



The 6th World Multiconference  
on Systemics, Cybernetics  
and Informatics

July 14-18, 2002  
Orlando, Florida, USA

**PROCEEDINGS**

Volume VII

Information Systems Development II

Organized by IIIS



International  
Institute of  
Informatics and  
Systemics

Member of  
International Federation of  
Systems Research IFSR

EDITED BY  
Nagib Callaos  
John Porter  
Naphtali Rishe

# Semantic Wrapping Tool for Internet \*

Naphtali RISHE, Tao HUANG, Maxim CHEKMASOV, Scott GRAHAM, Li YANG, Sheldon HIMMELSBACH

High-performance Database Research Center  
School of Computer Science  
Florida International University  
Miami, FL 33199, U.S.A.

## ABSTRACT

Semantic Wrapper is a set of wrapping tools that provides relational databases with a semantic interface based on the Semantic Binary Object-oriented Data Model (Sem-ODM), which is superior to other data models in expressive power and ease of use. This paper presents the system architecture of a Web application for accessing relational databases through the Internet via a Semantic Wrapper.

**Key words:** Semantic Wrapper, Semantic SQL, heterogeneous database, Three-tier Web architecture.

## 1. INTRODUCTION

In the past decades, various databases have been developed for commercial and research purposes. As a result, various data resides distributed in different databases that are maintained and developed independently. With the rapid development of distributed computing techniques like Internet computing, the demand for enterprise-level cross system data sharing in a global distributed environment becomes more and more pressing. The research being performed on heterogeneous databases satisfies this need by offering users a global transparent heterogeneous database layer sitting on top of all related distributed databases, thus providing an interface for users who would like to work on what appears to be single database instead of several databases with different data models in a distributed environment [20].

At the High Performance Database Research Center (HPDRC) [8], we undertook the Heterogeneous Distributed Database (HDB) project [19], which aims at integrating information from a variety of distributed heterogeneous data sources including structured data sources (e.g., relational databases, semantic databases) and semi-/un-structured data sources like information

from the World Wide Web (WWW or Web). In our solution, the Semantic Binary Object-oriented Data Model (Sem-ODM) [7] developed by HPDRC is selected to build the heterogeneous database layer. Sem-ODM has been shown to be very efficient when compared to relational database on a class of applications that utilize its semantic features [17, 18]. Most popular commercial databases, such as MS SQL server, Oracle, and IBM DB2 are relational databases (RDBMS) developed based on the Relational model [4]. As part of the HDB project, we developed a wrapper (*Semantic Wrapper*) [9, 11, 20] for relational database systems, which provides an interface to relational databases that is similar to a Semantic Database (Sem-ODB) [13, 14, 15].

*Semantic Wrapper* is a set of tools developed for accessing relational databases using the Sem-ODM data model and Semantic SQL query language (Sem-SQL) [12]. Our current *Semantic Wrapper* consists of three major components: *Semantic View Constructor*, *Sem-SQL Server* and *SDB JDBC/ODBC Driver*. We will address each component in the next section.

With *Semantic Wrapper*, a user views the united database through a semantic schema and accesses it via the standard interface of Sem-ODB. The advantages of such interfaces include friendlier and more intelligent generic user interfaces based on the stored meaning of the data, comprehensive enforcement of integrity constraints, greater flexibility, and substantially shorter application programs [11]. In addition, the semantic interface supports a client-server architecture, thus providing a client-server interface for accessing a centralized relational database like MS Access.

This paper presents the system architecture for a Web application that provides semantic access to relational databases. Users view the relational database through a semantic schema and access it via a standard interface of Sem-ODB, which lets users develop the relational database applications by taking the advantages of Sem-

\* This research was supported in part by NASA (under grants NAG5-9478, NAGW-4080, NAG5-5095, NAS5-97222, and NAG5-6830), NSF (CDA-9711582, IRI-9409661, HRD-9707076, and ANI-9876409), ONR (N00014-99-1-0952), and the FSGC.

ODB. As part of the system, we implemented the Web Sem-SQL query interface for relational databases.

The rest of paper is organized as follows. Section 2 introduces the architecture for Web access to relational databases via *Semantic Wrapper*. Section 3 discusses the design of the Web query interface. Section 4 gives the conclusions of our study.

## 2. WEB ACCESS TO DATA SOURCE WITH SEMANTIC WRAPPER

### System Architecture

In this section, we will present the architecture for Web access to relational databases via *Semantic Wrapper* and discuss each module in the architecture.

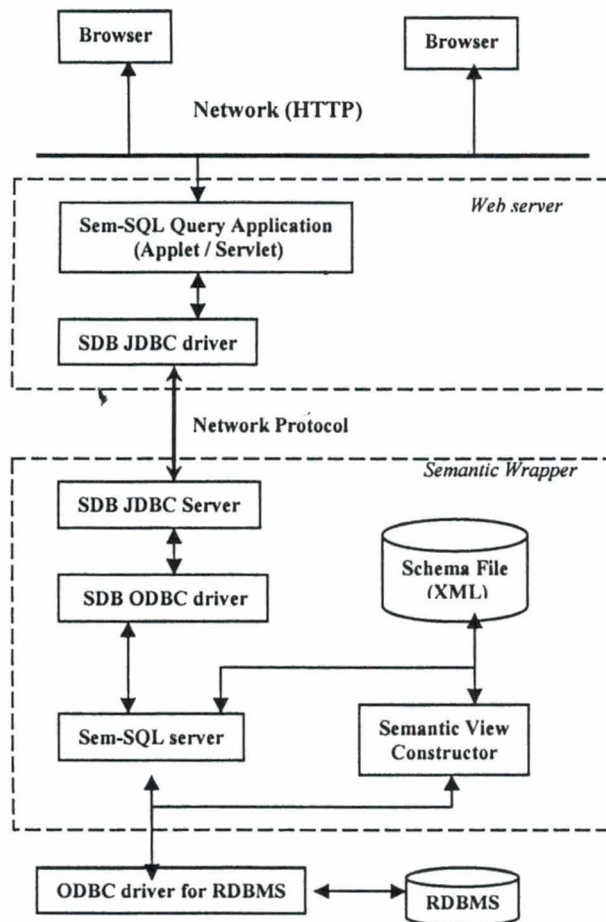


Figure 1. Overall Architecture of Web Semantic Access to RDBMS

Figure 1 shows the overall system architecture for accessing relational databases via the Web with *Semantic Wrapper*. We used a three-tier Web architecture [1] to develop our system.

- The first tier is the Web browser, which serves as our universal client. In the first phase of using the application, an HTML front-end was used for starting the application, which will provide a GUI for user-input and the display of database query results.
- The second tier of the application is implemented with a Web server capable of executing the program serving the server (e.g., Java Servlet program in our system).
- The third tier is composed of our back-end database server, on which *Semantic Wrapper* facilities are installed for accessing the actual relational database server that stores the information. The query application has to talk to *Semantic Wrapper* to get the desired information from the actual relational database.

We will briefly introduce each module in the architecture as follows.

**Browser:** A browser serves as the front-end client that loads the Web pages and programs from Web server to take user-input and display the query results.

**Sem-SQL Query Application:** This module is implemented with Java Servlet/Applet technologies, establishing the connection to database, retrieving the data, and providing a GUI for user-input and the display of query results. In order to retrieve information from RDBMS, this module needs to talk to SDB JDBC server for *Semantic Wrapper* through standard JDBC interface [5]. We will address this module in detail in the next section.

**Semantic View Constructor:** Before making a Semantic SQL query for relational database, the user has to create the semantic view (i.e., semantic schema) for the target relational database. Using this constructor, the user can express semantic information for the relational database by creating a semantic schema for it. Complex semantic expressions such as inheritance and m:m relations can be constructed by this component. All the mapping information will be saved in an Extensible Markup Language (XML) [6] file, which will be used to translate the semantic SQL statement into a standard SQL statement later on. This constructor assists the DBA to make intelligent design decisions in creating a complex semantic schema and also keeps the meta-data consistent.

**Schema File:** The schema file stores both the semantic and relational schemas along with derivation rules for query translation. XML technologies are adopted to save the schema information. By using XML technologies, we were able to easily capture complex semantic information along with the semantic schema from relational databases [10].

**Sem-SQL Server:** This component is the central processor of *Semantic Wrapper*, translating Semantic SQL queries (based on the semantic schema) into equivalent relational SQL queries based on the relational schema of the commercial RDBMS. It uses derivation rules as well as semantic and relational schema information stored in the XML file for this purpose. The relational SQL queries are transmitted to the RDBMS using ODBC interface. The Sem-SQL Server also provides the fundamental programming API for the JDBC/ODBC interface.

**SDB JDBC and ODBC:** This component provides the standard JDBC and ODBC interfaces, which support the Semantic SQL language, thus giving users the flexibility to develop their own applications on top of either interface. Our JDBC interface is designed to support a server/client structure, which means users can make semantic queries against any remote semantic data source through a semantic JDBC interface. Figure 1 illustrates the server/client structure of the semantic JDBC interface. Using the JDBC/ODBC interface, we provide a competitive solution for developing RDBMS applications while making use of the advantages of Semantic Databases.

**Performance**

As the data flow shows in Figure 1, in order to map the semantic information into actual relational data, the user application has to go through two ODBC interfaces. One is SDB ODBC interface through which Sem-SQL is translated to standard SQL. The other is the ODBC interface for the targeted database, which is used to retrieve the actual data from the relational database. One problem arises in retrieving information from the relational database: the query execution performance of the application might be lower than that of the application accessing the relational database only through its own standard interface. This means that users can only take advantage of the semantic interface via *Semantic Wrapper* at the expense of some minor performance losses. However, the advantages in use and development allow the high performance *Semantic Wrapper* module to be applied in commercial relational database applications.

In the next section, we will focus on how to develop the Web query application that provides an online Sem-SQL query interface for accessing relational database via our *Semantic Wrapper*.

**3. WEB SEM-SQL QUERY APPLICATION**

In this section, we introduce our online query application architecture. We used Java Applet and Servlet technology to implement the system.

**Related work**

Applets are Java programs that reside on a Web page and are dynamically loaded into the browser and executed when the Web page is viewed. Applets are discarded when the browser exits the Web page [3]. There are the following advantages in adopting applet technology for the implementation of the query interface program:

- The program enables a Web page to present dynamic content that can interact with the user. In our system, the applet program is used for user-input and displaying results.
- With any operating system, there is no need to install anything on the local machine to run the program.
- To start the program, the user only needs to browse the Web page that embeds the applet program. It is simple to run the program.

On the other hand, there are some security restrictions that restrict applet program access to some local and remote resources [3]. One of the restrictions is that applet programs cannot normally make a network connection to any host other than the Web server that served the applet [2]. One way of working around this restriction is to use a server application that executes on the applet's host. In our system, we used Java servlet [16] technology to develop the server application serving the applet. The servlet program can make the database connections, send queries to the database server, and retrieve the query results to the Applet.

In the rest of this section, we will address how to develop our query application.

**User Interface**

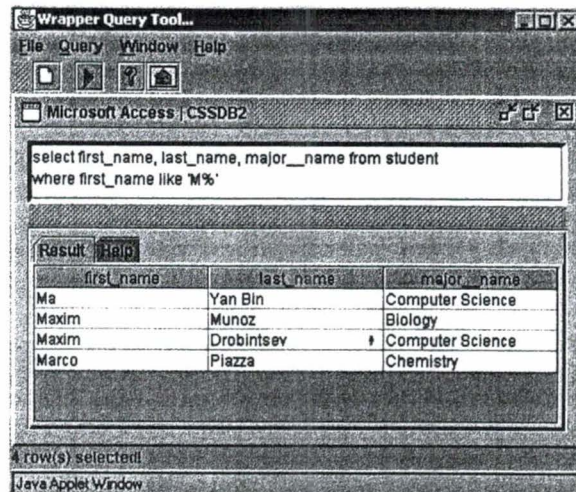


Figure 2 Main GUI for Web Sem-SQL Query Application

Figure 2 shows the main front-end graphical user interface (GUI) for our online Sem-SQL query

application, which provides a friendly interface for inputting semantic SQL statements and displaying query result in a table format.

### System Architecture

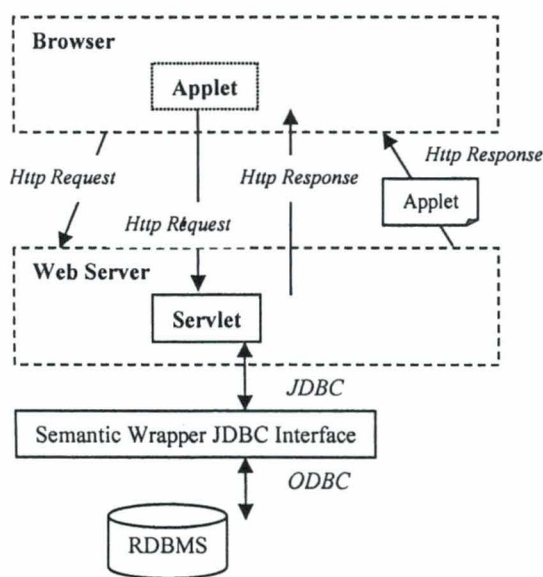


Figure 3 Architecture for Sem-SQL Query Application

As illustrated in Figure 3, we have implemented the Web query application via two Java programs: the Applet and Servlet programs. Both programs originally reside on the Web server: the Applet program will be loaded into browser and serve as the client side and the Servlet program will run on the server side. We briefly discuss the data flow of the query interface below.

- 1) When the user browses the Web page that embeds the Applet program, the Applet program is loaded into the Browser and is invoked on the client machine.
- 2) The Applet provides a GUI for user input of SQL statements and the necessary data source information such as the destination IP address, data source name and username/password. Applet will pack all of the user inputs as a HTTP message and send them to the Servlet program on the Web server.
- 3) After the message is received from the Applet, the Servlet will unpack the message and execute the query using the *Semantic Wrapper* JDBC interface. The Servlet will pack the query result as a HTTP message and send it back to the Applet.
- 4) Finally, the Applet will unpack the message and display the query result.

### Performance

Figure 3 shows the architecture to develop a general online query application for accessing semantic wrapped

(wrapped with *Semantic Wrapper*) relational databases via any machine that has access to the Internet. Our implementation of the system shows that the given architecture is practical and allows efficient development of query applications. Our query application can be used as a general query tool to allow Web users to access their relational databases as long as they create a wrapper for their relational database using our *Semantic Wrapper*.

### 4. CONCLUSION

In this paper, we discussed the architecture for our Web application that provides a query interface for accessing relational databases via the Internet. This interface was developed on top of *Semantic Wrapper*, which provides friendlier and more intelligent generic user interfaces for relational databases. Our approach of using Applet and Servlet technology together is shown to be a practical and convenient way to develop a high performance query interface.

### 5. REFERENCES

- [1] Silberschatz A., Korth H., Sudarshan S., *Database System Concepts*, 4th edition, McGraw-Hill, 2002, 786 pp.
- [2] Applet. <http://java.sun.com/docs/books/tutorial/applet/practical/workaround.html>
- [3] Java Applet. <http://java.sun.com/applets/>
- [4] E.F. Codd. *The Relational Model for Database Management*. Version 2. Reading, MA: Addison-Wesley, 1990.
- [5] Graham Hamilton. An overview of JDBC™.
- [6] "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation, October 2000, <http://www.w3.org/TR/2000/REC-xml-20001006>, work in progress.
- [7] Naphtali Rishe. *Database Design: the semantic modeling approach*. McGraw-Hill, 1992, 528 pp.
- [8] Naphtali Rishe, Wei Sun, David Bardou, Yi Deng, Cyril Orji, Michael Alexopoulos, Leonardo Loureiro, Carlos Ordonez, Mario Sanchez, Artyom Shaposhnikov. *Florida International University High Performance Database Research Center*. SIGMOD Record, 24 (1995), 3, pp.71-76.
- [9] Rishe N., Jun Yuan, Rukshan Athauda, Xiaoling Lu, Xiaobin Ma, "SemWrap: A semantic wrapper over relational databases, with substantial size reduction of user's SQL queries", EDBT 2000 VII. Conference on Extending Database Technology March 27-31, 2000 -- Konstanz, Germany
- [10] Naphtali Rishe, Li Yang, Maxim Checkmasov, "Semantic Mapping: from XML DTD to Semantic Schema", accepted by SCI 2002.

- [11] Naphtali Rische, Jun Yuan, Rukshan Athauda, Shu-Ching Chen, Xiaoling Lu, Xiaobin Ma, Alexander Vaschillo, Artyom Shaposhnikov, Dmitry Vasilevsky. "SemanticAccess: Semantic Interface for Querying Databases" Proceeding of the VLDB conf. , Cairo, Egypt, pp. 591-594, September 10-14, 2000.
- [12] Rische N., "Semantic SQL". Internal Document, High Performance Database Research Center, School of Computer Science, Florida International University, 1998.
- [13] HPDRC. *Semantic Binary Database C++ Interface Version 3*. High Performance Database Research Center. School of Computer Science, Florida International University, 1995.
- [14] HPDRC. *Semantic Binary Database Java Interface*. High Performance Database Research Center. School of Computer Science, Florida International University, 1995.
- [15] Sem-ODB, [http://hpdrc.cs.fiu.edu/\\_SemODB/text/](http://hpdrc.cs.fiu.edu/_SemODB/text/).
- [16] Java Servlet, <http://java.sun.com/products/servlet/>
- [17] Aslan G., McLeod D., "Semantic Heterogeneity Resolution in Federated Databases by Metadata Implantation and Stepwise Evolution", in VLDB Journal, Vol.8, No.2, 1999, pp.120-132.
- [18] Athauda, R., "Heterogeneity Resolution in MSemODB", Technical in Federated Databases by Metadata Implantation and Setpwise Evolution", In VLDB Journal, Vol. 8, No.2,1999, pp.120-132.
- [19] Athauda R., "Integration and Querying of heterogeneous, autonomous, distributed database systems", Ph.D's dissertation, School of Computer Science, Florida International University.
- [20] Lu X., "A Semantic Wrapper Used in Heterogeneous Database Systems", Master's thesis, School of Computer Science, Florida International University.