Available online at www.sciencedirect.com

ScienceDirect

Information Systems 32 (2007) 458–476

www.elsevier.com/locate/infosys

Efficient schema-based XML-to-Relational data mapping

Mustafa Atay*, Artem Chebotko, Dapeng Liu, Shiyong Lu, Farshad Fotouhi

Department of Computer Science, Wayne State University, Detroit, MI 48202, USA

Received 2 March 2005; received in revised form 4 December 2005; accepted 15 December 2005

Recommended by: Prof J. Van den Bussche

Abstract

Storing and querying XML documents using a RDBMS is a challenging problem since one needs to resolve the conflict between the hierarchical, ordered nature of the XML data model and the flat, unordered nature of the relational data model. This conflict can be resolved by the following XML-to-Relational mappings: schema mapping, data mapping and query mapping. In this paper, we propose: (i) a lossless schema mapping algorithm to generate a database schema from a DTD, which makes several improvements over existing algorithms, (ii) two linear data mapping algorithms based on DOM and SAX, respectively, to map ordered XML data to relational data. To our best knowledge, there is no published linear schema-based data mapping algorithm for mapping ordered XML data to relational data. Experimental results are presented to show that our algorithms are efficient and scalable.

Keywords: XML; Relational; Schema-based; Ordered; Mapping; Shredding

1. Introduction

XML has emerged as a standard for representing and exchanging data over the World Wide Web. The increasing amount of XML documents requires the need to store and query XML documents efficiently. Numerous researchers have proposed using relational databases to store and query XML documents [1–9]. The main challenge of this relational approach is that one needs to resolve the conflict between the hierarchical, ordered nature of the XML data model and the flat, unordered nature of the relational data model. This conflict can be resolved by the following XML-to-Relational mappings:

- *Schema mapping*: Either a fixed generic database schema (schema-oblivious XML storage) is used, or a database schema is generated from an XML schema or DTD (schema-based XML storage) for the storage of XML documents. To support the ordered nature of the XML data model, an order encoding scheme such as those proposed in [8] can be used and additional columns are introduced to store the ordinals of XML elements.
- *Data mapping*, which shreds an input XML document into relational tuples and inserts them into the relational database whose schema is generated in the schema mapping phase.

*Corresponding author. Tel.: +1 313 5776711; fax: +1 313 5776868.

E-mail addresses: matay@wayne.edu (M. Atay), artem@wayne.edu (A. Chebotko), dliu@wayne.edu (D. Liu), shiyong@wayne.edu (S. Lu), fotouhi@wayne.edu (F. Fotouhi).

0306-4379/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.is.2005.12.008

- *Query mapping*, which translates an XML query into its relational equivalent (i.e. SQL statements or relational algebra expressions), executes them against the database and returns the query result to the user. If the query result is to be returned as XML documents, then a reconstruction algorithm [10] is needed to reconstruct the XML subtrees rooted at the matching nodes.

While existing work has focused on the problems of schema mapping [1–7,9] and query mapping [8,11–19], there is no published linear schema-based data mapping algorithm for mapping ordered XML documents to relational data. Firstly, the schema-oblivious storage schemes [1–3,16] use a simple, fixed database schema for XML storage, and the data mapping problem in this context has been addressed by Grust et al. in [20]. Secondly, while the schema-based storage schemes [4–7,9] have presented different strategies to generate a good database schema from an XML schema, there has been no published work presenting algorithms for mapping XML documents to relational data that will fit into the generated database schema and preserve the XML document order. Tatarinov et al. [8] focus on the investigation of three order encoding schemes for storing and querying XML documents. Although it presents a brief discussion of schema-based order-preserving schema mapping, no algorithmic details are given for the schema-based *data mapping*. Thirdly, existing works on query mapping [8,11–15,17–19] assume that the database has already been populated with XML documents, and no algorithms have been published for shredding XML documents into relational data in the context where the database schema is generated from an XML schema. The data translation algorithm presented in [21] does not support recursive XML schemas and does not consider the ordered nature of XML documents. The data loading algorithms defined in [16,20] support the schema-oblivious storage scheme and use a SAX-based approach. Finally, our previous data mapping algorithm presented in [22] is not order-preserving and uses only a DOM-based approach.

Since the target database schema might be complex and its corresponding XML-to-Relational schema mapping is non-trivial, it is challenging to design an efficient schema-based data mapping algorithm. This is one major motivation of our research.

The main contributions of this paper are:

1. We propose a schema mapping algorithm, *ODTDMap*, which generates a database schema from an XML DTD for storing and querying ordered XML documents. Although the main idea of *ODTDMap* is similar to the shared inlining algorithm [4,8] and its variant [9], *ODTDMap* makes several improvements over them as discussed at the end of Section 4.
2. We propose an efficient DOM-based linear data mapping algorithm, *OXInsert*, which shreds and composes input XML documents into relational tuples and inserts them into the relational database according to the schema generated by *ODTDMap*. *OXInsert* is based on our previous data mapping algorithm *XInsert* [22], but it takes into account the ordered nature of the input XML documents and set-valued attributes that were not considered by *XInsert*.
3. We propose an efficient and linear SAX-based data mapping algorithm, *SDM*, which shreds and composes ordered XML documents into relational tuples and inserts them into the relational database according to the schema generated by *ODTDMap*.

Our experimental study shows that the proposed algorithms *ODTDMap*, *OXInsert*, and *SDM* are efficient and scalable. We show that our data mapping algorithms *OXInsert* and *SDM* are efficient under different schema mapping algorithms other than *ODTDMap* in the experimental study.

Although query mapping is an essential part of a complete mapping scheme, mapping XML queries into their SQL counterparts is not the focus of this paper. We refer the interested readers to recently proposed query mapping algorithms [8,11,12,14,15,17–19].

We assume the reader is familiar with XML [23] and its related technologies, such as DTD [23], DOM [24] and SAX [25].

Organization: The rest of the paper is organized as follows. Section 2 presents an overview of related work. The formalization of a schema-based relational XML storage system is given in Section 3. Section 4 gives a brief description of our schema mapping algorithm *ODTDMap*. Section 5 identifies the main issues for data mapping and describes our proposed data mapping algorithms *OXInsert* and *SDM*. Section 6 presents an experimental study of the time performance of *ODTDMap*, *OXInsert* and *SDM* algorithms. Finally, Section 7 concludes the paper and points out some potential future work.