# Adaptation Strategies for Case-Based Plan Recognition*

Alex Kass
The Institute for the Learning Sciences
Northwestern University
1890 Maple Avenue, Evanston, IL 60201

## Abstract

Figuring out what plan another agent might be executing is an important and difficult type of explanation problem which involves a special type of knowledge about plans and goals. Elsewhere, we have discussed an approach to the general explanation problem that involves adapting stored explanations to new situations. In this paper, we review that discussion very briefly and then examine the special issues that arise when applying that approach to explaining intentional actions, focusing on adaptation strategies that are specifically relevant to plan recognition.

## Introduction

Making sense of other agents' actions can be hard. Reconstructing the chain of reasoning that led someone to believe that an action might service its goals is complex for several reasons: (1) the other agent's goals may be obscure; (2) the set of goals that a given action might conceivably serve can be large; (3) the chain of reasoning needed to connect actions to goals can be long and complex; and (4), the other agent's knowledge may be incomplete or even incorrect.

Plan-recognition is a type of explanation problem in which the explanations are cast in terms of goals and intentional actions. The principal problem in designing any explanation-building system is how to bring the system's causal knowledge to bear on a problem so that it can efficiently infer an unseen cause of observed events. In [Kass *et al.*, 1986, Kass, 1990a, Kass, 1990b] we have discussed an approach to the explanation problem that involves adapting stored explanations to new situations. In this paper we review that discussion and then examine the special issues that arise when applying that approach to explaining intentional actions, focussing on adaptation strategies that are specifically relevant to plan recognition.

## Previous approaches

Approaches to plan recognition have essentially fallen into two categories. The first approach was to chain inferences together on the fly, in response to each new problem. Examples of this approach include [Rieger, 1975] and [Wilensky, 1978]. Researchers who take this approach are led to concentrate on methods of controlling the combinatorial explosion inherent in the inference-chaining process. Parallel methods of rule activation, bi-directional search, and application of either domain-dependent or abstract, thematic search-control knowledge are among their important methods.

There is another school of research that seeks to view the entire explanation process as an indexing problem. These researchers postulate the existence of ready-made knowledge schemas, such as scripts or frames ([Schank and Abelson, 1977], [Cullingford, 1978], [Minsky, 1975], [Charniak, 1977]) that will provide all of the proper inferences, if only the right structure can be found in memory. This is the "retrieve and apply" school of explanation construction. Its concerns revolve around how these knowledge structures are represented, and especially how they are organized in memory so that the right one can be found at the right time.

Both approaches have crippling limitations. The script-application approach originally arose out of the realization that the "chaining + control" method was ignoring issues of memory. Rule-chaining systems treat each experience as completely novel, dooming them to perform expensive analyses every time they recognized a new instance of a plan. However, the script-application approach can only be employed when memory contains a script that *precisely* matches the plan to be recognized. In order for the matching process itself to be relatively inference-free, it is important that the expectations represented in a script be at a very specific level. This means that a "retrieve and apply" system is stymied when confronted with situations that stray even slightly from its specific expectations. In short, "chaining + control" is a weak approach because it knows nothing of stereotypes, and "retrieve and apply" is limited because it knows *only* its stereotypes. What's needed is a system that knows stereotypes, but also knows how to go beyond them when necessary.

## Adaptation-based Explanation

It is possible to build a system that enjoys both the efficiency benefits of employing stored knowledge structures *and* the flexibility to handle novel input by extending script/frame theory in the following ways:

*This work was supported in part by the Defense Advanced Research Projects Agency and monitored by the Air Force Office for Scientific Research under contract number F49620-88-C-0058. The Institute for the Learning Sciences was established in 1989 with the support of Andersen Consulting, part of The Arthur Andersen Worldwide Organization.

- Augmenting the knowledge structures stored in memory, so that in addition to representing specific expectations to be matched, each structure also includes a representation of the causal inferences that underlie those specific expectations.
- Building into the system strategies that make use of this causal annotation to *adapt* the structures it retrieves when they don't precisely match the plan being recognized.

In this augmented theory, scripts evolve into variablized explanations (called Explanation Patterns, or XPs for short), and "retrieve and apply" evolves into "retrieve, evaluate, adapt, apply". This is essentially an application of the case-based methodology applied to constructing explanations. (For a discussion of XPs see [Schank, 1986], [Kass, 1986], and [Leake and Owens, 1986]. For examples of some previous work relevant to case-based reasoning see, [Kolodner *et al.*, 1985], [Simpson, 1985], [Hammond, 1986], and [Sycara, 1987]). The system uses its pre-stored stereotypes as a starting point, but when it determines that its stereotyped explanations don't quite fit a new sitution, it can learn a new variation on one of those explanations, which, in addition to enabling it to understand the current input, can be stored in memory to help process future inputs.

ABE (Adaptation-Based Explainer) [Kass, 1990b] is an implementation of this theory that can develop explanations of unexpected deaths and disasters by adapting XPs stored in its memory to new contexts. For instance, ABE can adapt an explanation of a human dying from the exertion of recreational jogging to explain the death of a famous racehorse, or it can explain death of a star athlete by adapting an explanation of why rock stars die of drug overdoses.

ABE employs a library of XPs containing the explanations that the system has generated to solve past problems, as well as a set of patterns that is hand-coded into it. ABE has a number of built-in XPs about death, some volitional, others non-volitional. For instance, ABE knows explanations of death involving drug overdoses, overexertion, and terrorism. In addition to its XP library, ABE contains a large knowledge-base of specific facts and causal rules, and a semantic hierarchy for objects and actions.

The core of ABE is its library of tweaking strategies. Each strategy encodes an algorithm for making a small change to an XP in response to a specific mismatch between the XP and a particular anomaly. There are three main categories of tweaking strategies: Those which replace slot fillers with causal equivalents, those which generalize over-specific explanations, and those which elaborate incomplete explanations. Tweaks are more general than domain-specific heuristics, but they are quite knowledge-based. Each tweak knows how to search the system's semantic memory for domain-specific knowledge, and to use that knowledge to produce a slightly altered variation of a given XP. Thus, tweaking strategies relevant to plan recognition do not encode knowledge that is specific to any particular planning domain (that knowledge is encoded in the knowledge base). However, they do encode general knowledge about planning, about what can make a plan-based explanation more plausible, and about how plan-related knowledge can be found in the knowledge base. Since explanations involving intentional actions involve special types of beliefs and rules, it is useful for a system to employ special memory-search and inference techniques when modifying these explanations. The remainder of this paper will be devoted to illustrating these techniques through a discussion of three such strategies.

## Three plan recognition strategies

Explanations that involve volitional actions only make sense if the agents involved could reasonably be imagined to have performed each of the actions that the explanation hypothesizes they performed. In other words, evaluating whether such an explanation makes sense involves simulating the various agents' decision-making processes. This requires some specialized knowledge about motivation and decision-making. The simple utilitarian model of decision-making which underlies all three of the strategies described here is that an agent weighs the positive effects of a prospective action against the negative effects, and then decides whether to perform the action based on whether the positive outweighs the negative. These weighting functions are very subjective, of course, so that in order to accept the hypothesis that a given agent might have carried out a particular plan, an evaluator must attempt to understand the situation from the other agent's point of view. The system must simulate the agent's ability to project the effects of the action, then it must decide whether the action would be worth performing from the agent's point of view by summing the projected positive and projected negative effects and seeing whether the grand total is positive or negative. When the system's simulation of the agent predicts that the agent would give a negative score to an action which explanation claims the agent performed, the explanation will not be acceptable. Building an acceptable variant on such an explanation involves accounting for the discrepancy.

Let's consider this in terms of an example story, and how explanations can be adapted to provide an explanation of the story.

- **The Suicide Bomber Story**

  A teenage girl exploded a car bomb at a joint post of Israeli troops and pro-Israeli militiamen in southern Lebanon. The bomber and a number of Israeli soldiers were killed by the blast.[1]

ABE knows a standard TERRORIST BOMBING XP, which essentially states that people who feel oppressed will sometimes explode bombs in the vicinity of those

[1]See [Ram, 1989] for an in-depth discussion of many issues related to understanding this story.

who they consider the oppressor. Applying that XP to this story yields a plausible reason why the bomber killed her victims. But that explanation has a major inadequacy when applied to this particular story; the fact that the bomber did something which resulted in her own death, which is a central part of the anomaly raised by this is not addressed by the XP. In other words, the standard TERRORIST BOMBING XP is inadequate because the negative side effects of the action seem to outweigh the benefits.

The TERRORIST BOMBING XP is clearly a reasonable starting point for understanding this story even though it does not explain the entire anomaly. It would be a mistake to ignore the XP entirely. But in order to produce an explanation which adequately covers this story the TERRORIST BOMBING XP must be amended with sub-explanations that explain why the bomber might have decided that objective(s) achieved outweighed the negative side-effects.

ABE knows several strategies for elaborating its explanations in cases like this. Each strategy encodes a method of producing a reason that the system's simulation of the agent's plan evaluator might have been inadequate. Each can be conceptualized as an algorithm for asking — and answering — a specific question about the explanation. Such strategies do inference chaining, which can potentially be expensive. However, since they only add small sections to otherwise preconstructed explanations, the expense is much less than chaining together entire explanations from scratch.

In the sections that follow, we describe three plan-recognition-related adaptation strategies in terms of the general question that each strategy asks, examples of how that question is instantiated to build variations on the TERRORIST BOMBING XP that account for the SUICIDE BOMBER STORY, and some of the details of the memory-search and inference algorithms the strategy uses to answer that question.

## Elaborate motivation : Downplay negative

**Underlying question:** TWEAK 1 involves asking, **"Why would the negative effects of the action be particularly unimportant to the agent?"**

**Example:** TWEAK 1 searches for rules that indicate reasons why one of the bad side-effects of a problematic action might have been given less weight by the agent than the evaluator would have expected. In the suicide-bombing example, dying might have been less important to the suicide bomber if she were convinced that she knew she would die soon anyway, or if she were depressed enough to have decided that it wasn't particularly important to go on living.

**Discussion:** The system's knowledge base contain a set of inference rules called value-system adjustment rules. These rules indicate ways in which the desirability of certain states can become raised or lowered for a particular individual. The subcategory of value-system adjustment rules used by this strategy are those concerned with reasons that the importance of a preservation goal can become reduced. They are called *goal-state value decreasers*, and have the following form:

- **General form of a goal-state value decreaser:** Preserving goal $G$, will decrease in importance for agent $A$ if fact $F$ is true of $A$.

Examples of goal-state value-decreaser rules:

1. The goal of living in a neighborhood with a good school system will become less important to $A$ when $A$'s children get beyond school age.
2. The goal of staying alive may become less important to $A$ if $A$ has a chronic, painful illness.
3. The goal of staying alive may become less important to $A$ if $A$ is deeply depressed.

Goal-state value decreasers are used by TWEAK 1 to build a causal chain linking a state that the agent is in, to a reason why he might not have attached much importance to the projected negative effect. For each of the projected negative side effects of the problematic action, the system collects the value-decreasers relevant to that event-type. It then tries using those rules to construct a causal chain linking a fact about the agent to a reason he might value the negative side effect less than the evaluator originally predicted.

When trying to process the suicide bombing story, the system would retrieve rules 2 and 3, and attempt to infer that the agent was depressed or terminally ill.

**Inference and memory search:** There are two components to the task of building the causal chain mentioned above. The first is to retrieve relevant value-decreaser rules, and the second is to build the inference chain necessary to satisfy the antecedent portion of the rules. Recall that each of the rules has three slots, $G$, $A$, and $F$, corresponding to the *goal* whose value is decreased by the rule, the *agent* for whom it is decreased, and the *fact* which must be true about the agent in order to put the rule into effect. Retrieving rules which may be applicable involves finding all the rules whose $G$ slot can be filled by any of the event types that any of the negative side effects of the action fall into.

The steps followed in the attempt to build this chain are as follows:

1. The first step in the process is to collect all the negative side-effects of the action.
2. These events are used to retrieve goal-state value-decreasing rules. Event categories all have links to rules which call for events of that category to fill their $G$ slots. (For example, the event category, M-DIE has links pointing to rules like 2 and 3 above).
3. If any appropriate rules are found, the system then checks to see if the fact which fills the $F$ slot is known to be true of the agent in question. This means checking to see if the knowledge-base explicitly states that this fact is true of the agent himself, or of a category which the agent is in. If no such fact is retrieved, a limited amount of bi-directional inference

is performed to try to infer that $F$ is true of $A$. The system forward chains from the facts which are explicitly stated about $A$, and backward chains from rules which would imply $F$, attempting to find an intersection between the two chains. The amount of chaining is an adjustable parameter set by the system at large.

### Elaborate motivation : Amplify positive

**Underlying question:** TWEAK 2 involves asking, **"Why would the positive effects of the anomalous action be particularly important to the agent involved?"**

**Example:** TWEAK 2 is like TWEAK 1 in that it also looks for value-system adjustment rules which might be relevant to the story at hand, but TWEAK 2 looks for a different kind of rule. While TWEAK 1 looks for a rule indicating why a bad side could have been less important than normally expected, TWEAK 2 assumes that the negative side effects have their normal value, and instead searches for a reason to believe that one of the positive effects might have been *more* important than expected.

For example, the TERRORIST BOMBING XP could be made suitable to the suicide bombing story if it were elaborated to include a sub-explanation that showed why the death of the Israeli soldiers was especially important to the bomber; so important that it outweighed the negative effect of the bomber's own death. For instance, since the system knows that the importance of killing an enemy will increase for an agent if the enemy kills a member of the agent's family or one of his close friends, and it knows that many Palestinians have had family and friends killed by Israeli soldiers, it could hypothesize that one of the soldiers might have killed a friend or family member of the bomber.

**Discussion:** It is important to recognize that the main focus of this strategy is *not* to identify a positive effect of the action which the evaluator didn't originally project (ABE has another strategy — not described in this paper — which does that). The main idea behind TWEAK 2 is to isolate one of the positive effects which the original explanation projects, and to propose a previously-overlooked reason why the agent involved might have placed more importance than expected on that positive effect; in other words, this strategy works by proposing that an agent employed a different value system than expected, rather than proposing that he had different expectations.

The rules used to suggest an increase in how much importance an agent will place on a certain outcome are called *goal-state value increasers.* They have the following form:

- **General form of a goal-state value increasers**: Achieving goal $G$ will increase in importance for agent $A$ if fact $F$ is true of $A$.

Some examples of value-system modification rules of this type are:

1. The goal of acquiring alcohol will increase in importance for $A$ if $A$ is an alcoholic.
2. The goal of living in a safe neighborhood will increase in importance for $A$ if $A$ has children.
3. The goal of killing a military enemy will increase in importance for $A$ if the enemy directly harms someone close to $A$.

TWEAK 2 uses goal-state value increasers in a manner very similar to the way in which TWEAK 1 uses the value decreasers; the rules guide the strategy in its search for a causal connection between the problematic action and a state which the agent might have had an exceptionally strong desire to achieve. TWEAK 2 examines the states which typically result from the action and then searches for a goal-state value increaser which admits one of those states in its $G$ slot and whose $F$ slot matches a fact true about the agent.

It is important to emphasize that the system does *not* attempt to foresee every conceivable effect of the action. This would be too hard (the combinatorial explosion which results from considering the effects, and then the effects of the effects, and so on would quickly overwhelm the system); it would also be inappropriate because the idea is to simulate the agent's thinking, and the agent cannot be expected to have thought through every possible implication of his actions either. The set searched is a smaller one: those indexed as **stereotypical** effects of the action, and those which are mentioned as effects of the action by either the XP or the current story.

With regard to the suicide bomber example, the system would examine the effects of the bombing, principal among these being the deaths it caused. It would then retrieve the third of the above rules. Next it would try to determine if the soldiers might have killed anyone close to the bomber. Since it doesn't know anything directly about the bomber herself, it would examine what it knew about categories the bomber is in. Since the bomber is a Palestinian, and the system knows that many Palestinians have had friends and family killed by Israeli soldiers, it would construct the hypothesis that perhaps the bomber had had someone close to her killed by the Israeli soldiers, and that because of this, the value of killing the soldiers had become elevated above the counter-balancing desire to keep herself alive.

**Inference and memory search:** The following outlines how these rules can be put to use by the adapter to build the sub-explanations needed. Most of the inference algorithm is quite analogous to the one used by TWEAK 1; the main differences are that TWEAK 2 uses positive effects and value increasers where TWEAK 1 uses negative effects and value decreasers. There is one significant addition to the algorithm; if the explicitly-mentioned positive effects don't index any useful value-increasers, the strategy adds an extra step, in which inference is performed to determine implications of the positive effects which are explicitly mentioned by the XP, in order to see if the implications link to any goal-state value increasers.

1. The process begins by collecting the events which the original XP mentions as causal consequences of the problematic action. These events serve as the first candidates in the search for an event which may have had special importance to the agent.
2. The system then uses these candidate events collected as indexes into the library of goal-state value increasing rules, to see if there any rules in the knowledge base which call for the collected event to fill the $G$ slot.
3. If any appropriate rules are found, the system then checks to see if the fact which fills the $F$ slot is true of the particular agent involved in this situation. This involves checking if the knowledge-base explicitly states that this fact is true of the agent himself, or of a category which the agent is in. If that doesn't pan out, a limited amount of forward chaining from the facts which are true of the categories the agent is in is done to see if $F$ can be inferred to be true about the agent (the amount of forward chaining is one of the adjustable parameters set by the system at large).
4. If any of the events collected in step 1 index a rule which can be connected up to a category which the agent is a member of, then a new variation on the XP is built, incorporating this new causal chain. If no such chain can be built then a step of forward chaining from the events mentioned in the XP is performed, to form a new set of events to use as candidates, and the process goes back to step 2 to look for relevant rules again. In other words, if the immediate effects of the action don't yield any results, then the effects of those effects are tried. The loop continues until the forward-chaining limit is met, or until a reasonable chain is found.

## Elaborate motivation : Identify gap

**Underlying question:** TWEAK 3 involves asking, **"Which negative effect of his decision might the agent not have known about, and why?"**

**Example:** This strategy can be applied to suicide bombing example to give a different sort of answer than that given by the previous two strategies. Instead of looking for reasons why she would not have cared about dying (as TWEAK 1 does) or reasons why she would have been especially anxious to kill the Israeli soldiers (as TWEAK 2 does), TWEAK 3 would elaborate the TERRORIST BOMBING XP by identifying possible reasons that the bomber would have failed to predict one of the negative effects of the action, and therefore wouldn't have brought its negative value into the equation at all. For instance, by using this strategy the system could hypothesize that perhaps the bomber did not predict that she would die because she didn't realize that she would be inside the car at the moment of the explosion. Perhaps the people who convinced her to perform the bombing had not told her that the car would explode while she was inside.

**Discussion:** TWEAK 3 puts forth the hypothesis that an agent performed an anomalous action because he failed to project some negative effect of his actions. It attempts to explain why this would be by identifying which of the pieces of knowledge needed to infer the negative effect the agent might have been missing

There are two phases to this strategy. So the simpler part of the process is identifying a knowledge gap which might have been involved. The harder part of the process is to produce an explanation of *why* the knowledge gap might have existed. It isn't necessary to do this to produce a coherent explanation, but if such a sub-explanation is included it will make the overall explanation more convincing.

**Inference and memory search:** The first phase involves identifying a fact, $F$, such that if the agent hadn't known $F$, it would have caused him not to predict one of the negative effects of the action. This works as follows:

- First, collect up each expected negative side-effect of the action, along with the causal chain which allows one to infer that the negative side-effect.
- For each negative side-effect, $F1$, there is a set of other beliefs which comprise the inference chain which leads to the belief in $F1$. Each belief, $F2$ which appears on the inference chain leading to $F1$ is a candidate for a hypothesis of the form, "The agent did not anticipate $F1$ because he did not know that $F2$ was true."

The second phase involves building a sub-explanation to explain why it would be the case that "The agent did not know that $F2$ was true." This involves explaining why the inference leading to $F2$ would not take place when the agent projected the results of his action.

Building sub-explanations must be handled by another strategy, the general sub-explanation builder (not described in this paper, see [Kass, 1990b]). Some of the kinds of explanations relevant to why an agent might not believe a particular fact are as follows:

- Is there a plausible contradictory belief that the agent might have held? For example, the bomber could have believed that the car wouldn't blow up while he was inside because he believed that there would be a time delay between when the car stopped and when it exploded.
- Could the agent have lacked the time necessary to infer the relevant belief. Perhaps in the heat of the moment the agent had failed to think about the fact that her plan implied that she would be inside the car at time of the explosion.
- Could the agent have lacked the inference rule necessary to infer $F2$? Could the suicide bomber have been missing the rule that proximity to the bomb is

what determines how much damage someone is likely to suffer, and therefore failed to realize that someone inside the car would be killed by the explosion?

A prospective knowledge gap is not considered a viable candidate if ignorance of the belief would have caused the agent to fail to predict the main positive effect of his action. For instance, it is not reasonable to hypothesize that the bomber did not know that explosives would cause damage — even though this knowledge gap would account for the bomber's failure to predict her own death — because this would also have caused the bomber to fail to predict that her enemy would be killed. On the other hand, it would be reasonable to entertain the hypothesis that the bomber did not know she would be inside the car when the bomb went off since this would break the causal chain which leads to one of the negative side effects without breaking the chain leading to the main desired effect.

## Conclusion

A case-based explainer depends on the availability of adaptation strategies that know enough about the structure of the knowledge base to efficiently perform the memory search and inference necessary to build variations on stored explanations in a variety of domains. Because plan recognition calls for explanations containing the special classes of inference rules and memory links that are relevant to understanding intentional actions, it also calls for a specialized set of adaptation strategies which know how to employ those classes of rules and links.

In this paper we have attempted to contribute to the understanding of case-based plan recognition by presenting three such strategies. While the three strategies described here represent only a fraction of what is necessary in order to account for the full range of failures that can occur when applying old explanations to new situations ([Kass, 1990b] describes 21 strategies; there are probably more waiting to be discovered) they illustrate the kinds of knowledge that the knowledge base must contain, and the kinds of search and inference techniques that can be used to apply such knowledge in building variations on stored explanations when attempting to understand novel plans.

## References

[Charniak, 1977] E. Charniak. Ms. malaprop: A language comprehension program. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, Cambridge, Mass., August 1977. IJCAI.

[Cullingford, 1978] R. Cullingford. *Script Application: Computer Understanding of Newspaper Stories*. PhD thesis, Yale University, 1978. Technical Report 116.

[Hammond, 1986] K.J. Hammond. *Case-based Planning: An Integrated Theory of Planning, Learning and Memory*. PhD thesis, Yale University, 1986. Technical Report 488.

[Kass *et al.*, 1986] A. M. Kass, D. B. Leake, and C. C. Owens. Swale: A program that explains. In *Explanation Patterns: Understanding Mechanically and Creatively*, pages 232–254. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[Kass, 1986] A. Kass. Modifying explanations to understand stories. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, Amherst, MA, August 1986. Cognitive Science Society.

[Kass, 1990a] Alex Kass. Adaptation-based explanation: Extending script/frame theory to handle novel input. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 143–147, Detroit, MI, August 1990. IJCAI.

[Kass, 1990b] A.M. Kass. *Developing Creative Hypotheses by Adapting Explanations*. PhD thesis, Yale University, 1990. Also available as ILS Tech Report No. 6, Northwestern University.

[Kolodner *et al.*, 1985] J. Kolodner, R. Simpson, and K. Sycara. A process model of case-based reasoning in problem solving. In A. Joshi, editor, *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 284–290, Los Angeles, CA, August 1985. IJCAI.

[Leake and Owens, 1986] D. Leake and C. Owens. Organizing memory for explanation. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, Amherst, MA, August 1986. Cognitive Science Society.

[Minsky, 1975] M. Minsky. A framework for representing knowledge. In P. Winston, editor, *The Psychology of Computer Vision*, chapter 6, pages 211–277. McGraw-Hill, New York, 1975.

[Ram, 1989] Ashwin Ram. *Question-driven understanding: An integrated theory of story understanding, memory and learning*. PhD thesis, Yale University, New Haven, CT, 1989. In preparation.

[Rieger, 1975] C. Rieger. Conceptual memory and inference. In *Conceptual Information Processing*. North-Holland, Amsterdam, 1975.

[Schank and Abelson, 1977] R.C. Schank and R. Abelson. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1977.

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

[Simpson, 1985] R.L. Simpson. *A Computer Model of Case-based Reasoning in Problem-solving: An Investigation in the Domain of Dispute Mediation*. PhD thesis, School of Information and Computer Science, Georgia Institute of Technology, 1985.

[Sycara, 1987] E. P. Sycara. *Resolving Adversarial Conflicts: An Approach Integrating Case-based and Analytic Methods*. PhD thesis, School of Information and Computer Science, Georgia Institute of Technology, 1987.

[Wilensky, 1978] R. Wilensky. *Understanding Goal-Based Stories*. PhD thesis, Yale University, 1978. Technical Report 140.