Chapter IV
XML Schema Evolution and Versioning: Current Approaches and Future Trends

Giovanna Guerrini
Università degli Studi di Genova, Italy

Marco Mesiti
Università degli Studi di Milano, Italy

ABSTRACT

The large dynamicity of XML documents on the Web has created the need to adequately support structural changes and to account for the possibility of evolving and versioning the schemas describing XML document structures. This chapter discusses and compares the support for schema evolution and versioning provided by commercial systems as well as the most relevant approaches and prototypes proposed and developed by the research community.

INTRODUCTION

XML is a markup language introduced by W3C (1998) that allows one to structure documents by means of nested tagged elements. XML is nowadays employed for the representation and exchange of information on the Web. Schema information can be represented through DTDs or XSDs (W3C, 2004). The XML Schema specification provides an extended set of data types and constraints that makes XSDs more suitable than DTDs to be used, for example, as a contract on the data types expected in a transaction, as a grammar for easily parsing application data, and as a database schema for document storing and query processing.

Both XML native (e.g., Tamino, eXist, TIMBER) and enabled (e.g., Oracle, IBM DB2, SQL Server) Database Management Systems (DBMSs) have been so far proposed (Bourret, 2007) for
storing and querying XML documents. Native DBMSs rely on a data model specifically conceived for the management of XML, whereas enabled DBMSs are relational or object-relational ones that have been extended for the treatment of XML. Enabled XML DBMSs are more mature than the native ones because they are supported by big vendors and the integration of XML data with other company data is easier. Some enabled DBMSs support XML Schema for the specification of a mapping between an XSD and internal relational or object-relational representation of XML documents (Florescu & Kossman, 1999). Relevant from many points of view is the use of XSDs in data management. They can be used for the optimization of query execution, for the specification of access control policies, and indexing structures.

New requirements may arise in an application domain that lead to updating the structure of data. Moreover, the integration of heterogeneous sources may require the schema to be modified. The complex structure, size, and dynamicity of schema information require exploiting database tools for their update and management.

Given an XML database, updates to the schema information, besides modifying the schema, can lead to schema evolution or to schema versioning. Schema evolution means that the original schema is replaced by an updated schema to which some update primitives are applied, and the effects of the update on instances and applications are faced. Specifically, as far as document instances are concerned, the following issues arise for a data management system: (i) how document revalidation with respect to the evolved schema can be performed efficiently; (ii) how document adaptation to the evolved schema can be (eventually automatically) performed for documents to make them valid again. Schema versioning means that original documents and schemas should be preserved and a new updated version of the schema is created. Therefore, document revalidation and adaptation are not an issue. However, the issue of handling different versions of the same schema needs to be addressed. Applications need to be able to handle document instances of different versions of the schema and queries, specified on a version of the schema, and also may need to be evaluated on other versions of the schema. A possible approach can rely on specifying mappings among schema versions.

In this chapter, we describe the facilities of main DBMSs to handle XML Schema and their level of support to schema evolution and versioning. Then, the chapter discusses how the research community is addressing the schema evolution and versioning issues and foresees future enhancement in the field.

The chapter starts by introducing the basics of XML documents and schemas as well as of evolution and versioning. Then, schema evolution and versioning of main XML-DBMSs are discussed. The most relevant research proposals related to schema evolution are then presented, and the X-Evolution system (Guerrini et al., 2007) is described as the most advanced approach for XML schema evolution and automatic document adaptation. Finally, schema versioning issues are considered and we conclude by discussing further research issues that still need to be addressed.

**BACKGROUND**

In this section, we first introduce some basic notions about XML documents and schemas and briefly present the languages for manipulating XML documents. Then, we discuss the role of schemas in XML data management systems. Finally, the concepts of schema modification, evolution, and versioning are introduced and contrasted.

**XML Documents and Schemas**

XML documents, as depicted in Figure 1(a), can be coupled with schema information either in
20 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage: www.igi-global.com/chapter/xml-schema-evolution-versioning/27777?camid=4v1


Related Content

Business Rules Management for Business Processes: From Modeling to Deployment
www.igi-global.com/chapter/business-rules-management-business-processes/35874?camid=4v1a

Design Recovery of Web Applications Transactions
Scott Tilley, Damiano Distante and Shihong Huang (2005). Advances in UML and XML-Based Software Evolution (pp. 1-19).
www.igi-global.com/chapter/design-recovery-web-applications-transactions/4928?camid=4v1a

Recent Advances and Challenges in XML Document Routing
www.igi-global.com/chapter/recent-advances-challenges-xml-document/27780?camid=4v1a

Introducing Non-functional Requirements in UML
www.igi-global.com/chapter/introducing-non-functional-requirements-uml/30540?camid=4v1a