Updating Relational Databases through XML Views

Vanessa de Paula Braganholo
Advisor: Carlos A. Heuser
Universidade Federal do Rio Grande do Sul - UFRGS
Instituto de Informática
e-mail: {vanessa,heuser}@inf.ufrgs.br

Abstract

This paper presents an overview of a process for updating relational databases through XML views. The process is based on a query language called R2X and on a concept of normalization for XML views. The R2X language constructs XML views from relational databases, while normalization removes the ambiguity in an XML view, thus solving the update problem. In order to normalize an XML view, we benefit from the relation between XML and non-first normal form relations (NF^2).

Keywords: XML views, update through views, non-first normal form relations

1 Introduction

Since its introduction, XML quickly became the universal format for publishing and exchanging data on the Web. Despite that, most of corporate data still resides in relational databases. This is due to the maturity of indexing, query processing and storing techniques of the relational systems.

In recent years, several types of applications appeared, where data stored in relational databases need to be translated to XML. Examples are WEB applications that separate content from presentation. Applications of this type represent content internally in XML and present data to the user in HTML. Another example are applications that use XML to interchange data.

In order to implement such types of applications, tools for generating XML views from relational data sources have been proposed in the literature. In this work, these tools are called *wrappers* because they can be thought as an intermediate layer between a user application and a relational database. These tools can be classified in two types.

Tools that use SQL to construct the views and map the relational view to XML. Such approach is suitable for users who are familiar with the relational model and the SQL language. In this approach the XML wrapper has a single task: the construction of XML instances from the result of SQL queries. Examples of such tools are Oracle XML SQL Utility [22], DB2XML [26], extension of SAX e DOM APIs [17], ODBC2XML [13], IBM DB2 XML Extender [6] and the XML/SQL language [27], among others.

Tools that map the relational model to XML and use XML query languages to build the views. This type of solution is suitable for users who are familiar with the XML model and its query languages. However there is a big semantic gap between queries in an XML language and the corresponding query in SQL. Due to this fact the construction of an XML wrapper in this approach is not a simple task. Examples of such tools are SilkRoute [9] and Xperanto [5].

Most of the existing proposals present limitations such as constraints on the nesting of XML elements and the need of procedural specification of the mapping from SQL to XML.

Another problem consists in updating the relational database through XML views. From the application point of view, it is natural that the changes made in the XML view be automatically mapped to the underlying relational database. However, few of these tools supply this kind of support [25, 22, 6, 28]. Additionally, tools that support updates generally present restrictions regarding

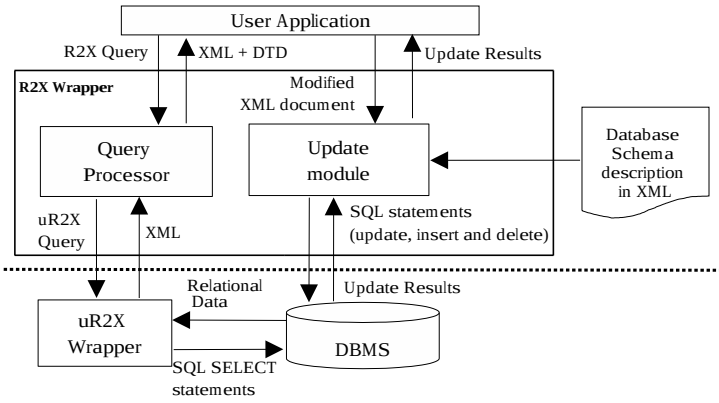

Figure 1: R2X wrapper architecture

the way changes are submitted, the kind of view that can be updated (usually only plan views) and also the kind of update operations that can be performed (some are restricted to only insertions).

This work fits in this context, since it proposes a mechanism to update relational databases through XML views. The mechanism is based in a language that constructs *updatable* XML views over relational databases. This language is called R2X (***R**elations **to X**ML*). It was designed over a more generic language called uR2X (***u**nderlying **R**elations **to X**ML*). uR2X [3] is an evolution of XML/SQL [27] and it constructs *read-only* XML views with arbitrary structure and nesting. R2X is also an evolution of a previous version [4]. Both R2X and uR2X follow the first approach, that is, they use SQL to build the views and map the relational view to XML.

This work is structured as follows. Section 2.1 presents the relation of R2X and NF^2 relations. Section 2.2 shows the update operations that can be performed in an XML view. Section 2.3 describes the next steps in the R2X design. Finally, section 3 presents some concluding remarks.

2 R2X Language

Besides constructing updatable XML views, R2X defines a mechanism to update relational databases through XML views. This mechanism is based in a wrapper architecture (Figure 1). In this architecture, a user application submits an R2X query to the wrapper. The wrapper processes the query and sends the resulting XML view and its DTD to the application. The user application modifies the XML view according to the DTD and sends it back to the wrapper. The wrapper identifies the changes performed on the XML view and generates update SQL statements so that the changes can be reflected in the database.

2.1 R2X and NF ^{2}

One of the design goals of R2X is the construction of updatable XML views with arbitrary nesting. In the database research area, the concept of nesting is not new, and appeared in the non-first normal form approaches (NF^2) [20, 1, 14].

With this in mind, we decided to use NF^2 relations in the R2X design. This decision was made because NF^2 relations can be easily mapped to XML [18]. Besides that, a normalized relation can be transformed into a non-normalized relation by the use of the *nest* operator (ν). The nest operator partitions relations and creates nested relations for each partition formed [24].

In this way, we defined an implementation model for R2X that works as follows. A view definition query expressed in classical relational algebra is transformed by a series of nest operations, obtaining an NF^2 view. This view is than mapped to XML (Figure 2). This process is based in the specification of R2X query language, which uses classical relational algebra and nest operations. Due to space restrictions, the R2X specification will not be shown here. Further details can be found in [3].

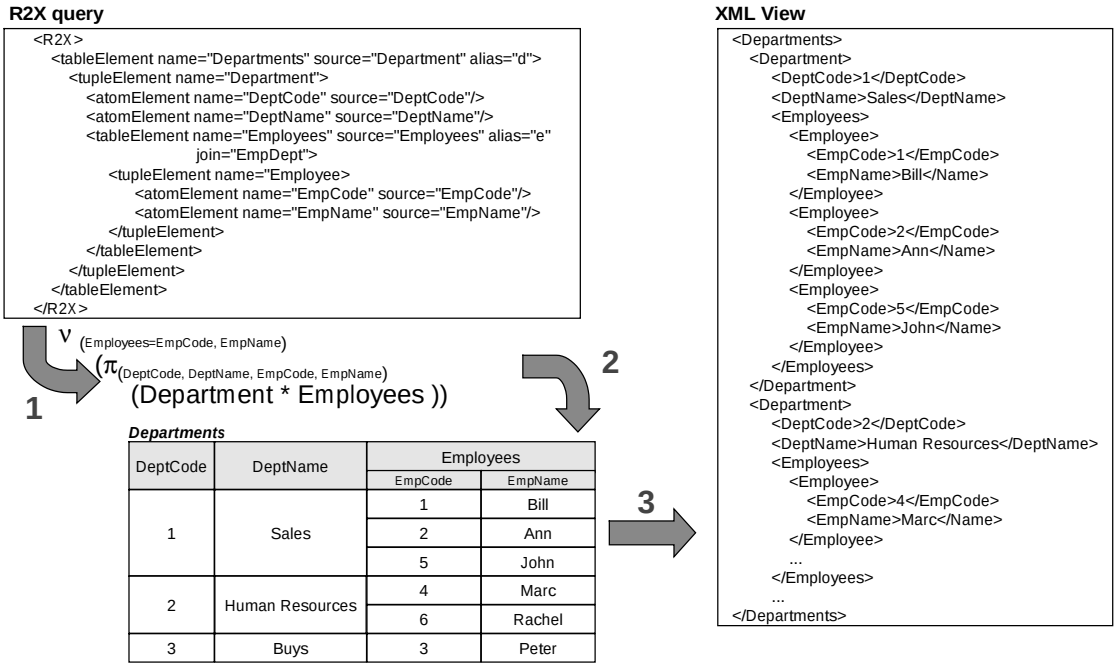

Figure 2: R2X and NF^2

2.2 Update Operations

R2X also defines the update operations that can be performed by a user application in the XML views.

INSERT: An insert operation consists of inserting a sub-tree in the XML instance.

DELETE: A delete operation removes a sub-tree of the XML instance.

UPDATE: An update operation modifies the content of an XML element.

MOVE: A move operation moves a sub-tree from a point in the XML instance to another point in the same instance.

These operations modify the content of the XML views, but they can not modify the view structure, that is, all the modifications made in the view have to be made according to the view DTD.

Examples of these operations are described in [4]. It is important to notice that these operations suppose that the XML view is updatable.

2.3 R2X Main Goals

For the R2X implementation model to be complete, it is necessary to *guarantee* that R2X produces only *updatable* views. The big problem in updating relational databases through views resides in the ambiguity that views can hold [7]. By removing the ambiguity, we can guarantee that the view is updatable. One alternative to obtain this guarantee is to study normal forms for NF^2 relations [20, 23, 12, 19], and among them, to choose one that best fits in the language. The properties of the chosen normal form must be checked against the obtained view, making that only views that follow these properties be accepted.

More specifically, to guarantee that R2X produces only updatable views, the following tasks must be accomplished.

Definition of functional dependency. It is necessary to define the functional dependency that will be used in the normal form definition. There are several definitions in literature. Among them are the ones used in the normal form definitions [20, 23, 12, 19] and the "Nested Functional Dependency", defined by [11].

Definition of a concept of normal form. Based in the functional dependency defined in the previous step, the concept of normal form that will be used in the thesis must be defined. Several

normal forms for NF^2 relations were proposed in literature: *Normal Form* (NF) [20], *Nested Normal Form* (NNF) [23], *Partitioned Normal Form* (PNF) [12] e NF-N3 [19], among others. There are also proposals of normal forms for semistructured schemas: *Normal Form for Semistructured Schema* (NF-SS) [29] e *XML Normal Form* (XNF) [8]. These proposals must be analyzed to check if they can be adapted to NF^2 relations.

Definition of the relation between XML and NF^2. In order to use NF^2 in the language specification, we formally defined the relation between XML and NF^2 [3].

Definition of what kinds of views are updatable. Using the concept of normal form defined in the second step, a method of identifying classes of updatable views and classes of non-updatable views must be defined. More specifically, it is necessary to verify what classes of views accomplish the properties of the normal form defined. Besides that, other features of updatable views must be observed. In literature, several works define classes of updatable views, and mechanisms to update databases through views. A classical work is presented in [16]. In addition, lots of works presents the properties a view must have in order to be updatable [15, 2, 21, 10].

The proposal presented here also includes the extension of the R2X language to support views with additional semantics. There are several applications in which a view has a specific semantic. They usually construct views according to an attribute value. A classical example is a view that shows the employees that plays in the company soccer time. In this case, the criteria for the view construction is *"play-soccer-time = yes"*. This makes the semantic of the update operations to be different from the usual. The exclusion of an employee in the XML view must be mapped to an update in the value of the *"play-soccer-time"* column to *"no"* rather than to the exclusion of the employee. Allowing these kinds of views would make R2X a more powerful language.

3 Concluding remarks

This paper presented a brief overview of a proposal for updating relational databases through XML views. So far, we have proposed the R2X language for building updatable XML views over relational databases and defined an implementation model that uses NF^2 relations. The next steps include using as much as possible the existing proposals for NF^2 relations in order to guarantee the generation of *only updatable XML views*. This process implies mapping NF^2 relations to XML.

The architecture shown in Figure 1 is being implemented in Java. Also, a web version will be available, allowing users to submit queries, modify the resulting XML view and submit the modified view back to the wrapper. The modifications will then be reflected in the database.

References

[1] ABITEBOUL, S., AND BIDOIT, N. Non first normal form relations to represent hierarchically organized data. In *PODS* (1984), pp. 191–200.

[2] BANCILHON, F., AND SPYRATOS, N. Update semantics of relational views. *ACM Transactions on Database Systems 6*, 4 (Dec 1981).

[3] BRAGANHOLO, V. Updating relational databases through xml views. Thesis proposal - in preparation, PPGC-UFRGS, Porto Alegre, 2002. Available at www.inf.ufrgs.br/∼vanessa/artigos/PropostaTese.pdf.

[4] BRAGANHOLO, V., HEUSER, C., AND VITTORI, C. Updating relational databases through xml views. In *Proceedings of Third International Conference on Information Integration and Web-based Applications & Services - IIWAS 2001* (Linz, Austria, 2001), pp. 85–94.

[5] CAREY, M. J., FLORESCU, D., IVES, Z. G., LU, Y., SHANMUGASUNDARAM, J., SHEKITA, E. J., AND SUBRAMANIAN, S. N. Xperanto: Publishing Object-Relational Data as XML. In *Third International Workshop on the Web and Databases* (Dallas, Texas, may 2000).

[6] CHENG, J., AND XU, J. Ibm db2 xml extender: An end-to-end solution for storing and retrieving xml documents. In *Proceedings of ICDE'00* (2000).

[7] DATE, C. J. *An Introduction to Database Systems*, 7th ed. Addison Wesley, 2000.

[8] EMBLEY, D. W., AND MOK, W. Y. Developing xml documents with guaranteed 'good' properties. In *Conceptual Modeling - ER 2001* (Yokohama, Japan, nov 2001), H. S. Kunii, S. Jajodia, and A. Solvberg, Eds., Springer, pp. 426–441.

[9] FERNÁNDEZ, M., TAN, W.-C., AND SUCIU, D. Silkroute: Trading between relations and xml. In *Proceedings of the Ninth International World Wide Web Conference* (2000).

[10] FURTADO, A. L., AND CASANOVA, M. A. Updating relational views. In *Query Processing in Database Systems*, W. Kim, D. S. Reiner, and D. S. Batory, Eds. Springer, Berlin, Heidelberg, 1985, pp. 127–142.

[11] HARA, C., AND DAVIDSON, S. Reasoning about nested functional dependencies. In *ACM Symposium on Principles of Database Systems (PODS)* (Philadelphia, Pennsylvania, may 1999).

[12] HULIN, G. On restructuring nested relations in partitioned normal form. In *16th VLDB Conference* (Brisbane, Australia, 1990), pp. 626–636.

[13] INTELLIGENT SYSTEM RESEARCH. Odbc2xml: Merging odbc data into xml documents. Available at http://www.intsysr.com/odbc2xml.htm, 2001.

[14] JAESCHKE, G., AND SCHEK, H. J. Remarks on the algebra of non first normal form relations. In *ACM SIGACT-SIGMOD Symposium on Principles of Database Systems* (1982), pp. 124–138.

[15] KELLER, A. M. Algorithms for translating view updates to database updates for views involving selections, projections, and joins. In *Fourth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems* (Portland, Oregon, Mar. 1985), ACM, pp. 154–1i63.

[16] KELLER, M. The role of semantics in translating view updates. *IEEE Computer 19*, 1 (Jan. 1986), 63–73.

[17] LADDAD, R. Xml apis for databases: Blend the power of xml and databases using custom sax and dom apis, January 2001. Available at http://www.javaworld.com/javaworld/jw-01-2000/jw-01-dbxml.html.

[18] LAENDER, A. H. F., RIBEIRO-NETO, B., DA SILVA, A. S., AND SILVA, E. S. Representing web data as complex objects. In *Proceedings of the 1st International Conference on Electronic Commerce and Web Technologies (EC-Web 2000)* (Greenwich, UK, Nov 2000), pp. 216–228.

[19] LING, T. W. A normal form for sets of not-necessarily normalized relations. In *22nd Hawaii International Conference on System Sciences* (Kailua-Kona, Hawaii, United States, jan 1989), IEEE Computer Society Press, pp. 578–586.

[20] MAKINOUCHI, A. A consideration on normal form of not-necessarily-normalized relation in the relational data model. In *VLDB* (Tokio, Japan, 1977), pp. 447–453.

[21] MEDEIROS, C., AND TOMPA, F. Understanding the implications of view update policies. In *13th International Conference on Very Large Databases* (1985), pp. 316–323.

[22] ORACLE CORPORATION. Oracle xml sql utility. Oracle Corporation, 2001. Available at http://technet.oracle.com/.

[23] OZSOYOGLU, Z. M., AND YUAN, L.-Y. A new normal form for nested relations. *ACM Transactions on Database Systems 12*, 1 (Mar. 1987), 111–136.

[24] ROTH, M. A., KORTH, H. F., AND BATORY, D. S. Sql/nf: A query language for ¬1nf relational databases. *Information Systems 12*, 1 (1987), 99–114.

[25] TATARINOV, I., IVES, Z., HALEVY, A., AND WELD, D. Updating xml. In *Proceedings of SIGMOD 2001* (Santa Barbara, California, May 2001).

[26] TURAU, V. Making legacy data accessible for xml applications, 1999. Available at http://www.informatik.fh-wiesbaden.de/∼turau/.

[27] VITTORI, C., DORNELES, C., AND HEUSER, C. Creating xml documents from relational data sources. In *Proceedings of EC-Web 2001* (Munich, Germany, Sep 2001), pp. 60–70.

[28] WAHLIN, D. Leveraging sql server's xml features. *XML Magazine 1*, 5 (2000).

[29] WU, X., LING, T. W., LEE, S. Y., AND MONG LI LEE, G. D. Nf-ss: A normal form for semistructured schema. In *International Workshop on Data Semantics in Web Information Systems - DASWIS 2001* (Yokohama, Japan, nov 2001), pp. 146–159.