In Leake, D., ed., 1996, *Case-Based Reasoning: Experiences, Lessons, and Future Directions.* Menlo Park: AAAI Press/MIT Press, 1996.

Chapter 1

CBR in Context: The Present and Future

David B. Leake

1 Reasoning from Remindings

A father taking his two-year-old son on a walk reaches an intersection and asks where they should turn. The child picks a direction, the direction they turned in at that intersection the day before to go to the supermarket. The child explains: "I have a memory: Buy donut."

Another Vietnam?

Recently, [this question has] been asked in discussions over a deeper U.S. involvement around the world – in Bosnia, in Somalia, in Haiti.

Ed Timms, *Dallas Morning News*

Windows 95: Microsoft's Vietnam?

Headline in the *IN Jersey* Web page

Reasoning is often modeled as a process that draws conclusions by chaining together generalized rules, starting from scratch. Case-based reasoning (CBR) takes a very different view. In CBR, the primary knowledge source is not generalized rules but a memory of stored *cases* recording specific prior episodes. In CBR, new solutions are generated not by chaining, but by retrieving the most relevant cases from memory and adapting them to fit new situations. Thus in CBR, reasoning is based on remembering. As the passages starting this section illustrate, remindings facilitate human reasoning in many contexts and for many tasks, ranging from children's simple reasoning to expert decision-making. Much of the original inspiration for the CBR approach came from the role of remindings in human reasoning (Schank 1982).

The CBR approach is based on two tenets about the nature of the world. The first tenet is that the world is regular: similar problems have similar solutions. Consequently, solutions for similar prior problems are a useful starting point for new problem-solving. The second tenet is that the types of problems an agent encounters tend to recur. Consequently, future problems are likely to be similar to current problems. When the two tenets hold, it is worthwhile to remember and reuse current reasoning: case-based reasoning is an effective reasoning strategy.

CBR can also be beneficial, however, when a reasoner must solve problems that are quite different from prior experiences. As a case-based reasoner applies cases to increasingly novel problems, the CBR process changes from simple reuse to more creative problem-solving. The child in the example starting this chapter performs very straightforward CBR; he remembers a previous path when confronted with an identical decision point—a previously-visited intersection—and suggests repeating a prior plan. The commentators who apply lessons of Vietnam to Bosnia, however, must do more subtle reasoning to determine whether and how Vietnam applies to the new situation. The wag who sees Vietnam in Windows 95 is applying a reminding to a very new context, and reasoning in a creative way.

Regardless of whether a case-based reasoner solves a routine or novel problem, and of whether the problem-solving outcome is success or failure, the case-based reasoner learns from its experience. Complementary with the principle of *reasoning by remembering* is the principle that *reasoning is remembered*—that reasoning and learning are intimately connected. The knowledge of a case-based reasoner is constantly changing as new experiences give rise to new cases which are stored for future use. A case-based reasoner learns from experience to exploit prior successes and avoid prior failures.

This chapter provides context for the remainder of this book, introducing key principles of CBR, its basic algorithm, and relationship to other approaches, and discussing the state of the field, new trends, and key challenges. The following chapter provides a tutorial introduction to the field and to the principles for developing CBR systems. Later chapters provide case studies of key issues, in the context of specific projects. They are followed by perspectives that examine lessons learned and provide visions of the future of case-based reasoning.

2 Why CBR?

The study of CBR is driven by two primary motivations. The first, from cognitive science, is the desire to model human behavior. The second, from artificial intelligence, is the pragmatic desire to develop technology to make AI systems more effective.

Interest in CBR as a cognitive model is supported by studies of human reasoning which demonstrate reasoning from cases in a wide range of task contexts. For example, studies support the importance of remindings of prior examples in learning a computer text editor (Ross 1984), learning programming (Pirolli & Anderson 1985), mathematical problem solving (Faries & Schlossberg 1994; Ross 1984), diagnosis by automobile mechanics (Lancaster & Kolodner 1987) and physicians (Schmidt, Norman, & Boshuizen 1990), explanation of anomalous events (Read & Cesa 1991), and decision-making under time pressure (Klein & Calderwood 1988; 1989). Understanding these processes requires developing and testing theories of how humans store, retrieve, and apply prior cases.

Observations that people use case-based reasoning have also spurred interest in CBR as an AI technology. Humans are robust problem-solvers; they routinely solve hard problems despite limited and uncertain knowledge, and their performance improves with experience. All of these qualities are desirable for real-world AI systems. Consequently, it is natural to ask how CBR can advance AI technology. Discussions of this question have identified five main problems that can be ameliorated by case-based reasoning:

1. **Knowledge acquisition:** A classic problem in traditional knowledge-based systems is how to provide the rules on which the systems depend. The rule acquisition process can be laborious and unreliable: it may be difficult to elicit rules, and there is no assurance that those rules will actually be sufficient to characterize expert performance.[1] In some domains, rules may be difficult to formalize or the number of rules required may be unmanageably large.

   Because case-based reasoners reason from complete specific episodes, CBR makes it unnecessary to decompose experiences and generalize their parts into rules. Some task domains are especially natural for CBR, with cases that are suitable for CBR already collected as part of standard problem-solving procedures. In those domains, the cost of knowledge acquisition for CBR is very low. Mark, Simoudis, & Hinkle (Chapter 14) describe their experience in one such domain, autoclave loading. Other reports corroborate comparatively rapid development times for other CBR applications (e.g., Simoudis & Miller, 1991).

   Of course, not all domains are natural CBR domains; cases may be unavailable, or may be available but in a hard-to-use form (e.g., cases described with natural language text). In these situations, applying CBR may depend on a significant "case engineering" effort to delimit the information that cases must contain, to define the representation for that information and to extract that information from available data. Likewise, applying CBR requires developing criteria for indexing and reapplying prior cases. (E.g., Mark et al., Chapter 14; Voss, 1994). However, even if this initial process requires considerable effort, CBR can still provide overall benefits for knowledge acquisition. First, experts who are resistant to attempts to distill a set of domain rules are often eager to tell their "war stories"—the cases they have encountered. This facilitates gathering the needed data for CBR. Second, as discussed in the following point, after the initial case engineering effort it is often simple to augment and maintain the knowledge a CBR system needs.

2. **Knowledge maintenance:** Defining an initial knowledge base is generally only the first step towards a successful AI application. Initial understanding of the problem is often imperfect, requiring system knowledge to be refined. Likewise, changes in task requirements and circumstances may render existing knowledge obsolete. Although refinement of case representations and indexing schemes may be required as a task becomes better understood, CBR offers a significant benefit for knowledge maintenance: a user may be able to add missing cases to the case library without expert intervention.

[1] See Forsythe and Buchanan (1989) for a discussion of some of the problems in knowledge elicitation.

Also, because CBR systems do incremental learning, they can be deployed with only a limited set of "seed cases," to be augmented with new cases if (and only if) the initial case library turns out to be insufficient in practice. A CBR system needs only to handle the *types of problems that actually occur in practice,* while generative systems must account for *all problems that are possible in principle.*

3. **Increasing problem-solving efficiency:** People achieve satisfactory problem-solving performance despite the fact that commonplace problems in everyday reasoning, such as explanation and planning, are NP-hard (Bylander *et al.* 1991; Chapman 1987). Reuse of prior solutions helps increase problem-solving efficiency by building on prior reasoning rather than repeating prior effort. In addition, because CBR saves failed solutions as well as successes, it can warn of potential problems to avoid.

4. **Increasing quality of solutions:** When the principles of a domain are not well understood, rules will be imperfect. In that situation, the solutions suggested by cases may be more accurate than those suggested by chains of rules, because cases reflect what really happens (or fails to happen) in a given set of circumstances. In medical reasoning, for example, anecdotes about specific cases go beyond codified knowledge, serving as "the as-yet-unorganized evidence at the forefront of clinical medicine" (Hunter 1986).

5. **User acceptance:** A key problem in deploying successful AI systems is user acceptance: no system is useful unless its users accept its results. To trust the system's conclusions, a user may need to be convinced that they are derived in a reasonable way. This is a problem for other approaches: neural network systems cannot provide explanations of their decisions, and rule-based systems must explain their decisions by reference to their rules, which the user may not fully understand or accept (Riesbeck 1988). On the other hand, the results of CBR systems are based on actual prior cases that can be presented to the user to provide compelling support for the system's conclusions.

Successful use of CBR depends on addressing issues in how to acquire, represent, index, and adapt existing cases. The next section highlights how these issues fit into the CBR process and how they are being addressed in current systems. The following section highlights how these methods relate to other approaches.

3 A sketch of the CBR process

Case-based reasoning tasks are often divided into two classes, *interpretive CBR* and *problem-solving CBR* (e.g., Kolodner, 1993; Rissland, Kolodner, & Waltz, 1989). Interpretive CBR uses prior cases as reference points for classifying or characterizing new situations; problem-solving CBR uses prior cases to suggest solutions that might apply to new circumstances.

3.1 Case-based interpretation

In interpretive CBR, the reasoner's goal is to form a judgment about or classification of a new situation, by comparing and contrasting it with cases that have already been classified (e.g., Ashley & Rissland, 1987). For example, interpretive CBR plays a fundamental role in interpreting legal concepts and applying laws in the American legal system (e.g., Ashley, 1990; Bain, 1986; Branting, 1991; Cuthill, 1992; Sanders, 1994). A tax lawyer arguing that his or her client should receive a "home office" deduction does so by using precedents: by showing that the deduction was granted in similar previous cases and showing that those cases are more relevant than cases in which the deduction was not granted. Interpretive CBR is also important for tasks such as diagnosis; a problem can be diagnosed by comparing and contrasting the current symptoms to those in previous cases to determine the best diagnosis (e.g., Bareiss, 1989).

In its simplest form, interpretive CBR involves four steps. First, the reasoner must perform *situation assessment* (Kolodner 1993; Owens 1991), to determine which features of the current situation are really relevant. Second, based on the results of situation assessment, the reasoner retrieves a relevant prior case or prior cases. Third, the reasoner then compares those cases to the new situation, to determine which interpretation applies. Finally, the current situation and the interpretation are then saved as a new case on which to base future reasoning.

3.2 Case-based problem-solving

The goal of problem-solving CBR is to apply a prior solution to generate the solution to a new problem. For example, case-based design, planing, and explanation systems all retrieve and adapt solutions of similar prior problems. Like interpretive CBR, problem-solving CBR involves situation assessment, case retrieval, and similarity assessment/evaluation. In addition, the similarities and differences between new and prior cases are used to determine how the solution of the previous case can be adapted to fit the new situation. For example, a case-based planning system generates a new plan by retrieving a prior plan for a similar goal, determining the differences between the old and new goals, and adapting the plan to take the new goals into account.

Case-based problem-solving can be seen as exploiting the relationship between two different types of similarity. These types of similarity apply to two different spaces, the space of problem descriptions and the space of problem solutions. We illustrate their role in the solution generation process in figure 1. When presented with a new problem, a CBR system does situation assessment to generate a problem description, and then searches for problems with similar problem descriptions. The solutions of those problems are used as the starting point for generating a solution to the new problem. With the right way of describing problem, similar problems will have solutions that are similar—i.e., easy to adapt to the new situation.

The figure also suggests another benefit of CBR: that multiple types of knowledge can be used to encode equivalent information (Richter 1995). Information is contained not only in the case representation/indexing scheme and case base, but also in the similarity metric and adaptation knowledge, and the information contained by these knowledge sources overlaps. Consequently, the system developer has the flexibility to choose the best alternative for representing the needed knowledge.

3.3 Learning from successes and failures

After a solution has been generated, the final step is to apply the solution, to repair it if necessary, and to learn from the experience. Learning in CBR systems is driven by both successes and failures, and encompasses both speedup learning and acquisition of new knowledge.

Success-driven learning: When the CBR process is successful, the resulting solution is stored for future reuse, avoiding the need to rederive it from scratch. When CBR is combined with generative problem-solving, it can provide speedup learning (Veloso discusses an experimental demonstration of this speedup in Chapter 8). If the generative system has an imperfect domain theory, the benefit goes beyond speedup. In that situation, stored cases provide information beyond the information contained in the original domain theory: they provide the information that a particular solution did or did not work in a specific real situation. In this way, case acquisition refines initial domain knowledge and allows the system to favor solutions that are more likely to be successful, based on its experience. In addition, if solutions are provided by an external source, storing cases with those solutions may increase the range of problems that the system can solve.

Failure-driven learning: CBR is committed to the value of learning from failures as well as successes. First, failures reveal that learning is needed. Second, failures help focus decisions about what to learn: the needed learning must help avoid future failures. CBR systems learn both from *task failures,* in which their solutions are unsuccessful, and *expectation failures* (Schank 1982), in which observed outcomes differ—for better or for worse—from predictions. For example, when a planner generates a plan that is expected to work and doesn't, there are two failures. The task failure prompts the system to try to learn a successful plan; the expectation failure prompts the system to learn how to anticipate similar problems in the future, in order to avoid them (e.g., Hammond, 1989a). When a planner generates a plan that is successful even beyond expectations, there is no task failure, but there is still an expectation failure, prompting learning about how to anticipate and perhaps harness the unexpected good effects.

In CBR systems, failures can trigger multiple types of learning. When a failed solution is repaired, the new solution is stored; this is simply learning from a new successful solution. In addition, however, information about the failure itself can stored as data for future analysis when new information becomes available (Riesbeck 1981; Schank 1982; 1986) or to provide a warning about possible future failures that should be avoided (e.g., Bareiss, 1989; Hammond, 1989a; Kolodner & Simpson, 1989), Failures can also prompt revision of indexing criteria,

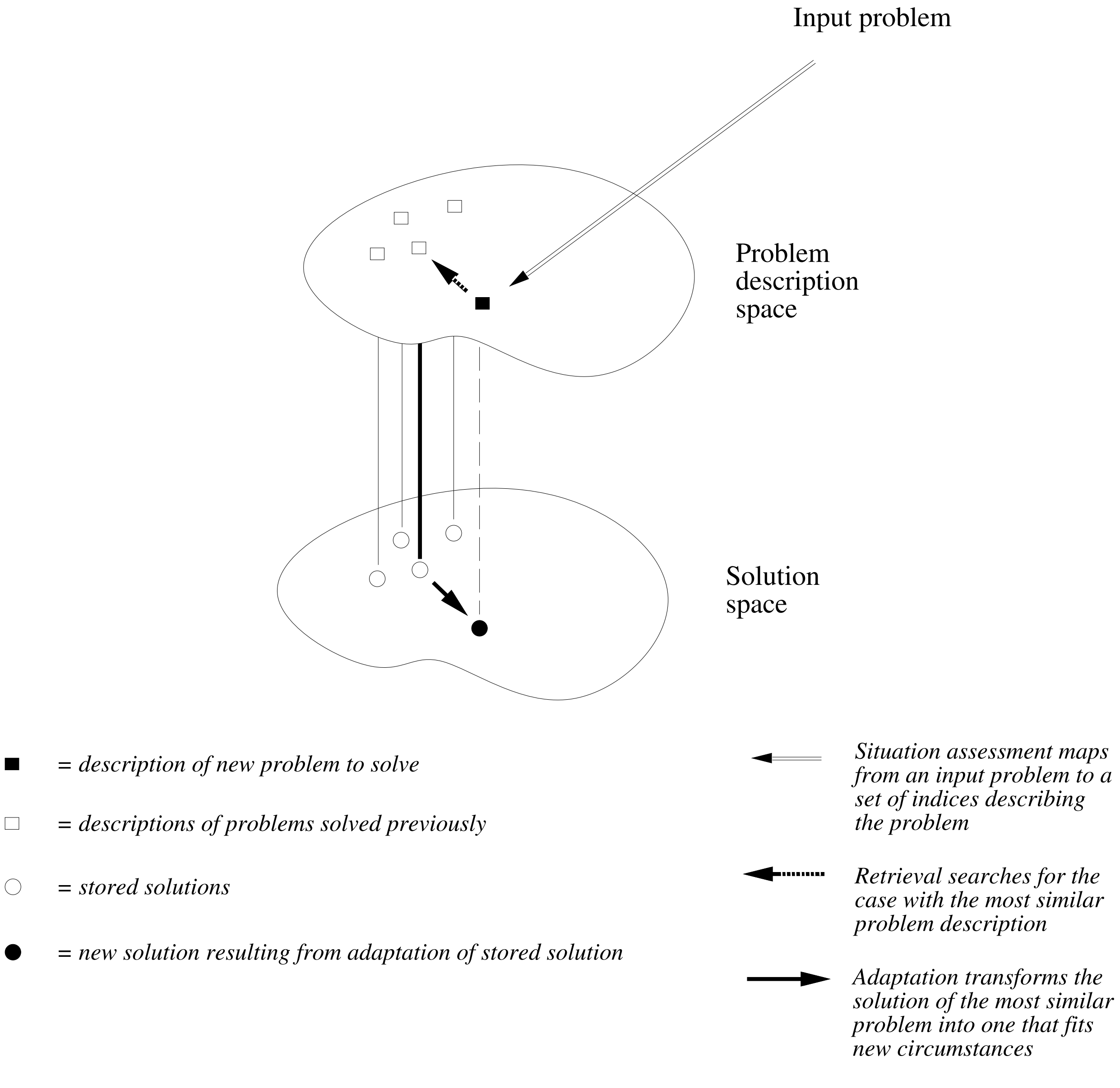

Figure 1: How case-based problem-solving generates a new solution.

to retrieve better cases in the future (e.g., Bhatta & Goel, 1993; Fox & Leake, 1995b, 1995c; Hammond, 1989a; Redmond, 1992; Sycara & Navinchandra, 1989).

3.4 A closer look

The previous section's description blurs many differences in CBR methods. This section illustrates a sampling of important variations in how fundamental issues are being addressed. The following tutorial chapter describes a number of them in more detail.

What cases contain: The previous section assumes that problem-solving systems will store and adapt prior solutions. An alternative approach is for them to store and reuse traces of how those solutions were derived, instead of the actual solutions. By capturing and replaying the reasoning trace involved in selecting problem-solving operators, rather than the problem solving steps themselves, the *derivational analogy* approach facilitates application of stored traces of processing to a wider class of problems (Carbonell, 1986; Veloso, 1994, Chapter 8). This approach has attracted interest not only for domain problem-solving tasks, but also in a number of systems that store and reuse reasoning traces for introspective reasoning and learning (e.g., Kennedy, 1995; Leake, Kinley, & Wilson, Chapter 11; Oehlmann, 1995; Ram & Cox, 1994).

How to retrieve: The previous sketch used purely top-down retrieval: A problem description was formed and used to select a relevant case. However, the indices needed are inextricably tied to the contents of the case library (which may change). Consequently, CBR research is also investigating the role of bottom-up influences to guide retrieval, favoring features that are useful to discriminate between the cases in memory (e.g., Cunningham, Bonzano, and Smyth, 1995; Owens, 1991). Although many CBR systems base retrieval on carefully constructed indexing vocabularies and problem descriptions, in order for retrieval to "zero in" on a small subset of the case library, other approaches exploit parallel hardware to maintain quick retrieval while considering large sets of cases (Kettler *et al.* 1994; Kolodner 1988b; Owens 1991; Stanfill & Waltz 1986). Retrieval based on nontraditional types of input information, such as bitmap images and CAD plans, is also being investigated (Voß 1994). As is discussed in Section 5.4, methods are also being developed make retrieval focus on cases that are likely to be easy to adapt.

Adaptation: Developing case adaptation criteria is a central open challenge for CBR (e.g., Allemang, 1993; Kolodner, 1991; Leake, 1994b). The case adaptation process in CBR systems is usually done by rule-based systems. Consequently, correct case adaptation requires that those rules capture both a theory of case adaptation, and the needed aspects of the domain theory to carry out changes. However, as has already been described, an important motivation for using CBR is often the lack of such a theory. As a result, developers defining adaptation rules must re-confront the knowledge acquisition problem for rule-based systems. Additional problems may arise because available cases can lack the internal structure needed for effective adaptation. For example, in case-based educational systems whose cases are video clips, the case content is simply not accessible.

Nevertheless, difficulties with case adaptation have led many CBR systems to simply dispense with adaptation, replacing the retrieve-evaluate-adapt cycle with *retrieve and propose* systems (e.g., Kolodner, 1991). Such systems exploit the memory processes developed in CBR research, while relying on a human user to adapt and evaluate solutions. This framework is the basis for many successful CBR applications, some examples of which are described in this volume (Kitano and Shimazu, Chapter 13 and Mark et al., Chapter 14).

New approaches are now being developed to overcome the adaptation problem. Because this is a central problem for the future of CBR, we devote section 5.4 to the promising new methods for addressing problems of automatic adaptation.

Similarity assessment: One issue in similarity assessment is how to determine the right features to compare. Decisions about which features are important are often based on explanations of feature relevance, but those explanations may be imperfect, leading to a need for robust similarity metrics that take the difficulties in specifying important features into account (e.g., Bento, 1994; Veloso, 1994). Another problem is that for some tasks, input problem descriptions are not sufficient to determine the similarity of old and new situations. For example, for the task of case-based explanation of anomalous events during understanding, the need to explain arises precisely *because* the input case is imperfectly understood. Thus situation assessment and similarity assessment may need to be combined. One method for the combination is *constructive similarity assessment*, which builds up a description of the input situation based on prior cases, and judges similarity by whether the retrieved case is adaptable to the new situation, rather than according to any static criteria (Leake 1992a; 1995b)

There is also growing recognition that the role of similarity judgments is to determine which cases are most *usefully* similar, given the desired results of the CBR process. A single set of static similarity criteria may not capture the right distinctions. Different cases may be most appropriate to consider depending on the relative importance of different dimensions for judging the success of the CBR process. For example, for case-based planning, some of the criteria might be reliability of the resulting plan, execution time for that plan, the time required to generate the solution, or even, if a creative solution is desired, the novelty of the result. The desire for similarity criteria to more strongly reflect intended use is reflected in approaches which replace traditional similarity judgments with judgments based instead on *adaptability* (Börner 1994; Leake 1992a; 1995b). For those approaches, similarity is aimed at facilitating solution generation.

Case evaluation: Like case adaptation, evaluation of the goodness of retrieved cases may be problematic for CBR systems, because evaluating candidate solutions may require considerable domain knowledge and reasoning effort. Although schemes have been developed to do rapid coarse-grained evaluation of some types of cases (e.g., Leake, 1992b), providing the right evaluation knowledge is difficult. An alternative approach is to base evaluation on the cases in the case library itself. Once a case is adapted to produce a new solution, similar cases can be retrieved and used as a dynamic benchmark for judging the quality of the adaptation: If similar solutions were unsuccessful, the cases provide a warning (Mark et al., Chapter 14).

Storage: Early CBR systems simply stored each case they generated. New work examines the effects of design decisions about the maximum size of case library (Santamaría & Ram, Chapter 12), as well as how to decide which cases must be stored in order to provide sufficient coverage (e.g., Smyth & Keane, 1995). Some systems also reason about which cases to try to acquire (Hunter 1989; Ram 1991).

4 Relationship to other approaches

Questions often arise about how case-based reasoning relates to areas such as memory-based reasoning, analogical reasoning, and other learning methods. This section highlights some relevant relationships and differences.

4.1 Memory-based reasoning

Memory-based reasoning (MBR) is often considered a subtype of CBR; MBR solves problems by retrieving stored precedents as a starting point for new problem-solving (e.g., Stanfill & Waltz, 1986; Waltz, 1989). However, its primary focus is on the retrieval process, and in particular on the use of parallel retrieval schemes to enable retrieval without conventional index selection. Parallel models can lead to very fast retrieval, but also raise new questions to address about the criteria for knowledge access (Kolodner, Chapter 16).

4.2 Analogical reasoning

Case-based reasoning can be viewed as fundamentally analogical: CBR solves new problems and interprets new situations by applying analogous prior episodes. As Burstein (1989) points out, cognitive models of analogy and CBR examine the same cognitive process; there is no clear line between research "on analogy" and "on CBR." Nevertheless, research on analogy was originally more concerned with abstract knowledge and structural similarity, while research on CBR is more concerned with forming correspondences between specific episodes based on pragmatic considerations about the usefulness of the result.

In addition, there have traditionally been differences in the scope of the process studied. Research on analogy has focused primarily on analogical mapping; CBR in addition studies related processes that occur both before and after mapping. For example, how to retrieve a source case is a fundamental part of CBR, while models of analogy may assume that source concepts are provided as input (e.g., Mitchell, 1993). Also, after a mapping between old and new situations suggests an analogous solution, CBR adapts that solution to fit the new situation, and stores it for future use.

If "analogy" is taken to refer only to analogical mapping, a possible description of the relationship between analogy and CBR is:

$$\text{Case-based reasoning} = \text{retrieval} + \text{analogy} + \text{adaptation} + \text{learning}$$

However, two caveats are necessary. First, some research on analogy takes a more extensive view, focusing not only on mapping but also seriously addressing related issues such as retrieval (e.g., Gentner & Forbus, 1991). Second, despite the breakdown of steps in this description of CBR, the steps of the CBR process are not independent. Considering them together provides an advantage over studying them individually, because their relationships can be exploited to facilitate and constrain processing in each one. For example, Leake (1995b) discusses how analogical mapping for explanations is facilitated by linking retrieval and mapping criteria, and Section 5.4 discusses the value of integrating other parts of the CBR process with case adaptation.

4.3 Databases and Information Retrieval Systems

Given that storage and retrieval are central aspects of CBR, a natural question is the relationship between CBR systems and databases or information retrieval systems (IR). Although an obvious difference is that full CBR systems adapt the cases they retrieve, the question is more subtle for case-based "retrieve-and-propose" systems or case-based educational systems that present cases but do not perform adaptation.

The retrieval process in CBR differs from that of information retrieval systems and standard databases by being more active. Database systems and IR systems leave the problem of how to formulate the right query largely to the user. In CBR systems, the system itself is often designed to start from an input description using features that are quite different from those included in the cases in memory, and to determine appropriate retrieval cues (e.g., Burke and Kass, Chapter 5; Rissland et al., Chapter 6; Wills & Kolodner, Chapter 4). The input description may also be incomplete (Cunningham, Bonzano, & Smyth 1995; Leake 1992a; 1995b; Owens 1991), forcing the system to determine what it needs to find out. Thus a crucial difference between IR and CBR is the importance of situation assessment and problem description processes in CBR.

Database systems are designed to do exact matching between queries and stored information, while the goal of CBR is to retrieve a "most similar" case or set of most similar cases. The most similar cases may include conflicts with some of the attributes that were specified in the retrieval query. In CBR, whether a particular case should be retrieved depends not only on the case itself, but whether there are better competitors. Despite these differences, databases can provide useful foundations for CBR memories and CBR can have useful synergies with information retrieval (e.g., Anick & Simoudis, 1993). For example, Kitano and Shimazu (Chapter 13) advocate the use of relational database management systems, combined with supplementary mechanisms to allow flexible query specification and partial matching during retrieval, to manage the case libraries for large-scale corporate CBR applications.

Likewise, techniques from CBR can be used to facilitate information retrieval, and the information available in information retrieval systems can be used to augment traditional case libraries. For example, Rissland and Daniels (1995) describe a retrieval approach in which CBR methods are used to retrieve a set of relevant cases from a richly-represented CBR case base, and the retrieved cases are in turn used as “seed” documents for the relevance feedback mechanism of a full-text information retrieval system. The IR system then retrieves additional cases from a large IR corpus of shallowly represented cases. The aim is twofold: to enable access to many more cases than normally available to CBR systems, and to improve recall and precision of retrieval from the IR corpus compared to standard IR techniques.

4.4 Learning methods

The learning done by CBR systems has interesting relationships with both inductive and explanation-based generalization methods.

Inductive learning: When case-based classification systems save exemplars of a concept, their learning can be viewed as a form of inductive concept learning. However, unlike traditional symbolic and neural network approaches to inductive learning, which define concepts by generalizations and discard the exemplars on which the generalizations are based, CBR systems define concepts entirely by the specific cases saved.

Retaining specific cases has important advantages. First, it makes decisions more explainable, by enabling a system to point to concrete cases supporting its decisions. Second, it makes the decisions more verifiable, because the user (whether a human or another system) can examine the cases directly to assess their applicability. Third, it is useful for resolving conflicts. For example, if the two most similar previous cases provide contradictory advice, it may be useful to know that they are contradictory and to explicitly compare and contrast them, balancing them against each other in light of the current situation, in order to decide which to follow. In systems that combine conflicting advice to offer only a single answer (e.g., neural networks), the conflict is hidden.

Another benefit of case learning in CBR is that it is incremental. No matter how few cases are contained in the case library, performance on those cases will be correct; as soon as a case has been stored by a CBR system, that case is available for use. As mentioned previously, this is an important advantage for applications, because it enables prototype CBR systems to function with a small set of “seed cases” and to add coverage by storing new cases incrementally if they prove to be needed (e.g., Mark et al., Chapter 14).

The CBR approach also contrasts with knowledge-poor inductive learning methods because it emphasizes the semantics of a domain, through similarity and retrieval criteria and case adaptation knowledge.

Instance-based learning (IBL), also called case-based learning, is an inductive learning method closely related to CBR. Rather than forming generalizations, IBL algorithms (Aha, Kibler, & Albert 1991) store previously-categorized episodes and use them to classify new inputs by assigning the same classification that was assigned to the most similar previous case (or cases). IBL systems forgo complex indexing, use feature-value representations, and do not address case adaptation, but they nevertheless appear very promising for certain applications (see Riesbeck, Chapter 17). They have also attracted attention as a form of CBR that is amenable to formal analysis (e.g., Jantke, 1992).

Explanation-based generalization: Explanation-based generalization (EBG) uses rules about a domain to explain why a training example has particular properties, and uses the explanation to guide generalization. The generalization is then stored for future use (DeJong & Mooney 1986; Mitchell, Keller, & Kedar-Cabelli 1986). Chunking (Laird, Rosenbloom, & Newell 1990), which collects traces of problem-solving steps and packages them for reuse, is a similar approach. Unlike inductive generalization, explanation-based generalization can do reliable learning from single examples.

CBR is similar to EBG in allowing single-example learning. However, CBR does not generalize cases at storage time. Instead, CBR adapts cases when adaptation is needed to solve a new problem. Thus CBR can be viewed as a form of *lazy learning* (e.g., Aha, 1996). (Because CBR does generalize *indices* (Hammond 1989a), ungeneralized cases can still be retrieved to deal with novel problems.)

Waiting to adapt cases avoids expending effort unless it is certain that the effort will help solve an actual problem. For example, the SWALE system, which uses a case-based method to build explanations for story understanding, stores its explanations without generalization, and generalizes them only if generalization is needed to subsume future situations. Even then, generalization is only done to the extent needed to subsume them (e.g., Kass, Leake, & Owens, 1986).

Another important difference between CBR and EBG is that adaptation is often much more flexible than explanation-based generalization. Adaptation can include operations other than generalization, such as specialization and substitution, and may involve modifications that are not guaranteed to be correct. For example, SWALE's adaptation process may use heuristics that include hypothesizing new causal rules. The flexibility of case adaptation precludes applying the "eager" approach of EBG generalization to case adaptation: it would be possible to generate an overwhelming number of variants for any candidate solution, many of them unreliable and most of them unlikely to be reused. However, because case adaptation in CBR is only done in response to the need to solve a specific new problem, and because adaptations are only done to the extent required by the new situation, the process is constrained, and the reasonableness of results can be verified in context (Leake 1995a).

5 Progress and Directions

To take stock of the state of CBR, this section looks at progress on general CBR issues, at some particularly noteworthy current task areas, and at work on the area of CBR that is least understood, and consequently the greatest research challenge: case adaptation.

5.1 Progress on general issues in applying CBR

Kolodner's (1993) CBR textbook concludes with a list of general challenges and opportunities for CBR, including knowledge engineering issues such as scaleup, evaluation, and developing CBR tools. Since that time, important progress has been made in each of those areas.

Scaling up: A vital question for applying ideas developed in testbed systems is whether they will "scale up" to large problems. The scale-up of CBR algorithms is now being tested in both CBR research and applications. For example, Veloso (Chapter 8) describes tests confirming successful scaleup of Prodigy/Analogy with a library of 1000 cases; Kitano and Shimazu (Chapter 13) describe the development and deployment of SQUAD, a software quality control advisory system with a case library of over 25,000 cases; Cassiopée, a case-based diagnostic aid for jet engines, uses 16,000 cases for its diagnosis process (Goodall 1995); and ALFA, a case-based system for power plant load forecasting, is in operation with a case library of 87,000 cases (Jabbour *et al.* 1988). These and other examples support that current technology is sufficient for CBR to be viable with large case bases. However, as Kolodner points out in Chapter 16, it is important to note that large case bases are not necessarily required by CBR. The size of the required case base depends strongly on the task being addressed. For some tasks, suitable performance may require only a few cases; for others, many thousands may be required.

Evaluation: Initial CBR research focused primarily on identifying key issues and methods for attacking them; progress was measured by qualitative advances in the types of problems that could be solved and by the insights they provided about human reasoning and reminding. As the field has matured, increased attention has been given to more quantitative evaluations of CBR systems and methods. Many case studies of evaluation of CBR systems and discussions of how to perform that evaluation are available in the proceedings of the 1994 AAAI Workshop on Case-Based Reasoning (Aha 1994). The chapters in this volume substantiate approaches with a mixture of qualitative and quantitative evaluations.

One difficulty in using quantitative evaluation to guide system construction is that CBR systems are complicated artifacts whose performance depends on many subtle interactions between components, as well as on the characteristics of the domain. Santamariá and Ram (Chapter 12) describe a methodology that addresses this problem by developing models of system performance, doing experimentation to validate those models, and using the models to guide design decisions.

From the perspective of applied CBR in a production setting, all evaluation criteria are subsumed in a single criterion: the effect on the bottom line. In order to be useful, CBR systems must be cost-effective. Many fielded applications attest to the cost-effectiveness of CBR applications and also on when and how CBR should be applied (e.g., Kitano and Shimazu, Chapter 13, and Mark et al., Chapter 14).

Tools: Because one of the motivations for CBR is to decrease the burden of developing intelligent systems, the ease of developing CBR systems is a crucial concern. The need for tools to enable an expert to participate directly in the case acquisition and case engineering process has been recognized from the early days of CBR (e.g., Riesbeck, 1988). An important part of current work on large-scale CBR projects is developing tools that manage basic parts of the CBR process (e.g., Kitano & Shimazu, Chapter 13, and Mark et al., Chapter 14). The FABEL project, for example, has developed a suite of both general and domain-specific tools to support case management, retrieval, assessment and adaptation of architectural designs (e.g., FABEL Consortium, 1993; Voss, 1994).

Some projects have also developed tools to ease the construction of particular classes of case-based systems. Examples include Design-MUSE (Domeshek *et al.* 1994), which eases construction of case-based design aids, REPRO (Mark et al., Chapter 14), which is a tool kit to help in the development of case-based advisory systems, and the ASK tool, for building browsable corporate memories (Ferguson *et al.* 1991). Tools have also been developed to help to build case-based teaching systems to facilitate students' case acquisition in new domains. For example, the GuSS tool facilitates building learning-by-doing systems that allow a student to do active learning in a low-risk, simulated social environment (Burke and Kass, Chapter 5).

Commercial CBR shells are available as well. CBR shells provide mechanisms to support case retrieval, such as nearest-neighbor retrieval or automatically generated decision trees, and may allow users to interactively provide additional information as needed during retrieval. They may also provide sophisticated interfaces to facilitate creating and editing the case base, as well as facilities for importing information in existing databases. Watson (1995) provides a comparative sketch of a number of tools including ART*Enterprise, Case-1, Casepower, the Inference CBR2 family, Eclipse, ESTEEM, KATE, ReCall, ReMind, and CBR Works. Althoff et al. (1995) provide a detailed comparative evaluation of five CBR shells: CBR Express, ESTEEM, KATE, ReMind, and CBR Works.[2] In this volume, Mark et al. (Chapter 14) discuss some experiences with commercial shells and the strategy of building components that add needed functionality "on top of" the functionality provided by a commercial CBR shell.

As Riesbeck (Chapter 17) points out, additional tools are needed to aid human indexing (see Goldstein, Kedar, & Bareiss, 1993, and Osgood & Bareiss, 1993, for examples of this type of tool), and another need is "catalogs" of the types of indices appropriate for particular tasks and domains (e.g., Domeshek, 1992; Leake, 1992b; Schank and Osgood, 1990). Likewise, tools are needed to facilitate acquisition of adaptation knowledge (one method under development is sketched in Leake, Kinley, & Wilson, Chapter 11).

Methodologies: Full acceptance of CBR by industry depends on establishing software development methodologies for CBR, to define how to organize and develop CBR projects. Lessons from CBR applications form a foundation for defining such methodologies. As

[2]CBR Works was previously named S^3-Case, and is referred to by that name in both the references.

Kitano and Shimazu describe, those lessons have already been used to define a methodology for building and maintaining large scale experience-sharing CBR systems at NEC.

One fundamental principle revealed by many experiences is the value of an iterative development process. Because CBR systems can provide useful results even with a partial case library, systems can be fielded with a set of seed cases that is augmented as gaps are revealed during use. Additional study is needed on issues in initial case engineering and case-base maintenance throughout the life-cycle of CBR applications.

5.2 Some noteworthy uses of CBR

CBR has been applied to a full spectrum of AI tasks, such as classification, interpretation, scheduling, planning, design, diagnosis, explanation, parsing, dispute mediation, argumentation, projection of effects, and execution monitoring. Many of these areas will be discussed in the following chapter. This section will discuss a few others that reveal noteworthy aspects of the CBR process and its relevance to important areas.

Creative reasoning: A common misconception about case-based reasoning is that it only applies if new problems are very similar to those solved in the past. Although CBR is a simple and effective method for that type of reuse, it is also an interesting framework for creative reasoning. Creativity can enter into the CBR process in flexible retrieval processes that result in novel starting points for solving new problems, in mapping processes that form novel correspondences, and in flexible case adaptation to generate novel solutions. These processes have been used as a basis for case-based models of creative explanation (e.g., Kass, 1990; Kass, 1992; Schank, 1986; Schank & Leake, 1986; Schank & Leake, 1989), design and problem-solving (e.g., Bhatta, Goel, & Prabhakar, 1994; Kolodner & Penberthy, 1990; Kolodner, 1994, Wills and Kolodner, Chapter 4), story generation (Turner 1994), and understanding (Moorman & Ram 1994).

Case-based aiding systems: Case-based aiding systems use automated case memories to support human reasoners. The case memories provide the experiences that human reasoners may lack, suggesting successful prior solutions and warning of prior failures. The human reasoners maintain final control, performing adaptation and evaluation of solutions. Not only does this interaction provide practical advantages, by avoiding the need for automatic case adaptation and evaluation, but humans readily accept and appreciate the availability of advice. A classic example is Lockheed's Clavier (Mark et al., Chapter 14), an aiding system which uses its case library both to suggest autoclave layouts and to provide feedback on user solutions. Another is the SQUAD system at NEC (Kitano and Shimazu, Chapter 13).

A task area with particularly active research is interactive decision-aiding for design (e.g., FABEL consortium, 1993; Griffith & Domeshek, Chapter 3; Gómez de Silva Garza & M. Maher, 1996; Hua & Faltings, 1993; Smith, Lottaz, & Faltings, 1995; Sinha, 1994; Sycara et al., 1991). Case-based design-aiding systems often support the design process not only with suggestions, but through mechanisms to facilitate case combination and adaptation by the user. There is also considerable interest in case-based decision-aiding for medical applications such as design of radiation treatments (e.g., Berger, 1995a; Kahn & Anderson, 1994, Macura & Macura, 1995).

A particularly active area in fielded applications is case-based help desk systems. Such systems provide a resource for human help desk employees, who can call upon an automated case library to present similar prior questions and answers. Case-based help desk systems can provide significant performance improvements with rapid development time. Compaq's SMART system (Acorn & Walden 1992), a case-based call tracking and problem resolution system that aids customer service representatives at a central help line, was built in six months and improved productivity sufficiently to pay for itself within a year. CBR aiding systems are also being used to provide direct support, bypassing the need for customer service representatives. Compaq's QuickSource, a CBR application for printer diagnosis (Nguyen, Czerwinski, & Lee 1993), was not only used as part of SMART but also shipped directly to customers with printers to allow them to perform their own diagnosis. Some issues in developing case-based help desks are discussed by Kriegsman & Barletta (1993) and Mark et al. (Chapter 14).

Corporate memories: Case bases are an appealing way to capture and share experiences of multiple agents. The case libraries accumulated by case-based help desk systems are one example of corporate memories, and are an interesting example of the use of cases for knowledge sharing. Case bases for particular help desk domains are now available as commercial products (Inference Corporation, 1995), providing a form of "instant experience" that can be augmented by adding cases if novel problems arise. In this volume, Kitano and Shimazu Chapter 13 describe the use of CBR as the basis of a large-scale corporate *experience sharing architecture.*

Case-based education: Large-scale efforts are also under way to apply lessons from the cognitive model of case-based reasoning to training and teaching. Although case studies already play a useful role in legal and medical education, students using them generally do not confront the complexity of real episodes and do not have the opportunity to act to execute, evaluate, and revise their solutions (Williams 1992). In Chapter 15, Schank examines the ramifications of CBR for education and argues for a new educational curriculum designed to support case acquisition through learning by doing. He proposes that learning be done in *goal-based scenarios* (Schank *et al.* 1993/1994), rich learning environments in which students learn skills and conceptual knowledge through activities in pursuit of compelling goals. Such learning environments can use CBR methods to facilitate students' own case acquisition, by presenting students with information about others' experiences, in the form of relevant cases, when they are likely to be useful. Burke and Kass (Chapter 5) describe a case-based teaching system reflecting this philosophy. More generally, the computational models developed by CBR can contribute to education by providing concrete suggestions about what makes a good problem, the range of problems that students should solve, and the kinds of resources that should be made available to student learners (Kolodner, Hmelo, & Narayanan 1996).

Knowledge navigation: The knowledge access issues that are crucial to CBR will also play a central role in developing "digital libraries" of on-line information. Consequently, a promising new area for applying the results of CBR is "knowledge navigation" to search and browse on-line repositories of information. For example, lessons learned about indexing and retrieval in CBR can be used to help in characterizing information and guiding information search.

The capability of CBR systems to describe and refine information needs by examples also promises to play an important role in making digital libraries easier to access. As Hammond (Chapter 7) points out, it is often natural to request information by reference to specific examples (e.g., when being shown a car by a car salesman, to ask for "something like that, but a little sportier"). CBR methods to support that type of query have the potential to significantly facilitate interaction with on-line repositories of information.

5.3 Opportunities for combining CBR with other methods

In many different task areas, attention is also being devoted to the combination of CBR with other methods. That combination can involve CBR systems using other methods for support, CBR systems integrated with other methods, or CBR systems in a purely support role.

Supporting CBR with other methods: The strong CBR stand towards cognitive modeling is that CBR is the central human reasoning process. Although other sources of knowledge and other reasoning processes may be used, their role is to support the CBR process (Kolodner, Chapter 16). An example of a combined system that uses other methods to support CBR is the case-based design system JULIA (Hinrichs 1992), which uses supporting systems such as a constraint poster (Stefik 1981) and a reason maintenance system (Doyle 1979) to support a fundamentally case-based design process. Other CBR systems fall back on rule-based reasoning as a backup to CBR, using rules when no relevant cases are available (e.g., Goel, et al., 1994; Koton, 1988).

Integrated systems: More balanced combinations of CBR with other reasoning methods are also being investigated. For example, the INRECA project focuses on combining CBR and inductive learning techniques to perform diagnosis (Auriol *et al.* 1995). Likewise, case-based and rule-based reasoning may be combined in many ways. Cases may guide interpretation of rules; cases may be used to focus rule-based reasoning; or the CBR system may be one component among equals in a multistrategy reasoning system (Althoff & Wess, 1991; Auriol, 1995; Bartsch-Spörl, 1995; Branting & Porter, 1991; Koton, 1988; Goel, 1989; Golding & Rosenbloom, 1991; Portinale & Torasso, 1995; Skalak & Rissland, 1991). Metareasoning about system performance, based on a self-model, can be used to guide learning to refine the CBR process itself (e.g., Arcos & Plaza, 1994; Birnbaum et al., 1991; Fox, 1995; Fox & Leake, 1995a, 1995b, 1995c; Leake, Kinley, and Wilson, Chapter 11; Ram & Cox, 1994). CBR may be also be applied in a fully integrated framework that performs strategic reasoning about each processing step (e.g., Aamodt, 1994; Armengol & Plaza, 1994). In this volume, Veloso (Chapter 8) describes the use of CBR within an integrated architecture.

Hybrid approaches have proven useful for applications as well. Mark et al. (Chapter 14) argue that CBR should be viewed as part of a technology mix, and Hammond (1993) has described the usefulness of a class of CBR systems—that he calls "CBR-lite^{tm}" systems—which exploit the most applicable parts of a number of technologies, including CBR, to maximize performance.

CBR to support other systems: Riesbeck (Chapter 17) proposes that a key future role of CBR will be for building "intelligent components" to improve the performance of a surrounding system with minimal development cost. Because CBR systems retrieve complete solutions, they offer an "anytime" ability to produce a first-pass solution rapidly, and then to refine it if the time constraints of the surrounding system allow additional processing to be done (Dean & Boddy 1988). Learning from actual processing episodes also automatically tailors the output of the intelligent component towards precisely what the surrounding system needs.

5.4 Case adaptation

A final research challenge and opportunity centers on one of the basic steps of CBR: case adaptation. Adaptation plays a fundamental role in the flexibility of problem-solving CBR systems; their ability to solve novel problems depends on their ability to adapt retrieved cases to fit new circumstances and on their ability to repair solutions that fail.

The difficulty arises in how to perform the adaptation. There are many ways to adapt a case; effective adaptation depends on having both knowledge of possible adaptations and ways to select those that will be appropriate and effective in a particular situation. The problem is illustrated by a joke concerning Michael Jordan, a basketball superstar. In 1993 he shocked his fans by announcing that he had decided to leave basketball for baseball. In 1995, he was frustrated by a baseball strike that resulted in the baseball team owners locking out their teams and hiring replacement players, and rumors suggested that he would soon return to basketball. A joke framed the decision as Jordan selecting an adaptation to repair the situation:

> Recent speculation is that Michael Jordan is switching back to basketball. We think there is a simpler explanation: He's trying to settle the baseball strike by using replacement owners.[3]

Central questions for adaptation are which aspects of a situation to adapt, which changes are reasonable for adapting them, and how to control the adaptation process. Answering those questions may require considerable domain knowledge, which in turn raises the question of how to acquire that knowledge. Many CBR systems depend on that knowledge being encoded a priori into rule-based production systems. Unfortunately, this approach raises the

[3] Tom Comeau, March, 1995.

same types of knowledge acquisition issues that CBR was aimed at avoiding. It has proven a serious impediment to automatic adaptation.

Recognizing that practical retrieval technologies are available, but that the general adaptation problem remains extremely difficult for CBR systems, experts in both CBR research (e.g., Kolodner, 1991) and applications (e.g., Barletta; 1994; Mark et al., Chapter 14) agree that the best use of CBR for today's applied systems is as advisory systems that rely on the user to perform evaluation and adaptation.

However, understanding case adaptation remains important both from a cognitive modeling perspective—for understanding human case-based reasoning—and from a practical one—for developing fully autonomous CBR systems. Recent calls have been made for renewed attention to case adaptation (Leake 1994b; Aha & Ram 1995), and some promising approaches are emerging. These new approaches fall into two categories. The first category focuses on the knowledge and methods used during the adaptation process itself. The second addresses the problem indirectly, by trying to decrease the need for adaptation. For example, the adaptation problem can be alleviated by retrieving cases that require less adaptation to fit the current task, or by revising the task to decrease the need for adaptation.

5.4.1 Improving adaptation capabilities

Most research on case adaptation has assumed that adaptation must be done in a completely autonomous way by rule-based systems. This results in a knowledge acquisition problem for adaptation rules. Two alternatives are to decrease the need for domain-specific adaptation rules, by making adaptation rules more flexible, or to avoid the need for adaptation rules by applying a case-based approach to the adaptation process itself:

- **Using flexible adaptation rules:** One of the problems in developing adaptation rules is how to balance the operationality and generality of adaptation rules. 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, 1989a; Hinrichs, 1991), but they may be hard to apply without additional specific domain knowledge. Specific rules, on the other hand, may be more operational, but cannot easily be applied to new tasks, forcing new rules to be coded for each new task and domain.

  For example, the adaptation rule *add a step to remove harmful side-effect* has been proposed to repair plans with bad side-effects in case-based planning (Hammond 1989a). This rule is widely applicable—it applies to any plan—but 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 to mitigate the bad effect may require considerable domain knowledge. An alternative is a very specific version of the rule, such as *add the step "rotate radiation sources" to remove harmful side-effect "excess radiation"* (Berger & 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.

One approach to the operationality/generality tradeoff is to replace traditional adaptation rules with *adaptation strategies* that operationalize abstract rules by packaging them with memory search information (Kass 1990; 1994). They strike a balance between domain-independent and domain-specific rules by providing domain-independent information about how to find the domain-specific information needed to solve a particular adaptation problem.

- **Derivational analogy:** Another alternative is to change the nature of the case that is stored. Rather than storing and directly reusing a solution itself, the CBR system can store a trace of how that solution was generated and replay it in the new situation. When the solution is replayed to solve future problems, the replay process can directly take into account differences between the old and new situations (e.g., Carbonell, 1986; Veloso, Chapter 8).

- **Using adaptation cases:** Because CBR has been shown to decrease the knowledge acquisition burden for domain knowledge in general, another appealing direction is case-based adaptation (e.g., Berger, 1995b; Leake, 1994a; Sycara, 1988). Problems remain, however, in how to acquire these adaptation cases, and how to apply adaptation cases to novel situations. Normally the reuse of adaptation knowledge is restricted to situations in which prior adaptations apply very directly.

- **Supporting adaptation with introspective reasoning:** Introspective reasoning about the adaptation process can be used to guide adaptation decisions and carry out adaptations and the search for needed information in a more flexible way (Leake, 1993a; Leake, 1995c; Leake, Kinley & Wilson, Chapter 11; Oehlmann, 1993; Oehlmann, 1995).

- **Combining rules and cases for adaptation learning:** Another new direction based on introspective reasoning is to combine rule-based and case-based adaptation, using reasoning from general heuristics when necessary, but whenever possible reusing more specific information from stored introspective reasoning traces for prior adaptations. This method allows flexible solution of new problems while relying on specific experiences when possible (Leake, Kinley, and Wilson, Chapter 11).

- **Hierarchical approaches and reuse of subcases:** Another way to facilitate adaptation is by representing cases hierarchically (e.g., Aha & Branting, 1995; Goel et al., 1994; Marir, 1995; Redmond, 1992; Smyth & Keane, Chapter 9). Hierarchical representations allow cases to be reused at the most specific level of abstraction that can be easily applied to the new situation. In addition, when individual subparts of a retrieved solution must be adapted, they can be adapted in context of the abstract outline of the entire solution. Francis & Ram (Chapter 10) describe a model of reuse of subcases in which an asynchronous memory mechanism retrieves relevant pieces of multiple prior cases to be spliced in as adaptation progresses.

5.4.2 Decreasing the need for adaptation

Case adaptation takes place within a larger context, including both the interaction with other components of the CBR system and with the user of the entire system. This context provides a range of possibilities for decreasing the need for adaptation.

Alleviating the adaptation burden by refining other components: One way to alleviate the problem is to tie other components of the CBR system more closely into the adaptation process. These methods aim at more perspicacious case retrieval and similarity assessment, as well as at stored cases that are easier to adapt:

- **Refining indices to favor more adaptable cases**

  Because the difficulty of case adaptation depends crucially on the cases that are retrieved, improvements in retrieval can significantly ameliorate the adaptation task. Fox and Leake (1994; 1995b; 1995c) apply introspective reasoning after problem-solving to evaluate whether the best case was retrieved, and, if not, to adjust retrieval criteria to focus future retrievals on more adaptable cases.

- **Basing retrieval directly on adaptability**

  Given that indexing and similarity criteria are simply proxies for adaptability, another promising direction is to integrate retrieval and similarity judgments with adaptation. Adaptation-guided retrieval, described by Smyth and Keane in Chapter 9, retrieves directly on the basis of evidence of likely adaptability.

- **Basing similarity judgments on adaptability**

  Many CBR systems use a two-step retrieval process, first retrieving a set of promising candidate cases, and then doing a finer-grained evaluation of the similarity of the retrieved cases and the new situation. Because the goal of their similarity judgment is to determine which cases can be applied to the new situations, it can be beneficial to integrate the similarity decision with the adaptation process, to favor cases not by the match between their features but instead by whether their features can be adapted to match (Börner 1994; Leake 1992a; 1995b).

- **Preparing for adaptation at storage time**

  CBR practitioners have long recognized the need for case representations to provide the information needed to facilitate future adaptation (e.g., Kass & Leake, 1988). This basic tenet for designing representations can be taken further, however, to guide preprocessing of specific cases at storage time in order to facilitate future adaptation. For example, Redmond's (1992) snippets facilitate reuse by making subparts of an episode individually accessible; Garland and Alterman (1995) propose that before plans are stored, they should be summarized and refined to remove superfluous information and inefficient steps, and then segmented into units expected to be useful.

- **Learning from user adaptation:** When the user manually adapts a case in an interactive CBR system, a trace of the user's adaptation process can be recorded for future use. That trace can then be replayed when needed for similar adaptation problems (Leake, Kinley, & Wilson, Chapter 11). This approach to adaptation learning can be viewed as a form of derivational analogy for reuse of case adaptations.

Supporting user adaptation: Applied CBR systems often forgo adaptation entirely. They function solely as memories, retrieving cases and presenting them to the user, who adapts them on his or her own. However, some recent projects have begun to take a middle approach. The idea is for the CBR system to support and facilitate user adaptation while still leaving the process primarily under user control. For example, the user may make high-level adaptation decisions, with the system using model-based information to suggest possible adaptation points and inform the user of relevant constraints or track important interactions (e.g., Bell, Kedar, & Bareiss, 1994; Smith, Lottaz, & Faltings, 1995; Sinha, 1994). After a case has been adapted to provide a new solution, the CBR system can also help evaluation of the result by presenting the user with similar prior solutions and their outcomes (Mark et al., Chapter 14).

Adapt the context, not the case: The goal of a CBR system is to generate a useful solution. Normally, this is accomplished by adapting a prior solution to apply to a new problem. An alternative method is to adapt the problem situation itself, so that the retrieved case can apply to the new problem without adaptation. For example, in CBR systems that retrieve and display video clips for educational purposes, no adaptation of the video clips is possible. However, for the purposes of such systems it is equally effective to *adapt the context,* by explaining why the retrieved video clip is relevant. "Bridging" generates a description of why a case is relevant, showing how the case applies. Burke and Kass (Chapter 5) describe a system which presents students with video clips and explanations of their significance. The "bridge" provided by that explanation makes the retrieved case useful.

6 The contents of this book

This book presents a selection of recent progress, issues, and directions for the future of case-based reasoning. It includes chapters addressing fundamental issues and approaches in indexing and retrieval, situation assessment and similarity assessment, and in case adaptation. Those chapters provide a "case-based" view of key problems and solutions in context of the tasks for which they were developed. It then presents lessons learned about how to design CBR systems and how to apply them to real-world problems. It closes with perspectives on the state of the field and the most important directions for future impact.

The case studies presented involve a broad sampling of tasks, such as design (Chapters 3, 4, and 9), education (Chapters 5 and 15), legal reasoning (Chapter 6), planning (Chapters 10, 11, 12), decision support (Chapters 3, 13 and 14), problem-solving (Chapters 4, 8 and 14), and knowledge navigation (Chapter 7). In addition, they experimentally examine one of the fundamental tenets of CBR, that storing experiences improves performance (Chapters 8 and 12). The chapters also address other issues that, while not restricted to CBR per se, have been vigorously attacked by the CBR community. These include creative problem-solving (Chapter 4), strategic memory search (Chapters 6 and 11), and opportunistic retrieval (Chapters 4, 5, and 10).

The discussion of research issues and results is complemented with experiences and lessons from building CBR applications for tasks such as experience sharing (Chapter 13), autoclave loading, diagnosis, help desk support (Chapter 14), and education (Chapters 5 and 17). These identify crucial issues and approaches for developing and deploying applied systems.

This book closes with perspectives on the state of case-based reasoning and its future impact. In Chapter 14, Mark et al. discuss insights about applying CBR, based on their experiences with a number of CBR applications. In Chapter 15, Schank examines the role of case acquisition in human learning and argues that case-based reasoning has profound implications for transforming education. In Chapter 16, Kolodner first identifies and dispels misconceptions that distort perceptions of CBR and then underlines key problems to attack in order to advance the field. In Chapter 17, Riesbeck presents a vision for the future of AI, the role CBR will play in that future, and the resulting challenges for the next generation of case-based reasoning systems. This volume provides a vision of the present, and a challenge for the future, of case-based reasoning research and applications.

7 Conclusion

This chapter has placed case-based reasoning in context, delineated some of its tenets, and pointed to new directions to be addressed by the case studies in the remainder of the book. The heart of CBR is the importance of experiences and lessons—of remembering and reusing specific experiences and the lessons that they provide. This volume applies that principle of CBR to examining CBR itself, by presenting experiences and lessons *in using CBR*.

Experiences with the current generation of CBR systems suggest central challenges for future research, such as the case adaptation problem; they also show how to apply CBR technology. Finally, they show where CBR may have the most impact. The following chapters present individual perspectives that illuminate important experiences, lessons, and future directions for advancing case-based reasoning.

8 Acknowledgments

The initial idea for this book came out of The AAAI-93 Workshop for Case-Based Reasoning (Leake 1993b), sponsored by the American Association for Artificial Intelligence. I would like to thank all the chapter authors for their contributions, and also to thank Ray Bareiss, Robin Burke, Eric Domeshek, Anthony Francis, Janet Kolodner, Ashwin Ram, Chris Riesbeck, Raja Sooriamurthi, and David Wilson, for their generous assistance as the book was being prepared. I am grateful to Ken Ford, Editor in Chief of the AAAI Press, for his enthusiastic support for this volume and his instrumental role in its coming into being, and to Mike Hamilton of AAAI Press, who helpfully guided arrangements and the production process. The author's work is supported by the National Science Foundation under Grant No. IRI-9409348.

9 Some CBR Resources

The tutorial in chapter 2 of this volume presents a more thorough discussion of key CBR principles and issues and how to develop CBR systems. Kolodner's (1993) textbook *Case-Based Reasoning* presents an extensive examination of CBR issues and survey of American CBR research. Riesbeck and Schank's (1989) *Inside Case-Based Reasoning*, and Schank, Riesbeck, and Kass's (1994) *Inside Case-Based Explanation*, present distillations of a number of influential dissertations on case-based reasoning research, in addition to "micro" versions of CBR programs developed to facilitate experimentation. Aamodt and Plaza's (1994) overview article includes an introduction to the field with highlights of American and international CBR research.

The most complete picture of the field is provided by the proceedings of the many case-based reasoning workshops. Proceedings are available for the larger workshops in the United States (Kolodner 1988a; Hammond 1989b; Bareiss 1991; Leake 1993b; Aha 1994) and in Europe (Wess, Althoff, & Richter 1994; Haton, Keane, & Manago 1995; Watson 1995), as well as for the First International Conference on Case-Based Reasoning (Veloso & Aamodt 1995).

There are also numerous electronic CBR resources, including discussion lists and archives of many CBR sources. The following list is a sampling of those available as of January 1, 1996.

9.1 Mailing lists/Newsletters

- **AI-CBR:** A mailing list including announcements, questions, and discussion about CBR, managed by Ian Watson and Farhi Marir at Salford University. To join, send an electronic mail message to `mailbase@mailbase.ac.uk` with "`join ai-cbr` *your name*" as the body of the message.

- **CBR-MED:** A mailing list for those interested in CBR for medical domains, including members of the CBR and medical communities. It is managed by Kurt Fenstermacher of the University of Chicago and Charles Kahn of Medical College of Wisconsin. To join, send a message to `listproc@cs.uchicago.edu` with "`subscribe CBR-MED` *your name*" as the body of the message.

- **CBR newsletter:** A quarterly electronic newsletter which originated as a publication of the Special Interest Group on Case-Based Reasoning (AK-CBR) in the German Society for Computer Science. It is managed by Dietmar Janetzko of the University of Freiburg and Stefan Wess of Inference Corporation. The home page for the newsletter is `http://wwwagr.informatik.uni-kl.de/~lsa/CBR/cbrNewsletter.html`.

9.2 Sites on the World Wide Web

The following sites include many references and links to other electronic CBR resources:

- David Aha at the Naval Research Laboratory maintains a site with URL `http://www.aic.nrl.navy.mil/~aha/research/case-based-reasoning.html`.
- Ralph Bergmann at the University of Kaiserslautern maintains a site with URL `http://wwwagr.informatik.uni-kl.de/~lsa/CBR/CBR-Homepage.html`.
- Ian Watson at the University of Salford maintains a site with URL `http://www.salford.ac.uk/survey/staff/IWatson/cbr01.htm`.

References

Aamodt, A. 1994. Explanation-driven case-based reasoning. In Wess, S.; Althoff, K.; and Richter, M., eds., *Topics in Case-Based Reasoning*, 274–288. Berlin: Springer Verlag.

Aamodt, A., and Plaza, E. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications* 7(1):39–52.

Acorn, T., and Walden, S. 1992. SMART: Support management automated reasoning technology for Compaq customer service. In *Proceedings of the Fourth Innovative Applications of Artificial Intelligence Conference*, 3–18. Menlo Park, CA: AAAI Press.

Aha, D., ed. 1994. *Proceedings of the AAAI-94 Workshop on Case-Based Reasoning.* Menlo Park, CA: AAAI Press. Technical Report WS-94-01.

Aha, D., ed. 1996. *Artificial Intelligence Review: Special Issue on Lazy Learning.* Boston: Kluwer. In press.

Aha, D., and Branting, K. 1995. Stratified case-based reasoning: Reusing hierarchical problem solving episodes. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 384–390. San Francisco: Morgan Kaufmann.

Aha, D.; Kibler, D.; and Albert, M. 1991. Instance-based learning algorithms. *Machine Learning* 6:37–66.

Aha, D., and Ram, A. 1995. Preface. In *Working Notes of the Fall Symposium on Adaptation of Knowledge for Reuse*, v. Cambridge, MA: AAAI.

Allemang, D. 1993. Review of the first European workshop on case based reasoning EWCBR-93. *Case-Based Reasoning Newsletter* 2(3). Electronic newsletter published by the special interest group AK-CBR of the German Society for Computer Science.

Althoff, K.-D.; Barletta, R.; Manago, M.; and Auriol, E. 1995. *A Review of Industrial Case-Based Reasoning Tools.* Oxford: AI Intelligence.

Althoff, K.-D., and Wess, S. 1991. Case-based knowledge acquisition, learning and problem solving for diagnostic real world tasks. In *Proceedings of the European Knowledge Acquisition Workshop*, 48–67.

Anick, P., and Simoudis, E., eds. 1993. *Proceedings of the 1993 AAAI spring symposium on Case-Based Reasoning and Information Retrieval.* Menlo Park, CA: AAAI Press. Technical Report WS-93-07.

Arcos, J., and Plaza, E. 1994. A reflective architecture for integrated memory-based learning and reasoning. In Wess, S.; Althoff, K.; and Richter, M., eds., *Topics in Case-Based Reasoning*, 289–300. Berlin: Springer Verlag.

Armengol, E., and Plaza, E. 1994. Integrating induction in a case-based reasoner. In *Proceedings of the Second European Workshop on Case-Based Reasoning*, 243–251.

Ashley, K. 1990. *Modeling legal argument: reasoning with cases and hypotheticals.* Cambridge: MIT Press.

Ashley, K., and Rissland, E. 1987. Compare and contrast, a test of expertise. In *Proceedings of the Sixth Annual National Conference on Artificial Intelligence*, 273–284. Menlo Park, CA: Morgan Kaufmann.

Auriol, E.; Wess, S.; Manago, M.; Althoff, K.-D.; and Traphöner, R. 1995. INRECA: A seamlessly integrated system based on inductive inference and case-based reasoning. In *Proceedings of First International Conference on Case-Based Reasoning*, 371–380. Berlin: Springer Verlag.

Bain, W. 1989. Judge. In Riesbeck, C., and Schank, R., eds., *Inside Case-Based Reasoning.* Lawrence Erlbaum. chapter 4, 93–140.

Bareiss, R. 1989. *Exemplar-Based Knowledge Acquisition: A Unified Approach to Concept Representation, Classification, and Learning.* San Diego: Academic Press.

Bareiss, R., ed. 1991. *Proceedings of the DARPA Case-Based Reasoning Workshop.* San Mateo: Morgan Kaufmann.

Barletta, R. 1994. A hybrid indexing and retrieval strategy for advisory CBR systems built with ReMind. In *Proceedings of the Second European Workshop on Case-Based Reasoning*, 49–58.

Bartsch-Spörl, B. 1995. Towards the integration of case-based, schema-based and model-based reasoning for supporting complex design tasks. In *Proceedings of First International Conference on Case-Based Reasoning*, 145–156. Berlin: Springer Verlag.

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. Hillsdale, NJ: Lawrence Erlbaum.

Bento, C. 1994. A similarity metric for retrieval of cases imperfectly explained. In Wess, S.; Althoff, K.; and Richter, M., eds., *Topics in Case-Based Reasoning*, 92–105. Berlin: Springer Verlag.

Berger, J. 1995a. *ROENTGEN: A Case-based Radiation Therapy Planner.* Ph.D. Dissertation, The University of Chicago. Computer Science Department.

Berger, J. 1995b. Using past repair episodes. Unpublished manuscript.

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

Bhatta, S., and Goel, A. 1993. Model-based learning of structural indices to design cases. In *Proceedings of the IJCAI-93 Workshop on Reuse of Design*, A1–A13. Chambery, France: IJCAI.

Bhatta, S.; Goel, A.; and Prabhakar, S. 1994. Innovation in analogical design: A model-based approach. In Gero, J., ed., *Artificial Intelligence in Design*, 57–74. Boston: Kluwer.

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., ed., *Proceedings of the DARPA Case-Based Reasoning Workshop*, 215–224. San Mateo: Morgan Kaufmann.

Börner, K. 1994. Structural similarity as guidance in case-based design. In Wess, S.; Althoff, K.; and Richter, M., eds., *Topics in Case-Based Reasoning*, 197–208. Berlin: Springer Verlag.

Branting, L. K. 1991. *Integrating rules and precedents for classificatin and explanation: Automating legal analysis.* Ph.D. Dissertation, University of Texas, Austin. Computer Science Department.

Branting, K., and Porter, B. 1991. Rules and precedents as complementary warrants. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, 3–9. Menlo Park, CA: AAAI Press.

Burstein, M. 1989. Analogy vs. CBR: The purpose of mapping. In Hammond, K., ed., *Proceedings of the DARPA Case-Based Reasoning Workshop*, 133–136. San Mateo: Morgan Kaufmann.

Bylander, T.; Allemang, D.; Tanner, C.; and Josephson, J. 1991. The computational complexity of abduction. *Artificial Intelligence* 49:25–60.

Carbonell, J. 1983. Learning by analogy: Formulating and generalizing plans from past experience. In Michalski, R.; Carbonell, J.; and Mitchell, T., eds., *Machine Learning: An Artificial Intelligence Approach*. Cambridge, MA: Tioga. 137–162.

Carbonell, J. 1986. Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. In Michalski, R.; Carbonell, J.; and Mitchell, T., eds., *Machine Learning: An Artificial Intelligence Approach*, volume 2. Los Altos, CA: Morgan Kaufmann. 371–392.

Chapman, D. 1987. Planning for conjunctive goals. *Artificial Intelligence* 32:333–377.

Cunningham, P.; Bonzano, A.; and Smyth, B. 1995. An incremental case retrieval mechanism for diagnosis. Technical Report TCD-CS-95-01, Department of Computer Science, Trinity College.

Cuthill, B. 1992. *Situation Analysis, Precedent Retrieval and Cross-Context Reminding in Case-Based Reasoning.* Ph.D. Dissertation, University of Connecticut. Technical report CSE-TR-92-3.

de Silva Garza, A. G., and Maher, M. 1996. Design by interactive exploration using memory-based techniques. *Knowledge-Based Systems* 9(1). In press.

Dean, T., and Boddy, M. 1988. An analysis of time-dependent planning. In *Proceedings of the seventh national conference on artificial intelligence*, 49–54. San Mateo, CA: Morgan Kaufmann.

DeJong, G., and Mooney, R. 1986. Explanation-based learning: An alternative view. *Machine Learning* 1(1):145–176.

Domeshek, E. 1992. *Do the Right Things: A Component Theory for Indexing Stories as Social Advice*. Ph.D. Dissertation, The Institute for the Learning Sciences, Northwestern University.

Domeshek, E.; Herndon, M.; Bennett, A.; and Kolodner, J. 1994. A case-based design aid for conceptual design of aircraft subsystems. In *Proceedings of the Tenth IEEE Conference on Artificial Intelligence for Applications*, 63–69. Washington: IEEE Computer Society Press.

Doyle, J. 1979. A truth maintenance system. *Artificial Intelligence* 12:231–272.

Fabel-Consortium. 1993. Survey of FABEL. FABEL Report 2, Gesellschaft für Mathematik und Datenverarbeitung mbH.

Faries, J., and Schlossberg, K. 1994. The effect of similarity on memory for prior problems. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, 278–282. Hillsdale, NJ: Lawrence Erlbaum.

Ferguson, W.; Bareiss, R.; Osgood, R.; and Birnbaum, L. 1991. ASK systems: An approach to story-based teaching. In *Proceedings of the 1991 International Conference on the Learning Sciences*, 158–164.

Forsythe, D., and Buchanan, B. 1989. Knowledge acquisition for expert systems: Some pitfalls and suggestions. *IEEE Transactions on Systems, Man, and Cybernetics* 19(3):435–442.

Fox, S. 1995. *Introspective Reasoning for Case-based Planning*. Ph.D. Dissertation, Indiana University. Computer Science Department.

Fox, S., and Leake, D. 1994. Using introspective reasoning to guide index refinement in case-based reasoning. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, 324–329. Hillsdale, NJ: Lawrence Erlbaum.

Fox, S., and Leake, D. 1995a. Learning to refine indexing by introspective reasoning. In *Proceedings of First International Conference on Case-Based Reasoning*, 431–440. Berlin: Springer Verlag.

Fox, S., and Leake, D. 1995b. Modeling case-based planning for repairing reasoning failures. In *Proceedings of the 1995 AAAI Spring Symposium on Representing Mental States and Mechanisms.* Menlo Park, CA: AAAI Press. Technical Report WS-95-05.

Fox, S., and Leake, D. 1995c. Using introspective reasoning to refine indexing. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 391–397. San Francisco: Morgan Kaufmann.

Garland, A., and Alterman, R. 1995. Preparation of multi-agent knowledge for reuse. In *Proceedings of the Fall Symposium on Adaptation of Knowledge for Reuse*, 26–33. AAAI.

Gentner, D., and Forbus, K. 1991. MAC/FAC: A model of similarity-based retrieval. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, 504–509. Hillsdale, NJ: Lawrence Erlbaum.

Goel, A. 1989. *Integration of case-based reasoning and model-based reasoning for adaptive design problem solving*. Ph.D. Dissertation, The Ohio State University. Computer Science Department.

Goel, A.; Ali, K.; Donnellan, M.; de Silva Garza, A.; and Callantine, T. 1994. Multistrategy adaptive path planning. *IEEE Expert* 9(6):57–65.

Golding, A. and Rosenbloom, P. 1991. Improving rule-based systems through case-based reasoning. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, 22–27. Menlo Park, CA: AAAI Press.

Goldstein, E.; Kedar, S.; and Bareiss, R. 1993. Easing the creation of a multipurpose case library. In *Proceedings of the AAAI-93 Workshop on Case-Based Reasoning*, 12–18. Menlo Park, CA: AAAI Press. Technical report WS-93-01.

Goodall, A. 1995. Supplier profile: AcknoSoft international. *AI Intelligence* 15–19. February.

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

Hammond, K., ed. 1989b. *Proceedings of the DARPA Case-Based Reasoning Workshop*. San Mateo: Morgan Kaufmann.

Hammond, K. 1993. Actualized intelligence. Invited talk, the 1993 AAAI Workshop on Case-Based Reasoning.

Haton, J.-P.; Keane, M.; and Manago, M., eds. 1995. *Advances in case-based reasoning: Second European Workshop*. Berlin: Springer Verlag.

Hinrichs, T. 1992. *Problem Solving in Open Worlds: A Case Study in Design*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hua, K., and Faltings, B. 1993. Exploring case-based design - CADRE. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing* 7(2):135–144.

Hunter, K. M. 1986. "There was this one guy:" Anecdotes in medicine. *Biology in Medicine* 29:619–630. Also appears in *Doctors' stories: The Narrative Structure of Medical Knowledge*, Princeton University Press, 1991.

Hunter, L. 1989. *Gaining Expertise Through Experience*. Ph.D. Dissertation, Yale University. Computer Science Department Technical Report 678.

Inference Corporation. 1995. Inference forms knowledge publishing division. Press release, Inference Corporation, November 17, 1995.

Jabbour, K.; Vega-Riveros, J. F.; Landsbergen, D.; and Meyer, W. 1988. ALFA: Automated load forecasting assistant. *IEEE Transactions on Power Apparatus and Systems* 3(3):908–914.

Jantke, K. 1992. Case-based learning in inductive inference. In *Proceedings of Fifth Annual ACM Workshop on Computational Learning Theory*, 218–223.

Kahn, C., and Anderson, G. 1994. Case-based reasoning and imaging procedure selection. *Investigative Radiology* 29:643–647.

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

Kass, A. 1992. Question asking, artificial intelligence, and human creativity. In Lauer, T.; Peacock, E.; and Graesser, A., eds., *Questions and Information Processing*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kass, A. 1994. TWEAKER: Adapting old explanations to new situations. In Schank, R.; Riesbeck, C.; and Kass, A., eds., *Inside Case-Based Explanation*. Lawrence Erlbaum Associates. chapter 8, 263–295.

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

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

Kennedy, A. 1995. Using a domain-independent introspection mechanism to improve memory search. In *Proceedings of the 1995 AAAI Spring Symposium on Representing Mental States and Mechanisms*, 72–78. Menlo Park, CA: AAAI Press. Technical Report WS-95-05.

Kettler, B.; Hendler, J.; Andersen, W.; and Evett, M. 1994. Massively parallel support for case-based planning. *IEEE Expert* 9(1):8–14.

Klein, G., and Calderwood, R. 1988. How do people use analogues to make decisions? In Kolodner, J., ed., *Proceedings of the DARPA Case-Based Reasoning Workshop*, 209–223. Palo Alto: DARPA.

Klein, G., and Calderwood, R. 1989. Decision models: Some lessons from the field. *IEEE Transactions on Systems, Man, and Cybernetics* 21(5):1018–1026.

Kolodner, J., ed. 1988a. *Proceedings of the DARPA Case-Based Reasoning Workshop*. San Mateo: Morgan Kaufmann.

Kolodner, J. 1988b. Retrieving events from a case memory: A parallel implementation. In Kolodner, J., ed., *Proceedings of the DARPA Case-Based Reasoning Workshop*, 233–249. San Mateo: Morgan Kaufmann.

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

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

Kolodner, J. 1994. Understanding creativity: A case-based approach. In Wess, S.; Althoff, K.; and Richter, M., eds., *Topics in Case-Based Reasoning*, 3–20. Berlin: Springer Verlag.

Kolodner, J., and Penberthy, L. 1990. A case-based approach to creativity in problem-solving. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, 978–985. Hillsdale, NJ: Lawrence Erlbaum.

Kolodner, J., and Simpson, R. 1989. The MEDIATOR: Analysis of an early case-based program. *Cognitive Science* 13(4):507–549.

Kolodner, J.; Hmelo, C.; and Narayanan, N. 1996. Problem-based learning meets case-based reasoning. In *Proceedings of the Second International Conference on the Learning Sciences*. Submitted.

Koton, P. 1988. Reasoning about evidence in causal explanations. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, 256–261. San Mateo, CA: Morgan Kaufmann.

Laird, J. E.; Rosenbloom, P. S.; and Newell, A. 1990. Chunking in Soar: The anatomy of a general learning mechanism. In Shavlik, J. W., and Dietterich, T. G., eds., *Readings in Machine Learning*. San Mateo, CA: Morgan Kaufmann. 555–572.

Lancaster, J., and Kolodner, J. 1987. Problem solving in a natural task as a function of experience. In *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, 727–736. Hillsdale, NJ: Lawrence Erlbaum.

Leake, D.; Kinley, A.; and Wilson, D. 1995. Learning to improve case adaptation by introspective reasoning and CBR. In *Proceedings of First International Conference on Case-Based Reasoning*, 229–240. Berlin: Springer Verlag.

Leake, D. 1992a. Constructive similarity assessment: Using stored cases to define new situations. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, 313–318. Hillsdale, NJ: Lawrence Erlbaum.

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

Leake, D. 1993a. Learning adaptation strategies by introspective reasoning about memory search. In *Proceedings of the AAAI-93 Workshop on Case-Based Reasoning*, 57–63. Menlo Park, CA: AAAI Press. Technical report WS-93-01.

Leake, D., ed. 1993b. *Proceedings of the AAAI-93 Workshop on Case-Based Reasoning.* Menlo Park, CA: AAAI Press. Technical Report WS-93-01.

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

Leake, D. 1994b. Workshop report: The AAAI-93 workshop on case-based reasoning. *The AI Magazine* 15(1):63–64.

Leake, D. 1995a. Abduction, experience, and goals: A model of everyday abductive explanation. *Journal of Experimental and Theoretical Artificial Intelligence* 7:407–428.

Leake, D. 1995b. Adaptive similarity assessment for case-based explanation. *International Journal of Expert Systems* 8(2):165–194.

Leake, D. 1995c. Representing self-knowledge for introspection about memory search. In *Proceedings of the 1995 AAAI Spring Symposium on Representing Mental States and Mechanisms*, 84–88. Menlo Park, CA: AAAI Press. Technical Report WS-95-05.

Macura, R., and Macura, K. 1995. MacRad: Radiology image resource with a case-based retrieval system. In *Proceedings of First International Conference on Case-Based Reasoning*, 43–54. Berlin: Springer Verlag.

Marir, F. 1995. Representing and indexing building refurbishment cases for multiple retrieval of adaptable pieces of cases. In *Proceedings of First International Conference on Case-Based Reasoning*, 55–66. Berlin: Springer Verlag.

Mitchell, M. 1993. *Analogy-Making as Perception.* Cambridge, MA: MIT Press/Bradford Books.

Mitchell, T.; Keller, R.; and Kedar-Cabelli, S. 1986. Explanation-based generalization: A unifying view. *Machine Learning* 1(1):47–80.

Moorman, K., and Ram, A. 1994. A model of creative understanding. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 74–79. Menlo Park, CA: AAAI Press.

Nguyen, T.; Czerwinski, M.; and Lee, S. 1993. Compaq QuickSource: Providing the consumer with the power of artificial intelligence. In *Proceedings of the Fifth Innovative Applications of Artificial Intelligence Conference*, 142–150. Menlo Park, CA: AAAI Press.

Oehlmann, R. 1995. Metacognitive adaptation: Regulating the plan transformation process. In *Proceedings of the Fall Symposium on Adaptation of Knowledge for Reuse*. AAAI.

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. Menlo Park, CA: AAAI Press.

Osgood, R., and Bareiss, R. 1993. Automated index generation for constructing large-scale conversational hypermedia systems. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 309–314. Menlo Park, CA: AAAI Press.

Owens, C. 1991. *Indexing and retrieving abstract planning knowledge*. Ph.D. Dissertation, Yale University.

Pirolli, P., and Anderson, J. 1985. The role of learning from examples in the acquisition of recursive programming skills. *Canadian Journal of Psychology* 39:240–272.

Portinale, L., and Torasso, P. 1995. ADAPtER: An integrated diagnostic system combining case-based and abductive reasoning. In *Proceedings of First International Conference on Case-Based Reasoning*, 277–288. Berlin: Springer Verlag.

Ram, A. 1991. A theory of questions and question asking. *The Journal of the Learning Sciences* 1(3 & 4):273–318.

Ram, A., and Cox, M. 1994. Introspective reasoning using meta-explanations for multistrategy learning. In Michalski, R., and Tecuci, G., eds., *Machine Learning: A Multistrategy Approach*. Morgan Kaufmann. 349–377.

Read, S., and Cesa, I. 1991. This reminds me of the time when ...: Expectation failures in reminding and explanation. *Journal of Experimental Social Psychology* 27:1–25.

Redmond, M. 1992. *Learning by Observing and Understanding Expert Problem Solving*. Ph.D. Dissertation, College of Computing, Georgia Institute of Technology. Technical report GIT-CC-92/43.

Richter, M. 1995. The knowledge contained in similarity measures. Invited talk, the First International Conference on Case-Based Reasoning, Sesimbra, Portugal.

Riesbeck, C. 1981. Failure-driven reminding for incremental learning. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, 115–120. Vancouver, B.C.: IJCAI.

Riesbeck, C. 1988. An interface for case-based knowledge acquisition. In Kolodner, J., ed., *Proceedings of the DARPA Case-Based Reasoning Workshop*, 312–326. San Mateo: Morgan Kaufmann.

Riesbeck, C., and Schank, R. 1989. *Inside Case-Based Reasoning*. Hillsdale, NJ: Lawrence Erlbaum.

Rissland, E., and Daniels, J. 1995. Using CBR to drive IR. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 400–407. San Francisco: Morgan Kaufmann.

Rissland, E.; Kolodner, J.; and Waltz, D. 1989. Case-based reasoning. In Hammond, K., ed., *Proceedings of the DARPA Case-Based Reasoning Workshop*, 1–13. San Mateo: Morgan Kaufmann.

Ross, B. 1984. Remindings and their effects in learning a cognitive skill. *Cognitive Psychology* 16:371–416.

Sanders, K. 1994. *Chiron: planning in an open-textured domain.* Ph.D. Dissertation, Brown University. Computer Science Department Technical Report 94-38.

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

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

Schank, R.; Fano, A.; Bell, B.; and Jona, M. 1993/1994. The design of goal-based scenarios. *The Journal of the Learning Sciences* 3(4):305–345.

Schank, R., and Kass, A. 1990. Explanations, machine learning, and creativity. In Kodratoff, Y., and Michalski, R., eds., *Machine Learning: An Artificial Intelligence Approach*, volume 3. San Mateo: Morgan Kaufmann Publishers. 31–60.

Schank, R. C., and Leake, D. 1986. Computer understanding and creativity. In Kugler, H.-J., ed., *Information Processing 86*, 335–341. North-Holland: International Federation for Information Processing.

Schank, R., 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., and Osgood, R. 1990. A content theory of memory indexing. Technical Report 2, Northwestern University Institute for the Learning Sciences.

Schank, R.; Riesbeck, C.; and Kass, A., eds. 1994. *Inside Case-Based Explanation.* Hillsdale New Jersey: Lawrence Erlbaum Associates.

Schmidt, H.; Norman, G.; and Boshuizen, H. 1990. A cognitive perspective on medical expertise: Theory and implications. *Academic Medicine* 65(10):611–621.

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

Sinha, A. 1994. Providing design assistance: A case-based approach. Artificial intelligence in management laboratory PITT-AIM-47, University of Pittsburgh.

Skalak, D., and Rissland, E. 1991. CABARET: Rule interpretation in a hybrid architecture. *International Journal of Man-Machine Studies* 34(6):839–887.

Smith, I.; Lottaz, C.; and Faltings, B. 1995. Spatial composition using cases: IDIOM. In *Proceedings of First International Conference on Case-Based Reasoning*, 88–97. Berlin: Springer Verlag.

Smyth, B., and Keane, M. 1995. Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 377–382. San Francisco: Morgan Kaufmann.

Stanfill, C., and Waltz, D. 1986. Toward memory-based reasoning. *Communications of the ACM* 29(12):1213–1228.

Stefik, M. 1981. Planning with constraints. *Artificial Intelligence* 16:111–140.

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. San Mateo: Morgan Kaufmann.

Sycara, K.; Guttal, R.; Koning, J.; Narasimhan, S.; and Navinchandra, D. 1991. CADET: a case-based synthesis tool for engineering design. *International Journal of Expert Systems* 4(2):157–188.

Sycara, K., and Navinchandra, D. 1989. Index transformation and generation for case retrieval. In Hammond, K., ed., *Proceedings of the DARPA Case-Based Reasoning Workshop*, 324–328. San Mateo: Morgan Kaufmann.

Turner, S. 1994. *The creative process: A computer model of storytelling.* Hillsdale, NJ: Lawrence Erlbaum.

Veloso, M. 1994. *Planning and Learning by Analogical Reasoning.* Berlin: Springer Verlag.

Veloso, M. and Aamodt, A., eds. 1995. *Case-Based Reasoning Research and Development: Proceedings of the First International Conference on Case-Based Reasoning.* Berlin: Springer Verlag.

Voß, A. 1994. The need for knowledge acquisition in case-based reasoning – some experiences from an architectural domain. In *Proceedings of the Eleventh European Conference on Artificial Intelligence*, 463–467. John Wiley.

Waltz, D. 1989. Is indexing used for retrieval? In Hammond, K., ed., *Proceedings of the DARPA Case-Based Reasoning Workshop*, 41–44. San Mateo: Morgan Kaufmann.

Watson, I., ed. 1995. *Progress in Case-Based Reasoning: First United Kingdom Workshop.* Berlin: Springer Verlag.

Watson, I. 1995. Case-based reasoning development tools: A review. Available on the World Wide Web at http://www.salford.ac.uk/docs/depts/survey/staff/IWatson/cbrtools.htm.

Wess, S.; Althoff, K.; and Richter, M., eds. 1994. *Topics in Case-Based Reasoning.* Berlin: Springer Verlag.

Williams, S. 1992. Putting case-based learning into context: Examples from legal, business, and medical education. *The Journal of the Learning Sciences* 2(4):367–427.