Databases, But Not As We Know Them

Gillian Dobbie

Department of Computer Science
University of Auckland,
Private Bag 90210, Auckland, New Zealand,
Email: gill@cs.auckland.ac.nz

The Web has become a very useful tool to disseminate information to a diverse audience. Portals and web sites are springing up with all kinds of information, e.g. Rural News (*FarmneT* 2003), Angling in Otago (*The Otago Anglers Association Inc.* 2003), and Typhoon Information (*Digital Typhoon: Typhoon Images and Information* 2003). The information can either be gathered dynamically when requested or stored in some kind of repository. If it is stored in a repository, the question that arises is "what kind of repository should be used". The obvious answer is a native XML database because XML databases have revolutionized how businesses operate. They are the hottest topic in database research since entity relationship diagrams. Or are they?

XML itself is a very simple language that allows us to format documents, allowing us to describe them in a hierarchical manner, defining our own tags to describe the different parts of the documents. We can use either DTD or XMLSchema to define the structure of these documents. Most of the early research was into how XML documents could be mapped to relational databases and vice versa. Since then companies like Oracle have included support for XML documents in Oracle9i. Such databases are called XML Enabled databases. Most recently there has been a drive to build native XML databases, that is databases that store XML documents in their native format.

In this talk, we argue that the move to native XML databases opens up new research areas, which range from how to structure the data to how to ensure reasonable performance from the new systems. We also raise questions related to whether native XML databases will solve all our problems and how new the underlying concepts really are.

What is the motivation for native XML databases? Many companies are building web sites and portals in order to disseminate information more easily. However, without careful design and management of the information, these sites can quickly become very difficult to maintain and the companies are back to square one. One way to manage the data is to store it in a relational database, but the kinds of data that people are attempting to store is not simple. Many sites now want to store video, images, sound etc. Everyone will have their own examples but the Typhoon site (*Digital Typhoon: Typhoon Images and Information* 2003) that I mention above is a good example of the use of mixed media. Another way to manage the data is to store it in an XML-enabled database. In these systems, the underlying storage is still a relational database so there must be some overhead in the mapping from one data structure to another. There may also be some issues with round-tripping.

Where can you find out more about native XML databases? The most referenced source on native XML databases seems to be a web site built and maintained by Ron Bourret (Bourret 2000). At this site, Ron describes a native XML database as a database that:

- Defines a (logical) model for an XML document – as opposed to the data in that document – and stores and retrieves documents according to that model. At a minimum, the model must include elements, attributes, PCDATA, and document order. Examples of such models are the XPath data model, the XML Infoset, and the models implied by the DOM and the events in SAX 1.0.
- Has an XML document as its fundamental unit of (logical) storage, just as a relational database has a row in a table as its fundamental unit of (logical) storage.
- Is not required to have any particular underlying physical storage model. For example, it can be built on a relational, hierarchical, or object-oriented database, or use a proprietary storage format such as indexed, compressed files.

We distinguish between native XML databases and XML-enabled databases, which are databases with front ends that map data between XML documents and their own internal data model. Ron's website is again a good source if you are interested in specific native XML databases. Chaudhri et al. (Chaudhri, Rashid & Zicari 2003) also has information about Tamino (*Tamino XML Server* n.d.) and eXist (*eXist Open Source Database* 2003).

We have also had limited experience with some open source native XML database systems. eXist is an open source system and is tightly integrated with existing XML development tools like Apache's Cocoon. The developers are very active so new versions come out every six months. We have used eXist in two projects. In one project we built a content management system. In this project we used Cocoon and XSLT to publish the web pages. eXist supports the query language XPath. eXist was well suited to this project. In another project we attempted to build a prototype access control mechanism on top of eXist. We found that the internal representation that the access control mechanism used was different from that used in eXist. X-Hive/DB is another open source system that differs from eXist in that it supports XQuery. We used it to run data mining experiments. It proved to be easy to install and use.

We will be addressing three research challenges in this talk. The first challenge involves defining the

Copyright ©2004, Australian Computer Society, Inc. This paper appeared at Fifteenth Australasian Database Conference (ADC2004), Dunedin, New Zealand. Conferences in Research and Practice in Information Technology, Vol. 27. Klaus-Dieter Schewe and Hugh Williams, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

schema for XML documents. The second addresses the issue of access control in XML databases, and finally the third addresses the issue of mathematical foundations for XML databases. Traditionally when databases are designed, the real world semantics are first captured in a conceptual model, such as the entity relationship (ER) model, and the logical and physical database models are derived from the conceptual model. If we are to follow the same process for XML databases, can we use the same conceptual data models or do we need something different? As we have seen, requirements from the real world map quite naturally to the ER data model. How well does the ER data model map to the XML or semistructured data model? We would argue that it does not map very well. Firstly the notion of hierarchy is cumbersome and even in the most extended of the ER data models, there is no concept of ordering of entities or attributes. We believe that one of the implicit concepts of ER modelling is that an entity is defined by its attributes. This is not a concept that holds in the XML data model. In fact, in the XML data model, it is assumed that entities of the same entity set are heterogeneous i.e. they can have quite different attributes. Another field that is now well understood in the area of relational databases is normalization. Is some kind of normalization necessary for XML databases? We would argue that it is. In traditional data models, normalization is used to redesign the schema such that there is controlled redundant data in the corresponding data instance. This reduces the number of insertion, deletion and update anomalies. Are these anomalies just as likely to arise in XML databases, and if so, how can they be reduced? We argue that they are just as likely to occur so we need something like normalization for XML databases. What facets of the theory of normalization can be retained for semistructured data, and what needs to be changed? Researchers have been working in this area for five or more years. A sample of the work in this area appears in (Lee, Lee, Ling & Kalinichenko 1999, Wu, Ling, Lee, Lee & Dobbie November 2001, Embley & Mok 2001, Arenas & Libkin 2003).

The second challenge that we address concerns access control in native XML databases. Traditional authorization schemes require defining complicated views on a per user basis that essentially limits access to an entire set of columns of a relation in an all or nothing fashion. In recent years, there has been renewed interest in role based access control (RBAC) models (Sandhu, Soyne, Feinstein & Youman Feb 1996) and multilevel secure (MLS) systems (Jajodia & Sandhu May 1991). The central notion of RBAC is that permissions are associated with roles, and users are assigned to appropriate roles. Typically relationships, such as the inheritance relationship can be defined between roles. The multilevel secure systems have data classified to different security levels, and database users are assigned security clearances. The multilevel secure database system in turn assures that each user gains access to only those data for which he has proper clearance.

Many of the proposals for access control for XML data and XML databases have a flavour of RBAC and MLS. Two access control languages have been defined for XML documents, XACL (*IBM Alphaworks XACL* 2002) and XACML (*Sun XACML implementation* 2003). The most complete research work in this area to date is that of Elisa Bertino (Bertino, Castano, Ferrari & Mesiti 2000).

Research into the previous challenges mentioned is progressing. Algorithms for normalization and proposals for access control mechanisms exist. However, what they lack is a formal foundation that can be used to prove the correctness of the algorithms and the mechanisms. The first work in this area was based on graph theory, and was mainly concerned with the answering of queries, e.g., (Abiteboul, P.Buneman & D.Suciu 2000). Some work in this area has addressed representing the semantics of the data, including (Buneman, Davidson, Fan, Hara & Tan 2002, Buneman, Fan & Weinstein 1998, Buneman, Fan & Weinstein 1999). This work provides a number of very solid foundations. However the constraints being modelled are closely linked with previous database data models. We believe that this is a place where basic assumptions from the database area need to be revisited and lessons could be learnt from other fields such as object-oriented modelling. Another weakness of this work is that different algorithms are based on slightly different mathematical foundations. More recent research undertaken by Mani (Mani 2003),using regular tree grammars, studies different issues with respect to data modelling with XML, in particular subtyping, mapping from XML to a relational representation, and the reverse mapping. It is possible that his work could be used as a basis for studying normalisation algorithms and access control mechanisms. Other papers that require further investigation include (Neven 2002, Hosoya & Pierce 2001).

In order to focus this talk we have concentrated the discussion on three areas in which research is currently being undertaken. There are many other areas that we haven't addressed. There are those directly related to native XML databases, such as indexing and query optimization, performance (Matthias Nicola 2003), benchmarking (Chaudhri et al. 2003), transaction processing, data mining (Braga, Campi, Klemettinen & Lanzi 2002, Meo & Psaila 2002), etc. and those in allied fields, such as data integration, extending XML (Tatarinov, Ives, Halevy & Weld 2001), storing XML documents in a relational database (Tatarinov, Viglas, Beyer, Shanmugasundaram, Shekita & Zhang 2002), etc. After writing this paper I am left pondering the following questions:

- Should we take XML databases seriously or are they a passing fad?
- Is this research limited to XML databases or does it apply to any systems that deal with XML documents?
- Is the database community targeting the right problems or is our view too data centric?
- Are we replacing the steel wheel with a wooden wheel?
- Is XML rich enough?
- Are XML databases just databases, as we know them?

These questions represent a cynical view. There are new challenges. Methods and technologies have changed in the last 30 years so we can't just dust off our old papers. Perhaps, XML isn't exactly what we want. There are some features missing but there is a hope that that based on our current research it will evolve to what is needed to represent semistructured data and that the appropriate databases will be built to efficiently manage semistructured data.

References

Abiteboul, S., P.Buneman & D.Suciu (2000), *Data On the Web-From Relational to Semistructured Data and XML*, Morgan Kaufman Publishers, San Francisco, California.

Arenas, M. & Libkin, L. (2003), An information-theoretic approach to normal forms for relational and xml data, *in* 'Proceedings of the 2003 ACM SIGMODD-SIGACT-SIGART symposium on Principles of database systems', ACM Press.

Bertino, E., Castano, S., Ferrari, E. & Mesiti, M. (2000), 'Specifying and enforcing access control policies for XML document sources', *World Wide Web* **3**(4).

Bourret, R. (2000), 'Papers about xml', http://www.rpbourret.com/xml/.

Braga, D., Campi, A., Klemettinen, M. & Lanzi, P. (2002), Mining association rules from xml data, *in* '4th International Conference on Data Warehousing and Knowledge Discovery, (DaWaK)', Vol. 2454 of *Lecture Notes in Computer Science*, Springer.

Buneman, P., Davidson, S. B., Fan, W., Hara, C. S. & Tan, W. C. (2002), Reasoning about keys for xml, *in* G. Ghelli & G. Grahne, eds, 'Database Programming Languages, 8th International Workshop, DBPL 2001, Frascati, Italy, September 8-10, 2001, Revised Papers', Vol. 2397 of *Lecture Notes in Computer Science*, Springer.

Buneman, P., Fan, W. & Weinstein, S. (1998), Path constraints in semistructured and structured databases, *in* 'Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 1-3, 1998, Seattle, Washington', ACM Press, pp. 129–138.

Buneman, P., Fan, W. & Weinstein, S. (1999), Interaction between path and type constraints, *in* 'Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 31 - June 2, 1999, Philadelphia, Pennsylvania', ACM Press, pp. 56–67.

Chaudhri, A., Rashid, A. & Zicari, R. (2003), *XML Data Management: Native XML and XML-Enabled Database Systems*, Addison Wesley Professional.

Digital Typhoon: Typhoon Images and Information (2003), http://agora.ex.nii.ac.jp/digital-typhoon/.

Embley, D. W. & Mok, W. Y. (2001), Developing xml documents with guaranteed "good" properties, *in* 'Proceedings of the 20th International Conference on Conceptual Modeling (ER 2001)', Vol. 2224 of *Lecture Notes in Computer Science*, Springer.

eXist Open Source Database (2003), http://exist.sourceforge.net.

FarmneT (2003), http://www.farm.net.nz/.

Hosoya, H. & Pierce, B. (2001), Regular expression pattern matching for XML, *in* 'Proc. of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages'.

IBM Alphaworks XACL (2002), http://www.trl.ibm.com/projects/xml/xacl/.

Jajodia, S. & Sandhu, R. (May 1991), Toward a multilevel secure relational data model, *in* 'Proc. of the Conf. on Management of Data', ACM Press, Denver, CO, pp. 50–59.

Lee, S., Lee, M., Ling, T. & Kalinichenko, L. (1999), Designing good semi-structured databases, *in* 'Proc. 18th International Conference on Conceptual Modeling', pp. 131–145.

Mani, M. (2003), Data Modelling Using XML Schemas, PhD thesis, University of California, Los Angeles.

Matthias Nicola, J. J. (2003), Xml parsing: A threat to database performance, *in* 'Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management', ACM.

Meo, R. & Psaila, G. (2002), Toward xml-based knowledge discovery systems, *in* 'Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002)', IEEE Computer Society.

Neven, F. (2002), Automata, logic and xml, *in* 'Proceedings of the 16th International Workshop of Computer Science Logic, CSL 2002, 11th Annual Conference of the EACSL', Vol. 2471 of *Lecture Notes in Computer Science*, Springer.

Sandhu, R. S., Soyne, E. J., Feinstein, H. L. & Youman, C. E. (Feb 1996), 'Role-based access control models', *IEEE Computer* **29**(2), 38–47.

Sun XACML implementation (2003), http://sunxacml.sourceforge.net/.

Tamino XML Server (n.d.), http://www.softwareag.com/tamino/.

Tatarinov, I., Ives, Z. G., Halevy, A. Y. & Weld, D. S. (2001), Updating xml, *in* 'Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data', ACM Press.

Tatarinov, I., Viglas, S. D., Beyer, K., Shanmugasundaram, J., Shekita, E. & Zhang, C. (2002), Storing and querying ordered xml using a relational database system, *in* 'Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data', ACM Press.

The Otago Anglers Association Inc. (2003), http://www.dissco.co.nz/oaa/.

Wu, X., Ling, T., Lee, S. Y., Lee, M. L. & Dobbie, G. (November 2001), NF-SS: A normal form for semistructured schemata, *in* 'Proceedings of the International Workshop on Data Semantics in Web Information Systems (DASWIS/ER2001)', Springer-Verlag.