ReQueSS: Relational Querying of Semi-Structured Data

Rajshekhar Sunderraman
Department of Computer Science
Georgia State University
Atlanta, Georgia 30303-3083
raj@cs.gsu.edu

Abstract

We present a prototype of a Web querying interface which is capable of searching and querying unified Web sources of data that have sufficient hidden relational structure. The system converts query-related parts of Web pages into relational data and provides for SQL-like or QBE-like querying capability. The relational query is parsed for relevant information such as selection conditions and table names. This information is then used in conjunction with a knowledge base containing the network locations of HTML documents at the unified Web source to retrieve only the relevant documents for conversion to relational form. The system is being developed using Oracle 8.0 and Oracle Application Server 4.0 (JWeb Cartridge).

1. Introduction

We present a prototype implementation of a Web querying system that allows the user to pose relational queries on Web sources of data. The system requires (1) a database server (such as Oracle8) which will store relational data that is dynamically converted from HTML or XML format and (2) Web-database integration tool such as Oracle Application Server 4.0, the software platform on which the query interface will be built.

The relational schema for the hidden structure behind the Web data is provided as input to the system. Based on this relational schema, the system provides several user interfaces for ad-hoc querying and searching. Users can express their queries in SQL or QBE (a visual query language). Users can also form their queries by choosing among several form elements and providing certain selection criteria.

The conversion of the data from HTML or XML format and the subsequent loading of the relational tables is query-driven and is done dynamically. Wrappers or converters for various Web sources of data are created and are independent of the querying system.

2. System architecture

The overall system architecture of our approach to querying Web sources of data is presented in Figure 1. The system consists of two main components: the query subsystem and the data converters.

2.1. Query sub-system

The Query sub-system provides several user interfaces for the user to express their ad-hoc queries. The query can be expressed in SQL or QBE or can also be formed using a query page which provides the user several selection criteria to choose from. In all cases, the query is translated into SQL. The query is also parsed for useful information regarding the relational tables necessary to answer the query and selection conditions which may be used by the data converters to limit the data conversion process. The query sub-system is responsible for loading the relational database with the dynamically converted data. It is also responsible for sending the query to the database server, receiving the query results back from the database server, and displaying the results to the user.

The query sub-system has been developed entirely in Java under the JWeb cartridge of the Oracle Application Server 4.0. This sub-system will remain constant when querying different Web sources of data for which individual wrappers have to be developed.

2.2. Data converters

To use our system with a particular Web source of data, we need to develop data converters for the data source.

0-7695-0506-6/00 $10.00 © 2000 IEEE

These converters are essentially programs that automatically convert the Web data in HTML or XML into relational form. We have developed converters for several Web sources in Java.

The data converters are provided with the names of relational tables for which data is to be converted. They are also provided with useful conditions that restrict the number of HTML pages that need to be processed to answer the query. The data converters are equipped with a knowledge base of URLS for the data source along with pertinent information relating the URLs to the database tables and columns.

Developing these data converters for sources of Web data that are very strict in their HTML formatting is very simple. Data items are found near specific HTML tags and can be easily extracted. However, if the HTML formatting of the data source is not consistent or if it changes, the converters do not work well. We are hoping that XML data formatting standard catches on and becomes a universal language for representing Web data. Once the data sources make their data available in this universal format instead of the free formatting HTML that is prevalent nowadays, data conversion becomes a simple task that can be easily automated.

2.3. Observations

1. Our system is an easily configurable tool for any source of Web data. In order to configure the tool for a particular source of Web data, we need to provide (1) the relational schema and (2) a set of modules (one per relation) which will convert the Web data in HTML or XML format into relational form.

2. The conversion of the semi-structured Web data into a relational form is query-driven in our approach. One main advantage of this query-driven approach is that we always access current Web data maintained at the Web sites. However for relatively static Web sources, we could modify our system to periodically load the database with the current contents of the Web source.

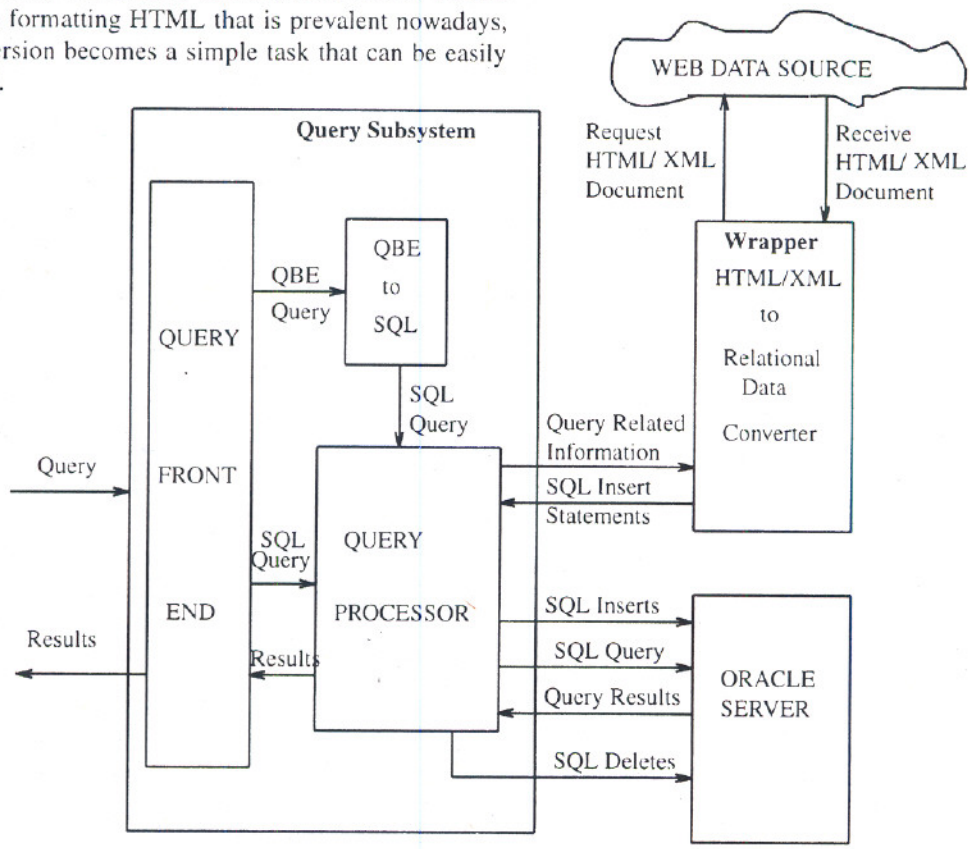

Figure 1: System Architecture