Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1995 Proceedings

Americas Conference on Information Systems (AMCIS)

8-25-1995

Enhancing Data Management Support for Case-based Reasoning Systems

Radha Mahapatra
University of Alabama in Huntsville, rmahapat@asb1.asb.uah.edu

Arun Sen
Texas A&M University

Enhancing Data Management Support for Case-based Reasoning Systems

(An Extended Abstract)

Radha Mahapatra
email:
rmahapat@asb1.asb.uah.edu
Ph: (205) 895-6857

Department of MIS/MSC
College of Administrative Science
University of Alabama in Huntsville
Huntsville, AL 35899

Arun Sen
sen@tamvm1.tamu.edu

Business Analysis & Research
College of Business Administration and Graduate School of Business
Texas A&M University
College Station, TX 77843-4217

Introduction

Case-based reasoning offers a novel approach to develop knowledge based systems. A case-based system (CBS) stores problem solving expertise as cases in its casebase. A case captures a problem description and the description of a solution to the problem. A CBS solves a problem by starting with an approximate solution found in a case in its casebase. When presented with a problem, a CBS analyzes it to extract salient features relevant for problem solving. It searches the casebase to identify cases with similar features. All such cases are retrieved and compared with the problem to select the best matching case. The solution in the best case is adapted to develop a solution to the problem. The proposed solution is evaluated. A new case is formed by combining the problem with the proposed solution. This case, if found suitable, is stored in the casebase. A CBS, thus, augments its casebase with new cases as it solves new problems.

Case-based reasoning has been used in a wide range of application domains to develop problem solving and advisory systems. A limitation of these systems is that they lack adequate data management support for casebases. Most current CBS are small memory-resident systems. They use small casebases, which are loaded into primary memory during processing. This limits the size of the casebase and restricts the scope of the CBS. Since a CBS develops a solution by starting with an approximate solution from its casebase, its problem solving ability depends to a great extent on the variety and number of cases available in its casebase. It is more likely to find a closely matching case for a given problem in a large casebase compared to that in a smaller casebase. A CBS, therefore, needs a large casebase to operate at an acceptable level of expertise. As a CBS solves new problems, it adds new cases to its casebase. Thus the casebase keeps growing with the daily use of the system. A major research issue confronting CBS research is how to create large systems that can handle large casebases comprising hundreds and

thousands of cases (Kolodner 1993). Our research addresses this important issue of providing data management support to large casebases.

Research Issues

Following the database research paradigm we have proposed to store the casebase in secondary memory A Casebase Management System (CBMS) provides data management functions for this casebase. Instead of loading the whole casebase to primary memory during processing, only cases relevant to a problem are retrieved from the casebase during problem solving. While a secondary memory provides significantly large storage space compared to a primary memory, with regard to data access time the former is considerably slower compared to the latter. Therefore, to implement the casebase in secondary memory, it is necessary to ensure efficient retrieval of cases from the casebase so that the overall system response time is reasonable. Clustering and indexing are used as standard techniques in Database Management Systems (DBMS) to improve data retrieval time. Similar strategies are needed for a CBMS.

Clustering is the process of storing frequently co-referenced data records in the same page. This reduces the number of pages to be retrieved from the secondary memory to access a set of data records and results in faster data retrieval. Relational systems cluster tuples of different relations that frequently participate in joint operations (Astrahan et al. 1976). Nested relational systems cluster components of a nested relational record (Schek et al. 1990). Object-oriented databases use clustering to improve object retrieval time (Kim et al. 1987, Deux et al. 1990). Clustering techniques used in DBMS exploit schema related information to perform clustering. These techniques are not appropriate for clustering cases, because cases must be clustered based on inter-case similarity. Case retrieval is based on the similarity between a query and the cases stored in the casebase. A multidimensional approach is used for measuring similarity (Kolodner 1993). The match between a case and a query is measured along each dimension relevant for a problem domain. This information is combined with the significance level of each dimension to compute the overall degree of similarity between a query and a case. All cases with a similarity above a threshold are retrieved from the casebase. This notion of similarity and similarity-based retrieval is not found in databases.

DBMS maintain secondary indexes to facilitate associative access to data records. Indexes are maintained on attributes that frequently appear in user queries. These are used to speed up query processing in large databases (Ullman 1988). There is a major difference between query execution on a database and that on a casebase. A database query retrieves records that exactly match with the attribute values specified in the query. In the absence of such a match, a database query would not return any record. Execution of a case query returns partially matched cases when an exact matching case is not found. A partially matched case is better than no case, because a CBS fails to solve a problem if it is unable to locate a case that can provide an approximate solution. Identifying partially matched cases is an essential functionality of a CBMS and must be supported by its indexing system. Partial matching is performed using domain knowledge. This knowledge may change with the acquisition of new knowledge. The indexing system in a

CBMS must be designed to capture the domain knowledge required for performing partial match. It should be flexible to accommodate changes to this knowledge.

Results

A similarity-based clustering algorithm has been developed to cluster cases based on inter-case similarity. This algorithm determines the most appropriate placement of a new case in secondary memory. It tries to place a new case in the same physical page that stores its best matching case. If a case cannot be placed in its target page because of inadequate space, the cases in the page are split into two groups while ensuring that each group holds more similar cases. The two groups of cases are written to two different pages. Thus, the algorithm ensures that more similar cases are placed in the same page. The case retrieval performance is improved because fewer data pages need to be accessed by the system during a retrieval operation. A *feature tree* is used to capture domain knowledge required for identifying partially matched cases. It can be modified to reflect changes in domain knowledge. Indexes are maintained on the *feature tree* to facilitate retrieval of partially matched cases.

A prototype CBMS called *alpha* has been developed using C language in a UNIX environment on a SunSparc Station. It supports storage and retrieval operations for cases in a casebase which resides in secondary memory. The case insertion function in *alpha* uses the case-clustering algorithm to cluster cases. The index management system in *alpha* implements the *feature tree* and supports partial matching of cases. The case retrieval function is designed to retrieve similar cases for a query. The performance of the system has been evaluated and found to be satisfactory. The performance results are reported.

Conclusion

This paper proposes the use of secondary memory based persistent casebases to scale up case-based systems. This will eliminate the restriction imposed on the size of the casebase by the main memory of a computer system and will facilitate sharing of the casebase by multiple reasoners. It presents a similarity-based clustering algorithm to improve case retrieval time. An indexing system is proposed to facilitate retrieval of partially matched cases. These clustering and indexing strategies have been implemented in a prototype system. Results from the performance evaluation of this system demonstrates the effectiveness of the clustering and indexing strategies. *alpha* is the first version of a prototype CBMS. Currently it operates in a single user mode and provides case storage and retrieval functions to case-based reasoners. Our long term plan is to enhance it with multi-user support features and to connect it to multiple case-based reasoners in a client-server mode. This paper addresses an important research issue in case-based reasoning research. It also makes a contribution to database research by creating a specialized database system for case management.

References

Astrahan, M.M. et al. (1976),"System R : Relational Approach to Database Management," *ACM TODS*,1(2), June 1976, pp. 97-137.

Deux, O. et al. (1990),"The Story of O2," *IEEE KDE* , 2(1), pp. 91-108.

Kim, W., Banerjee, J., Chou, H.,Garza, J.F., Woelk, D. (1987),"Composite Object Support in an Object-Oriented Database System," *OOPSLA Proceedings*, pp. 118-125.

Kolodner, J.L. (1993), *Case-based Reasoning*, Morgan Kaufmann, San Mateo, CA.

Schek, H.-J., Paul, H.-B., Scholl, M.H., Weikum, G. (1990),"The DASDBS Project: Objectives, Experiences, and Future Prospects," *IEEE KDE*, 2(1), March 1990, pp. 25-43.

Ullman, J.D. (1988), *Principles of Database and Knowledgebase Systems*, Volume I, Computer Science Press, Rockville, MD.