Indexing, Elaboration and Refinement: Incremental Learning of Explanatory Cases*

Ashwin Ram

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

Abstract

This article describes how a reasoner can improve its understanding of an incompletely understood domain through the application of what it already knows to novel problems in that domain. Case-based reasoning is the process of using past experiences stored in the reasoner's memory to understand novel situations or solve novel problems. However, this process assumes that past experiences are well understood and provide good "lessons" to be used for future situations. This assumption is usually false when one is learning about a novel domain, since situations encountered previously in this domain might not have been understood completely. Furthermore, the reasoner may not even have a case that adequately deals with the new situation, or may not be able to access the case using existing indices. We present a theory of incremental learning based on the revision of previously existing case knowledge in response to experiences in such situations. The theory has been implemented in a case-based story understanding program that can (a) learn a new case in situations where no case already exists, (b) learn how to index the case in memory, and (c) incrementally refine its understanding of the case by using it to reason about new situations, thus evolving a better understanding of its domain through experience. This research complements work in case-based reasoning by providing mechanisms by which a case library can be automatically built for use by a case-based reasoning program.

Machine Learning, 10(3):201–248, 1993.

1 Case-based learning

Case-based reasoning programs deal with the issue of using past experiences or *cases* to understand, plan for, or learn from novel situations (e.g., see [Kolodner, 1988; Hammond, 1989]). This happens according to the following process: (a) Use the problem description (problems, anomalies, goals) to get reminded of the old case; (b) Retrieve the results (lessons, explanations, plans) of processing the old case and give them to the problem-solver, understander or planner; (c) Adapt the results from the old case to the specifics of the new situation; and (d) Apply the adapted results to the new situation. In some case-based reasoning programs, there is a further step (e) in which the old and new solutions are generalized to increase the applicability of the solutions.

The intent behind case-based reasoning is to avoid the effort involved in re-deriving these lessons, explanations or plans by simply reusing the results from previous cases. However, this process relies on three assumptions: (1) A case is available for the situation at hand; (2) The case is correctly indexed in memory so that it can be retrieved for use in the current situation using the cues that the situation provides; and (3) The case is well understood and provides good "lessons" to be used in this and future situations.

In other words, case-based reasoning programs rely on the existence of a case library that can provide the right cases when they are needed, since it is these very cases that determine the performance of the system in new situations. This assumption is usually false when one is learning about a novel domain, since cases encountered previously in this domain might not have been understood completely. Instead, it is reasonable to assume that the reasoner would have gaps in the domain knowledge represented by its cases. This could happen in three ways, corresponding to the three assumptions above:

1. **Novel situation:** In a truly novel situation, an applicable case may not be available. The reasoner simply does not have a prior experience that provides it with a case that is relevant to the current situation.
2. **Mis-indexed cases:** The reasoner may have a case that is applicable to the current situation, but may be unable to retrieve it since the case is not indexed under the cues that the situation provided.
3. **Incorrect or incompletely understood cases:** Previous experiences, especially in novel and complex domains, may not have been completely understood, and so cases corresponding to them may be incomplete or incorrect.

This article addresses a fundamental problem in case-based reasoning: how can a case library for a novel domain be built automatically through experience despite the existence of such gaps? We propose that learning consists of the incremental revision of previously existing case knowledge in response to successes and failures when using that knowledge in case-based reasoning in complex and ill-understood domains. We present a theory of incremental learning of explanatory case knowledge, and discuss the types of knowledge that can be acquired when the reasoner encounters a gap in its case knowledge during case-based reasoning.

We argue that past cases, even if not completely understood, can still be used to guide processing in new situations. However, in addition to using the past case to understand the *new* situation, a reasoner can also learn more about the *old* case itself, and thus improve its understanding of the domain. This is an important problem that has not been addressed in case-based reasoning research, and one that is suited to a machine learning approach in which learning occurs incrementally as gaps in the reasoner's case library are filled in through experience. Learning may also occur through the learning of new indices for old cases, as the reasoner discovers new contexts in which its cases are applicable.

As mentioned earlier, some case-based learning programs do include a learning step in which the results of case application are generalized (e.g., [Kass and Owens, 1988]). This gives these programs the ability to improve their case knowledge in an incremental fashion. However, in most programs this involves the

generalization of cases that embody correct and complete solutions to past or current problems. Thus these programs do not get around the problem of reasoning with incompletely understood cases.

Furthermore, these programs do not deal with the issue of where the initial cases come from. In novel and poorly understood domains there will always be situations for which there are no cases in memory. A reasoner cannot expect to have an experience relevant to every possible situation that might be encountered. In such situations, the reasoner must reason from scratch based on general knowledge. At the end of the process, however, the reasoner is left with a new case representing this specific experience. This case, in turn, is modified further as it is used in future situations.

1.1 Outline

This article presents a case-based story understanding system that improves its case library through experience. The article is organized into four parts. Part I (case representation) deals with the nature of explanatory cases in an understanding system, and discusses how cases are represented, indexed in memory, and used in case-based explanation. The case representation for explanation-based tasks, including story understanding, depends on a representational theory of causal explanations. Any such theory has two parts, *representational structure* and *representational content*. Although the content of our representations are specific to our domain (understanding motivations of terrorists), the theory of learning can be applied to any domain in which causality can be represented using explanatory cases and abstract schemas with the causal structure that we describe here.

Part II (case learning) deals with the learning of a new, and perhaps partially understood, case in a situation for which no case previously existed. We introduce a learning method called explanation-based refinement, a type of explanation-based learning in which causal constraints from a novel story are used to specialize or refine an existing explanation schema (which may be very general) to a more specific schema that applies to the kind of situation that the story is about. Whereas the specialized schema is less widely applicable than the abstract schema that the reasoner started with, it is easier to recognize and provides a better and more detailed explanation for the specific type of situation that it applies to. Such specialized causal schemas are precisely the explanatory cases used in case-based explanation when reasoning about new situations.

Finally, we introduce explanation-based methods for index learning, and for the incremental modification of cases. Part III (incremental case modification) discusses the incremental modification of incompletely understood cases, including newly learned cases, through their use in understanding stories about novel situations. This is done through the generation of questions (the system's representation of what it needs to know to complete its understanding of the case), and the answering of these questions during future situations through case-based explanation with cases about which previous questions are pending. The question-answering process may be incidental to the case-based explanation task, or, as is the case in our system which reads for the purpose of learning, the very focus of this task. Part IV (index learning) discusses the learning of new indices for cases. Index learning is a special kind of question answering in which the system attempts to answer a question of the type "In which types of situations is such a case likely to be applicable?" We discuss how a system can learn new indices for cases that it already has by using them in novel contexts.

Together, the learning techniques result in a program that can gradually evolve a better understanding of its domain through experience, even in the presence of the three problems mentioned earlier. The theory is illustrated with a series of examples demonstrating the improved behavior of the program on input stories that it could not understand adequately at the outset. We conclude with a discussion of the strengths and weaknesses of our approach.

1.2 The AQUA system

The theory presented here has been implemented in the AQUA system, a story understanding program that learns about terrorism by reading newspaper stories about unusual terrorist incidents.[1] AQUA retrieves past explanations from situations already represented in memory, and uses them to build explanations to understand novel stories about terrorism. In doing so, the system refines its understanding of the domain by filling in gaps in these explanations, by elaborating the explanations, by learning new indices for the explanations, or by specializing abstract explanations to form new explanations for specific situations. This is a type of *incremental learning*, since the system improves its explanatory knowledge of the domain in an incremental fashion rather than by learning complete new explanations from scratch.

The performance task in AQUA is to "understand" human-interest stories about terrorist acts, i.e., to construct explanations of the actions observed in the story that causally relate the actions to the goals, plans, and beliefs of the actors and planners of the actions. Such an explanation is called a *volitional explanation*, and the process of constructing these explanations is called *motivational analysis*. In general, an explanation consists of several inference rules connected together into a graph structure with several antecedents and one or more consequents. Construction of such explanations is typically done by chaining together inference rules through a search process (e.g., [Rieger, 1975; Wilensky, 1981; Morris and O'Rorke, 1990]), through a weighted or cost-based search (e.g., [Hobbs *et al.*, 1990; Stickel, 1990]), or through a case-based reasoning process in which previous explanations for similar situations are retrieved and adapted for the current situation (e.g., [Schank, 1986; Kass *et al.*, 1986; Ram, 1989; Ram, 1990a]).

The latter method, which is the basis for AQUA's approach to motivational analysis, is similar to the use of explanation schemas to build explanations (e.g., [Mooney and DeJong, 1985]) since it relies on the instantiation of "large" knowledge structures (cases or schemas) rather than the chaining together of "small" knowledge structures (inference rules). Rather than defend the case-based reasoning approach here, we will simply state the assumptions implicit in this approach:[2]

> **A-1: Efficiency assumption:** It is more efficient to retrieve and apply larger knowledge structures than to construct them from scratch out of smaller knowledge structures or inference rules each time.
>
> **A-2: Content assumption:** There are too many possible ways in which inference rules can be connected together, many of which will be irrelevant or meaningless. The content of the explanations produced from cases is likely to be better than those produced through exhaustive search through inference rules, because cases contain experiential knowledge about the ways in which the rules are actually connected together in real situations.
>
> **A-3: Typicality assumption:** Situations encountered in the real world are typical of the kinds of situations that are likely to be encountered in the future. Thus it is worthwhile creating a new case to represent novel experiences, because remembering this case will make it easier (by virtue of A-1 and A-2) to process similar situations in the future.

The performance of a case-based reasoning clearly depends on having the right cases indexed in memory in the right ways. This article focusses on methods for learning cases and indices through experience. Before elaborating further on what this entails, let us consider a few examples. Consider the following story (*New York Times*, Nov 27, 1985, page A9) from the domain of the AQUA program:

[1] AQUA stands for "Asking Questions and Understanding Answers." This article focusses on the learning aspects of AQUA. Further details of this program may be found in Ram [1989; 1991].

[2] While to our knowledge these assumptions have not been stated explicitly in this manner, most case-based reasoning approaches described in the literature do make these assumptions.

S-1: Suicide bomber strikes Israeli post in Lebanon.

SIDON, Lebanon, November 26 — A teenage girl exploded a car bomb at a joint post of Israeli troops and pro-Israeli militiamen in southern Lebanon today, killing herself and causing a number of casualties, Lebanese security sources said. ...

A statement by the pro-Syrian Arab Baath Part named the bomber as Hamida Mustafa al-Taher, born in Syria in 1968. The statement said she had detonated a car rigged with 660 points of explosives in a military base for 50 South Lebanon Army men and Israeli intelligence and their vehicles.

Suppose that AQUA has never encountered a suicide bombing story before. This story is unusual because it involves unusual goal priorities:

S-2: Why was Hamida willing to sacrifice her life in order to destroy the Israeli military base?

The explanation here is one that most people are familiar with: "Because she was a religious fanatic." However, when reading about suicide bombing for the first time, it is reasonable to assume that the system does not know about religious fanaticism to begin with. The system must fall back on its general knowledge (in this case, about the sacrificing of one goal for another) to understand the story. Once this is done, however, the system can learn its first case of religious fanaticism, and use it to process future stories about religious fanaticism.

In the above example, the system did not have a previous case that dealt with the problem at hand. Now consider the same system (by now reasonably expert in religious fanaticism) reading the following story (New York Times, April 14, 1985):

S-3: Boy Says Lebanese Recruited Him as Car Bomber.

JERUSALEM, April 13 — A 16-year-old Lebanese was captured by Israeli troops hours before he was supposed to get into an explosive-laden car and go on a suicide bombing mission to blow up the Israeli Army headquarters in Lebanon. ...

What seems most striking about [Mohammed] Burro's account is that although he is a Shiite Moslem, he comes from a secular family background. He spent his free time not in prayer, he said, but riding his motorcycle and playing pinball. According to his account, he was not a fanatic who wanted to kill himself in the cause of Islam or anti-Zionism, but was recruited for the suicide mission through another means: blackmail.

Let us now assume that the system does know about suicide bombing, religious fanaticism, blackmail, and so on. However, story S-3 is novel, not because one has never heard of blackmail, but because one has never seen it used in this context before. One usually does not think of blackmail when reading a story about suicide bombing. Many situations involve novel uses of known cases, and it is unreasonable to expect a reasoner's cases to be correctly indexed for all possible situations in which they are likely to be applicable. In this example, AQUA learns a new index for blackmail to allow it to retrieve this explanation in such situations in the future.

These examples illustrate two of the problems mentioned earlier, those of missing cases and mis-indexed cases. The final problem is that of incompletely understood cases. Although in the above example AQUA has learned a new case involving a novel use of blackmail, its understanding of the new case is incomplete. The explanation for the bomber's actions has a gap in it: "What could the bomber want more than his own life?" This gap corresponds to an unanswered question, a missing piece of the causal chain underlying the explanation of the bomber's motivations. The third type of learning presented in this article involves the incremental elaboration of incompletely understood cases through the generation and answering of questions.

In general, any learning system, whether case-based or otherwise, would have incomplete knowledge of its domain, since by definition it is still learning about its domain. In a case-based system, incomplete

domain knowledge is manifested through missing, mis-indexed or incompletely understood cases, each of which could lead to poorer performance. In this article, we are concerned with methods for improving the quality of the explanations produced by a case-based understanding system such as AQUA. Since it is difficult to measure the quality of volitional explanations in a quantitative manner, the performance of the system is determined by the quality of its output rather than by quantitative measures such as the speed of explanation construction. We show improved performance of our system by demonstrating qualitative improvement in the range of stories that it can understand adequately, as well as the depth of its understanding of these stories.

Part I

Case representation

2 Explanation patterns: The nature of explanatory cases

Before we can discuss the learning process, we must describe what needs to be learned. This in turn depends on the purpose to which the learned knowledge will be put. In other words, the task of motivational analysis for story understanding, and the case-based explanation method for performing this task, impose functional constraints on the nature of cases and indices to be learned. Although the details of the task and method, and justifications for the representations, are outside the scope of this article, we will present those aspects that are required to understand the assumptions underlying the learning theory.

The need for an explanation arises when some observed situation does not quite fit into the reasoner's world model, i.e., the reasoner detects an *anomaly*. An explanation is a causal chain that demonstrates why the anomalous proposition might have occurred by introducing a set of premises that causally lead up to that proposition. If the reasoner believes the premises, the proposition ceases to be anomalous since the causal interactions underlying the situation can now be understood.

An explanation-based understanding system must be able to detect anomalies in the input, and to resolve them by building volitional 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. This process characterizes both "story understanders" that try to achieve a deep understanding of the stories that they read, as well as programs that need to understand their domains in service of other problem-solving tasks. Explanations represent *causality*, or sets of causal relationships between the basic elements of the domain. For example, a bombing action results in the destruction of the target of the bombing. This is represented using a causal relation of type `physically-results` between the `bombing` (the antecedent of the relation) and the `destroyed-state` of the `target` (the consequent of the relation).[3] Such a relation is called a *primitive inference rule* because it is an inference rule with no internal structure. There is no further explanation of how or why bombing physically results in the destruction of the target.

An explanation typically consists of several primitive inference rules that, when connected together, provide a causal description of how the antecedents of the explanation led up to the consequent(s). In a case-based explanation system, such explanations are constructed by remembering previous cases with known explanations (called *explanatory cases*), and using the cases as a basis for constructing new explanations. Explanatory cases represent standard patterns of causality that are observed in previously encountered situations, and are represented using *explanation patterns* [Schank, 1986]. When the understander sees a

[3] We use the mathematical terms `domain` and `co-domain` to refer to the "left-hand side" and "right-hand side" of a relation. `Typewriter font` represents actual vocabulary items used by the AQUA program. Further details of the representation may be found in Ram [1989].

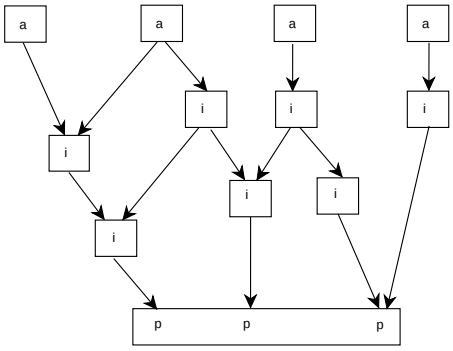


Figure 1: The structure of explanation patterns. Nodes labelled a represent XP-ASSERTED-NODES, corresponding to the premises of the explanation that is built by instantiating the XP. Nodes labelled i are INTERNAL-XP-NODES, and those labelled p are PRE-XP-NODES (the conclusions of the explanation). INTERNAL-XP-NODES and LINKS taken together compose the internal structure of the explanation. XPs with no internal structure are equivalent to primitive inference rules.

situation for which it has an explanation pattern (XP), it tries to apply the XP to avoid detailed analysis of the situation from scratch. Thus an XP represents a generalization based on the understander's experiences that can be used as a basis for reasoning about similar situations in the future.

Our main contribution to Schank's theory of explanation patterns is that the case-based explanation process in AQUA, while similar to that used by the SWALE program [Kass *et al.*, 1986], is formulated in a question-based framework. Our emphasis is on the questions that underly the creation, verification, and learning of explanations, and not on the creative adaptation process described by Kass *et al.* Furthermore, we focus on the use of possibly incomplete XPs with questions attached to them, and the learning that occurs as these questions are answered. We also focus on the learning of indices for XPs.

To support these enhancements, we introduce a graph-based representation of the structure of explanation patterns. Finally, to enable us to apply our theory to the task of motivational analysis, we propose a content theory of volitional explanations that serves as the basis for the explanatory cases used in AQUA. The content theory is based on the theory of *decision models*, which describe the planning process that an agent goes through when considering whether to perform an action. Although we use decision models as our domain for learning, the theory of learning can be extended to any domain in which causality can be represented using explanatory cases of the type that we describe here.

2.1 The structure of explanation patterns

Explanation patterns in AQUA have four components (figure 1):

- **PRE-XP-NODES:** Nodes that represent what is known before the XP is applied. One of these nodes, the EXPLAINS node, represents the particular action being explained.
- **XP-ASSERTED-NODES:** Nodes asserted by the XP as the explanation for the EXPLAINS node. These compose the premises of the explanation.
- **INTERNAL-XP-NODES:** Internal nodes asserted by the XP in order to link the XP-ASSERTED-NODES to the EXPLAINS node.
- **LINKS:** Causal links asserted by the XP. These taken together with the INTERNAL-XP-NODES are also called the internals of the XP.

An explanation pattern is a directed, acyclic graph of conceptual nodes connected with causal LINKS, which in turn could invoke further XPs at the next level of detail. The PRE-XP-NODES are the sink nodes (consequences) of the graph, and the XP-ASSERTED-NODES are the source nodes (antecedents or premises). The difference between XP-ASSERTED-NODES and INTERNAL-XP-NODES is that the former are merely asserted by the XP without further explanation, whereas the latter have causal antecedents within the XP itself. An XP applies when the EXPLAINS node matches the concept being explained and the PRE-XP-NODES are in the current set of beliefs. The resulting hypothesis is confirmed when all the XP-ASSERTED-NODES are verified.

Ultimately, the graph structure underlying an XP bottoms out in primitive inference rules of the type used by MARGIE [Rieger, 1975] or PAM [Wilensky, 1978]. Schank [1986] describes XPs as the "scripts" of the explanation domain. Unlike scripts, however, XPs are flexible in the sense that their internal structure allows them to be useful in novel situations, while retaining the advantages of pre-stored structures in stereotypical situations. Access to an XP's causal internals is essential to the incremental learning process described later.

2.2 Domain theory: The content of explanation patterns

The particular content of the causal knowledge represented in explanation patterns depends, of course, on the domain of interest. AQUA deals with volitional explanations, which link actions that people perform to their goals and beliefs, yielding an understanding of the motivations of the characters. For example, in the suicide bombing story S-1, the understander needs to explain why the girl performed an action that led to her own death. An explanation for this anomaly, such as the religious fanatic explanation, must provide a motivational analysis of the reasons for committing suicide.

AQUA has two broad categories of explanatory knowledge:

1. **Abstract explanation schemas** for why people do things. These are standard high-level explanations for actions, such as "Actor does action because the outcome of the action satisfies a goal of the actor."

2. **Explanatory cases.** These are specific explanations for particular situations, such as "Shiite Moslem religious fanatic goes on suicide bombing mission."

For example, an explanation of type 1 for story S-1 might be "Because she wanted to destroy the Israeli base more than she wanted to stay alive." An explanation of type 2 would be simply "Because she was a religious fanatic." The internal causal structure of the latter explanation could then be elaborated to provide a detailed motivational analysis in terms of explanations of the first type if necessary.

Both types of explanatory knowledge are represented using volitional XPs with the internal structure discussed in the previous section. Volitional XPs relate the actions in which the characters in a story are involved to the *outcomes* that those actions had for them, the *goals*, *beliefs*, *emotional states*, and *social states* of the characters as well as priorities or *orderings* among the goals, and the *decision process* that the characters go through in *considering* their goals, goal-orderings, and likely outcomes of the actions before deciding whether to perform those actions. A volitional explanation involving the planning decisions of a character is called a *decision model* [Ram, 1990a]. A detailed example, showing the representation of the religious fanatic explanation pattern, is shown in figures 2 and 3.

Further details of the particular representations used by the AQUA program are irrelevant for the purposes of this article. We discuss them only as an example of the kinds of explanatory structures that underly cases in causal domains. What is essential is that the domain be describable using patterns of causality represented using graph structures with the four components discussed in section 2.

1. **Explains (PRE-XP-NODES):**
   Why `volitional-agent` A did a `suicide-bombing` M, with `results` =

   (BS) `death-state` of A

   (GS) `destroyed-state` of `target`, a `physical-object` whose `owner` is an opponent religious group.

2. **Premises (XP-ASSERTED-NODES):**

   (1) A believes in the `religion` R.

   (2) A is a `religious-fanatic`, i.e., A has `high-religious-zeal`.

3. **Internals (LINKS and INTERNAL-XP-NODES):**

   (1) A is religious and believes in the `religion` R (an `emotional-state`, perhaps caused by a `social-state`, such as upbringing).

   (2) A is strongly zealous about R (an `emotional-state` represented as `high-religious-zeal`).

   (3) A wants to spread his religion R (a `goal`, initiated by (1) and (2)).

   (4) A places a high priority on his goal in (3), and is willing to sacrifice other goals which we would normally place above the religion goal (a `goal-ordering`, initiated by (1) and (2)).

   (5) A believes that performing a suicide bombing against opponent religious groups will help him achieve his goal in (3) (a belief or `expected-outcome`).

   (6) A knows that the performance of a suicide bombing may result in a negative outcome for him (an `expected-outcome`).

   (7) A weighs his goals (3), goal-orderings (4), and likely outcomes (5) and (6) (a `consideration`).

   (8) A decides to do the suicide bombing M (a `chooses-to-enter` decision, based on the considerations in (7)).

   (9) A does the suicide bombing M (an action or `mop`, whose `actor` is A).

   (10) The suicide bombing has some outcome for A, which is either positive or negative as viewed from the point of view of A's goals and goal-orderings (a `self-outcome`).

Figure 2: An English description of the religious fanatic explanation pattern shown in figure 3. Concepts in `typewriter font`, and the labels A, M, GS, and BS, correspond to the representational elements of figure 3.

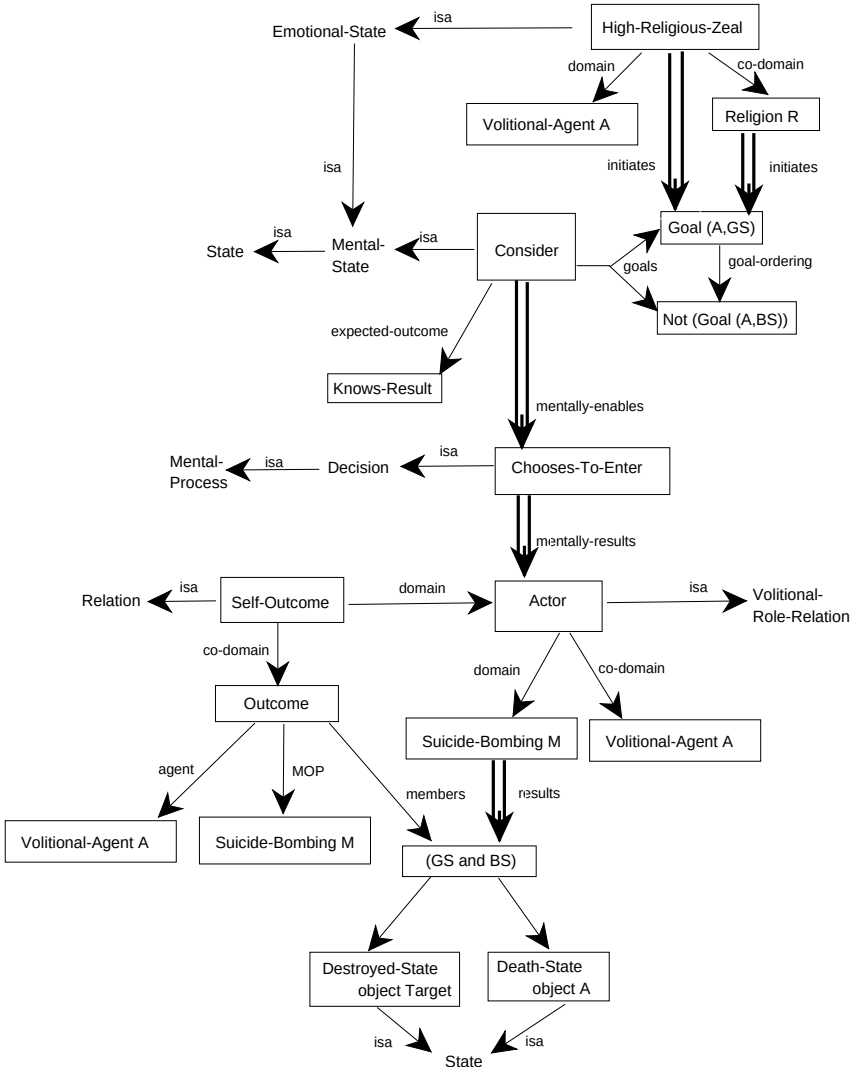


Figure 3: Representation of the religious fanatic explanation pattern, `xp-religious-fanatic`. A is the agent, R his religion, M the action he chooses to do, and GS and BS the good and bad outcomes for A as a result of doing that action. A considers his goals to achieve GS and to prevent BS, the relative priorities of the two goals, and volitionally chooses to perform (`chooses-to-enter`) M knowing both expected outcomes of M, the `death-state` of A and the `destroyed-state` of the `target`. Double arrows highlight the main elements of the causal chain comprising the volitional explanation.

2.3 Process model for case-based explanation

The process of case-based explanation consists of the following steps (see table 1). The input to the process is a `volitional-role-relation`, which is defined as a relation between an action or MOP (the `domain` of the relation) and a `volitional-agent` (the `co-domain` of the relation). The relation represents the fact that the agent is the `actor` or `planner` of an action (the two types of facts that require motivational analysis).

Anomaly detection: Anomaly detection refers to the process of identifying an unusual fact or situation description that needs explanation. The fact may be unusual in the sense that it violates or contradicts some piece of information in memory. Alternatively, the fact may be unusual because, while there is no explicit contradiction, the reasoner fails to integrate the fact satisfactorily in its memory. Anomaly detection in AQUA is done through a series of anomaly detection questions based on the goals, goal-orderings, plans, beliefs and decisions represented in AQUA's decision models [Ram, 1991]. For example, the question "Did the actor want the outcome of his action?" allows AQUA to notice a `goal-violation` anomaly in which an agent performs an action that violates the agent's own goals.

Details of the anomaly detection process are not relevant to this article, which focusses on the learning aspects of AQUA. Here, we may assume that the anomaly a is detected by an external anomaly detection algorithm. The anomaly index I_a is determined by looking up a in a table associating anomalies with abstract explanation schemas that form anomaly category indices for XPs representing specific explanatory cases. For example, the anomaly `goal-violation` is associated with the abstract XPs `xp-not-know-outcome` and `xp-goal-sacrifice`. More details of XP-based approaches to anomaly detection may be found in Ram [1989; 1991] and Leake [1989a; 1989b].

Explanation pattern retrieval: When faced with an anomalous situation, AQUA tries to retrieve one or more previously known explanatory cases or, if no cases are available, abstract explanation schemas that would explain the situation. An applicable XP is one whose PRE-XP-NODES can be unified with the current situation, with the EXPLAINS node being unified with the particular action being explained. Since it is computationally infeasible to match the PRE-XP-NODES of every XP with every action being explained, AQUA uses a set of indices as a heuristic to identify potentially relevant explanatory cases. Learning the right indices for an XP is therefore an important component of AQUA's learning process.

In general, XPs are indexed by stereotypical descriptions of their EXPLAINS nodes, and a description of the anomaly to be explained. For example, in order to explain an action M performed by a volitional agent A, AQUA uses three types of indices to retrieve potentially relevant XPs: (1) I_a, the anomaly category index, which identifies classes of XPs relevant to the given anomaly a, (2) I_s, the situation index, which identifies XPs relevant to a particular situation (action or MOP) M, and (3) I_c, the character stereotype index, which identifies XPs relevant to a particular stereotype that the agent A can be viewed as. These are described in more detail later.

Explanation pattern application: Once a set of potentially applicable XPs is retrieved, AQUA tries to use them to resolve the anomaly. This involves instantiating the INTERNAL-XP-NODES and LINKS of each XP, and filling in the details through elaboration and specification. The PRE-XP-NODES of the XP are merged with corresponding nodes in the story representation. The instantiated XP is called an *explanatory hypothesis*, or simply hypothesis (labelled H in table 1). If there are gaps in the XP, represented as pending questions attached to the XP, the questions are instantiated and the story representation is checked to see if the questions can be answered.

Input: R, a volitional role relation (`actor`, `planner`) between an action or MOP M and a volitional agent A. By definition, A appears in the R slot of M.

Output: T, a hypothesis tree (see figure 4).

Algorithm:

- Invoke anomaly detection algorithm to determine whether R is anomalous (see accompanying discussion).
- If so, create a root node for T and place the anomaly a at the root.
- Identify the set of anomaly indices $\{I_a\}$ based on the anomaly a.
- Determine the set of situation indices $\{I_s\}$ by retrieving abstractions of M.
- Determine the set of character stereotype indices $\{I_c\}$ by matching A to known character stereotypes.
- $\forall$ $\{I_a, I_s, I_c\}$ combinations, retrieve any explanation pattern XP that is indexed by this combination (explanation pattern retrieval). This provides the set of potentially applicable explanation patterns $\{XP\}$.
- $\forall$ XP in this set $\{XP\}$, match the EXPLAINS node of XP to R. Retain the set of applicable explanation patterns $\{XP\}$ for which this match succeeds.
- $\forall$ XP in the new set $\{XP\}$, create hypotheses H as follows (explanation pattern application):
  - instantiate XP
  - unify EXPLAINS node of XP with R
  - instantiate INTERNAL-XP-NODES and LINKS of XP
  - instantiate pending questions attached to XP, if any
  - create a new node in T to represent the hypothesis H and attach it as a child of the root node representing the anomaly a.
- $\forall$ H in the set of hypotheses, verify H as follows (hypothesis verification):
  - instantiate the XP-ASSERTED-NODES n of the XP that was instantiated to form H
  - create a hypothesis verification question HVQ from each n that is not already known to be true in the story
  - create a new node in T for each HVQ of H and attach it as a child of the node representing H
  - invoke hypothesis evaluation algorithm to determine current best hypothesis (see accompanying discussion).
- When all the HVQs of any hypothesis H are verified (question answering), verify the hypothesis H and refute its competitors. Note that questions may be answered later while processing this or other stories.

Table 1: AQUA's case-based explanation algorithm.

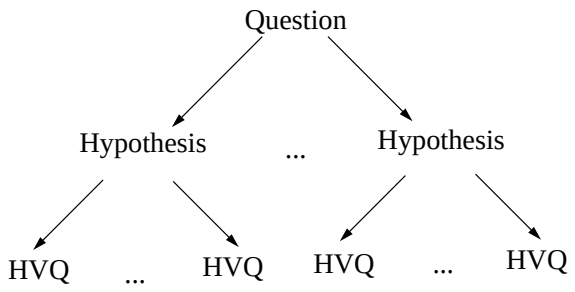


Figure 4: The structure of a hypothesis tree.

Hypothesis verification: The final step in the explanation process is the confirmation or refutation of possible explanations, or, if there is more than one hypothesis, discrimination between the alternatives. A hypothesis is a causal graph that connects the premises of the explanation to the conclusions via a set of intermediate assertions. The premises of the explanation are the XP-ASSERTED-NODES of the XP. XP-ASSERTED-NODES which assert facts that are not already known to be true in the story are turned into hypothesis verification questions (HVQs) for the hypothesis. If all the HVQs are confirmed, the hypothesis is confirmed (and its competitors refuted). If any HVQ is disconfirmed, the hypothesis is refuted.

The reasoner may use other methods for evaluating candidate hypotheses as well. Ram and Leake [1991] discuss several explanation evaluation methods, including those used by AQUA. Details of these methods are outside the scope of this article; here, we assume that an external algorithm is invoked to evaluate candidate hypotheses. At the end of this step, the reasoner is left with one or more alternative hypotheses. Partially confirmed hypotheses are maintained in a data dependency network called a *hypothesis tree* (labelled T in table 1), along with questions representing what is required to verify these hypotheses (figure 4).

2.4 Representation of questions

Questions in AQUA's memory represent gaps in AQUA's model of the domain. These questions serve as *knowledge goals*, the system's goals to acquire knowledge in order to learn more about the domain. Some questions, such as the ones in figure 4, arise from unconfirmed hypotheses that the system is entertaining, or has entertained in a previous story. Other questions arise from other kinds of gaps in the system's knowledge or other kinds of difficulties during processing. Our theory of questions and knowledge goals is discussed in Ram [1989; 1991] and Ram and Hunter [1992]. For the purposes of this article, questions may be thought of as identified gaps in the system's memory or knowledge base, representing what the system needs to know for the purposes of the reasoning tasks that it is carrying out. Question representations have two parts:

- **Concept specification:** the object of the question, i.e., the desired information. 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 concept specification completely. The answer could specify more than the question required, of course.
- **Task specification:** what to do with the information once it comes in, which depends on why the question was generated. This may be represented either as a procedure 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.

When a question is posed, AQUA searches its memory for a knowledge structure that matches the concept specification of the question. If one is found, the question is immediately answered; if not, the question is indexed in memory and the task is suspended until an answer is found. An answer to a question is a node that matches the concept specification of the question and provides all the information required by the concept specification. The answer node may be created to represent new information provided by the story, or internally generated through inference during other processing. When a question is answered, the answer node is merged with the concept specification, and the task associated with it is run. The management of questions, including methods for indexing, retrieval and matching, is outside the scope of this article; further details may be found in Ram [1989].

3 What needs to be learned

For the purposes of this article, the important steps in the case-based explanation algorithm (section 2.3) are those of XP retrieval and XP application. XP retrieval involves the identification of XPs whose EXPLAINS nodes are unifiable with the description of the anomalous situation to be explained. XP application involves the unification of the internals of the XP with the representation of the story through partial matching. Both these steps are affected by gaps in the system's understanding of the domain, and both steps provide an opportunity for the system to learn.

3.1 XP retrieval

When faced with an anomalous situation, the reasoner tries to retrieve one or more explanation patterns that would explain the situation. Since it is computationally expensive to match the EXPLAINS node of every XP to every situation, a set of *indices* is used to retrieve XPs that are likely to be relevant to a given situation. Furthermore, following the typicality assumption A-3, these indices do not encode generalized applicability conditions representing all possible circumstances in which an XP might be applicable, but rather stereotypical descriptions of actual situations in which the XP has been applicable in the past.

For example, consider the applicability conditions for `xp-blackmail`. In general, blackmail is a possible explanation whenever "someone does something he doesn't want to do because not doing it results in something worse for him." But trying to show this in general is very hard since it requires a lot of inference. Thus, in addition to general applicability conditions, a reasoner must learn specific, sometimes superficial, features that suggest possibly relevant XPs even though they may not completely determine the applicability of the XP to the situation. For example, a classic blackmail situation is one where a rich businessman who is cheating on his wife is blackmailed for money using the threat of exposure. If one read about a rich businessman who suddenly began to withdraw large sums of money from his bank account, one would expect to think of the possibility of blackmail. However, one does not normally think of blackmail when one reads a story about suicide bombing, although theoretically it is a possible explanation.

In general, an index to an XP is a set of descriptions of the EXPLAINS node (the node that will be unified with what needs to be explained) that represent stereotypical situations in which the XP is likely to be applicable based on the past experiences of the reasoner. In AQUA's domain, the EXPLAINS node consists of a description of an action and an actor. For example, the religious fanatic XP described in figure 2 describes a `bombing` action with `death-state(actor)` and `destroyed-state(target)` outcomes, and a `volitional-agent` actor. This XP should be retrieved in situations that match this description.

Thus indices to volitional XPs consist of stereotypical situations or contexts in which the XP might be encountered, and stereotypical categories of actors to whom the XPs might be applicable. These are called *situation indices* and *character stereotype indices*, respectively. In other domains, an index to an explanatory case would consist of a typical configuration of its EXPLAINS node, which would represent the kind of

Read the story. Leaving aside the natural language aspects of the task, this is equivalent to processing a sequence of input facts representing the individual events in the story.

Explain each action in the story using the algorithm of table 1. Build hypothesis trees representing possible explanations for the motivations of the `actor`, `planner`, and any other volitional agents involved in the action.

Suspend the explanation task until all the HVQs of one of the hypotheses in a hypothesis tree are confirmed, or one of the HVQs is refuted.

Restart the suspended task when this happens. Confirm or refute the associated hypothesis, as appropriate.

Learn when a hypothesis is confirmed, using the algorithms described in tables 3, 4, and 5.

Table 2: The overall control structure in AQUA involves three main steps: **read**, **explain**, and **learn**. The interaction between these steps is managed through a question-based agenda system, in which tasks are **suspend**ed if there is insufficient information to run them, and **restart**ed when the questions seeking the missing information are answered. The **learn** step is discussed in the following sections.

situation in which it would be appropriate to use this case. A third type of index, the *anomaly category index*, represents the category of the XP required to explain a given type of anomaly. Thus the XP retrieval step may be thought of as a pre-filter for the XP application step.

AQUA can learn new indices to an explanatory case based on an experience in which the XP representing the explanatory case is applied in a novel context. AQUA can also fall back on general knowledge and learn a new explanatory case (as well as indices to this case) if there is no appropriate case in memory.

3.2 XP application

XP application involves the unification of the EXPLAINS node of the retrieved XPs with the anomaly node representing the current situation. The INTERNAL-XP-NODES and XP-ASSERTED-NODES must not be contradicted during this unification; they must either match nodes in the story representation, or be asserted into this representation. In addition, unconfirmed XP-ASSERTED-NODES must be justified in some manner external to the XP, either through a recursive explanation step using another XP, or by reading the story further.

An incompletely understood explanatory case is represented by an XP that has pending questions attached to it. If this occurs, these questions are instantiated during XP application and used to focus the understanding process. If the instantiated questions are answered by reading the story, answers to the questions are generalized and used to modify the original XP by answering the general questions attached to the XP.

The overall understanding and learning cycle of AQUA is shown in table 2. Let us now discuss the **learn** step in which AQUA learns XPs and indices to XPs.

Part II

Case learning

AQUA learns new explanatory cases through the incremental modification of XPs it already knows, a form of incremental case learning. There are two types of modifications:

- **Elaboration:** If an existing but inadequately understood explanatory case is retrieved, it is elaborated through a process of question generation and question answering.
- **Refinement:** If no explanatory case is available for this specific situation, an abstract explanation schema (abstract XP) is used and a new explanatory case (specific XP) is created by refining or specializing the abstract XP.

Although the system resorts to using general knowledge if no specific case is available, the same case-based explanation process is used to instantiate and apply both specific and abstract XPs. This is in contrast to typical explanation-based learning systems in which new explanations are constructed by exhaustive backchaining through primitive inference rules if an applicable schema is not available (e.g., [DeJong and Mooney, 1986]). AQUA's approach relies on the efficiency and content assumptions, A-1 and A-2, discussed earlier.

4 Explanation-based refinement

Let us start with the situation in which no specific XP is available, and a new case must be created by reasoning from abstract knowledge. Explanation-based generalization programs can learn new explanation schemas through the generalization of causal features from a novel story (e.g., [DeJong and Mooney, 1986]). This technique provides a method of creating new XPs by generalizing the details of specific novel explanations. However, XPs can also be created by specializing or *refining* abstract explanations in memory to create situation-specific XPs for different stereotypical situations. *Explanation-based refinement*[4], or EBR, is a type of explanation-based learning in which causal constraints from a novel story are used to refine an existing explanation schema, which may be very general, to a more specific case that applies to the kind of situation that the story is about.

5 Using an abstract explanation schema

For example, consider the suicide bombing story S-1 (section 1.2, page 4). Suppose this was the first suicide bombing story that the reasoner had ever read. Most people would have had no trouble understanding this story, even though it is a novel type of terrorism. We hypothesize that this is because people already have an abstract notion of *goal sacrifice*, which allows them to build an explanation for this story even though they may not fully understand why the bomber's life goal would have a lower priority than some other goal. AQUA's repertoire of explanation patterns includes an abstract XP called `xp-goal-sacrifice` that represents its understanding of a situation in which an agent pursues a high priority goal at the expense of a goal that is less important (see figure 5).

The abstract XP, `xp-goal-sacrifice`, is useful because it applies in a wide variety of situations. But it is also limited by its generality; it does not allow AQUA to predict the goal that the bomber was pursuing at the expense of her life goal. Trying to derive this in general would be impossible without detailed and domain-specific knowledge about terrorists in the Middle East. All that can be predicted is that the bomber probably has a goal that she values above her own life, which, while true, is not very useful at this level of generality. AQUA must still rely on the story to refine its expectations by telling it what this goal was, how the bomber came to have this unusual goal configuration, and so on. Furthermore, it would always have to go through this process for each suicide bombing story it encountered, unless it could learn a more specific goal sacrifice XP that applied to suicide bombing stories.

[4] This term is due to DeJong and Mooney [1986].

1. **Explains:** Why `volitional-agent` A did an action M, with `results` =

   (1) a `state` S1 such that A has the `goal` to achieve S1, i.e., there is a `goal` G1 with `goal-actor` = A and `goal-object` = S1.

   (2) a `state` S2 such that A has the `goal` to prevent S2, i.e., there is a `goal` G2 with `goal-actor` = A and `goal-object` = `not`(S2).

2. **Premises (XP-ASSERTED-NODES):**

   (1) A places a higher priority on G1 than on G2, i.e., there exists a `goal-ordering` between G1 and G2.

3. **Internals (LINKS and INTERNAL-XP-NODES):**

   (1) A believes that performing the action M will result in the state S1, thus achieving his goal G1 (an `expected-outcome`).

   (2) A believes that performing the action M will result in the state S2, thus violating his goal G2 (an `expected-outcome`).

   (3) A weighs his goals, goal-orderings, and likely outcomes (a `consideration`).

   (4) A decides to perform the action M (a `decision`, based on the considerations in (3)).

   (5) A performs the action M (represented as a `mop`).

   (6) M has an outcome for A, which is positive from the point of view of the goal G1, and negative from the point of view of the goal G2 (a `self-outcome`).

Figure 5: The abstract goal sacrifice explanation pattern, in which an agent *A* trades off one goal for another. The newly learned `xp-religious-fanatic` is a refined version of `xp-goal-sacrifice`.

Since by assumption AQUA does not have an explanatory case corresponding to `xp-religious-fanatic` at this point, it uses `xp-goal-sacrifice` to explain story S-1:[5]

```
Applying XP-GOAL-SACRIFICE to explain THE SUICIDE BOMBING
    Unifying EXPLAINS node
    Installing NODES
    Installing LINKS
        THE GIRL DECIDED TO DO THE SUICIDE BOMBING
        because THE GIRL WANTED TO DESTROY THE ISRAELI POST MORE THAN THE
                GIRL WANTED TO PRESERVE THE LIFE STATE OF THE GIRL.

    Installing questions for hypothesis verification
        DOES THE GIRL WANT TO ACHIEVE A GOAL MORE THAN THE GIRL WANTS TO
        PRESERVE THE LIFE STATE OF THE GIRL?

Finished installing XP-GOAL-SACRIFICE
```

Now suppose that the story mentioned that the bomber performed the mission in the cause of her religion. This fact answers the hypothesis verification question generated above. The explanation is now complete; the system now understands the motivations of the bomber. Following the typicality assumption A-3 discussed earlier, it would be useful at this point to build a more refined goal sacrifice XP which, while applicable to a lesser range of situations, would provide better and more detailed expectations about the situations to which it did apply. The refined XP is what we normally think of as the "religious fanatic" explanation. The abstract goal sacrifice XP would still be at hand to allow the system to deal with novel goal sacrifice situations which the refined XPs could not deal with.

[5] AQUA uses a simple template-based natural language generator to describe concepts in memory. The program traces presented here are the actual output of the AQUA program, except that the generator output has been cleaned up slightly for the sake of readability.

```
Answering question:
    DOES THE GIRL WANT TO ACHIEVE A GOAL MORE THAN THE GIRL WANTS TO
    PRESERVE THE LIFE STATE OF THE GIRL?
with:
    THE GIRL WANTED TO SPREAD HER RELIGION MORE THAN THE GIRL WANTED TO
    PRESERVE THE LIFE STATE OF THE GIRL.

Confirming XP-GOAL-SACRIFICE:
    THE GIRL DECIDED TO DO THE SUICIDE BOMBING
    because THE GIRL WANTED TO SPREAD HER RELIGION MORE THAN THE GIRL
            WANTED TO PRESERVE THE LIFE STATE OF THE GIRL.

This was a novel explanation for SUICIDE BOMBING!
Invoking EBR to refine XP-GOAL-SACRIFICE

Copying XP-GOAL-SACRIFICE to XP-GOAL-SACRIFICE-SUICIDE-BOMBING
Installing new refinement of XP-GOAL-SACRIFICE
Indexing XP-GOAL-SACRIFICE-SUICIDE-BOMBING in memory
```

Here, AQUA has read a new fact about the girl's goal priorities in the story, and has matched the fact to the concept specification of a pending question, thereby answering the question. The question was posed as a verification question (HVQ) for the goal sacrifice hypothesis. Since all the HVQs of this hypothesis are now answered, the hypothesis is confirmed. Since this is a novel explanation for suicide bombing, a new XP can now be built by refining `xp-goal-sacrifice`. The XP, representing the system's memory of this case, is then indexed in memory for future use.

Explanation-based refinement (EBR) is a model of the process by which specialized explanatory cases are built from abstract explanation schemas, following an experience in which an abstract schema is applied to a novel situation. Let us consider the steps in this process.

6 Building a novel explanation

Consider how AQUA learns the religious fanatic XP as a refined version of `xp-goal-sacrifice` after reading its first story about a religious fanatic performing a suicide bombing mission. There are two competing hypotheses that are built to explain the story. The simple explanation based on `xp-want-outcome` ("Actor performs action because actor wants the outcome of the action") is refuted since the bomber would not *want* her own death. This inference is made from the `preserve-life` goal that people are known to have. The correct explanation is the more complicated one based on `xp-goal-sacrifice`, in which the bomber trades off her `spread-religion` goal in favor of her `preserve-life` goal. Since this story is (by assumption) the first suicide bombing story that the system has read, the goal sacrifice explanation is a novel volitional explanation for the `actor` of a `terrorist-act`. This triggers the learning process.

7 Learning a new refinement

At this point, the goal sacrifice explanation has been instantiated, the details of this explanation have been filled in, and the explanation has been installed in memory. Since there is no matching goal sacrifice XP indexed under the concept `suicide-bombing`, AQUA realizes that the newly confirmed explanation is a novel explanation. The next step is to create a new XP that is a refined version of `xp-goal-sacrifice`. The new XP is indexed under the `xp-goal-sacrifice` category, which represents the abstract category of goal sacrifice explanations. The situation index is `suicide-bombing`, and a newly created stereotype forms the character stereotype index. (Index learning is discussed in part IV.) The XP is named `xp-goal-sacrifice-suicide-bombing`, and represents what we would think of as the religious fanatic explanation.

Creating the new refinement involves the following steps (see table 3):

Input: XP_a, an abstract XP; XP_i, the instantiation of XP_a for the situation at hand.

Output: XP_r, a refined XP of type XP_a.

Algorithm:

- Copy XP_a to XP_r (XP copy).
- Substitute EXPLAINS node of XP_r with a more refined node based on the instantiation of this node in XP_i.
- $\forall$ XP-ASSERTED-NODES and INTERNAL-XP-NODES n in XP_r, substitute n with a more refined node n', based on the instantiation of n in XP_i (node substitution).
- $\forall$ XP-ASSERTED-NODES n with an external explanation with XP-ASSERTED-NODES e, internalize n by making it an INTERNAL-XP-NODE of XP_r. Generalize the nodes e and add them to the XP-ASSERTED-NODES of XP_r (node internalization).
- $\forall$ LINKS l between nodes $n1$ and $n2$, elaborate l by substituting it with an XP in which $n2$ is the EXPLAINS node and $n1$ is an XP-ASSERTED-NODE (node and link elaboration).
- Learn indices for XP_r (index learning, see part IV).

Table 3: Explanation-based refinement. "Substitution" is the straightforward operation of replacing part of the explanation graph with a new subgraph. "Generalize" and "refine" are discussed in the accompanying text.

XP copy: A copy XP is made of the abstract XP.

Node substitution: Nodes in the new XP are substituted with newly created nodes, based on the instantiation of these nodes in the story.

Node internalization: Nodes in the new XP may be internalized, and new nodes added in order to "grow" the XP at the fringes.

Node and link elaboration: Nodes and links in the new XP are elaborated and replaced with sets of nodes and links, respectively.

XP indexing: The new XP is indexed in memory.

The new XP is named by concatenating the name of the abstract XP and the situation index. This of course is arbitrary, since the system does not care about the names of its XPs. The substitution, internalization, and elaboration processes form the core of EBR, and are discussed below.

7.1 Node substitution

The node substitution step involves the creation of a new node for each node in the original XP, and the substitution of each new node into its appropriate place in the new XP:

Substitute EXPLAINS node with a more refined node representing the anomaly in the current story.

Substitute each XP-ASSERTED-NODE and INTERNAL-XP-NODE with a more refined node representing the particular situation being explained.

The issue here is, what should these refined nodes look like? If the nodes are not generalized away from the specific details of the story, the new XP will only be applicable to this exact story. On the other hand, we still want to retain the specificity of the suicide bombing scenario; an overly generalized XP would take us right back to `xp-goal-sacrifice` where we started. In order to resolve this tradeoff, consider the functional

role of XPs in the story understanding process. XPs are used to generate causal explanations in order to resolve anomalies that arise during the understanding process. Furthermore, XPs are designed to provide an efficient way to reason about stereotypical situations in which similar anomalies are likely to occur. Thus the new XP should be as general as possible while still providing the specific information required to resolve the anomaly encountered in the story. Furthermore, the nodes in the new XP should not violate the causal constraints specified by the LINKS of the XP.

Based on these arguments, the rule used for refinement in table 3 is:

> **R-1: Refinement rule:** Each node is replaced by the most abstract node below it in the multiple inheritance hierarchy such that the node provides the specific inferences required by the particular causal structure observed in the story, including the structure underlying the anomaly.

The causal constraint used in this rule is similar to the identification and generalization of relevant features in explanation-based generalization through goal regression (e.g., [Mitchell *et al.*, 1986]). In the above example, the anomaly to be resolved is one of `goal-violation`: Why would an actor perform an action that resulted in her own death? The LINKS in `xp-goal-sacrifice` specify that the `decision` to perform the action was a `mental-result` of a `mental-process` in which the `actor` `consider`ed her `goals`, `goal-orderings`, and the `expected-outcomes` of the `suicide-bombing`, and decided to sacrifice her `preserve-life` goal for her `spread-religion` goal which she valued above her `preserve-life` goal. Based on the causal structure of this anomaly and the corresponding explanation, the following node substitutions are made to the contents of `xp-goal-sacrifice` shown in figure 5 (page 16):

- `xp-goal-sacrifice` requires the `agent` A to be any `volitional-agent`. In the particular story, the `agent` is a specific instance whose abstractions (`isa`) belong to the set {`girl`, `terrorist`}, each of which `isa` {`volitional-agent`}. Whereas `girl` does not provide any specific expectations for the two goals (`preserve-life` and `spread-religion`) implicated in the anomaly, `terrorist` does specify an expectation for `spread-religion` that is not specified by the general `volitional-agent` concept. Thus the `agent` node A is replaced by a `terrorist` node.
- The general `mop` M is replaced by `suicide-bombing` though a similar chain of inferences. `suicide-bombing` is the most general abstraction of the particular instance of suicide bombing in the story that still provides the particular expectations required by the anomaly: that the agent be a `terrorist` (determined above), and that the outcome involve a `state` that achieves the `spread-religion` goal (which the destruction of an enemy target is known to do), and another state which violates the `preserve-life` goal of the agent (inferred from the fact that `suicide-bombing` is a type of `suicide`). Further abstractions of `suicide-bombing` (`suicide`, `bombing`, `destroy`, `mop`) do not provide all these inferences.
- The outcomes S1 and S2 are replaced by `destroyed-state(target)` and `death-state(actor)`, respectively.
- The goals G1 and G2 are replaced by `spread-religion(actor)` and `prevent(actor,death-state(actor))` (i.e., `preserve-life(actor)`), respectively.

In general, some nodes may not be substituted by more refined nodes if the story does not provide any specialized information about that node (e.g., the `decision` node `chooses-to-enter` remains as it is since there is no further information about the kind of decision it was). A node may be also substituted by a collection of nodes; this is called node elaboration and is discussed later.

7.2 Node internalization

The second type of process involved in EBR is node internalization, in which the XP is elaborated by adding nodes to explain one or more of its previously unexplained XP-ASSERTED-NODES, thus pushing back the explanation at its fringes. Notice that this type of refinement involves further specification of the XP through elaboration rather than through specialization down an inheritance hierarchy. This involves the following process:

> **For each XP-ASSERTED-NODE** (i.e., for each node with no explanation within the XP itself) that has been explained in the story:
>
> **Internalize the node** by deleting it from the set of XP-ASSERTED-NODES of the XP and adding it to the set of INTERNAL-XP-NODES of the XP
>
> **Create a new node** by generalizing the node that causally explains the node being internalized
>
> **Add the new node** to the set of XP-ASSERTED-NODES of the XP

Thus each premise of the XP for which a causal explanation has been found is turned into an internal node of the XP, and the premises of the causal explanation (i.e., the nodes that are causally linked to the node being internalized) are made the new premises of the XP. The generalization process for the new premise node is the same as that for node substitution, and uses similar constraints to determine the right level of generalization from the story. The main difference is that, since this node is outside the limits of the original XP, there is no abstract node specified by the original XP to specialize from.

Since case-based reasoning relies on specific experiences of the system (here, specific XPs known to the system), these experiences should not be overly generalized because that would lose much of the power of the case-based reasoning method. Based on this argument, the following rule is used for generalization:

> **R-2: Generalization rule:** Each node is replaced by the most specific node above it in the multiple inheritance hierarchy such that the node provides the specific inferences required by the particular causal structure observed in the story, including the structure underlying the anomaly.

In both R-1 and R-2, the heuristic used is to modify the existing XP as little as possible. Again, this is justified by the typicality assumption A-3 of the case-based reasoning approach. Since existing XPs represent the system's past experiences, it makes sense to modify them incrementally based on new experiences rather than making radical changes to the system's representation of its past experiences.

In the present example, the newly built `xp-goal-sacrifice-suicide-bombing` can be further elaborated as follows:

- Internalize the `goal-ordering` node (`goal-ordering(actor,(spread-religion(actor)), preserve-life(actor))`), making it an INTERNAL-XP-NODE instead of a PRE-XP-NODE.
- Add one or more new PRE-XP-NODES that explain how this `goal-ordering` arises. In the religious fanatic example, the new PRE-XP-NODE would be `high-religious-zeal(actor)` since this node has a `mentally-initiates` link to the `goal-ordering` node.
- Add the causal relationship (`mentally-initiates`, specified by the explanation built for the story) between `high-religious-zeal(actor)` and the internalized `goal-ordering` to the LINKS of the XP.

At this point, the refined XP is identical to the religious fanatic XP that was described earlier. Thus AQUA has learned its first explanatory case about religious fanaticism through the application of the abstract explanation schema `xp-goal-sacrifice` to a novel situation.

7.3 Node and link elaboration

The above processes form the core of the EBR algorithm. Other methods of refinement can be also used to augment the algorithm. One such method is node and link elaboration. A node in the XP may be replaced by a collection of nodes, representing a more detailed understanding of that part of the causal structure of the XP. An XP may also be elaborated by replacing one of the LINKS with a set of LINKS between the same nodes, with new intermediate nodes being added in between these nodes. For example, a LINK specifying a direct causal relationship between two nodes $n1$ and $n2$ may be replaced by a more refined LINK that invokes an XP in which $n2$ is the EXPLAINS node and $n1$ is one of the XP-ASSERTED-NODES. This has the effect of replacing a primitive LINK with a more detailed explanation, yielding a better understanding of the internal causality of the XP. Though the representation supports the elaboration of XPs in this manner, the implementation of this extension to the basic algorithm is not yet complete.

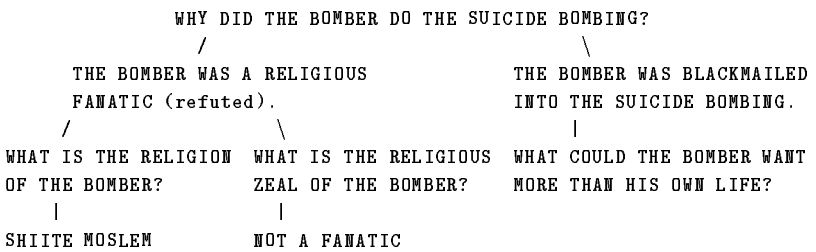
The final step, learning indices for the new XP, is done using the same algorithm that is used to learn new indices for existing XPs, and is discussed in part IV.

Part III

Incremental case modification and elaboration

The second type of case learning occurs when AQUA learns new XPs through the incremental modification and elaboration of XPs it already knows. When a situation is encountered for which an explanatory case is known but is inadequate, either because it was incompletely understood in the past or because it does not fit the new situation well, application of the XP representing the case causes the system to generate questions corresponding to gaps in its domain knowledge. Learning occurs through the generation and answering of these questions, corresponding to XP modification through gap filling. The modification is constrained by the causality underlying the situation being explained as in other types of explanation-based learning.

Revision of previously existing case knowledge is a fundamental issue for incremental learning in complex domains, in which intermediate results are likely to be incomplete or even incorrect. For example, suppose AQUA has just read the blackmail story S-3 (section 1.2, page 4). After reading this story, AQUA builds the following hypothesis tree in memory, representing an anomaly (*Why would the bomber perform an action that resulted in his own death?*), alternative hypotheses constructed by applying known XPs to the anomalous situation (*religious fanatic* and *blackmail*), questions that would verify these hypotheses, and answers to these questions, if any:



The final explanation built for this story involves a novel application of a stereotypical XP, `xp-blackmail`, that is already known to the system (see figure 6).[6] In this example, even though AQUA already knows about

[6] This story explicitly mentions blackmail as the explanation for the suicide bombing action. In general, how the correct hypothesis is determined is not relevant to this article, which focusses on what is learned after the hypothesis tree is built.

(1) The blackmailee has a goal G1.

(2) The blackmailer has a goal G2, and the blackmailee does not have the goal G2 (since otherwise he or she would satisfy the goal without needing to be threatened).

(3) The blackmailee has a goal G3, which he or she values above goal G1.

(4) The blackmailer threatens to violate G3 unless the blackmailee performs an action A that satisfies G2, even though the action would have a negative effect of violating G1.

Figure 6: Internals of the blackmail explanation pattern.

blackmail, it learns a new variant of this XP (`xp-blackmail-suicide-bombing`), based on the particular manner in which `xp-blackmail` was adapted to fit the story. AQUA also learns indices to the new XP. Both kinds of learning are important in a case-based reasoning system [Ram, 1990c].

The algorithm for incremental case modification and elaboration is shown in table 4. The process involves question generation and question answering, and is discussed below.

8 Associating new questions with XPs

Suppose AQUA reads the blackmail story S-3 with only the religious fanatic XP for suicide bombing in memory. When reading this story, AQUA is handed an explanation for the suicide bombing: the story explicitly mentions that the bomber was blackmailed. In a sense, then, the story has been understood since an explanation for the bombing has been found. However, one could not really say that AQUA had understood the story if it did not ask the question, *What could the boy want more than his own life?* Unless this question is raised while reading the story, one would have to say that AQUA had missed the point of the story.

Operationally, although the blackmail XP is available, application of this XP to the story leaves open questions that must be answered before the explanation is complete. Such questions correspond to gaps in the explanation structures that are built during the understanding process (figure 7). XP-related questions arise from two sources: (1) A question left over from a previous story is instantiated in the current story when the XP to which the question is attached is applied, and (2) a new question is raised when an XP is independently confirmed but some of its HVQs have not yet been answered (see table 4). These questions are associated with the XP, and may be answered later in the story or when the XP is applied to a future story. When they are answered, the understander can elaborate and modify the XP, thus achieving a better understanding of the causality represented by the XP.

In the present example, the question *What could the boy want more than his own life?* is generated as an HVQ for the blackmail hypothesis, since the desired goal state is one of the XP-ASSERTED-NODES of `xp-blackmail`. Since this question is not answered by the story, it is retained as a question on `xp-blackmail-suicide-bombing`.

9 Incremental refinement of XPs by answering questions

In addition to raising new questions, of course, an understander must answer the questions that it already has in order to improve its knowledge of the domain. This involves the instantiating of pending questions during the XP application process, and the answering of the instantiated questions during story understanding

Question generation

Input: XP, an explanation pattern; XP_i, the instantiation of XP for the situation at hand; independent confirmation for XP_i.

Output: XP', an elaborated version of XP with one or more gaps g identified.

Algorithm:

- Confirm XP_i in the hypothesis tree and refute its competitors.
- Copy XP to XP'.
- $\forall$ HVQs of XP_i that are not answered, mark the corresponding XP-ASSERTED-NODE g in XP' as a gap. Install a question whose concept specification is g (question generation through identification of new gaps).

Question answering

Input: XP, an explanation pattern with a gap g; XP_i, the instantiation of XP (with corresponding gap g_i) for the situation at hand; s_i, a later fact that matches the concept specification g_i.

Output: XP', an elaborated version of XP with the gap g filled.

Algorithm:

- Instantiate g for XP_i and build a new question whose concept specification is the new instantiation g_i (question generation through instantiation of pending questions).
- Index instantiated question in memory and suspend. The concept specification of this question is the desired information g_i, and the task specification is the restarting of this question-answering process.
- When the question is answered (g_i is merged with a story node s_i), restart suspended process (question answering).
- Generalize s_i to yield an answer s to the original question whose concept specification was g.
- Merge s with g in XP, and remove the tag identifying this node as a gap.

Table 4: Incremental case modification and elaboration, involving question generation (gap identification) and question answering (gap filling).

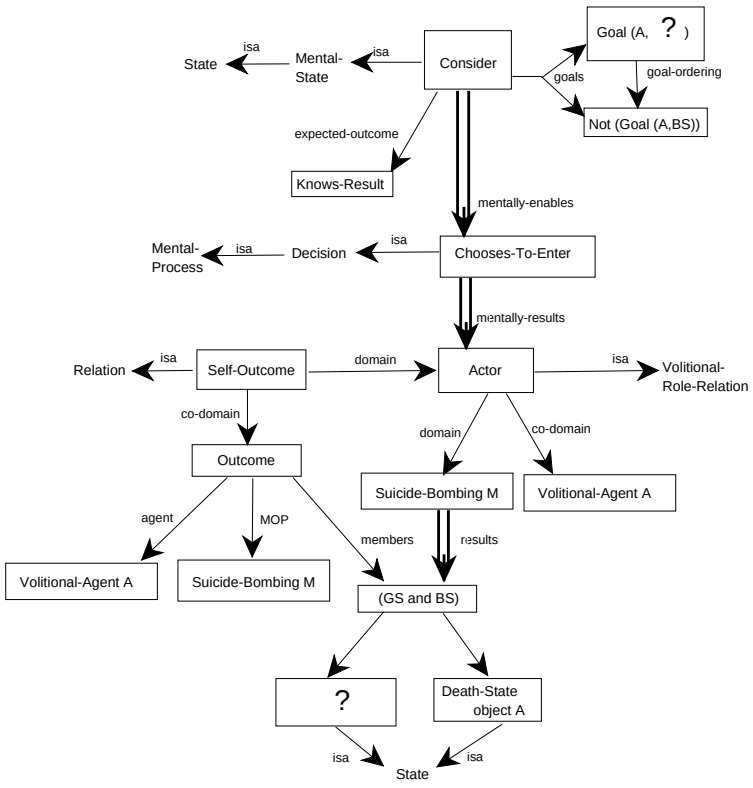


Figure 7: Associating new questions with XPs. The XP represents a situation in which an `agent` A volitionally performs (`chooses-to-enter`) an action whose outcome is known (`knows-result`) to be the `death-state` of A, as well as an unknown state that A wants more than he wants to avoid his `death-state` (the `goal-ordering`). The unknown `goal` represents the new question, *What could the actor want more than his own life?* This is depicted as an empty box, representing a gap in the program's knowledge. The XP is elaborated by filling in this gap when this question is answered.

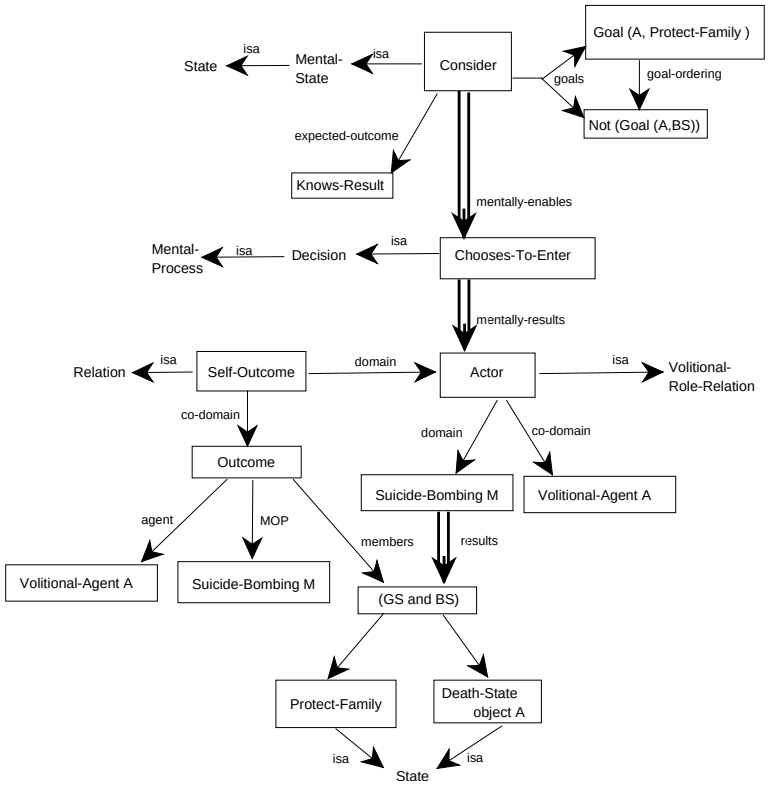


Figure 8: Elaborating an XP through incremental learning. The changed portion is depicted as a newly filled-in box, representing the answering of the question that was indexed at that point (compare with figure 7).

(table 4). A question is answered by merging it with a node that matches the concept specification of the question and provides all the information required by the concept specification. The answer node is built to represent new information provided by the story, or internally generated through inference. The answer node is then generalized to yield an answer to the original question attached to the XP.

For example, consider the following story:

> **S-4:** JERUSALEM — A young girl drove an explosive-laden car into a group of Israeli guards in Lebanon. The suicide attack killed three guards and wounded two others. ...
>
> The driver was identified as a 16-year-old Lebanese girl. ... Before the attack, she said that a terrorist organization had threatened to harm her family unless she carried out the bombing mission for them. She said that she was prepared to die in order to protect her family.

When this story is read, AQUA retrieves the new `xp-blackmail-suicide-bombing` (figure 7) and applies it to the story. The question that is pending along with this explanation is also instantiated. When the question is answered, it is replaced by a new node representing the `protect-family` goal, and becomes part of `xp-blackmail-suicide-bombing` (figure 8). Since no explanations are known for the newly added node, this in turn becomes a new question about the elaborated XP (not shown in the figure). The question is seeking a reason for the unusual `goal-ordering` of the `actor`, in which `protect-family` is given a higher priority than `preserve-life`.

When the elaborated XP is applied to a new suicide bombing story, the new node will now be one of the premises of the hypothesis, causing AQUA to ask whether the actor was trying to protect his family.

This reflects a deeper understanding of this particular scenario. The new question will also be instantiated, causing AQUA to look for an explanation for the unusual `goal-ordering`. Should new questions be raised and then answered during future stories, AQUA will again be able to elaborate this XP in a similar manner. Thus AQUA evolves a better understanding of the "blackmailed into suicide bombing" scenario through a process of question asking and answering.

Part IV

Index learning

10 Learning indices for explanatory cases

Regardless of whether a new explanatory case is learned from scratch or by applying an existing XP to a new situation, the case needs to be indexed in memory appropriately so that it can be used in future situations in which it is likely to be useful. As described earlier, XPs are associated with stereotypical situations and people in memory. An understander needs to learn the stereotypical categories that serve as useful indices for volitional explanations. This is a type of inductive category formation [Dietterich and Michalski, 1981]; however, the generalization process is constrained so that the features selected for generalization are those that are causally relevant to the explanations being indexed [Barletta and Mark, 1988; Flann and Dietterich, 1989].

XPs are indexed in memory using stereotypical descriptions of the EXPLAINS node. These descriptions represent the types of situations in which an XP has been useful in the system's experience, and are created through explanation-based generalization of the story node that is unified with the EXPLAINS node of a particular XP. In AQUA's domain, the EXPLAINS node represents a `volitional-role-relation`, since an XP provides a volitional explanation for why an agent was the `actor` or `planner` of a given action. Thus AQUA indexes volitional XPs in memory using typical classes of actions or contexts in which the XPs might be encountered (*situation indices*), as well as character stereotypes representing typical categories of people to whom the XPs might be applicable (*character stereotype indices*). As discussed earlier, a third type of index, the *anomaly category index*, represents the category of the XP required to explain a given type of anomaly.

In the above example (story S-3), AQUA learns a new context for blackmail (suicide bombing), as well as a new character stereotype representing the type of person who one might expect to see involved in a "blackmailed into suicide bombing" explanation. Let us discuss how AQUA learns these indices (see table 5).

11 Learning situation indices

AQUA learns new contexts (e.g., "suicide bombing") for stereotypical XPs (e.g., "blackmail"), which are then used as situation indices for these XPs in the future. The main issue here is how far the context should be generalized before it is used as an index. In the above example, should the new situation index for blackmail be `suicide-bombing`, `suicide`, `bombing`, `destroy`, or indeed any MOP (action) with a negative side effect for the actor? As discussed earlier, XP theory relies not on generalized reasoning about all possible conditions under which an XP might be applicable, but rather on specific reasoning about stereotypical situations that have actually been encountered by the reasoner. After reading story S-3, for example, one would expect to think of blackmail when one reads another story about a `suicide-bombing` attack. However, one would

Input: XP, an explanation pattern; XP_i, the instantiation of XP for the situation at hand.

Output: Situation (I_s), character stereotype (I_c) and anomaly category (I_a) indices for XP.

Algorithm:

- Let n_s be the node representing the situation, and n_c be the node representing the actor, in the story node n unified with the EXPLAINS node of XP. By definition, n_s is the `domain` of n, and n_c is the `co-domain` of n.
- Generalize n_s to create situation index I_s for XP.
- Generalize n_c to create character stereotype index I_c for XP.
- Create pointers to XP using I_s and I_c indices.
- If XP is a newly refined version of an abstract XP XP_a, let anomaly category index I_a be the abstract category of XPs represented by XP_a; else if XP is a modified version of another specific explanation pattern, retain the old index I_a.

Table 5: Index learning. This algorithm is used to learn stereotypical descriptions of components of the EXPLAINS node (here, the situation and actor components), and the abstract category of the explanation, to be used as indices to the XP. The pointer creation operations are straightforward, and are illustrated in figures 9 and 10. "Generalize" is discussed in the accompanying text.

probably not think of blackmail on reading any story about `suicide`, say, a teenager killing himself after failing his high school examinations, even though theoretically it is a possible explanation. Furthermore, it would not be useful to index the new XP under `bombing` in general (as opposed to `suicide-bombing` in particular), since the particular goal violation of the `preserve-life` goal is central to this explanation.

Following this argument, the rule used for generalization in table 5 is as follows:

> **R-3: Index generalization rule:** Each node is replaced by the most specific node above it in the multiple inheritance hierarchy that belongs to the category of stereotypical nodes for that type of index. This category is identified by the content theory of indices for the domain. (This rule will be elaborated in the next section, but is sufficient as stated at present.)

In the present example, AQUA uses `suicide-bombing`, as opposed to the abstractions `suicide` or `bombing`, as the situation index for the new variant of `xp-blackmail` (figure 9). After reading several stories about blackmail, AQUA would know about different stereotypical situations in which to use the blackmail explanation, rather than a generalized logical description of every situation in which blackmail is a possible explanation. In other words, AQUA would have indexed a copy of `xp-blackmail` under all the MOPs for which it has seen `xp-blackmail` used as an explanation. Whenever these MOPs are encountered, AQUA would retrieve the new blackmail XP (if the other indices are also present).[7] The reason that a copy of the original XP is used is that the XP, once copied, will need to be modified for that particular situation, as discussed below.

It should be noted that the generalization process for index construction is based on a content theory of the kinds of indices that are useful for the reasoning task in the context of which the learning algorithms are being invoked. Here, the theory of case-based explanation that specifies what kinds of indices are useful. This set of specifications may be viewed as the *operationality criterion* [Keller, 1988] for the explanation-based generalization process, and is in contrast to the use of purely syntactic criteria such as "generalize as far as

[7] AQUA can still understand other blackmail situations that it has not learned about as yet, as it did while reading the story in this example. Thus not having a situation index for an XP does not necessarily mean that the XP cannot be applied to the situation, but rather that this XP is not one that would ordinarily come to mind in that situation. Additional cues, or explicit external mention of the XP, would be needed to retrieve the XP in such situations.

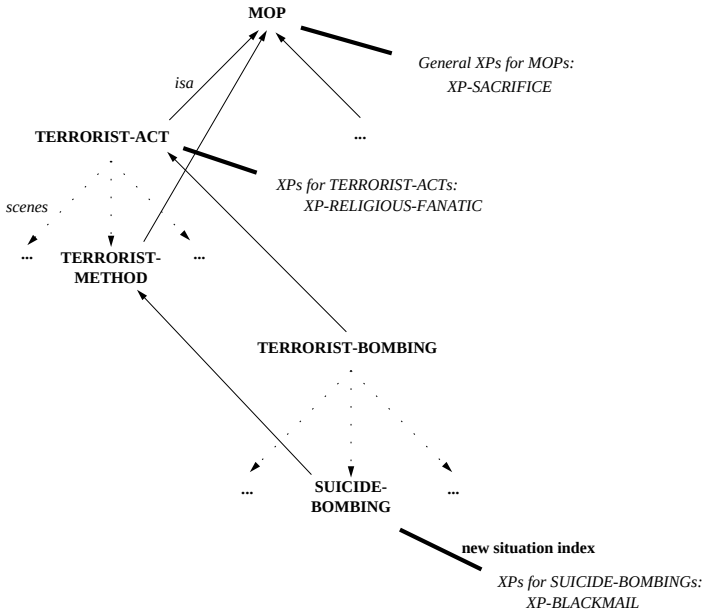


Figure 9: Learning situation indices for XPs. Upward lines represent `isa` links, and downward dotted lines represent `scenes` of MOPs. Heavy lines represent situation indices, which point from MOPs to XPs. Here, AQUA has just built a situation index from `suicide-bombing` to a copy of `xp-blackmail`.

possible" or "always express the generalization as a disjunct of three conjuncts" that provides the constraints, or *bias* [Mitchell, 1980; Michalski, 1983], on the generalization process in many empirical learning methods.

12 Learning character stereotype indices

The main constraint on a theory of stereotype learning is that the kinds of stereotypes learned must be useful in retrieving explanations. In other words, they must provide the kinds of discriminations that are needed for indexing XPs in memory. Since volitional explanations are concerned with goals, goal-orderings, plans, and beliefs of characters, the learning algorithm must produce typical collections of goals, goal-orderings, plans and beliefs, along with predictive features for these elements. Such a collection is called a *character stereotype*.

Character stereotypes serve as motivational categories of characters and are an important index for XPs in memory. Continuing with the blackmail example, AQUA learns a new stereotype (`stereotype.79`) representing a typical Lebanese teenager who might be blackmailed into suicide bombing, which is used to index the blackmail XP. The stereotype is built from the novel blackmail explanation by generalizing the features of the `volitional-agent` involved in that explanation:

```
Answering question: WHY DID THE BOY DO THE SUICIDE BOMBING?
              with: THE BOY WAS BLACKMAILED INTO DOING THE SUICIDE BOMBING.

Novel explanation for A SUICIDE BOMBING!
```

```
Building new stereotype STEREOTYPE.79:
    Typical goals:    P-LIFE (in)
                      A-DESTROY (OBJECT) (out)
                      AVOIDANCE-GOAL (STATE) (question)
    Typical goal-orderings:
                      AVOIDANCE-GOAL (STATE) over P-LIFE (question)
    Typical beliefs:  RELIGIOUS-ZEAL = NOT A FANATIC (in)
    Typical features: AGE = TEENAGE AGE (hypothesized)
                      RELIGION = SHIITE MOSLEM (hypothesized)
                      GENDER = MALE (hypothesized)
                      NATIONALITY = LEBANESE (hypothesized)

Indexing XP-BLACKMAIL-SUICIDE-BOMBING in memory
    Category index   = XP-GOAL-SACRIFICE
    Stereotype index = STEREOTYPE.79
    Situation index  = SUICIDE-BOMBING
```

The label `in` (`out`) marks features that are known to be true (false) of this stereotype [Doyle, 1979]. These features are definitional of the stereotype. The label `question` marks features that are `in` but incomplete. In this case, `(AVOIDANCE-GOAL (STATE))` refers to an unknown goal that needs to be filled in when the information comes in. This is represented as a `goal` with an unknown `goal-object`.

Finally, the label `hypothesized` marks features that were true in this story but were not causally relevant to the explanation. These features are retained for the purposes of recognition and learning. Since AQUA does not assume that its explanations are complete, there is the possibility of learning more about this explanation in the future that would help to determine whether these features have explanatory significance. These features are similar to the "possibly relevant" features described by Barletta and Mark [1988]. In their system, such features are used as "secondary indices" and are refined through induction. Our emphasis has been on the learning of "primary indices"; it would be relatively straightforward to add Barletta and Mark's "secondary indices" to AQUA. Ideally, however, hypothesized features should be refined, not just through induction, but through explanation-based processes as well. This is an important issue for further research.

The stereotype is used to index the new explanation in memory (figure 10). After reading this story, AQUA uses the new stereotype to retrieve the blackmail explanation when it reads other stories about Lebanese teenagers going on suicide bombing missions. The stereotype is built through generalization under causal constraints from the hypotheses that were considered, including the ones that were ultimately refuted. The causal constraints are derived both from the successful explanation (blackmail) as well as from unsuccessful hypotheses, if any (here, religious fanaticism). The rule for index generalization described in the previous section applies here as well, but needs to be elaborated. The rule as stated identifies the category of stereotypical nodes for the given type of index. While this is sufficient to determine the situation index (e.g., `suicide-bombing`), this rule would simply identify the abstract node `stereotype` as the character stereotype index. To specify the details of the stereotype, the generalization rule must also provide a means for identifying the features of the generalized node (i.e., the goals, goal-orderings, beliefs, and other features of the stereotype). This identification relies on causal constraints similar to those used for the identification of relevant features in explanation-based learning (e.g., [Mitchell *et al.*, 1986; DeJong and Mooney, 1986]), with the difference that the intent here is to produce generalizations that are as specific as possible within the causal constraints of the observed situation.

> **R-3: Index generalization rule (elaborated):** Each node n is replaced by the most specific node above it in the multiple inheritance hierarchy that belongs to the category of stereotypical nodes for that type of index. This category is identified by the content theory of indices for the domain.
>
> The features of the node n are identified by generalizing the specific nodes m in the instantiated representation of n as follows: Each node m that is a feature in the instantiation of n is

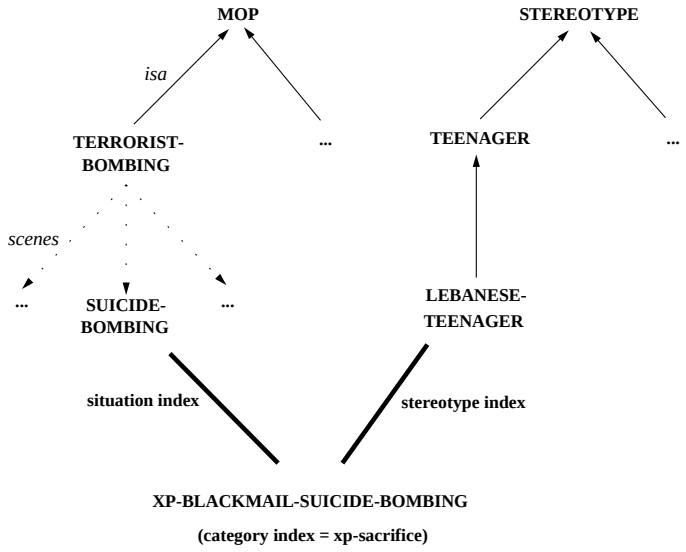


Figure 10: Learning character stereotype indices for XPs. Upward lines represent `isa` links, and downward dotted lines represent `scenes` of MOPs. Heavy lines represent indices to XPs. Here, AQUA has just built a character stereotype index from `stereotype.79`, representing a `lebanese-teenager`, to `xp-blackmail-suicide-bombing`.

replaced by the most specific node above it in the multiple inheritance hierarchy that provides the specific inferences required by the particular causal structure observed in the story (in particular, the successful and failed explanations that n plays a role in).

Notice that the generalization rule for index learning is very similar to the generalization rule R-2 (section 7.2). Here too the most specific node is sought, rather than the most abstract node as in the refinement rule R-1 (section 7.1). The reason for this is that index learning seeks to learn situation-specific indices to explanatory cases, and therefore generalizes as little as possible from the observed situation. This ensures that a newly learned case will be remembered in other situations that are as similar as possible to the case, within the constraints provided by a causal analysis of the case. The elaborated index generalization rule is illustrated below.

Learning from successful explanations: Clearly, much of `stereotype.79` comes from the motivational aspects of the blackmail explanation. AQUA retains those goals, goal orderings, and beliefs of the character in the story that are causally implicated in the blackmail explanation. Since blackmail relies on a goal ordering between two goals, one of which is sacrificed for the other, the stereotype must specify that the character has a goal that he or she values above `preserve-life`. The stereotype also specifies that the character would normally not have the goal of performing terrorist missions, since this is part of the blackmail explanation. Based on the rule R-3, AQUA infers the following goals and goal-orderings for the actor (corresponding to (1), (2) and (3) of `xp-blackmail`; see figure 6, page 22):

```
Typical goals:     P-LIFE (in)
                   A-DESTROY (OBJECT) (out)
                   AVOIDANCE-GOAL (STATE) (question)
Typical goal-orderings:
                   AVOIDANCE-GOAL (STATE) over P-LIFE (question)
```

Learning from failed explanations: Many explanation-based learning programs learn only from positive examples (e.g., [Mooney and DeJong, 1985; Segre, 1987]). However, it is also possible to apply this technique to learn from negative examples (e.g., [Mostow and Bhatnagar, 1987; Gupta, 1987]). AQUA uses refuted hypotheses to infer features that should *not* be present in the newly built stereotype. These are features that, if present, would have led to the hypothesis being confirmed.

For example, in the blackmail story, AQUA knows that the person being blackmailed is not a religious fanatic, since the religious fanatic explanation, which depended on this fact, has been refuted. The kind of person likely to be blackmailed into suicide bombing is, therefore, not a religious fanatic.[8] Using rule R-3, the new belief is generalized from the beliefs of the boy in the story (corresponding to (2), the failed premise, of `xp-religious-fanatic`, figure 2). This belief is recorded in the newly built stereotype:

```
Typical beliefs:  RELIGIOUS-ZEAL = NOT A FANATIC (in)
```

The reason that learning from the failed explanation works in this example is that the blackmail explanation specifies that the person being blackmailed would normally not have the goal to perform that action. This rules out other explanations that would result in this goal. Our theory does not deal with the issue of multiple successful explanations; more research needs to be done in this area.

13 Learning anomaly category indices

The *anomaly category index*, or simply category index, represents the category of the XP required to explain a given type of anomaly. For example, if the anomaly was one where an actor performed an action that violated one of the actor's own goals ("goal violation"), the reasoner might look for a "goal sacrifice" XP (such as a religious fanatic sacrificing her life for the cause of her religion), or an "actor didn't know outcome" XP (such as a gullible teenager not realizing what the outcome of her action was going to be). However, the category of goal sacrifice XPs would be inappropriate for an anomaly in which the actor failed to perform an action which only had a good outcome for the actor; in this case, a "missed opportunity" XP might be chosen. As described earlier, AQUA determines anomaly category indices by looking up the anomaly (e.g., `goal-violation`) in a table associating anomalies with abstract XPs that form category indices for specific XPs (e.g., `xp-goal-sacrifice` and `xp-not-know-outcome`).

When a known XP is modified or elaborated through question answering, the category index of the XP does not change. In this case, the old category index is retained for the modified XP. When an abstract XP is refined, the new XP (such as `xp-goal-sacrifice-suicide-bombing`, the newly learned religious fanatic explanation) is placed in the category of XPs defined by the abstract XP (such as `xp-goal-sacrifice`). Thus religious fanaticism is learned as a type of goal sacrifice explanation, and is recalled when "goal sacrifice" is an appropriate type of explanation for the particular anomaly at hand.

Part V

Discussion

The underlying theme of our research is a focus on the learning goals of the reasoner. In particular, we are developing a theory of *knowledge goals*, which represent the goals of a reasoner to learn by acquiring new

[8] As before, this is a stereotypical inference and not a logically correct one. A religious fanatic could indeed be blackmailed into suicide bombing; however, on reading a story about a religious fanatic going on a suicide bombing mission, blackmail would not normally come to mind. This means that `xp-blackmail-suicide-bombing` should not be indexed under `religious-fanatic`, at least on the basis of this example.

knowledge or reorganizing existing knowledge by learning new indexing structures [Ram, 1991].[9] Knowledge goals arise from gaps in the reasoner's knowledge that are identified when the reasoner encounters difficulties during processing. Our approach is in contrast to other approaches that rely on properties of the domain to determine what needs to be learned rather than on the goals of the reasoner. For example, one might propose a rule, similar to that discussed by DeJong [1983], that the understander generalize a new schema whenever it reads a story in which a preservation goal (P-GOAL) is violated in a novel manner. But this should be so only if noticing violations of this P-GOAL is actually useful to the program. Any such rule must make a statement about the goals of the program, not just about the content of the domain. A similar argument can be made for the use of knowledge goals to focus the inferencing process for understanding, explanation, or diagnosis [Ram, 1990d; Ram and Leake, 1991; Ram and Hunter, 1992].

Identification of knowledge goals is essential to the solution of the problems of when learning should be triggered, what knowledge should be learned, and how the appropriate level of generalization should be determined. We are investigating the kinds of knowledge goals that arise out of gaps in a reasoner's knowledge, and the learning methods that could be used to learn in situations involving these knowledge goals. For example, a reasoner often makes simplifying assumptions when dealing with complex situations. When these assumptions fail, the reasoner may be able to learn applicability rules for its simplified assumptions. For example, one does not explicitly decide to keep one's shoes on when entering a restaurant; this is part of our set of implicit assumptions about social situations. However, if one is asked to take off one's shoes in a traditional Japanese restaurant, one can learn the situations in which this assumption ought to be checked explicitly. Other failures arise out of other kinds of shortcomings in the reasoner's model of the domain. In general, there are several types of knowledge goals that might arise out of difficulties during processing, and different types of learning that correspond to these knowledge goals. We are developing learning algorithms that deal with different types of processing failures, and investigating the extent to which these learning algorithms can be integrated into a single multistrategy learning system [Cox and Ram, 1991; Ram and Cox, 1993].

In this article, we have presented explanation-based learning techniques for building and improving the case library for a case-based story understanding task within this general framework. Learning occurs incrementally when the understander encounters the following difficulties:

1. **Novel situation:** In a truly novel situation, an applicable case may not be available. The reasoner simply does not have a prior experience that provides it with a case that is relevant to the current situation. In this situation, abstract explanation schemas are applied to the situation, resulting in the creation of a new explanatory cases through explanation-based refinement.

2. **Mis-indexed cases:** The reasoner may have a case that is applicable to the current situation, but it may be unable to retrieve it since the case is not indexed under the cues that the situation provided. In this case, new indices are learned to an existing explanatory case to allow it to be used in novel contexts.

3. **Incorrect or incompletely understood cases:** Previous experiences, especially in novel and complex domains, may not have been completely understood, and so cases corresponding to them may be incomplete or incorrect. In these situations, explanatory cases are incrementally modified and elaborated through a process of question generation and gap filling.

The net result of these processes is that the case-based understanding system incrementally improves its understanding of its domain through experience. It uses its current knowledge, even though it is incomplete, to process the new situation as best as it can, and improves the quality of its cases and their indices.

[9] Since knowledge goals are often voiced out loud in the form of questions, we use the terms *questions* and *knowledge goals* interchangeably in this discussion.

14 Evaluation of the methods

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. AQUA can understand several variations of 10 basic types of stories, one for each stereotypical XP that it has. For example, AQUA can understand stories about religious fanatics, depressed teenagers, Kamikazes, and so on. The story can be varied to include different actors, actions, outcomes, and so on, as long as frames for these actors, actions, outcomes, etc., are represented in the system's memory. AQUA has been run on five or six different newspaper stories about terrorism, which have been simplified to fit within the English language constructions that AQUA can deal with. AQUA has also been run on several variations on these stories, such as story S-5 below, and a few other stories that have not been taken from actual newspapers.

In addition, AQUA can also understand stories which involve one or more of its abstract XPs, such as stories about people performing actions to achieve their goals, people planning actions that they do not want to perform themselves by using other actors, and so on. Any of the 50 or so MOPs represented in memory can occur as the underlying action, and of course a story could involve more than one such action. Thus in addition to the stereotypical stories mentioned above, AQUA can understand a large range of basic "goal-based" stories based on its abstract XPs. For example, AQUA can read a variation of the blackmail story in which a gullible teenager is tricked into performing a suicide mission without knowing its outcome:

> **S-5: Terrorists recruit boy as car bomber.**
> A 16-year-old Lebanese got into an explosive-laden car and went on a suicide bombing mission to blow up the Israeli army headquarters in Lebanon. ...
> The teenager was a Shiite Moslem but not a religious fanatic. He thought he was being recruited as a limousine driver. He did not know that the deadly mission would result in his own death.

This story is initially processed in the same manner as the blackmail story. However, in this case it is `xp-not-know-outcome` that is confirmed, resulting in the learning of a new stereotypical XP through explanation-based refinement, and of new indices to this XP.

The learning algorithms in this article are fully implemented, with the exception of the node and link elaboration step in the EBR algorithm. AQUA learns from stories in which an XP with attached questions is retrieved and used to build explanations, or a known XP provides an explanation in a new context involving an action that the XP is currently not indexed under. AQUA does not learn new MOPs, nor does it learn new XPs that are not incremental variations on old ones that it already knows. Thus the MOPs and XPs in memory also provide a constraint on the stories that can be processed. Although the abstract XPs that have actually been implemented are fairly complete for the kinds of motivational stories that AQUA is designed to deal with, additional XPs for planner-actor relationships would be required to understand stories with complex goal interactions. The motivational aspects of most of the suicide bombing stories that have appeared in the newspapers over the past several years fall within variations of the religious fanatic and coercion themes that AQUA knows about, but AQUA would need several more stereotypical XPs and MOPs (and a larger natural language lexicon) in order to read a wider range of stories.

Theoretical evaluation: Theoretical (as opposed to implementational) strengths and limitations of the program are discussed next. The performance task in AQUA is somewhat different to that of most machine learning systems. We are more concerned with the functional utility of learned concepts rather than their accuracy. We are concerned with being able to learn the types of XPs required by the case-based explanation

program that underlies the story understanding system, and, as demonstrated by the examples, to be able to improve the quality of the explanations produced by the program. The theory of case-based explanation provides a context for, and hence provides constraints on, the learning algorithms. In particular, we rely on the *content theory* of explanatory cases to determine the levels of generalization and refinement.

AQUA is not designed to learn the "right" concepts underlying terrorism or the "correct" definition of religious fanaticism. Instead, its task is to become "better" at understanding stories in its domain. AQUA learns the concepts that are useful in performing this task. One of the claims of our theory of question-driven understanding is that asking good questions is as important to understanding as is answering them [Ram, 1989; Ram, 1991]. Learning the questions to ask when the case is next applied is a central issue since this allows the system to reason about what it does not yet know but needs to find out. These questions focus the understanding process during future stories that might answer the questions. Here again there are no "right" questions, only "better" ones (those that help the system to learn) and "worse" ones (those that miss the point of the input stories).

Since it is difficult to measure the quality of questions or explanations in a quantitative manner, the performance of the system is determined by the quality of its output (questions, explanations) on different sequences of input stories rather than by quantitative measures (such as the speed of explanation construction), which do not correlate with the quality of the explanations produced or the depth of the system's understanding of the stories. The scope of our theory is determined by the behavior of the system on stories that provide "boundary conditions" for the program, in the sense that they represent interesting borderline cases with which the theory can be tested.

Quality of output: First, let us consider the issue of how AQUA's questions and hypotheses change as it reads several stories about suicide bombing. Since AQUA has more knowledge about the domain, it can ask more questions about a new story that it reads. On the other hand, fewer stories would be novel since AQUA already knows a lot about that domain, and so fewer new questions would be raised.

In other words, as AQUA gets more "expert" in its domain, most common stories that it sees fit in fairly well with what it already knows. They raise very few questions, and so these stories are not very interesting to the program. On the other hand, AQUA asks more and better questions about stories that are novel. Furthermore, the unanswered questions that are pending in AQUA's memory are more sophisticated than the ones that it started out with, and reflect a better understanding of the domain.

This is demonstrated by AQUA's improved ability to understand the examples used earlier to illustrate the learning algorithms. It is difficult to quantify this improvement. Traditional learning curves (e.g., "speed of learning") and performance curves (e.g., "speed of understanding" or "number of explanations") do not adequately capture the kind of qualitative improvement that we are seeking. Developing methods to evaluate such systems in a more precise manner is an important issue for further research. Here, we use a series of examples to demonstrate the improvement in AQUA's performance as it learns about its domain.

Quality of explanations: Consider the quality of the explanations built by the system for the input story used earlier to illustrate the EBR learning process. There are two types of explanations one could build for any event, abstract and specific, as discussed earlier. The specific explanation based on `xp-religious-fanatic` is easier to construct because this XP provides more details than `xp-goal-sacrifice`, and easier to use because it provides better and more specific expectations about the story. The predictions from `xp-goal-sacrifice` are so general as to be virtually useless. On reading about a suicide bombing attack, for example, `xp-goal-sacrifice` would only predict that there was some unknown goal of the agent that was more valuable to the agent than his or her own life, but not what the unknown goal was, nor why it was more important than life. Furthermore, considerable inference is required to complete the explanation because inferring what the unknown goal might be is very difficult. Thus learning the specific explanatory case,

`xp-religious-fanatic`, enables the system to construct more detailed explanations with less inferential effort for this and other suicide bombing stories. Furthermore, as discussed below, the ability to form better and more specific predictions facilitates the process of story understanding.

Thus EBR results in better, more detailed and easier to construct explanations, without sacrificing the abstract planning and decision knowledge embodied in the abstract XPs that represent more general decision models. Whereas the refined schema is less widely applicable than the abstract schema that the reasoner started with, it provides a more detailed explanation for the specific situation that it applies to. Note that the underlying causality is still accessible because the internals of the XP are accessible to the system. Thus the internal causal structure of this explanation can be elaborated to provide a detailed motivational analysis in terms of abstract volitional explanations if necessary.

In addition to being more useful on functional grounds, EBR-built explanations (e.g., "Because she was a religious fanatic") correspond to our intuitions about the level of explanations normally used by people. We hypothesize that people tend to prefer the specific explanation because it summarizes the generic details of `xp-goal-sacrifice` (a commonly known explanation) and focusses attention on the `high-religious-zeal` of the agent (the unusual aspect of this particular explanation).

Understanding a new example: Let us further demonstrate AQUA's improved understanding of its domain by examining its performance on a new story with and without the benefit of its experiences with the example stories discussed in this article. Consider the following story (New York Times, February 27, 1986):

> **S-6: Lebanon car bomb kills driver, hurts 7 at Palestinian site.**
>
> BEIRUT, Lebanon, February 26 — A car bomb exploded today at the entrance of the largest Palestinian refugee district in southern Lebanon, killing the driver and wounding seven people.
>
> The police said the explosion occurred outside the Ain Khilwe camp, near the port of Sidon ... . A guard at the entrance of the camp ... said he saw the driver trying to get out of the car. "He struggled with the door, then the whole car exploded with him inside," the guard said.

On the surface, this story looks like a stereotypical suicide bombing story. The only quirk in this story is that the driver appears to have changed his mind. But this is pretty understandable; perhaps he was frightened and changed his mind at the last minute. Or perhaps it was part of the plan all along that the driver would jump out of the moving vehicle. These possibilities lead to two different explanations for the "struggling to open the door" action. AQUA builds the first explanation using an abstract explanation schema that says that people perform actions (trying to open the door) to disenable their own plans (driving the car) if they change their minds about carrying out the plan, and the second explanation using an explanatory case involving a known plan for terrorist car bombing whose final scene involves the actor jumping out of the car. In this story, neither explanation can be confirmed since there is insufficient information to do so.

Range of understanding: Now consider what happens when AQUA reads the same story after having read the blackmail story S-3. In this case, AQUA can view this story as confirmation of the hypothesis that suicide bombers do not volunteer for these missions, but instead are forced into them by extortion (or perhaps exhortation). This story does not prove that the hypothesis is "true," of course. The point is that the questions and hypotheses currently in memory can affect one's interpretation of a story. In this case, AQUA's experience with the blackmail story causes it to build an interesting hypothesis for story S-6 which it otherwise could not have built. This illustrates a *wider* understanding of the domain as a result of learning.

Depth of understanding: Next, suppose AQUA had already read story S-4 about the girl who was willing to die to protect her family. Recall that this story had answered the question of what could be more important than life for these agents. On reading story S-6 after this, AQUA would not only build the hypothesis that the driver may have been forced into going on this mission through coercion, but would also ask what the driver's family relations were like. This question illustrates a *deeper* understanding of the domain as a result of learning. Not only has AQUA learned a wider range of explanatory possibilities for terrorism (coercion), but also it has a deeper understanding of these possibilities (the role of family ties). It can build more elaborate explanations and ask more sophisticated questions.

Again, there is no guarantee that the family-relations question is the right one for this story. However, asking the question is desirable for two reasons. Firstly, it illustrates a deeper understanding of the domain since it results from the elaborated coercion hypothesis shown in figure 8. This hypothesis often occurs to people as a possible explanation for story S-6 after reading these stories in succession. Secondly, from the computational point of view, asking the question sets up an expectation that makes it easier for the system to understand the story if the question does turns out to be relevant. (The role of expectations in understanding was pointed out by Schank [Schank, 1978; Schank and Abelson, 1977].) Suppose the story is modified to continue as follows:

> **S-6:** (continued) In a statement issued the next day, [the driver's] family said that their son loved them very much and would do anything for their sake.

If AQUA has not read stories S-3 and S-4, it cannot create a coercion hypothesis for S-6. In this case, the last sentence does not fit into any of the hypotheses (religious fanaticism, changing one's mind at the last minute, part of the plan all along). Thus AQUA cannot connect this sentence to the motivations of the boy in performing the terrorist attack. There are two alternatives in the design of story understanding systems at this point: (1) to not integrate this sentence with the rest of the story representation, thereby not understanding part of the story, or (2) to use exhaustive search using a chaining-type inferencer, which would eventually find the blackmail possibility if the inference rules were set up correctly, but only after considerable inference. However, if the question is present, it is easy to match the new statement to the pending question and thereby confirm the coercion hypothesis. The ability to use past experiences to provide expectations for new situations is precisely the point of case-based reasoning.

Scope of the methods: To summarize, then, AQUA's learning methods result in a qualitative improvement in the explanations and questions produced by the system, which in turn allow the system to come to a deeper understanding of a wider range of input stories with less effort. Both from the cognitive or depth-of-understanding viewpoint, as well as the computational viewpoint, the indexing, elaboration, and refinement mechanisms we have presented lead to improved performance as the system learns through experience. Ultimately, the evaluation of the explanations produced by the system (and therefore of the performance of the system) must be done with respect to the goals of the system, that is, with respect to the reasons for which the explanation is being produced in the first place [Ram and Leake, 1991]. For this purpose, AQUA can be set up with a set of "initial questions" to be answered, which define the initial learning goals of the system. AQUA's learning algorithms formulate those hypotheses, and generate those new questions, that help to answer its questions, whether pre-programmed as initial questions or self-generated as knowledge goals. As illustrated by the above examples, AQUA's answers to these questions become better as it learns.

The scope of our theory is discussed next in the context of the "boundary condition" stories mentioned earlier. These stories represent interesting borderline cases with which the theory can be tested.

Bizarre stories: The learning methods presented here have important implications for the design of case-based reasoning systems that can learn, refine, and index cases, and thereby build their own case libraries

through experience. The strength of these methods derive from their incremental and knowledge-intensive nature. However, these properties are also responsible for the major limitations of the methods. Although AQUA can understand novel stories, stories that are too deviant fall outside its range. A story that is too bizarre, in the sense that it does not relate to any case or abstract XP that the program knows about, is difficult to understand precisely because the program does not have the knowledge structures to even begin to process the story.

AQUA learns through incremental modification of its XPs. Thus in order to learn from a novel story, it should be possible to understand these novel aspects using the kinds of modifications that AQUA is capable of performing. Again, a story that requires large, non-incremental modifications to existing knowledge structures would fall outside the scope of AQUA's methods.

Stereotypical stories: At the other extreme, a story that fits perfectly within existing knowledge structures is easy to understand. But by the same token, such a story is not very interesting since it does not say anything new. When AQUA reads a story that fits well with what it already knows, no new questions are raised. The processing questions that arise are easily answered. Although AQUA can read these stories, therefore, it will not have learned anything new as a result, nor would it have asked a new question. This is to be expected since if no processing difficulties are encountered, no learning is needed.

Misleading stories: Stories are often biased, incorrect, untrue, or otherwise misleading. This may be intentional (e.g., mystery stories are designed to mislead the reader) or unintentional (e.g., newspaper coverage of ongoing events). AQUA has no mechanisms for questioning the validity of the facts presented in the story, or for reasoning about the motivations of the author of the story. Although stories may be incomplete (e.g., a story may raise a question but not answer it), explanations explicitly provided by the story are assumed to be correct. For example, AQUA does not reason about the possibility that blackmail may in fact not be a possible explanation for suicide bombing, since it has read a story in which it was stated to be the explanation for an instance of suicide bombing.

15 Comparison with other work

In this article, we have presented three classes of reasoning difficulties — those arising from missing, misindexed, and incomplete knowledge — that may be encountered during case-based reasoning, and we have presented algorithms that allow a reasoner to learn through experience with situations involving these difficulties. We now discuss the main points of our theory in the context of other related work in case-based reasoning and machine learning.

AQUA's case-based explanation process is similar to that used by SWALE [Kass *et al.*, 1986], but is formulated in a question-based framework that provides a basis for integrating explanation, natural language understanding, memory, and learning [Ram, 1989]. Although both programs are based on Schank's [1986] theory of explanation patterns, the emphasis in AQUA has been on the questions that underly the creation, verification, and learning of explanations, and not on the creative adaptation process described by Kass, *et al.* Furthermore, unlike SWALE, AQUA can use incomplete XPs that have pending questions attached to them, and learn as these questions are answered. AQUA can also learn new indices to its XPs. A final difference between AQUA and SWALE is AQUA's ability to use XPs representing both stereotypical explanatory cases as well as abstract explanation schemas to build explanations for new situations.

Explanation-based refinement is related to theory-based concept specialization [Flann and Dietterich, 1989; Mooney,], which involves the inductive specialization of a concept defined by a domain theory. The emphasis in TBCS, however, is on the correctness of the learned concept, whereas EBR is more concerned

with the *quality* of the concepts, and the *functional utility* of these concepts with respect to the task for which the concepts are being learned in the first place. (Schank *et al.* [1986] call this a *pragmatic* constraint.) AQUA's use of its questions to focus the learning process cause it to learn those concepts that are useful in answering its questions, even though an alternative characterization may exist that may be provably correct in a theoretical sense but irrelevant from the point of view of the program's goals. EBR is also related to the explanation-based specialization algorithm used in PRODIGY [Minton, 1988] to map problem-solving traces into explanations; however, EBR concentrates more on the adaptation of previously known abstract explanations (using mechanisms of substitution, internalization and elaboration) rather than the generalization of problem-solving traces.

We are also interested in the *content* of the explanations, which is constrained by the needs of AQUA's XP-based understanding algorithm. There are many correct specializations of an abstract schema such as goal sacrifice; the issue is which one to learn given the functional role that it will play in the theory of XP-based understanding. Although we use story understanding as the task, the approach could be used to learn specific causal patterns in any case-based reasoning situation in which a system uses explanatory cases to reason about novel experiences.

Content theories also play an important role in AQUA's index learning algorithms. AQUA's use of explanation-based learning methods for identifying indices is similar to Barletta and Mark's [1988] explanation-based indexing (EBI) algorithm. An important difference, however, is that we have identified classes of stereotypical concept descriptions that constitute good indices in our domain. These descriptions comprise a *content theory* of indexing. In a content-based approach to index learning, one enumerates good indices for the domain that the system is dealing with, and develops methods that can be used to learn the particular kinds of knowledge that are known to make good indices in the domain. For example, in predicting the motivations of people for a story understanding task, it is useful to categorize the characters in the story into stereotypical groups that tend to use particular kinds of plans in achieving their goals. Causal explanations for motivations can then be indexed using these stereotypes, and index learning can be viewed as the problem of learning effective characterizations of stereotypes of people. (However, we still require that the index learning algorithm should use one or a small number of generalization rules, such as R-3, for all types of indices.)

A *structure-based* approach to index learning, on the other hand, uses heuristics based on the structure of the learned knowledge to extract indices that are likely to be useful. For example, since causally prior events are likely to predict later events in a story, it is useful to index explanatory schemas by the antecedents of the explanations represented in the schemas. This heuristic is independent of the actual content of the explanations, or the domain of applicability of the schemas, relying instead on the nature of causal relationships in the domain (e.g., [Bhatta and Ram, 1991]).

Barletta and Mark's EBI algorithm identifies indices using a process similar to goal regression in explanation-based generalization (EBG) [Mitchell *et al.*, 1986]. Any feature identified in this manner can be used as an index. However, such an approach could select features that do not make good indices because they are not easy to observe directly (e.g., "actor's intent was to jump out of the car at the last minute"), because they are expensive to compute or prove (e.g., "actor is gullible"), or because they do not provide adequate discrimination (e.g., "actor is a human being"). Instead, our approach has been to use stereotypical sets of indices that are easy to observe or infer and are likely to be predictive (e.g., "actor is a teenager"). Within this constraint, our methods are similar to EBG. The disadvantage of our approach is that new *classes* of indices cannot be learned, only new indices within existing classes. We are currently investigating the use of combinations of structure-based and content-based approaches to overcome the limitations of each.

AQUA learns from positive and negative hypotheses resulting from episodes of case-based reasoning using XPs. However, it does not perform any explicit comparison between its hypotheses, unlike Falkenhainer's system which exploits differences between similar situations to focus search and generate plausible hypotheses [Falkenhainer, 1988]. It would be instructive to use a difference-based reasoning method similar

to Falkenhainer's to improve the quality of the generalizations produced to be used as indices in AQUA's index learning algorithm.

Although AQUA is not a concept learning program in the traditional sense, it is useful to contrast its approach to indexing with other case-based concept learning programs that also focus on the indexing problem. For example, PROTOS uses exemplar differences, censors, and prototypicality measures to retrieve exemplars with a high degree of match similarity to the current problem [Porter *et al.*, 1990]. While PROTOS's task is to classify the input by assigning it to one of the categories in a pre-enumerated list, AQUA's task is to explain anomalies by building causal explanations. This requires assessment, not of the match similarity of features of cases, but rather of the applicability of known explanations to a given anomalous situation. For comparison purposes, AQUA's indexing method may be characterized the assessment of the similarity of the causal structure of cases in terms of the abstract language of decision models, as opposed to the assessment of the similarity of the feature vectors that describe cases in terms of the specific language of the domain.

AQUA's use of explanations to constrain learning is similar to the use of causal knowledge in explanation-based generalization. However, traditional work in explanation-based learning has focussed on the creation of new schemas by the generalization of explanations (e.g., GENESIS [Mooney and DeJong, 1985]) or problem-solving traces (e.g., LEX2 [Mitchell, 1983]) created through backchaining or other exhaustive search processes. In contrast, AQUA's approach relies on an incremental modification of explanation structures while they are used to construct explanations through case-based reasoning. Unlike GENESIS and LEX2, therefore, AQUA is incremental and case-based.

AQUA's approach to learning is empirical but theory-driven. One extension of this approach would be to combine it with correlational learning methods. For example, OCCAM also specializes schemas by combining correlational information between events and prior causal theories which explain regularities between events [Pazzani *et al.*, 1986]. At present, AQUA must rely on incremental theory-driven modification of its XPs through single experiences. Correlational information could be used, for example, to decide the status of superficial features (such as the fact that the bomber in a particular story was young) that cannot be causally linked to a relevant aspect of the explanation. As Barletta and Mark [1988] point out, a reasoner will in general not be able to show that every feature is relevant or irrelevant in situations where the reasoner's theory of the domain is incomplete. Their system retains "possibly relevant" features as "secondary indices," which are refined through induction. We are working on developing algorithms by which a reasoner can later re-classify these features as "relevant" or "irrelevant," based on future experiences in the domain. Unlike Barletta and Mark's algorithm, however, we are focussing on the use of explanation-based methods for index refinement.

We are also exploring better methods for evaluating our approach. For example, Minton and Carbonell's PRODIGY system sometimes slows down with learning [Minton, 1988]. While slowing down is not necessarily bad in itself, better evaluation methods are needed to evaluate the relative merits of producing simpler explanations in less time as opposed to better (and perhaps more complex) explanations that may take longer to compute. Ultimately, the evaluation of the desired explanation must be done with respect to the goals of the system, that is, with respect to the reasons for which the explanation is being produced in the first place (e.g., see [Leake, 1989c; Ram, 1990b; Ram and Leake, 1991]). The effect of this constraint on the evaluation of the learning algorithms is an open issue. Also needed are finer grained evaluation techniques to isolate the evaluation of AQUA's learning algorithms from the evaluation of the case-based reasoning paradigm itself (including the individual effects of the A-1, A-2, and A-3 assumptions).

16 Conclusions

Understanding requires the ability to construct explanations for novel and anomalous situations. In the case-based reasoning paradigm, explanations are constructed by applying stereotypical packages of causality

from similar situations encountered earlier (explanatory cases, represented as stereotypical XPs) and from general domain knowledge (explanation schemas, represented as abstract XPs). This article addresses the issue of the formation of a case library of explanatory cases in a novel and complex domain. Much work in explanation-based learning has focussed on the problem of learning through the generalization of causal structures underlying novel situations. However, it is difficult to determine the correct level of generalization. Furthermore, many stories do not provide enough information to prove that the explanation is correct. The understander must often content itself with two or more competing hypotheses, or otherwise jump to a conclusion. This means that in a real world situation, an explanation-based learning system may still need to deal with the problem of incomplete or incorrect domain knowledge.

In general, the system's memory of past experiences will not always contain "correct" cases or "correct" explanations, but rather one or more hypotheses about what the correct explanation might have been.[10] These hypotheses often have questions attached to them, representing what is still not understood or verified about those hypotheses. As the understander 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, its understanding of the old case gradually gets refined. New indices are learned as the understander learns more about the range of applicability of the case. The case is re-indexed in memory and is more likely to be recalled only in relevant situations. Each type of learning leaves the reasoner a little closer to a complete understanding of its domain. Each type of learning could also result in a new set of questions as the reasoner realizes what else it needs to learn about, which in turn drives the reasoner towards further learning.

Thus XP learning is an incremental process of theory formation, involving both case-based reasoning and explanation-based learning processes. We have presented a theory of case-based learning through the incremental modification and indexing of existing XPs, using explanation-based learning techniques to constrain the learning process. The modifications involve the adaptation and elaboration of existing XPs, the refinement of abstract XPs, as well as the learning of indices for XPs. The theory is implemented in the AQUA program, which learns about terrorism by reading newspaper stories about unusual terrorist incidents.

Acknowledgments

This research was supported in part by the National Science Foundation under grant IRI-9009710. Part of this research was carried out while the author 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.

[10] Actually, a single story or episode can provide more than one "case," each case being a particular interpretation or dealing with a particular aspect of the story. For an explanation program such as AQUA, each anomaly in a story, along with the corresponding set of explanatory hypotheses, can be used as a case.

References

[Barletta and Mark, 1988] R. Barletta and W. Mark. Explanation-Based Indexing of Cases. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 541–546, St. Paul, MN, August 1988.

[Bhatta and Ram, 1991] S. Bhatta and A. Ram. Learning indices for schema selection. In M. B. Fishman, editor, *Proceedings of the Fourth Florida Artificial Intelligence Research Symposium*, pages 226–231, Cocoa Beach, FL, April 1991. Florida AI Research Society.

[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 Multistrategy Learning*, pages 217–230, Harpers Ferry, WV, November 1991. Center for Artificial Intelligence, George Mason University, Fairfax, VA.

[DeJong and Mooney, 1986] G. F. DeJong and R. J. Mooney. Explanation-Based Learning: An Alternative View. *Machine Learning*, 1(2):145–176, 1986.

[DeJong, 1983] G. F. DeJong. An Approach to Learning from Observation. In R. S. Michalski, editor, *Proceedings of the 1983 International Machine Learning Workshop*, pages 171–176, Monticello, IL, June 1983. Department of Computer Science, University of Illinois, Urbana-Champaign.

[Dietterich and Michalski, 1981] T. G. Dietterich and R. S. Michalski. Inductive Learning of Structural Descriptions: Evaluation Criteria and Comparative Review of Selected Methodologies. *Artificial Intelligence*, 16:257–294, 1981.

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

[Falkenhainer, 1988] B. Falkenhainer. The Utility of Difference-Based Reasoning. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 530–535, St. Paul, MN, August 1988.

[Flann and Dietterich, 1989] N. S. Flann and T. G. Dietterich. A Study of Explanation-Based Methods for Inductive Learning. *Machine Learning*, 4:187–226, November 1989.

[Gupta, 1987] A. Gupta. Explanation-Based Failure Recovery. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 606–610, Seattle, WA, July 1987.

[Hammond, 1989] K. J. Hammond, editor. *Proceedings of the Second Case-Based Reasoning Workshop*, Pensacola Beach, FL, May 1989. Defense Advanced Research Projects Agency, Morgan Kaufmann, San Mateo, CA.

[Hobbs *et al.*, 1990] J. R. Hobbs, M. Stickel, D. Appelt, and P. Martin. Interpretation as Abduction. Technical Note 499, SRI International, 1990.

[Kass and Owens, 1988] A. Kass and C. Owens. Learning New Explanations by Incremental Adaptation. In *Proceedings of the 1988 AAAI Spring Symposium on Explanation-Based Learning*, Stanford, CA, 1988.

[Kass *et al.*, 1986] A. Kass, D. Leake, and C. Owens. SWALE: A Program That Explains. pages 232–254. 1986. In [Schank, 1986].

[Keller, 1988] R. M. Keller. Defining Operationality for Explanation-Based Learning. *Artificial Intelligence*, 35:227–241, 1988.

[Kolodner, 1988] J. L. Kolodner, editor. *Proceedings of a Workshop on Case-Based Reasoning*, Clearwater Beach, FL, May 1988. Defense Advanced Research Projects Agency, Morgan Kaufmann, San Mateo, CA.

[Leake, 1989a] D. Leake. Anomaly detection strategies for schema-based story understanding. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, pages 490–497, Ann Arbor, MI, 1989.

[Leake, 1989b] D. Leake. *Evaluating Explanations.* Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, 1989.

[Leake, 1989c] D. B. Leake. The Effect of Explainer Goals on Case-Based Explanation. In *Proceedings of a Workshop on Case-Based Reasoning*, Pensacola Beach, FL, May 1989. Morgan Kaufmann, Inc.

[Michalski, 1983] R. S. Michalski. A Theory and Methodology of Inductive Learning. *Artificial Intelligence*, 20:111–161, 1983.

[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.

[Mitchell *et al.*, 1986] T. M. Mitchell, R. Keller, and S. Kedar-Cabelli. Explanation-Based Generalization: A Unifying View. *Machine Learning*, 1(1):47–80, 1986.

[Mitchell, 1980] T. M. Mitchell. The need for biases in learning generalizations. Technical Report CBM-TR-117, Rutgers University, Department of Computer Science, New Brunswick, NJ, 1980.

[Mitchell, 1983] T. M. Mitchell. Learning and Problem Solving. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 1139–1151, Karlsruhe, West Germany, 1983. Morgan Kaufman.

[Mooney,] R. J. Mooney. Explanation-based learning as concept formation. Presented at the Symposium on Computational Approaches to Concept Formation.

[Mooney and DeJong, 1985] R. J. Mooney and G. F. DeJong. Learning Schemata for Natural Language Processing. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 681–687, Los Angeles, CA, August 1985.

[Morris and O'Rorke, 1990] S. Morris and P. O'Rorke. An Approach to Theory Revision using Abduction. In *Proceedings of the AAAI Spring Symposium on Automated Abduction*, Stanford, CA, 1990.

[Mostow and Bhatnagar, 1987] J. Mostow and N. Bhatnagar. FAILSAFE – A Floor Planner that uses EBG to Learn from its Failures. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 249–255, Milan, Italy, August 1987.

[Pazzani *et al.*, 1986] M. Pazzani, M. Dyer, and M. Flowers. The Role of Prior Causal Theories in Generalization. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 545–550, Philadephia, PA, August 1986.

[Porter *et al.*, 1990] B. W. Porter, R. Bareiss, and R. C. Holte. Concept Learning and Heuristic Classification in Weak-Theory Domains. *Artificial Intelligence*, 45(1-2):229–263, 1990.

[Ram and Cox, 1993] A. Ram and M. T. Cox. Introspective Reasoning using Meta-Explanations for Multistrategy Learning. In R. S. Michalski and G. Tecuci, editors, *Machine Learning: A Multistrategy Approach, Volume IV.* Morgan Kaufman Publishers, Inc., 1993. To appear. Also available as Technical Report GIT-CC-92/19, College of Computing, Georgia Institute of Technology, Atlanta, GA, 1992.

[Ram and Hunter, 1992] A. Ram and L. Hunter. The Use of Explicit Goals for Knowledge to Guide Inference and Learning. *Applied Intelligence*, 2:47–73, 1992.

[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*, pages 867–871, Chicago, IL, August 1991.

[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, Hillsdale, NJ.

[Ram, 1990b] A. Ram. Goal-Based Explanation. In *Proceedings of the AAAI Spring Symposium on Automated Abduction*, Palo Alto, CA, March 1990.

[Ram, 1990c] 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, 1990d] 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, Hillsdale, NJ.

[Ram, 1991] A. Ram. A Theory of Questions and Question Asking. *The Journal of the Learning Sciences*, 1(3&4):273–318, 1991.

[Rieger, 1975] C. Rieger. Conceptual Memory and Inference. In R. C. Schank, editor, *Conceptual Information Processing.* North-Holland, Amsterdam, 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 *et al.*, 1986] R. C. Schank, G. Collins, and L. E. Hunter. Transcending Inductive Category Formation in Learning. *The Behavioral and Brain Sciences*, 9(4), 1986.

[Schank, 1978] R. C. Schank. Predictive Understanding. In R. N. Campbell and P. T. Smith, editors, *Recent Advances in the Psychology of Language — Formal and Experimental Approaches*, pages 91–101. Plenum Press, New York, 1978.

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

[Segre, 1987] A. M. Segre. *Explanation-Based Learning of Generalized Robot Assembly Tasks.* Ph.D. thesis, University of Illinois at Urbana-Champaign, Urbana, IL, January 1987. Technical Report UILU-ENG-87-2208.

[Stickel, 1990] M. E. Stickel. A Method For Abductive Reasoning In Natural-Language Interpretation. In *Proceedings Of The AAAI Spring Symposium On Automated Abduction.* AAAI, Menlo Park, CA, 1990.

[Wilensky, 1978] R. Wilensky. *Understanding Goal-Based Stories.* Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, 1978.

[Wilensky, 1981] R. Wilensky. PAM. In R. Schank and C. K. Riesbeck, editors, *Inside Computer Understanding: Five Programs plus Miniatures.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1981.