The MacWrap Project: Using an Object-Relational DBMS for an Open Multimedia Database System

Wolfgang Lindner *Andreas Heuer*

Dept. of Computer Science,
University of Rostock
D-18051 Rostock, Germany
{wl, ah}@informatik.uni-rostock.de

Abstract

A large quantity of data and documents in the world are not stored in database management systems. There are many specialized systems emerging to store and search for particular data types, including image and video management systems, etc. However, multimedia digital library systems e.g., benefit not only from combining information from these various systems, but also from using metadata (structured data), in addition to multimedia documents in one system. In our approach, we combine different sources providing multimedia documents and structured data by using an object-relational DBMS. We use a Multimedia-Compensator-Wrapper architecture to integrate specialized servers into the ORDBMS. In this paper, we present an architectural overview and address its benefits for realizing an open multimedia database system. Therefore, we do not address all the issues concerning multimedia DBMSs e.g., content-based retrieval or indexing of multimedia data.

1 Introduction

Today, much research is dedicated to the development of systems that store, retrieve, or process multimedia data or structured data. However, much less effort is invested in interrelating specialized systems, like media servers or databases, in order to benefit not only from combining information, but also from using metadata (structured data), in addition to multimedia documents in one system. More specifically, the collaboration of database management systems (DBMSs) and media servers would be especially useful because traditional DBMSs are widely used and can help efficiently manage structured data. Furthermore, media servers are also commonly used and can provide efficient support for media data management. However, the integrated management of standard structured, as well as media data (e.g., time-dependent data), is most desirable, but it is still too complex of a task to be covered by a single system and may, in fact, never work out. Therefore, coupling distinct systems, each doing only what it can do best, seems to be a better way of building a multimedia database management system.

The intention of this paper is to introduce a new and efficient method of building a complex multimedia database architecture based on commercially available database as well as existing legacy systems, respectively media servers. We use an object-relational database management system as a "federation" platform. ORDBMSs have the advantage of providing full DBMS services such as transactions, query languages, and optimizers. On the other hand, they are "open" in the sense that different data or document types can be integrated as so-called "database extenders" (DB2) [3], "data blades" (Informix), or "data cartridges" (Oracle). Instead of integrating database data and multimedia documents or different types on top of database systems, we use a Multimedia-Compensator-Wrapper (MacWrap) architecture to use the extender or blade technology for integration. While the wrapper only polishes the interfaces, parameters, and data types, the compensator has to implement some services not provided by the data or document source (like transaction information or specific query operations).

2 Related Work

The Garlic project [2] presents an approach to build a multimedia information system capable of integrating data that resides in different database systems as well as in a variety of non-database servers. To federate the different sources, Garlic uses a so-called middleware architecture. With this approach, "all the work" is done in the middleware [5], or so to say, "on top" of the database system. A query engine accesses the different sources, transforms data, and optimizes and executes queries. But what about using the database system for some of this work ?

Compared to Garlic and other projects (like AMOS [9], FMDBS [1], etc.), which also follow the middleware approach, we are introducing a new and promising way to build a MMDBS. In our approach we "extend" an existing database system in such a way so that different media servers, as well as other databases, can be combined (integrated) to form a complex multimedia information system. This allows the usage of database capabilities the system provides (e.g. the built-in optimizer), eliminates redundant functionality, and hides the heterogeneous environment.

3 System Architecture

The main component is the compensator where all the subsystems will be plugged-in as shown in Fig. 1. The users can query the ORDBS which then accesses the different data sources registered with the compensator transparently as needed. The core of the global schema is the multimedia standard of SQL99 [6]. The schema can be dynamically extended by user defined functions (UDF) and user defined types (UDT) of plugged-in media servers.

3.1 Database System

The database is an object-relational Database System. Besides the database capabilities, it supports the definition and usage of UDF and UDT and serves two purposes. It is used as the main database, i.e. as a working database for the clients and also to store the required metadata of all connected systems.

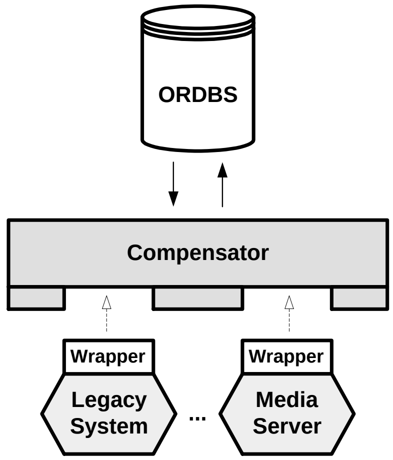

Figure 1: Compensator Approach

3.2 Compensator

The compensator is basically a main extender that is able to federate different subsystems while maintaining the independence of the systems and without creating copies of their data. Depending on the scope of database capabilities the subsystems are able to provide, the compensator has to compensate more or less. This means, if a media server supports some sort of query capabilities or transactional control, the compensator takes advantage of it. If the subsystem e.g. does not support transaction control, the compensator will provide it for any interaction with the database system. In summary, the functions of the compensator are to:

- register all subsystems dynamically and store their metadata such as UDT, UDF, and costs in the database,
- interrelate data from different sources into a single query and provide some optimization in compliance with various quality of service requirements,
- provide transactional control for the subsystems, if needed,
- support real-time audio/video streaming as well as interaction with such continuous data [8].

This allows e.g., the combination of different image extenders in the compensator as demonstrated in the middleware of the HERON-project [7].

3.3 Wrappers

Wrappers are needed to abstract the different media server interfaces to a common interface, the compensator. This allows the database to store or retrieve data or just use special functions from the media servers. Wrappers are system specific and don't provide any additional functions or features beyond the scope of the subsystems.

3.4 Subsystems

Before using a subsystem, it has to be registered with the compensator. This means the compensator has to know what kind of UDT and UDF it supports and how expensive each function is. Subsystems can vary from a plain storage system or specialized application running as a demon, up to a multimedia information system, like an audio/video server with real-time capability. Even a database system can serve as a plug-in for this architecture as introduced in the project PowerDB [4].

4 Experience and Future Work

We implemented a first prototype of our MacWrap architecture based on an object-relational DBMS, DB2. The disadvantages of the extender or data blade technology available up to now are the lack of adequate development tools and the black-box approach of these kinds of database extensions. For a scenario like MacWrap, it is much easier to implement a compensator if an extender can be used within another extender. For example, an XML extender could then be based on a video extender (realizing video documents within an XML metadata description).

On the other hand, we currently develop a model for the Multimedia-Compensator-Wrapper architecture that is based on object types and methods of a formal object-relational database model to semi-automatically derive compensating functions within a framework similar to SQL-99. Using MacWrap to support realtime aspects of multimedia servers is described in [8].

References

[1] H. Berthold. Design of a Federated Multimedia Database System. In *Proc. 11. GI-Workshop "Grundlagen von Datenbanken", 25.-28. Mai, 1999, Luisenthal, Germany*, 1999.

[2] M. Carey, L. Haas, and et al. Towards Heterogeneous Multimedia Information Systems: The Garlic Approach. In *RIDE-DOM, Taiwan, 3/95*, 1995.

[3] D. Chamberlin. *A Complete Guide to DB2 Universal Database.* Morgan Kaufmann, San Francisco, 1998.

[4] T. Grabs, K. Böhm, and H.-J. Schek. A Document Engine on a DB Cluster. In *High Performance Transaction Processing Systems Workshop USA 9/99*, 1999.

[5] L. Haas and et al. Transforming Herterogeneous Data with Database Middleware: Beyond Integration. In *Data Engineering Bulletin*, 1999.

[6] *ANSI/ISO/IEC International Standard (IS) Database Language SQL – Part 2: Foundation (SQL/Foundation), ISO/IEC 9075-2:1999 (E)*, September 1999.

[7] W. Kiessling, K. Erber-Urch, W.-T. Balke, T. Birke, and M. Wagner. The HERON Project - Multimedia Database Support for History and Human Sciences. In *28. German Computer Society Conference, Magdeburg, 9/98*, 1998.

[8] W. Lindner, H. Berthold, F. Binkowski, A. Heuer, and K. Meyer-Wegener. Enabling Hypermedia Videos in MMDBS coupled with Realtime Media Servers. In *2000 International Database Engineering and Applications Symposium, IDEAS 2000, 18 - 20 September, 2000, Yokohama, Japan, Proceedings. IEEE Computer Society*, 2000.

[9] T. Rakow, W. Klas, and E. Neuhold. Research on Multimedia Database Systems at GMD-IPSI. In *IEEE Multimedia Newsletter, Vol. 4, No. 1, pp. 41-46, 4/96*, 1996.