Case Base Management Systems: Providing Database Support to Case-based Reasoners

Radha Mahapatra and Arun Sen
Texas A&M University

Case-based reasoning offers a new approach for developing knowledge-based systems. Unlike the rule-based paradigm, in which domain knowledge is encoded in the form of production rules, in the case-based approach the problem solving experience of the domain expert is captured by cases. These cases are stored in a knowledge base, called a casebase. Current case-based systems are mostly research prototypes (programming-in-the-small), designed primarily to refine the concepts in case-based reasoning. A number of research issues need to be resolved to facilitate the transition of these prototypes to large application systems (programming-in-the-large). The primary issue that will help the transition is to develop data management support for these prototypes. In this paper, we analyze this data management support and propose a new concept called a casebase management system to perform data management functions for case-based systems. We discuss the research issues that should be addressed to develop casebase management systems.

Case-based reasoning offers a novel approach to develop knowledge-based systems. Unlike the rule-based paradigm, in which domain knowledge is encoded in the form of rules, in the case-based approach the expert’s problem solving experience is captured in the form of cases which are stored in a casebase. Case-based systems solve problems by reusing problem solving experience stored in cases. These systems are especially suitable for domains with the following characteristics [Kolodner & Riesbeck, 1990]:

- problems are ill defined and/or incompletely specified;
- knowledge used for problem solving is obscure;
- it is difficult to formulate domain rules, but examples are easily available;
- rule-based processing is expensive because of large number of rules; and,
- problems have large sets of possible solutions.

These domain characteristics make the case-based approach very attractive for developing knowledge-based solutions to problems that are not amenable to rule-based problem solving.

Case-based reasoners (CBRs) have been developed in several application domains including legal reasoning [Ashley 1987, Bain 1986], planning [Hammond 1989], classification [Bareiss 1989], design [Hinrichs 1991], model formulation [Vellore et al. 1993], and business planning [Sullivan & Yates 1988]. Most of the CBRs developed so far follow the programming-in-the-small philosophy of systems development. These are small, single user systems developed as research prototypes. The primary objective of these systems is to develop and refine tools to support case-based reasoning, and to establish case-based approach as a viable method for developing knowledge-based systems in a wide range of application domains. As the technology moves from the research labs into real-life business applications, there will be a need to develop large systems that will support multiple applications accessed by multiple users (programming-in-the-large). A number of new problems will be encountered as the technology moves from the research labs into real-life business applications. One of these is the need to provide data management support to case-based systems. In this paper we identify this data management need of case-based systems and propose a new concept, called a casebase management system (CBMS), that will fulfill such needs. We outline the research issues that need to be addressed to design a CBMS.
An Overview of Case-Based Reasoning

In business organizations and in our day-to-day lives, we often come across problems that have some degree of resemblance with problems encountered and solved earlier. In such situations, decision makers usually recall and make use of their past problem solving experience to solve new problems. This process of problem solving is called case-based reasoning [Riesbeck & Schank 1989, Kolodner 1991]. A case-based system incorporates a computational model of case-based reasoning in its problem solving process. Since mid-eighties, the Artificial Intelligence (AI) community is very much involved in developing and refining CBRs.

The Mechanism of Case-based Reasoning

A CBR solves problem by reusing past problem solving experience stored in the form of cases in its casebase. The basic control flow in such a system is shown in Figure 1. The casebase is a repository of cases. Cases typically have two major components: (1) the problem description, and (2) the solution description. The problem description captures the features of the problem relevant for decision making. The solution description captures the decision made, or the solution steps used to solve the problem. To facilitate case retrieval, they are indexed by problem features and solution features relevant for decision making. An analysis of the problem solving process in several case-based systems [Riesbeck & Schank 1989, Kolodner & Riesbeck 1990] reveals the following six major steps:

- step-1. Problem analysis and feature extraction,
- step-2. Case retrieval,
- step-3. Base case selection,
- step-4. Solution adaptation,
- step-5. Solution evaluation, and

In the problem analysis step, the input problem is analyzed and features relevant for decision making are extracted. These are then used in the case retrieval phase for retrieving similar cases from the casebase. Case retrieval is based on feature matching. Cases with features that partially or completely match the features of the input problem are retrieved from the casebase. There are three possible outcomes of this phase: (a) no matching case is found in the casebase; (b) a single matching case is found; and (c) more than one matching cases are found. If no matching case is found in the casebase, then the system has no prior experience of solving problems similar to the input problem. The CBR will not be able to solve the problem and the processing will terminate with failure. This signals the inadequacy of the casebase and a need to enhance it with more cases. If a single matching case is found, that case will be used

![Figure 1: The Basic Control Flow in a CBR](image-url)