Adapting Abstract Knowledge*

Eric K. Jones
The Institute for the Learning Sciences
Northwestern University
Evanston, IL 60201
jones@ils.nwu.edu

Abstract

For a case-based reasoner to use its knowledge flexibly, it must be equipped with a powerful case adapter. A case-based reasoner can only cope with variation in the form of the problems it is given to the extent that its cases in memory can be efficiently adapted to fit a wide range of new situations.

In this paper, we address the task of adapting abstract knowledge about planning to fit specific planning situations. First we show that adapting abstract cases requires reconciling incommensurate representations of planning situations. Next, we describe a representation system, a memory organization, and an adaptation process tailored to this requirement. Our approach is implemented in BRAINSTORMER, a planner that takes abstract advice.

Introduction

Most knowledge-based systems are unable to use their knowledge flexibly: they can only solve problems that are stated in exactly the right way. This is one aspect of the *brittleness* problem [Feigenbaum *et al.*, 1971], which limits the scope of many existing artificial intelligence systems both as practical tools and as cognitive models.

Even if a system contains the knowledge it needs to solve a problem, it may not be able to bring this knowledge to bear. At one extreme are systems that use radically incomplete inference procedures such as schema or script application [Cullingford, 1977], which although efficient, can only solve a limited class of problems. At the other extreme are systems that countenance multistep inference chaining [Rieger, 1976; Wilensky, 1978]. In principle, these systems are capable of solving a large number of problems, but in practice they make so many useless inferences that the number of problems they can actually solve under reasonable resource constraints remains small.

Case-based reasoning has been proposed as a promising middle ground [Hammond, 1986; Kolodner *et al.*, 1985]. Like schema appliers, case-based reasoners trade completeness for efficiency. Unlike schema appliers, however, case-based reasoners come with an adaptation component for resolving mismatches between past cases and current problematic situations. If the adapter can efficiently resolve an interesting class of commonly-occurring mismatches, then the case-based reasoner will be able to solve a wider range of problems than a schema applier. The flexibility of a case-based reasoner thus depends heavily on the power of its adapter.

In this paper, we consider a particular adaptation task: adapting abstract knowledge about planning to fit specific situations. A real-world activity in which this task arises is taking advice. People often communicate advice about planning in terms of high-level culturally-shared models of the planning process [White, 1987]. The vocabulary of these models may be very different from the vocabulary of the representations the person actually uses to solve the problem. In other words, the initial form of the advice is not necessarily operational for problem solving. The person taking the advice is therefore faced with the task of adapting abstract knowledge to fit her current problem solving needs.

We have built the BRAINSTORMER system to investigate the problem of adapting abstract knowledge [Jones, 1990; Jones, 1991]. BRAINSTORMER is a planner that takes advice about problems in the domain of terrorist crisis management. Whenever it encounters a difficulty, it asks for advice. A user then presents advice in the form of a proverb or aphorism, represented in a high-level vocabulary of culturally-shared planning concepts. BRAINSTORMER adapts the advice it receives by converting it into specific information in the operational vocabulary of the planner, which the planner can then use to resolve its difficulty and continue planning.

We have focused on proverbs for two reasons. First, proverbs are a well-defined class of abstract cases that encode useful culturally-shared knowledge about the idiosyncrasies of human planning and social interaction [Owens, 1988; Schank, 1986; White, 1987]. *The grass is always greener on the other side of the fence*, for example, expresses a peculiarity of the human planning process: when comparing options, people tend to be biased in favor of the unfamiliar or the unpossessed.

Second, representing a large number of proverbs has proved an effective strategy for developing and testing our representational vocabulary for advice. *The grass is always greener on the other side of the fence*, for example, not only encodes an important truth about human planning, but also points to the need for a conceptual

*This research was supported in part by DARPA contract F49620-88-C-0058 and AFOSR contract 89-0493. The Institute for the Learning Sciences was established in 1989 with the support of Andersen Consulting, part of The Arthur Andersen Worldwide Organization. Additional funding is provided by Ameritech, an Institute Partner, and by IBM.

vocabulary sufficient to encode the idea of *choosing between options in terms of estimates of their utility.*

BRAINSTORMER is capable of accepting a range of advice expressed in its high-level, culturally-shared vocabulary of planning concepts: proverbs are just one useful class of expressions representable in this vocabulary. Thus our approach to abstract case adaptation not only contributes to the field of case-based reasoning, but also addresses the more general problem of flexible understanding of arbitrary abstract advice.

The Problem

Taking abstract advice imposes particular inferential requirements that constrain the design of an adapter. BRAINSTORMER's descriptions of the problematic situations it faces are typically encoded in a very different vocabulary than the advice it receives; therefore, its adapter has to be able to infer the connection between the two. For example, one of BRAINSTORMER's goals is preventing terrorism, which it pursues by trying to analyze and counterplan against particular past terrorist attacks. There are many ways that a piece of abstract advice might characterize a terrorist attack, including *an act of frustration, an act of defiance, an act of revenge, an attack on civilians, a goal conflict, an illegal action, an immoral action, part of a propaganda campaign*, and *a political statement*, among others.

Conversely, there are many ways in which any of the above abstractions could be plausibly inferred from different specific planning situations as a planner might initially represent them. For example, a goal conflict can be plausibly inferred by observing an action on the part of one person that violates a goal of someone else, or from the knowledge that two different people hold incompatible goals, or by noticing that someone is angry.

It follows that adapting abstract knowledge requires being able to reconcile descriptions of given objects expressed in incommensurate vocabularies. This is essentially a problem of bidirectional search. As the above examples indicate, this problem is potentially serious, because the branching factor is high in both directions: on the one hand, there are many possible abstractions of a given concrete situation; on the other, there are many ways to plausibly recognize an instance of any particular abstract concept in a concrete situation.

In the remainder of this paper, we present an approach to adaptation sufficient to solve this problem. First, we give an example showing how the need to reconcile representations in incommensurate vocabularies arises in BRAINSTORMER. Second, we show that our underlying representation system is sufficiently powerful to represent the results of this process. As we will see shortly, this is a nontrivial requirement. Third, we sketch a memory organization that allows alternate encodings of a given situation in different vocabularies to be efficiently related. Finally, we specify an algorithm for this task, which works by redescribing one representation in the vocabulary of the other.

The Need for Redescription: An Example

A typical interaction between BRAINSTORMER and a user is as follows. BRAINSTORMER issues a query for an explanation of a particular past terrorist attack, as part of trying to come up with plans for preventing terrorism. The user responds with advice in the form of a representation of a proverb; let us suppose that the proverb is *no trouble but a priest is at the bottom of it*, which BRAINSTORMER represents as a causal connection between religious leaders and goal conflicts. Figures 1 and 2 depict the query and the initial representation of the advice.[1] It is the task of BRAINSTORMER's adapter to turn the advice into an answer to the query.

```
?explanation
   explained terrorist-attack
             actor group
                   typical-elt terrorist
```

Figure 1: A query from the planner.

```
cause
   cause  action
            actor religious-figure
   caused goal-conflict
```

Figure 2: Initial advice representation.

The adapter begins by transforming the advice into a form that could conceivably match the system's query. As the query asks for an explanation, the adapter converts the initial causal relation into an explanation structure based on that causality, as shown in figure 3.

The need for redescription arises in the next stage of adaptation, when the system tries to match or unify this newly created explanation structure with the query. No difficulty is encountered until an attempt is made to match the `terrorist-attack` to the `goal-conflict`. At that point, the matcher complains that these are instances of incommensurate concepts and starts trying to redescribe the terrorist attack in terms of goal conflicts.

```
causal-explanation
  explained goal-conflict =gc
  causals   (cause
               cause  action
                        actor religious-figure
               caused =gc)
```

Figure 3: The advice transformed into an explanation.

[1] Brainstormer uses a frame-based representation system with a slot-filler notation:

```
<frame>
    <slot1> <filler1>
    <slot2> <filler2>
    ...
```

Answers to queries are bindings to the free variables in the query, which are prefixed by "?"s. Variable binding constraints are expressed using the notation =<var>.

Redescription, Vagueness, and Dynamic Concept Formation

Before we can talk about how to efficiently carry out this redescription, we have to make clear what we want the result to look like. What did the user really mean in using the proverb *no trouble but a priest is at the bottom of it* to explain a terrorist attack? Presumably, that the religious figure caused the terrorist attack as part of causing a goal conflict. Notice that the intended meaning is *vague*, in that it doesn't specify exactly how the religious figure was involved. The output of the adapter should be a representation of this information at the right level of abstraction. What should this look like? We start with three inadequate answers to this question that point the way to a more acceptable one.

In BRAINSTORMER's ontology, a goal conflict exists if two different agents hold incompatible goals, the pursuit of one of which negatively impacts the other. Here we can safely assume that the terrorists had the goal to carry out a terrorist attack and that carrying out the attack violated goals of other agents: goals of people injured or killed in the attack, for example, and also BRAINSTORMER's goal to prevent terrorism. Therefore, various goal conflicts obtain as a consequence of carrying out the terrorist attack.

A first attempt to represent the result of adaptation might involve a four-step causal chain: a religious figure incited the terrorists to carry out the terrorist attack, causing the attack to occur, which violates a goal of someone else, thereby producing a goal conflict. Unfortunately, this representation is more specific than the user intends by the advice, as it asserts that the religious figure actually caused the terrorists to have the goal to carry out the attack. What the user really intends is vaguer. In particular, the religious figure may have been causally implicated in producing the terrorist attack in other ways as well: by facilitating its execution, for example, or by helping to ensure that its results would be effective.

A second possibility is to represent the advice using a one-step causal chain: the religious figure caused a goal conflict involving a terrorist attack. This representation, however, is too general. The output of adaptation is supposed to explain the terrorist attack, not the goal conflict. The proposed representation would cover a situation in which a religious figure caused someone to have the goal to prevent the terrorist attack; while this may well cause a goal conflict to exist once the terrorist attack occurs, it is hardly an explanation for the terrorist attack.

A third possibility is a disjunctive representation: list all of the ways that the religious figure could have helped the goal conflict come about that were also causally implicated in producing the terrorist attack, and represent the result of adaptation as the assertion that the religious figure interfered in one of these ways. While this approach adequately captures the meaning of the advice, it is still somewhat unsatisfying. Representing a vague piece of advice by cashing out all the ways it can be made more specific, while epistemologically adequate, is hardly concise or easy to reason with.

A truly satisfactory approach should allow us to represent the intended vague meaning of the advice nondisjunctively, by creating a first-class object to represent "the thing that the religious figure caused, leading to the terrorist attack and the goal conflict." That way, the advice can be represented and reasoned about directly, without having to enumerate the various ways it could be specialized.

But what in fact does the advice assert that the religious figure caused? The answer, we believe, is an instance of a brand new concept: an instance of the class of actions which, like the terrorist attack, are performed intentionally and violate a goal of someone else, thereby signaling a goal conflict. In other words, we suggest that the advice asserts that a religious figure caused the terrorist attack *under a new description* in terms of a set of recognition conditions or features for plausibly inferring a goal conflict.

More generally, an important part of using knowledge flexibly is being able to create and reason with novel concepts. Even if BRAINSTORMER has never before been asked to describe a terrorist attack in terms of goal conflict, it should be able to do so if asked. This imposes new demands on our representation language. Many knowledge representation systems are incapable of representing new concepts on the fly. Usually, all possible concepts are hard-wired into a type hierarchy in advance, and the system can only categorize inputs as being instances of these fixed concepts.

BRAINSTORMER, in contrast, can dynamically extend its base set of concepts by the mechanism of λ-abstraction [Sowa, 1984]: $\lambda(a)$(*representations mentioning* a) defines a new concept, whose instances are all of the a's that satisfy the conditions in the body of the lambda. In BRAINSTORMER, we represent λ-abstractions using the notation of *views*, which encode relationships between incommensurate representations of a single situation. Views are instances of λ-abstractions that permit one concept to be redescribed in terms of recognition conditions for another.[2]

Figure 4 shows our preferred representation of the advice after it has been adapted to fit the planner's query. The `part-whole-view` frame in the figure encodes a redescription of the `terrorist-attack` as an instance of the dynamically-constructed abstract concept $\lambda(a)$(*a is an intentional action of some agent that violates a goal of someone else*). This new concept was

[2] Actually, there are two kinds of views: *part-whole views* and *whole-whole views*. Only part-whole views create new concepts dynamically. Whole-whole views, in contrast, reference only preexisting concepts. For example, a terrorist attack can also be redescribed in terms of the preexisting concept of illegal action using a whole-whole view. Whole-whole views are similar in spirit to the views in Jacob's ACE system [Jacobs, 1987].

```
causal-explanation
  explained --
   part-whole-view =view
     source      terrorist-attack =attack
                   actor group =terrorists
                           typical-elt terrorist
     recog-conds (cause
                   cause achieve-goal =g1
                           actor =terrorists
                           state =attack
                   caused =attack
                 violates-goal
                   state =attack
                   goal prevent-goal =g2
                         actor brainstormer
                         state terrorist-attack)
     target      goal-conflict
                   actor1 =terrorists
                   actor2 brainstormer
                   goal1  =g1
                   goal2  =g2
     concept     (gc-schema2 lhs3)
  causals (cause
            cause  action
                     actor religious-figure
            caused =view)
```

Figure 4: The trace of redescription.

built out of a configuration of plausible recognition conditions for goal conflicts (determined, as we will see in the next section, by the schema **gc-schema2**). The filler of the **recog-conds** slot shows how this concept is instantiated by the terrorist attack: the terrorists had the goal to carry out the terrorist attack, and the attack violates BRAINSTORMER's goal to prevent terrorism. The **source** slot stores the **terrorist-attack** prior to redescription, while the **target** slot holds the **goal-conflict** that follows from the instantiated recognition conditions. The latter is the original **goal-conflict** of the advice, fleshed out to include the information supplied by the terrorist attack.

The **part-whole-view** in our example reifies "the thing the religious figure caused" as an instance of a new class of action, so that the system can reason about it without making any definite commitment as to how the religious figure caused it. This representation of the advice compactly encodes an answer to the planner's original query, which BRAINSTORMER can use to continue planning. As we saw above, the only way that BRAINSTORMER could represent this somewhat vague information without the benefit of views is in terms of a clumsy disjunction of all the ways the advice might be more specific. Of course, the system might later wish to postulate one or more of these specializations itself; if it does, the information necessary to do so is stored in the **recog-conds** slot of the view.

Memory Organization for Redescription

Now that we have specified the results we want from adaptation, we demonstrate a memory organization that

```
gc-schema2
  lhs1 goal =goal1              ; Someone has a goal
         actor agent =actor1    ; that an action occur
         state action =act
  lhs2 cause =cs                ; which causes the
         cause  =goal1          ; action to occur.
         caused =act
  lhs3 action =act
         actor agent
  lhs4 violates-goal =v         ; The action violates a
         state =act             ; goal of someone else.
         goal  =goal2
  lhs5 goal =goal2
         actor agent =actor2
  rhs  goal-conflict
         actor1 =actor1
         actor2 =actor2
         goal1  =goal1
         goal2  =goal2
  abductively-infer (=goal1 =cs)
```

Figure 5: A viewing schema for goal conflicts.

allows results like this to be efficiently computed. The basic task to be addressed is redescribing specific planning situations in terms of abstract concepts present in advice. As we have seen, this involves bidirectional search, and unfortunately, the branching factor is high in both directions. Therefore, neither chaining backwards from the advice nor forwards from the planning situation looks attractive.

One way to achieve greater efficiency would be to make the abstract concept work together with the specific planning situation to guide inference. This suggests the following general strategy for organizing memory to support redescription inference: create schemas for resolving differences in vocabulary and index them in terms of the endpoints of the inference chains they help to construct. Search can then take place in a much smaller space of ways to instantiate these schemas.

We have followed this approach in BRAINSTORMER. Associated with each of BRAINSTORMER's hard-wired abstract concepts such as goal conflict are **viewing schemas**, which are special rules for inferring that abstract concept. The antecedents of these rules are plausible recognition conditions for instances of the abstract concept; their consequents specify how the abstract concept should be instantiated. A viewing schema that can relate the terrorist attack to the goal conflict is shown in figure 5. The **lhs** slots are the schema's antecedents, the **rhs** slot its consequent. The **abductively-infer** slot is described below.

Viewing schemas are indexed in memory in multiple ways, under every conjunction of the form $(concept, recog_i)$, where *concept* is the concept that the schema infers, and the $recog_i$ are components of the schema's recognition conditions whose presence is predictive of the schema's applicability. The schema shown in figure 5, for example, is indexed in terms of (**goal-conflict**,**action**), (**goal-conflict**,**goal**), and (**goal-conflict**,**violates-goal**).

The Redescription Inference Process

BRAINSTORMER's memory of viewing schemas helps it to adapt abstract advice it receives to fit queries from the planner: whenever the system tries to match pairs of incommensurate concepts, it uses viewing schemas to efficiently resolve its difficulty. For example, we saw BRAINSTORMER attempt to match a **goal-conflict** in advice it is handed to a **terrorist-attack** in a query. These two concepts are incommensurate, so BRAINSTORMER retrieves all viewing schemas indexed in terms of them, or generalizations thereof.

The system's predefined concepts are arranged in a type hierarchy; **action** dominates **terrorist-attack** in the hierarchy.[3] Therefore, the system retrieves **gc-schema2**, as shown above, using (**goal-conflict**, **terrorist-attack**) as an index. Next, the system matches the **goal-conflict** in the advice to the consequent of the viewing schema and matches the **terrorist-attack** to its **lhs3** antecedent, then attempts to satisfy the schema's remaining antecedents.

BRAINSTORMER tries to match all of these antecedents to existing memory representations, but is prepared to abductively infer some of them if necessary. First, the system attempts to retrieve items from memory that satisfy the antecedents of the viewing schema. In this example, it retrieves its own goal to prevent terrorist attacks. Next, the system tries one-step backward chaining to satisfy the remaining antecedents. Limiting the system to one-step backward inference is somewhat arbitrary, but it has proved sufficient for our needs, while putting a hard limit on the adapter's computation. Finally, if some antecedents are still unsatisfied, then the system uses the **abductively-infer** slot of the viewing schema to determine if it is reasonable to abductively hypothesize them: abductive inference can proceed if the remaining antecedents are a subset of the filler of this slot. (See figure 5.)

If all antecedents of the viewing schema are now satisfied, the resulting variable bindings are then used to instantiate the schema's consequent, which is already bound to a component of the advice (here, the **goal-conflict**). Finally, a **part-whole-view** frame is built from this instantiated schema, as shown earlier in figure 4, and redescription inference is complete.

Matching antecedents of a viewing schema can recursively trigger further redescription inference. In the current example, the **lhs3** recognition condition of the viewing schema requires that the actor of the **action** be of type **agent**. The actor of the **terrorist-attack**, however, is a *group* of terrorists, and groups are not agents. BRAINSTORMER is, however, capable of redescribing a group of agents as a composite agent by retrieving and applying an appropriate viewing schema.

The possibility of recursive redescription inference means that BRAINSTORMER's matching algorithm is actually a kind of *means-ends analysis* process [Fikes and Nilsson, 1971], where the differences to be resolved are incommensurate type labels on concepts being matched, and the operators for resolving differences are viewing schemas. BRAINSTORMER can thus engage in a nontrivial amount of search in attempting to match two incommensurate items. It is important to note, however, that this means-ends analysis approach is considerably more efficient than any obvious forward or backward chaining alternative, because search is much more tightly focused. Inference is only attempted if an important recognition condition of a concept is already known, and the system already wants to relate this condition to the concept. Figure 6 sketches BRAINSTORMER's algorithm for matching advice to queries.

(MATCH *adv qry*):
  IF the type labels of *adv* and *qry* match
  THEN recursively MATCH corresponding
       slot fillers of *adv* and *qry*
  ELSE (REDESCRIBE *adv qry*).

(REDESCRIBE *adv qry*):
  1. Retrieve all viewing schemas indexed under (*adv'*, *qry'*), where *adv'* and *qry'* are generalizations of *adv* and *qry* in the type hierarchy
  2. Until successful, or no more viewing schemas, pick a viewing schema and
     a. MATCH *adv* to the schema's consequent; MATCH *qry* to the appropriate antecedent.
     b. Attempt to satisfy the other antecedents:
        i. Try memory retrieval.
        ii. Try one-step backward chaining.
        iii. Try abductive inference.
     c. IF all antecedents are satisfied
        THEN schema application is successful. Construct an appropriate view frame as the binding associated with *adv*.

Figure 6: Matching and the redescription process.

Discussion

BRAINSTORMER's task is adapting abstract cases to fit planning specific situations; we have seen that this requires being able to dynamically categorize these situations as instances of abstract concepts. Existing representation systems are generally inadequate to this task. Like many frame-based representation systems, BRAINSTORMER starts with a fixed set of base concepts arranged in a taxonomic hierarchy that supports property inheritance. The system's representations of planning situations and advice are initially encoded as instances of these concepts. In contrast to many other systems, however, the redescription process allows BRAINSTORMER to flexibly and dynamically redescribe instances in terms of other concepts, thereby categorizing them in new ways. BRAINSTORMER can categorize instances both in terms of other base concepts and in terms of new concepts that it dynamically constructs from recognition conditions for other concepts.

BRAINSTORMER's redescription mechanism relates to three areas of past work in cognitive science and ar-

[3] Two type labels are commensurate if and only if they can be related in the system's type hierarchy.

tificial intelligence. First are investigations into the idea of viewing or redescription inference, starting with MERLIN [Moore and Newell, 1973] and KRL [Bobrow and Winograd, 1977] in the 1970's. More recently, Jacobs has resurrected these ideas in the context of natural language generation [Jacobs, 1987]. His ACE system redescribes concepts with no associated generation method in terms of other concepts that have one. ACE cannot create new concepts on the fly, but it would be pointless to do so in any case, as the newly created concepts would lack generation methods. Redescription in BRAINSTORMER, in contrast, is driven by the need to relate abstract advice to concrete situations. We have seen that BRAINSTORMER needs to introduce new concepts to adequately express the intended meaning of abstract advice as it applies to a specific situation.

Our research also relates to past work on analogical reasoning. For example, recategorizing a terrorist attack in terms of recognition conditions for a goal conflict can be thought of as answering the question "in what way is a terrorist attack like a goal conflict?" Redescription maps the terrorist attack into a target concept that is simultaneously (1) an instance of a covering class of intentional actions that violate goals of others and (2) a set of conditions sufficient for inferring a goal conflict. This is a kind of analogical reasoning. Unlike many existing systems, however (e.g., Gentner's structure mapping engine [Falkenhainer *et al.*, 1986]), BRAINSTORMER does not depend on preexisting structural or predicate correspondences between base and target domains. All that is required is that the source can be extended to a set of plausible recognition conditions for the target.

Finally, our work extends earlier research on case adaptation in the field of case-based reasoning. In particular, Kass describes his research on ABE [Kass, 1989] as extending script/frame theory to handle a wider range of input situations; similarly, we are interested in flexibly relating past cases to new situations. BRAINSTORMER's cases, however, are much more abstract than ABE's. Consequently, a principal concern of BRAINSTORMER is making abstract descriptions more specific; in contrast, ABE spends most of its time trying to replace components of past explanations that do not apply in a new situation with different components that do. Therefore, while dynamic redescription is of central importance in BRAINSTORMER, it does not exist in ABE, and if it did, it would probably play a peripheral role.

Conclusion

In this paper, we have focused on the problem of adapting abstract knowledge about planning to fit specific planning situations. We have shown this to entail redescribing aspects of particular planning situations in terms of a dynamically-constructed abstract concepts. We have presented a memory organization and a redescription process that efficiently accomplishes this task. Our redescription process is implemented as part of the BRAINSTORMER system, a planner that takes abstract advice in the context of ongoing problem solving.

Acknowledgements. Thanks to Eric Domeshek, Andy Fano, Bruce Krulwich, and David Pautler for comments on drafts of this paper.

References

[Bobrow and Winograd, 1977] D. G. Bobrow and T. Winograd. An overview of KRL, a knowledge representation language. *Cognitive Science*, 1(1):3–46.

[Cullingford, 1977] R. Cullingford. *Script Application: Computer Understanding of Newspaper Stories.* PhD diss., Yale Univ.

[Falkenhainer *et al.*, 1986] B. Falkenhainer, D. K. Forbus, and D. Gentner. The structure-mapping engine. In *Proceedings AAAI-86*, 272–277, Philadelphia, PA. AAAI.

[Feigenbaum *et al.*, 1971] E. A. Feigenbaum, B. Buchanan, and J. Lederberg. On generality and problem solving: a case study using the DENDRAL program. In B. Meltzer and D. Michie, editors, *Machine Intelligence 6*, 165–190. Edinburgh University Press, Edinburgh.

[Fikes and Nilsson, 1971] R. E. Fikes and N. J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208.

[Hammond, 1986] K. J. Hammond. *Case-based Planning: An Integrated Theory of Planning, Learning and Memory.* PhD diss., Yale Univ.

[Jacobs, 1987] P. S. Jacobs. Knowledge-intensive natural language generation. *Artificial Intelligence*, 33:325–378.

[Jones, 1990] E. K. Jones. BRAINSTORMER: A model of advice taking. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, 269–276, Cambridge, MA. Cognitive Science Society.

[Jones, 1991] E. K. Jones. Knowledge refinement using a high-level, non-technical vocabulary. In *Machine Learning: Proceedings of the Eighth International Workshop*, Chicago, IL. Morgan Kaufmann.

[Kass, 1989] A. Kass. Adaptation-based explanation: Extending script/frame theory to handle novel input. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 143–147, Detroit, MI. IJCAI.

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

[Moore and Newell, 1973] J. Moore and A. Newell. How can MERLIN understand? In L. W. Gregg, editor, *Knowledge and Cognition*. Lawrence Erlbaum, NJ.

[Owens, 1988] C. Owens. Domain-independent prototype cases for planning. In *Proceedings of a Workshop on Case-Based Reasoning*, 302–311, Clearwater, FL. Defense Advanced Research Projects Agency, Morgan Kaufmann.

[Rieger, 1976] C. Rieger. An organization of knowledge for problem solving and language comprehension. *Artificial Intelligence*, 7:89–127.

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

[Sowa, 1984] J. F. Sowa. *Conceptual Structures: Information Processing in Mind and Machines.* Addison-Wesley, Reading, MA.

[White, 1987] G. M. White. Proverbs and cultural models. In Dorothy Holland and N. Quinn, editors, *Cultural Models in Language and Thought*, 151–172. Cambridge University Press, New York.

[Wilensky, 1978] R. Wilensky. *Understanding Goal-Based Stories.* PhD diss., Yale Univ.