Using Logic to Reason with Cases[1]

Kevin D. Ashley and Vincent Aleven
University of Pittsburgh, Pittsburgh, Pennsylvania 15260
ashley@vms.cis.pitt.edu, (412) 648-1495, 624-7039

Before people can decide whether to rely on the advice provided by a Case-Based Reasoning (CBR) system, they must understand the criteria according to which the system asserts that a case is relevant to a problem and that it is more relevant than other cases. In our application, tutoring students to reason with cases, our intelligent tutoring system, CATO, needs to explain its relevance criteria and illustrate them with examples. Moreover, it needs to deal with a variety of relevance criteria, some of which involve relations among multiple cases. The system designers, and eventually, teachers and students, need to be able to understand, use and modify the program's concepts for assessing case relevance and constructing case-based arguments. A logical representation of the relevance criteria provides the expressiveness and flexibility to make that possible.

At an operational level, a CBR program needs to compute the relevance of cases. Although in most CBR programs to date, relevance concepts have been operationalized by structuring a program's memory (e.g., as a discrimination net) and building procedures to sort, select and filter cases (e.g., see Koton's CASEY and Sycara's MEDIATOR), such concepts can be expressed in first-order logic and implemented by a deductive pattern-matcher. In developing CATO, an intelligent tutoring system that teaches law students to argue with cases, we have employed the knowledge representation system Loom to represent relevance and argument concepts declarativelly in logic expressions.

The declarative logical representation offers important advantages for our CBR tutoring application, allowing us to: (1) Specify relevance criteria in terms of relationships among multiple cases, (2) Deal with multiple relevance criteria, (3) Communicate relevance criteria and illustrate them with examples, and (4) Support user queries of the case base. To insure that the declarative representation did not lead to computational inefficiency, we have conducted a series of experiments to measure the efficiency of our queries using synthetic case bases that vary in size from 26 to 250. We have found that the costs of representing relevance concepts logically have not been prohibitive.

The primary significance of this work, however, has to do with explanation in CBR systems. We have needed to deal with relevance criteria in ways that have been relatively unusual in CBR work so far, but which we anticipate will become increasingly useful for CBR beyond our tutoring application. The CBR community has not adequately addressed the question: how can case-based reasoners explain their reasoning and convince users of the plausibility of the system's conclusions? There are at least five ways:

1. Show the user a similar precedent. Such an explanation may involve mapping and adapting an explanation from the precedent to the problem as in SWALE, CASEY, GREBE or integrating the precedent into a rule-based explanation as in CABARET. The precedent, however, is only part of the warrant represented by the case comparison.
2. Some CBR programs, like CASEY, can justify why the precedent matches the original.
3. In addition, a program could explain why the particular precedent is a better match then other candidates (HYPO);
4. In addition, the program could explain its criterion for justifying the match or for considering one case to be better than another.
5. In addition, the program could explain why the criterion matters in terms of the theory of the domain and task.

This work on CATO focuses on the third and fourth methods. By representing relevance criteria declaratively, we have made some progress in enabling a program to explain aspects of its relevance criteria by example. As CBR system designers confront the problem of building programs that can explain and justify their results, we believe that a declarative logical representation of relevance concepts will be useful.

[1]This work is supported by an NSF Presidential Young Investigator Award and grants from the National Center for Automated Information Research and Digital Equipment Corporation.