] Since interestingness depends on one's goals, the heuristics presented here do not cover interests that arise from goals that lie outside the scope of the basic understanding and learning tasks that AQUA performs. For example, a parent would be interested in the report card of his child. Since AQUA's goals do not include caring for children, it would not have any reason to be interested in a report card, unless the report card was anomalous with respect to AQUA's beliefs.

Interestingness is a guess at what one thinks one might learn from paying attention to a fact or a question. The guess must be made without processing the fact or question in detail, because otherwise the purpose of focussing attention to control inferences would be defeated. Thus the interestingness heuristics described below are indeed *heuristics* rather than precise measures of the value of thinking about a fact or a question.

This is a functional approach to the problem of interestingness [Hidi and Baird, 1986; Schank, 1979] from the perspective of our theory of knowledge goals. A similar approach can be used for reasoning systems performing other cognitive tasks, such as planning, since these systems would also need to focus their attention on inferences that were relevant to goals arising from their tasks.

In AQUA, interest in a concept is triggered by its likely relevance to questions or knowledge goals, and continuing interest is determined by its continuing significance to these goals. This is related to the "goal satisfaction principle" of [Hayes-Roth and Lesser, 1976], which states that more processing should be given to knowledge sources whose responses are most likely to satisfy processing goals, and to the "relevance principle" of [Sperber and Wilson, 1986], which states that humans pay attention only to information that seems relevant to them. These principles make sense because cognitive processes are geared to achieving a large cognitive effect for a small effort. To achieve this, the understander must focus its attention on what seems to it to be the most relevant information available [Sperber and Wilson, 1986]. The Hayes-Roth and Lesser paper prefigures the approach presented here. The additional step suggested here is to mediate the influence of processing goals on attentional decisions by using explicit characterizations of desirable knowledge. The reason for this is the multiplicity of sources of knowledge goals, and their diverse effects throughout a learning or inference system. As was made clear in the case of IVY, it is not generally possible to calculate all of the potential impacts on processing goals every time an inference is made. Knowledge goals embody the results of that calculation so that it does not have to be repeated for every new input.

AQUA's knowledge goals are used to evaluate the interestingness of various aspects of the stories being read. They also allow the system to evaluate the interestingness of its questions. Once the interestingness of the input has been determined, AQUA uses it to guide processing by focussing its resources on the more interesting aspects of the story. Since interestingness-determining heuristics are geared towards learning, this ensures that AQUA spends its time on those aspects of the story that are most likely to result in something useful being learned. Without its interestingness heuristics, AQUA would still learn the same things, but it would spend a lot more time drawing inferences that ultimately turn out to be irrelevant. Readers interested in this aspect of question-driven understanding are referred to Ram [1990c] for more details.

6 Knowledge planning: Learning through the satisfaction of knowledge goals

Knowledge goals can be used both to control inference, and to direct explicit knowledge actions, based on the metaphor of robot planning for physical goals. The generation and representation of goals to learn is only the beginning of the learning process. The theoretical justification for generating them depends on their effectiveness at constraining combinatorics of learning from complex experience. Our idea is to use AI planning techniques for making decisions about which learning actions should be taken in what order to achieve the knowledge goals of an actor situated in the world. Generally speaking, these decisions are based on knowledge about available resources, knowledge about actions and knowledge about the current state of the world (including the actor's current knowledge state). The actions that people take to acquire knowledge span a tremendous range, from looking up an answer in a reference book to designing and running scientific experiments. In order for a planner to select actions appropriate to goals, the actions must be annotated with the resources that they require, preconditions to executing the actions and expected outcomes of the actions, and perhaps information about possible alternative outcomes and relative probabilities of the alternatives.

In a system capable of taking a large number of possible actions, hierarchies of action classes can improve the combinatorics of the planning process. Classes of knowledge actions are, in effect, hypotheses about the component cognitive processes involved in learning. A proposal for a taxonomy of learning actions can be found in Hunter [1990b].

With unlimited resources, planning is trivial. Unfortunately, there are always limits. Physical planners have to manage resources like energy, money and time. Learners are similarly constrained, although the resources are different. In particular, learners have limitations on the amount of memory they have and on the amount of time they can spend on inference. Other resources may also come into play (e.g., database access may cost money), or there may be limits on the amount of network traffic a learner can generate in pursuit of information. Planners may have strict limits on resource consumption, or may merely try to avoid waste.

The question of managing resources in learning raises the issue of learning over time. Existing machine learning research has focused on learning from a particular dataset. Conversely, human-like learning occurs over an entire lifetime. Learners need to decide not only whether and what to learn, but when to learn. IVY and AQUA are able to keep "questions in the back of their mind," in the form of unsatisfied knowledge goals, which are satisfied as opportunities arise.

Learning, then, can be viewed as the incremental revision of previously existing knowledge in response to the successes and failures when using that knowledge to understand novel situations or reason about novel problems. In order to learn effectively in this manner, the reasoner needs to be able to model the gaps in its own knowledge explicitly. *It must know what it needs to know, and why.* When there is a difficulty or error in processing a novel situation, the reasoner must be able to identify the type of gap that resulted in the problem, and invoke the appropriate learning strategy to learn from the experience. For example, AQUA can (a) use domain knowledge that may not be completely understood to understand novel stories, (b) maintain an explicit model of what it needs to know to complete its understanding of the problem, i.e., of the "gaps" in its knowledge base, (c) learn by filling in these gaps when the information it needs becomes available, and hence (d) gradually evolve a better understanding of the domain [Ram, 1990b; Ram, 1992].

Thus the learning process is focussed by the knowledge goals of the system. Reading can be thought of as one type of knowledge action. More sophisticated planners might manage a complex and interacting set of learning goals and available knowledge actions, making decisions about when to pursue a particular goal, based on its relationship to the program's other learning and performance goals and on on the current state of the world. These issues are being explored further in the INVESTIGATOR [Hunter, 1990b; Hunter, 1990a] and META-AQUA [Cox and Ram, 1991; Ram and Cox, 1992] projects.

7 Comparison to other approaches

Other cognitive theories have also included reference to desires for knowledge, although there are significant differences between those prior theories and our theory of knowledge goals.

The conceptual dependency representation proposed by Schank and Abelson [1977] included D-KNOW, a goal to "change knowledge state," or to learn something. Examples of D-KNOW goals were to find out the location of food (in order to go to it and then eat it) or to find out the price of an item (in order to buy it). The generation of D-KNOW goals was always tied very specifically to a physical supergoal (e.g., satisfy hunger), and were not mentioned in the author's later theories of learning (e.g., [Schank, 1982]).

Other theories, particularly from the animal learning psychology literature, have proposed diffuse motivations to learn: a "will to perceive" (Thorpe), a "motivation for learning" (Thacker), and a "search by an information hungry organism" (Pribram, all reported in Livesey [1986], p. 20–21). As discussed above, an important feature of knowledge goals are their specificity. Merely desiring knowledge generally is not sufficient for use in focussing attention, or in other important decisions during learning.

Social psychologists have used various "goal orientations" as explanatory phenomena in theories of attention, recall and judgement. These goal orientations are close in spirit to our knowledge goals. However, social psychologists' goal orientations are generally specified at a very abstract level (e.g., "form an impression," or "make predictions"). As Zukier's [1986] review notes, "In general, however, little systematic research is available on goal orientation in inference, and no comprehensive taxonomies of 'middle-level' or concrete goals have emerged from these studies." Our work has described a much more detailed taxonomy of knowledge goal types, and proposed methods for their generation and use in processing.

Lenat's [1976] AM program had a method for focusing its attention that is related to our knowledge goals. AM maintained a queue of concepts to modify, and had a set of possible modifications that could be applied. It chose which concept and which modification based on a heuristic evaluation of the interestingness of the concept. Each concept was tagged with an interestingness number, which was used to order the queue of concepts to change. AM's search of concept space was undirected; it was not trying to learn anything in particular, and therefore cannot be said to have goals for specific knowledge. On the other hand, AM's interestingness heuristics contained an implicit characterization of what knowledge was desirable, for example, concepts with a few instances (not a single instance and not many instances). All of these characterizations were, however, syntactic; that is, they described the structure of interesting concepts, not their content. Nevertheless, this approach is more compatible with the use of knowledge goals than many systems, since, at least locally, the program made selections among various possible actions based on characterizations of the knowledge that would result from those actions.

Goal-directed planning has been investigated in the context of other reasoning tasks. For example, Tong's work on goal-directed planning in knowledge-based design addresses issues in task prioritization and inference control in the context of the design task [Tong, 1987]. Tong's focusses on the development of knowledge-based models for design, and presents a framework for organizing, evaluating and developing such models from the perspectives of the knowledge embodied in the process model, the functionality of the design process, and the implementation of the design process as an actual program. In contrast, we are interested in developing goal-directed process models, as well as content theories, for learning across a variety of different reasoning tasks. Thus our work deals with cognitive domains rather than physical domains. Inference control in Tong's model is performed through the propagation of design constraints, and attempts to maintain consistency among specification details that must be true of a solution description. Our work focusses, not so much on the truth or correctness of inferences, as on the *utility* of these inferences for reasoning and learning.

Our process of planning to learn does not involve detailed reasoning about subproblem interactions (e.g., [Sussman, 1975; Sacerdoti, 1975]) and constraint propagation (e.g., [Stefik, 1981]) that is the focus of much work in conventional planning and problem solving. Our work is closer to research in opportunistic planning (e.g., [Hayes-Roth and Hayes-Roth, 1979; Hammond, 1988]), and deals with the opportunistic pursuit of cognitive goals (e.g., [Birnbaum and Collins, 1984; Dehn, 1989]), and the use of such goals to focus inference and learning. In order to incorporate a complete model of knowledge planning using learning actions, we are currently investigating issues such as prioritization of knowledge goals, and contention for resources such as time and storage space. Although knowledge goals do not directly interfere with each other (e.g., learning one thing does not disable the preconditions for learning another), we expect there will be significant interactions between goals due to resource constraints. The extent to which least commitment approaches to goal-directed planning in physical domains (as discussed, e.g., by Tong [1987]) will generalize to planning with knowledge goals in cognitive domains remains an open issue.

Also related to our claims is the work of Horvitz, *et al* [1989]. They present a calculus for deciding when to do more inference (versus when to act) in medical decision making. Although based on highly idealized functions for estimating the expected value of additional inference (in their model, inference includes data gathering), it provides an attempt to model content-based decisions about when it is worthwhile to acquire knowledge. Although their model does not specify what is worth learning, it may be useful in deciding whether it is worth learning at all, potentially reducing the size of the potential hypothesis space to zero.

Minton [1988] also proposes a model of judging whether it is worth learning, although his model involves computing the effect of learning on future performance after the new concept is formed, and is hence not useful for focussing attention.

8 Conclusion: Automating curiosity

We view learning as an incremental process of belief formation, involving a wide variety of interrelated learning processes. Most learning systems have incomplete knowledge of their domains; these gaps give rise to difficulties during processing, and we propose that the difficulties should give rise to explicit motivations to learn. Such a system learns in an incremental manner, by noticing interesting aspects of its experiences, generating knowledge goals based on those observations, and devoting some of its resources to achieving those goals. The process of satisfying those goals generally involves the selection of both an appropriate learning method and the focussing of attention on potentially relevant information sources.

Knowledge goals specify both desired knowledge and what to do with that knowledge once it is found. The use a new piece of knowledge is put to (or, similarly, where that knowledge is stored in memory) depends on the motivation for acquiring that knowledge in the first place. In AQUA, a new piece of knowledge could result in a new explanation in memory; it could be used to fill in a gap in an existing explanation; it could be used to elaborate an existing explanation if that explanation was not detailed enough to deal with the new situation; or it could be used to reorganize or re-index knowledge in memory to allow the reasoner to use what it already knows in novel situations to which that piece of knowledge had not been applied before. In IVY, a new piece of knowledge can change provide specific information to replace a general weak method for recognition, specification or distinction; it can generate a new subclass of a disease type in memory; it can be used to change the perceived value of gathering other pieces of information during diagnosis; or it can also cause the augmentation or reorganization of existing knowledge of indicative features of a disease or other entity. Each type of learning leaves the system a little closer to a complete understanding of its domain. Each type of learning can also result in a new set of knowledge goals. The satisfaction of one goal can lead to the identification of many other pieces of information that have become useful as a result.

An important class of knowledge goals for AQUA are those that are intended to test hypotheses that the program has generated. Hypotheses often have questions attached to them, representing what is still not understood or verified about those hypotheses. As the program reads new stories, it is reminded of past cases, and of old explanations that it has tried. In attempting to apply these explanations to the new situation, it also remembers unanswered questions that it had thought of previously. The system's understanding of its cases gradually gets refined as these questions get answered. Details of this process may be found in Ram [1989; 1990b; 1992].

Much human learning seems subjectively to be a process of this type. Adults learn by modifying what they already know, using little pieces of new information as they come along. They have topics that they are interested in, and that they expend energy to pursue. People who are always asking new questions, and always on the lookout for new knowledge, are termed curious.

What would it take for a computer program to be curious? Any system that asks questions or gathers data about the world can be said to be curious in a very basic way: namely, it acts to acquire information. This kind of behavior is not a very interesting model of curiosity. Even people who are considered gluttons for knowledge do not infer everything that can be inferred; they focus on particular aspects of their environment. There is a well known psychiatric case of a person who had immense recall, but did not distinguish between relevant and irrelevant material [Luria, 1968]; his pathology is not considered a kind of wild curiosity. Human-like curiosity seems to us to require motivated pursuit of knowledge, or active learning, and not just the simple absorption of data and their consequences. Curiosity involves specific (although often abstract) desires for knowledge, not merely a diffuse drive of some sort. When to attribute goals to computer programs is a

difficult philosophical question (see, e.g., [Dennett, 1987]), but we believe that programs that make decisions about what to learn and how to learn it have taken an important step toward genuine automated curiosity.

Our intent was not to have AQUA or IVY learn the "correct" understanding of terrorism or lung cancer, but rather to be able to wonder about unusual things they encounter and be motivated by those encounters to seek out new information. As they learn more about their domains, they ask better and more sophisticated questions. We suggest that both programs can be seen as simple models of human-like curiosity, and propose that the more practical or functional reasons for goal-directed learning processes, as discussed in this paper, be viewed as an explanation for the utility of this type of behavior.

The theory of knowledge goals presented in this paper brings together both cognitively motivated processes and functionally justified resource constraints, and provides a basis for designing practical reasoning systems that can represent and reason about their own goals.

Acknowledgements

Ashwin Ram's research was supported in part by the National Science Foundation under grant IRI-9009710. Part of the research described was conducted while Dr. Ram was at Yale University, and supported by the Defense Advanced Research Projects Agency and the Office of Naval Research under contract N00014-85-K-0108, and by the Air Force Office of Scientific Research under contracts F49620-88-C-0058 and AFOSR-85-0343.

References

[Birnbaum and Collins, 1984] L. Birnbaum and G. Collins. Opportunistic Planning and Freudian Slips. In *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, pages 124–127, Boulder, CO, 1984. Institute of Cognitive Science and University of Colorado, Boulder.

[Cox and Ram, 1991] M. Cox and A. Ram. Using Introspective Reasoning to Select Learning Strategies. In R. S. Michalski and G. Tecuci, editors, *Proceedings of the First International Workshop on Multi-Strategy Learning*, pages 217–230, Harpers Ferry, WV, November 1991. Center for Artificial Intelligence, George Mason University, Fairfax, VA.

[Dehn, 1989] N. Dehn. *Computer Story Writing: The Role of Reconstructive and Dynamic Memory.* Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, 1989. Research Report #792.

[DeJong, 1979] G. F. DeJong. *Skimming Stories in Real Time: An Experiment in Integrated Understanding.* Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, May 1979. Research Report #158.

[Dennett, 1987] D. Dennett. *The Intentional Stance.* Bradford Books/MIT Press, Boston, MA, 1987.

[Dietterich, 1989] T. G. Dietterich. Limitations on Inductive Learning. In *Proceedings of Sixth International Workshop on Machine Learning*, pages 125–128, Ithaca, NY, June 1989. Morgan Kaufman.

[Doyle, 1979] J. Doyle. A Truth Maintenance System. *Artificial Intelligence*, 12:231–272, 1979.

[Dyer, 1982] M. G. Dyer. *In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension.* Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, May 1982. Research Report #116.

[Hammond, 1988] K. J. Hammond. Opportunistic Memory: Storing and Recalling Suspended Goals. In J. L. Kolodner, editor, *Proceedings of a Workshop on Case-Based Reasoning*, pages 154–168, Clearwater Beach, FL, May 1988. Morgan Kaufmann, Inc., San Mateo, CA.

[Hayes-Roth and Hayes-Roth, 1979] B. Hayes-Roth and F. Hayes-Roth. A Cognitive Model of Planning. *Cognitive Science*, 2:275–310, 1979.

[Hayes-Roth and Lesser, 1976] F. Hayes-Roth and V. Lesser. Focus of attention in a distributed logic speech understanding system. In *Proceedings of the IEEE International Conference on Accoustics, Speech and Signal Processing*, pages 416–420, Philadephia, PA, April 1976. IEEE, New York, NY.

[Hidi and Baird, 1986] S. Hidi and W. Baird. Interestingness — A Neglected Variable in Discourse Processing. *Cognitive Science*, 10:179–194, 1986.

[Hoffman *et al.*, 1981] C. Hoffman, W. Mischel, and K. Mazze. The role of purpose in the organization of information about behavior: Trait-based versus goal-based categories in person cognition. *Journal of Personality and Social Psychology*, 39:211–255, 1981.

[Horvitz *et al.*, 1989] E. Horvitz, G. Cooper, and D. Heckerman. Reflection and action under scarce resources: Theoretical principles and empirical study. Report KSL-89-1, Knowledge Systems Laboratory, Stanford University, 1989.

[Hunter, 1989] L. E. Hunter. *Knowledge Acquisition Planning: Gaining Expertise Through Experience.* Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, January 1989. Research Report #678.

[Hunter, 1990a] L. E. Hunter. Knowledge Acquisition Planning for Inference from Large Datasets. In B. D. Shriver, editor, *Proceedings of the Twenty Third Annual Hawaii International Conference on System Sciences*, pages 35–45, Kona, HI, 1990. IEEE Computer Society Press, Los Alamitos, CA.

[Hunter, 1990b] L. E. Hunter. Planning to Learn. In *Proceedings of the Twelvth Annual Conference of the Cognitive Science Society*, pages 26–34, Boston, MA, July 1990.

[Kass *et al.*, 1986] A. Kass, D. Leake, and C. Owens. *SWALE: A Program That Explains*, pages 232–254. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[Lenat, 1976] D. B. Lenat. *A.M.: An artificial intelligence approach to discovery in mathematics as heuristic search.* Ph.D. thesis, Stanford University, Artificial Intelligence Laboratory, 1976.

[Livesey, 1986] P. Livesey. *Learning and emotion: A biological synthesis*, volume 1 of *Evolutionary Processes.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[Luria, 1968] A. Luria. *The Mind of a Mnemonist.* New York, 1968.

[Minton, 1988] S. Minton. *Learning effective search control knowledge: An explanation-based approach.* Ph.D. thesis, Carnegie-Mellon University, Computer Science Department, Pittsburgh, PA, 1988. Technical Report CMU-CS-88-133.

[Ram and Cox, 1992] A. Ram and M. Cox. Introspective Reasoning using Meta-Explanations for Multistrategy Learning. In R. Michalski and G. Tecuci, editors, *Machine Learning IV: A Multistrategy Approach.* Morgan Kaufman Publishers, Inc., 1992. In preparation.

[Ram and Leake, 1991] A. Ram and D. Leake. Evaluation of Explanatory Hypotheses. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, Chicago, IL, August 1991.

[Ram, 1987] A. Ram. AQUA: Asking Questions and Understanding Answers. In *Proceedings of the Sixth Annual National Conference on Artificial Intelligence*, pages 312–316, Seattle, WA, July 1987. Morgan Kaufman Publishers, Inc., Los Altos, CA.

[Ram, 1989] A. Ram. *Question-driven understanding: An integrated theory of story understanding, memory and learning.* Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, May 1989. Research Report #710.

[Ram, 1990a] A. Ram. Decision Models: A Theory of Volitional Explanation. In *Proceedings of the Twelvth Annual Conference of the Cognitive Science Society*, pages 198–205, Cambridge, MA, July 1990. Lawrence Erlbaum Associates.

[Ram, 1990b] A. Ram. Incremental Learning of Explanation Patterns and their Indices. In B. W. Porter and R. J. Mooney, editors, *Proceedings of the Seventh International Conference on Machine Learning*, pages 313–320, Austin, TX, June 1990. Morgan Kaufman Publishers, Inc.

[Ram, 1990c] A. Ram. Knowledge Goals: A Theory of Interestingness. In *Proceedings of the Twelvth Annual Conference of the Cognitive Science Society*, pages 206–214, Cambridge, MA, July 1990. Lawrence Erlbaum Associates.

[Ram, 1991] A. Ram. A Theory of Questions and Question Asking. *The Journal of the Learning Sciences*, 1(3&4), 1991. In press.

[Ram, 1992] A. Ram. Indexing, Elaboration and Refinement: Incremental Learning of Explanatory Cases. *Machine Learning*, 1992. To appear. Also available as Technical Report GIT-CC-92/03, College of Computing, Georgia Institute of Technology, Atlanta, GA.

[Rieger, 1975] C. Rieger. Conceptual Memory and Inference. In R. C. Schank, editor, *Conceptual Information Processing*. North-Holland, Amsterdam, 1975.

[Sacerdoti, 1975] E. D. Sacerdoti. A structure for plans and behavior. Technical Report 109, Stanford Research Institute, Artificial Intelligence Center, 1975.

[Schank and Abelson, 1977] R. C. Schank and R. Abelson. *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.

[Schank, 1979] R. C. Schank. Interestingness: Controlling Inferences. *Artificial Intelligence*, 12:273–297, 1979.

[Schank, 1982] R. C. Schank. *Dynamic Memory: A Theory of Learning in Computers and People*. Cambridge University Press, New York, NY, 1982.

[Schank, 1986] R. C. Schank. *Explanation Patterns: Understanding Mechanically and Creatively*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[Sperber and Wilson, 1986] D. Sperber and D. Wilson. *Relevance: Communication and Cognition*. Language and Thought Series. Harvard University Press, Cambridge, MA, 1986.

[Srull and Wyer, 1986] T. Srull and R. Wyer. The Role of Chronic and Temporary Goals in Social Information Processing. In R. Sorrentino and E. Higgins, editors, *Handbook of Motivation and Cognition: Foundations of Social Behavior*, pages 503–549. The Guilford Press, Guilford, CT, 1986.

[Stefik, 1981] M. J. Stefik. Planning with constraints (MOLGEN: Part 1). *Artificial Intelligence*, 16(2):111–140, 1981.

[Sussman, 1975] G. J. Sussman. *A Computer Model Of Skill Acquisition*, volume 1 of *Artificial Intelligence Series*. American Elsevier, New York, 1975.

[Tong, 1987] C. Tong. Towards An Engineering Science Of Knowledge-Based Design. Ai/Vlsi Project Working Paper 49, Rutgers University, Department Of Computer Science, New Brunswick, Nj, 1987.

[Utgoff, 1986] P. Utgoff. Shift Of Bias For Inductive Concept Learning. In R. S. Michalshi, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning*, page 107. Morgan Kaufman, Los Altos, Ca, 1986.

[Wilensky, 1986] R. Wilensky. Knowledge Representation — A Critique and A Proposal. In J. L. Kolodner and C. K. Riesbeck, editors, *Experience, Memory and Reasoning*, chapter 2, pages 15–28. Lawrence Erlbaum Associates, Hilldale, NJ, 1986.

[Yesner and Carter, 1982] R. Yesner and D. Carter. Pathology of Carcinoma of the Lung: Changing Patterns. *Clinics in Chest Medicine*, 3(2):257–289, 1982.

[Zukier, 1986] H. Zukier. The Paradigmatic and Narrative Modes in Goal-Guided Inference. In R. Sorrentino and E. Higgins, editors, *Handbook of Motivation and Cognition: Foundations of Social Behavior*, pages 465–502. Guilford Press, Guilford, CT, 1986.