Proceedings of the AAAI-93 Workshop on Case-Based Reasoning, AAAI Press, Menlo Park, CA, pp. 57–63, 1993.

Learning Adaptation Strategies by Introspective Reasoning about Memory Search

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

Abstract

In case-based reasoning systems, the case adaptation process is traditionally controlled by static libraries of hand-coded adaptation rules. This paper proposes a method for learning adaptation knowledge in the form of *adaptation strategies* of the type developed and hand-coded by Kass [90] . Adaptation strategies differ from standard adaptation rules in that they encode general memory search procedures for finding the information needed during case adaptation; this paper focuses on the issues involved in learning memory search procedures to form the basis of new adaptation strategies. It proposes a method that starts with a small library of abstract adaptation rules and uses introspective reasoning about the system's memory organization to generate the memory search plans needed to apply those rules. The search plans are then packaged with the original abstract rules to form new adaptation strategies for future use. This process allows a CBR system not only to learn about its domain, by storing the results of case adaptation, but also to learn how to apply the cases in its memory more effectively.

Introduction

The flexibility of case-based reasoning systems depends on their retrieval and adaptation processes. To direct case adaptation, CBR systems have traditionally relied on built in adaptation knowledge. However, developing models of how adaptation strategies are learned is important for both theoretical and practical reasons. From a theoretical viewpoint, understanding how adaptation knowledge is learned is an important issue for case-based reasoning as a cognitive model. From a practical viewpoint, developing mechanisms for CBR systems to learn adaptation knowledge is important because hand-coding adaptation knowledge is a difficult practical problem that can impede development of CBR systems. The practical difficulty is evidenced both by recent proposals that developers of CBR applications should focus on building decision-aiding systems that function primarily as memories, with a human user performing case adaptation (e.g., [Kolodner, 1991]), and by the many current systems that do not perform case adaptation (e.g., [Blevis *et al.*, 1991; Domeshek and Kolodner, 1991; Hennessey and Hinkle, 1991; Simoudis and Miller, 1991; Slator and Riesbeck, 1991]). Although focusing on the retrieval aspects of CBR is probably now the best way to build CBR applications, it has the practical drawback of requiring user expertise to apply retrieved cases—the user must be a partner in the reasoning process. In some contexts, this may itself require considerable user experience and domain knowledge, limiting the usefulness of the CBR system to naive users.

This paper addresses the question of how to learn case adaptation knowledge within the framework of case-based explanation construction (e.g., [Schank, 1986; Kass *et al.*, 1986; Kass and Leake, 1988; Schank and Leake, 1989]). It proposes an idea for automatically generating *adaptation strategies* [Kass, 1990]. Adaptation strategies combine abstract adaptation rules with memory search procedures that guide the search for domain-specific information needed to apply the abstract rules. The aim of the adaptation strategy approach is to provide generality (rather than being tied to a particular domain, the strategies can be used in any domain to which the general search strategies apply) while retaining operationality (adaptation strategies allow efficient access to the domain-specific information needed in a particular context).

In the proposed learning method, the CBR system's adaptation component begins with general information about the system's memory organization and with a small library of abstract adaptation rules. In response to specific adaptation problems, memory search plans are generated for finding the information needed to apply the abstract rules. The memory search plans are then packaged with the original abstract adaptation rules to form new adaptation strategies for future use. Thus in this model, adaptation strategy learning is done by storing the results of introspective reasoning about the information needed to adapt a case and about ways to search memory for information to satisfy those needs.

We note that the knowledge contained in the new adaptation strategies is not necessarily new information, in the sense that the memory search plans in the adaptation strategies combine memory search rules that were already available to the system. The role of the new adaptation strategies is primarily to make useful memory search paths explicitly available, but in addition they may package the results of learning about the applicability of particular combinations of memory search strategies to different types of adaptation problems.

We begin by discussing the motivation for using adaptation strategies rather than other types of adaptation rules. We next sketch the proposed method for learning adaptation strategies. We then illustrate the proposed method with a concrete example of how it could permit a CBR system to learn a sample adaptation strategy hand-coded into existing case-based explanation systems. We close by highlighting the fundamental research problems that must be addressed, some directions for addressing them, and some additional issues for failure-driven refinement of existing adaptation strategies.

Adaptation Rules versus Adaptation Strategies

The operationality/generality tradeoff is a classic problem for defining useful rules: General rules are often hard to apply; specific rules may be easier to apply but lack generality. Case adaptation rules exhibit this tradeoff: Abstract case adaptation rules have good generality, with a small set characterizing a wide range of possible adaptations (e.g., [Carbonell, 1983; Koton, 1988; Hammond, 1989; Hinrichs, 1991], but they may be hard to apply without additional specific domain knowledge [Kass, 1990].

The problem with general case adaptation rules is shown by the adaptation rule *add a step to remove harmful side-effect* which has been used in case-based planning [Hammond, 1989]. Although the rule is widely applicable, at that level of abstraction it gives no guidance about *how* to find the right step to add in order to mitigate a given side-effect. For example, if the case-based planning system is attempting to build a plan for X-ray treatment, and the X-ray dose needed to destroy a tumor will result in an excessive radiation dose to healthy tissue, finding the right step to add may require considerable domain knowledge.

More specific versions of the rule can reduce the cost of applying the rule, but at the expense of its original generality. For example, a more specific version of *add a step to remove harmful side-effect* can be tailored to adaptation problems for X-ray treatment plans, resulting in rules such as *add the step "rotate radiation sources" to remove harmful side-effect "excess radiation."* [Berger and Hammond, 1991]. Such rules can be applied effectively, but hand building such rules in advance requires intimate knowledge of a domain. In addition, an enormous number of rules may be needed, especially in systems that reason about multiple tasks and domains.

Kass argues for addressing the operationality/generality tradeoff by replacing traditional adaptation rules with *adaptation strategies* [Kass, 1990]. Adaptation strategies operationalize abstract rules by packaging them with memory search information. (In what follows, we use the phrase *adaptation rules* to refer to traditional adaptation rules, and *adaptation strategies* to refer specifically to the combination of transformations and memory search information developed by Kass.)

Hand-coded adaptation strategies were used to control case adaptation in the case-based explanation systems SWALE (e.g., [Kass *et al.*, 1986; Kass and Leake, 1988; Schank and Leake, 1989]) and ABE [Kass, 1990]. To clarify the nature of adaptation strategies, and to introduce the adaptation example we use to illustrate our proposal for adaptation strategy learning, we describe a case adaptation example from those systems. The example arose when the systems attempted to explain the death of Swale, a 3-year-old superstar racehorse who died unexpectedly at the peak of his career.

One candidate explanation SWALE retrieves for Swale's death is the death of the runner Jim Fixx, who, like Swale, died when in peak physical condition. The explanation associated with Fixx's death is the following: Fixx was doing recreational jogging, which caused a high exertion level and overtaxed a hereditary heart defect, leading to a fatal heart attack. That explanation does not apply directly to Swale: Swale was not a recreational jogger. Consequently, the explanation must be adapted to fit the circumstances of Swale's death.

Because the problem is that the action "jogging" does not apply to Swale, SWALE's adaptation component attempts to find a more plausible cause of exertion by Swale that can be substituted for the hypothesized jogging. In a system using abstract adaptation rules, the adaptation rule to apply would be a rule along the lines of *substitute evidence* [Koton, 1988]. That rule can be used in any domain but gives no guidance as to how to find the evidence to substitute. Alternatively, in a system using very specific rules tailored to the domain of horse-racing, the rule for finding the substitution might be *if evidence for a racehorse's exertion is unconvincing, substitute racing for the previous evidence*—a rule that is easy to apply but that has no applicability to other types of actions and actors.

Instead of either the type of traditional adaptation rule, SWALE and ABE use an adaptation strategy called *Replace-action: Use agent theme links,* which suggests trying to find substitute actions by examining actions habitually associated with the actor. We can view this adaptation strategy as operationalizing the abstract rule *substitute evidence* by combining it with memory search information. In SWALE and ABE, the memory search information is contained in a hand coded procedure that searches memory for a replacement action that would be plausible for the agent of the implausible action to perform. The search is conducted by following links in memory between the agent and the agent's *role themes* [Schank and Abelson, 1977], to find actions stereotypically associated with the agent.[1]

The adaptation strategy *Replace-action: Use agent theme links* retains generality, because it can be applied whenever an explanation depends on an implausible action. In addition, in any particular situation that adaptation strategy provides fairly specific guidance for memory search: it provides a strategy for finding information appropriate to the specific actor involved. For example, when the rule is used to find a replacement action for Swale's jogging, it suggests searching for role themes associated with racehorses and then considering those themes' standard actions as possible replacements for the jogging. In SWALE's memory, one of the theme actions of racehorses is running in races, so the strategy's memory search finds horse racing as a candidate replacement action. Replacing jogging with horse racing as the cause of exertion leads to a plausible explanation for why Swale died: that the exertion of his racing overtaxed a hidden heart defect.

Thus adaptation strategies appear to be a promising method for guiding case adaptation. Unfortunately, however, developing such strategies is a difficult problem requiring intimate knowledge of the domain and the CBR system's memory organization. The following sections suggest a method for learning such strategies automatically by (1) using knowledge planning [Hunter, 1990] to generate memory search plans, and (2) storing those plans, associated with the abstract adaptation rules that they operationalize, as new adaptation strategies for future use.

Adaptation strategy learning

Because memory search procedures are the central part of adaptation strategies, developing a system that can learn adaptation strategies depends on building mechanisms for generating and learning the memory search procedures they contain. Given those mechanisms, an adaptation system can start with general memory search information and a small library of abstract adaptation rules (since a small set of abstract rules appears to cover a wide range of possible adaptations) and operationalize the rules into adaptation strategies by combining the rules with relevant memory search procedures.

In the proposed framework, adaptation strategy learning is need-driven, performed when no existing adaptation strategies are applicable to a particular task or when failures occur during search for information needed for adaptation. This paper concentrates on the method for generating new strategies when no previous strategies apply, but the final section discusses some of the issues involved in building mechanisms for failure-driven refinement of existing adaptation strategies.

The proposed process for generating new adaptation strategies involves four steps:

1. Input a case and a description of a problem to be solved by adaptation.
2. Attempt to retrieve relevant existing adaptation strategies. If success, done—no new strategy is needed. If failure, retrieve a relevant abstract adaptation rule.
3. Use a knowledge planning process to generate memory search plans for the specific information needed to apply the abstract rule.
4. Package the new search plan with the rule to form a new adaptation strategy. Store the new strategy for future use.

Figure 1 sketches the stages of this process and its relationship to failure-driven refinement of existing adaptation strategies.

An example

The following example illustrates how the previous steps could be applied to the problem of finding an action to substitute for the implausible hypothesis that Swale's death was caused by jogging. We assume that the system starts with only abstract adaptation rules and simple memory search strategies based on local knowledge of the system's memory organization. The result of its processing is both a solution for the current adaptation problem—the suggestion that horse racing is a good substitute action—and the learning of an adaptation strategy corresponding to SWALE's hand-coded strategy *Replace action: Use agent-theme links.*

1. **Input an explanation and a description of a problem requiring adaptation.**

   The explanation of Jim Fixx's death applied to Swale is input to the system, along with the problem description *agent-action mismatch* for the hypothesis that Swale was jogging.

2. **Attempt to retrieve existing adaptation strategies indexed as relevant to the problem description. If success, done. If failure, retrieve abstract adaptation rules applicable to the class of problem.**

   Initially the system would have only abstract adaptation rules, so initially the search for adaptation strategies would fail, forcing the system to start from an abstract adaptation rule.

   Abstract adaptation rules such as *substitute evidence, remove evidence,* and *add support* are all potentially applicable to repairing the hypothesis that Swale was jogging. Consequently, a method is needed for deciding which rule to attempt to apply. The case-based explanation system CASEY

[1] A role theme represents stereotyped knowledge about the plans and goals associated with actors in certain societal roles. For example, a racehorse runs in races; a policeman performs actions such as directing traffic and investigating crimes.

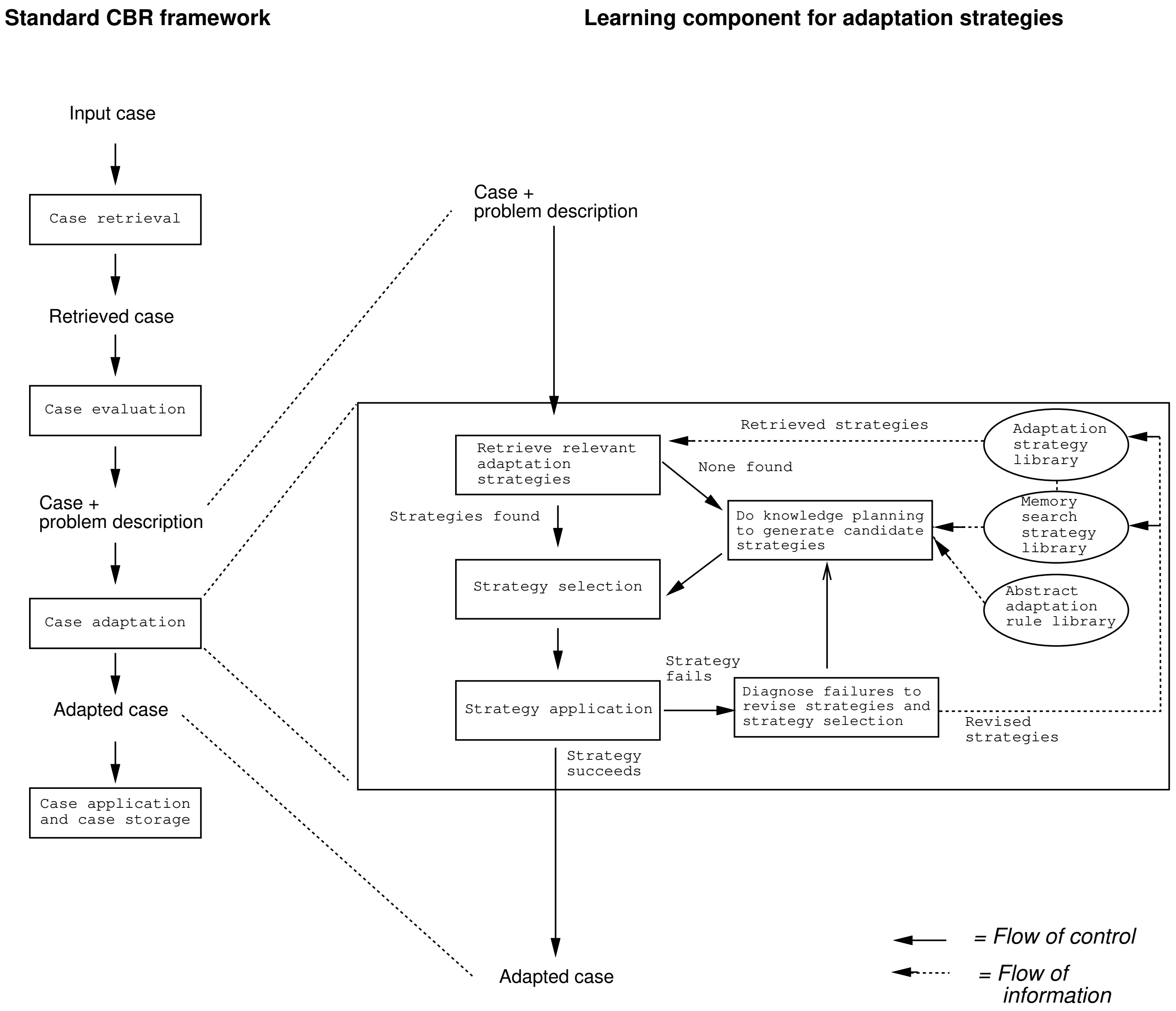

Figure 1: Adaptation strategy learning within the case-based reasoning process.

[Koton, 1988] includes some general criteria for deciding which abstract rules to apply, and those criteria form a starting point for abstract strategy selection criteria. For this example, however, *add support* does not apply—no supports exist for a racehorse jogging—and *remove evidence* immediately results in an explanation with insufficient support. This leaves *substitute evidence* as the only remaining candidate.

3. **Use a knowledge planning process to generate memory search plans.**

   The system attempts to generate a memory search plan to find evidence to substitute for "Swale jogging," the problem antecedent in the candidate explanation.

   (a) By analyzing the current problem description (*action-agent mismatch*), the system generates a knowledge goal [Hunter, 1990]: the goal to find a new causal antecedent that can be substituted to alleviate the current problem (i.e., to avoid the agent-action mismatch), and to provide support for the same part of the explanation that the implausible action initially supported. In the current example, the knowledge goal is to find a plausible antecedent for Swale having a high exertion level.

   (b) The knowledge goal is passed to the memory search planner. Because the adaptation component will begin with only abstract adaptation rules, it will initially need to generate a search plan from scratch.[2]

   For this example, suppose that the planner starts its processing with the memory search rule *to find a plausible antecedent for an actor's state, search for a plausible action by the actor that causes that state.*[3] Applying the rule requires finding an action that causes exertion and is plausible for Swale to have performed. To find an action that is plausible for Swale, other memory search rules must be applied. A relevant rule is *to find a plausible action by an actor, consider actions that the actor habitually does*, which generates the sub-goal of finding habitual actions. This can be done by applying the rule *to find habitual actions of an actor, search for actions associated with the actor's role themes.* This chain is operational—it involves directly-executable steps—so the search plan is complete.

4. **Package the search plan with the rule as a new adaptation strategy.**

   The abstract rule *substitute evidence* is combined with the memory search plan developed above. The result is equivalent to the adaptation strategy *Replace-action: Use agent theme links.*

Thus, results of the process are both a solution to a particular adaptation problem and a new adaptation strategy that can be applied to a wide range of future situations.

Directions towards adaptation strategy learning

Developing the proposed framework into a practical model of adaptation strategy learning depends on developing new theories of how to reason about information needs and how to guide memory search. In particular, the framework depends on extending current CBR models in five main ways:

1. *Developing a vocabulary for characterizing information needs for case adaptation.* Reasoning about how to search memory for needed adaptation information requires building a taxonomy of the types of information needs associated with particular adaptation tasks. (In the explanation context, these must reflect the constraints associated with abstract classes of modifications such as substituting, adding, or specifying hypotheses in an explanation, as well as constraints associated with specific types of problems that can arise in explanations with different types of content). The effort to formulate this taxonomy can build on current research on characterizing reasoning failures (e.g., [Leake, 1992; Kass, 1990; Birnbaum *et al.*, 1990; Ram *et al.*, 1992]).

2. *Building a model of planful memory search.* Previous research has developed methods for flexible memory traversal through an index elaboration process [Kolodner, 1984]. We propose going further in treating the search for adaptation knowledge as a planning process in which needs for information are represented as explicit goals to be achieved through plans based on reasoning about the system's information-seeking abilities [Hunter, 1990; Pryor and Collins, 1991]. The planful memory search framework requires the additional development of an internal model that the CBR system can use to reason about its own processing [Birnbaum *et al.*, 1991]. In particular, a model is needed for the system's own memory search process.

   Two types of knowledge are required for a suitable memory model: (1) abstract domain-independent knowledge about the structure of the memory (e.g., that abstractions of a concept can be obtained by following abstraction links), and (2) content-specific rules (e.g., that one way to find normative actions of an actor is by searching for actions associated with the actor's role themes). Luckily, the task of representing such information is facilitated because memory rules need only specify local relationships

---

[2] Once a library of strategies is available, case-based reasoning could be applied to building memory search plans.

[3] Many alternative memory search plans are possible here, and an important issue is developing search plans and criteria for selecting the plans to apply. Also, note that the required memory search knowledge includes both domain-independent and content-specific rules; we discuss this knowledge further in part 2 of the following section.

between concepts; more distant relationships can be derived as needed using the knowledge planning process. In addition, because the memory model itself can be task-independent, the resulting model should be applicable to a wide range of tasks and contexts.

3. *Developing methods for monitoring memory search.* This paper has only suggested the process for constructing new adaptation strategies from scratch. However, because it controls memory search with an explicit planning process that is accessible to system reasoning, the framework makes memory search strategies themselves accessible to failure-driven plan repair methods (e.g., [Sussman, 1975; Hammond, 1989; Birnbaum *et al.*, 1990]).

   Although existing models of failure-driven plan repair should be readily applicable to this task, one new issue that arises is how to detect the memory search problems themselves. The issue here is one of identifying and diagnosing failures of the system's own internal processing (e.g., [Birnbaum *et al.*, 1990; Birnbaum *et al.*, 1991]).

   To be able to recognize failures of adaptation strategies, the system must be able to detect memory search failures such as:

   - The search plan cannot be completed (e.g., due to failure to find intermediate information).
   - The search plan can be completed, but no specific information is available (e.g., if the search leads to a general class of objects for which no specific instances are known).
   - The search returns a result, but the result fails to fit additional constraints (e.g., the search finds a causal antecedent of a node in the explanation, but that antecedent fails to fit because of inconsistencies with the rest of the explanation).
   - The cost of the search differs from predictions. If criteria can be established for predicting the effort required to retrieve particular types of information, failures of those expectations can suggest that existing adaptation strategies must be modified or new strategies created.

4. *Determining the learning strategies appropriate for different types of memory search failures:* In order for an adaptation system to respond effectively to current failures, classes of memory search failures will need to be associated with strategies for recovering from and learning to avoid similar failures in the future (e.g., [Birnbaum *et al.*, 1991; Ram and Cox, 1993]).

5. *Indexing and retrieval of the adaptation strategies that are learned.* Learned adaptation strategies must be organized in memory to be retrieved at appropriate times. The ideas on indexing adaptation strategies described in [Kass, 1990] provide a starting point for addressing those memory organization issues and issues in adaptation strategy selection.

Perspective on other methods for learning adaptation knowledge

We note that some previous systems have the capability for learning knowledge useful for guiding adaptation. For example, although CHEF [Hammond, 1989] has a static library of domain-independent plan repair strategies, it augments that library with learned *ingredient critics* for suggesting adaptations appropriate to particular ingredients. Likewise, PERSUADER [Sycara, 1988] uses a combination of heuristics and case-based reasoning to guide adaptation, searching memory for similar prior adaptations to apply. In these systems, however, the adaptation information learned is quite domain and task specific—the cases learned must be applied to future problems in the same domain and with very specific similarities. The proposed method makes it possible to learn the more general adaptation information contained in adaptation strategies.

Conclusions

The flexibility of CBR systems to deal with new situations depends on their ability to do case adaptation. However, developing appropriate case adaptation strategies is both a theoretical and practical problem. This paper suggests a means to address that problem. In the proposed model, adaptation strategies with cross-domain applicability are built by (1) reasoning about the information needed to adapt an explanation (2) generating memory search plans for finding that information, and (3) packaging those plans as adaptation strategies for future use.

The goal of the approach is for a CBR system to start with only abstract adaptation rules and local knowledge about its memory organization, and then to build and refine adaptation strategies appropriate to its current task and the specifics of its knowledge and of the problems that it addresses. The memory search plans in new adaptation strategies may reflect specifications of the initial abstract memory search knowledge, showing how those abstract rules are combined to form effective search strategies for particular adaptation problems, even if those combinations of abstract rules may not be effective under all circumstances.

We note that with this approach, adaptation strategies produced are accessible to refinement by both success-driven and failure-driven learning processes. Thus not only can new adaptation strategies be generated, but they can be refined in response to experience.

Although the framework has been illustrated in the context of adaptation for case-based explanation construction, its planful memory search process has wider applicability. For example, abstract adaptation rules of the type that our framework requires as a starting point have already been developed for case-based planning [Hammond, 1989, pp. 135-136], and our method could be applied to those abstract rules to learn adaptation strategies for use in case-based planning.

In addition, this method for reasoning about memory search and learning memory search strategies has potential applicability far beyond the confines of case adaptation. The development of an introspective model of planful memory search would enable memory systems to recover from memory search failures and to improve their performance by generating, re-using and refining memory search strategies in response to their needs.

References

Berger, J. and Hammond, K. 1991. ROENTGEN: a memory-based approach to radiation therapy treatment. In Bareiss, R., editor 1991, *Proceedings of the Case-Based Reasoning Workshop*, San Mateo. DARPA, Morgan Kaufmann, Inc. 203–214.

Birnbaum, L.; Collins, G.; Freed, M.; and Krulwich, B. 1990. Model-based diagnosis of planning failures. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, MA. AAAI. 318–323.

Birnbaum, L.; Collins, G.; Brand, M.; Freed, M.; Krulwich, B.; and Pryor, L. 1991. A model-based approach to the construction of adaptive case-based planning systems. In Bareiss, R., editor 1991, *Proceedings of the Case-Based Reasoning Workshop*, San Mateo. DARPA, Morgan Kaufmann, Inc. 215–224.

Blevis, E.; Burke, R.; Glasgow, J.; and Duncan, N. 1991. The life analysis & depreciation integrated exemplar system (LADIES). *International Journal of Expert Systems* 4(2):141–155.

Carbonell, J. 1983. Learning by analogy: Formulating and generalizing plans from past experience. In Michalski, R.; Carbonell, J.; and Mitchell, T., editors 1983, *Machine Learning: An Artificial Intelligence Approach*. Tioga Publishing Company, Cambridge, MA.

Domeshek, E. and Kolodner, J. 1991. Toward a case-based aid for conceptual design. *International Journal of Expert Systems* 4(2):201–220.

Hammond, K. 1989. *Case-Based Planning: Viewing Planning as a Memory Task*. Academic Press, San Diego.

Hennessey, D. and Hinkle, D. 1991. Initial results from clavier: A case-based autoclave loading assistant. In Bareiss, R., editor 1991, *Proceedings of the Case-Based Reasoning Workshop*, San Mateo. DARPA, Morgan Kaufmann, Inc. 225–232.

Hinrichs, T. 1991. *Problem Solving in Open Worlds: A Case Study in Design*. Ph.D. Dissertation, Georgia Institute of Technology. College of Computing Technical Report GIT-CC-91/36.

Hunter, L. 1990. Planning to learn. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, Cambridge, MA. Cognitive Science Society. 261–268.

Kass, A. and Leake, D. 1988. Case-based reasoning applied to constructing explanations. In Kolodner, J., editor 1988, *Proceedings of the Case-Based Reasoning Workshop*, Palo Alto. DARPA, Morgan Kaufmann, Inc. 190–208.

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

Kass, A. 1990. *Developing Creative Hypotheses by Adapting Explanations*. Ph.D. Dissertation, Yale University. Northwestern University Institute for the Learning Sciences, Technical Report 6.

Kolodner, J. 1984. *Retrieval and Organizational Strategies in Conceptual Memory*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Kolodner, J. 1991. Improving human decision making through case-based decision making. *The AI Magazine* 12(2):52–68.

Koton, P. 1988. Reasoning about evidence in causal explanations. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, Minneapolis, MN. AAAI, Morgan Kaufmann Publishers, Inc. 256–261.

Leake, D. 1992. *Evaluating Explanations: A Content Theory*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Pryor, L. and Collins, G. 1991. Information gathering as a planning task. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, Chicago, IL. Cognitive Science Society. 862–866.

Ram, A. and Cox, M. 1993. Introspective reasoning using meta-explanations for multistrategy learning. In Michalski, R. and Tecuci, G., editors 1993, *Machine Learning: A Multistrategy Approach*. Morgan Kaufmann. In Press.

Ram, A.; Cox, M.; and Narayanan, S. 1992. An architecture for introspective learning. In *Machine Learning: Proceedings of the Ninth International Conference*, Aberdeen, Scotland. Workshop on Computational Architectures.

Schank, R.C. and Abelson, R. 1977. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Schank, R.C. and Leake, D. 1989. Creativity and learning in a case-based explainer. *Artificial Intelligence* 40(1-3):353–385. Also in Carbonell, J., editor, *Machine Learning: Paradigms and Methods*, MIT Press, Cambridge, MA, 1990.

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

Simoudis, E. and Miller, J. 1991. The application of CBR to help desk applications. In Bareiss, R., editor 1991, *Proceedings of the Case-Based Reasoning Workshop*, San Mateo. DARPA, Morgan Kaufmann, Inc. 25–36.

Slator, R. and Riesbeck, C. 1991. TaxOps: A case-based advisor. *International Journal of Expert Systems* 4(2):117–140.

Sussman, G. 1975. *A computer model of skill acquisition*, volume 1 of *Artificial Intelligence Series*. American Elsevier, New York.

Sycara, K. 1988. Using case-based reasoning for plan adaptation and repair. In Kolodner, J., editor 1988, *Proceedings of the Case-Based Reasoning Workshop*, Palo Alto. DARPA, Morgan Kaufmann, Inc. 425–434.