The ARANEUS Web-Base Management System *

G. Mecca
D.I.F.A. - Università della Basilicata
via della Tecnica, 3 - 85100 Potenza, Italy
mecca@dia.uniroma3.it

P. Atzeni, A. Masci, P. Merialdo, G. Sindoni
Dipartimento di Informatica e Automazione
Università di Roma Tre
{atzeni,masci,merialdo,sindoni}@dia.uniroma3.it

http://poincare.dia.uniroma3.it:8080

1 Introduction

The paper describes the ARANEUS *Web-Base Management System* [1, 5, 4, 6], a system developed at Università di Roma Tre, which represents a proposal towards the definition of a new kind of data-repository, designed to manage Web data in the database style.

We call a *Web-Base* a collection of data of heterogeneous nature, and more specifically: (*i*) highly structured data, such as the ones typically stored in relational or object-oriented database systems; (*ii*) semistructured data, in the Web style. We can simplify by saying that it incorporates both databases and Web sites. A *Web-Base Management System (WBMS)* is a system for managing such Web-bases. More specifically, it should provide functionalities for both database and Web site management. It is natural to think of it as an evolution of ordinary DBMSs, in the sense that it will play in future generation Web-based Information Systems the same role as the one played by database systems today. Three natural requirements arise here:first, the system should be fully distributed: databases and Web sites may be either local or remote resources; second, it should be platform-independent, i.e., it should not be tied to a specific platform or software environment, coherently with the nature of the Internet; finally, all system functionalities should be accessible through a hypertextual user interface, based on HTML-like markup languages, i.e., the system should be a site itself.

We can list three main classes of applications that a WBMS should support, in the database spirit: (1) *queries*: the system should allow to access data in a Web-base in a declarative, high-level fashion; this means that not only structured data can be accessed and queried, but also semistructured data in Web sites; (2) *views*: data coming from heterogeneous sources should be possibly reorganized and integrated in new Web-bases, in order to provide different views over the original data, to be navigated and queried by end-users; (3) *updates*: the process of maintaining Web sites is a delicate one which should be carefully supported; semi-structured data need to be updated to reflect changes in the outside world, and sometimes even restructured.

Our system meets all of the above requirements. Original features of the system are: (*i*) a new data model for Web documents and hypertext; (*ii*) several new languages for wrapping, querying, creating and updating Web sites; (*iii*) a number of new techniques for publishing data on the Web. We have tested our approach on several real-life applications, ranging from high-level access to Web sites, including the Database and Logic Programming Bibliography Server at Trier [9], to Web site integration [3], to Web site design and development [2]; these concrete experiences helped us to better shape our ideas and gave a fundamental contribution towards the development of the system.

2 Related Work

To the best of our knowledge, the only system that is similar in spirit to ours is STRUDEL [8], which however adopts a radically different approach in which no explicit data model is used to describe the scheme of a hypertext and graphs are used to store data and pages.

Some aspects of our work are also similar to features of commercial DBMS systems, which often provide an interface to the Web. However, also in that case, no data model is used to describe pages and hypertexts. Moreover, these proposals tend to adopt a pull approach to Web publishing, whereas we also support materialized approaches. In the following section we compare these two techniques and discuss the respective advantages and disadvantages.

3 The ARANEUS Web-Base Management System: Architecture

The ARANEUS WBMS introduces a number of new tools and techniques for managing Web-bases. The overall architecture is shown in Figure 1. The system is implemented in Java and runs on any Java-enabled platform. The **User Interface** is completely written in HTML, so that end-users and administrators can access the system from any client on the network. Due to the heterogeneous nature of Web-bases, several data models and languages are used. In the following, we briefly discuss the key features and the corresponding components of the system.

3.1 Data Models

In our perspective, structured data are essentially database tables; hence, the data model used to describe structured

[*] Work partially supported by Università di Roma Tre, MURST, and CNR.

SIGMOD '98 Seattle, WA, USA
© 1998 ACM 0-89791-995-5/98/006...$5.00

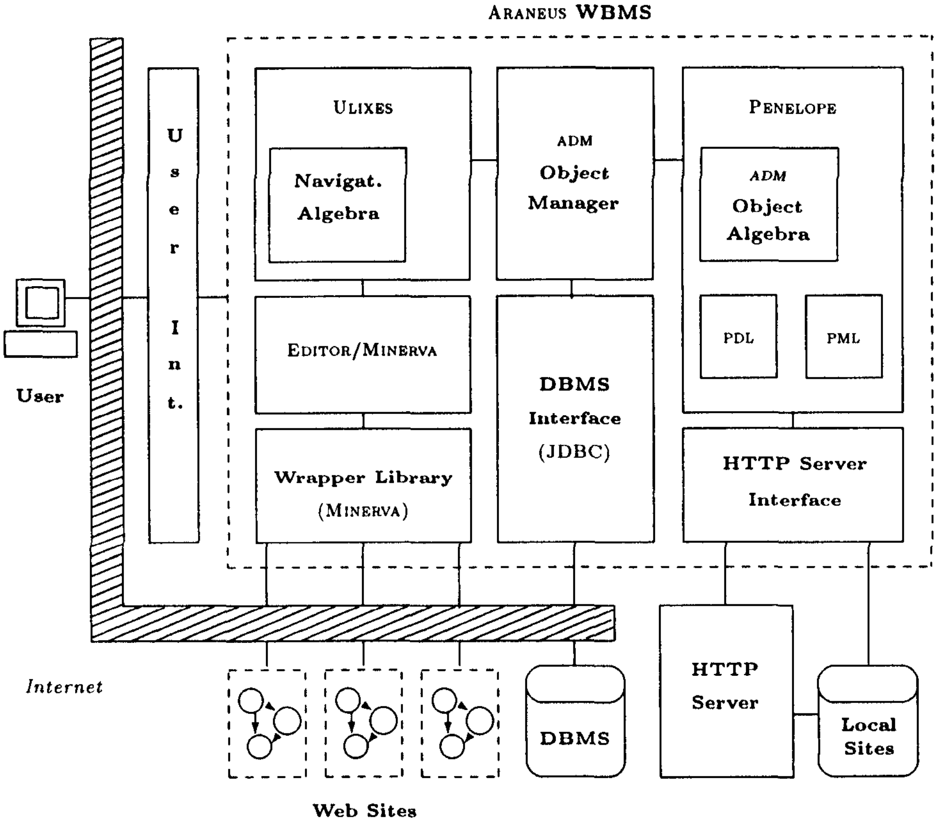

Figure 1: Architecture of the ARANEUS WBMS

data is the relational model, and the corresponding language is SQL. Note that other models for structured data were possible, for example object-based. However, the adoption of a relational model allows us to leverage standard, wide-spread, and robust technology, and at the same time to accomplish a real platform independence. In fact, we assume that the system has access to a (possibly remote) DBMS — as shown in Figure 1—and uses it to store and manipulate tables. Strictly speaking, the DBMS is not part of the WBMS itself; it is more appropriate to say that the WBMS relies on a DBMS to handle tables. Any table-based DBMS—relational or object-oriented—fits in the system; the standard SQL-based protocol JDBC is used to communicate with the database.

A key component of the system is the use of the ADM data model, an ODMG-like model for describing Web-hypertexts. We say that the model is *page-oriented*, in the sense that pages play a central role. Each page is seen as an object with an identifier, the URL, and a number of attributes. Attributes may be either atomic, like strings, images, and links to other pages, or multivalued; multivalued attributes are essentially lists of tuples, possibly nested. Thus, each page is seen as a URL-identified nested tuple. Pages are grouped into *page-schemes*, which recall the notion of relation scheme or class in databases, i.e., collection of objects of homogeneous structure. It is possible to see an instance of a page-scheme as a set of nested tuples, one for each page of the corresponding type. A *Web site* is a collection of page-schemes connected by links.

Beside these rather standard notions, the model has several peculiar features. For example, since access to pages is based on actual navigation, a site has a number of distinguished page-schemes, which act as entry-points to the site; at least the home-page falls in this category. Inheritance is not provided, in favor of *heterogeneous union*, which in our experience better models semi-structured hypertexts. Also, other types such as *forms* are introduced to model Web-specific features. A number of examples of ADM schemes can be found on our Web site.

In the system, ADM objects are handled by the ADM **Object Manager** module (see Figure 1); it manipulates nested objects by decomposing them and using the DBMS to store them in flat tables.

3.2 Queries over Hypertexts

The system allows to query Web-sites using a suitable language called ULIXES; note that such sites can be either external or local. However, querying remote sites that are not under the system's direct control requires a more complex process, as follows.

When an external site is to be queried, an ADM description of the site has to be derived by analyzing its content. Then, the site needs to be wrapped in order to extract data from pages and see them as instances of page-schemes. Wrappers are written using a text-management tools called EDITOR [4] and MINERVA [7]. They allow to search and restructure semi-structured documents. Regions in documents can be declaratively parsed using regular expressions, or procedurally restructured using "cut & paste" instructions, in the attempt to find a good compromise between high-level functionalities and flexibility in managing exceptions and errors. The system uses an extensible **Wrapper Library** to store wrappers; whenever a page of a certain page-scheme is to be accessed, its source is downloaded from the network and then processed by the corresponding wrapper, which extracts attribute values and stores the resulting ADM object using the **Object Manager**.

ULIXES allows to express and evaluate queries over a Web site. It implements a **Navigational Algebra** [10] over nested structures. *Navigational expressions* are used to express paths in the ADM schemes, along which selections, projections and joins select data of interest and return a set of tuples. In this way, data can be extracted from a site and stored in the database; these database abstractions over sites can be further processed for reorganization, integration or application development. Several examples of ULIXES queries can be seen on our Web site; these queries can also be executed using an on-line version of the system.

Note that, as discussed in the following section, the system also allows to create and manage new sites from scratch. For these sites, wrappers are generated in a completely automatic way, thus making them immediately available for querying purposes.

3.3 Hypertext Creation and Management

The second major domain of application of ARANEUS is the creation and administration of new sites. Here, the idea is that data to be published on the site are stored in the database. Two different approaches to Web publishing are possible in this context. Products on the market adopt *pull* techniques, in which pages contain calls to the DBMS, which, when the user requests a new page, are evaluated, generate the page on the fly and return it to the browser. The main advantage of this approach is that pages always reflect the most recent database state; however, there are at least two limitations associated with it. First, if the underlying database has to be used also for other ends—for example, like a repository for a company information systems—frequent accesses to the Web site may considerably increase the load on the system and can slow down the overall performance. Second, the resulting Web site is strongly platform dependent: the HTTP server needs a specific DBMS as a back end to serve pages, which often contain non-standard tags to invoke the execution of scripts; this means, for example, that such a site cannot be mirrored or distributed over the network, or moved to another platform without also migrating the DBMS.

An alternative is represented by a *push* approach, in which data are materialized in HTML files and 'pushed' to the site. This clearly solves the problems above, since the resulting site is standard and the HTTP server works independently from the DBMS; however, in this case, the management of pages in presence of updates is more complex; in fact, when the database is updated, also materialized HTML files need to be correspondingly maintained to reflect the change; this imposes the development of a *page update language*.

The ARANEUS WBMS supports both approaches: in a site, pages can be kept virtual, and generated on-the-fly, or materialized in HTML files. The module that supports the process of defining and maintaining new sites is called PENELOPE. It incorporates the ADM **Object Algebra**, a nested relational algebra with URL invention that can be used to generate ADM views, and therefore HTML pages, over database tables. These views are declaratively defined using the PENELOPE *Definition Language* (PDL), and maintained using the PENELOPE *Manipulation Language* (PML). When a new site is to be generated, its ADM scheme is first designed. This process is supported by a specific design methodology [6]. Then, the ADM structure is mapped to the database using PDL, which provides a `Define-Page` command to describe the structure of a page-scheme in terms of database tables; page-schemes can be arbitrarily nested and linked using the ADM **Object Algebra**. Then, the actual site is generated using PML, which provides two main instructions: `Generate` and `Remove`, which can refer to (*i*) the whole site; (*ii*) all instances of a page-scheme; or (*iii*) pages that satisfy a condition in a `Where` clause. To maintain pages, the system allows to execute *mixed transactions*, in which SQL updates to the database and PML updates to pages are combined in order to guarantee consistency between the two.

References

[1] The ARANEUS Project Home page. http://poincare.dia.uniroma3.it:8080/Araneus

[2] Faculty of Engineering at University of Basilicata. http://www.ing.unibas.it. In italian.

[3] The Integrated Web Museum. http://poincare.dia.uniroma3.it:8080/penelope/webmuseumNEW/webmuseum.html.

[4] P. Atzeni and G. Mecca. Cut and Paste. In *Sixteenth ACM SIGMOD Intern. Symposium on Principles of Database Systems (PODS'97), Tucson, Arizona*, pages 144–153, 1997.

[5] P. Atzeni, G. Mecca, and P. Merialdo. To Weave the Web. In *International Conf. on Very Large Data Bases (VLDB'97), Athens, Greece, August 26-29*, pages 206–215, 1997.

[6] P. Atzeni, G. Mecca, and P. Merialdo. Design and maintenance of data-intensive Web sites. In *VI Intl. Conference on Extending Database Technology (EDBT'98), Valencia, Spain, March 23-27*, 1998.

[7] V. Crescenzi and G. Mecca. Grammars have exceptions, 1998. Submitted for Publication.

[8] M. Fernandez, D. Florescu, J. Kang, A. Levy, and D. Suciu. STRUDEL – a Web site management system. In *ACM SIGMOD International Conf. on Management of Data (SIGMOD'97), Tucson, Arizona*, 1997. Exhibits Program.

[9] M. Ley. DataBase systems and Logic Programming bibliography site. http://dblp.uni-trier.de.

[10] G. Mecca, A. Mendelzon, and P. Merialdo. Efficient queries over Web views. In *VI Intl. Conference on Extending Database Technology (EDBT'98), Valencia, Spain, March 23-27*, 1998.