Tables and Trees Don't Mix (very well)

Erik Wilde
ETH Zürich

ABSTRACT

There are principal differences between the relational model and XML's tree model. This causes problems in all cases where information from these two worlds has to be brought together. Using a few rules for mapping the incompatible aspects of the two models, it becomes easier to process data in systems which need to work with relational and tree data. The most important requirement for a good mapping is that the conceptual model is available and can thus be used for making mapping decisions.

Categories and Subject Descriptors: H.2.1 [**Information Systems**]: Database Management — *Logical Design*

General Terms: Design, Management

1. INTRODUCTION

The predominant method used for data modeling today is based on the *relational model* first proposed by CODD [2] for this application area. Many applications today are based on relational data models, and many modeling methodologies include a way to map their data model to a relational model.

On the contrary, XML-based data exchange is based on XML's model of trees, and XML expresses most things via hierarchy and sequence, which are both absent from the relational model. XML also allows references across the tree structure, but in many scenarios these references are only weakly defined and may not cross document boundaries.

The relational model and XML's tree model do not map very well. There have been several approaches to define models for XML, and this is possible because XML is only a syntax and does not prescribe any data model. However, when working in an environment with applications based on relational structures and XML-oriented data exchanges based on tree structures, there often arises the problem of how to map the models on the conceptual layer (regardless of the underlying formalism). In this paper, the principal mapping problems are described, and possible solutions are suggested.

The approach of mapping models on the conceptual layer is very different from the approach to map a conceptual model in one domain to a logical model in the other domain. This has been done by a generic mapping of relational data to XML structures by SHANMUGASUNDARAM et al. [3], and by a generic mapping of XML data to relational structures by AMER-YAHIA et al. [1].

The conceptual mapping approach presented here attempts to transfer as much model information as possible. The goal is to start with a model inside one domain and design a "good" counterpart of it in the other domain. Since there is no "XML modeling language" available today, we assume that the "XML model" is some kind of schema definition, such as an XML Schema, accompanied by documentation.

WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
ACM 1-59593-332-9/06/0005.

2. FROM TREES TO TABLES

When starting from an XML-based model and trying to map it to a relational model, the biggest problem is the hierarchical nature of XML, which has no direct correspondence in the relational world.

2.1 Relationship vs. Hierarchy

Problem: Relationships are expressed in hierarchy as well as in node-to-node references. Because of XML's tree-based nature, only 1:n relationships can be expressed hierarchically, others must be mapped to node-to-node references. However, some hierarchical relationships are pure container issues and not data model relationships.

Solution: All relationships (hierarchies and references) in an XML model must be analyzed and classified. In many cases the documentation will be an indispensable source of information for this process.

2.2 Choices

Problem: XML content models may contain choices, which allow hierarchies to specify alternatives of content.

Solution: Choices have to be classified according to the content contained in the choice. If the choice contains a small list of simple content models, it may make sense to model it as a set of attributes with additional constraints. If the choice is large and/or has complex content, there is no generally applicable mapping to relational structures.

2.3 Ordered Content

Problem: XML's trees are inherently ordered, and in all cases where the order is not predetermined by the schema, this order may convey relevant information (in case of XML Schema this applies to `all` groups and any particle with `maxOccurs` > 1).

Solution: If the order is relevant (and not just an artifact of the XML encoding), it can be represented by sequence numbers attributed to the individual particles. This method is easy to implement, but using the sequence information in queries and in particular any update of sequence-numbered information may result in significant performance issues.

2.4 Mixed Content

Problem: Mixed content is a special case of ordered content, where some of the children are text nodes instead of elements. Mixed content is an important concept for document-oriented XML, and since it usually is specified in an open way (as inherited from the poor ability of DTDs to restrict mixed content), it is a relationship with a lot of variability.

Solution: In most cases, it is not a viable solution to map mixed content to various tables containing text children and all possible elements of the mixed content. One solution of modern databases is to use `XML` attributes (as introduced by SQL/XML), but their content cannot participate in relationships. If the relationships in mixed content should be represented in the relational model, the mixed content either has to be fully shredded into relational tables, or the relevant part of mixed content may be redundantly stored in an additional table, while the complete mixed content is still retained in one column. This latter solution is very efficient for queries, but rather expensive for updates.

3. FROM TABLES TO TREES

When starting from a relational model, a straightforward mapping of tables to trees can be used. This, however, may lead to XML that is awkward to work with, and using model information, the mapping can be improved. Some of XML's strengths (inherent ordering and mixed content) will not be implemented at all, because they do not have counterparts in the relational model.

3.1 Models vs. Documents

Problem: The relational model is not confined to any specific container, while XML is tightly bound to the concept of documents, which are self-contained units of data.

Solution: Anything going beyond the boundary of one document cannot be easily specified using today's schema languages. XML is still very document-centric, and there is no established framework for inter-document references or even inter-schema dependencies. Anything within these areas must be documented and implemented by hand, and thus should be avoided if possible.

3.2 Relationship Strength

Problem: Relationships in the relational world have to mapped to the tree-based model of XML, where relationships can be expressed by hierarchy or by references. Deciding which kind of XML relationship should be used is an important aspect of creating an adequate XML model.

Solution: When mapping relationships, their "strength" should be considered. For example, UML's distinction of *association*, *aggregation*, and *composition* is a good guideline. Compositions are good candidates for hierarchies, while aggregation is better mapped as references.

3.3 1:1 Relationships

Problem: 1:1 relationships are symmetric in the sense that they associate exactly two entities. The mapping question is which of the entities should make the XML reference, and which should be the target?

Solution: If the relationship has a well-defined directionality, then the source should carry the reference, and the target should be referenced. If there is no well-defined directionality, it may be possible to make assumptions about the more frequent usage pattern, and to model the XML accordingly.

3.4 Relationship Multiplicities > 1:1

Problem: If a relationship has multiplicities greater than one, but is not a composition and thus the entity should not be embedded, then there must be a way to create more than one reference.

Solution: The DTD mechanism of `IDREFS` allows multiple references, but does not allow to specify the targets or to limit the number of references. In XML Schema, using individual elements with attributes for each reference, the multiplicity can be specified using `maxOccurs`, and the reference's target can be specified using an identity constraint.

3.5 Relationship Multiplicities > 1:n

Problem: If a relationship has both multiplicities greater than one, then the pattern described above can be used, but now each referenced entity may occur in more than one reference.

Solution: DTDs and XML Schema do not allow to specify the constraint that a node should be referenced a specified number of times. To specify this constraints, additional methods are required, either implementation in the application logic, or complementary schema languages which provide better constraint specification capabilities.

3.6 Relationships with Attributes

Problem: If a relationship has attributes, the reference-based approach presented above does not work, because the references cannot carry any additional information.

Solution: In this case, relationships must be modeled as standalone components, containing their attributes. The connections with entities are then implemented by using references. This leaves open the question whether the relationship should have an ID and should be referenced by the entities, or vice versa. The decision how to represent this should be guided by the relationship's directionality and assumptions about the most frequent usage pattern for the relationship.

4. CONCLUSIONS

The list of mapping problems presented here is a short overview of the general problem of how to model and process data in an environment using relational and tree-based components. XML's increasing popularity highlights some of the areas where XML technologies so far have failed to deliver practical solutions, and one of the most important of these areas is how to model XML. By investigating the mismatches and possible mappings, the problem of how to integrate XML-oriented data and relational data models can be approached from a practical point of view.

5. REFERENCES

[1] Sihem Amer-Yahia, Fang Du, and Juliana Freire. A Comprehensive Solution to the XML-to-Relational Mapping Problem. In Alberto H. F. Laender, Dongwon Lee, and Marc Ronthaler, editors, *Proceedings of the 6th ACM International Workshop on Web Information and Data Management*, pages 31–38, Washington, D.C., November 2004.

[2] Edgar F. Codd. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13(6):377–387, June 1970.

[3] Jayavel Shanmugasundaram, Eugene Shekita, Rimon Barr, Michael Carey, Bruce Lindsay, Hamid Pirahesh, and Berthold Reinwald. Efficiently Publishing Relational Data as XML Documents. *The International Journal on Very Large Data Bases*, 10(2-3):133–154, December 2001.