Formal Framework of XML Document Schema Design

Zurinahni Zainol, University of Hull, UK, & Universiti Sains Malaysia, Malaysia
Bing Wang, University of Hull, UK

ABSTRACT

Designing “good” XML documents is a very difficult task for a database designer. Although many theories for XML database design have proposed, none of commercial design tool for XML document design has been developed to assist the XML document designer. In this paper, the authors present a formal framework of XML document design by incorporating a conceptual model of XML schema called Graph-Document Type Definition (G-DTD) with a theory of database normalization. This framework is designed as a blueprint to help the XML database designers to perform the XML document schema design quickly and accurately. The G-DTD is used to describe the structure of XML documents at the schema level. A set of normal forms for G-DTD on the basis of rules proposed by Arenas and Libkin and Lv. et al is used to provide a guideline to a well-designed schema for XML documents. They develop a prototype of XML document schema design using a Z formal specification language. Finally, using a case study, this formal specification is validated to check for correctness and consistency of the specification. Thus, this gives a confidence that the authors’ prototype can be implemented successfully to generate an automatic XML document design.

Keywords: Document Type Definition (DTD), Formal Method, Graph-Document Type Definition (G-DTD), Normal Form, XML Design

1. INTRODUCTION

With the wide utilization of the web and the availability of a huge amount of electronic data, XML (eXtensible Markup Language) has been used as a standard means of information representation and exchange over the Web. It usage has increased extensively in many commercial applications with complex data structures such as Manufacturing, Bioinformatics, B2B (Business to Business), Medicine and Geographical data (Powell, 2007; Ma & Yan, 2007; Pankowski, 2009). Thus, effective means of the management of XML documents as databases are needed for query, consistent and efficient storage. Various databases, including relational, object-oriented, and object-relational databases have been used for mapping to and from XML documents (Florecsu & Kossmann, 1999; Runapongsa & Patel, 2002). Among this kind of database, most researchers use a relational database as a persistent storage since it is a more promising alternative, because of its maturity. However, this approach has disadvantages, since it does not support well complex data structures such as scientific data because it cannot retain the original of XML documents.
With such problem, has led to the development of native XML database systems for a number of applications and its use is increasingly rapidly because its ability to hold and manage highly complex data structures (Bourret, 2007; Kohler, 2004; Lee et al., 2010). Such applications may use native XML database facilities (Kanne & Moerkotte, 2000) to store and update XML data (Tatarinov et al., 2001). The native XML database stores XML documents directly without performing any conversion or reformatting the XML documents into another format thus reduce processing time and provide better performance. However, native XML database is still in its infancy and not as mature as traditional databases (e.g., relational database), hence many important problems and questions remain unanswered, especially on the principles of XML database design (Arenas, 2006; Schewe, 2005; Libkin, 2007).

It is important to design good XML document for the sake of readability and manageability. The good design means there are no duplicate information, store correct and complete information. This is because duplicate information (also called redundant data) will wastes space and increases the likelihood of errors and inconsistencies. While the correctness and completeness of information is important because, if the document contains incorrect information, any reports that pull information from the document will also contain incorrect information. As a result, any decisions made based on those reports will then be misinformed. Like managing traditional database, the management of XML documents requires capabilities to handle with integrity, consistency, data dependency, redundancy, views, access rights, integration, and normal forms (Yu & Jagadish, 2008; Libkin, 2007; Arenas & Libkin, 2004; Dobbie, 2000; Feng et al., 2002). Amongst the important problem related to XML document design are data redundancies and update anomalies.

As an example, consider Figure 1 which illustrates the example of an XML document. For instance, for tno = ‘123,’ the lecturer named Bing who teaches the course number (cno) csc101 is stored twice, causing redundancy in XML document. As with a relational database, if we would like to update the name of the lecturer to full name “Bing Wang” in the XML document, this name needs to be updated twice. This type of update is called update anomalies Codd (1972). This problem occurs because of the structure design of XML document is based on the DTD in Figure 2. As with a relational database, such data redundancies and anomalies presented can be avoided by designing a good DTD. Those redundancies could be eliminated if we could refine the original DTD schema to a new DTD schema by eliminating some semantic constraint from XML documents.

However, to formulate the criteria for good of XML document schema design is very challenging for the following reasons:

- The structure of XML documents is different from that of relational databases which contain a complicated hierarchical structure, so it is difficult to see whether it contains redundancies (Arenas, 2006; Libkin, 2007).
- Expression of semantic constraints of XML documents imposed by schema such as DTD and XML Schema Definition (XSD) is limited (Yu & Jagadish, 2008).
- There are problems of ensuring that data and semantic constraint of designed schema are not lost and preserved after the process of normalization of the schema (Kolahi, 2007).
- No acceptable notion of an XML updates as yet exists, comparable to the notion of the relational updates, which makes it hard to say what makes an update anomaly (Libkin, 2007).
- There is no standard query language for XML document compared to relational algebra for relational databases (Libkin, 2007).

Thus, the challenges and issues motivate us to investigate further the needs and requirements to achieve a good XML document.
Fragmentation in Distributed Database Design Based on Ant Colony Optimization Technique