Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society (FLAIRS) Conference. AAAI Press, Menlo Park, 2002, Pp. 106-110.

Managing Multiple Case Bases: Dimensions and Issues*

David B. Leake
Computer Science Department
Indiana University
Lindley Hall 215
150 S. Woodlawn Avenue
Bloomington, IN 47405, U.S.A.
leake@cs.indiana.edu

Raja Sooriamurthi
Department of Computer Science
University of West Florida
11000 University Parkway
Pensacola, FL 32514, U.S.A
sraja@cs.uwf.edu

Abstract

Case-based reasoning (CBR) models the process of reasoning from specific experiences acquired by an agent, and contained in the agent's case-base. When multiple agents acquire cases, opportunities arise for sharing their case-bases, with accompanying issues for how to apply others' experiences effectively. This paper examines issues for multi-case-base reasoning (MCBR), the reasoning process needed for a CBR system to exploit external case-bases reflecting similar but different tasks and task environments. The paper summarizes the component processes required, the dimensions along which these processes may differ, and some of the key research issues that must be addressed for successful MCBR systems. It closes with a perspective on the relationships of case-based reasoning and multi-case-base reasoning, examining the analogy between reasoning about cases in CBR and case-bases in MCBR.

Introduction

Case bases have long been recognized as a valuable knowledge resource, and as a potential medium for knowledge sharing (Inference 1995). Case-based reasoning research, however, has focused largely on the issues involved in managing a single, task-dedicated case base. Research in distributed CBR has examined issues in drawing on well-standardized external case bases, and case-base maintenance research has examined how to improve case-base standardization. However, increasing numbers of fielded systems promise growing numbers of nonstandardized case bases, for different (but related) tasks, which may not be practical to standardize or merge. Consequently, exploiting this resource may require reasoning about issues such as when to draw on particular case-bases, which case-bases are most appropriate to access for a given task, and how to revise solutions according to the case-bases from which they were derived. Such reasoning processes form the heart of *multi-case-base reasoning* (MCBR) (Leake & Sooriamurthi 2001).

* This research is supported in part by NASA under award No NCC 2-1216.

MCBR is a framework in which the basic CBR processes are augmented with facilities for *problem dispatching*, in which problems may selectively be assigned to external case-bases, and *cross-case-base adaptation*, in which proposed solutions are revised according to the properties of the case-bases that suggested them. In this paper we examine the architectures required for such systems, the issues that arise in their design, and the dimensions for addressing them. The goals of the paper are twofold: To examine the nature of MCBR systems, and to point to crucial issues for future research.

Motivations

Moving from single-case-base to multiple-case-base models may be worthwhile both to augment local case information, and to facilitate management of information that is naturally available from different sources with differing task or environmental characteristics. Specific motivations may include:

- *Increasing efficiency and coverage:* Multiple smaller case bases may increase retrieval efficiency or make storage requirements more manageable.
- *Easing maintenance:* Accessing cases from unmerged local sources, rather than merging information into a single case-base, automatically captures local case updates and facilitates responses to case-base relationship changes.
- *The benefits of least-commitment processing:* Potential case uses may be too hard to anticipate to standardize in advance.
- *The benefits of retaining implicit information:* Important task and environment characteristics may be implicit in case collections.
- *Exploiting information available only on demand:* If case bases become commercial knowledge resources, different sources may have exclusive rights that prevent merging. Thus MCBR may be necessary to exploit the growing availability of distributed web-based information sources.

The Basic Framework

In standard case-based reasoning, the reasoning system draws on a single case-base of experiences. In our previous work on multi-case-base reasoning (Leake & Sooriamurthi 2001), we considered situations in which a CBR system can augment its local case-base by drawing on a single external case-base when needed. However, in general an MCBR system could have many case-bases to choose from. For example, a system to predict product prices might draw on the catalogs of multiple stores; an on-line travel guide could draw on repositories of traveler comments published by many different groups, etc. Because those case-bases may have differing coverage characteristics, a multi-case-base reasoner may need to reason about the case-base from which to draw cases. Likewise, because different case-bases may reflect different problem circumstances, a multi-case-base reasoner may need to adapt cases based on their sources in addition to responding to differences in the stated problem descriptions. Thus any MCBR system requires components to perform four basic functions:

1. *Problem dispatching:* The dispatching component determines the set of case-bases to which a problem is sent. This may be based on criteria such as estimated likelihood that the case-base will contain a relevant case, expected speed of response, or cost of requesting information from a commercial case source.
2. *Case selection:* The selection component determines which cases to consider as candidates for contributing to the solution. For example, selection might return the first case retrieved; might wait for all queried case-bases to respond and return the most similar case, or all cases within a set similarity threshold; or might return cases from all case-bases that respond within a given time limit.
3. *Solution merging:* The solution merging component determines how the solutions from the set of selected cases will be combined into the final solution. For example, for a numerical prediction task, numerical values might be averaged; for a planning task, portions of the retrieved plans might be merged according to preference criteria.
4. *Cross-case-base adaptation:* When case-bases deal with differing tasks or task environments, the solutions that they suggest may need additional adjustment in response to those differences (e.g., a case-base of products might return prices in dollars when the desired prices are in euros). The cross-case-base adaptation component adjusts solutions for these inter-case-base differences.

Architectures for MCBR

One possible MCBR architecture is the simple "fallback" architecture, shown in Figure 1. In this architecture, a CBR system draws on external cases when it determines that its own case library is insufficient. Here dispatching could be based on the level of similarity between current problem and most similar local case; case selection based on whether the external case actually addresses a more similar problem, and cross-case-base adaptation based on sampling the external case-base to estimate inter-case-base solution differences.

More generally, an MCBR system may draw on any number of case-bases, accessed according to strategic criteria. Figure 2 shows one such architecture. When a problem is input to the MCBR system, a dispatcher determines case bases to query and a strategy for how to pursue the query

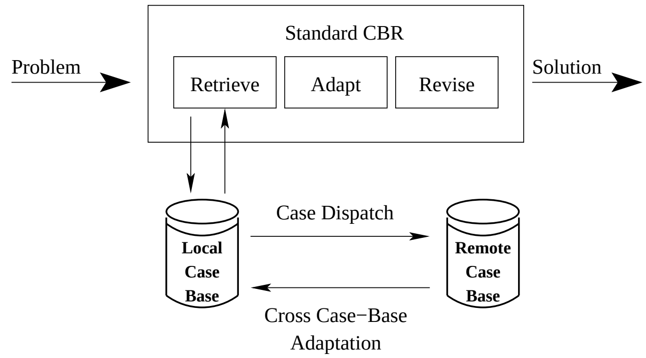

Figure 1: Drawing on an external case-base when the local case-base is insufficient.

- Case-base characterization
- Problem dispatching
- Case selection
- Solution merging
- Cross-case-base adaptation
- Multi-case-base maintenance

Table 1: Issues for MCBR design.

sequence. The system then selects results to consider, performs cross-case-base adaptation to adjust for inter-case-base differences, and merges solutions if multiple solutions will be used. (Depending on the system strategy, these steps may be ordered in different ways.) The result is then passed on to a standard CBR process.

We note that this general architecture includes the previous architecture as a special case. A "local" case-base need not be distinguished from other case-bases, except as encoded by the dispatching strategy. For example, a local case-base would presumably have very low access cost, causing it to be favored by a cost-based dispatching strategy, and would require only the identity function for cross-case-based adaptation, causing it to be favored by a dispatching strategy favoring cases for similar problems.

Issues for Multi-Case-Base Reasoning

Applying the MCBR architecture requires addressing both the standard CBR issues—for processing of the cases that are retrieved—and addressing the new issues that arise from multiple case bases, as summarized in Table 1. The following sections summarize each of these issues and highlight the dimensions of how they may be addressed.

Case-Base Characterization

Just as case indexing plays a crucial role in CBR, case-base characterization is vital to MCBR. As for case indexing, case-base characterization must provide the information needed to estimate the usefulness of the case-base. Because

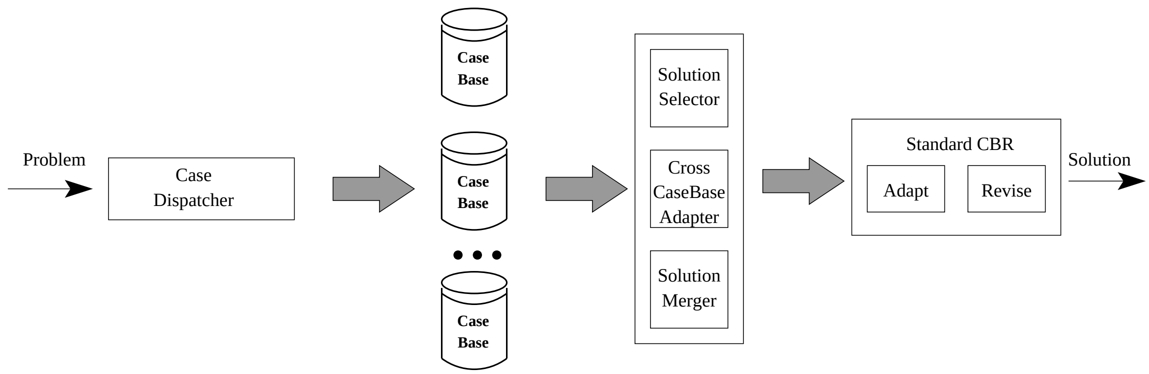

Figure 2: Multi-case-base reasoning framework for drawing on a set of case-bases.

the question is the relevance of the case-base, rather than of a case, the information must be predictive of the likelihood of the case-base to contain cases addressing a particular problem. This includes *coverage information* expressing the likelihood of the case-base containing cases for the problem at hand. At a coarse-grained level, the characterization might include global competence information; at a finer-grained level, it might include descriptions of more specific expertise. Likewise, it includes *task information*, to compare to the task at hand to estimate the difficulty of cross-case-base adaptation. In addition, because different case-bases have different access characteristics (in terms of how to formulate queries, availability, expected response time, etc.), it must include *access characteristics*.

Problem Dispatching

Dispatching policies use case-base characterizations to determine which case-base(s) should be used to solve a particular problem. For purposes of dispatching, no distinction is made between the local case-base and external case-bases: Choices of when to use the local case-base are made according to the same types of reasoning processes used to decide when to access an external case-base. In this way, the dispatching policies simultaneously address the questions of "when to dispatch" and "where to dispatch."

The dispatcher's decision-making may depend on arbitrarily sophisticated reasoning. It may be based on any aspects of the problem to solve (e.g., using the characterization to decide which case-base is most relevant), on the overarching system task and constraints (e.g., needs for rapid response), and on information about external constraints and circumstances (e.g., that an overburdened external case-base should be accessed only when necessary). The dispatching process may be targeted at retrieving a single case, or a set of cases; multiple retrievals are resolved during the selection/merging phase.

Dispatching policies fall into four possible categories:

- *One-shot dispatching:* Closest to traditional CBR, dispatching may simply choose to dispatch a problem to one or more of the external case-bases, for the results to be processed directly by the case selection/merging phase.
- *Sequential dispatching:* Sequential dispatching prioritizes dispatching targets and determines a sequence of queries, based on the sequence of results, to implement an information-seeking strategy. For example, a simple two-step policy might solve problems locally if the local case-base has a sufficiently similar case, and otherwise send the problem to a remote case-base.
- *Parallel dispatching:* Parallel dispatching broadcasts requests simultaneously to a set of case-bases.
- *Hybrid dispatching:* Hybrid dispatching methods involve a sequence of dispatches, combining parallel and sequential dispatching steps.

If an MCBR system is drawing on the case-bases of other MCBR systems, we may imagine a federation of communicating systems. In such systems, another dimension is *dispatcher integration*—the level of centralization or information sharing between individual MCBR systems. For example, to illustrate two extremes, multiple systems might be designed to communicate with a central registry of available case-bases and their characteristics, or each maintaining their own dispatching information.

Another issue, arising from the perspective of the case-bases being queried, is *query acceptance*. With sufficient reasoning capability and rich-enough queries, a case-base might assess its own willingness to process a query and competence in the query area, possibly to itself dispatch the problem to another source.

Case Selection and Solution Merging

Because an MCBR system may send simultaneous queries to multiple case-bases, policies are needed for how to select the returned cases to consider and to merge their suggested solutions. A first-pass selection policy might simply use the first case that is returned; a somewhat more subtle approach might accumulate cases with a cut-off policy (e.g., ignoring future returned cases when a certain number of cases has been retrieved, or when a case of sufficient quality has been retrieved). The availability of multiple cases also presents opportunities for applying ensemble methods to multiple solutions.

The selection/merging process becomes more subtle when the application of a prior case begins before retrievals are complete. In that situation, later retrievals may provide useful information to shape or even replace the current reasoning process. This merging requires reasoning not only about the usefulness of a case, but how much it contributes beyond the current system state. For example, even if much effort has been expended on a candidate solution, it may be appropriate to discard that work if a complete prior solution has been found.

Cross-Case-Base Adaptation

In order for an MCBR system to effectively use case-bases that may have been developed in different ways, for different tasks or task environments, methods are needed to adjust retrieved cases for local needs. The cross-case-base adaptation process adapts suggested solutions from one case base to apply to the needs of another. It corresponds to the case adaptation process for a CBR system, but with the difference that its role is to adapt in response to the differences in sources of cases, rather than differences between the current and prior problems. Those problem-based differences are accounted for by the standard CBR process, after cross-case-base adaptation has produced a case compatible with the local CBR system. Each pairing of a CBR system with a destination case-base may require different adaptation strategies.

Cross-case-base adaptation may need to address syntactic or semantic representational differences, as well as differences in tasks and task environments that simply affect attribute values. In order to select appropriate cross-case-base adaptation strategies, a system must have (or be able to derive) information about the relationships between solutions in its own case-base and the external case-base. To enable this, a fundamental issue is how to derive the needed metadata. We are currently exploring sampling methods for comparing case-base characteristics in order to select appropriate cross-case-base adaptation strategies.

Multi-Case-Base Maintenance

Traditional case-base maintenance focuses on issues of individual case bases, such as how to standardize case-base contents or compact the case-base. MCBR must address these issues for individual case-bases, as well as additional issues that arise in the MCBR context:

- *Standardization:* Case-base standardization can function as an “eager” analogue to cross-case-base adaptation: It can adjust for systematic differences that otherwise would need to be addressed as cases are applied.

- *Split and reform:* MCBR maintenance adds the choice of how many case-bases to use and their makeup. Splitting a large case-base into smaller, task-focussed case-bases may enable more efficient retrievals; Merging two case-bases may yield increased breadth, as coverage increases, or depth, as multiple case-bases become available for the same part of the case-base.

Analogies to CBR: The Continuum from Reasoning about Cases to Reasoning about Case-Bases

The previous sections highlight the operations that MCBR systems require, beyond those required for standard CBR. Operations such as case-base indexing, and cross-case-base adaptation, may be viewed as simply the standard CBR processes, applied to a case-base of case-bases. Given this correspondence, we may view a CBR system as an MCBR system, in which each of the MCBR system’s case-bases contains a single case, and the MCBR system’s dispatching process corresponds to normal case retrieval. Likewise, any MCBR system can be transformed into a standard CBR system by merging its cases. Thus we can view CBR and MCBR as falling on a continuum. At one extreme, there is a single, unified case-base; at the other, there is a different case-base for every domain case. Processing of both endpoints in the continuum is isomorphic, with case indexing in CBR corresponding to case-base indexing for the singleton case-bases. The MCBR split and reform maintenance operations provide a way to move along this continuum.

The principles of MCBR can also apply to multiple retrievals from a single case-base. For example, (Riesbeck 1996) proposed a CBR model in which processing begins based on an initial retrieval, with retrieval processes continuing in case more appropriate case-bases are available. This model corresponds to MCBR with multiple retrievals from the same case-base, and utility-based case solution merging.

In terms of maintenance, MCBR’s cross-case-base adaptation can be seen as a form of “lazy” case standardization: Cases are converted to the form of the local system as needed. (Leake & Wilson 1999) discusses an analog to MCBR’s splitting operation, applied to standard CBR: Identifying “hot spots” in the problem space, based on current problems, to selectively place cases less likely to be used in a secondary case-base if storage capacity is limited or utility problems interfere with retrieval efficiency.

Related Research

Methods for managing sharing of standardized case-bases have been studied in research on distributed CBR (e.g., (Martin, Plaza, & Arcos 1999)), as have methods for facilitating large-scale case distribution (Hayes, Cunningham, & Doyle 1998). However, these methods do not address the cross-case-base adaptation needed for non-standardized cases. Relevant to dispatching issues, (McGinty & Smyth 2001) highlight the value of case-base specific expertise in drawing on external case-bases, and describe a system that dispatches cases to provide personalized recommendations.

The use of multiple case-bases in MCBR has a number of analogs in the database community, in research areas such as component database systems (Dittrich & Geppert 2001), and in studies of management issues for heterogeneous database systems (Elmagarmid, Rusinkiewicz, & Sheth 1999). How to address structural and syntactic differences between heterogeneous database systems has been investigated extensively in research on interoperability and schema integration. This work, and AI research in areas such as learning to translate between ontologies and data integration (e.g., (Domingos 2000)), are likely to prove highly relevant to cross-case-base adaptation issues.

Conclusion

Multi-case-base reasoning augments CBR with capabilities for reasoning about which case bases to use, and how to use them. Its selective, lazy access of nonstandardized cases can provide a useful supplement to a local case base, and the framework it requires provides a uniform way to access distributed case resources. The MCBR framework can also be applied to combine a number of CBR systems, each generating and drawing on shared case resources.

At a high level, many of the processes needed for MCBR parallel those of the normal CBR process, showing the applicability of lessons from CBR, but many specific multi-case-base issues arise as well. These point to new research areas and new opportunities for intelligent sharing and reuse of case knowledge from multiple sources.

References

Dittrich, K., and Geppert, A. 2001. *Component Database Systems*. Morgan Kaufmann.

Domingos, P. 2000. Data integration: A "killer app" for multi-strategy learning. In *Proceedings of the Fifth International Workshop on Multistrategy Learning*. Menlo Park, CA: AAAI Press.

Elmagarmid, A.; Rusinkiewicz, M.; and Sheth, A. 1999. *Management of Hetergenous and Autonomous Database Systems*. Morgan Kaufmann.

Hayes, C.; Cunningham, P.; and Doyle, M. 1998. Distributed CBR using XML. In *Proceedings of the KI-98 Workshop on Intelligent Systems and Electronic Commerce*.

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

Leake, D., and Sooriamurthi, R. 2001. When two case bases are better than one: Exploiting multiple case bases. In *Case-Based Reasoning Research and Development: Proceedings of the Fourth International Conference on Case-Based Reasoning, ICCBR-01*. Berlin: Springer-Verlag.

Leake, D., and Wilson, D. 1999. When experience is wrong: Examining CBR for changing tasks and environments. In *Proceedings of the Third International Conference on Case-Based Reasoning*, 218–232. Berlin: Springer Verlag.

Martin, F.; Plaza, E.; and Arcos, J.-L. 1999. Knowledge and experience reuse through communications among competent (peer) agents. *International Journal of Software Engineering and Knowledge Engineering* 9(3):319–341.

McGinty, L., and Smyth, B. 2001. Collaborative case-based reasoning: Applications in personalised route planning. In *Case-Based Reasoning Research and Development: Proceedings of the Fourth International Conference on Case-Based Reasoning*. Berlin: Springer Verlag.

Riesbeck, C. 1996. What next? The future of CBR in postmodern AI. In Leake, D., ed., *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. Menlo Park, CA: AAAI Press. 371–388.