An Indexing Vocabulary for Case-Based Explanation*

David B. Leake
Department of Computer Science
Indiana University
Bloomington, IN 47405
leake@cs.indiana.edu

Abstract

The success of case-based reasoning depends on effective retrieval of relevant prior cases. If retrieval is expensive, or if the cases retrieved are inappropriate, retrieval and adaptation costs will nullify many of the advantages of reasoning from prior experience. We propose an indexing vocabulary to facilitate retrieval of explanations in a case-based explanation system. The explanations we consider are explanations of anomalies (conflicts between new situations and prior expectations or beliefs). Our vocabulary groups anomalies according to the type of information used to generate the expectations or beliefs that failed, and according to *how* the expectations failed. We argue that by using this vocabulary to characterize anomalies, and retrieving explanations that were built to account for similarly-characterized past anomalies, a case-based explanation system can restrict retrieval to explanations likely to be relevant. In addition, the vocabulary can be used to organize general explanation strategies that suggest paths for explanation in novel situations.

Introduction

The fundamental strategy of case-based reasoning (CBR) systems is to address new situations by re-using the applicable portions of prior experiences. If similar situations have been processed in the past, this approach can significantly facilitate processing. However, its success depends on the ability of the retrieval process to efficiently select cases likely to apply. If a system cannot find the most relevant cases in its memory, adaptation of retrieved cases is unnecessarily costly, and efficiency advantages of case-based reasoning are reduced. Focusing retrieval is a well-known problem; [Hammond, 1989] provides an overview of retrieval issues and some current approaches.

*This work was conducted at Yale University, supported in part by the Defense Advanced Research Projects Agency, monitored by the Office of Naval Research under contract N0014-85-K-0108 and by the Air Force Office of Scientific Research under contract F49620-88-C-0058.

When understanding systems encounter anomalous events, they need to explain them, in order to have an accurate picture of what has happened and what is likely to occur in the future. A promising way of generating the needed explanations is by case-based reasoning, adapting explanations of similar prior events to fit the new situation [Schank, 1986]. To aid effective explanation retrieval, we have developed a vocabulary for characterizing anomalies. The vocabulary items organize case memory: explanations in memory are indexed under the vocabulary items for the anomalies they explain. When an anomaly is encountered, a description is generated in the anomaly vocabulary and used to index into explanations with similar characterizations.

Our vocabulary includes nine top-level categories, which account for a wide range of everyday anomalies. We anticipate that classifying all anomalies in everyday events would involve roughly 15 top-level categories. We have also identified important sub-classes under many of our main classes, and defined categories corresponding to them. In a case-based explanation system with a large library of explanations, the sub-classes could be further specified, with different sub-categories providing finer-grained guidance towards specific explanations in memory.

Our theory is implemented in ACCEPTER, a story understanding program that detects anomalous events in the stories it processes, and characterizes the anomalies to facilitate explanation retrieval and adaptation [Leake, 1990].[1] ACCEPTER has been used to characterize anomalies in a range of anomalous events, such as the explosion of Space Shuttle Challenger, the accidental shootdown of an Iranian airliner by the warship Vincennes, and unexpected celebrity deaths.

We begin with a brief discussion of the focus needed to explain anomalies. We then describe our theory of anomaly characterization, sketch the top-level categories, and show how the categories can be associated with knowledge structures whose components reflect the important features of anomalous situations. We then discuss the generality of our anomaly categories.

[1] Early versions of the program were embedded in the case-based explanation system SWALE [Kass *et al.*, 1986].

The problem: focusing explanation

When the world conforms to an understander's expectations, the understander has no reason to explain. However, when expectations fail, the failure reveals a gap in its knowledge, requiring explanation to fill that gap [Schank, 1982]. An argument for case-based reasoning is that a CBR system will be able to improve performance with experience, as its case library grows and it becomes more likely to have applicable cases. However, larger case libraries make indexing crucial. Since any retrieved case may require expensive adaptation to fit the current situation, the system needs a way of guiding retrieval towards the right cases. Even if cases that fit the current situation are found, success is not guaranteed: not all applicable cases are appropriate.

For example, suppose we are given the statement "John used a blowtorch to break into the First National Bank's automatic teller yesterday night." If we ask "why?," many explanations that might apply, such as:

- John's uncle wouldn't lend him any more money.
- Crowbars aren't enough to open modern ATMs.
- The bank's security camera was broken.
- The torch worked because it was a new model that melts anything instantly.

In a specific instance, only a few of the applicable explanations will be appropriate (for example, John's parents will be more interested in his motivations than his burglary techniques.) Especially when the explainer has the capability to adapt near-miss explanations, the range of potential candidates is overwhelming. To make the search for explanations tractable, we must have a way to direct search towards the right explanation.

Anomaly-centered retrieval

In most systems that explain unexpected events, explanation is directed towards accounting for the event, basically without considering the prior context (*e.g.*, [Mooney and DeJong, 1985]). However, many explanations can be generated for a given situation, and not all of them provide the needed information. The aspects of a new situation that merit explanation are those relevant to the the specific expectation failure, since they show which parts of the situation the system misjudged or overlooked.

For example, a burglar might be surprised that the torch *could* penetrate the ATM, and need the explanation that focuses on its capabilities. On the other hand, if the burglar was a friend of John's, and knew that John's *modus operandi* was to use a crowbar, the choice of the torch would need explanation. Although the same event is explained in both cases, different aspects are surprising, and require different explanations. To satisfy an understander's particular needs for explanation, a case-based explanation system must retrieve cases that not only involve similar *events*, but involve similar *anomalies* [Leake, 1988]. Accounting for anomalies requires showing why the reasoning leading to the contradicted expectation or belief failed to apply: in our examples, that a new type of torch had been developed, or that John had made an unsuccessful attempt with the crowbar the night before.

To focus retrieval towards explanations that address the failed beliefs, indices for explanations must reflect the anomalies to explain. We discuss below how our characterization reflects both events and anomalies.

A vocabulary for anomalies

The purpose of a vocabulary for anomalies is to facilitate explanation: the value of the categories is the explanatory information they provide. The basic explanation retrieval process in our system involves three steps: (1) an anomaly is detected (the detection process is beyond the scope of this paper; see [Leake, 1990]), (2) the raw anomaly description is input to an anomaly characterizer, which outputs an anomaly characterization in the anomaly vocabulary, (3) the characterization is input to an explanation retriever, which uses it to index into a library of explanations, and outputs explanations with similar descriptions. This process is summarized in figure 1.

For the retrieval scheme to be successful, anomalies must be characterized in similar ways exactly when they have similar explanations. If similar characterizations reliably suggest that similar explanations apply, the scheme will be able to suggest near-miss explanations when no perfect explanation is available in memory, and will not suggest irrelevant candidates.

ACCEPTER's anomaly classes reflect two main aspects of similarity of anomalous situations. The first is the event itself; similarly-characterized anomalies must concern similar events. If we are explaining a robbery, we will want to retrieve explanations of other robberies.

The second is the type of knowledge underlying the expectation, and how it failed— similar characterizations must reflect similar failures. Returning to the example "John used a blowtorch to break into the First National Bank's automatic teller yesterday night," which explanation applies depends on the particular anomaly. By characterizing the type of knowledge that provided the failed expectations, and how the expectations failed, we obtain an anomaly description that corresponds to a class of explanations, and restricts search. We list below some knowledge areas that could be relevant to expectation failures prompted by the example, and show a relevant explanation for each one.

- Planning choice: We might have expected John to get money by borrowing from a relative, requiring an explanation to account for why he tried robbery instead. The category for such problems is SURPRISING-PLAN-CHOICE.

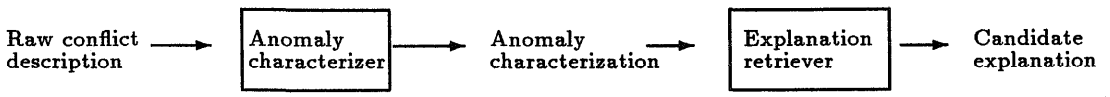

Figure 1: Processing sequence from anomaly to explanation.

- Plan instantiation: We might have expected John to use a crowbar instead of a torch. The category for anomalies involving use of unusual tools or other objects is SURPRISING-PROP-CHOICE.
- Plan and action execution: We might have known of John's robbery plan in advance, and known that he planned to rob another bank, or to do the robbery at another time. The category for explanations of deviations in a plan's details, once it has begun, is PLAN-EXECUTION-FAILURE.

  If we previously expected the plan to be blocked (*e.g.*, by new security measures at the bank), and it succeeded anyway, the explanation would also have to account for the BLOCKAGE-VIOLATION.
- Information transmission: We might have previously read that the robbery took place at a different bank, making the anomaly our BAD-INFORMATION.
- Models of physical, biological or chemical processes: We might not have expected the blowtorch to be able to melt the ATM door quickly enough. The category for deviations from models of progressive processes, such as unusually rapid melting, is PROCESS-EXECUTION-FAILURE.
- Models of device function: We might have expected the ATM closed-circuit TV to alert the bank before the break-in could take place. The failure to do so is an instance of DEVICE-FAILURE.
- Inductive generalizations about object features: We might think that all ATMs are run by other banks, in which case it is anomalous that the owner of the ATM is First National Bank. The category for such anomalies is UNUSUAL-OBJECT-FEATURE.
- Generalizations about persistence of features [McDermott, 1982]: Successful theft from an ATM would be surprising if we believed that the machine contained no money, and would not be refilled until the next day. This anomaly is STRANGE-FEATURE-CHANGE.

When ACCEPTER detects a conflict with expectations or beliefs, the memory structure for the expectation or belief, and the conflicting information, are passed to its anomaly characterizer. To decide on the proper anomaly class, ACCEPTER first determines the source of the prior belief. For its explicit expectations, such as those generated during schema application, ACCEPTER maintains a pointer to the expectation source in its representation of the expectation, and can simply follow that pointer. For implicit expectations, such as generalizations about standard object features, the memory search process that detects the conflict also notes the source of the expectation, and passes it to the anomaly characterizer along with the conflict. Once ACCEPTER has identified the expectation source, it selects the anomaly category associated with that type of source and conflict. This category determines the basic type of explanation to consider.

Requiring explanations to address a particular expectation source is a strong constraint on the decision of which explanations apply. For example, the four explanations we first considered for John's robbery each address different knowledge sources; we can choose between then based on the source of the knowledge that failed. This is an important component of ACCEPTER's decision of which explanations to apply. For example, when an anomaly is described in terms of SURPRISING-PLAN-CHOICE, only explanations with that categorization are considered for retrieval.

In addition, each category is associated with a structure for describing anomalies of that category. Once the category is chosen, the characterizer uses category-specific rules to fill the slots of the characterization structure, and specify the characterization if possible. We describe below the structures, their specifications, and how they are used.

The structure of anomaly descriptions

Retrieving the best explanation requires selecting the important features of a conflict. Unfortunately, without knowing the explanation, it is impossible to know which features *are* important. One of the functions of anomaly categories is to guide selection of important features, before an explanation is available. Each anomaly category is associated with a knowledge structure, with slots for information likely to be important.

For example, take SURPRISING-PLAN-CHOICE. Since that anomaly type characterizes conflicts with predictions about plans, its structure must include the information that affects expectations for an actor's planning process: the actor, the goal for which the actor is planning, the prior expectation for the actor's plan, and the surprising plan. Table 1 shows an example of the characterization for the anomaly of an actor driving to New York, when he was expected to take the bus. The components of this structure direct retrieval toward unusual plans that the actor (or similar actors) selected in similar circumstances.

SURPRISING-PLAN-CHOICE

Slot	Filler
Assumed goal	Going to New York
Actor	John
Plan	Car travel
Conflict description	Car travel instead of bus

Table 1: Components of the anomaly characterization for a SURPRISING-PLAN-CHOICE anomaly.

Finer-grained descriptions

The categories above describe only *what* knowledge fails, rather than *how* it fails. Since the description includes specific information from the anomaly characterization structure, it may be sufficient to retrieve explanations addressing the same anomaly. For example, if John travels by car instead of bus a second time, the first explanation to consider is the one that accounted for the previous failure. However, when specific features do not match past experience, an intermediate level of characterization is needed: one that reflects the underlying similarity of superficially different situations, but retains enough specificity to still suggest reasonably specific types of explanations.

For example, although very different plans are involved in getting a graduate degree and getting lunch, cancellation of those plans may share common types of reasons, such as lack of resources, or competing obligations. Delays in their completion may also share similar types of reasons, such as the need to complete preconditions that are usually satisfied (respective examples include taking background courses, and going to a bank to get cash before going to the restaurant).

These similarities are in *how* plan execution differed from expectations: in the first case, the plan was cancelled; in the second, delayed. Plans being cancelled, or delayed, respectively correspond to two sub-categories of plan execution failure, that can be used to describe anomalies more specifically than the simple PLAN-EXECUTION-FAILURE description, suggesting more precisely-applicable explanations.

These types of failures correspond to two subcategories PLAN-EXECUTION-FAILURE: PLAN-CANCELLATION, and PLAN-DELAY. Describing an anomaly in terms of these categories gives more precise guidance towards explanations likely to be relevant. Figure 2 sketches how sub-categories are organized in memory under the top-level categories, with specific explanations stored under them.

We illustrate the major anomaly categories and their specifications in table 2. Each category and subcategory has an associated knowledge structure, with fixed slots to be filled in to form a characterization. (For a description of all these anomaly categories, see [Leake, 1990].) We consider these the top levels of an abstraction hierarchy, that could be specified further

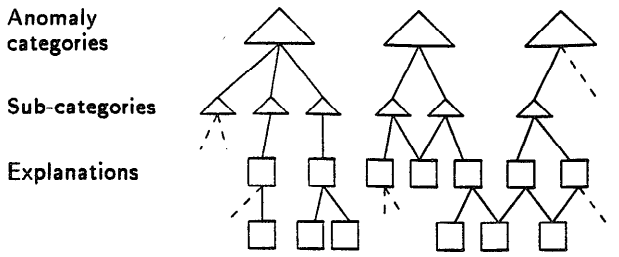

Figure 2: ACCEPTER organizes explanations in an abstraction net under its anomaly categories and subcategories.

to provide a discrimination tree to direct search in a large library of explanations.

- **SURPRISING-PLAN-CHOICE**
  IRRELEVANT-PLAN
  REDUNDANT-PLAN
  PREFERENCE-FAILURE
  BLOCKED-PLAN
- **SURPRISING-PROP-CHOICE**
- **PLAN-EXECUTION-FAILURE**
  PLAN-DELAY
  PLAN-SPEEDUP
  PLAN-CANCELLATION
  PLAN-OUTCOME-FAILURE
- **BLOCKAGE-VIOLATION**
  INADEQUATE-ROLE-FILLER
  UNAVAILABLE-ROLE-FILLER
- **PROCESS-EXEC-FAILURE**
  PROCESS-DELAY
  PROCESS-SPEEDUP
  PROCESS-OUTCOME-FAILURE
- **DEVICE-FAILURE**
- **BAD-INFORMATION**
- **UNUSUAL-OBJECT-FEATURE**
- **STRANGE-FEATURE-CHANGE**

Table 2: Main anomaly categories, with selected subcategories.

Using the vocabulary

In ACCEPTER's retrieval process, the primary index for explanation retrieval is the anomaly category. The system will only consider retrieving explanations that address the same category of anomaly. If possible, the system will retrieve explanations for the same subcategory as well.

Similarity of explanations within a category is judged by the similarity of individual slot-fillers of their

anomaly characterization structures. The stored explanations with the most specifically matching characterizations are retrieved first. Specificity of match is measured by closeness of the slot-fillers' types in the type hierarchy of the system's memory net. Since any combination of slots of a particular anomaly characterization could be abstracted, anomaly characterizations of specific explanations can be considered to be organized in an implicit net, with the possibility of a single characterization having multiple abstractions. The abstraction net is searched breadth-first.

As an example of how the categories constrain ACCEPTER's search process, consider again the characterization in table 1, which describes the anomaly "John was expected to take the bus to New York, but drove instead." ACCEPTER first tries to retrieve an explanation with exactly the same anomaly characterization, which would give explanations for other instances of John choosing driving over the bus to New York (*e.g.*, he might drive whenever he plans to work late). If no explanation is found, ACCEPTER tries to find explanations under the same anomaly type, but with generalizations of the fillers in the current instance. For example, if John were a commuter, one of the indices ACCEPTER would try would be the anomaly description for "A commuter was expected to take the bus to New York, but drove instead;" A further abstraction would be the description "A commuter was expected to take public transportation somewhere, but drove instead," which might index the explanation "commuters often drive on cold days, to avoid waiting outside for public transportation."

Even if no specific explanations were found for the SURPRISING-PLAN-CHOICE anomaly, simply knowing the anomaly type could suggest very general characterizations of what might cause surprising plan choices, such as "the usual plan is blocked" or "another agent is dictating the plan to use." These characterizations suggest general explanation strategies [Hammond, 1987], which guide search by suggesting features of the situation to investigate. For example, to see if the first explanation is applicable, an explainer might search for standard explanations for how the plan of taking a bus might be blocked. If buses are often full, that explanation could be retrieved and used to account for John's travel by car. However, ACCEPTER's memory contains only specific explanations; the use of these general strategies has not been implemented in the system.

Although retrieved explanations will account for anomalies similar to the current situation, they are not guaranteed to be appropriate. For example, an explanation may be relevant to the anomaly, but too vague to be useful— someone trying to increase the reliability of a bank's security system may need to know not just that the TV camera was broken, but how a burglar could know. Consequently, the explanations' appropriateness must be checked further before a retrieved explanation is applied. (See [Leake, 1990] for details of this process.) Likewise, an explainer with special expertise would be expected to have additional specifications of ACCEPTER's categories, organized below them in memory. For example, a television repairman might know additional features, beyond those in ACCEPTER's standard structure for characterizing device failures, to use in discriminating explanations of television problems.

When no appropriate explanations are found, an explanation retriever must be able to search for near-miss explanations. In this case, the structure can suggest the significant features for a partial match: some of the slots can be forced to match, and others allowed to vary. For example, the structure in table 1 suggests that if no explanations can be found for why a particular actor chooses a particular surprising plan, we might look for why other actors choose the plan, or why the particular actor chooses other plans.

Evaluation of the vocabulary

Our anomaly categories guide retrieval of explanations for real-world anomalies, focusing search through an explanation library. "Evaluation" in standard terms cannot be applied to this task. In both case-based reasoning systems and people, what is retrieved, like what is considered anomalous, is idiosyncratic and experience-based; at this point, we cannot generate a representative set of real-world anomalies and stored explanations, on which to test system performance. However, on other grounds we can substantiate both the need for anomaly characterization, and the particular categories proposed.

As discussed above, building explanations from scratch is an intractable problem. The case-based approach proposes re-using stored explanations, but presents control problems of its own: we have shown that any particular situation admits countless possible explanations, few of which may be relevant to the anomaly. To make case-based explanation tractable, we must focus the search process. This requires a way to describe what is sought, and to organize memory around features of the goal description. Our anomaly types provide criteria for which explanations are potentially relevant— those addressing the current anomaly type— and consequently narrow the search space. Associating anomaly categories with particular characterization structures provides further guidance, since the slots in the structures identify features that are likely to be important in explanation search.

Having shown that *some* anomaly vocabulary is needed, we turn to our specific categories. They were developed to characterize examples of anomalies collected during initial phases of our research on case-based explanation.[2] To facilitate explanation, the categories must group anomalies likely to require similar

[2]The data was a set of approximately 180 anomalies and explanations, gathered informally at the Yale Artificial In-

explanations, and must guide discrimination between explanations for a particular anomaly type.

One way of judging success for both purposes is to see whether the anomaly types, which were based on many concrete examples, are useful for more than the set of specific instances for which they were devised—whether it is possible to describe general explanation strategies that can be applied to any anomaly of a given type. If each type corresponds to a set of abstract strategies, and those sets are disjoint, the types partition the set of explanations according to abstract causal properties relevant to their explanations. Such a partition suggests that the categories correspond to differences that are likely to apply to a wide class of problems, rather than coincidentally working on the specific anomalies and explanations tested. For each of our anomaly categories, we have identified a set of general explanation strategies; each sub-category of anomalies is also associated with additional, more specific strategies that apply in addition to the general ones. Any particular application of explanation strategies gives domain-specific information, but the strategies themselves are general enough to be applied across different domains. They suggest types of causes to look for, giving domain-independent guidance about how to look for domain-specific information. Two examples, discussed above for surprising plan choice, are looking for impediments to the standard plan, and looking for a controlling agent other than the actor ([Leake, 1990] describes the strategies for each anomaly type). The existence of these disjoint sets of strategies suggests that the categories in fact have the desired generality.

Conclusion

The benefits of case-based reasoning depend on being able to efficiently select an appropriate case. This in turn requires a way to identify which features in a situation are important, to formulate indices for retrieval, and to search memory for relevant cases. We have argued that for case-based explanation, the central index for explanation retrieval is the anomaly to be explained. We have formulated a vocabulary for describing anomalies in terms of the *knowledge underlying expectations that fail*, and *how* the expectations fail. This vocabulary provides an abstract summary of the problem involved. Each category is associated with a particular knowledge structure, so vocabulary elements suggest specific features of the situation that are likely to be important for retrieval.

The vocabulary, and associated knowledge structures, are used to organize explanations stored in ACCEPTER's memory. After generating the characterization of an anomaly, the system uses that characterization to index into the library of explanations, to facilitate retrieval of relevant candidates.

When an explainer has previously encountered a particular anomaly, the anomaly vocabulary used to describe it is relatively unimportant, as long as the previous episode was classified the same way: any characterization will be sufficient for retrieval. However, when the current situation differs from all those explained previously, the characterization will only be useful if near-miss characterizations suggest near-miss explanations, to give general guidance about how to explain the current case. In our vocabulary, the categories themselves suggest general strategies to guide explanation, when no similar specific cases can be found. This makes it possible to benefit from the characterization, even when no matching cases are available.

Acknowledgements

I would like to thank the AAAI referees for their helpful comments on a draft of this paper.

References

Hammond, K. 1987. Learning and reusing explanations. In *Proceedings of the Fourth International Workshop on Machine Learning*, Irvine, CA. Machine Learning. 141–147.

Hammond, K., editor 1989. *Proceedings of the Case-Based Reasoning Workshop.* Morgan Kaufmann, Inc., San Mateo.

Kass, A. and Leake, D. 1987. Types of explanations. Technical Report 523, Yale University Department of Computer Science.

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

Leake, D. 1988. Evaluating explanations. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, Minneapolis, MN. AAAI, Morgan Kaufmann Publishers, Inc. 251–255.

Leake, D. 1990. *Evaluating Explanations.* Ph.D. Dissertation, Yale University. Computer Science Department Technical Report 769.

McDermott, D.V. 1982. A temporal logic for reasoning about processes and plans. *Cognitive Science* 6:101–155.

Mooney, R. and DeJong, G. 1985. Learning schemata for natural language processing. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA. IJCAI. 681–687.

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

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

telligence laboratory, and other examples that arose during work on SWALE. See [Kass and Leake, 1987] for a list of many of those anomalies and explanations.