

Chapter X

Integrity Constraint Checking for Multiple XML Databases

Praveen Madiraju

Marquette University, USA

Rajshekhar Sunderraman

Georgia State University, USA

Shamkant B. Navathe

Georgia Institute of Technology, USA

Haibin Wang

Winship Cancer Institute, USA

ABSTRACT

Global semantic integrity constraints ensure the integrity and consistency of data spanning distributed databases. In this chapter, we discuss a novel representation technique for expressing semantic integrity constraints for XML databases. We also provide the details of XConstraint Checker, a general framework for checking global semantic constraints for XML databases. The framework is augmented with an efficient algorithm for checking these global XML constraints. The algorithm is efficient for three reasons: 1) the algorithm does not require the update statement to be executed before the constraint check is carried out; hence, we avoid any potential problems associated with rollbacks, 2) sub constraint checks are executed in parallel, and 3) most of the processing of algorithm could happen at compile time; hence, we save time spent at run-time. As a proof of concept, we present a prototype of the system implementing the ideas discussed in this paper.

INTRODUCTION

XML (eXtensible Markup Language) has now been adopted as a standard for representation and exchange of data on the web. XML based data exchange occurs in many applications such as finance, health, e-commerce and other application areas. A major goal of a database is to ensure consistency of the data. Integrity constraints are rules which guarantee the consistency of a database. We consider XML constraints in the setting of distributed XML databases. A single update (XUpdate (Tatarinov *et al.*, 2001), (Laux & Martin, 2000)) on one site might cause a global constraint (*global XConstraint*) to be violated. By global XConstraints, we mean global semantic integrity constraints affecting multiple XML databases. We need an approach to check for such constraint violations. In the XML database setting, the majority of the times, users are interested in generating (updating), integrating and exchanging data. So, frequent updates on XML data may cause frequent global constraint violations. *Hence we need an approach that will efficiently and speedily check for such global constraint violations.*

There are two major approaches to this problem. The *first* would be to translate the XML document into relational data using methods such as those found in Shanmugasundaram *et al.* (1999), Chen *et al.* (2003) and Fong and Wong (2004). And then, map the updates and constraints on the XML data to corresponding updates and constraints on the relational data (Chen *et al.*, 2002a). Now the problem of constraint checking on XML data is pushed to the problem of constraint checking on relational data. There are well established models for constraint checking in the relational world. However, this approach suffers from the overhead cost involved in transforming XML data into relational data (Kane, Su & Rundensteiner, 2002). The *second* approach would be to check for constraint violations on the XML data without

transforming to relational data. It should be noted that using the *first* approach vs. *second* depends on the application being considered. If the application contains millions of records and if it benefits to use relational database features such as querying, fast indexing, etc., it is worth while to consider the first one otherwise the second approach suffices for a normal sized application. In this chapter, we consider the second approach.

A naïve solution would first update an XML document and then check for constraint violations. If a constraint is violated, we can rollback. However, such a naïve solution suffers from the overhead of time and resources spent on rollback. Also, the update statement is checked against all the constraints with the total new updated database state. However, in an incremental constraint checking strategy (Fan, 2005), (Bouchou *et al.*, 2005), constraints are checked incrementally only on the updated document. Hence, we need an approach that would check for constraint violations before updating the database and therefore obviates the need for rollback situations.

In our constraint checking procedure, constraint violations are checked at compile time, *before* updating the database. Our approach centers on the design of the *XConstraint Checker*. Given an XUpdate (Tatarinov *et al.*, 2001), (Laux & Martin, 2000) statement and a list of global XConstraints, we generate *sub XConstraint* checks corresponding to local sites. Sub XConstraint is an XML constraint, expressed as an XQuery, local to a single site (more details in Section 4). The results gathered from these sub XConstraints determine if the XUpdate statement violates any global XConstraints. Our approach is *efficient*; since we do not require the update statement to be executed before the constraint check is carried out and hence, we avoid any rollback situations. Our approach achieves *speed* as the sub constraint checks can be executed in parallel.

18 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/integrity-constraint-checking-multiple-xml/4298

Related Content

An Event-Oriented Data Modeling Technique Based on the Cognitive Semantics Theory

Dinesh Batra (2012). *Journal of Database Management* (pp. 52-74).

www.irma-international.org/article/event-oriented-data-modeling-technique/76666

A Quantitative Function for Estimating the Comparative Values of Software Test Cases

Yao Shi, Mark L. Gillenson and Xihui Zhang (2022). *Journal of Database Management* (pp. 1-33).

www.irma-international.org/article/a-quantitative-function-for-estimating-the-comparative-values-of-software-test-cases/299559

Resource Discovery Using Mobile Agents

M. Singh, X. Cheng and X. He (2009). *Semantic Mining Technologies for Multimedia Databases* (pp. 419-448).

www.irma-international.org/chapter/resource-discovery-using-mobile-agents/28844

Measuring Data Quality in Context

G. Shankaranarayanan and Adir Even (2009). *Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends* (pp. 385-395).

www.irma-international.org/chapter/measuring-data-quality-context/20723

Modeling and Analyzing Perspectives to Support Knowledge Management

Jian Cai (2007). *Research Issues in Systems Analysis and Design, Databases and Software Development* (pp. 185-205).

www.irma-international.org/chapter/modeling-analyzing-perspectives-support-knowledge/28437