Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence

Enhancing Case Adaptation with Introspective Reasoning and Web Mining

David Leake and Jay Powell
School of Informatics and Computing, Indiana University
Lindley Hall, Bloomington, IN 47405, U.S.A.
{leake,jhpowell}@cs.indiana.edu

Abstract

Case-based problem-solving systems reason by retrieving relevant prior cases and adapting their solutions to fit new circumstances. The ability of case-based reasoning (CBR) to reason from ungeneralized episodes can benefit knowledge acquisition, but acquiring the needed case adaptation knowledge has proven challenging. This paper presents a method for alleviating this problem with just-in-time gathering of case adaptation knowledge, based on introspective reasoning and mining of Web knowledge sources. The approach combines knowledge planning with introspective reasoning to guide recovery from case adaptation failures and reinforcement learning to guide selection of knowledge sources. The failure recovery and knowledge source selection methods have been tested in three highly different domains with encouraging results. The paper closes with a discussion of limitations and future steps.

1 Introduction

Case-based reasoning (CBR) systems solve problems by reasoning from experiences. Given a problem, they generate a solution by retrieving a relevant prior case—a stored record of a previous problem-solving episode—and adapting its solution to fit the new situation (e.g., [Mantaras *et al.*, 2005]). One of the attractions of CBR as a methodology for applications is to ease knowledge capture, because records of cases may be readily available and need not be distilled into rules to be applied in CBR systems. However, applying cases to new situations requires additional knowledge for similarity assessment and case adaptation. CBR systems often implement case adaptation with adaptation rules, which may be difficult or costly to develop. Consequently, there is a tradition of CBR applications focusing on retrieval, providing users with cases to adapt themselves (e.g. [Kolodner, 1991]).

Advancing automated case adaptation requires developing methods to alleviate the knowledge acquisition problem. A number of previous projects have explored methods for learning adaptation knowledge from existing cases and other known sources [Wilke *et al.*, 1997; Patterson *et al.*, 1999; Hanney and Keane, 1997; Craw *et al.*, 2001; d'Aquin *et al.*, 2007], including some work on acquiring case knowledge on line in response to failures (e.g., [Cordier *et al.*, 2008]). This paper describes research on just-in-time mining of Internet knowledge for case adaptation, starting with a general knowledge planning process but minimal knowledge about the domain and Web sources, as implemented in the WebAdapt system [Leake and Powell, 2008; 2010a]. Just-in-time harvesting reduces effort by enabling a system to acquire only the knowledge actually needed for new problems, and also enables the system to profit from the frequent updating of community-maintained sources.

This paper begins with an introduction to the adaptation knowledge acquisition problem and the motivations for the WebAdapt approach. It then provides a high-level sketch of WebAdapt's methods. The basic framework has been augmented with two recent additions aimed at increasing the system's robustness and cross-domain effectiveness: an introspective reasoning component (cf. [Arcos *et al.*, 2008; Cox and Raja, 2008]) to diagnose and respond to failures in the adaptation process, and a reinforcement learning component to guide selection of Web sources. The paper describes evaluations of these additions which test their generality with trials on three task domains, travel itinerary planning, menu planning, and software recommendation. The paper closes by putting the project into perspective, including discussing its limitations and potential future steps.

2 Acquiring Adaptation Knowledge

Acquisition of case adaptation knowledge has long been a central challenge for CBR. Automated case adaptation may require extensive domain knowledge, but CBR is often applied precisely because general domain knowledge is difficult or expensive to capture. It may be difficult even to identify in advance what adaptation knowledge will be needed: the role of adaptation knowledge is to adapt previously-seen cases to solve problems beyond past experience.

Multiple approaches have been pursued to address the problem of acquiring case adaptation knowledge, for example, learning from knowledge already in the system's case base (see [Mantaras *et al.*, 2005] for an overview). Such methods are useful, but do not address how to enable the system to handle truly novel problems. Likewise, without constant maintenance pre-coded knowledge may miss changes that frequently occur in external knowledge sources. For example, in the travel domain, short-term museum exhibits frequently open or close.

The growing availability of knowledge on the Web provides a lazy alternative to traditional knowledge acquisition for case adaption: Instead of endowing the CBR system with adaptation knowledge, the system may be endowed with capabilities for just-in-time mining of Web knowledge sources such as Wikipedia, which themselves are continually updated. Applying this approach requires that the CBR system be able to identify its information needs, be able to determine how to satisfy them, and to update its understanding of the sources as they change.

In addition, harnessing Web sources fundamentally challenges implicit assumptions about the nature of case adaptation. Traditionally, the adaptation process is seen as finding one candidate adaptation. Given Web sources, there may be many alternatives from which to choose. The availability of these alternatives naturally introduces considerations of the fit between particular adaptations and particular users, pointing to a new area, the personalization of adaptations.

The WebAdapt project [Leake and Powell, 2007; 2008; 2010a; 2010b] aims to develop a general domain-independent framework for applying case adaptation. The project tests the ability for systems to perform adaptations starting with minimal domain knowledge, but we note that the framework would be equally applicable to augmenting internal knowledge as needed.

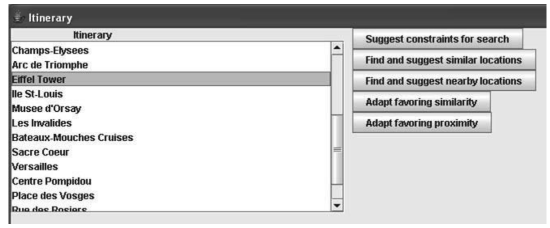

Figure 1: WebAdapt's adaptation interface.

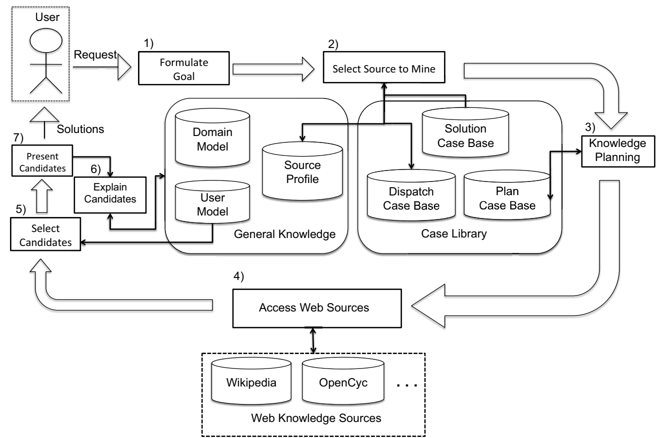

Figure 2: WebAdapt's adaptation procedure and knowledge sources.

3 The WebAdapt System and Approach

3.1 Characterizing Adaptations

A difficulty for encoding adaptation knowledge is the operationality/generality tradeoff: General rules with wide applicability provide little guidance about how to apply them to the specifics of a problem; specific rules are easy to apply but have limited applicability. Kass [1994] proposed the response of representing adaptation knowledge as the combination of general transformations (substitutions, additions, and deletions) and general pre-defined procedures for searching memory to gather specific information needed to apply them. WebAdapt's adaptation process is in a similar spirit, but replaces pre-defined memory search procedures with a flexible search process using *knowledge planning* [Ram and Hunter, 1992]. Given an adaptation problem, it generates explicit knowledge goals and plans to achieve them using operators for gathering, transforming, and applying knowledge.

3.2 WebAdapt's Basic Process

WebAdapt is designed to perform adaptations in concert with a human user. However, in contrast to the model of "leave adaptation to the user," the user's role is to identify items to adapt and to choose between proposed substitutions. To perform the adaptation, the system hypothesizes which characteristics of the items are relevant, selects Web sources to search for substitutions, generates search plans for the information, filters the results based on learned information about user preferences, presents them to the user (on request, explaining why they are considered relevant), and learns about user preferences from the user's choice. During processing the system also learns about user preferences, Web source appropriateness, and search strategies for future use.

The WebAdapt system has been tested in three domains, but the primary illustration we will use in this paper is from the travel itinerary domain. For this domain, the system begins with a case-base of travel itineraries from Frommer's guide to Paris, and adapts them by replacing sights on the itineraries based on user input. Figure 1 shows the user interface. The left side shows the list of sights in an itinerary, from which a user may select one to replace; the right side provides options to guide the replacement.

Figure 2 illustrates WebAdapt's adaptation process. The process begins when a user requests a substitution for a role-filler of a case (e.g., for "Louvre"). WebAdapt generates top-level knowledge goals for finding a substitution (step 1), describing hypothesized relevant characteristics of the existing item (the Louvre), in light of any learned information about the user's preferences. Subsequently, the system may generate subgoals in service of the top-level goal.

For example, consider the task of adapting an itinerary to find a substitution for the Louvre (e.g., for a user who has already visited it and wants an alternative). For this substitution, WebAdapt's top-level goal is to find a ranked set of candidate substitutions; lower-level knowledge goals in service of this goal include *hypothesize constraints satisfied by the Louvre*, and *find unranked candidate substitutions for it*.

WebAdapt next identifies knowledge sources to search for the desired knowledge (step 2). The system first relies on its own experience, if a similar prior goal resulted in storing a search case about a source successfully chosen for the prior search. Otherwise, it selects a source based on a learned profile of the source's previous performance.

The knowledge goal and source(s) are then input to WebAdapt's knowledge planning component (step 3). The component begins by attempting to use CBR—retrieving a prior plan satisfying the given knowledge goals. If no suitable plan is found, it applies UCPOP [Penberthy and Weld, 1992] to generate a plan from scratch, and the plan is executed to obtain Web information (step 4).

For example, if WebAdapt identifies Wikipedia as the source to mine for substitutions for "Louvre", WebAdapt's plan includes the following steps: Generate a Google query to find the Wikipedia page for "Louvre," parse the returned Wikipedia entry for links to category pages, which form a core set of candidate substitutions, and follow those links and parse the resulting pages for category-relevant information, to generate a set of potentially relevant attributes from the categories found (e.g. "Visitor attractions in Paris" and "Art museums in Paris"). For reasons of space, the knowledge planning process is not described further in this paper, but a full description is available in [Leake and Powell, 2010a].

We note that in WebAdapt's formulation of the travel planning domain, the substitutions of interest are simply keywords representing locations. For the purpose of presenting alternatives to a user, no deeper representation is necessary. However, the framework applies to more structured knowledge as well.

During processing, the system's domain model is updated with information about the categories discovered (building an internal model of the domain) and the specific substitutions discovered during the search process. When a set of candidate substitutions is displayed to the user, information about the user's selection is used to refine a model of the user. This model is then used to personalize future suggestions to focus on the types of categories of interest to the user (e.g., "art museums").

3.3 Internal and External Knowledge Sources

As illustrated in Figure 2, WebAdapt's adaptation process relies on a mixture of internal and external knowledge. Internal system knowledge includes three types of general knowledge: a domain model, a user model, and a collection of source profiles describing sources.

- *Domain model:* The domain model, built incrementally during adaptation, records the hierarchies and specific items discovered during Web search.
- *User model:* The user model records information about the categories of substitutions chosen by the user.
- *Source profiles:* For each Web source accessed by the system, a source profile captures statistical information from the system's experiences including average access times, uptimes, estimated coverage (percent of queries that result in substitutions), and a measure of diversity of results.

WebAdapt's internal knowledge also includes three case bases recording three types of experience: A dispatching case base, recording cases of choices of Web sources to access to satisfy particular queries (and their outcomes); a solution case base, recording specific prior adaptation problems and selected solutions (including derivational information); and a case base of plans used to search for particular types of adaptations. Note that there is a tradeoff between the possible efficiency gains from re-use of prior cases and the responsiveness to external updates available from searching from scratch. We discuss this tradeoff further, and propose strategies to address it, in [Leake and Powell, 2010b].

WebAdapt draws on external knowledge sources both to identify categories of items to substitute and to find specific items for substitutions. To access a source, WebAdapt requires that a small set of basic operators be defined on the source for operations such as traversing the source's abstraction hierarchy and extracting specific items. These can be defined to act on structured information (such as the knowledge in OpenCyc) or on less structured textual sources (such as Wikipedia). For example, WebAdapt's queries to Wikipedia are performed by Google queries, nodes are Wikipedia categories, and leaves are Wikipedia entries. Queries to OpenCyc are performed using the Cyc query engine; nodes are collections and leaves are individuals. Queries to the Geonames database are performed as SQL queries, where nodes are Geonames feature classes and leaves are locations. Once the basic operators have been defined, sources are navigated by the source-independent knowledge planning process.

We note that WebAdapt does not attempt to do deep natural language processing. When mining text sources such as Wikipedia, WebAdapt extracts keyword identifiers which it places in a mined concept hierarchy (e.g., finding a reference to "Sainte Chapelle" and inferring that it is an instance of a church and a historic site).

4 Introspective Failure Recovery

WebAdapt's knowledge plans may fail, for example, when the system is missing background knowledge or queries an inappropriate source. The system detects and responds to such failures with a three-step process, shown in Figure 3: (1) monitoring, (2) blame assessment, and (3) recovery. Monitoring tracks the executing plan and detects failures when solutions cannot be found (e.g., if a source is temporarily unavailable), if the knowledge sought cannot be found in the knowledge source, or if the user rejects a proposed solution.

For each failure WebAdapt generates a failure diagnosis from its reasoning trace. It uses a failure taxonomy based on Cox and Ram [1999] to categorize reasoning failures (cf., [Arcos *et al.*, 2008]). The diagnosis is provided as input for the blame assessment stage, with failure categories used to suggest points to repair, as illustrated in Table 1.

5 Learning Sources to Use

Realizing the benefits of Web access requires selecting good knowledge sources [Leake and Scherle, 2001]. Some Web sources are highly specialized; others are general-purpose. Rather than assuming pre-coded information about the types of sources to use for particular queries, WebAdapt learns two types of information for guiding source selection: (1)

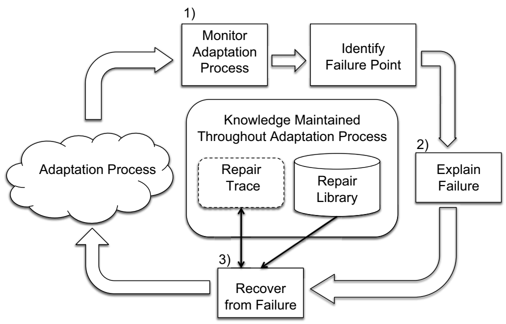

Figure 3: WebAdapt's introspective process and knowledge.

Table 1: Sample knowledge goals and possible sources of failure

Failed Knowledge goal	Failure Point(s)	Explanation
Discover item in source	Dispatching	Source choice
Hypothesize seed constraints	Dispatching	Source choice
	Planning	Strategy choice
Hypothesize candidate substitutions	Dispatching	Source choice
	Planning	Strategy choice

"dispatching cases" with records about the performance of a source for a particular query, and (2) "source profiles," collecting generalized information about source performance. Such learning has proven useful, but there is a the danger of early successes biasing later choices [Leake and Powell, 2008]. Consequently, methods are needed to balance source exploration and exploitation. We address this by adding a Q-learning component to WebAdapt's model, to guide exploration/exploitation as it determines dispatching actions.

If WebAdapt has no dispatching case for a similar knowledge goal and no source profile for a source, it selects a source at random. If it has no dispatching case but has a source profile, it relies on the source profile information. If it has both dispatching cases and source profiles, it chooses according to Q-values which are associated with the available sources and dispatching cases, as described in [Leake and Powell, 2010a].

6 Evaluation

Initial evaluations of WebAdapt's basic adaptation process were encouraging for the ability of the approach to provide reasonable substitution candidates using only Web sources and minimal initial knowledge [Leake and Powell, 2007; 2008]. Recent experiments test our extensions for introspective failure recovery and for automatically learning which sources to use. This evaluation investigates two key points:

1. *Effects of introspective failure recovery on problem-solving:* An ablation study was conducted to test how the addition of introspective reasoning affects WebAdapt's ability to solve problems.
2. *Usefulness of exploration:* How does the addition of reinforcement learning about searching external sources affect WebAdapt's ability to successfully exploit them? An ablation study was conducted to test the effects of reinforcement learning to guide source selection.

To test the generality of the approach, the experiments were conducted for five different Web sources, two generalized sources (Wikipedia and OpenCyc), and three specific sources (Geonames, USDA SR20, and an Apple Downloads library). For each source, simple procedures were coded to enable accessing the source's hierarchical structure.

6.1 Evaluation of Introspective Reasoning

The evaluation of introspective reasoning performed an ablation study with three conditions: (1) Introspective reasoning disabled, learned problem dispatching enabled, (2) Exhaustive search, and (3) Both introspective reasoning and learned problem dispatching enabled. In condition (2), WebAdapt tried each of its Web sources until a solution was found. This provided a baseline for the maximum ability of the sources to provide solutions.

Thirteen features randomly chosen from WebAdapt's cases were selected for adaptation (five itinerary items, five ingredients, and three widgets). Each item was adapted ten times using each configuration (WebAdapt picks a source at random when it has no prior dispatching knowledge). This was repeated for ten rounds of testing resulting in 520 adaptations.

For conditions (1), (2), and (3), WebAdapt was able to suggest adaptations 48%, 100%, and 100% of the time, respectively. With introspection, WebAdapt suggested substitutions 100% of the time, regardless of whether dispatching knowledge learning was enabled. In addition, configuration (3) resulted in a 76% decrease in reasoning failures over configuration (2). Thus learned dispatching combined with introspective reasoning provided the same ability to suggest an adaptation as exhaustive search, with a substantial improvement over learned problem dispatching alone.

6.2 Evaluation of Reinforcement Learning for Source Selection

The value of reinforcement learning for source selection was evaluated by comparing four conditions: (1) Exploration disabled, (2) exploration using source profile knowledge only, (3), exploration using learned source profile knowledge and learned dispatching cases, with the dispatching case library built from scratch during the experiments, and (4) exploration with a dispatching case base built from a previous round of testing on the same problems. The goal of (4) was to evaluate the impact on WebAdapt's failure rate when exploring new sources versus always choosing the sources suggested by past experience for a system which had addressed matching prior problems.

Each configuration included introspective reasoning. When case-based dispatching was included, a source which successfully solved a sufficiently similar problem in the past was always re-used in the future. Initial sources were selected at random. Fifty-two case features were randomly chosen for adaptation. Each was adapted ten times for each configuration, resulting in 2,080 adaptations.

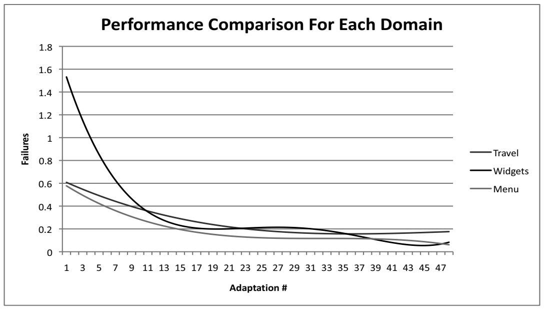

Figure 4: Reasoning failures for individual domains.

With no exploration, the system had an average of .24 failures per problem. Adding exploration based on source profiles alone substantially increases the failure rate (to .84), but exploration with learned source profile knowledge results in fewer failures (.34). Thus in this configuration, exploration is possible with modest penalty. We expect this rate to decrease as experience is gained. Using full dispatching knowledge from a prior round of testing (simulating the effects of extensive experience) reduced average failures to .07. Thus there is not a significant amount of overhead associated with learning dispatching knowledge from scratch. Introspection and learning lead to a 76% decrease in failures compared to the exhaustive search method.

6.3 Evaluation of Generality of the Approach

Because WebAdapt is intended to provide a domain-independent approach to capturing adaptation knowledge, an important question is whether its approach is equally effective across multiple domains. To test this, forty-eight features, spanning all of WebAdapt's domains, were randomly chosen for adaptation. The sequence of adaptations was performed ten times, with randomly selected initial sources, for a total of 480 adaptations. WebAdapt learned both problem dispatching and source profile knowledge from scratch during each round of testing, with introspection enabled to repair reasoning failures.

Figure 4 shows the average number of failures per feature for the travel, widget, and menu domains (with average standard deviations of 0.44, 0.53, and 0.39, respectively). The menu and travel domains can each be solved by three of the five knowledge sources and have similar failure rates. Because the Mac Widget domain can only be be solved by one knowledge source its initial failure rate is high, but source learning rapidly reduces this rate. Source selection was based on an ϵ-greedy policy. An ϵ-greedy policy selects the action that has the maximum estimated value most of the time and selects non-optimal actions with probability $1 - \epsilon$. In the experiments, the value of ϵ for a source never reached zero, resulting in occasional incorrect source choices and a final error rate of .2.

7 Limitations and Future Directions

Results to date are encouraging for WebAdapt's approach. However, many areas remain for future research. One of these is controlling the time required to mine adaptation knowledge. Web knowledge presents a vast space to mine, but in an interactive context, mining delays could seriously interfere with acceptance of the system. One approach to this problem (already implemented) is to apply case-based reasoning to the knowledge planning process, reusing prior searches and their results. In rapidly-changing domains this raises issues of the tradeoff between gaining speed from reuse and foregoing the benefits of ongoing updates to external sources. Retaining search cases also raises issues about managing the proliferation of search cases. We have begun to address this problem with methods for selective retention of search cases [Leake and Powell, 2010b]. Another approach to improving response time would be to replace the system's current sequential solution generation and presentation steps with incremental presentation of each candidate as it is found.

Another issue is how best to generate knowledge goals for adaptations. WebAdapt's automatic domain-independent method, selecting replacements belonging to categories similar to the original object, avoids the need for detailed knowledge when substitutions fall into standard categories, but does not reflect specialized needs which may not map well to standard hierarchies (e.g., if desired itinerary substitutions fall into an ad-hoc category such as "places I visited as a child."). Given descriptions of special-purpose constraints, it would be comparatively easy to integrate them into the filtering process for candidate solutions. Ideally such information would also be integrated into the original search process to guide candidate generation, but any such methods could require extensive knowledge and reasoning.

WebAdapt's adaptation process focuses on performing a single substitution at a time, with its choices based only on the item to be replaced and its user model. However, the broader context provided by a case as a whole may significantly affect the suitability of a substitution (e.g., an itinerary should avoid including two highly similar sights), and multiple substitutions may need coordination (e.g., to keep within budget, expensive activities could be balanced with inexpensive ones). Likewise, modifications to one feature in a case may necessitate adaptation of other parts of a case (e.g., to shorten one stop to compensate for increased time at another). Thus developing a framework for handling related adaptations is another area for future research.

WebAdapt currently performs minimal textual analysis of its natural language sources. Thus another extension would be to augment this capability and to exploit additional structured knowledge sources made available by the Open Data Movement, such as DBpedia (`dbpedia.org`), Freebase (`freebase.com`), and Search Monkey (`developer.yahoo.com/searchmonkey`).

8 Conclusion

The ability to adapt prior cases to new circumstances is fundamental to the flexibility of case-based reasoning systems, but may present a substantial burden for knowledge acquisition and maintenance. Enabling case-based reasoning systems to search for needed adaptation knowledge on the Web can help alleviate this burden, replacing in-advance knowledge capture with just-in-time mining of new information in the context of the current task. The WebAdapt system applies domain-independent methods for introspective reasoning and just-in-time Web mining to find personalized information for substitution adaptations.

This paper has presented the WebAdapt approach and described steps for increasing its generality, by enabling it to introspectively correct failures in its adaptation process and to improve its selection of Web sources to mine. Experiments on multiple domains support the generality of these approaches. Future steps include developing richer models of the formulation of knowledge goals and integrating the approach with more knowledge-rich methods.

Acknowledgments

The paper adapts and extends Leake & Powell 2010a, which received the Best Paper Award at Ninth International Conference on Case-Based Reasoning. It is based on work supported by the National Science Foundation under Grant No. OCI-0721674 and by a grant from the Data to Insight Center of Indiana University.

References

[Arcos *et al.*, 2008] J.-L. Arcos, O. Mulayim, and D. Leake. Using introspective reasoning to improve CBR system performance. In *Proceedings of the AAAI 2008 Workshop on Metareasoning: Thinking About Thinking*, pages 21–28, 2008.

[Cordier *et al.*, 2008] A. Cordier, B. Fuchs, L. Lana de Carvalho, J. Lieber, and A. Mille. Opportunistic acquisition of adaptation knowledge and cases - the IAKA approach. In K.D. Althoff, R. Bergmann, M. Minor, and A. Hanft, editors, *Advances in Case Based Reasoning: 9th European Conference*, Berlin, 2008. Springer Verlag.

[Cox and Raja, 2008] M. Cox and A. Raja. Metareasoning: A manifesto. Technical Memo 2028, BBN, 2008.

[Cox and Ram, 1999] M. Cox and A. Ram. Introspective multistrategy learning: On the construction of learning strategies. *Artificial Intelligence*, 112(1-2):1–55, 1999.

[Craw *et al.*, 2001] S. Craw, J. Jarmulak, and R. Rowe. Learning and applying case-based adaptation knowledge. In D.W. Aha and I. Watson, editors, *Proceedings of the Fourth International Conference on Case-Based Reasoning*, pages 131–145, Berlin, 2001. Springer Verlag.

[d'Aquin *et al.*, 2007] M. d'Aquin, F. Badra, S. Lafrogne, J. Lieber, A. Napoli, and L. Szathmary. Case base mining for adaptation knowledge acquisition. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 750–755, San Mateo, 2007. Morgan Kaufmann.

[Hanney and Keane, 1997] K. Hanney and M. Keane. The adaptation knowledge bottleneck: How to ease it by learning from cases. In *Proceedings of the Second International Conference on Case-Based Reasoning*, Berlin, 1997. Springer Verlag.

[Kass, 1994] A. Kass. Tweaker: Adapting old explanations to new situations. In R.C. Schank, C. Riesbeck, and A. Kass, editors, *Inside Case-Based Explanation*, chapter 8, pages 263–295. Lawrence Erlbaum, 1994.

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

[Leake and Powell, 2007] D. Leake and J. Powell. Mining large-scale knowledge sources for case adaptation knowledge. In R. Weber and M. Richter, editors, *Proceedings of the Seventh International Conference on Case-Based Reasoning*, pages 209–223, Berlin, 2007. Springer Verlag.

[Leake and Powell, 2008] D. Leake and J. Powell. Knowledge planning and learned personalization for Web-based case adaptation. In K.D. Althoff, R. Bergmann, M. Minor, and A. Hanft, editors, *Advances in Case Based Reasoning: 9th European Conference*, pages 284–298, Berlin, 2008. Springer Verlag.

[Leake and Powell, 2010a] D. Leake and J. Powell. A general introspective reasoning approach to web search for case adaptation. In I. Bichindaritz and S. Montani, editors, *Proceedings of the Ninth International Conference on Case-Based Reasoning*, Berlin, 2010. Springer Verlag. In press.

[Leake and Powell, 2010b] D. Leake and J. Powell. On retaining web search cases. In *WebCBR: Reasoning from Experiences on the Web (ICCBR 2010 Workshop)*, pages 87–94, 2010.

[Leake and Scherle, 2001] D. Leake and R. Scherle. Towards context-based search engine selection. In *Proceedings of the 2001 International Conference on Intelligent User Interfaces*, pages 109–112, 2001.

[Mantaras *et al.*, 2005] R. Mantaras, D. McSherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M. Maher, M. Cox, K. Forbus, M. Keane, A. Aamodt, and I. Watson. Retrieval, reuse, revision, and retention in CBR. *Knowledge Engineering Review*, 20(3), 2005.

[Patterson *et al.*, 1999] D. Patterson, S.S. Anand, W. Dubitzky, and J.G. Hughes. Towards automated case knowledge discovery in the M^2 case-based reasoning system. In *Knowledge and Information Systems: An International Journal*, pages 61–82. Springer Verlag, 1999.

[Penberthy and Weld, 1992] J. Penberthy and D. Weld. UCPOP: A sound, complete, partial order planner for ADL. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, pages 103–114. Morgan Kaufmann, 1992.

[Ram and Hunter, 1992] A. Ram and L. Hunter. The use of explicit goals for knowledge to guide inference and learning. *Applied Intelligence*, 2(1):47–73, 1992.

[Wilke *et al.*, 1997] W. Wilke, I. Vollrath, K.-D. Althoff, and R. Bergmann. A framework for learning adaptation knowledge based on knowledge light approaches. In *Proceedings of the Fifth German Workshop on Case-Based Reasoning*, pages 235–242, 1997.