Using ODD for Multi-purpose TEI Documentation

Flanders, Julia
Julia_Flanders@brown.edu
Women Writers Project, Brown University, USA

Bauman, Syd
Syd_Bauman@brown.edu
Women Writers Project, Brown University, USA

The philosophy of "literate programming" (Knuth 1984), on which the TEI ODD is founded, proposes that code and documentation be written and maintained as a single integrated resource, from which both working programs and readable documentation can be generated. As currently designed, the TEI ODD system supports these goals, and there exist several good examples of extended project documentation written using the ODD customization file (see (Trolard 2009), (Burnard and Sperberg-McQueen 2006), (Burnard et al. 2010) and also the TEI Guidelines themselves). However, at present these examples only assume and demonstrate the ability to generate two types of documentation: a prose narrative and a set of reference documentation. As text encoding projects develop and mature they generate a variety of documentation that may include training tutorials and reference documentation, public documentation of editorial and encoding practices, documentation of their TEI customization, and internal documentation of the encoding decisions that have resulted in their current encoding rationale. Many of these other forms of output (such as training tutorials) have not been tried in practice and the current ODD processor, Roma, does not explicitly support them. In this paper we explore the possibility of generating more complex and varied forms of documentation using the TEI ODD customization file.

As background for this discussion we should begin by describing the nature and role of this customization file. Underlying any TEI-encoded document is a schema defining the terms of its validity, and underlying that schema is a further specification: the ODD customization file, which is the source file from which the schema is generated (documented in detail in chapters 22 and 23 of the TEI Guidelines (TEI 2007)). The ODD file serves several important functions:

1. To express the specific choices that are being made with respect to the TEI system as a whole: which TEI modules are to be included in the generated schema, which elements and attributes from these modules are to be included or omitted, changes to content models, etc.
2. To document those choices: for instance, to explain the meaning of controlled vocabularies for attribute values, or to express the rationale for applying an element only in specific contexts or with a slightly broader or narrower significance than described in the Guidelines.
3. To permit the generation (using these two kinds of information) not only of a custom TEI schema but also of custom documentation.

This custom documentation includes a re-expression of the TEI reference documentation: that is, the second volume of the printed TEI Guidelines, the portion containing separate entries for each element, attribute, class, and macro. This custom reference documentation includes only references to elements, attributes, and classes that are actually present in the custom schema, and includes as part of these entries some of the additional documentation expressed as part of point 2 above (e.g. glosses of specific attribute values, etc.).

The goal of the TEI customization mechanism as a whole, then, is two-fold. First, it aims to make TEI schemas self-documenting, by encapsulating all of the choices made in a separate document (the customization ODD file) that can be stored, maintained, exchanged. And second, to a certain extent the mechanism is intended to permit *encoding systems* to be self-documenting, in the sense that the documentation of the encoding practice can be bundled together with the raw materials for creating and maintaining the custom schema. This is true in a very straightforward manner to the extent that information about encoding practice can be directly associated with specific schema modifications. But more

complex documentation is also possible: since the ODD file is a TEI document, one can also include more detailed documentation that is not directly associated with a schema modification, simply by adding prose to that TEI document. When the ODD file is processed, the resulting documentation will include that additional prose. This more detailed approach is currently uncommon, but this is primarily because the Roma web interface does not currently support the authoring of such documentation; users need to author the ODD directly in order to create more complex documentation forms. Examples of this more detailed approach include the TEI in Libraries best practices documentation (Hawkins and Bauman 2009), and also the documentation for TEI Lite (Burnard and Sperberg-McQueen 2006).

With this in mind, however, it is tempting to extend this process even a step further: to use the ODD file as a way of writing documentation of other sorts that are even less closely attached (in their methods of organization) to the schema specifications. For example, for many purposes a project may need to maintain both reference-style documentation for each element or encoding concept, and also tutorial-style documentation whose emphasis is on leading the reader through a pedagogically structured narrative. Encoders learning to transcribe manuscript diaries might need to learn first the specific set of structural elements that will be used to encode the overall structure of the document (, <opener>, <dateLine>, etc.) and then the set of elements having to do with transcriptional difficulties (<gap>, <unclear>, <add>, <del>). At the same time, in other areas of the project it might be essential to have documentation of the underlying rationale for the encoding approach, or a high-level narrative with links to specific entries. More importantly, different tutorials or forms of documentation might need to use particular portions of the specification in different orders: the tutorial format might take specific sets of elements and treat them as groups, while a more comprehensive narrative documentation might treat the same encoding concepts in alphabetical order, or by conceptual grouping, or by TEI module (to take just a few examples).

To support the generation of multiple documentation narratives from a single ODD requires two changes to the way the ODD is written. First, additional prose (from which the various narratives will be generated) will need to be included in the ODD, and the ODD itself may need to be organized somewhat differently to accommodate this prose. Second, and more challengingly, some mechanism is required by which the different narrative orderings can be expressed in the ODD and then processed to produce the various appropriate forms of output. These different forms of documentation could of course be maintained separately (as is currently the case) but the value of the ODD system lies in its philosophy (expressed somewhat obscurely in the word "ODD", or "one document does-it-all") of having a single document that expresses and documents all aspects of a TEI encoding scheme. Rather than creating separate prose documentation for these additional forms, there is clear value in being able to generate multiple forms of documentation serving different purposes, from a single ODD.

We are not aware of any examples of this latter type, but the utility of this approach is clear and the ODD language – because it is part of the TEI and thus can use the full expressive resources of the TEI language as a whole – contains the features necessary to support it. In this paper we present an initial implementation in which we construct a set of tutorials on specific encoding topics in areas where the WWP has customized the TEI (for instance verse, title pages, and notes), using the ODD mechanism. The approach we explore entails encoding the components of the various narratives using standard TEI prose and documentation elements, and constituting each individual narrative using stand-off markup to identify and assemble the pieces in the appropriate order for the documentary form in question. We present the proposed ODD encoding for these tutorials, using the Women Writers Project internal documentation as a testbed. We also present prototype stylesheets that will generate multiple documentation narratives from the single ODD source.

References

Burnard et al.. *An Encoding Model for Genetic Editions*. http://tei.svn.sourceforge.net/viewvc/tei/trunk/genetic/geneticTEI.xml?view=markup.

Burnard, L. and Rahtz, S. (2002). 'The Role of the TEI in the Authoring and Interchange of XML Documents'. *Proceedings of elpub2002*. Berlin, 2002. http://elpub.scix.net/data/works/att/02-22.content.pdf.

Burnard, L. and Sperberg-McQueen, M. (2006). *TEI Lite: Encoding for Interchange: an introduction to the TEI*. http://www.tei-c.org/release/xml/tei/custom/odd/teilite.odd.

Hawkins, K. and Bauman, S.. *TEI in Libraries. Digital Library Federation, 2009*. http://github.com/sydb/TEI-in-Libraries/tree/master/BestPractices.

TEI (2007) (2007). *Text Encoding Initiative Guidelines for Electronic Text Encoding and Interchange*. TEI Consortium. http://www.tei-c.org/release/doc/tei-p5-doc/en/html/index-toc.html.

Trolard, Perry (2009). *TEI Tite: A recommendation for off-site text encoding*. http://www.tei-c.org/release/xml/tei/custom/odd/tei_tite.odd.

Knuth, Donald E. (1984). 'Literate Programming'. *The Computer Journal*. **27(2)**: 97-111. http://www.literateprogramming.com/knuthweb.pdfdoi:10.1093/comjnl/27.2.97.