On Maintaining XML Linking Integrity During Update

Eric Pardede$^1$, J. Wenny Rahayu$^1$, and David Taniar$^2$

$^1$ Department of Computer Science and Computer Engineering, La Trobe University, Bundoora VIC 3083, Australia
{ekpardede, wenny}@cs.latrobe.edu.au

$^2$ School of Business Systems, Monash University Clayton VIC 3800
David.Taniar@infotech.monash.edu.au

Abstract. It is a fact that XML update has become more important with the rise of XML Database usage. How update operations affect XML documents needs to be investigated further. In this paper we propose a methodology to accommodate update without violating the XML document’s constraints. The constraints maintained are those that are defined using XML linking language: xlink and xpointer. This language, which is standardized by W3C, is used to provide referential purpose among XML documents or nodes.

Since XML link is embedded as an attribute in an XML instance, our proposal can be used for schema-less documents and for instance-based reference. We propose a set of functions that perform checking mechanisms before updates. The proposed method can be implemented in various ways, and in this case we use XQuery language.

1 Introduction

XML update is a considerably new research for semi-structured data community. There was a perception that XML document does not need frequent update [1]. However, due to the dynamic nature of the web application, we have witnessed a growing number of XML documents that require regular update.

XML update methodologies have been discussed in a few works [1, 11]. Moreover, the researches on constraints preservation during update operations are even fewer. [7] discussed the issues on capturing semantic constraints during XML update. However, it is applicable to schema-based XML documents.

Unfortunately, very frequently we have to store schema-less XML document in our database repository. Furthermore, for some cases –even to the XML with schema- the constraints are not schema-based but more likely to be instance-based.

In this paper we aim to propose a methodology to update XML document without schema bound. The methodology will preserve the referential integrity constraint that exists through some XML linking technologies: XML Linking Language (XLink) [14] and XML Pointer Language (XPointer) [15].

After the introduction, in section 2 we will discuss the motivation for this work. Section 3 provides basic information regarding the XML linking technologies. Section 4 proposes the methodologies for the update operations insertion, deletion and
replacement. In section 5 we provide some analysis of the work including our contributions and limitations. Finally our work will be concluded in section 6.

2 Motivation

XML document is a collection of information that is structured in a tree of nodes. Most of the time, we find an XML node contains information that is not explicitly stated in that particular node. The actual information is stored in another resource and the former node just refers to that resource. The different location between the two sources has raised the issues of referential integrity. How the reference links between the sources are maintained becomes a crucial task.

We have been familiar with the concept of primary and foreign keys since the early era of relational model. The excellent support of referential integrity in this data model is the strength that has given RDB an important position in the database communities. Since then, any emerging data model requires referential integrity maintenance support as a basic requirement. It also applies for the XML data model that has been mentioned as the new era of data format in the database communities.

The common way of referential integrity implementation in XML data model is ID/IDREF or key/keyref. For update, some XML database products use ID/IDREF to maintain the referential integrity [3, 4, 5, 9, 10]. Another approach uses key/keyref and embeds the referential integrity maintenance in XML query language [7]. Despite the contribution and usability, there is a limitation to these approaches. The system needs to know the ID/IDREF and key/keyref through a schema before it can employ the rules. There are two main cases where the current approaches will not work.

- **XML documents without schema.** The non-schema-based XML document is mostly used by the document-side of the XML community [1]. Even though the XML document is rarely structured, its elements can refer to other sources and thus referential integrity is still necessary.

  For example, it is unlikely to store the following XML with a schema. Inside one element, there is a link that refers to another source. If we need to update the document that affects these link types, the referential integrity might be violated.

  ```xml
  <Article title="Linking Language in XML Document">
  <firstchapter>
    <para> This work is the continuation of our previous work on 
    <article (!--link to another source)> Referential Mechanism in 
    XML Data Model </article>. Our main interest..........<para>
    ..</firstchapter>..</Article>
  ```

- **Instance-based reference instead of schema-based.** In this case, the same element of two instances that are validated by a schema refers to different sources. Thus, identifying the reference through IDREF/keyref will not be correct. For example, the following schema-based XML document cannot use the IDREF/keyref because the reference links are instance-based. The element authors actually refer to different sources type.