Chapter I
Closing the Gap Between XML and Relational Database Technologies: State-of-the-Practice, State-of-the-Art, and Future Directions

Mary Ann Malloy
The MITRE Corporation, USA

Irena Mlynkova
Charles University, Czech Republic

ABSTRACT

As XML technologies have become a standard for data representation, it is inevitable to propose and implement efficient techniques for managing XML data. A natural alternative is to exploit tools and functions offered by relational database systems. Unfortunately, this approach has many detractors, especially due to inefficiency caused by structural differences between XML data and relations. But, on the other hand, relational databases represent a mature, verified and reliable technology for managing any kind of data including XML documents. In this chapter, the authors provide an overview and classification of existing approaches to XML data management in relational databases. They view the problem from both state-of-the-practice and state-of-the-art perspectives. The authors describe the current best known solutions, their advantages and disadvantages. Finally, they discuss some open issues and their possible solutions.

Copyright © 2009, IGI Global, distributing in print or electronic forms without written permission of IGI Global is prohibited.
INTRODUCTION

Without a doubt, the extensible markup language (XML) (Bray et al., 2006) is one of the most popular contemporary formats for data representation. It is well-defined, easy-to-use and involves various recommendations such as languages for structural specification, transformation, querying, updating, etc. This wide popularity naturally has evoked intense effort to propose faster and more efficient methods and tools for managing and processing XML data. Soon it became possible to distinguish several different directions. The four most popular approaches are: methods that store XML data in a classical file system; methods that store and process XML data using a relational database system; methods that exploit a pure object-oriented approach; and, native XML methods that use special indices, numbering schemes and/or data structures particularly suitable for the tree structure of XML data. Naturally, each of these approaches has both keen advocates and detractors who emphasize its particular advantages or disadvantages.

The situation is not good especially for file system-based and pure object-oriented methods. The former approach suffers from an inability to query without any additional pre-processing of the data; whereas the latter approach fails in particular in finding a corresponding efficient and comprehensive implementation. As expected, the highest-performance techniques are the native ones, since they are tailored particularly for XML processing and do not need to artificially adapt existing structures to a new purpose. Nevertheless, the most practically used methods exploit features of relational databases. Although researchers have already proven that native XML strategies perform much better, they still lack one important aspect: a robust implementation verified by years of both theoretical and practical effort.

If we consider this problem from an alternative viewpoint, we realize that considerable amounts of data in practical use are still stored in relational databases. Legacy relational stores are well-established and reliable enough that their existence is entrenched and they are unlikely to disappear anytime soon (Bruce, 2007). Developers must sustain existing investments in applications predicated on a relational architecture while, at the same time, adapting them to the heterogeneous and message-driven nature of XML. A typical use case may involve mapping Web document content from an XML representation into a relational database. Not only does this help insulate naïve Web clients from the underlying and perhaps less familiar XML technologies, it also positions the information for storage and query via the more mature technologies associated with RDBMSs. Alternatively, middleware may permit XML sophisticates to view and query relational contents as though they were XML documents, and vice versa. For the foreseeable future, some hybrid of these solutions is likely to be developed, although the major relational database vendors are already providing embedded XML support.

Consequently, currently there are many efforts focused on database-centric XML data management. The researchers focus on more efficient strategies to query evaluation, database vendors more and more support XML and even the SQL standard has been extended by SQL/XML which introduces a new XML data type and operations for XML data manipulation. But, although the amount of existing solutions is large, there are still unsolved problems, open issues and aspects to be improved.

Goals of the Chapter

The main aim of the chapter is to study the problem of XML and relational technology convergence from various points of view. Firstly we view the problem fairly generically, in terms of the various data sources, integration approaches, processing models and types of applications to be considered. Secondly, we review commercial off-the-shelf (COTS) offerings that might affect a business’