A la Carte Schema: A Case Study Comparison of the Application of DTDs and XML Schema to the Carte Calendar Project Template

Ingrid Daneker *(idaneker@ukonline.co.uk)*

School of Library, Archive and Information Studies, UCL

Claire Warwick *(c.warwick@ucl.ac.uk)*

School of Library, Archive and Information Studies, UCL

Introduction

This proposal describes research which compares the validation capabilities of conventional DTDs and XSD Schemas using the *Carte Calendar Project* template of the Oxford Digital Library (ODL) as a case study.

The *Carte Calendar* is a manuscript consisting of 75 volumes, cataloguing political papers on English and Irish history of the 17th century, originally collected and put together in chronological order by Thomas Carte, and converted into Library records by the Librarian Edward Edwards in the 19th century. The ODL is in the process of creating a digital record of Edwards' catalogue of Carte's collection, covering the period between 1660 and 1688, converting about 17,500 individual manuscript pages into XML-encoded transcripts using an EAD-DTD (1998) based transcription template.

The manuscript contains structured data which is repeated on almost every page, such as the shelf-mark, the unit-date, a document number, and a page number, as indicated by the keying sample below, showing the content of the `<unittitle>` element.

```
<!--shelfmark-->
<unitid type="shelfmark">MS. Carte 45,
fol(s). 67</unitid>
<!--document number-->
<unitid type="docno">34</unitid>
<!--pencil page number-->
<unitid type="page">451</unitid>
<!--red number-->
<unitid type="redno">?</unitid>
<unitdate> 22 May 1661</unitdate>
<physdesc>
<genreform>Holograph</genreform>
</physdesc>
```

These data-fields lend themselves well to the application of user-defined data-types, which Schema language can facilitate, thus making possible the automated verification of the data.

At any one time four freelance keying operators with varying levels of experience and three permanent staff might be working on the transcription of the project. Consistency of data entry is an ongoing concern, as it requires not only familiarity with the author's 19th century hand-writing and the subject matter, but also a good knowledge of the historical background.

The aim of this research was therefore to test whether the use of XML Schema could help to improve the accuracy of data entry by introducing stricter validation constraints than are mandated by the EAD-DTD. It will highlight some of the benefits of Schema definition language for the document type definition of the Carte template, compared to conventional DTD syntax. It will also consider whether the adoption of Schema based document type definitions is easily achievable for users of encoding standards such as EAD and TEI, who may have little prior knowledge of Schema.

This research is particularly timely, since in June 2004, Daniel Pitti announced that work was about to begin on developing one or more official EAD schemas (Pitti, 2004), as it provides a case study of how such a Schema might work in practice, and could inform work on the much broader EAD schema standard.

Methodology

The researcher analysed the predefined element structure of the *Carte* project, keying template and the intellectual content provided by the Carte Manuscript to determine areas which would potentially benefit from the addition of Schema definitions, by restricting allowable (valid) data-input and element/attribute use. A template-specific DTD was then created based on SGML/EBNF syntax, containing specifications for only those elements/attributes of the EAD standard used in the template. Although significant changes to the original tag-set of the template were avoided in order to remain backward compatible with the original EAD-DTD, some alterations had to be made so that the addition of Schema's data-typing capabilities could later be facilitated.

The 'bespoke' DTD applied much stricter definitions than the original EAD-DTD with regards to the use of allowable elements/attributes in the template, the number of occurrences and their possible location within the document. This allows for stricter validation of the document structure and content,

providing more rigid constraints for keying staff, whose objective is to produce a valid document following the entry of data.

The new DTD was converted into XSD Schema language using XML editing software; *oXygen* XML editor 4.2 and *Altova's XMLSPY 2004 Enterprise Edition* (XMLSPY v2004 rel. 4 U). These specific tools were chosen because other packages such as the *XMetal Home Edition* (version 2004) provided limited support for XSD files which contain, for instance, multiple namespaces.

With the aim of applying even stricter rules to the allowable elements/attributes of the template's metadata structure, the researcher then introduced more rigid input data specifications to the resulting Schema file, facilitated by the additional capabilities of Schema definition 'language', which goes well beyond the potential of conventional DTDs. Further, in order to explore the benefits of the inbuilt namespace support Schema offers, two additional Schema files were created and associated with the main Schema using the `<xs:import>` element.

Findings

Use of the new DTD

The new project-specific DTD ensured a higher level of uniformity in structure and data entry verification. It did so by stipulating, where possible, the order and number of occurrences of EAD elements used in the template, as well as by specifying the requirement of all the attributes and possible attribute values. It also eliminated the very imprecise and inflexible definition structure of mixed content specifications, determined by the underlying syntax of conventional DTDs. This project-specific DTD should, however, only be used for the validation stage of the project. To assure interoperability and compatibility with other projects, such as its inclusion in the A2A (Access to Archives – `<http://www.a2a.org.uk>`) records, the instance document must remain backward compatible with the official EAD-DTD standard. Thus the more generic DTD file must be associated with the template again after the proofreading process.

Benefits of Schema

The Schema allowed for the use of data-typing to increase the control over data entry by applying stricter validation parameters to the instance document. For example, when a user-defined global simple type specification such as 'content' (see the 'content' type definition below) is included in the element definition of, for example `<geogname>`, it ensures that an element must have content.

The user-defined type definition 'content':

```
<xs:simpleType name="content">
<xs:restriction base="xs:string">
<xs:pattern
value="([a-zA-Z?{$])+(.)*"></xs:pattern>
</xs:restriction>
</xs:simpleType>
```

This is not enforceable using DTD syntax, which allows the user either to define empty elements or to specify parsable character data (`#PCDATA`), which includes whitespace. The inclusion of the 'content' datatype in an element's content model will make sure that — should a transcriber forget to enter data for the element in question — validation against the Schema produces an error message pointing at the element's location in the instance document.

Further benefits were highlighted by the use of namespaces, which can help to avoid potential ambiguity problems, particularly where document exchange and integration is concerned. Schema allows the user to declare different namespaces, which can then be applied to individual element declarations, and thus used to differentiate between the varying element content models of the same element. In the *Carte* Template, for example, the `<num>` element has four different instances. This ambiguity was resolved by making use of additional namespaces for three of the four `<num>` elements.

Where different names describe the same set of data, Schema offers document authors the opportunity to define substitution groups which allow for increased compatibility of different instance documents. In the case of the *Carte* project this occurs with regard to `<genreform>` and `<physfacet>`, which are both allowable child elements of `<physdesc>` according to the original EAD-DTD.

On a more basic level, unlike DTD syntax, Schema does allow for sequencing of child elements, in a mixed content element model. Occurrence indicators for elements can be set much more specifically through the use of numerical values ranging from 0 to infinity, and global type definitions guarantee re-usability of individual type definitions and element declarations. All of these features proved useful in the Carte case study, and will be described in the proposed paper.

Conclusion

The Schema-based keying template was tested by three experienced encoders at SERS (Systems and Electronic Resources Service). Their data-input would have validated against the original EAD-DTD. However, on average 3-4 errors were found by the *oXygen* 4.2 parser, pointing at entry format inconsistencies. For example if the unitdate was entered as *1*

January 1661 instead of *01 January 1661* (or *10 January 1661*), the validator highlighted the missing digit as an error, in accordance with the pre-defined entry parameters provided by the underlying Schema. Similarly, missing content for the element `<geogname>` caused an error message (as the Schema's element declaration includes the global type 'content', discussed above).

These specifications were incorporated in the Schema to make sure that the encoder double checks the manuscript for such data fields, and testing proved it had been successful. Nevertheless a balance must be maintained between accuracy and speed of transcription in the case of the *Carte* project. Extensive data-typing of the `shelfmark`, `docno`, `pencilpage`, `redno` and `unitdate` fields might cause transcribers to spend more time trying to figure out which data format creates a valid document than is spent with the actual data input for each individual record.

The experimental DTD and resulting Schema have proved successful in helping transcribers to avoid errors and improve consistency. Yet despite the benefits of using Schema language over DTD syntax for XML document declarations, its complexity and to the untrained user's eye rather complicated element content modelling structure, underpinned by the W3C standard recommendation, might discourage potential users from trying to learn it. Nevertheless, this research has shown that software already available can help users to make the transition from DTD to Schema. While the needs of individual projects must always be considered carefully, the case study of the *Carte* project template shows that the benefits of Schema use should be taken seriously, despite its complexity.

Bibliography

Costello, Roger L. *XML Technologies Course, XML-Schemas: A downloadable schema tutorial.* xFront, 2003. Accessed 2005-02-27. <http://www.xfront.com/xml-schema.html>

Deitel, H.M., et al. *XML: how to program.* Upper Saddle River, NJ: Prentice Hall, 2000.

Encoded Archival Description. *EAD Tag Library for Version 1.0.* . Accessed 2005-03-21. <http://www.loc.gov/ead/tglib1998/>

LEADERS – Project (October 2001 to March 2004: Linking EAD to Electronically Retrievable Sources). Accessed 2005-03-21. <http://www.ucl.ac.uk/leaders-project/>

Mertz, David. *"TEI - the Text Encoding Initiative": An XML dialect for archival and complex documents.* September 4, 2003. Accessed 2005-02-27. <http://www-106.ibm.com/developerworks/library/x-matters30.html>

Pitti, Daniel. *EAD and W3C XML Schema.* Cover Pages. Accessed 2005-02-27. <http://xml.coverpages.org/ead.html>

Pitti, Daniel. Encoded Archival Description List (EAD@LISTSERV.LOC.GOV), 4 June 2004. <http://listserv.loc.gov/cgi-bin/wa?A2=ind0406&L=ead&P=R605&D=0&I=-3>

Rahtz, Sebastian. *Converting to schema: the TEI and Relax NG.* Text Encoding Initiative. Accessed 2005-03-21. <http://www.tei-c.org.uk/Talks/xmleurope2002/>

Sperberg-McQueen, C.M., and Lou Burnard. *Tel P4: Guidelines for Electronic Text Encoding and Interchange.* Text Encoding Initiative, 2001. Accessed 2005-02-27. <http://www.tei-c.org/P4X>

Tennant, Roy, ed. *XML in libraries.* New York: Neal-Schuman Publishers, Inc., 2002.

Thompson, Henry S. "XML Schema types and equivalence classes." Paper presented at the XML Europe 2000 conference. 2000. Accessed 2005-02-27. <http://www.gca.org/papers/xmleurope2000/papers/s06-01.html>

Van der Vlist, Eric. *Using W3C XML Schema.* O'Reilly xml.com. Accessed 2005-02-27. <http://www.xml.com/pub/a/2000/11/29/schemas/part1.html?page=1>

Watt, Andrew, and R. Allen Wyke. *XML Schema Essentials (e-book).* New York: John Wiley & Sons, Inc., 2002.

Women Writers Project. "Encoding Practice." Rhode Island: Brown University. Accessed 2005-02-27. <http://www.wwp.brown.edu/encoding/research/NASSR/WWP.html#Heading3>

Women Writers Project. "Methodological Issues." Rhode Island: Brown University. Accessed 2005-02-27. <http://www.wwp.brown.edu/encoding/research/NASSR/WWP.html#Heading4>