Supporting Conversational Case-Based Reasoning in an Integrated Reasoning Framework

David W. Aha, Leonard A. Breslow, and Tucker Maney
Navy Center for Applied Research in Artificial Intelligence
Naval Research Laboratory
Washington, DC 20375
lastname@aic.nrl.navy.mil

Abstract

Conversational case-based reasoning (CCBR) has been successfully used to assist in case retrieval tasks. However, behavioral limitations of CCBR motivate the search for integrations with other reasoning approaches. This paper briefly describes our group's ongoing efforts towards enhancing the inferencing behaviors of a conversational case-based reasoning development tool named NaCoDAE. In particular, we focus on integrating NaCoDAE with machine learning, model-based reasoning, and generative planning modules. This paper defines CCBR, briefly summarizes the integrations, and explains how they enhance the overall system.

Conversational Case-Based Reasoning

Our research focuses on enhancing the performance of conversational case-based reasoning (CCBR) systems (Aha & Breslow, 1997). CCBR is a form of case-based reasoning where users initiate problem solving *conversations* by entering an initial *problem description* in natural language text. This text is assumed to be a partial rather than a complete problem description. The CCBR system then assists in eliciting refinements of this description and in suggesting solutions. Its primary purpose is to provide a focus of attention for the user so as to quickly provide a solution(s) for their problem. Figure 1 summarizes the CCBR problem solving cycle.

Cases in a CCBR library have three components:

1. *Summary*: A brief textual description of the case.
2. *State*: A set of ⟨question,answer⟩ pairs.
3. *Solution*: A sequence of actions for responding to this state.

The text entered by the user is matched against the summary in each stored case. The reasoner then ranks the cases using a similarity function and identifies the top-ranking k cases. The solutions of these cases are displayed to the user, along with the top-ranking q unanswered questions in the states of these cases. Example case and question ranking functions are described in (Aha & Breslow, 1997). The display sizes k and q are user-defined parameters.

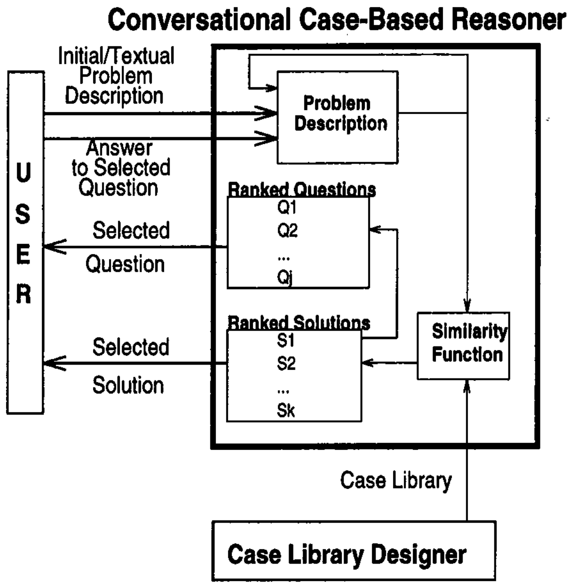

Figure 1: Problem Solving Using Conversational Case-Based Reasoning

The user can select one of the displayed questions to answer. This ⟨q,a⟩ pair will be added to the concept description, and another similarity computation will update the case rankings and, subsequently, the contents of the two displays. Alternatively, the user can select one of the displayed solutions to apply to their problem.

System performance can be measured by retrieval *precision*, defined as the relative frequency with which a selected solution solves a given problem, and retrieval *efficiency*, defined as an inverse function of the number of questions answered before a solution is retrieved.

The CCBR approach is, by far, the most successfully deployed style of case-based reasoning (Watson, 1997), primarily because it was targeted for a niche market that it fits well: interactive help-desk and WWW diagnostic tasks.[1] However, commercial CCBR tools cur-

[1] In particular, Inference Corporation has licensed their CCBR tool tool to over a half million end users involved with over 650 corporate contracts.

rently have limitations:

1. *Case Authoring*: Authoring CCBR libraries is difficult, and incurs a steep learning curve.
2. *Dialogue Inferencing*: CCBR Conversations are limited in their ability to derive inferences from problem descriptions.
3. *Applicability*: They are limited to problems that can be represented as case retrieval tasks.

A few researchers have focused on integrating additional reasoning capabilities in CCBR systems. Shimazu et al. (1994) describe three types of user interfaces for eliciting information from the user. Trott and Leng (1997) briefly mentioned how an integer programming approach can be used to induce question weights, but their approach is restricted to highly engineered case libraries. Racine and Yang (1997) describe promising techniques for case-based maintenance, but these do not require additional inferencing capabilities. Several publications describe applications of this approach; perhaps Nguyen et al.'s (1993) award-winning paper is the best known among these. We include a discussion of our own research on CCBR integrations in the next section.

This paper describes how the three limitations listed above can be addressed by integrating CCBR with alternative reasoning methods. In particular, we summarize ways in which machine learning approaches can be used to facilitate case authoring, how models can be used to enhance dialogue inferencing, and how CCBR can assist a generative planner. These integrations are being developed in the context of our CCBR tool, named NACoDAE (Navy Conversational Decision Aids Environment) (Breslow & Aha, 1997).

Multimodal Reasoning Integrations

This section briefly describes how CCBR tools can benefit from integrations with learning, model-based reasoning, and generative planning components.

Learning

There are several motivations for integrating machine learning strategies with conversational case-based reasoners. We detail one role and then outline others.

CCBR tools can be successfully used for some applications, but they do not eliminate the need for systematic knowledge acquisition. In particular, claims that using case-based approaches simplifies the knowledge elicitation process (i.e., because experts can more easily provide cases than rules) are misleading. While it is frequently true that cases can be more easily obtained from experts, this does not mean that they can be easily authored so as to ensure good CCBR performance.

The reason that case authoring is difficult for CCBR libraries is that cases are typically small in size (i.e., having fewer than eight ⟨question,answer⟩ pairs in their states), and two arbitrary cases may have few or no questions in common. Thus, some reliance is necessary to ensure that textual case descriptions assist in

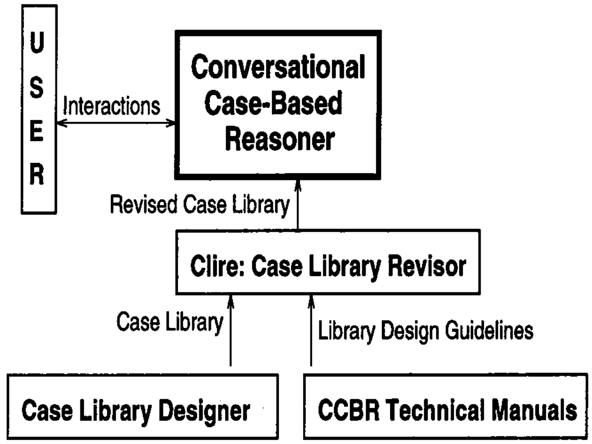

Figure 2: CLIRE Revises Case Libraries to Enforce Guidelines

case matching. Furthermore, care is needed in selecting questions, and in distinguishing cases by using the same question(s) but different answers. This task's complexity increases with case library size.

To address this case authoring problem, CCBR vendors provide guidelines on how to author case libraries. Unfortunately, these guidelines are numerous and conflicting; they are difficult to master. Thus, after creating a (typically) poor-performing case library, customers tend to either seek additional consulting assistance or abandon the use of CCBR entirely.

We instead suggested using a machine learning approach to assist customers (Aha & Breslow, 1997). In particular, we introduced the following general approach for enforcing guidelines in an existing case library:

1. *Hierarchy Creation*: Create a structured hierarchy from the case library.
2. *Editing*: Modify this hierarchy to enforce specific library design guidelines.
3. *Case Extraction*: Extract a modified set of cases from the edited hierarchy.

We implemented this approach in CLIRE (Case Library Revisor), which uses a novel learning algorithm to induce a decision tree in the first step. CLIRE's relationship with NACoDAE is summarized in Figure 2. Aha and Breslow (1997) reported that CLIRE performed well on a set of case libraries: NACoDAE's retrieval precision and efficiency both increased when using the CLIRE-transformed libraries.

Several other machine learning integrations could prove useful for CCBR tools. For example, ⟨q,a⟩ pairs in case states are often annotated with weights, similar to feature weights, so as to bias the similarity function. A machine learning approach could be used to automatically tune these weights, as has been done for classification and planning tasks (Wettschereck et al., 1997; Muñoz-Avila & Hüllen, 1996). This would further simplify the case authoring task, allowing users to ignore the challenging problem of assigning weights manually.

Learning could also be used to model user behavior. For example, it could be used to determine whether users are experts or novices, and invoke different sets of questions appropriately. Alternatively, learned user models could bias the ranking of case solutions according to past problem solving interactions. Other learning opportunities could also be identified.

Model-Based Reasoning

One problem with the CCBR approach is that no capability is provided for deriving inferences from a partial problem description. Thus, questions cannot be answered from the user's text, nor from other answered questions. For example, using a fairly well-designed and paradigmatic case library for troubleshooting computer printer problems, a user might enter the phrase "black streaks," and the system might respond by displaying the questions "What does the print quality look like?" and "Are you having print quality problems?" If the CCBR tool was designed to perform dialogue inferencing, then it could automatically answer these questions with "black streaks" and "yes" respectively.

One solution to this problem is to allow users to insert rules that specify these relationships (i.e., if the text includes "influence diagrams," then answer "What topic?" with "Decision theory"). However, even for small case libraries, the number of rules that must be inserted can be huge because several synonymous phrases can exist for domain objects and several questions can be interrelated.

An alternative solution involves integrating NaCoDAE with a model-based reasoning component (Aha & Maney, 1998). The idea is to interactively assist the user in creating a model of the case library domain, which highlights the domain objects, their relationships, and how the library's questions relate to these objects. Models are represented using graph structures, and questions are answered by determining the bindings for variables in specific subgraph structures.

Figure 3 summarizes this approach. This integration involves the following modules:

1. *Text Pre-Processor*: This transforms the user's textual problem description into a canonical form.
2. *Model Builder*: This interactively builds the library model.
3. *Rule Generator*: This yields *text rules*, which relate text to ⟨q,a⟩ pairs, and *implication rules*, which define implications between ⟨q,a⟩ pairs.
4. *Parka-DB*: This is a fast relational querying system (Hendler et al., 1996). It inputs a *knowledge base* (i.e., set of binary assertions) and a query (i.e., "Which new answers can be derived given this problem description?"). It outputs a set of newly implied answers to previously unanswered questions.

The benefits of this approach derive from the library model's comparably compact size. It will be easier to

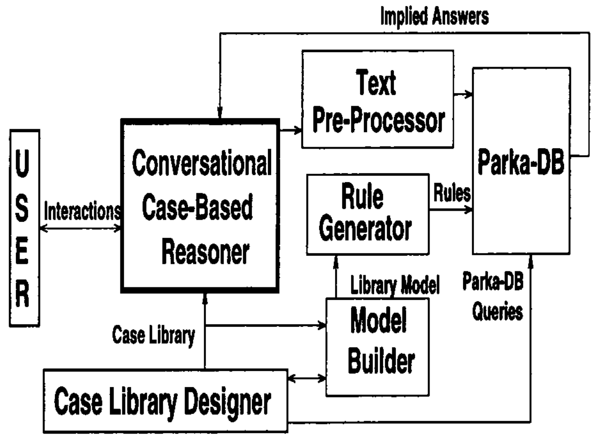

Figure 3: Model-Based Dialogue Inferencing for CCBR

maintain than a set of unrelated rules; when the domain evolves (e.g., as new cases are added), it will be simpler to edit the model rather than to determine which rules should be edited, and how to edit them. The drawback is that the user must supply information on how to construct the model, but this information can be obtained in a semi-automated fashion by prompting the user for information derived from a text analysis of case descriptions and questions.

Initial results with this integration have been promising; answers are automatically derived as users would expect from implications in the problem description. In summary, this integration demonstrates a way in which a model-based component, supported by text processing and relational querying systems, can improve the quality of CCBR conversations.

Generative Planning

Several researchers have argued that generative planning algorithms can be made more robust by integrating them with a case-based reasoning component (Veloso, 1992; Branting & Aha, 1995). In particular, planning effort can be reduced by retrieving and adapting plans stored in cases, assuming that the combined effort of retrieval and adaptation is less than that of generative planning.

To our knowledge, no research effort has focused on *conversational* case-based planning. We anticipate a need for conversational variants of hybrid case-based and generative planners when (1) user interaction is required and (2) the user cannot provide a complete problem description (i.e., initial state) without the system's assistance.

Our efforts to develop a conversational case-based planner are driven by crisis response planning and scheduling tasks. These tasks are characterized by limited resource availabilities, non-independent goals, and the fact that actions can affect other actions. We are currently investigating a hybrid planning approach with the following behavior:

- Planning is separated from and precedes scheduling.
- The planning step yields a plan tree, where interior nodes define planning goals and leaves identify resources required to satisfy those goals.
- Cases are annotated with the goal(s) that their solutions address.
- Users interact by conducting a conversation that leads to the selection of a case for expanding an interior node, where the goal of the selected case's plan must match the selected node's goal.
- State information can be collected from both users and available sensors. Answers to some questions involve interaction with question-specific interfaces (e.g., point and click maps).

When either the user or system determines that no stored case suffices for a particular goal (e.g., due to insufficient similarity), the generative planning algorithm is tasked with expanding the selected node. Cases can be derived from the results of generative planning, corresponding to either a set of (possibly ordered) child nodes or a more elaborate plan subtree.

Two aspects of decision analysis should later prove useful for this extension of NACoDAE. First, question ranking can be profitably guided by ranking questions according to their expected utilities, which will be a combination of evaluation cost and the expected information gain from their answers (e.g., to further distinguish the top-ranking cases). Second, we plan to investigate a representation for cases that includes a set of *monitors* for determining whether the preconditions of a planning step are dynamically violated. Decision theory can be used to model uncertainty in whether these violations require replanning and/or rescheduling.

Status and Future Research

CLIRE, the case library revisor, has been implemented and systematically evaluated (see Aha & Breslow, 1997). Future research goals include designing it to enforce library guidelines as libraries are being created, and to investigate other opportunities for applying machine learning techniques in CCBR.

The dialogue inferencing tool has been partially implemented, and we are currently integrating it with NACoDAE. We recently reported its initial empirical evaluation, demonstrating that this approach improves retrieval efficiency without sacrificing retrieval precision for two case libraries (Aha et al., 1998). We also plan to evaluate the time required to build case library models and the additional computational overhead required to derive implied answers (which we expect will be minimal, due to Parka-DB's high efficiency).

We have developed and implemented a first version of our conversational case-based planner, and applied it to a task concerning the development of plans for responding to hazard materials spills and fires (Gervasio et al., 1998). However, its evaluation has not yet been completed. Also, we plan to address other crisis response planning tasks (e.g., for planning noncombatant evacuation operations), and will integrate it with Gervasio et al.'s non-conversational case-based planning system.

Acknowledgments

This paper describes joint planning research with Jim Ballas. This research is supported by a grant from the Office of Naval Research and a Cooperative Research and Development Agreement with Inference Corporation. NaCoDAE, implemented in Java, can be obtained from the first author.

References

Aha, D. W., & Breslow, L. A. (1997). Refining conversational case libraries. *Proceedings of the Second International Conference on Case-Based Reasoning* (pp. 267–278). Providence, RI: Springer.

Aha, D. W., & Maney, T. (1998). A model-based approach for supporting dialogue inferencing in a conversational case-based reasoner. To appear in *Proceedings of the AAAI-98 Spring Symposium on Multimodal Reasoning*. Menlo Park, CA: AAAI Press.

Aha, D. W., Maney, T., & Breslow, L. A. (1998). *Supporting Dialogue Inferencing in Conversational Case-Based Reasoning* (Technical Report AIC-98-008). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.

Branting, L. K., & Aha, D. W. (1995). Stratified case-based reasoning: Reusing hierarchical problem solving episodes. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 384–390). Montreal, Canada: Morgan Kaufmann.

Breslow, L., & Aha, D. W. (1997). *NaCoDAE: Navy Conversational Decision Aids Environment* (Technical Report AIC-97-018). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.

Gervasio, M. T., Iba, W., & Langley, P. (1998). Case-based seeding for an interactive crisis response assistant. To appear in D.W. Aha & J.J. Daniels (Eds.) *Case-Based Reasoning Integrations: Papers from the 1998 Workshop* (Technical Report WS-98-02). Menlo Park, CA: AAAI Press.

Hendler, J., Stoffel, K., & Taylor, M. (1996). *Advances in high performance knowledge representation* (Technical Report CS-TR-3672). College Park, MD: University of Maryland, Department of Computer Science.

Muñoz-Avila, H., & Hüllen, J. (1996). Feature weighting by explaining case-based planning episodes. *Proceedings of the Third European Workshop on Case-Based Reasoning* (pp. 280–294). Lausanne, Switzerland: Springer.

Nguyen, T., Czerwinskai, M., & Lee, D. (1993). COMPAQ QuickSource: Providing the consumer with the power of artificial intelligence. In *Proceedings of the Fifth Annual Conference on Innovative Applications of*

Artificial Intelligence (pp. 142–150). Washington, DC: AAAI Press.

Racine, K., & Yang, Q. (1997). Maintaining unstructured case bases. *Proceedings of the Second International Conference on Case-Based Reasoning* (pp. 553–564). Providence, RI: Springer.

Shimazu, H., Shibata, A., & Nihei, K. (1994). Case-based retrieval interface adapted to customer-initiated dialogues in help desk operations. *Proceedings of the Twelfth National Conference on Artificial Intelligence* (pp. 513–518). Seattle, WA: AAAI Press.

Trott, J. R., & Leng, B. (1997). An engineering approach for troubleshooting case bases. *Proceedings of the Second International Conference on Case-Based Reasoning* (pp. 178–189). Providence, RI: Springer.

Veloso, M. M. (1992). *Learning by analogical reasoning in general problem solving.* Doctoral dissertation, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Watson, I. (1997). Applying Case-Based Reasoning: Techniques for Enterprise Systems. San Francisco: Morgan Kaufmann.

Wettschereck, D., Aha, D. W. & Mohri, T. (1997). A review and empirical comparison of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review, 11*(1–5), 273–314.

Appendix

1. **Integration name**: NaCoDAE
2. **Performance Task**
   Machine learning: Retrieval precision and conversational efficiency
   Model-Based Reasoning: Same
   Generative Planning: Reactive planning
3. **Integration Objective**
   Machine learning: Revise case libraries
   Model-Based Reasoning: Deduce implied answers to questions
   Generative Planning: Generate a tree of actions and required resources
4. **Reasoning Components**
   Machine learning: Decision tree inducer, editor, and case extractor
   Model-Based Reasoning: Query-retrieval system (PARKA-DB)
   Generative Planning: Multiple CCBR agents
5. **Control Architecture**
   Machine learning: ML performed off-line
   Model-Based Reasoning: CBR as master
   Generative Planning: CBR directed by planning goals
6. **CBR Cycle Step(s) Supported**: Retrieval, reuse (in planning)
7. **Representations**
   Machine learning: Cases temporarily represented in a decision tree
   Model-Based Reasoning: Model represented as a semantic net
   Generative Planning: Tree hierarchy of actions/goals and resources
8. **Additional Components**: Scheduler for generative planning task
9. **Integration Status**
   Machine learning: Completed (Aha & Breslow, 1997)
   Model-Based Reasoning: Initial evaluation completed (Aha et al., 1998)
   Generative Planning: Implemented, informally applied
10. **Priority future work**
    Machine learning: Address other learning tasks
    Model-Based Reasoning: Complete empirical evaluation, formal analysis, implement the interactive model constructing module
    Generative Planning: Evaluation, integration with scheduler