UPDATEABLE EDUCATIONAL APPLICATIONS BASED ON COMPRESSED XML DOCUMENTS

Tomasz Müldner, Christopher Fry
Jodrey School of Computer Science, Acadia University, Wolfville, N.S., Canada

Jan Krzysztof Miziołek
IBI AL, University of Warsaw, Warsaw, Poland

Keywords: XML, Compression, Query, Update, Educational applications.

Abstract: Many recent educational applications, including stand-alone and client-server applications, use XML-based data. Unfortunately, XML data are verbose, which results in large memory overhead and a low throughput for network-based applications. In addition, most XML-based educational applications do not support an essential feature, namely updates of data. In this paper we describe a specific type of XML compression and outline its application to education. The use of the proposed approach will result in a new generation of XML-based educational applications, which do not suffer from the afore-mentioned problems.

1 INTRODUCTION

A large number of educational applications (Navro et al., 2000), (Buendia et al., 2002), (Song et al., 2008) use the XML format (XML, 2010) for underlying data. There are several reasons why XML is a perfect choice for a data format. First, XML is the de-facto standard for storing data, and there are many tools (often available at no cost) to process XML, such as XML parsers. Second, unlike HTML, which provides only markups for format specifications, XML provides structure for data. The formatting of the data, be it PDF or HTML is left to tools such as CSS (CSS, 2010) or even better XSLT (XSL, 2010). Third, XML satisfies the need for supporting internationalization of educational applications, i.e. for making them available in various languages without having to modify the underlying data (Müldner, 2004). However, existing applications suffer from two major problems, which severely limit their usefulness. First, XML is verbose and as a result, on average, XML-based representation is often as much as 10 times larger than the original data. This increase in size, combined with a large amount of data to be maintained, results in XML files that are hundreds of megabytes in size. Specifically, some educational applications (for example, a typical set of courseware for an education institution) require large amounts of data, consume vast amounts of disk space, and cannot be used on increasingly more popular wireless devices such as PDAs and mobile phones. In addition, retrieval of any required data, for example midterms used for a specific courseware, will result in a query issued on the underlying XML data, which will have a slow response time. The second problem faced by XML-based educational applications is that typically they are *static*, i.e. they do not support updates of the XML data (this limitation is caused by the fact that currently most XML tools do not support updates). However, the ability to update educational applications is of paramount importance, for example in order to modify a specific lesson or to allow the instructor or the student to add notes or bookmarks.

There is a considerable body of research on XML data compression, and more recently on queryable XML compression with lazy decompression. At the same time, there is little research on updateable XML compressors with lazy decompression (Müldner, 2010). However, there is no research known to the authors of this paper that provides an indication as to how queryable and updateable XML compression with lazy decompression may be used for educational purposes. This paper attempts to fill in this gap and it describes such a compressor and its applications to education.

Contributions. The main contribution of this paper is a description of an application for educational purposes of a queryable and updateable XML compressor, called XSAQCT.

This paper is organized as follows. Section 2 describes XSAQCT (Section 2.1) and then it outlines the applications of XSAQCT for educational purposes (Section 2.2). Section 3 provides conclusions and it describes our future work.

2 XSAQCT AND ITS APPLICATIONS

In this section, we briefly describe the architecture of XSAQCT, and then outline its applications for education.

2.1 Architecture of XSAQCT

In the first stage, we use a SAX-parser to separate structure and contents. In addition, the structure is *summarized* to provide a concise representation amenable for querying and updates. Specifically, given a document D, a single SAX traversal of D creates an annotated tree $T_{A,D}$, in which all *similar* paths (i.e. paths that are identical, possibly with the exception of the last component, which is the data value) are merged into a single path. Every tree node $T_{A,D}$ is labeled with an *annotation* representing father-children relations via an integer sequence. For a sample XML document shown in Fig. 1, the corresponding XML tree and its concise representation in the form of an annotated tree are shown in Fig. 2.

The tree representing the structure summary in XQueC, another well-known XML compressor, is additionally annotated with indices into the array, called *ID sequences* storing pairs of the form <path from the structure summary, a list of pre/post order numbers for each occurrence of the first element of this path>. As a result, XQueC's structure summary is less compact than XSAQCT's. For the sample XML document shown in Fig. 2, annotations in XSAQCT consist of 12 integers, while XQueC's structure summary requires 20 integers. This difference in size is much more profound for very large XML documents.

```
<a>
    <b>
      <c>
      </c>
    </b>
    <b>
      <c>
      </c>
      <d>
      </d>
   </b>
   <b>
      <e>
      </e>
      <e>
      </e>
      <c>
      </c>
  </b>
</a>
```

Figure 1: Sample XML document.

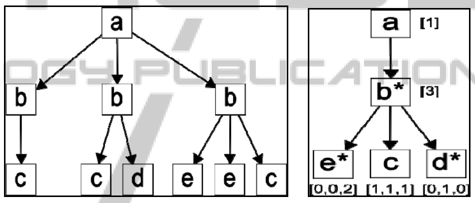

Figure 2: The XML document and its annotated tree.

In XSAQCT, data values are written to various containers based on the rule that values belonging to similar paths are written to the same container and subsequently compressed.

Queries over XML data are implemented in XSAQCT using a small subset of (XPath, 2010). Our future work will extend this set.

Various *update operations* on the document D are available.

2.2 Applications of XML Compression to Education

In this section, we describe a typical educational application and outline how XML compression can be used to facilitate the implementation of such application.

Consider an educational institution, which maintains a set of courseware. Each courseware consists of a number of lessons, and each lesson consists of an optional pre-test, a number of pages, and an optional post-test. In addition, we assume that each page may have a number of notes associated with this page (for example, storing instructor's suggestions as to what should be studied more carefully, or student "to-do" lists).

The structure of courseware presented in Fig. 3 shows that it is well suited for compression with XSAQCT, since as we noted before this compressor has the best performance for regular structures.

Our previous experience with creating XML-based courseware of the above type showed that the user's interface should provide the following functions:

1. Author's interface to support building the initial courseware
2. Learner's interface to support access to specific courseware, or just specific lessons or pages within the lesson
3. Author's interface to support insertion/deletion of new pages/lessons, modifications of selected pages and insertion/deletion/modification of author's notes
4. Learner's interface to support insertion/deletion/modification of learner's notes

In what follows, we outline a design of the implementation of the above interface based on XSAQCT used as an XML compressor. Implementation of the author's interface (point 1. above) can use basic XML tools, such as XML parsers. When the process of courseware creation is completed, XML data created are passed on XSAQCT for compression and storage. Implementation of the learner's interface listed in 2. above uses standard queries, which will result in partial decompression of the required courseware; the entire courseware will continue to be saved in a compressed form. Points 3. and 4. listed above will be implemented via the update functionality of XSAQCT with lazy decompression.

3 CONCLUSIONS

In this paper we showed how XML-based educational applications can be implemented in more efficient way using compressed XML documents, which can be updated and queried with minimal decompression. Our future work we will design and test a more complete educational application based on compressed XML data.

REFERENCES

Navrro, A., Sierra, J-L., Fernandez-Manjon, B., Fernandez-Valmayor, A, 2000. XML-based Integration of Hypermedia Design Component-Based Techniques in the Production of Educational Applications. In: Manuel Ortega, José Bravo (eds.) *Computers and Education in the 21st Century*, Kluwer 2000.

Buendia, F., Diaz, P., Benlloch, J. V., 2002. A Framework for the Instructional Design of Multi-Structured Educational Applications. In: P. Barker & S. Rebelsky, Eds., *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2002*, Chesapeake, VA, 210-215.

Chang-xin, S., Qinghai, K. M., 2008. Applications of Data Mining in the Education Resource Based on XML, In *Advanced Computer Theory and Engineering, 2008. ICACTE '08 International Conference*, 20-22 Dec. 2008, pp. 943 - 946

Extensible Markup Language (XML) 1.0 (3rd ed.), 2010. http://www.w3.org/TR/REC-xml/, retrieved on September 20, 2010.

Cascading Style Sheets, 2010 http://www.w3.org/Style/CSS/ retrieved on September 20, 2010.

XSL Transformations (XSLT) W3C Recommendation; http://www.w3.org/TR/xslt, retrieved on September 20, 2010.

XPath, 2010, http://www.w3.org/TR/xpath/, Retrieved on September 20, 2010.

Müldner, T., Wong, F., Benoit, D., 2004. My Webpage can Speak Many Languages. In *ED-Media 2004*, Lugano, Switzerland, 2004. AACE Proceedings, pp. 2040-2046.

Müldner, T., Fry, C., Miziołek, J. K., Corbin, T., 2010. Updates of Compressed Dynamic XML Documents. In *The Eighth International Network Conference (INC2010)*, Heidelberg, Germany, July 2010.