Chapter 14
DocBase:
Design, Implementation and Evaluation of a Document Database for XML

Arijit Sengupta
Wright State University, USA

Ramesh Venkataraman
Indiana University, USA

ABSTRACT

This chapter introduces a complete storage and retrieval architecture for a database environment for XML documents. DocBase, a prototype system based on this architecture, uses a flexible storage and indexing technique to allow highly expressive queries without the necessity of mapping documents to other database formats. DocBase is an integration of several techniques that include (i) a formal model called Heterogeneous Nested Relations (HNR), (ii) a conceptual model XER (Extensible Entity Relationship), (ii) formal query languages (Document Algebra and Calculus), (iii) a practical query language (Document SQL or DSQL), (iv) a visual query formulation method with QBT (Query By Templates), and (v) the DocBase query processing architecture. This paper focuses on the overall architecture of DocBase including implementation details, describes the details of the query-processing framework, and presents results from various performance tests. The paper summarizes experimental and usability analyses to demonstrate its feasibility as a general architecture for native as well as embedded document manipulation methods.

MOTIVATION

The growth of electronic documents in the Internet era has been phenomenal. In early studies by Lawrence and Giles (1998, 1999) the approximate size of the web was reported to be about 320 million in 1997 and had grown to 800 million by 1999. With the explosive growth of the Internet that is understood to double about every five years following Moore's Law, it is hard to determine the current size of the Internet, one can easily assume that there over 10 billion unique web pages on the Internet. The primary markup language for documents on the Internet is HTML, but because

DOI: 10.4018/978-1-4666-2044-5.ch014

of its layout-driven nature and its limitations for use as a format for document interchange, new languages are being developed and used, primary among them being XML (eXtensible Markup Language) (Bray et al., 2008). XML is also being used to structure data-exchange among businesses, e.g., through the use of the ebXML standard (Grangard et al., 2001). Further, emerging web services standards such as SOAP (Gudgin et al., 2007), WSDL (Christensen et al., 2001) and UDDI (Clement et al., 2004) all use XML for achieving their required functionality. Hence, it is not surprising that XML is a key component of advanced software development frameworks such as Sun Microsystem's (now acquired by Oracle) J2EE and Microsoft's .NET, and is the backbone of emerging architectures such as Service Oriented Architecture (SOA).

Use of XML, however, is not limited to the "back end" of systems. XML is playing an increasing larger role in the area of document management. For example, many academic conferences now require that the final submissions are submitted as an XML document. This allows the proceedings to seamlessly be converted to various presentations formats (HTML, PDF etc.). At the same time, it allows for the creation of a searchable repository of these articles for use in electronic document databases, e.g., ABI/Inform or INSPEC. Thus, it is not surprising that XML documents are playing a significant role in modern day libraries (Tennant, 2002). XML is also being used to transform the way financial information is collected and reported. Extensible Business Reporting Language (XBRL) is a language to enable standardized communication of business and financial information around the world (http://www.xbrl.org). Many companies such as, Edgar Online, Reuters, Microsoft etc. are now reporting and archiving financial information using XBRL. A similar standard, XBITS (XML Book Industry Transaction Standards) is taking root in the book industry to enable "bi-directional electronic data interchanges within the book manufacturing supply chain" (http://www.idealliance.org/xbits).

With the growth in the use of XML, both in terms of quantity and variety of applications, it is important that techniques be developed that will allow for the flexible as well as efficient management of XML data and documents. In particular, there is a critical need to examine the issues surrounding the storage and retrieval of XML data.

With regard to storage, researchers have proposed techniques that range from storing XML documents using existing file-based systems (e.g., Gonnet & Tompa, 1987) to storing them in object-oriented and relational databases (e.g., Christophides et al., 1994). Native XML data management (Fiebig et al., 2002) has also emerged as a viable alternative to relational or object-oriented databases. From a querying perspective, the most common method for searching information in XML databases is using the standard released by the World Wide Web Consortium (W3C) - XQuery (Boag et al., 2007). However, given the popularity of declarative languages like SQL for querying databases, the jury is still out on whether a query language like XQuery can serve the needs of all constituencies.

Let us use a motivating example which in fact started this research. Suppose a reference librarian has acquired the Chadwyck-Healey English poetry database (Chadwyck-Healey, 1994) and needs to make the data available to patrons who have no background in XML and related standards such as XQuery, XPath, etc. A simple option would be to index the documents using a standard web search engine that will immediately allow keyword searches through the collection. However, the poems have an interesting meta-data structure that a standard web search cannot easily perform, unless the documents are converted into other formats. DocBase was designed from this perspective, with the goal that the librarian can drop the XML documents into a file system folder and start treating the documents as a database that can be queried using an SQL-based language.