Scientific Data Integration: Wrapping Textual Documents with a Database View Mechanism and an XML Engine

Zoé Lacroix *

SurroMed Inc., 1060 East Meadow Circle, Palo Alto CA 94303, USA

zlacroix@surromed.com

Abstract

Nowadays scientific data is inevitably digital and stored in a wide variety of formats in heterogeneous systems. Scientists need to access an integrated view of remote or local heterogeneous data sources with advanced data analyzing and visualization tools. Building a digital library for scientific data requires accessing and manipulating data extracted from flat files or documents retrieved from the Web. We present an approach to querying Flat Files as well as Web datasources through an object database view based a database system and a wrapper. Generally a wrapper has two tasks: it first sends a query to the source to retrieve data and, secondly builds the expected output with respect to the virtual structure. Scientific data servers and in particular the ones publicly available on the Web, usually provide information retrieval techniques to access data. Our wrappers are composed of a retrieval component based on an intermediate object view mechanism called search views *mapping the source capabilities (including full-text retrieval accesses) to attributes, and a XML engine to perform respectively these two tasks. If the retrieval component is specific to each datasource, this approach shows that the extraction component, the XML engine, can be common.*

We describe our system and focus on the retrieval component of the Object-Web Wrapper (OWW) for Web sources. The originality of our approach consists of (1) a common wrapper architecture for flat files and Web datasources sharing a XML engine for data extraction, (2) a generic view mechanism to access datasources with limited capabilities, and (3) the representation of hyperlinks as abstract attributes in the object view as well as their use in the search view. Our approach has been developed and demonstrated as part of the multidatabase system supporting queries via uniform Object Protocol Model (OPM) interfaces.

*The work presented in this paper was done when the author was a Senior Research Scientist at Gene Logic Inc.

1 Introduction

A large part of the work of scientists today consists in querying a scientific digital library, integrated view of remote or local heterogeneous data sources with advanced data analyzing and visualization tools. In each domain, the access to the relevant data, combining datasources and coping with their distribution and heterogeneity is a tremendously difficult task. Biological datasources obey the common rule. Biological data is available in a wide variety of formats, annotated, and stored in flat-files and relational or object-oriented databases. Datasources are heterogeneous and do not provide the user with a real query language. Access to Flat File sources such as GenBank or SwissProt [BA99] are mandatory to the biologists. Access to Web sources such as LENS (Linking Expressed Sequence Tags and their associated Name Space) datasource [LEN], Saccharomyces Genome Database [SGD], or PubMed [Pub] provide the biologists with a necessary complement of information. These sources are textual and of restricted access facilities. Their structure varies from ASN.1 data exchange format for GenBank to poorly structured HTML format for LENS.

Data of interest to the scientists is widely distributed and constantly updated in heterogeneous remote public data sources or local private ones. Integration of data means accessing up-to-date data across multiple available data sources. The database community has devoted a large amount of work on integration of data either materialized within data warehouses or non-materialized. For the specific integration of biomolecular data, one can cite BioKleisly [DOTW97], the multi-database system based on the Object Protocol Model (OPM) [CKMS98] or TAMBIS [BBB+98]. BioKleisly enables on-line queries against integrated datasources but requires additional implementation of programs in order to use retrieval facilities provided by Web data servers (see [Won98]). TAMBIS provides an integrated view of data sources through a pool of concepts but offers no link to information retrieval facility and requires the implementation of a new wrapper for each new data source. The OPM multi-database system is based on the Object Protocol Model (OPM) [CM95] to design object views [CKMS97] of the sources and its query language, an OQL-like query language. It has a convenient architecture and APIs for its extension to new wrappers, as CORBA servers.

The whole problem of designing a virtual database schema is to provide the user with the best object view of the datasource when the access to the data is through a restricted view constrained by both *retrieval* and *extraction* facilities. Indeed, the evaluation of a query against any of these sources has to process both retrieval and extraction. The retrieval step consists in selecting documents out of (a large number of) documents. A document is either a textual flat file or a HTML output. Retrieval for flat files may be done through an indexing system such as SRS [EA93]. For a Web source, the retrieval is restricted by the provider with CGI calls or other available accesses. Extraction means the analysis of the data contained in the retrieved documents for final data manipulation to build the output as specified in the query. The sources we consider return formatted textual documents. The extraction step consists in parsing and identifying the relevant data they contain.

We propose an architecture including an object database system, a retrieval and an extraction components to access either flat file or Web sources. The retrieval component processes the query and extracts retrieval conditions. A retrieval condition can be a search on indexed fields or a Blast search call on a flat file source, or a CGI call to a Web source. Obviously the retrieval component is specific to each source. In the paper, we present retrieval component of the Object-Web Wrapper (OWW) based on an intermediate object view mechanism called *search views* [Lac99, Lac00]. Search views express the mapping between the object view and the access capabilities available to retrieve the data from Web sources. Search views can map *direct calls* retrieving a document that completely describes an object, *search calls* to the search engines retrieving a document containing a list of objects and *abstract calls* corresponding to hyperlinks between objects. We describe and illustrate our approach with the LENS datasource [LEN] and PubMed [Pub].

The extraction component processes the retrieved files and returns the output in its virtual format to the database system. Contrary to the retrieval component, the extraction component does not necessary depend on the datasource if the processed documents have a unique format. The emergence of eXtensible Markup Language (XML) as a new standard for data representation and exchange on the Web motivates us to choose XML [BSM97] as an intermediate format. Our extraction component is an XML engine, an interface between the database system on one side and the retrieved XML documents on the other side. Since none of our actual documents are generated in XML, we use parsers that translate them into XML. But we expect that soon data providers will deliver data in XML format. To an OPM*QL query sent by the database system, the XML engine returns records built from data extracted from XML documents.

In the next section, we describe the general architecture and focus on the retrieval component built for Web sources. In Section 3 we conclude and discuss future extensions.

2 Architecture

The architecture is composed of a multidatabase system, a XML engine and a retrieval component. The XML engine and the retrieval component constitute a wrapper for a given source. The design of the database view results from the retrieval and extraction capabilities of the wrapper. We first describe how the retrieval component of the OWW only consists of an intermediate view mechanism. In Section 2.3, we introduce the XML engine common to all wrappers. Finally, we explain how the object view is designed in Section 2.4.We illustrate our approach with queries to Web sources such as LENS, SGD and PubMed (resp. [LEN, SGD, Pub]).

2.1 An OPM*QL query to LENS

The LENS (Linking Expressed Sequence Tags and their associated Name Space) datasource [LEN] is one of several databases created and maintained at the Computational Biology and Informatics Laboratory at the University of Pennsylvania. Like many other datasources, LENS is only available on the Web. To query LENS, a user has no other option than to use a browser, type the URL of the datasource and fill a form where he can give a clone name or identifier and specify a type. Each type corresponds to a CGI call and a key. Suppose a user wants to know the name of a clone identified by "25002" in I.M.A.G.E. datasource, the date of its entry in the datasource and its corresponding identifier in GDB. To ask the query, he selects type "IMAGE ID", types "25002" and submits. After waiting for the document to be retrieved and displayed he can read the page and gets the expected output.

Now suppose that the user is querying an OPM view of the LENS datasource consisting of a class

Clone and attributes clone_name, gdb_id, image_id, etc. He submits an OPM*QL query (see below) to the OPM multidatabase system. The multidatabase system processes and sends the query to the OWW. The retrieval component builds and sends a call to retrieve the document. The document is parsed and cached in XML format. The XML engine extracts the data necessary to answer the query and returns it to the database system.

```
select  clone_name = @c.clone_name,
        gdb_id = @c.gdb_id,
        date_of_entry = @c.date_of_entry
from    @c in LENS:Clone
where   @c.image_id = "25002"
```

2.2 Retrieval Component

In this section, we focus on the retrieval component of the OWW, specific to Web datasources or textual sources accessible through a search engine. The task of the retrieval component consists in sending retrieval queries (calls) to the Web source when getting an OPM query as input. In general, the task of wrapping data from a database system to another is a difficult task, but at least the data model and query language of both sides are known. Textual and Web sources do not have a data model, but are rather semistructured [Abi97, Bun97]. If textual datasources can provide a real language to retrieve data thanks to a sophisticated indexing system, the search facilities available for Web sources are rather poor. Each Web datasource has entry points (keys in a CGI form, for example) which allow to retrieve documents either describing completely an object (*direct calls*) or providing a list of relevant objects (*search calls*). A retrieval query must correspond to one of these entry points. The retrieval component has the hard task to map a real database query with simple CGI calls. To accomplish its tasks, the Web retrieval component of the OWW is a view mechanism based on a *search view* introduced in [Lac99].

In the search view, direct calls are represented as *key attributes*[1], when search calls are associated with *search attributes* [Lac00]. Most Web datasources only provide data embedded into a HTML shell for display (as the current version of LENS). However, there are sources where HTML, and in particular its hyperlinks, is used to structure the data. Hyperlinks can be often represented in the object-oriented database framework as isa relationships or as abstract attributes. The search view and the associated view mechanism presented in [Lac99, Lac00] of the OWW are extended to take advantage of abstract attributes as described in the following.

[1] Key attributes were called *search attributes* in [Lac99].

```
DATASOURCE LENS
 CLASS  Clone
 WRAPPER wrapper.perl
 KEY ATTRIBUTE image_id
   CGI http://www.cbil.upenn.edu/cgi-bin/
       lens/viewCloneId.perl?all_types.
       type=I.M.A.G.E.+ID&all_types=
   PARSER LensClone
 KEY ATTRIBUTE gdb_id
   CGI http://www.cbil.upenn.edu/cgi-bin/
       lens/viewCloneId.perl?all_types.
       type=GDB+ID&all_types=
   PARSER LensClone

DATASOURCE PUBMED
 CLASS  Citation
 WRAPPER wrapper.perl
 KEY ATTRIBUTE pubmed_id
   CGI http://www.ncbi.nlm.nih.gov/cgi-bin/
       Entrez/referer?/htbin-post/
       Entrez/query%3fdb=m&form=6&uid=
   PARSER PubMed
 KEY ATTRIBUTE medline_id
   CGI http://www.ncbi.nlm.nih.gov/cgi-bin/
       Entrez/referer?/htbin-post/
       Entrez/query%3fdb=m&form=6&uid=
   PARSER PubMed
 SEARCH ATTRIBUTE text_search
   CGI http://www.ncbi.nlm.nih.gov/htbin-post/
       Entrez/query?form=4&db=m&term=
   PARSER PubSearch
 ABSTRACT ATTRIBUTE similar_articles
   ASSOCIATED ATTRIBUTE pubmed_id
   TARGET DATASOURCE PUBMED
   TARGET CLASS Citation
   CGI http://www.ncbi.nlm.nih.gov/htbin-post/
       Entrez/query?db=m&form=6&uid=
   PARSER PubSearch
```

Figure 1: Search View

An *abstract attribute* links an object to one or several objects of a target datasource and class. It can be either single-valued or set-valued. A single-valued abstract attribute is associated with a direct call and the attribute whose value is used to build the call. A set-valued attribute is associated with a search call and the attribute whose value is used to build the call.

Consider PubMed [Pub], the National Library of

Medicine's search service that provides access to over 10 millions citations. It provides direct access to citations given a PubMed or a MEDLINE identifiers. It also enables typed search calls (through author names, journal titles, etc.) as well an untyped search call (fulltext index through the whole description of the citation). Given a citation, it also provides links to similar documents and NCBI databases. The latter links can be represented as abstract attributes. The object view of PubMed defines an abstract attribute `similar_articles` linking the object class `Citation` to itself. In the search view of PubMed (see Figure 1), `similar_articles` is represented as an abstract attribute, associated to a search call (the abstract attribute is set-valued) that requests the PubMed identifier (associated attribute) to retrieve the citations (target class) of PubMed (target datasource).

Given a query, the retrieval components generates a query plan based on search conditions extracted from the `WHERE` clause of the query and associated with search values. Selected search conditions generate retrieval queries (calls) to the Web datasource. The query planning described in details in [Lac00] is extended to generate retrieval queries corresponding to abstract attributes in a second step. When all retrieval queries corresponding to key and/or search attributes are generated and documents are processed and cached in XML, new retrieval queries are generated. Before sending any call to the Web source, the retrieval component checks whether or not the document(s) are already cached. It sends the output name(s) back to the XML engine. The current implementation assumes that the query can be completely evaluated. It implies that (1) there are enough search conditions in the query to generate retrieval queries, and (2) all retrieval queries success.

2.3 XML Engine

Common component of all wrappers, the XML engine extracts data from XML documents and return it to the multidatabase system for query evaluation. It only extracts the data necessary to answer the query. The input of the XML engine consists of a query and a set of file names. The files are XML documents either flat files locally stored in XML format or files retrieved from the Web and cached in XML format.

In order to use a single XML engine to query both Flat File and Web sources, when none of them are in XML format we need to parse and translate documents into XML. The approach to generate XML documents from flat files and Web datasources are rather different. Flat files such as SwissProt and GenBank are stored locally and their structure is well known. Their translation into XML is done beforehand when loading the source. Each local flat file datasource is parsed and translated into XML format, and indexed and stored in XML.

Retrieval queries to Web datasources, retrieve HTML documents. As soon as retrieved from the Web, documents are parsed and cached in XML. However our experience has proven that the output of Web datasources often is not in "pure" HTML format, and parsers for HTML documents usually fail. Moreover, HTML tags do not structure the output: the data is mostly textual and often only embedded into an HTML shell for display. Most of the Web datasources we wrapped are *semi-structured* [Abi97, Bun97]: we do not know what their format really is, unexpected attributes show up, etc. For all these reasons, the parsers we use to process documents retrieved from the Web are currently manually generated PERL subroutines. This situation is obviously not satisfactory. A knowledge-based approach (see for instance [CINS97, AK97, CGM98, GRVB98]) using a knowledge-based representation of the datasource could be used for an automatic wrapper generation. But we mostly expect that soon data providers will deliver data in XML format. When it happens, the problems mentioned above will no longer exist. The necessary ability of our system to parse and extract relevant data will be significantly improved thanks to an XML engine able to translate the XML documents and manipulate the data they contain to cache them with respect to our XML specification. The approach will be similar to XML-QL [DFF+98], Strudel [FFK+97] or Lorel [AQM+97, GMW99].

2.4 A Database View in 2 easy steps!

We briefly present some features of the OPM Multidatabase Query System based on the OPM model [CM95]. It follows the ODMG standards [Ba97] with objects identified by a unique object identifier, qualified by attributes and classified into classes. Attributes are either simple or tuple of simple attributes and classes are organized within a sub-classing relationship. OPM*QL is an object-oriented query language similar to OQL [Ba97]. We do not discuss any of the multidatabase aspects of the system in the paper and invite the reader to read [CKMS98] for a complete description.

The design the OPM schema of the source is restricted to the retrieval and extraction capabilities of the wrapper. Retrieval capabilities are expressed in the *search view* as key, search or abstract attributes associated to classes. Each search class defines an OPM class. All attributes mentioned in the search view must be defined at their corresponding OPM class. All attributes corresponding to XML elements (tags) are also defined at their corresponding OPM class. To summerize, thanks to the OWW, the design of an OPM view of a Web datasource is really done in two easy steps:

(1) write the search view of the Web datasource (identify direct and search calls and hyperlinks and write parsers);

(2) write the resulting OPM view of the Web datasource (and register it in the database directory).

3 Conclusion

Querying scientific data, and in particular biomolecular data, requires the integration of widely distributed heterogeneous data sources and advanced data analysis and visualization tools. A large amount of scientific data is semi-structured and stored in flat-files or published on the Web and must be integrated with structured data available in relational or object-oriented databases.

In [CDY95, DM97], query processing and optimization techniques are proposed for a coupling of a relational database system and a text retrieval system. The use of textual similarity is proposed in [Coh98] in the case of a much tighter integration between relational databases and statistical IR queries. The Web-Supported (Databases) Queries (WSQ) proposed in [GW99] leverages results from Web searches to enhance SQL queries assuming that search engines return large amounts of results with ranks.

The Object-Web Wrapper (OWW) presented is this paper is based on the notion of *Search View* presented in [Lac99] extended to access most of the source capabilities. Key attributes represent direct calls when search attributes are linked to a search engine and allow users to express full-text conditions. Hyperlinks are associated with abstract attributes. Our approach allows typed search (with wildcards on attributes) or untyped search (with the `text_search` attribute) but it could be further extended to allow a user to search over the object structure of the view like in OQL-doc [ACC+97] with generalized path expressions. Such an extension would enable the user to express queries without a full explicit knowledge of the object structure. In addition, query planning could allow more flexibility as proposed in [LC00].

The design of a search view of a Web data source only requires to know what are the available calls or parameterized URLs to access the search engine and to write a parser to translate retrieved documents into XML format. Besides the implementation of the parsers (that can be performed automatically by one of the several parser generators currently available), no additional programming effort is required to access any new Web data source thanks to the OWW search view mechanism. Moreover, we expect that soon data providers will deliver data in XML format. When data is published in XML, since OWW is using XML as an internal format, parsers will no longer be needed and only an XML translation will be performed by the XML engine.

Acknowledgments

The author wishes to thank in particular Anthony Kosky, Paula Ta-Shma and I-Min Chen for answering all questions regarding the OPM multidatabase system, the OPM wrapper for Flat-File sources and their implementation.

References

[Abi97] S. Abiteboul. Querying semi-structured data. In *Proc. of Intl. Conf. on Database Theory*, pages 1–18, Delphi, Greece, January 1997. LNCS 1186, Springer.

[ACC+97] S. Abiteboul, S. Cluet, V. Christophides, T. Milo, G. Moerkotte, and J. Siméon. Querying documents in object databases. *Journal on Digital Libraries*, 1997.

[AK97] N. Ashish and C. Knoblock. Wrapper generation for semi-structured internet sources. In *ACM SIGMOD Workshop on Management of Semistructured Data*, Tucson, Arizona, May 1997.

[AQM+97] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J.L. Wiener. The lorel query language for semistructured data. *Journal on Digital Libraries*, 1997. ftp://db.stanford.edu/pub/papers/lorel96.ps.

[Ba97] D. Bartels and al. *The Object Database Standard: ODMG 2.0.* Morgan Kaufmann, San Francisco, 1997.

[BA99] A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence databank and its supplement TrEMBL. *Nucleic Acids Res*, 1(27):49–54, January 1999. http://www.expasy.ch/sprot.

[BBB+98] P. Baker, A. Brass, S. Bechhofer, C. Goble, N. Paton, and R. Stevens. TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. An Overview. In *Sixth International Conference on Intelligent Systems for Molecular Biology (ISBM98)*, 1998.

[BSM97] T. Bray and C. M. Sperberg-McQueen. *The XML Specification.* W3C, 1997. available at http://www.w3.org/pub/WWW/TR/WD-xml.html.

[Bun97] P. Buneman. Semistructured data. In *Proc. ACM Symp. on Principles of Database Systems*, Tucson, 1997. Invited tutorial.

[CDY95] S. Chaudhuri, U. Dayal, and T. Yan. Join Queries with External Text Sources: Execution and optimization Techniques. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 410–422, San Jose, California, 1995.

[CGM98] T. Critchlow, M. Ganesh, and R. Musick. Automatic Generation of Warehouse Mediators Using an Ontology Engine. In *Proceedings of the 5th KRDB Workshop*, Seattle, WA, 1998.

[CINS97] T. Catarci, L. Iocchi, D. Nardi, and G. Santucci. Conceptual Views over the Web. In *Proceedings of the 4th KRDB Workshop*, Athens, Greece, 1997.

[CKMS97] I.A. Chen, A.S. Kosky, V.M. Markowitz, and E. Szeto. Constructing and Maintaining Scientific Database Views. In *Proceedings of the 9th Conference on Scientific and Statistical Database Management*, August 1997.

[CKMS98] I.A. Chen, A.S. Kosky, V.M. Markowitz, and E. Szeto. Exploring Heterogeneous Biological Databases: Tools and Applications. In *Proceedings of the 6th Conference on Extending Database Technology*, March 1998. http://www.genelogic.com/opm.htm.

[CM95] I.A. Chen and V.M. Markowitz. An Overview of the Object-Protocol Model (OPM) and OPM Data Management Tools. *Information Systems*, 20(5):pp 393–418, 1995.

[Coh98] W. Cohen. Integration of Heterogeneous Databases Without Common Domains Using Queries Based on Textual Similarity. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 201–212, Seattle, Washington, 1998.

[DFF+98] A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu. XML-QL: a query language for XML. In preparation - Available at http://www.w3.org/TR/NOTE-xml-ql /, 1998.

[DM97] S. Dessloch and N. Mattos. Integrating SQL databases with Content-Specific Search Engines. In *Proceedings of the 23rd International Conference on Very Large DataBases (VLDB)*, pages 276–285, Athens, Greece, 1997.

[DOTW97] S. Davidson, C. Overton, V. Tannen, and L. Wong. BioKleisli: A Digital Library for Biomedical Researchers. *Journal of Digital Libraries*, 1997.

[EA93] T. Etzold and P. Argos. SRS, An Indexing and Retrieval Tool for Flat File Data Libraries. *Computer Applications of Biosciences*, 9(1):49–57, 1993. See also http://srs.ebi.ac.uk.

[EST] EST. http://genome.wustl.edu/est/esthmpg.html. WashU-Merck Human EST Project.

[FFK+97] M. Fernandez, D. Florescu, J. Kang, A. Levy, and D. Suciu. STRUDEL: A Web-site Management System. In *ACM SIGMOD – Research prototype demonstration*, Tucson, Arizona, May 1997.

[GMW99] R. Goldman, J. McHugh, and J. Widom. From Semistructured Data to XML: Migrating the Lore Data Model and Query Language. In *ACM SIGMOD Workshop on the Web and Databases (WebDB'99)*, 1999. available at http://www-db.stanford.edu/lore.

[GRVB98] J-R. Gruser, L. Raschid, M. Vidal, and L. Bright. Wrapper Generation for Web Accessible Data Sources. In *CoopIS*, 1998.

[GW99] R. Goldman and J. Widom. WSQ/DSQ: A Practical Approach for Combined Querying of Databases and the Web. Technical report, Stanford University Technical Report, October 1999.

[Lac99] Z. Lacroix. Object views through search views of web datasources. In *International Conference on Conceptual Modeling (ER99)*, Lecture Notes in Computer Science, Paris, France, November 1999.

[Lac00] Z. Lacroix. Querying annotated scientific data combining object-oriented view and information retrieval. In *6th International Conference on Content-Based Multimedia Information Access (Recherche d Informations Assistée par Ordinateur) RIAO 2000*, Paris, France, April 2000.

[LC00] C. Li and E. Chang. Query planning with limited source capabilities. In *16th International Conference on Data Engineering (ICDE2000)*, pages 401–412, San Diego, California, February 2000.

[LEN] LENS. http://www.agave.humgen.upenn.edu/lens. Center for Bioinformatics, University of Pennsylvania.

[Pub] PubMed. http://www.ncbi.nlm.nih.gov/pubmed/. National Library of Medecine.

[SGD] SGD. http://genome-www.stanford.edu/saccharomyces/. Department of Genetics, Stanford University.

[Won98] L. Wong. Some MEDLINE Queries Powered By Kleisli. In *ACCESS*, June 1998.