Chapter XC
Adaptive XML–to–Relational Storage Strategies

Irena Mlynkova
Charles University, Czech Republic

INTRODUCTION

Without any doubt, the eXtensible Markup Language (XML) (Bray et al., 2006) is currently one of the most popular formats for data representation. Its wide popularity naturally invoked an enormous endeavour to propose faster and more efficient methods and tools for managing and processing of XML data. Soon it was possible to distinguish several different directions. The four most popular ones are methods which store XML data in a classical file system, methods which store and process XML data using a relational database management system, methods which exploit a pure object-oriented approach and native methods that use special indices, numbering schemas and/or data structures proposed or suitable particularly for tree structure of XML data.

Naturally, each of these approaches has particular advantages or disadvantages. In general, especially the popularity of file system-based and pure object-oriented methods is low. The former ones suffer from inability of querying without any additional pre-processing of the data, whereas the latter approach fails especially in finding a corresponding efficient and comprehensive tool.

The highest-performance techniques are the native ones, since they are proposed 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 the scientific world has already proven that native XML strategies perform much better, they still lack one important aspect – a reliable and robust implementation verified by years of both theoretical and practical effort.

Currently there exists a plenty of existing works concerning database-based[a] XML data management. All the major database vendors support XML and even the SQL (Structured Query Language) standard has been extended by a new part SQL/XML (ISO/IEC 9075-14, 2003) which introduces new XML data type and operations for XML data manipulation. But, although the amount of existing works is enormous, there is probably no universally efficient strategy. Each of the existing approaches is suitable for selected applications, but, at the same time, there can usually be found cases when it can be highly inefficient. An illustrative example is updatability of data, where the efficient storage strategies significantly differ if the feature is required or not.

The aim of this text is to provide an overview of existing XML-to-relational storage strategies. We will overview their historical development and provide a more detailed discussion of the currently most promising ones – the adaptive methods. Finally, we will outline possible future directions.

BACKGROUND

The main concern of the database-based XML techniques is the choice of the way XML data are stored into relations, so-called *XML-to-relational mapping* or *schema decomposition to relations*. The strategy in first approaches to XML-to-relational mapping, so-called *generic* (e.g. Florescu

et al., 1999), was based purely on the data model of XML documents. The methods were able to store any kind of XML data since they viewed XML documents as general labelled trees. But, the efficiency of query evaluation was quite low due to numerous join operations or the increase of efficiency was gained at the cost of increase of space overhead.

Hereafter, the scientists came with a natural idea to exploit structural information extracted from XML schemas of XML data, usually expressed in DTD (Document Type Description) (Bray et al., 2006) or XML Schema (Thompson et al., 2004; Biron et al., 2004) language. All the so-called *schema-driven* approaches (e.g. Shanmugasundaram et al., 1999) were based on the same idea that the structure of the target relational schema can be created according the structure of the source XML schema. Assuming that a user specifies the XML schema as precisely as possible to specify the related data, we can get also more precise relational XML schema. The problem is that DTDs are usually too general. The extensive examples are recursion or * operator which, in general, enable to specify infinitely deep or wide XML documents. According to analyses of real-world XML data (Mlynkova et al., 2006) in both the cases the respective XML documents are much simpler and, thus, the effort spent on processing all the complex schema constructs is useless.

A slight solution to the problem brought schema-driven approaches exploiting information extracted from XML Schema definitions (e.g. Mlynkova et al., 2004). The XML Schema language enables to describe the structure of XML data more precisely, especially in case of data types. But, although its expressive power is higher, most of the new constructs can be considered as "syntactic sugar". Hence, exploitation of XML Schema constructs does not have a key impact on efficiency of XML processing.

A different type of improving of the fixed mapping strategies brought *constraint-preserving* methods (e.g. Chen et al., 2003; Davidson et al.,

6 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/adaptive-xml-relational-storage-strategies/20771

Related Content

CORBA-Based Run-Time Architectures for Workflow Management Systems
J. A. Miller, A. P. Sheth, K. J. Kochutand X. Wang (1996). *Journal of Database Management (pp. 16-27).*
www.irma-international.org/article/corba-based-run-time-architectures/51159

Multi-Level Modeling of Web Service Compositions with Transactional Properties
K. Vidyasankarand Gottfried Vossen (2013). *Innovations in Database Design, Web Applications, and Information Systems Management (pp. 139-170).*
www.irma-international.org/chapter/multi-level-modeling-web-service/74392

Knowledge Extraction From National Standards for Natural Resources: A Method for Multi-Domain Texts
Taiyu Ban, Xiangyu Wang, Xin Wang, Jiarun Zhu, Lvzhou Chenand Yizhan Fan (2023). *Journal of Database Management (pp. 1-23).*
www.irma-international.org/article/knowledge-extraction-from-national-standards-for-natural-resources/318456

Principles on Symbolic Data Analysis
Héctor Oscar Nigroand Sandra Elizabeth González Císaro (2009). *Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends (pp. 74-81).*
www.irma-international.org/chapter/principles-symbolic-data-analysis/20690

Unified Modeling Language: A Complexity Analysis
Keng Siauand Qing Cao (2001). *Journal of Database Management (pp. 26-34).*
www.irma-international.org/article/unified-modeling-language/3259