What's Next in XML and Databases?

Minos Garofalakis[1], Ioana Manolescu[2], Marco Mesiti[3],
George Mihaila[4], Ralf Schenkel[5],
Bhavani Thuraisingham[6], and Vasilis Vassalos[7]

[1] Bell Labs, USA
minos@research.bell-labs.com
[2] INRIA, France
ioana.manolescu@inria.fr
[3] University of Milan, Italy
mesiti@dico.unimi.it
[4] IBM Watson Research Center, USA
mihaila@us.ibm.com
[5] Max Planck Institute, Germany
schenkel@mpi-sb.mpg.de
[6] NSF, USA
bthurais@nsf.gov
[7] Athens University of Economics and Business, Greece
vassalos@aueb.gr

1 Introduction

Since the time XML became a W3C standard for document representation and exchange over the Web, many efforts have been devoted to the development of standards, methodologies, and tools for handling, storing, retrieving, and protecting XML documents. The purpose of this panel, held during the international EDBT'2004 workshop on "*database technologies for handling XML information on the Web*" [3], is to discuss the current status of the research in XML data management and to foresee new trends towards the XML-ization of database research.

The panel included Minos Garofalakis (Bell Labs, USA), Ioana Manolescu (INRIA, France), George Mihalia (IBM Watson Research Center, USA), Ralf Schenkel (Max Planck Institute, Germany), Bhavani Thuraisingham (NSF, USA), and Vasilis Vassalos (Athens University of Economics and Business, Greece). Marco Mesiti (University of Milano, Italy) served as moderator.

This paper reports the main research topics discussed during the panel. It is opinion of the panelists that many efforts should be devoted to the definition and implementation of retrieval systems that cope with both the XQuery query language and the emerging requirements of identifying fast, approximate and ontology-based answers for Web users' queries. Moreover, the use of XML as main means for the representation and exchange of information on the Web introduce novel issues for the security and privacy communities. Finally, a lot of attention should be devoted to the issues of XML data integration, schema versioning and stream processing.

The panel conclusions are reported along with the answers to the following "controversial" questions posed to the panelists:

W. Lindner et al. (Eds.): EDBT 2004 Workshops, LNCS 3268, pp. 318–324, 2004.

1. "*Is it always so relevant XML data management for the database community*?".
2. "*Are Xschemas really used or DTDs are still predominant*?"
3. "*Will native databases have market in the next future? In which application areas?*"
4. "*There is a great emphasis on 'approximate queries for XML documents'. What are real applications that take advantages from this research*?"

2 Approaches for XML Documents Retrieval

Retrieval of XML information is a broad field of research that range from the definition and implementation of query languages for XML databases, to the definition of mechanisms for the retrieval of approximate results and the use of ontologies for the retrieval of documents distributed over heterogenous sources on the Web. Ioana, Minos and Ralf discussed key research directions on this research field.

Development and Implementation of XQuery Language. Ioana sees that research should target the XQuery language. XQuery has evolved into a very complex language. By now, it fully subsumes XPath; covers all SQL-style data transformations; allows for ordered and unordered querying, is being extended for text search, and is based on a functional-style data model. This complexity caters to widely different requirements, and makes XQuery useful in many contexts, such as: application integration, persistent database management, stream processing, document database management, information retrieval etc. This richness of application domains is likely to promises XQuery a long and fruitful future.

XQuery is soon to be issued as a W3C standard. This opens up new issues on XQuery implementations: defining storage and execution models, optimization techniques, and cost models. Among the promising applications of XML data management, warehousing of XML data and Web services deserve significant attention, due to the increase of XML data on the Web.

Data Reduction and Approximation. Minos sees the issue of XML data reduction and approximation as a promising direction for future research. With the rapid growth of available XML data, one can expect a proliferation of on-line decision support systems that enable the interactive exploration of large-scale XML repositories. In a typical exploratory session, a domain expert poses successive queries in a declarative language (such as XQuery or XSLT), and uses an appropriate visualization of the results in order to detect interesting patterns in the stored data. Obviously, the successful deployment of decision-support systems depends crucially on their ability to provide timely feedback to users' queries. This requirement, however, conflicts with the inherently expensive evaluation of XML queries which involve complex traversals of the data hierarchy, coupled with non-trivial predicates on the path structure and the value content.

Generating *fast, approximate answers* based on precomputed, concise synopses of the XML data is a cost-effective solution for offsetting the high evaluation cost of XML queries. (Of course, another natural application for such XML-data synopses is as an effective tool for generating *approximate, compile-time selectivity estimates* during the optimization of complex user queries over XML databases.) Ideally, such approximate answers are computed very fast (since they only use concise summaries of the data) and are also accurate, in the sense that they preserve the key statistical traits of the true result with low error. Users can then examine this approximate "preview", assess the information content of the true answer, and decide whether it needs to be retrieved by executing the query over the base data. Overall, by providing users with fast and accurate feedback on the form of the results, the system can reduce the number of queries that need to be evaluated in order to effectively support the data exploration task.

Clearly, the effectiveness of such an approximate XML-query answering system hinges upon the existence of accurate synopsis structures that capture the key statistical characteristics of the base XML data and can thus produce low-error approximate answers to user queries. Data-reduction and approximation problems are now fairly well understood in the context of flat, relational databases, and a number of proposals exist for building relational-data synopses (based, for example, on histograms, wavelets, or random samples) and using them to approximate SQL query results. The proposed techniques and summarization methods, however, are suitable only for flat, relational data and do not easily extend to the case of general XML hierarchies. More recent work has proposed novel, effective synopsis mechanisms for large XML documents, in the context of both XPath selectivity estimation and approximate answering of XML "twig" queries (see, for example, [2, 5–7]). Still, these proposals only represent the first steps of the database community in this exciting research area, and several important problems remain wide open. Examples of directions for future work in this area follow.

- *Summarizing XML-Document Values.* Thus far, the primary focus of XML-synopsis proposals has been on summarizing the *label-path structure* of the underlying document, typically paying little attention to the *value content* of XML elements. Some initial ideas on summarizing both structure and value content are described in [5], but only for the case of numeric values. The problem of designing effective synopsis mechanisms for XML documents with *textual values* remains open. This domain opens up many interesting research questions and possibilities for cross-fertilization of ideas and concepts from Information Retrieval.
- *Generic Framework for Characterizing XML Data Synopses.* The goal here is to identify the key dimensions of the design space for XML data synopses, and classify existing proposals along these dimensions. Such a generic framework has already been proposed for the case of relational histogram summaries and, in fact, has directly led to identifying novel, effective classes of histograms (representing unexplored "points" in the underlying design space) [8]. Of course, the design-space characterization problem becomes significantly more complex in the case of XML data synopses, since such synopses need to effectively capture *the structure as well as the values* in an XML database.

Ontology-Based Retrieval of XML Documents on the Web. Ralf moved the discussion to querying XML on the Web. Such a case is radically different from querying XML databases consisting of XML with a fixed and known schema. As there is no universal standard for representing data in XML (and it is unlikely that there will ever be such a standard), schemas used to represent data widely vary across different Web sites, and some do not provide a schema at all. Widely adopted query languages for XML like

XPath and XQuery are no longer appropriate for searching in such an environment as they cannot cope with the diversity of data. This opens up some very interesting and important research questions.

Instead of the existing query languages, Ralf thinks that we need a new query paradigm with an expressiveness between the powerful, but complex and schema-dependant XQuery and the limited, keyword-based search that today's Web search engines provide. As users don't know (and typically don't care about) how the schema looks like, queries should not explicitly specify the structure of the data, but express the "information need" of users, a kind of "find what I mean" approach. Queries should express the user's guess how the data may be structured, and it is the system's task to find documents that match this guess.

From a research point of view, this leads to the application of both structural and ontological similarity measures to match documents and queries. Additionally, as the system's notion of good results may not coincide with the user's, relevance feedback must be a core part of a system to establish user-based instead of system-induced similarity measures. Finally, as the introduction of similarities highly increases the number of potentially relevant results, any efficient system must apply algorithms for query evaluation that are optimized to compute the most relevant results first.

3 XML Data Integration, Versioning and Stream Processing

XML Data Integration. Vasilis argues that XML is uniquely well-suited to contribute to the solution of the large problem of data integration and web services integration. With the proliferation of information and the increased connectivity, the problem of putting together information from disparate, heterogeneous, autonomous data sources is ever more pressing. XML is uniquely qualified to act as the global data model for heterogeneous information: it is self-describing, which allows information from different sources to encapsulate its description, it natively supports missing or duplicate information elements, and it has a less rigid structure than the object-oriented or relational data model. This last feature is critical, as flexibility is needed to combine rich data from different sources, even if each source is rigidly structured. At the same time, flexible schema languages, most notably XML Schema, have evolved, which provide the benefits of schema-based processing to XML data without sacrificing the flexibility necessary to put together disparate information.

The use of XML for integration has been predicted since the invention of XML. Jon Bosak in [1] explains that XML will be used for "applications that require the Web client to mediate between two or more heterogeneous databases" and "applications ... which ... tailor information discovery to the needs of individual users". But the evolution of XML use followed the evolution of XML processing software. Very quickly after its introduction XML gained wide acceptance as a message exchange format. XML message exchange needs little technical infrastructure and standards beyond the XML definition and an XML parser. XML messages need to be filtered and XML data transformed, so XPath and XSLT evolved to satisfy these needs along with XPath and XSLT processors.

The underlying infrastructure for supporting ad-hoc queries and views on distributed heterogeneous data is taking longer to develop, due to the complexity of the query processing problems involved, but Jon Bosak's prediction is slowly becoming a reality, with the introduction of products such as Enosys Software's Integration Server (which now powers BEA's Liquid Data product). The whole family of XML standards, such as XQuery for querying, XML Schema for defining the structure of the views and the underlying heterogeneous information, RDF to define semantics, and WSDL to define and call remote computational services, come into play to solve the integration problem.

As for XML as a native storage and retrieval format, it has significant benefits for data that have strong semistructured characteristics such as (most importantly) bioinformatics data, and therefore a significant opportunity to take hold in such domains.

Versioning and Stream Processing. George highlighted several areas of XML research that still require considerable work. Thus, in the area of data federation, the ability to define XML views over distributed collections of both XML and relational data will be very important, since a great deal of the world's data will continue to be stored in relational databases. For this vision to become a reality, efficient algorithms for distributed XML query processing need to be developed.

In the same context, that of distributed systems and specifically of Web services-based architectures, it is also increasingly important to be able to efficiently process streaming XML data. One promising direction for improving the performance of applications based on streaming XML data is converting the data to a binary message format. In addition to dramatically reducing the parsing time, a carefully designed binary format can also embed query processing hints that will enable applications to effectively skip irrelevant portions of large messages.

Another aspect that will become critical as Web services become widespread, is the ability to reason about evolving schemas of XML messaging formats. Robust versioning schemes, with built-in provisions for backward compatibility will be essential in a world where applications designed for different versions of a messaging standard will have to interoperate seamlessly.

Finally, and probably most important, as XQuery is becoming the query language of choice for XML data, a lot of attention will need to be given to efficient indexing structures for XML data, efficient structural join algorithms, as well as advanced query optimization algorithms for XQuery.

4 Security and Privacy in the Semantic Web

Bhavani addressed open issues in the protection of XML documents in the semantic web. The major components for the semantic web include web infrastructures, web databases and services, and ontology management and information integration. A lot of work has been done in these three areas, but little consideration has been devoted to security. Since XML is used for the representation of information, and RDF as a language for describing ontology, a lot of work should be devoted to securing XML and RDF information in the these three areas.

Furthermore, XML and RDF could be used to specify policies. We also need to examine the Web Rules Language to specify security policies. Closely related to security is privacy. We also need to examine the specification and enforcement of privacy policies.

Finally we need to develop trust mechanisms for the semantic web. This also includes developing a trust negotiation language as well as developing mechanisms for enforcing trust. In summary, security, privacy and trust for the semantic web are important research areas.

5 Conclusion

A general consideration, arisen from the panel discussion (that also answer the first trivial question), is that XML data management is a relevant research direction for the database community and it will become more and more relevant in the future. This is also perceived by the increase in the number of XML or XML-ized application data sets. Application-generated data is usefully structured in XML; Web service messages follow XML syntax; business standard formats are in XML. These various XML data sources need to be organized, queried, stored, analyzed, mined for information.

The second question has been addressed by Ioana. Relying on her experience, DTDs are still predominant, due to the perceived important complexity of XSchemas (thus high learning cost). A recent study [4] shows that 40% of the XML documents on the Web have a DTD, and less than 1% have an XSchema, most of which are WAP documents. So, it seems that DTDs are there to stay.

For what concern the third question, native databases are likely to have a market in enterprise-wide warehousing of various documents, which may or may not have been initially XML, but which are amenable to an XML format reflecting their internal structure. Also, native XML databases are a precious tool when the data is highly textual (e.g. news contents, documentary databases) and/or of very complex and variable structure (e.g. biological data). Finally, it could be guessed that in some years, all the XML querying market will belong to native databases - as it seems more and more clear that the complexity of current XML query languages exceeds the capability of RDBMSs.

Ioana addressed the last question. She claims that the approximation of the document structure is useful as it allows the users to deal with a lower cognitive overhead: it is easier to "think of a simpler object", thus document. In this direction, techniques and tools from the field of Knowledge Representation and Information Retrieval are likely to be useful. Also, fuzzy querying (by means of text search-like queries) is currently getting a lot of attention, within the framework of the XQuery language.

References

1. J. Bosak. XML, Java and the Future of the Web. *WWW Journal* 2(4): 219-227, 1997.
2. J. Freire, J. Haritsa, M. Ramanath, P. Roy, and J. Simeon. StatiX: Making XML Count. *ACM SIGMOD Conf.*, 2002.
3. M. Mesiti, B. Catania, G. Guerrini, and A. Chaudhri. *DataX: International Workshop on Database Technologies for Handling XML information on the Web*. In conjunction with Int'l Conference on Extending Database Technology (EDBT 2004). March 14, 2004, Heraklion (Crete) Greece. http://www.disi.unige.it/person/MesitiM/dataX/
4. L. Mignet, D. Barbosa, and P. Veltri. The XML Web: a first study. *WWW Conf.*, 2003.
5. N. Polyzotis and M. Garofalakis. Statistical Synopses for Graph-Structured XML Databases. *ACM SIGMOD Conf.*, 2002.

6. N. Polyzotis and M. Garofalakis. Structure and Value Synopses for XML Data Graphs. *VLDB Conf.*, 2002.
7. N. Polyzotis, M. Garofalakis, and Y. Ioannidis. Approximate XML Query Answers. *ACM SIGMOD Conf.*, 2004.
8. V. Poosala, Y. Ioannidis, P. Haas, and E. Shekita. Improved Histograms for Selectivity Estimation of Range Predicates. *ACM SIGMOD Conf.*, 1996.