XML Documents Normalization Using GN-DTD

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

ABSTRACT

Designing a well-structured XML document is important for the sake of readability, maintainability and more importantly to avoid both data redundancies and update anomalies. This paper proposes to improve and simplify XML structural design using a normalization process. To achieve this, Graphical Notation for Document Type Definition (GN-DTD) is used to describe the structure of XML document at the schema level. Multiple levels of normal forms for GN-DTD are proposed and the corresponding normalization rules to transform from poorly designed into well-designed XML documents. A case study is presented to show the application of these normal forms and normalization algorithm.

Keywords: DTD, Graphical Notation, Normal Form, Normalization Algorithm, XML Model and Design

1. INTRODUCTION

Like managing traditional database, the management of XML documents require capabilities to handle with integrity, consistency, data dependency, data redundancy and anomalies (Arenas & Libkin, 2004; Dobbie et al., 2000; Embley & Mok, 2001; Vincent et al., 2004, 2007; Wang & Topor, 2005; Yu & Jagadish, 2008). Amongst the important problem related to XML design that needed to be looked into are data redundancy and update anomalies (Arenas & Libkin, 2004; Yu & Jagadish, 2008). Data redundancy and anomalies can happen in XML document if the schemas: DTD (Document Type Definitions) (Powell, 2007) or XML schema (Tompson et al., 2008), are not well-designed. These schemas allow one to define the structure, the content and the semantic of XML document. In this work, we consider only DTD since it is expressive enough for a large variety of applications and has been widely studied in database theory (Arenas & Libkin, 2004).

The design problem for XML document has recently been a subject of interest for database researchers, and a number of normal forms have been proposed to reduce data redundancies caused by functional or multi-valued dependencies (Arenas & Libkin, 2004; Feng et al., 2002; Mani et al., 2001; Vincent et al., 2004; Wang & Topor, 2005; Yu & Jagadish, 2008). However, XML normal form XNF, proposed by (Arenas & Libkin, 2004) achieves the best possible design form the point of view of eliminating redundancies in XML documents (Kolahi, 2007). But, there are problems with this definition of normal

DOI: 10.4018/ijirr.2011010105
forms and normalization process. Firstly, the notions of XML normal forms are presented in a difficult term to be understood and hard to be implemented practically because of the lack of graphical notations for the proposed theories. Secondly; the current approach of XML normal form did not show the tremendous benefit to practitioners and thirdly; the normalization algorithms only works for the existing normal form which has a very limited semantic expressiveness.

In this paper, we propose to improve and simplify XML structural design and normalization process using graphical model called GN-DTD (Zainol & Wang, 2010). We define multi levels normal forms for GN-DTD to allow users to find an ‘optimal’ structure of XML elements/attributes and to produce a correct, complete and consistent representation of the real world XML data which may interest to the user.

GN-DTD is a graphical modelling approach for describing XML documents. For GN-DTD itself, we define a complete set of syntax and structure which incorporate of attribute identity, simple data type, complex data type and relationship types among elements. Furthermore, semantic constraints are also precisely defined in order to capture semantic meaning among those defined objects. The significance of using GN-DTD model is that, it helps user to arrange the content of XML document in order to give a better understanding of its corresponding DTD structures. DTD commonly represented as textual representation, hence it is particularly difficult to be analysed and understood. In practise, it often causes difficulties when even designing a simple DTD. This is partly due to textual form of the grammar itself. Because of using these GN-DTD’s notations, we found it is particularly easily to improve XML structural design and more importantly, make the XML normalization procedure simpler and practical.

The rest of the paper is organized as follows. Section 2 provides a brief overview of GN-DTD’s syntax and semantic. Section 3 presents multiple levels of normal forms for GN-DTD. The normalization procedure of XML document based on GN-DTD is presented in section 4. A case study is given in section 5 to illustrate the application of normal forms and normalization rules. Section 6 and 7 discuss the related work.

2. BACKGROUND

Some of the notations of GN-DTD have been adopted and improved from the current data model ORA-SS (Dobbie et al., 2000) notations and conventional ER model (Chen, 1976). The significant difference between GN-DTD and the other models is complex element, simple element, and attribute nodes are explicitly distinguished. This is because such distinction is crucial for clarifying the semantics of the data. More importantly, GN-DTD provides an explicit semantic relationships definition to allow the user to define semantic relationship between nodes. The GN-DTD allows users to define the structure of nodes in a hierarchical way. We have made the ordering of sub-elements significant by treating them as a set, sequence (list) or disjunction. The GN-DTD model is briefly discussed in the following section. The details of the GN-DTD specification can be found in (Zainol & Wang, 2010).

2.1. Syntax and Semantics of GN-DTD

Graphical Notation for Data Type Definition (GN-DTD) graphical modelling (Zainol & Wang, 2010) consists of six basic components:

- **Complex element nodes**

A complex element node is used to represent a set of elements which have another sub element and attribute. The complex element node is illustrated as labelled rectangle box. This notation is adopted from the ER model (Chen, 1976) which is similar to entity. The label is written in the rectangle box as a tuple `<name,level>`, where `name` represents the name of the node and `level` represent the depth of the node in GN-DTD. The name is mandatory.