elpub2002 Proceedings — J. A. Carvalho et. al. (Eds.) — VWF Berlin, 2002

Creating XML Documents for Electronic Journal Publication

Brian Rosenblum

Scholarly Publishing Office
University Library, University of Michigan

Abstract. The Scholarly Publishing Office (SPO) at the University of Michigan Library works closely with a number of scholarly journals to develop methods and models for online publication. One of our biggest challenges is converting a steady stream of incoming content from a number of different sources and formats into XML to prepare for online delivery. The paper begins with background information on SPO. Then it summarizes the three major stages in our text conversion process: 1) creation and submission of the documents and related metadata; 2) editing and structuring of the document; and 3) transformation of the content to XML. The paper then discusses some of the issues and challenges we face and lessons we have learned, including: our experience working with content providers, managing multiple processes for multiple publications, and a discussion of some of the conversion tools that we use.

1 Background

The Scholarly Publishing Office (SPO), a unit of the University Library, University of Michigan, provides tools and services to facilitate the electronic publication and distribution of scholarly content. SPO supports the publication of traditional print publications in an online environment, as well as scholarly work specifically designed for electronic delivery.

SPO publications are encoded in XML and made available online through DLXS, the suite of tools and search engine that provides the foundation for digital library services at the University of Michigan. The DLXS search engine, XPAT, is a powerful SGML/XML aware tool which supports searching, indexing and retrieval of encoded information. The DLXS "middleware" tools are designed to process and support access to various types of digital collections (called "classes") and various "behaviors" associated with these collections (such as different types of searches, browse, and remembering search histories.) Most of SPO's publications are published as part of DLXS's "textclass", which uses a DTD based on the Text Encoding Initiative (TEI) rules for encoding scholarly text (other DLXS classes include bibliographic data, digital images and image metadata, EAD-encoded finding aids, and encyclopedic reference works).

SPO currently has about 10 publications in production, over 15 projects in development, and has worked extensively on a number of large academic publishing projects, such as the ACLS History E-Book Project. SPO is a small production unit, with four full time employees (including an interface designer and a programmer) and occasional part-time students to assist in the text markup process and other activities.

This paper will focus on the transformation process and other issues related to converting content to level 4 TEI-based XML for use in the DLXS search engine.[1] First I will outline the general preparation and conversion process; then I will discuss some of our primary challenges and lessons learned in developing this process.

2 The Conversion Process

There are three general stages in our document conversion process: 1) submission of the document and creation of metadata; 2) thorough editing and structuring of the document with appropriate paragraph and character styles; and 3) the transformation of the content to XML. The following sections outline these three stages as they take place in their ideal form.

2.1 Submission of Documents

The first stage consists of the content providers submitting to SPO the content and a metadata record for each article or digital object. One method for this is through a web-based submission system created by SPO. The system allows the content provider to log into the system, create a record with basic bibliographic information about each item, edit records, upload files in various formats and, when they are ready, mark them for publication. In some cases, marking items for publication automatically begins the subsequent stages of the transformation process, and the content appears online within twenty-four hours with little or no intervention by

[1] TEI encoding levels provide guidelines for describing text used for different purposes. Level 1 is minimal encoding for wholly automated text creation. Level 4 encoding is highly detailed markup at the structural level but does not require the assistance of content experts. Level 5 requires scholarly analysis and markup at the semantic level

SPO staff. In other cases marking them for publication simply notifies SPO staff that new content is ready for processing and we need to physically move it to the next stage of production.

The web-based submission system works well for scholarly journals, or those publications with a small amount of articles that are published periodically. For many publications and projects, however, this system is not appropriate or the most efficient method of submitting the files–as in the case, for example, of an annual collection of conference proceedings, in which there are one or two hundred papers at a time, once per year. In such cases the files are delivered to us electronically or on disk, along with a database with the article metadata. We encourage use of FileMaker for this purpose, and we often place databases on a server for ease of use, access and sharing.

Before setting up a submission process for a particular publication or collection, SPO works closely with the content provider to ensure that proper metadata is created and provided. This includes arranging a suitable numbering and identification scheme; anticipating the types of searching and browsing desired in the final publication and making sure the necessary metadata exists; and, if appropriate, coordinating metadata information with existing library catalog records. The records in the database will be used to create TEI headers to attach to the article after it has been converted to XML.

2.2 Structuring the Documents

The second stage involves structuring the content (if it has not already been structured by the content provider) to prepare it for transformation to XML. First, the documents are converted to MS Word or RTF format, then, using MS Word as an editor, paragraph and character styles are applied to the entire text. These styles identify the structural elements of the articles (such as title, author, block quote and epigraph) rather than the layout characteristics (bold, indent, font sizes, etc.) We have created a general template of about 15 basic styles that will suffice for most of our publications. We work through each article paragraph by paragraph and apply the styles. When possible, we work with the content providers to structure the text, either by providing the templates to them or encouraging them to structure their content consistently using their own methods and templates. If they are able to provide us with the text already structured, we can often skip this stage and move directly to the transformation stage, greatly speeding up and automating the process. Often, however, for reasons explained below, we must structure the text ourselves. It is a process that we need to do largely by hand, but with the use of templates and a basic palette of styles, a straightforward 20 page article can be completed in a matter of minutes.

2.3 Transformation to XML

Once the articles are in RTF format and have styles applied, we place them on a Unix server. To do the transformation to "textclass" we use a combination of tools: Logictran's RTF Converter, Perl scripts, and sp with various DTDs for validation, all brought together with Unix shell scripts.

RTF Converter is conversion program that allows the user to specify tags associated with each paragraph and character style. In its most simple use, it can easily create a flat tagged structure, which can then be further modified using Perl scripts. With a little more effort and some light programming skills, the RTF Converter also allows the user to use string and numerical variables, conditional statements and functions, and embedded Perl scripts to output a complex hierarchical, tagged document. In such cases, some additional clean-up with Perl is often useful and can simplify the scripting.

The RTF converter works particularly well for us because of its ability to quickly batch process entire directories, as well as to incorporate specific configuration files for specific directories. This makes it useful to transform entire issues or runs of journals, each with its own particular needs and customizations.

The RTF converter and associated Perl and shell scripts convert the RTF documents to a tagged format and attach the appropriate header. We use a intermediary encoding DTD for initial validation purposes. If an article does not validate against this intermediary DTD, we need to do one of several things: fix errors or simplify some of the structure or styles in the original article; modify one or more of the transformation scripts; or modify the DTD. Once the content is valid, a Perl script transforms the file to its final incarnation conforming to the "textclass.dtd" (the TEI-based DTD). It is validated once again and is then ready for normalizing and indexing by the search engine.

This two-step double-DTD transformation process is necessary because of the very general nature of the TEI DTD. TEI is a very loosely structured DTD designed to handle a wide range of texts. It is very flexible for describing text, but it is not very suitable for initial encoding and validation purposes. Since nearly any element may appear nearly anywhere, an article may be "valid", but it still may not reflect the actual structure of the original article, or the structure that we expect, or even any readable structure at all. Therefore we use an intermediary DTD for encoding, one which has a smaller set of elements, is more specific and restrictive regarding where it allows elements to appear, and is intended to ensure that a given group of articles share a similar structure. This DTD may vary from publication to publication, and may require regular modification, especially in the initial stages of setting up a conversion process for a new collection.

3 Challenges and Lessons Learned

The preceding section outlined how the conversion process works in a general way. In practice however it is more complex, with numerous technical and workflow challenges doing their best to make things go awry.

We work with a number of different content providers and the type of material, content, format, scale, structure and publication schedule of each particular publication can vary widely, as can the technical expertise of each content provider. And while work on each of the separate publications follows the general outline described above, managing many ongoing publications in parallel, each with its own minor or major variations, can be a challenge. There is often a tension between, on the one hand, making the system as generic as possible so that our small unit can work efficiently and scale up the amount of publications we handle, and, on the other hand, customizing the process for each publication's particular needs, allowing the publisher to retain a greater control over the look, feel and functionality of the entire sight. The following sections outline some of the challenges and issues we face as we work with multiple content providers and a diverse range of content.

3.1 Working with Content Providers

Ideally, content would be submitted to SPO already encoded in XML, or at least well-structured in a word processing or page layout program. The transformation of already well-structured documents can be largely automated. SPO is prepared to work closely with content providers to coordinate our respective workflows, provide XML or MS Word templates, and to instruct content providers in creating XML. This has proved very successful in a several cases, such as with *The Medieval Review* and *Philosophers' Imprint.* In both cases, the editors submit articles to us as the articles become ready for publication, and the articles are published online the following day through a process that is either entirely automated or has minimal intervention. Significantly, both of these are electronic-only journals whose production processes are fairly flexible.

With established print journals we have found that we ourselves have to take on much of the work of structuring the documents. Files received from typesetters often are of little use to us because they are designed according to how they should look on the page rather than for their structural design. In such cases, we have found it simpler to simply strip the document of all styles and apply our own styles from scratch. The established print journals we work with are not as flexible in adapting their own processes, heavily dependent as they are on typesetters, production schedules, and structuring their content for print. Still unsure about their identity and role in the online world, and restricted by tight budgets, they are reluctant to commit to radical change in their own production processes. SPO has more flexibility than these publishers, and to facilitate publishing them on line we have to adapt to their production processes, rather than the other way around. One of our goals for the future is to push this data structuring up earlier in the process (to the content provider's end, if possible) by instructing them about the use of XML, and working with them to better coordinate our processes.

3.2 Workflow Issues

Maintaining many processes simultaneously has implications for the scale of our future activities. How much will we be able to scale up our operations? With most of our publications involving continuous, regular, ongoing submissions, our goal is to automate as much as possible. Enhancing our web-based submission system will greatly help in this area. Use of this centralized system allows us to automate significant portions of the process, such as the creation of article headers, the attachment of the headers to the articles, and the loading of the articles onto the servers. In the future we plan to make this system more robust by enabling it to accept a wider variety of file formats, allowing content providers to create their own metadata fields, and increasing editorial tools for creating, editing and tracking records and files. This will enable a greater number of journals to use this centralized system, standardize and consolidate many disparate processes, and simplify many of our own difficulties in managing several processes at once.

3.3 Conversion Tools

Finding the right tools to use also took some time, since when we began there was little information about how to get material into XML in a low-cost, efficient way. We proceeded largely by trial and error, and tried various software tools, including Avenue.quark in the hope that it would help us with the large amount of material that we receive in Quark format. But Avenue proved to be completely unworkable for our needs, having severe limitations and being extremely cumbersome to use. Eventually we settled on the Logictran RTF Converter, which was the best tool we found at the time that could batch process large amounts of material quickly, give us complex, hierarchical tagged output, be configured to work with Perl scripts and other tools, and run on Unix, Mac or PC platforms. It is inexpensive and provides good support, but it also involves a significant learning period, and requires some light programming skills to take full advantage of its capabilities. But it has proven to be quite powerful and suitable for our needs for the time being, especially when combined with some Perl scripts to help clean up the transformed files. There are a number of other tools that we have seen which will get material into a flat tagged structure, which can then be modified using Perl, but some that we have seen lack RTF Converters batch processing ability.

3.4 Lessons Learned

After one-and-a-half years of developing our conversion system we are able to make a few generalizations:

- Text conversion can be semi-automated but it is still a very hands-on process, especially during the initial set-up stages. It is unrealistic for us to expect, as we originally had hoped, to fully automate the process, especially if we want to continue to make our services available to a large cross-section of the scholarly community on campus.
- There is a constant need to deal with new and ongoing technical issues, learn new software, standards and procedures. This can range from dealing with seemingly minor issues (for instance, overcome a bug in exporting text from Quark) to learning entire new technologies, such as XSLT.
- Flexibility is important. The process needs to be flexible and robust enough to run and re-run the transformation scripts, track variations and changes in the files, integrate with Perl and shell scripts, and be configurable and easily modified to handle unanticipated elements in new documents and publications. Our intermediary, encoding DTDs are not written in stone, nor are our Perl transformation scripts. We continue to make regular changes to them as necessary to accommodate new document structures.
- The bulk of the work is in structuring the documents, which we try to do as early in the production process as possible. If the documents are well-structured then the conversion is usually very straightforward.

4 Conclusion

Despite the technical and organizational challenges our process has largely been successful. We are able to convert a steady stream of material in a short amount of time with a small staff (our conversions are done with one person preparing the data and maintaining the scripts part-time, with the occasional assistance of part-time students applying styles to the documents, managing the databases, and other activities). Some of our publications require us to walk through the conversion process step-by-step and do thorough editing on the document. With others, in particular those who publish only in electronic format, we have been able to automate the bulk of the conversion process.

Useful Links

- Digital Library Production Service (DLPS) www.umdl.umich.edu
- Digital Library Extension Service (DLXS) www.dlxs.org
- Logictran www.logictran.com
- The Medieval Review www.hti.umich.edu/t/tmr
- Michigan Quarterly Review www.hti.umich.edu/m/mqr
- Philosophers' Imprint www.philosophersimprint.org
- Scholarly Publishing Office (SPO) spo.umdl.umich.edu
- Text Encoding Initiative (TEI) www.tei-c.org
- TEI Text Encoding in Libraries: Guidelines for Best Encoding Practices www.indiana.edu/~letrs/tei