

# IDB: Unified Query Interface for Information on the Web

**Jaewoo Kang**

Computer Science Department  
University of Wisconsin-Madison  
Madison, WI 53706  
jaewoo@cs.wisc.edu  
Advisor: Jeffrey Naughton

October 22, 1999

## 1 Proposal

XML [4] is likely to become the primary vehicle of the information interchange on the Web. Organizations will publish and export their data in XML to facilitate inter- and intra-organization information sharing. Businesses will publish their product information in XML for their customers or for software agents for on-line shopping services. This is already beginning to happen: a number of ontologies have been developed and adopted by many financial and research communities, including Open Financial Exchange (OFX) [18], Mathematical Markup Language (MathML) [12], Biopolymer Markup Language (BIOML) [2], and others. Note that XML will not just be used for publishing static documents; queryable information sources will provide access to the data they store by defining an external view of it in XML [5, 20]. Ideally, the union of all these static XML pages together with XML views of queryable information sources will make the web viewable as an enormous XML repository.

Unfortunately, this vision is a long way from reality. First, there is no uniform interface to both static documents and queryable information sources; to date, research has focussed either on searching/querying static documents, or on searching/querying queryable information resources, but not on a combination of the two. Second, the attempts that have been provided for unifying views over multiple queryable information sources provide dismal scaleup characteristics. These approaches simply do not extend to the scale of the present Internet, let alone the Internet of the future. In our thesis work we propose to prototype and evaluate approaches that do present a scalable, unified view of the Internet's XML documents and queryable information resources.

We believe that we have identified a promising approach toward solving these problems. First, in our previous work, we have implemented an integration engine, IDB (Internet Data Base) that is a much more

scalable approach toward information integration than other approaches (discussed in the related work section.) Of course, this scalability does not come for free; IDB is more scalable because it solves a more targeted problem than most integration work. It is our belief that this targeted problem provides most of the functionality people will want, while avoiding the difficult 10% of functionality that makes preceding efforts non-scalable. We discuss the IDB in more detail in Section 3.

Second, as part of the NIAGRA project, we have been a member of the team that has implemented a structured search engine for static XML documents. This structured search engine, unlike commercial HTML search engines such as Hotbot or AltaVista, lets you ask for keywords in context ("find all book author elements that contain the name 'Naughton'"). This greatly improves the selectivity of the searches.

The immediate focus of our research is to integrate the XML search engine into the IDB query processing. The integration can be done by rewriting the IDB user query into a search engine query. By passing the rewritten query to the search engine, we can effectively "push down" the selection conditions to the bottom of the execution plan. In the case of querying static documents, the search engine returns only the list of urls of documents that satisfy the selection conditions and that are relevant to the user query. The IDB query engine will then retrieve the documents and generate the final answer. In the case of querying dynamic documents (queryable databases), the search engine will identify the relevant information sources and return their resource descriptions to the IDB query engine. The IDB query engine, then, will reformulate the original user query into multiple subqueries, one for each local data source, based on the resource descriptions. Next, it will integrate the results from the local data sources after each subquery is processed.

## 2 Related Work

Decades of research on *structured* (as opposed to *semistructured*) data management have been extremely successful and have yielded many fruitful results including relational, object-oriented, and object-relational database systems. Many novel techniques have been developed for efficient modeling and querying structured data. The techniques, however, can not be directly applied to the ever growing World Wide Web (WWW) to help manage the eclectic nature of data on the Web. This problem has led to a significant volume of recent research in the area of semi-structured data management. The area has been gaining popularity as the volume of data available on the Web has grown.

The semistructured data have been largely influenced by the popularity of HTML and can be characterized as follows: 1) the structure is irregular and implicit 2) the schema is often lacking, even if there is some schematic information available it is often very large, and a posteriori - as opposed to the a priori schemas of the structured data management systems.

A significant body of research on semistructured data management has focused on the application of modeling and querying the web, specifically HTML. In the early stage of research, several direct Web-query languages were proposed, including W3QL [13], Web-SQL [16]. These languages depend on text patterns and link patterns appearing in the documents to query. While these languages are document-oriented, there

are also languages querying the structure of the data. These languages, including Lorel [1], UnQL [6], and StruQL [9, 10], provide a way to access to the structure of the data and manipulate them. Especially, UnQL and StruQL went further to feature the ability of restructuring data graph, which made it possible to create a new Web-site by writing a declarative view definition<sup>1</sup> [9, 8].

In parallel to the work on modeling and querying the web, there has been a large effort to integrate heterogeneous information sources on the web. Several systems have been built with the goal of answering queries using logical views, including the Information Manifold [15, 21], and TSIMMIS [11, 21]. These systems consists of four primary components. The first component is the *global schema*. It defines the view of global information. The second component is the *source description*. It defines the view of information sources. The third component is the *wrapper* and it facilitates interaction between the source and the query execution engine. The last component is the *query engine* (or *mediator*) and it processes the user queries.

The global schema of TSIMMIS [11, 7] is generated in a bottom-up fashion from sources to mediators. Some source level changes could affect the upper level intermediate schema and they therefore change the global schema. This seriously restricts its scalability when the number of information sources is dynamically changing. The wrapper is an interface program. The creation and maintenance of wrapper programs were a serious obstacle for the scalability. Both the Information Manifold [15, 14] and TSIMMIS depend on the wrapper programs to interact with information sources.

The Information Manifold queries over global predicates while TSIMMIS queries over the schema generated from mediators. Both systems require expensive conjunctive query containment tests to generate candidate query plans. Further, the number of generated plans (for IM) or terms in plans (for TSIMMIS) are exponential in the size of the query, and the result of the query is the union of all the output of such plans.

### 3 Overview of Ongoing Work

As we have stated, the current focus of our research is to integrate the IDB query engine and the XML search engine. As discussed in Section 1, the core of the integration is the rewriting of an IDB user query into a search engine query. We push down the selection predicates in the IDB query to the bottom of the execution plan tree by evaluating the rewritten search engine query and identifying the relevant information sources. We have developed prototypes of both the IDB query engine and the XML search engine. Both prototypes are currently working as a stand-alone system.

The current prototype of the IDB query engine consists of three primary components: the **global schema** provides a standardized view of the information, the **resource description** defines the view of the information sources using ontologies defined in the global schema and describes how to interact with the sources for extracting information, and the **query engine** processes user queries in standard SQL.

---

<sup>1</sup>Both UnQL and StruQL support graph transformation queries, but only StruQL applied this feature in the specific application of Web-site creation.

The IDB global schema is a collection of IDB ontologies. An IDB ontology is a grouping of terms (or vocabularies) describing a concept. The terms in the global schema are fully reusable. When defining an ontology, one can borrow existing terms from other ontologies in the global schema as well as creating new ones. The reuse of the vocabulary is somewhat different from the inheritance concept in the object-oriented database schema. Unlike ODB schemas, an ontology can selectively inherit (or reuse) only a subset of the parent ontology. Inheritance from multiple ontologies is also allowed. The IDB global schema is an a priori schema as opposed to the a posteriori schema of TSIMMIS [19]. It imposes no restriction in describing concepts and their relations, and allows a schema designer full expressive power in the data modeling level.

In light of this, the XML Namespace [3] is a perfect candidate for the IDB global schema. By adopting the XML Namespace as the global schema representation, we can reuse large number of widely used namespaces as our schema without reinventing them. If such namespace for an ontology is not available, either the system administrator or a user of the IDB can introduce a new IDB ontology by defining a new namespace for the ontology of their interest.

The IDB interacts with information sources using resource descriptions. The role of a resource description is two fold: describing contents and capability of a source in terms of the global schema and providing information of how to extract data from the source. Two alternative formats are used for the resource description. One is the Resource Description Framework(RDF) [5] which is the XML standard for resource description. The other is the IDB resource description format which is a simplified version of the RDF. The resource description guides the query engine to extract information from the source by providing *regular tag expression* that map the set of local data into the tuples of the global schema. The regular tag expression is a combination of regular expressions and tag expressions. The tag expression is a regular expression over the alphabet of meta language tags [17].

The resource description can use terms from one or more ontologies (or namespaces) in the global schema. It need not conform to any namespace nor have any restriction on choosing a set of vocabularies from various namespaces. This allows the user to describe sources as close as possible to the original semantics of contents of information sources and avoids losing or adding information due to a lack of expressive freedom.

A user query is formulated over the global schema and the IDB query engine processes it using the resource descriptions. The query engine generates a bushy plan to optimize a multinary union operation that is the most important operation in the integration of large number of sources that belong to a global schema in the query. Every operator in the generated execution plan runs in a separate thread and uses **push model** to propagate the tuples up in the plan tree. A synchronized queue is employed to serve as a producer/consumer buffer between two operators. The IDB extended standard SQL by adding a set of predicates to facilitate convenient string containment tests.

The current XML search engine prototype consists of three primary components: the **crawler** that traverses the Web and discover new documents, the **index manager** that builds and maintains inverted index for documents discovered by the crawler, and the **query engine** that executes the search query against the

inverted index.

When a new document is discovered by the crawler, the url of the document is passed to the index manager for indexing. The index manager maintains three kinds of inverted lists: a text inverted lists, an element inverted lists, and a dtd inverted lists. Each group of inverted lists are associated with the matching lexicons. For instance, with an element name *book*, the element lexicon can identify an inverted list of all documents that contain a *book* element. Further, each entry in an inverted list is structured to record a list of positions/position pairs along with a document id to facilitate the element containment tests. It allows significantly more expressive queries than the traditional text search engine and, in turn, can identify an effective subset of documents that satisfy the selection conditions.

## References

- [1] Serge Abiteboul, Dallan Quass, Jason McHugh, Jennifer Widom, and Janet L. Wiener. The lorel query language for semistructured data. *Int. J. on Digital Libraries*, 1(1):68–88, 1997.
- [2] Ronald Beavis. The biopolymer markup language-bioml working draft proposal. <http://www.proteometrics.com/BIOML/bioml-toc.html>, 1999.
- [3] Tim Bray, Dave Hollander, and Andrew Layman. Namespaces in xml. <http://www.w3.org/TR/REC-xml-names>, 1999.
- [4] Tim Bray, Jean Paoli, and C. M. Sperberg-McQueen. Extensible markup language (xml) 1.0. <http://www.w3.org/TR/REC-xml>, 1998.
- [5] Dan Brickley, R.V. Guha, and Andrew Layman. Resource description framework (rdf) schema specification. <http://www.w3.org/TR/WD-rdf-schema>, 1998.
- [6] Peter Buneman, Susan B. Davidson, Mary F. Fernandez, and Dan Suciu. Adding structure to unstructured data. In *Proc. Int. Conf. on Database Theory*, pages 336–350, 1997.
- [7] Sudarshan S. Chawathe, Hector Garcia-Molina, Joachim Hammer, Kelly Ireland, Yannis Papakonstantinou, Jeffrey D. Ullman, and Jennifer Widom. The tsimmi project: Integration of heterogeneous information sources. In *Proc. Information Processing Society of Japan Conference*, pages 7–18, 1994.
- [8] Mary F. Fernandez, Daniela Florescu, Jaewoo Kang, Alon Y. Levy, and Dan Suciu. Strudel: A web-site management system. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 549–552, 1997.
- [9] Mary F. Fernandez, Daniela Florescu, Jaewoo Kang, Alon Y. Levy, and Dan Suciu. Catching the boat with strudel: Experiences with a web-site management system. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 414–425, 1998.

- [10] Mary F. Fernandez, Daniela Florescu, Alon Y. Levy, and Dan Suciu. A query language for a web-site management system. *SIGMOD Record*, 26(3):4–11, 1997.
- [11] Hector Garcia-Molina, Yannis Papakonstantinou, Dallon Quass, Anand Rajaraman, Yehoshua Sagiv, Jeffrey D. Ullman, Vasilis Vassalos, and Jennifer Widom. The tsimmi approach to mediation: Data models and languages. *Journal of Intelligent Information Systems*, 8(1):117–132, 1997.
- [12] Patrick Ion, Robert Miner, Stephen Buswell, Stan Devitt, Angel Diaz, Nico Poppelier, Bruce Smith, Neil Soiffer, Robert Sutor, and Stephen Watt. Mathematical markup language (mathml). <http://www.w3.org/TR/1998/PR-math-19980224/>, 1998.
- [13] David Konopnicki and Oded Shmueli. W3qs: A query system for the world-wide web. In *Proc. Int. Conf. on Very Large Data Bases*, pages 54–65, Zurich, Switzerland, 1995.
- [14] Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Query-answering algorithms for information agents. In *Proc. of the AAAI Thirteenth National Conference on Artificial Intelligence*, pages 40–47, 1996.
- [15] Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proc. Int. Conf. on Very Large Data Bases*, pages 251–262, 1996.
- [16] Alberto O. Mendelzon, George A. Mihaila, and Tova Milo. Querying the world wide web. *Int. J. on Digital Libraries*, 1(1):54–67, April 1997.
- [17] Robert C. Miller and Krishna Bharat. Sphinx: A framework for creating personal, site-specific web crawlers. In *Proc. of WWW7*, 1998.
- [18] Open financial exchange (ofx). <http://www.ofx.net/ofx>, 1999.
- [19] Yannis Papakonstantinou and Jennifer Widom Hector Garcia-Molina. Object exchange across heterogeneous information sources. In *ICDE*, pages 251–260, 1995.
- [20] Henry S. Thompson, David Beech, Murray Maloney, and Noah Mendelsohn. Xml schema. <http://www.w3.org/TR/xmlschema-1/>, 1999.
- [21] Jeffrey D. Ullman. Information integration using logical views. In *Proc. Int. Conf. on Database Theory*, pages 19–40, 1997.