IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB7284

Chapter XX

Organizing Multimedia Objects Using a Class Algebra Database

Daniel J. Buehrer
National Chung Cheng University, Taiwan, R.O.C.

This chapter considers the question of how to locate appropriate multimedia objects. The traditional use of Web search engines is limited by the fact that there are as of yet no standard XML tags describing the many various types of multimedia objects. The users would much prefer to choose from menus, especially from a classification hierarchy where subclasses are all listed. Traditional SQL relational databases are generally not flexible enough to permit dynamic addition of attribute/value pairs to describe these multimedia objects, and to display the resulting schema. The solution suggested in this chapter is the sharing of class and binary relation definitions, including the binary superclass/subclass relation of the IS-A classification hierarchy of object-oriented technology. This object-oriented database implementation is based on class algebra. Similar users can see each other's objects, classes, and relations, and add their own objects into appropriate places in both the class hierarchy and the binary relations. The binary relations look like subdirectories, and the available binary relations and methods can be chosen from a menu. The distributed database system, therefore, looks like a typical network operating system, and it is expected that average users may be able to make database queries and updates without any need of programmers.

INTRODUCTION

Multimedia will include online versions of the entertainment industry, the publishing industry, the advertising industry, and the education industry, among others. Obviously, it will affect a major portion of our lives. To date, however, bandwidth limitations have prevented the widespread use of interactive television, games, and educational material.

This chapter appears in the book, Distributed Multimedia Databases: Techniques and Applications by Timothy K. Shih.

So far, only MP3 music has been widely distributed by peer-to-peer file sharing programs such as Napster. Now that Napster is joining major entertainment companies such as Bertelsmann, the third largest media conglomerate, there is little doubt that its peer-to-peer file sharing techniques will enable distribution of other kinds of multimedia material to far more than its current 38 million users (Gibney, 2000). Similar programs, such as Aimster (http://www.aimster.com/), will also be used to distribute other kinds of multimedia among friends. However, one key question is how to find and distribute "suitable" multimedia and software. There are many possibilities, some of which may be worth the membership fee that these companies will soon be charging. Of course, there are also many other services besides search that the membership fee will cover. But the key service is the ability to quickly download cached versions of material that should be of interest within a group of similar users.

A distributed class algebra database provides one answer for locating relevant material. It allows similar users to share their IS-A classification hierarchies and other binary relations, along with the multimedia files and their applications. Users can import other users' classifications, and they can make their own classifications based on a Boolean algebra of classes. They can also make use of shared binary relations for creating common models of reality. Finally, they can develop and share Java applications for classifying, searching, editing, renting, buying/selling, evaluating, and chatting about multimedia, or playing distributed interactive games. Unlike most current C++ object-oriented databases, these applications will be cross-platform, executable on any CPU and OS with a single click.

To see an example of how distributed class algebra databases can help to share multimedia, we will describe how to use the Abia Cadabia database (A Basic Interface with Audio for a Class Algebra DAtaBase Inference Agent) (Chen, Lo, Hsieh, Hou and Buehrer, 2000; Buehrer and Lee, 1999; Buehrer, 1995; Buehrer, Liu, Hong, and Jou, 1996; Buehrer, 1994). This database is capable of sharing editors, viewers, games, and other software as well as the multimedia itself. Perhaps more importantly, however, it can also union together many users' classification hierarchies and binary relations.

Even though the database is implemented in Java, which has single inheritance, it supports the multiple inheritance of class algebra. The IS-A hierarchy is a directed, acyclic graph. Although different users may have different classifications of objects, based on different attributes, these classification graphs are unioned together when one user logs into other users' databases. Besides unioning the IS-A superclass/subclass relation, other users' relations are also unioned together if they have the same relation name.

The Abia user interface for the Cadabia database is basically a binary relation editor, and a user can easily define new subclasses or new binary relations to either his own objects or other users' objects. Following a binary relation is much like opening a subdirectory. The relations may either be retrieved by using the Abia menu-driven interface, or they may be retrieved by client applets via a convenient API that allows class algebra queries to be included within Java programs, similar to the way SQL statements are included within database applications.

The focus of this chapter is on the usefulness of such a distributed superclass/subclass relation and other distributed binary relations. We hope to show the advantages of class algebra (and distributed object-oriented databases, in general) over relational algebra and SQL databases, and give several examples of how class algebra queries can easily retrieve the desired kinds of multimedia, including multimedia that has been reviewed by various evaluation techniques, including human reviewers. An example is given of how reviewers can earn credits for their evaluations, and how reviewers themselves can be evaluated, being paid different credits based on the accuracy of their evaluations.