Background
The electronic editions listed above all aspire to become MLA Electronic Editions, and plans to submit them are underway. The MLA Scholarly Editions committee has a list of requirements for Electronic Editions (attached) that includes TEI encoding. TEI is an application of XML. There are several reasons for encoding documents in XML. During the first stage of digitizing, people created electronic editions that are now no longer readable (Mary Lynn Johnson describes such a problem with her ill-fated digital Blake edition). Editions that were originally encoded in HTML, such as the Penn Frankenstein, cannot participate in Web 2.0. That is, the documents themselves do not contain the rich bibliographic data that allows them to be maximally findable on the web, savable using programs for generating bibliographies such as Zotero or EndNote, or exportable to I-Phones, handheld, or e-Readers. No matter what new devices are invented in the future that require new formats, XML can be transformed into that new format. Also, one can export out of the XML document information that can be entered into a database and fed through programs to create information visualizations.

Thus one of the strengths of XML is that it can be transformed into other codes for various viewing devices. It is especially important in the case of scholarly editions that there be ONE master document of a text to ensure smooth workflow. If there is more than one master, errors can be introduced and corrections made only in one document and not its copies, creating multiple versions that are out of synch. It is the technical editor’s job to ensure that there is a workflow and to set up the systems to handle it, train project staff to work with it, and write documentation so that these procedures can be followed consistently. A schematic of this workflow (Figure 1) is at the end of this case study.

Part of this workflow is the creation of scripts to transform the XML into various formats as described above. My role is to create these scripts that transform the XML. To do this, I must be fully conversant, not only with the content of the scholarly project, but with its goals and objectives. That way I can work with the
editors of the project to design the TEI/XML so that the rest of the web site can be created. For example, if I’m creating a web page from a TEI-encoded letter to Robert Bloomfield, I have to be able to write the HTML, CSS, and Javascript used to create such a page as well as the TEI of the original document, and then also learn the declarative programming language that changes one into the other.

Scholarly Editorial Practice:

The editions for which I have consented to be technical editor are all of the highest scholarly quality. That quality has been adjudged in two ways. The first is peer review. Two venues for presenting scholarly works relevant to the field of nineteenth-century studies and Romanticism, Romantic Circles (http://www.rc.umd.edu) and NINES (http://www.nines.org) have peer reviewed most of these editions. The second means for judging quality is the funding received: Patricia Fumerton’s English Broadside Ballad Archive has been funded by an NEH Preservation Grant and awarded a prize for best electronic edition in the eighteenth century by the British Society for Eighteenth Century Studies.

The editors of these editions are scholars in the field:

The Bloomfield Letters, ed. Tim Fulford, Linda Pratt
http://www.rc.umd.edu/editions/bloomfield_letters
The Letters of Robert Southey, ed. Linda Pratt, Tim Fulford
http://www.rc.umd.edu/editions/southey_letters
The Penn Frankenstein, ed. Stuart Curran
http://www.rc.umd.edu/editions/frankenstein
http://morrisedition.lib.uiowa.edu/
The Poetess Archive, eds. Laura Mandell, Kathy Harris, Virginia Jackson, Eliza Richards
http://unixgen.muohio.edu/~poetess
The Yellow Book Project, eds. Denis Denisoff and Lorraine Janzen Kooistra
http://www.ryerson.ca/1890s/yellowbook.htm
The English Broadside Ballad Archive, ed. Patricia Fumerton
http://emc.english.ucsb.edu/ballad_project/

In each case, these editions describe their editorial procedures in creating scholarly texts.

Technical Editorial Practice:

One of the first things incumbent upon a technical editor is to determine the parts of the TEI code that will be used for any particular document. The TEI tag set of XML is large – over 300 elements: it makes possible encoding any kind of document you can imagine, from dictionaries to interviews. A subset of this code must be selected and
then a schema created which ensures that the people doing the coding are doing it consistently

Creating a TEI schema has been made easy by the TEI Consortium – they have developed a tool for it – but doing so requires making editorial decisions with wide-reaching effects. It is this TEI schema that I create in consultation with the scholarly editors, and then instruct their staff in how to create the TEI documents. Not only do I create documentation in HTML, but I have been experimenting with creating videos, which project staff have found extremely useful and often easier to follow than purely textual documentation. Some examples of documentation are at:

http://www.muohio.edu/chat
http://www.users.muohio.edu/mandellc/xml
http://unixgen.muohio.edu/~chat/xml

Sometimes training graduate students to become digital editorial assistants requires creating a whole manual, as I have done here:

My work as a technical editor has led me to work in the area of software development. My goal is to make it easier for scholarly editors to do some of the work I do without my assistance. I am developing this software in collaboration with computer scientists to serve two goals: to allow scholarly editors to create their own schemas and tagging manuals simply by making editorial decisions described in plain English. This tool is the topic of a co-authored article currently under review.

After the schema has been created, then the XSLTs or transformers can be written that will manipulate the edited pages of a scholarly edition in all the ways specified in Figure 1: to create web pages, to export data from them, to make them maximally searchable, to visualize them, and to make them readable on multiple devices.

In the case of each one of these scholarly editions, I have made sure to meet the requirements for an MLA electronic scholarly edition. That is, on every web page of these documents, one can click on a button that allows you to see the TEI master, the original, underlying code. This capacity constitutes editorial transparency of a sort not possible in print. A reader of Southey can see and print a finished letter, but can also go behind the scenes and see every editorial decision made by the editors for presenting the document. Also, the technical editing procedures are detailed either in the “editorial procedures” sections of the editions, along with scholarly procedures, as in Bloomfield and Southey, or in special sections such as:
http://unixgen.muohio.edu/~poetess/about/principles.html

MLA also requires making available the scripts used to transform the document (using a language called XSLT). I have been able to make them available at some sites: see for example
http://unixgen.muohio.edu/~poetess/bijou/Bijou1828corpus.xml-front2.html

Scholarly Value:
While it may seem that designing codes, coding, and writing programs to transform documents into multiple forms is sheerly mechanical work, in fact the set of routines constitutes a rigorous reading practice. As many critics have argued, to code a document is to interpret it. Literary critics interpret documents using many methods – feminism, new historicism, psychoanalytic theory. My reading has been through the lens of an emerging discipline called “Code Studies.” Slicing up texts up into chunks of smaller units and attempting to organize those units requires asking the following kinds of questions: A) what are the units of text, small and large, and what are their relationships to each other? B) can the relationships among these units be articulated in any given abstract language (code), and if some cannot, which ones and why not? And finally, C) what impact upon meaning is effected by rearranging these units differently?

I have produced one work of literary criticism that uses the code-studies reading methodology, called, “What Is the Matter? Or What Literary Theory Neither Hears Nor Sees,” and published it in *New Literary History* (attached). My book manuscript, the prospectus under review at several presses, also included in the dossier, is titled, *Print Cognition: the Secret Life of Information*.

---

2 This term was coined by Mark Merino.
The blue lines coming out of the octagon represent the master document coded in XML being transformed into those other kinds of documents coded in A, B, C, D, and E. The transformation process needs to be automated so that no errors are introduced into the original document’s content.

The scripting language used to transform the document into all these things is XSLT, a declarative programming language.

The workflow for creating Electronic Scholarly Editions: http://unixgen.muohio.edu/~chat/xslt/workflow.html