XML Data Integration in Action*

The Sigmod Record Distinguished Profiles in Databases Interviews

Maarten Marx and Anne Schuth
ISLA, Informatics Institute, University of Amsterdam
Science Park 904, 1098 XH Amsterdam, The Netherlands
maartenmarx@uva.nl, anne.schuth@uva.nl

ABSTRACT

We indicate how several theoretical XML technologies developed within the FP7 ICT FET Foundations of XML (FoX) project (`http://fox7.eu`)can directly be applied to a real life problem: making a collection of interviews easily accessible. To make the solution of the problem scientifically interesting we worked under the following constraints:

1) all processing must be done automatically (scalability);
2) only XML technology may be used (uniformity);
3) data format should allow for defining views and OLAP (re-usability);
4) data can directly be added to the Linked Open Data cloud (connectivity);
5) users are brought directly to the right place, depending on their information need (productivity).

We describe the XML technologies that were applied. A working system is available at `http://xml.politicalmashup.nl/sigmod`.

1. INTRODUCTION

The aim of this paper is to show the benefits of using XML technology by creating a small but completely worked out real life example, while satisfying the constraints laid down in the abstract. This project has shown two important advantages: First, the clear separation between form and content which comes natural through XML stimulates reuse and sharing of data. Second, maintenance and updates are facilitated because the complete application runs under W3C standardized XML technology and all within one application. We now briefly describe the real life example.

Marianne Winslett has created a corpus of 32 interviews with distinguished database researchers held between 2002 and 2010. These interviews are published in Sigmod Record and also available from the ACM website in three formats: PDF text, MP3 audio and Flash video format. At present the only connection between these 32×3 files is that they are listed on one web page[1] grouped by interview.

*The authors acknowledge the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the FET-Open grant agreement FOX, number FP7-ICT-233599.
Thanks are due to Jaap Kamps.

[1] `http://www.sigmod.org/publications/interview`

Our goal was to integrate all this data into one system allowing users to query the text of the interviews and be directed to the relevant spots in the videos, compare answers of different researchers to the same question, do analytical querying, etc. We succeeded in building a system which meets this goal. Apart from constraint 1 (alignment of text and video was done manually), we did so satisfying all constraints mentioned in the abstract. Instead of describing it, we invite the reader to try it out.This paper describes the most important technical aspects of the system.

2. STRUCTURE EXTRACTION WITH XSLT

In order to satisfy constraints 3 and 5 we made the implicit structure in the interviews explicit in XML markup. Interviews have a very regular structure: they consist of a sequence of question-answer pairs with some introductory information. Long questions and answers may be partitioned into several paragraphs. The Relax NG schema corresponding to this model is in Figure 1. We briefly describe the

```
start =
  element interview {
    element title { text },
    element intro { text },
    element author { text },
    element qa {
      element q { p+ },
      element a { p+ }
    }+
  }
p = element p { text }
```

Figure 1: Relax NG schema for interviews.

transformation from interviews in PDF format to XML valid with respect to this schema. First we transform the PDF to well-formed XML using off-the-shelf open source software. This yields a rather flat XML consisting of text elements for every line of text. Only page, position, font size and font style (bold, italic) is preserved. Schematically this can be represented as a sequence of the form

$$\mathtt{T}_1\mathtt{T}_2\mathtt{T}_3\mathtt{T}_4\mathtt{T}_5\mathtt{T}_6\mathtt{T}_7\mathtt{T}_8\mathtt{T}_9\mathtt{T}_{10}, \tag{1}$$

where each `T` is an element containing text like

```
<text>Welcome to this instalment of</text>.
```

We transform this sequence in two steps.[2] First, using Boolean

[2] These can be done in one pass over the data, but it is better explained in this way.

XPath expressions, we mark text elements as being the start of a question, an answer, or a new paragraph. After this step (1) looks like

$$\mathtt{Q}_1\mathtt{T}_2\mathtt{A}_3\mathtt{T}_4\mathtt{T}_5\mathtt{P}_6\mathtt{T}_7\mathtt{Q}_8\mathtt{A}_9\mathtt{T}_{10}. \quad (2)$$

In the second step we nest this sequence according to the schema in Figure 1, yielding

```
<qa>
  <q> <p>1,2</p> </q>
  <a> <p>3,4,5</p> <p>6,7</p> </a>
</qa>
<qa>
  <q><p>8</p></q>
  <a><p>9,10</p></a>
</qa>,
```

where the numbers correspond to the positions of the `T` elements in (1) and indicate the text-value of these elements.

`http://politicalmashup.nl/2011/01/text-preserving` contains a full description of the transformation including all XSLT code. Within the FoX project the Universities of Hasselt and Dortmund are investigating ways to automatically learn such text-preserving transformations from positive examples. XML data exchange is an important topic within FoX, studied by the universities of Edinburgh, Oxford and Warsaw. The present state-of-the-art declarative XML data mapping formalisms (implications of tree patterns) [1] are however not expressive enough for the transformation described above.

3. BEST ENTRY POINT RETRIEVAL

Full text search in a collection of XML files yields an opportunity that standard Google style search does not have. Whereas with standard web search the unit of retrieval is fixed —always the complete document— in XML search any XML-element can in principle be returned. In the ideal case, the search engine decides which parts of the document are most relevant ánd most specific to a query and returns the corresponding XML element(s). This search scenario is studied within INEX [2]. The FoX project studies foundational aspects of it and creates large document-centric XML corpora with rich semantic structure.

This XML search technique has been applied to perform the task called *best entry point retrieval*: given a document and a query, what are the best points in the document to start reading? For audio and video search, this technique is ideal because these media are hard to quick-scan for humans. We applied it is as follows: We decided that every question could be an entry point to the video and only these. Thus we aligned the XML and the videos by adding a time code to each question element. We then created four full-text indexes: on questions, on answers, on question-answer pairs, and one the complete interviews. This had two purposes. First, users can specify in which part they want to search: in questions, in answers, or in the combined question-answer pairs. Second, we combine the different indexes to improve ranking [4]. For example, (3) shows how we, given a search term `t` and a question-answer pair `QA` from an interview `I`, combine the score from the question-answer pair index and the complete document index.

$$\mathrm{score}(\mathtt{t},\mathtt{QA}) = \lambda \cdot \mathrm{score}_{\mathrm{qa}}(\mathtt{t},\mathtt{QA}) + (1-\lambda)\cdot \mathrm{score}_{\mathrm{int}}(\mathtt{t},\mathtt{I}) \quad (3)$$

Setting λ to 0.6 has been shown to give good results before [4] and was also used by us.

4. ADVANTAGES OF XML-BASED SYSTEMS

Working with the source data in XML has two important advantages.

First, it enforces a strict separation between form and content. The schema in Figure 1 contains only semantic markup. Using XSLT this can be styled in any desired manner. This separation eases making global changes and reuse of the data.

Second, a complete application can be build using only W3C standardized XML technology. Moreover, the whole application needs only one piece of software, the XML DBMS, and data is only stored there. These two aspects greatly facilitate maintenance, updates and portability of the system.

We now briefly discuss the front and backend of the system.

Front end. The front end consists of two web-pages. There is a search engine result page (SERP) [3] with a search box and a relevance ranked list of hits. Hits are entry points to the video, consisting of a small text-snippet with the search terms highlighted. They are grouped by interview and at most 3 hits per interview are shown. The second page shows the result in a style familiar from YouTube: A video player with the transcript of the currently playing question-answer pair below it. To the right is a timeline of the interview with all questions which allows for fast navigation and browsing.

Both pages are generated using just one single XQuery. Styling is done with CSS, and Javascript is used for the dynamic features of the page. We needed to use three extensions to XQuery 1.0: full-text search [5], updates [6] and a module for obtaining parameters from HTML forms.

Backend. The eXist XML DBMS was used as a backend. eXist comes with full text search implemented using Lucene, with virtually the same syntax as the W3C recommendation. There is no duplication of data needed. Each interview is only stored once as an XML file in eXist. We also serve the interviews in PDF format in the ACM proceedings style: these are generated at query time using a combination of XSLT and LaTeX 2_ε.

5. REFERENCES

[1] S. Amano, L. Libkin, and F. Murlak. XML schema mappings. In *Proc. PODS '09*, pages 33–42, 2009.

[2] N. Fuhr, J. Kamps, M. Lalmas, and A. Trotman, editors. *Focused Access to XML Documents, Proc. INEX 2007*, volume 4862 of *LNCS*. Springer, 2008.

[3] M. Hearst. *Search User Interfaces*. Cambridge University Press, 2009.

[4] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. An element-based approach to XML retrieval. In *Proc. INEX 2003*, pages 19–26, 2004.

[5] World-Wide Web Consortium. XQuery and XPath Full Text 1.0 W3C Proposed Recommendation 25 January 2011. `http://www.w3.org/TR/xpath-full-text-10/`.

[6] World-Wide Web Consortium. XQuery Update Facility 1.0 W3C Proposed Recommendation 25 January 2011. `http://www.w3.org/TR/xquery-update-10/`.