Proceedings of the Thirteenth National Conference on Artificial Intelligence, AAAI Press, Menlo Park, CA, 1996. In press.

Acquiring Case Adaptation Knowledge: A Hybrid Approach

David B. Leake, Andrew Kinley, and David Wilson
Computer Science Department
Lindley Hall 215, Indiana University
Bloomington, IN 47405

Abstract

The ability of case-based reasoning (CBR) systems to apply cases to novel situations depends on their case adaptation knowledge. However, endowing CBR systems with adequate adaptation knowledge has proven to be a very difficult task. This paper describes a hybrid method for performing case adaptation, using a combination of rule-based and case-based reasoning. It shows how this approach provides a framework for acquiring flexible adaptation knowledge from experiences with autonomous adaptation and suggests its potential as a basis for acquisition of adaptation knowledge from interactive user guidance. It also presents initial experimental results examining the benefits of the approach and comparing the relative contributions of case learning and adaptation learning to reasoning performance.

Introduction

Case adaptation plays a fundamental role in the flexibility of case-based reasoning systems. The ability of CBR systems to solve novel problems depends on retrieving relevant prior solutions and adapting them to fit new circumstances. Considerable domain knowledge may be needed to guide this adaptation process (e.g., Kolodner, 1993), and the need for such knowledge in turn raises the difficult question of that knowledge should be acquired and applied. Most CBR systems depend on a static library of built-in adaptation rules that are applied by rule-based production systems. Unfortunately, because CBR is often applied to domains that are poorly understood or difficult to codify, developing adaptation rules is particularly difficult. The problem is so acute that experts in CBR research and applications agree that it is not currently practical to deploy CBR applications with automatic adaptation. Consequently, new methods are needed for acquiring case adaptation knowledge.

We describe research on an approach for facilitating acquisition of useful adaptation knowledge. In our approach, a CBR system begins with a small set of abstract transformation rules and memory search methods. When presented with a new adaptation problem, it first selects a transformation to apply. It then performs memory search to find the information needed to operationalize the transformation rule and apply it to the problem at hand (e.g., given a *substitution* transformation, finding what to substitute). The system improves its adaptation capabilities by case-based reasoning applied to the case adaptation process itself: a trace of the steps in solving an adaptation problem is saved to be reused when similar adaptation problems arise in the future (Leake 1995). In this way, a CBR system doing adaptation can acquire specific adaptation knowledge by using "weak methods" for adaptation when no specific knowledge is available.

When autonomous adaptation is unsuccessful, this framework can also be used as a basis for interactive acquisition of adaptation cases from a human user. We are developing an interface that allows a user to guide transformation selection and memory search for a particular adaptation problem. A trace of the user's adaptation process is then added to the adaptation case library for future use.

We begin by sketching our testbed system's architecture. We next summarize the adaptation process and the knowledge sources it uses. We discuss preliminary results concerning the relationship of different types of memory search strategies, adaptation learning, and case learning, as well as the relationship of our method to previous approaches. We conclude by highlighting strengths of our approach and questions for further study.

DIAL System Overview

We are developing our model of adaptation learning in the context of a case-based planner in the domain of disaster response planning. Disaster response planning is the initial strategic planning used to determine how to assess damage, evacuate victims, etc., in response to natural and man-made disasters such as earthquakes and chemical spills. Accounts of human disaster response planning suggest that case-based reasoning is important in response planning by human disaster planners (Rosenthal, Charles, & Hart 1989).

Our testbed system, DIAL, processes conceptual representations of a news story describing the initial events in a disaster, and proposes a response plan by retrieving and adapting the response plan for a similar prior disaster. DIAL includes a schema-based story understander, a response plan retriever and instantiator, a simple evaluator for candidate response plans, and an adaptation component to adapt plans when problems are found. The system's case-based planning framework is based in a straightforward way on previous case-based planners such as CHEF (Hammond 1989). Consequently, we will not discuss DIAL's planning process per se, but instead will focus entirely on the system's case adaptation and adaptation learning.

Summary of DIAL's Adaptation Process

DIAL's adaptation component starts with a library of domain cases—disaster response plans from previous disasters—and general (domain-independent) rules about case adaptation and memory search. DIAL's case-based planner provides the adaptation component with two inputs: an instantiated disaster response plan and a description of the problems in the response plan that must be repaired. When the response plan has been successfully adapted, DIAL stores both the new response plan and two types of adaptation knowledge for use in similar future adaptation problems: *memory search cases* encapsulating information about the steps in the memory searches performed during adaptation, and *adaptation cases* encapsulating information about the adaptation problem as a whole, the transformations and memory search cases used when solving it, and the solution to the adaptation problem. Thus, the system learns not only new response plan cases but also new ways of adapting existing cases to new situations.

To adapt a case, DIAL's adaptation component performs the following steps:

1. **Case-based adaptation:** DIAL first attempts to retrieve an adaptation case describing the successful adaptation of a similar previous adaptation problem. If retrieval is successful, the adaptation process traced by that case is re-applied and processing continues with step 3.
2. **Rule-based adaptation:** When no relevant prior case is retrieved, DIAL selects a transformation associated with the type of problem that is being adapted. For example, it may decide to *substitute* a new plan step for one that does not apply. Given the transformation, the program generates a *knowledge goal* (Hunter 1990; Ram 1987) for the information needed to apply the transformation. (E.g., when performing a substitution, the knowledge goal is to find an object that satisfies all the case's constraints on the object being replaced.) The knowledge goal is then passed to a planning component that uses introspective reasoning about possible memory search strategies (Leake 1994) to guide search for the needed information. If the needed information is found, the transformation is applied. If it is not found, the process continues with step 4, manual adaptation.
3. **Plan evaluation:** The adapted response plan is evaluated by a simple evaluator that checks the compatibility of the current plan with explicit constraints from the response plan. A human user performs backup evaluation. If the new response plan is not acceptable, other adaptations are tried.
4. **Manual adaptation:** If autonomous case adaptation fails to generate an acceptable solution, an interface allows the user to guide the adaptation process, selecting a transformation and suggesting features to consider. During the adaptation, the system records a trace of the adaptation process in the same form as the traces of system-generated adaptations. This trace is added to the adaptation case library for future use.
5. **Storage:** When adaptation is successful, the resulting response plan, adaptation case, and memory search plan are stored for future use.

The basic principles of the adaptation process are shown by an example of developing a response plan for the story of a 1994 flood in Allakaket, Alaska. When DIAL processes that story, it retrieves the closest disaster case in memory, a flood in Bainbridge, Georgia. Part of the Bainbridge response was to build walls of sand bags to protect the area from water damage as the flood waters rose. In Bainbridge, volunteers helped to build the sand walls, and DIAL generates a knowledge goal to find people who could fill the same role in an Allakaket response plan. However, most of the able-bodied people in Allakaket are unavailable because they are helping to fight fires in the northwest. This prompts a new problem, that the desired role-fillers are unavailable. DIAL has no similar adaptation cases, so it falls back on rule-based memory search to attempt to find a substitution. It checks constraints on possible role-fillers and finds that the previous volunteers were under the authority of the police. Searching for others who are under the authority of the police, it finds prisoners as a possible substitution. Prisoners are suggested to build the flood walls, and, when they are judged a reasonable substitution by a human user, the replacement of able-bodied volunteers with prisoners is saved for future use.

Guiding Adaptation

In order to reason about adaptation problems, a uniform framework is needed for characterizing the case adaptation problem. DIAL's rule-based case adaptation treats the case adaptation process as involving two parts: applying *structural transformations* (e.g., additions, substitutions, and deletions) and performing *memory search* to find the information needed to apply the transformations (Kass 1990). Accordingly, two types of case adaptation knowledge are needed: abstract transformations and memory search strategies. A small set of transformations is sufficient to characterize a wide range of adaptations (Kolodner 1993), but much domain-specific reasoning may be required to find the information to apply those transformations. Consequently, a key issue is learning how to find the needed domain-specific knowledge. This involves a process of generating knowledge goals, using them to focus search for information, and packaging the reasoning trace to guide future adaptations.

Knowledge Goals: DIAL generates knowledge goals to obtain information necessary for a specific adaptation. For example, the national guard was called out to prevent riots after the Los Angeles earthquake. When using the response to that earthquake as the basis for the response to an earthquake in Liwa, Indonesia, a problem arises: Indonesia has no national guard. Consequently, the Los Angeles response plan must be adapted. In response to the inappropriate value problem, a knowledge goal is generated to find a substitute for the national guard in the Liwa earthquake context.

Memory Search Plans: DIAL's memory search process starts from an input knowledge goal and uses a combination of rule-based and case-based reasoning to guide the traversal of memory to find the needed information. DIAL's memory is frame-based, hierarchically organized by *memory organization packages* (MOPs) (Schank 1982). MOPs representing event sequences include roles such as *actor* and *object* (e.g., the MOP for a flood disaster includes a role for the rescuers), constituent sub-events, called *scenes* (e.g., rescuers traveling to the victims, rescuers evacuating the victims), and constraining relationships between the roles in a main MOP and roles in its scenes (e.g. the fillers of the *victims* role of the flood MOP are the *evacuees* in its evacuation scene). DIAL's MOPs may also include explicit relationships in which role-fillers of a MOP are involved (e.g., because the police responding to a particular flood are directed by the mayor, the MOP represents that they participate in an authority relationship with the mayor). All these of relationships may suggest pathways to be pursued during memory search. Corresponding memory search operations exist to examine roles, scenes, explicit relationships between role-fillers, or the meanings of those relationships; to examine MOPs or response plans that are nearby in the abstraction hierarchy; and to examine a representation of the problem prompting memory search.

Memory search cases: When memory search is successful, a trace of the search process is packaged as a memory search case. A memory search case consists of a sequence of primitive memory search operations which was previously used to find some information in memory. Initial memory search cases are built up interactively by recording traces of manual adaptation. When similar knowledge is next needed in a similar context, a retrieved search case provides an initial strategy for finding the needed information. Memory search cases are indexed under adaptation cases that have successfully used them and the knowledge goals they satisfy, so that they can be used as operators when building up future memory search plans.

Adaptation Cases: Adaptation cases package the results of a successful adaptation. They package both a transformation type (e.g., substitute, add, delete) and the memory search steps used to find the information needed to apply the transformation. An adaptation case consists of three parts: indexing information, adaptation information, and evaluation information. The indexing information includes a representation of the type of problem to adapt as the primary index, along with information about the response plan context in which the adaptation case was generated. Thus, appropriate adaptations can be retrieved to deal with new adaptation problems.

The type of problem to be repaired by adaptation is described in terms of a vocabulary similar in spirit to the problem vocabularies used to guide adaptation in other CBR systems (e.g., Hammond, 1989; Leake, 1992). For example, DIAL's problem types include the following problems involving role-fillers in a candidate plan: UNAVAILABLE-FILLER (e.g., a police commissioner may be out of town and unable to be reached in an emergency situation), FILLER-MISMATCH (e.g., if a workplace response plan involving notifying the victims' union is applied to a school disaster, whose victims—children—do not have unions), UNSPECIFIED-FILLER (e.g., if a plan calls for a rescue without specifying who will carry it out). These categories are used to index adaptation cases.

How Adaptation Knowledge is Acquired

DIAL acquires adaptation cases in two ways. First, it generates them autonomously, based on the adaptations it performs. Second, they can be entered into the system with a manual adapter, in which a human user interactively builds an adaptation case in response to a problem, selecting the type of transformation to apply and the types of memory links to follow. The manual adapter lets a user input cases which implicitly contain the user's knowledge of important features to consider when performing a particular type of adaptation. The specific adaptation and a trace of how it was derived are saved for future use. At present, the use of the manual adapter requires considerable knowledge of the system's memory organization, but a future research direction is to allow the user to provide general suggestions to be operationalized by the system's own memory search mechanisms.

Effects of Adaptation Learning

To obtain initial indications of the potential value of adaptation learning, we compared the system's adaptation efficiency under six different conditions. In the first four conditions, all memory search during case adaptation was done by "local search," with different combinations of learning methods: (1) No learning of either cases or adaptations, (2) Learning of response plan cases only (this is the standard learning of CBR systems), (3) No learning of response plan cases, but learning of adaptation cases, and (4) learning of both response plan cases and adaptation cases. Conditions (5) and (6) replaced "local search" with memory search planning to find needed information. In (5), DIAL performed response plan learning only, and in (6), it learned both response plans and adaptation cases.

The memory for the trials included nodes for 870 concepts. The initial case library included 6 response plans, for the following disasters: an earthquake in Los Angeles, an air quality disaster at a manufacturing plant, a flood in Bainbridge, Georgia, a chemical disaster at a factory, a flood in Izmir, Turkey, and an air quality disaster in a rural elementary school. The system processed conceptual representations of 5 stories taken from the Clarinet News Service newswire and the *INvironment* newsletter for air quality consultants: An indoor air quality disaster at Brookview School; A chemical disaster at Johnson School; An air quality disaster at the Kirtland military base; A flood at Allakaket, Alaska; and an earthquake in Liwa, Indonesia. Appropriate response plans for each of these can be generated by adapting one of the prestored plans. For example, one change to adapt the plan for the Bainbridge flood to apply to Allakaket is that the Salvation Army—which provided shelter during the Bainbridge flood, but does not exist in Allakaket—is replaced by the Red Cross.

Each of the input problems required multiple adaptations. In the trials including adaptation learning, the system built 30 adaptation cases. Efficiency of adaptation was compared across the six conditions by counting both the number of primitive memory operations performed and the number of memory nodes visited; each gives an indication of memory search effort. Table 1 shows the results for a single problem order, but changes in problem order did not appear to have a significant effect.

Multiple search strategies combined with adaptation learning performed best overall in terms of memory operations performed. This contrasts with the relatively poor behavior of using multiple search strategies without adaptation learning, which will be discussed below. The same general pattern follows in results based on the number of memory nodes visited.

The poor performance of multiple strategies without learning, compared to local search without learning, was initially surprising. However, it can be explained by the types of adaptations that are most common to these problems and the contents of memory: near-by substitutions, such as sibling nodes, were often appropriate fillers, and these fillers could be found directly by local search.

Conversely, a problem that is difficult for local search is often easier for the other strategies, which perform operations such as attempting to find substitute fillers based on explicit constraints (e.g., as in the case of having to notify children's parents instead of their unions). Another illustration involves adapting the Los Angeles earthquake response plan into a plan for Liwa, Peru. The Los Angeles plan involved the Red Cross sending supplies in by truck, but the roads to Liwa are impassable so the Red Cross cannot deliver the supplies. (In the real episode, the solution was a military airlift.) Using local search to replace the Red Cross with an agency that can deliver the supplies is costly, because the Red Cross and the military are distant in the system's memory. A search based on other strategies identifies an old case where "lack of access" was an impediment, uses this case to identify vehicles that can make the trip, and then looks for actors who control these vehicles. This leads to the suggestion of the Liwa military after minimal search. Such problems did not arise often, however. When adaptation cases based on both local search and other strategies are saved and reused, average performance is better than for either method individually.

	Memory Ops			Nodes visited		
	Avg	Max	Min	Avg	Max	Min
Using "local search" to find needed information						
1. *No learning*	103	226	7	53	114	4
2. *Plan learning only*	80	214	7	41	108	4
3. *Adaptation learning only*	68	181	4	40	92	3
4. *Plan + Adaptation Learning*	66	181	4	39	92	3
Using multiple strategies to find needed information						
5. *Plan Learning*	548	812	42	56	83	5
6. *Plan + Adaptation Learning*	59	312	1	26	50	1

Table 1: Average, maximum and minimum effort expended adapting the five sample cases.

As was expected, when no adaptation cases are learned, learning additional response plan cases makes the system able to solve new problems with less adaptation effort—more similar cases are available. This is the foundation for the benefits of learning found in most CBR systems.

The table also shows that for this small test set, adding response plan learning to adaptation case learning produced a very small (and probably insignificant) benefit. This requires further investigation, but suggests that for achieving good performance from CBR systems, it is not sufficient to consider only the effects of learning domain cases: serious attention must be paid to the interaction of retrieval, similarity, and adaptation criteria. In our trials, the best performance came from simultaneously learning both response plans and adaptations.

We plan to follow up on these initial tests with a more controlled analysis of the effects of learning for a larger set of problem examples. We also intend to study the tradeoff between adaptation effort and adaptation quality, and the effects of adaptation learning on the quality of solutions generated.

Relationship to Other Approaches

Some early case-based reasoning systems included components for learning adaptation knowledge. For example, CHEF (Hammond 1989) bases its adaptations on both a static library of domain-independent plan repair strategies and a library of special-purpose *ingredient critics,* which suggest steps that must be added to any recipe using particular ingredients (e.g., that shrimp should be shelled before being added to a recipe). CHEF uses special-purpose procedures to learn new ingredient critics. PERSUADER (Sycara 1988) uses a combination of adaptation heuristics and previously-stored adaptation episodes to suggest adaptations. In both these systems, learned adaptations can only be reused in very similar situations; the adaptation cases learned by DIAL can be reused more flexibly. In addition, DIAL can perform adaptations from scratch when necessary to augment its library of cases.

DIAL's flexible approach to memory search is inspired by the memory search process of CYRUS (Kolodner 1984), and also relates to Oehlmann's 1993 question-based approach to introspective reasoning for guiding adaptation.

Our use of both case-based planning and case-based case adaptation provides advantages of both derivational (Veloso & Carbonell 1994) and standard transformational approaches to CBR. Transformational CBR approaches store and adapt a *solution* to a problem, while derivational approaches store and replay a *derivational trace* of the problem-solving steps used to generate a previous solution. For CBR tasks such as disaster response planning, derivations of solutions are not generally available, and planning from scratch is not satisfactory because domain theories are inaccurate and intractable. However, examples of prior solutions are readily available in news stories and casebooks used to train disaster response planners (e.g., Rosenthal et al., 1989). This favors a transformational approach to reusing disaster response plans. However, derivational approaches can simplify the reapplication of a case to a new situation, and the rationale for the system's choice of particular steps during adaptation of prior cases *is* available.

There is growing interest in alleviating case adaptation problems through interactive user adaptation (e.g., Bell, Kedar & Bareiss, 1994; Goel et al. 1991), including presenting the user with derivational traces (Goel et al., 1996). However, because those systems do not capture the results of the user's adaptations for future use, DIAL contributes a new approach to acquiring adaptation knowledge.

Conclusion

We have described ongoing research on a method for facilitating the acquisition of case adaptation knowledge. The method depends on representing adaptations as combinations of abstract transformations with memory search plans for finding the information needed to apply them. When no specific adaptation knowledge is relevant, reasoning from scratch is used to search memory for the information needed to perform an adaptation. When a similar adaptation has been performed in the past, case-based reasoning is used. This hybrid method makes it possible for a CBR system to acquire adaptation expertise through case-based reasoning about adaptation, and also to reason from scratch when needed to solve novel adaptation problems that would be beyond the scope of previous case-based adaptation methods. The view of adaptations as involving transformations plus memory search has also been used as the basis for interactive acquisition of adaptation knowledge.

Preliminary studies show speedup benefits from adaptation learning for an initial sampling of problems. More extensive studies are needed to corroborate these benefits, to investigate the affects of adaptation learning on the quality of adaptations suggested, and to examine whether the indexing scheme for adaptation cases is sufficient to make the approach resistant to the utility problem as large numbers of cases are learned. Another issue requiring study is how similarity assessment criteria should change as adaptations are learned (Leake, Kinley, & Wilson 1996). Nevertheless, we believe that combining transformational CBR for planning with derivational CBR for performing adaptation is a promising approach.

Acknowledgments

This work was supported in part by the National Science Foundation under Grant No. IRI-9409348. We thank the AAAI reviewers for their helpful comments.

References

Bell, B.; Kedar, S.; and Bareiss, R. 1994. Interactive model-driven case adaptation for instructional software design. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, 33–38.

Goel, A.; Kolodner, J.; Pearce, M.; and Billington, R. 1991. Towards a case-based tool for aiding conceptual design problem solving. In Bareiss, R., ed., *Proceedings of the DARPA Case-Based Reasoning Workshop*, 109–120. San Mateo: Morgan Kaufmann.

Goel, A.; Garza, A.; Grue, N.; Murdock, J.; Recker, M.; and Govindaraj, T. 1996. Explanatory interface in interactive design environments. In *Fourth International Conference on AI in Design*. In press.

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

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

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.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Kolodner, J. 1993. *Case-Based Reasoning.* San Mateo, CA: Morgan Kaufmann.

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

Leake, D. 1994. Towards a computer model of memory search strategy learning. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, 549–554.

Leake, D. 1995. Combining rules and cases to learn case adaptation. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, 84–89.

Leake, D.; Kinley, A.; and Wilson, D. 1996. Linking adaptation and similarity learning. In *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society.* In press.

Oehlmann, R.; Sleeman, D.; and Edwards, P. 1993. Learning plan transformations from self-questions: A memory-based approach. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 520–525.

Ram, A. 1987. AQUA: Asking questions and understanding answers. In *Proceedings of the Sixth Annual National Conference on Artificial Intelligence*, 312–316.

Rosenthal, U.; Charles, M.; and Hart, P., eds. 1989. *Coping with crises: The management of disasters, riots, and terrorism.* Springfield, IL: C.C. Thomas.

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

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

Veloso, M., and Carbonell, J. 1994. Case-based reasoning in PRODIGY. In Michalski, R., and Tecuci, G., eds., *Machine Learning: A Multistrategy Approach.* Morgan Kaufmann. Chapter 20, 523–548.