**HOLMES — Whatever**

<0 images>

**Whatever Happened to Interchange?**

Holmes, M.

**Interchange and Interoperability**

In his 2011 paper at the Balisage conference, Syd Bauman (2011) draws a clear and useful distinction between two terms that are often lazily conflated: *interoperability* and *interchange*. Interoperability, he says, involves ‘little or no human intervention’; data is pulled from one site or system and plugged unchanged into another. Interchange, on the other hand, involves human actions, either through direct communication between individuals, or through indirect human mediation in which the re-user of data makes an effort to learn about it through reading documentation supplied with it, and then modifies or transforms it in order to put it to use in his or her own work.

The Text Encoding Initiative, as Unsworth (2011) points out, originally aimed at interchange rather than interoperability, with the ‘I’ in TEI standing in the Guidelines document itself for ‘Interchange’ rather than ‘Initiative’. Perhaps because of the increasing range of big-data projects and tools looking for texts and corpuses they can absorb and process, the focus nowadays is on interoperability and on the difficulties and failures we encounter with it (see, for example, McDonough [2009] for a detailed discussion of the chequered history of interoperability in XML encoding standards used in libraries and archives). But interoperability is hard, especially in the case of literary and historical encoding projects with unique interests and research agendas producing densely encoded texts. The TEI schema is as large and complex as it is because its user community has a wide variety of precise requirements and the need to express them at different levels of specificity. As Bauman reminds us, ‘Interoperability and expressiveness are competing goals constantly in tension with each other’, and encoding with interoperability in mind results in encoding likely to be ‘substandard, less faithful to the document, or outright incorrect’. We obviously do not want this; but when Desmond Schmidt (to take one example) complains that ‘because the choice of tags is guided by human interpretation, TEI-XML encoded files are in general not interoperable’, it is this very expressiveness he is complaining of.

In this paper I would like to re-focus on interchange, for several reasons. Interchange is a common way for TEI files to be re-used, and this is especially common in teaching environments and among novice users. Despite this, other than insisting that detailed documentation is essential and expecting re-users to read that documentation, we do little in the way of encouraging and assisting in the re-use of our data. Paying more attention to the needs of users who might want to take and work with our TEI files has a number of significant benefits to the source project, not only in terms of ordinary documentation, but also with regard to clarifying, streamlining, and normalizing practices and schemas. Finally, I would argue that making sincere efforts to enhance the practicality of interchange actually contributes significantly towards enabling interoperability, the much harder problem. I will be primarily concerned with TEI here, but the principles involved are generalizable to any formal data structure.

**Barriers to Interchange**

If we share our XML at all (and many projects do not), we typically share a single version of it: the version we created and customized in order to be as expressive, as faithful to the data, and as useful for the goals of our project as possible. On the face of it, this is rather strange. Most modern web-based digital edition projects (to take one large subset of the XML encoding world) provide a wide variety of output formats, including various web views (for desktops and mobile devices), printable formats such as PDF, and even plain text. Interactive web views of documents often allow the user to choose between a variety of display options—TAPAS, for instance, allows the user to switch between diplomatic and normalized views, and between ‘TAPAS Generic’ and ‘TEI Boilerplate’ renderings.1 It is regarded as good practice to try to predict the variety of

users who will visit a project, and to answer their needs as flexibly as possible. And yet, when it comes to our XML data, we do not do this; we typically provide a single XML dump for a document (in the TAPAS case, ‘Download TEI’).

In most projects, this XML view is likely full of peculiarities of encoding, rarely used elements and attributes, intra-project links, and other features that, even if well-documented, will present barriers to re-use. A brief analysis of the encoding in our Map of Early Modern London project shows many such issues, from which I examine three:

• Use of many project-specific private URI schemes for linking:

**<pb facs=‘moleebo:13311|2’/>**

• Complex encoding of dates with custom dating attributes:2

**<date notBefore-custom=‘1438-08-31’ notAfter-custom=‘1439-08-30’ datingMethod=‘mol:julian’ calendar=‘mol:regnal’>**17thof

**<name ref=‘mol:HENR2’>**Henry VI**</name></date>**

• Standoff markup in the form of pointers to personographies, taxonomies, etc. in other files within the project:

**<l>**The honour due by graue Antiquitie,**</l>**

**<l>**Then giuen to **<ref target=‘mol:LOND5’>**London**</ref>**s

**<name ref=‘mol:DRAP3’ type=‘org’>**Draperie**</name>**,**</l>**

**<l>**By Royall **<name ref=‘mol:RICH2’>**Richard**</name>**, who in me,**</l>**

All of these encoding practices are perfectly reasonable, and are well-documented, but they are likely to stand in the way of scholars wishing to take one of our documents out of its surrounding infrastructure and work with it in another context. The solution is obvious: in all of these cases, we can replace the problem feature with a more common encoding. For example, to make our date encoding more broadly useful, we could easily convert the date from the Julian calendar to the proleptic Gregorian, and remove the **@datingMethod** attribute:

**<date notBefore=‘1438-09-09’ notAfter=‘1439-09-08’ calendar=‘mol:regnal’>**17th of

**<name ref=‘mol:HENR2’>**Henry VI**</name></date>**

Similarly, we can expand the private URI schemes into full URLs, so that **‘moleebo:13311| 2’** becomes:

**<pb facs=‘http://eebo.chadwyck.com/fetchimage?**

**vid=13311&page=2&width=1200’/>**

Intra-project links to personographies, gazetteers, bibliographical references, and similar resources can be expanded in similar ways. However, many users will prefer a version of the document that does not depend on resources in linked files. We can provide for this by importing external resources into the appropriate locations in the host document, creating, for example, **<particDesc>** containing a **<listPerson>** in the header, and a **<listBibl>** for a bibliography in the **<back>**.

With strategies such as these, we can create a complete, standalone version of the encoded document that eschews unusual project-specific encoding practices.

**Targeting Specific Output Formats**

Producing a more standardized and self-contained document is one step on the path to effective interchange, but it will be useful only to a subset of potential users. The TEI schema, even if we restrict ourselves to the more common encoding practices, is still huge and daunting particularly to novice users, and problematic for the tools they are likely to use (TEI stylesheets, collation tools, rendering tools such as TEI Boilerplate,3 and so on). We can go further by providing output in simpler, more constrained versions of TEI, such as TEI Lite4 or the nascent TEI Simple.5

Naturally, there are tradeoffs here. We might characterize such conversions as lossy downsampling; most highly expressive project-specific encoding strategies will suffer in the translation to a much simpler and more constrained format. For example, the current version of the TEI Simple proposal states that ‘We will preclude use of **@rend** and **@style’**, and ‘We will produce a closed list of values for **@rendition** using a pseudo-protocol of “simple:”’. Our project uses **@style** and **@rendition** extensively with CSS code to describe the appearance, layout, and typography of primary source documents. To convert this to TEI Simple, we will have to create a complex system that analyses that CSS code and converts it to the nearest possible representation in ‘simple:’ tokens, discarding information that cannot be represented. Similarly, Simple (currently) lacks the **<mentioned>** element, which we use extensively; this will have to be mapped to a more generic representation, such as **<seg type=‘mentioned’>**.

This is an uncomfortable prospect, perhaps, but it does have some positive aspects. First of all, it causes us to evaluate all our encoding practices and to consider whether we really do need to use a particular formulation when something plainer might do. Second, downsampling or simplification in the process of rendering a widely used constrained format constitutes what we might call ‘enacted documentation’; by mapping our encoding to a simpler representation, we provide another level of description of our encoding practice. For example, this template, part of a conversion to TEI Lite, shows how the <supplied> element, not present in TEI Lite, is replaced with a <seg> element, documenting in the process not only how the output is produced but how the MoEML project typically uses <supplied> and its attributes:

<!-- No <supplied>. Have to make do with <seg>. -->   
 <xsl:template match="supplied">  
 <seg type="supplied" n="{concat(@reason, if(@evidence) then concat('; evidence: ', @evidence) else '')}">  
 <xsl:apply-templates select="@\*|node()"/>  
 </seg>  
 </xsl:template>

It might seem that such conversion processes should be undertaken by the end user, not the source project. However, in dealing with unusual encoding practices or rare elements and attributes, the source project’s encoders will have a much better understanding of what is intended, and how best to represent it in the target output; the example above, for instance, shows that in MoEML documents, <supplied> can be assumed always to have @reason, and often also @evidence. An outsider coming to the project may not be willing or able to undertake the work involved in reading the project documentation and building an appropriate transformation such as this; and even worse, an automated tool may simply throw away important information without comment. For instance, the Abbot Text Interoperability Tool,6 in converting one of our project documents to TEI Lite, simply throws away without comment the five **<egXML>** elements that are the heart of the document; these could actually be represented quite well using <eg>, which is available in TEI Lite. When lossy conversions are written by the progenitors of a project’s data, they are likely to be considerably less lossy.

**Conclusion**

While it has been argued that TEI XML has not lived up to its early promise with regard to interoperability, the focus on interoperability (unmediated re-use) has largely eclipsed interchange (human-mediated re-use), which is far more practical in most cases. By putting some effort into providing a range of different versions of our XML source encodings, targeted at different use-cases and user groups, we can greatly facilitate the re-use of our data in other projects; this process causes beneficial reflection on our own encoding practices, as well as an additional layer of documentation for our projects. Furthermore, by generating specific highly constrained output formats such as TEI Simple, we also help to facilitate interoperability.

In the Map of Early Modern Londonproject, we have been enhancing the variety of XML renderings of our documents, and now provide four different views: ‘Original,’ ‘Standard,’ ‘Standalone’, and TEI Lite; see, for example, http://mapoflondon.uvic.ca/METR1.htm. TEI Simple will be added when it stabilizes. A further documentation page7 provides an explanation of how each of these formats is derived, and suggests what kind of end-use it might be appropriate for. We are aiming to present a model example of how we might stimulate interchange and contribute to interoperability. Based on the demonstrable value of outputs created specifically for interchange, I would argue that such outputs should be incorporated as a formal component of plans and grant applications for all digital edition projects, and that granting agencies should be encouraged to look for and value this factor in project plans.

**Notes**

1. See, for example, http://tapasproject.org/digital-mitford/lettertotntalfourd28oct1821/16145.

2. See Holmes, Jenstad, and Butt (2013) for an argument that sophisticated encoding of historical dates is essential for some types of interoperability.

3. http://teiboilerplate.org/.

4. http://www.tei-c.org/Guidelines/Customization/Lite/.

5. https://github.com/TEIC/TEI-Simple.

6. http://abbot.unl.edu/cocoon/vicar/Vicar.html.

7. http://mapoflondon.uvic.ca/xml\_outputs.htm.

**References**

**Bauman, S.** (2011). Interchange vs. Interoperability. In *Proceedings of Balisage: The Markup Conference 2011*. Montréal, Canada, http://www.balisage.net/Proceedings/vol7/html/Bauman01/BalisageVol7-

Bauman01.html.

**Holmes, M., Jenstad, J. and Butt, C.** (2013). Encoding Historical Dates Correctly: Is It Practical, and Is It Worth It? In Digital Humanities 2013 Conference Abstracts, Lincoln, NE, http://dh2013.unl.edu/abstracts/ab-179.html.

**McDonough, J.** (2009). XML, Interoperability and the Social Construction of Markup Languages: The Library Example. *Digital Humanities Quarterly,* **3**(3),

http://digitalhumanities.org/dhq/vol/3/3/000064/000064.html.

**Rahtz, S.** (2014). TEI Simple: Summary. *TEI Simple GitHub Repository*,

https://github.com/TEIC/TEI-Simple/blob/master/advisory/report1.xml.

**Schmidt, D.** (2014). Towards an Interoperable Digital Scholarly Edition. *Journal of the Text Encoding Initiative,* **7**, 10.4000/jtei.979.

**Unsworth, J.** (2011). Computational Work with Very Large Text Collections:

Interoperability, Sustainability, and the TEI. *Journal of the Text Encoding Initiative,* **1** (June), 10.4000/jtei.215.