Journal of the Text Encoding Initiative

Issue 4 (March 2013)
Selected Papers from the 2011 TEI Conference

Joel Fredell, Charles Borchers IV and Terri Ilgen

TEI P5 and Special Characters Outside Unicode

Electronic reference

Publisher: Text Encoding Initiative Consortium
http://jtei.revues.org
http://www.revues.org

Document available online on:
http://jtei.revues.org/727
Document automatically generated on 08 August 2013.

Joel Fredell, Charles Borchers IV and Terri Ilgen

TEI P5 and Special Characters Outside Unicode

1. Introduction

1 One of the major challenges facing TEI encoders of older documents (ancient and medieval manuscripts, early print, manuscripts transcribed in modern print editions) is the range of special characters and abbreviations that they contain. This issue is especially critical for online documentary editions, where the goal is to present the digital facsimile of a manuscript alongside its electronic transcription. Many of the characters in the facsimile will have no exact correlatives even in Unicode, let alone in standard fonts (e.g. Times New Roman, Garamond, Palatino). In response, scholars have fallen back on deeply problematic strategies, such as substituting images for glyphs or using rough, modern equivalents that are usually historically and/or linguistically inaccurate. A "normalized" transcription, which translates all such characters to some modern equivalent, does have its uses—for example, as a reader edition and/or as an accessible edition for the visually impaired (both of which would otherwise be complicated by medieval abbreviation)[1]—and should, therefore, be part of any scholarly edition presented on the Web. However, such a transcription is far less useful to researchers interested in examining special characters as part of the dialectical and paleographical studies important to early periods; students and non-students alike also deserve a full documentary transcription that preserves the historical character of the original manuscript. And, quite simply, for those of us who work on texts surviving in unique copies, the ideal of reproducing in XML a text with all of its artefactual manifestations (so that the encoded description could substitute for the artefact if the latter is lost) is particularly resonant.

2 The stated goal of TEI—to develop and maintain "a standard for the representation of texts in digital form"—sets up a double challenge: machine-readable code on one side; its visual display on the other. Both forms of representation are crucial to translating and preserving texts in digital form. Nonetheless, our experience with TEI has shown a dramatic emphasis on the former part of the challenge—developing code that works seamlessly for XML—and surprisingly little support for the latter—developing code that guarantees accurate displays of complex characters, displays that users can identify easily with equivalent glyphs in the facsimile.[2] Reinforcing this imbalanced emphasis is the *TEI P5* standard for representation of non-standard characters and glyphs, which sets up a complex methodology for representing non-Unicode characters that is largely unconcerned—and at odds—with the practical issue of how these characters might actually be displayed (TEI Consortium 2012, chap. 5).[3] Still, as more and more editions of texts move to the Web the relationship between XML and HTML, between coded representation and visual display, has become increasingly important. In this respect the TEI Guidelines are not ready to serve documentary sites trying to go live now, and to provide accurate display for transcriptions of their source manuscripts. This problem, crucial for scholars of texts from pre-modern and less-dominant language groups, we view as an ongoing challenge deserving wider attention among TEI members. In the interim, we offer some immediate technical solutions using Unicode Private Use Area (PUA) characters. We derive our PUA characters from the Medieval Unicode Font Initiative (MUFI), but our strategies for displaying these characters remain interchangeable with similar Unicode recommendations or character sets across disciplines and thus, we believe, offer hope to a broad range of projects struggling with issues of display left unresolved by Unicode and TEI.

2. TEI and the Challenges of Current Web Display

3 Those of us working with ancient and medieval texts face a set of Unicode charts where most of the punctuation, abbreviation marks, and other characters typical in manuscript production are missing or sequestered in PUAs ignored by virtually all fonts. Our Web-based project

currently features a complete digital facsimile of the sole manuscript for *The Book of Margery Kempe*. As we refine our planned parallel diplomatic edition, one of our areas of focus has been developing strategies not only to encode special characters in compliance with the *TEI Guidelines* and best practice, but also to ensure that those characters display properly. We will not accept a single blank box in our electronic transcriptions or force visitors to our site to download and install special fonts before they can access these transcriptions fully.

4 Part of the imbalance in TEI between representation and display solutions for special characters may be a consequence of print bias: many scholarly teams working on texts from the print era have not faced substantial bodies of characters that are non-standard for modern fonts. Consequently, these scholars may not mind the occasional blank box popping up in their displays of online text, particularly if the blank box (which frequently indicates the user's browser's failure to find a character among those fonts installed on his/her computer) can be resolved by the user's installation of a font downloadable from a project's Web site. As print artefacts massively outnumber surviving works in manuscript form, scholars working with print-era texts dominate the humanities and, most likely, membership in TEI.[4] Thus TEI members from this latter group may see text to be coded (especially when the characters that they use have Unicode counterparts) as having a more direct relationship between initial representation in XML and ultimate display in HTML.[5] Of course, one intent of TEI is to facilitate (blind) interchange and interoperability of humanities data with support for a wide variety of use cases, not only HTML Web pages. The foremost challenge in this respect is generating metadata for systems such as COinS or RDF that can be used to render large bodies of data across multiple projects into interoperable and searchable assets: text strings in these metadata formats cannot tolerate markup outside Unicode, so standardization for these systems is essential. Consequently, newly-recognized special characters cannot simply remain in PUAs indefinitely, but must be put into the Unicode pipeline for review, consideration, and eventual acceptance if texts using what are now non-standard characters are to be included in larger databases.

5 Nonetheless, scholars working on ancient and medieval manuscripts face two more immediate problems. First, though many non-alphabetic characters are included in Unicode, these characters will display only in combination with fonts that include a glyph for the appropriate codepoint—and such support is far from guaranteed. Among users worldwide, choices of software vary widely, and even different versions of the same operating system, office suite, or Web browser can affect which fonts are available to the user and how (if at all) these fonts will render a character. Second, vast numbers of special characters are not included in Unicode nor have they been proposed for inclusion in Unicode. Among medievalists, for instance, one standard resource for interpreting frequent abbreviations by scribes, Adriano Capelli's *Dizionario di abbreviature latine ed italiane* (1996), includes thousands of examples of such abbreviations. Some of these abbreviations can be recreated with superscripted letters or combining diacritical marks, but many cannot, and those that can often play fast and loose with the semantics of the characters drafted. Editors often simply present their expanded versions of the abbreviations without the original scribal elements, but this strategy introduces a large mediating disjunction from the visual object. A similar problem exists with medieval punctuation, which uses both characters and pauses quite differently from their modern counterparts and can only be roughly reconstructed with Unicode characters.[6] Critical editions in traditional book form, restricted by the limits of print technology, regularly normalize on both counts by introducing expansions and modern punctuation equivalents; yet falling back on these old accommodations abandons one of the most important features of Web editions: visuality.

6 As Kathryn Sutherland has observed, the "only aspect of the book-bound text that the computer appears to simulate with any high degree of success is the visual" (Sutherland 2009, 20). Digital facsimiles are an inevitable and highly attractive consequence of scholarly editions moving to the Web. One of the principal forms in which medieval editions are migrating to the Web is diplomatic editions of single manuscripts, in the digital form now called "documentary editions" (Pierazzo 2011). In large part this trend is practical: the edition itself

is straightforward transcription rather than complex creation of a critical edition out of many texts. Rather than force us to depend entirely on the old print mechanism of the scholarly apparatus when transcribing works that survive in multiple witnesses, the Web enables us to reproduce every witness. But this opportunity is not without its challenges: in Chaucer's case, we must consider eighty-four separate manuscript witnesses (pre-1500) for the *Canterbury Tales*. As a result a number of early projects for digital critical editions have either stalled or moved to Web-based facsimiles and documentary editions for individual manuscripts.[7] Hence the problem of display has become an immediate concern.[8]

3. Applying the *TEI Guidelines* to *The Book of Margery Kempe*: A Case Study

7 In our case, we are encoding for a Web site, currently in prototype, that offers a high-resolution facsimile of the manuscript for *The Book of Margery Kempe* and a facing diplomatic transcription.[9] *The Book of Margery Kempe* survives in a single manuscript, but one for which a diplomatic edition was never published—the only standard edition is a hybrid: a critical edition with many silent editorial interventions, such as normalized text.[10] Our first objective, then, must be a production of the diplomatic edition that has never existed. Furthermore, as a mystical text, *The Book of Margery Kempe* is of great interest not just to academics and medievalists but to a wide body of general users; consequently our transcriptions could not simply represent specialized graphemes such as abbreviations and medieval punctuation with direct representations or with the standard scholarly approximations. We wanted all users working with a variety of platforms and browsers to have immediate and transparent access to the transcription; to have the ability to see the text representing as exactly as possible the characters in the manuscript as they were presented there; to be able to switch between this direct representation with abbreviations to an expanded version to ease the reading process for non-specialists; and to be able to separate the original text from several layers of commentary added by later medieval hands.

8 Clearly facsimiles with diplomatic editions are a good fit for the Web, but our global culture's move away from print consciousness is far from complete, and TEI P5 offers some valid, but incomplete, solutions to the problems of display faced by projects such as *The Book of Margery Kempe*. The charge that TEI is not well designed to address visuality, including bibliographic codes (in the term familiar from McGann 2001, 56) such as the many graphemes unique to the practice of medieval scribes, has been leveled by a number of critics, including some deeply sympathetic with the goals of TEI. Katherine Hayles has summarized this line of criticism well, citing the origins of TEI in the structuralist assumptions of OHCO—the text as an ordered hierarchy of content objects—assumptions that can be seen as a kind of New Critical desire for a platonic ideal of the text freed from the vagaries of its material manifestations (Hayles 2005, 89–96). James Cummings, while quoting Hayles's criticisms approvingly, points out that *TEI P5* in some respects addresses this criticism, although a longstanding principal concern for him and other scholars is the problem of coding competing physical hierarchies such as page breaks in a system designed to encode semantic hierarchies such as chapters and paragraphs (Cummings 2008; Renear, Mylonas, and Durand 1993). Remaining undiscussed are issues of transformation both theoretical and practical: how can (and should) XML represent the specific visual manifestations of non-print graphemes?

3.1 Coding Special Characters

9 *TEI P5* does, in fact, offer vastly improved guidance for coding abbreviations for display —introducing the <choice> tag, that can allow users to view the text in abbreviated or expanded form. Maintaining these options, particularly for side-by-side viewing of the facsimile and the manuscript, facilitates a direct transcription of the manuscript's bibliographic codes (such abbreviations are an important feature of the material culture of reading) and ease of use for non-specialist readers (who can toggle back and forth between a clear representation of what they see in the facsimile and what they can more easily understand in semantic terms).

This strategy usually presents little challenge for common abbreviations such as *wyth* that have no special characters

```
<choice>
  <abbr>w<hi rend="superscript">t</hi></abbr>
  <expan>w<ex>yth</ex></expan>
</choice>
```

but can occasionally lead to very long strings of code.[11] For example, the six-character word "dowtyr" requires seventy-three characters to encode when utilizing a <choice> containing both abbreviation and expansion elements. This length can actually double (to 154 characters) or even quintuple (to 373 characters) when highlighting and/or multiple-line, drop-capital characters are part of the word.[12]

fol. 9r, "dowtyr" coded with abbreviation and expansion (72 characters):

```
<choice><abbr>dowt &#x1DD1;</abbr><expan>dowt<ex>yr</ex></expan></choice>
```

fol. 9v, "Dowtyr" coded with highlighting, abbreviation, and expansion (154 characters):

```
<choice><abbr><hi rend="highlight" rendition="#RIA">D</hi>owt
&#x1DD1;</abbr><expan><hi rend="highlight"
rendition="#RIA">D</hi>owt<ex>yr</ex></expan></choice>
```

fol. 44v: "DOwtyr" coded with dropcap, highlighting, abbreviation, and expansion (373 characters):

```
<handShift new="#SALTHOWS"/><hi rend="prompt"
n="3">d</hi><choice><abbr><add rend="overwriting"
hand="#RED_INK_ANNOTATOR"><hi rend="dropcap" n="3">D</hi></add><hi
rend="highlight" rendition="#RIA">O</hi>wt&#x1DD1;</abbr><expan><add
rend="overwriting" hand="#RED_INK_ANNOTATOR"><hi rend="dropcap"
n="3">D</hi></add><hi rend="highlight"
rendition="#RIA">O</hi>wt<ex>yr</ex></expan></choice>
```

10 Wherever possible, we have turned to automation to facilitate choice encoding—creating standardized strings of code that replace found strings and that typically complete 80% of our coding, leaving mostly second-pass tasks to our encoders.[13]

11 Our success in encoding choices prompted us to consider the problem of display for many of the glyphs used in medieval manuscript abbreviation and punctuation. Where *TEI P5* describes ways to encode these glyphs in XML, these ways are complicated and verbose—requiring 1) <gaiji> in the body of a text; 2) character declarations in the header of a text; and/or 3) entity declarations in a project's schema file with instructions on what to do with these entity declarations when they are encountered in the project's XSLT, without prescription for how these characters should/could actually be displayed in HTML. One older strategy for representing scribal glyphs, used by the editors of the Auchinleck manuscript website and many editions on CD, is to provide a specially-created font for installation by the editions' users —an invasive solution that brings with it ease-of-use issues. Other editors, such as those at *The Newton Project* involved in the transcription of early modern alchemical manuscripts, have the advantage that Unicode provides at least some alchemical symbols. Since these symbols are unavailable in most fonts, however, the editors elect to use image files planted in the text to represent special characters. However, in Web browsers these image files do not scale when text is resized and become distorted when the entire page is resized, and so, generally, this approach is problematic.

12 A more recent strategy, used by the editors of the new Malory Project site, is to rely on Unicode exclusively for special characters and make do with whatever is available that displays more or less like the medieval facsimile. Again, problems across platforms emerge because a vast number of scribal characters simply are not available in Unicode. On the site, abbreviations are also left unexpanded, effectively limiting the site's use to medieval scholars. A bigger

problem for non-specialist users is the site's attempt to account for scribal features for which no print equivalent has been created: the site depends on characters in Unicode not created for medieval scribal glyphs. In one case, the editors use the print character barred-l or "ł" (U+019A from the "Non-European and Historic" Latin Extended-B chart) to indicate an otiose hairline (a common habit for this scribe) through the letter "l," so that "all" is rendered "ał." This approach does offer a somewhat analogous correlative from print for a scribal habit. Still, the chosen character has no relationship with otiose hairlines, and the resulting code is confusing both visually for the non-expert user and semantically for search.[14] We hasten to add that none of us can pretend to be pure about semantics for special characters at this point.[15]

13 Ideally we would have a set of Unicode characters that could represent all medieval glyphs, since authors of worldwide importance, such as Chaucer or Dante, along with hundreds of other literary figures from the period, need diplomatic transcriptions in digital documentary editions. And this is why PUAs exist: to provide a resource to scholars with which to propose the adoption of special characters (in Early Hungarian printing, for instance, or medieval manuscripts) by Unicode. As a result, some special characters have taken their place in the Unicode pipeline (http://unicode.org/alloc/Pipeline.html) and/or have become established within a PUA in the hope of eventual Unicode approval, as is the case for a cluster of characters developed by the Medieval Unicode Font Initiative (MUFI).[16] The problem with such private codepoints for characters has long been that no standard fonts support them, and that their display on the Web has been impossible without a specialized font downloadable and installable by the user.

14 *TEI P5* guidance on the use of PUA characters strongly emphasizes the in-text use of <gaiji>, which associate XML references with XML IDs described in a text's header—and which may omit any character reference to a PUA codepoint (that might be used in HTML) entirely —ostensibly so that these special characters can be identified consistently by other XML encoders, searched, and easily replaced in the event that Unicode creates sanctioned codepoints for them. However, the approaches we have seen in practice so far, while soundly reasoned, range from the very expansive, like this:

```
<char xml:id="eg-efa3">
        <desc>LATIN SMALL LIGATURE AF</desc>
        <charProp>
            <localName>entity</localName>
            <value>aflig</value>
        </charProp>
        <mapping type="MUFI" subtype="Unicode">af</mapping>
        <mapping type="MUFI" subtype="PUA">U+EFA3</mapping>
        <mapping type="standardized">af</mapping>
        <graphic url="images/efa3.png"/>
    </char>

    <g ref="#efa3">af</g>
```

Figure 1: TEI approach to coding special characters[i]

which utilizes character declarations, combinations of Unicode and PUA/MUFI mappings, "standardized" expansions of abbreviations, images, and gaiji to mark up and display representations of characters, to the more abbreviated, like this:

```
<!ENTITY aolig "&#xEF93;">
<!--Entity declaration for Latin Small Ligature AO-->
```

which still utilizes entity declarations (within a project's schema file) and entities in place of Unicode/hexadecimal code (reportedly so that, if a character in the PUA becomes a Unicode character, its new code point need only be updated once, in the entity declaration in the schema file—though, arguably, there are other ways to automate this process in XML/HTML code).

Nonetheless, both approaches only displayed special characters when the characters were supported by a font (or fonts) installed on the user's computer—substituting blank boxes when that font (or those fonts) was (were) not.

15 For its part in the discussion, TEI seems to fall somewhere between these two approaches. First, *TEI P5*'s instructions frown on expansive coding:

> For brevity of encoding, it may be preferred to predefine internal entities such as the following:
>
> ```
> <!ENTITY r1 '<g ref="#r1">r</g>' >
> <!ENTITY r2 '<g ref="#r2">r</g>' >
> ```
>
> which would enable the same material to be encoded as follows:
>
> ```
> Wo&r1;ds in this manusc&r2;ipt are sometimes written in a funny way.
> ```
>
> (*TEI P5* at 5.3)

16 Second, *TEI P5*'s instructions do suggest that there are ways to display special characters on the Web, but focus upon their descriptive markup in XML without any consideration at all for how they might be displayed in HTML (other than as images or Unicode characters). In the process these instructions cast doubts on PUA clusters such as MUFI as a viable alternative or even a supplement to Unicode for anything more than "local processing," as in the only discussion TEI offers about creating new characters:

> The creation of additional characters for use in text encoding is quite similar to the annotation of existing characters. The same element g is used to provide a link from the character instance in the text to a character definition provided within the charDecl element. This character definition takes the form of a char element. The element g itself will usually be empty, but could contain a code point from the Private Use Area (PUA) of the Unicode Standard, which is an area set aside for the very purpose of privately adding new characters to a document (*TEI P5* at 5.4).

17 The *Guidelines* go on to say that complex special characters may use pre-existing Unicode to construct the character as "a sequence of code points" in existing Unicode or "some locally-defined PUA character (say **) for local processing only." According to the *Guidelines*, however, neither of these approaches is desirable since "the former loses the fact that the sequence of composed characters is regarded as a single object [and] the second is not reliably portable" (*TEI P5* at 5.4).

3.2 Portability and Font Embedding

18 PUA characters can be reliably portable when custom fonts can be embedded directly in a Web page, where they can be loaded and rendered automatically by the Web browser—an approach that we have proven can work across browsers (e.g. Chrome, Firefox, Internet Explorer, Opera, and Safari) in our prototype. The option to specify which font or which set of fonts—called a "font family"—the Web browser should use to display text on a Web page has long been a fixture of popular Web design software. Importantly, however, this option merely notes a designer's font preferences and, in point of fact, offers little control over how text is actually displayed by the Web browser. Theoretically, if a designer specifies a font family that includes Georgia, Times New Roman, and Times, the Web browser will first attempt to display text using Georgia and, if that font is not available —that is if that font is not present or installed on the site visitor's computer— then Times New Roman, and, if that font is not available, then Times. If Times is not available, then the Web browser is supposed to default to some standardized font. But "standardized font" is a misnomer. Even among widely-used fonts—

like Times New Roman—there can exist substantial variation between different desktop operating systems and applications. And if we extend our discussion to mobile operating systems—those on tablet devices and smart phones—the problem is only further complicated.

19 Neither Unicode nor MUFI directly addresses this complication. The working groups for each are tasked with deciding how, if at all, a character should be represented in their respective standards or recommendations and, if so, at which code point. Though these groups are invaluable in this regard, they are not tasked with producing—or regulating—the fonts that will actually support approved characters. That responsibility falls to software and/or font developers. Accordingly, adding to the complication, not all code points are supported by all fonts. This is especially true—and, frankly, should be expected—with Unicode, which contains nearly 250,000 assigned code points. But even were we to narrow our focus just to MUFI's character recommendations, which contain far fewer code points (just over 1,500), we would find that only four fonts currently support the latest version of that standard. The first challenge, then, in resolving the font complication is finding and selecting a font that supports all desired code points. Ensuring that that font is actually displayed—and displayed correctly—by the Web browser is only possible at this point through font embedding.

20 Font embedding depends on the cascading style sheet, or CSS, code for the @font-face rule:[17]

```
@font-face
{
font-family: font-name;
src: url('font_name.ttf'),
     url('font_name.eot'); /* Embedded Font Type REQUIRED by IE 8 */
}
```

Figure 2: The @font-face rule

21 There is nothing particularly complicated about this code. TTFs, or TrueType fonts, have been around for decades and can be installed and used across operating systems and applications. EOT, or Embedded OpenType, is a font type invented by Microsoft for use with Internet Explorer versions 4–8, which did not and do not support embedded TTFs.[18] So if we want our IE users to see it, our selected font must have an EOT variant. If our selected font does not have an EOT variant—and most fonts will not—our next step might be to create one from the TTF that we selected. While there are a number of tools available to do this, we need to review our font's license to ensure that such conversion is permitted—noting that even if our font's license does permit conversion, that conversion may be flawed or fail altogether.[19] Equally important is that the font's license permits embedding. The alternative to using an existing font is, of course, creating one, but that only results in a new font, one that still has to be embedded.

22 Embedding a font so that it renders consistently across Web browsers is actually not as simple as the @font-face code above suggests—because different Web browsers read the same CSS code in different ways. Recent security changes in Firefox, for example, have necessitated that 1) the style sheet containing the @font-face rule(s), 2) the font(s) referenced by the rule(s), and 3) the Web page(s) that will use them share the same folder and that 4), in some cases, EOTs, which are not even processed by Firefox, actually precede other embedded font types in the rule(s).[20] Accordingly, to make the @font-face rule work, we have to rewrite it:

```
@font-face
{
font-family: Andron;
src: url('andron_scriptor_web_3_1.eot'); /* Embedded Font Type REQUIRED by IE 8 */
}
@font-face {
font-family: Andron;
src: url('andron_scriptor_web_3_1.woff'); /* Embedded Font Type Used by Other Browsers */
}
.medieval {
font-family: Andron;
}
```

Figure 3: Rewritten @font-face rule

23 We use this code to embed fonts on our project. The addition of the CSS class medieval is used to mark which text should be rendered with our embedded font, Andron Scriptor Web, on pages where both medieval and non-medieval text coexist.[21] More recently, we have also begun "wrapping" the TTF version of the font that we use for "other Web browsers" in the Web Open Font Format (WOFF), now supported by the current versions of all major Web browsers (except Internet Explorer 8). This, in addition to enhancing the security of the font, which we license, enables us to compress the enclosed TTF and to reduce its file size—improving site performance.

24 For visitors to our site who do not have the Andron Scriptor Web font installed on their computers, the code above causes the font to be loaded from our server and rendered by their Web browsers automatically and invisibly. No GIFs, JPGs, or other image types are used to display any of the text in our electronic transcriptions, so special characters can be enlarged alongside other text without distortion and without affecting relative proportions between even large embedded characters, like our drop capitals, and the body text.[22] The embedded font also offers other important advantages over its installed counterpart—including tighter control over versioning, which can and frequently does affect how special characters are rendered.[23]

25 Once a suitable font has been embedded in a Web page, displaying a PUA character becomes as simple as encoding its character reference—for example, ** (the same as it would be in XML)—in HTML. Accordingly, wrapping a character reference like ** in a <gaiji> (e.g. <g ref="#ur-leminskate"></g>) or even omitting the character reference completely (e.g. <g ref="#ur-leminskate"/>) in XML for later processing through XSLT (which will, effectively, restore the element to a simple character reference), strikes us as rather paradoxical—especially since the characters referenced are now demonstrably and reliably "portable" using embedded fonts.

26 No doubt, the objection that will be raised here is that our definition of portability does not actually meet the portability test. Our counter-objection would be that the definition of portability found in *TEI P5* does not always meet the portability test itself. Respecting characters, the portability test in *TEI P5* is really the Unicode test: does this character exist in Unicode? If it does, *TEI P5* allows the encoder simply to record a character reference: & for an ampersand, ‒ for an en dash, — for an em dash, and so on. For common characters, this reference is, generally, reliably portable: often, even the XML editor can display such a reference as a familiar glyph. But for less common characters and especially for new or uncommon ones, display may not be possible (for the reasons we discuss above), and the uninitiated encoder may be left scratching his/her head trying to figure out what ⎡ represents. Perhaps the character reference is defined somewhere in the XML document's header or in a schema. Perhaps not. In either case, the likelihood is that the encoder will have to look the character reference up in Unicode to understand its visual significance. Putting aside reliance upon an external source, to say that this approach constitutes portability depends on the continued existence and accessibility of Unicode. If Unicode ceases to exist (replaced by a new standard, for instance), undergoes significant revision resulting

in the reassignment of code points, institutes changes in how it is accessed, or becomes inaccessible (for whatever reason), the Unicode character reference ceases to be portable.

27 Admittedly, these are only hypotheticals, and all are highly unlikely. But they do illustrate a double standard in *TEI P5*'s definition of portability: if Unicode character references—which implicitly reference an external source—are considered portable, then why can't the character references from an alternative and/or emerging standard, like MUFI, also be considered portable? The question strikes us as particularly pertinent given that MUFI is largely a collection of Unicode character recommendations, updated as medieval characters are adopted by Unicode. Further, by adopting MUFI and the Andron Scriptor Web font, we have been able to ensure that every character reference included in our XML is defined (through consensus by an international body of academics, scholars, graphic designers, and information technology professionals) and displayable in one of three ways: 1) by visiting our site; 2) by saving site pages for offline viewing; or 3) by installing the Andron Scriptor Web font.

4. Conclusions

28 TEI P5 recommends using the empty <gaiji> in 1) the creation of a combined Unicode entity that has no semantic association with a specific historical character and/or 2) coding nonstandard characters, including characters for which there is reasonable hope of future inclusion in Unicode. The first recommendation clearly does not live up to the ideal of reproducing in XML a text complete with its artefactual manifestations so that the encoded description could substitute for the artefact if the latter is lost. And the second demands effectively that XML wrap with <gaiji> every character not absolutely standard—surely a difficult task dependent to some extent on guesswork.

29 Of the use of the <gaiji> to tag single PUA characters, the *TEI Guidelines* suggest:

> In the fullness of time, a character may become standardized, and thus assigned a specific code point outside the PUA. Documents which have been encoded using the mechanism must at the least ensure that this changed code point is recorded within the relevant char element; it will however normally be simpler to remove the char element and replace all occurrences of g elements which reference it by occurrences of the newly coded character (*TEI P5* at 5.5).

30 As we have previously argued, other mechanisms exist to find occurrences of any text string and replace it with another in XML. Moreover, both suggestions seem to proceed from the assumption that PUA characters are useful for "local processing" (e.g. processing by fonts installed on the user's computer) only—which is certainly no longer the case.

31 While we do concur that, for maximum transparency, these PUA characters (and their character references) should be defined somewhere, we wonder if, given the realities of font embedding, there is not some other way to define the character references themselves without having to resort to what seems unnecessary—or unnecessarily verbose—tagging. In her tagging of special characters for the Robert Southey Edition at Romantic Circles, Laura Mandell suggests what may well be such an alternative in the editorial declarations of texts' TEI headers:[24]

```
<editorialDecl>
        <normalization>
                <p>&amp;amp; has been used for the ampersand sign.</p>
                 <p>£ has been used for £, the pound sign</p>
                 <p>All other characters, those with accents, non-breaking
                    spaces, etc., have been encoded in HTML entity decimals.</p>
        </normalization>
</editorialDecl>
```

32 No doubt some in TEI will balk at this method of defining character references —preferring, at a minimum, the more verbose method of gaiji tags and character declarations. However, Mandell's method does, in fact, define these character references—and, arguably, in a way that is much more transparent (e.g. to non-encoders) than the methods suggested in the *TEI P5 Guidelines* on special characters (*TEI P5* at 5). which, again, we would argue were based on a different reality at the time that they were devised.

33 For the purposes of combined characters—be they PUA or those with assigned Unicode points—meant to represent a single "composed" character and for Unicode characters that have not been employed as prescribed by Unicode, we agree that additional tagging should be considered. For example, <g ref="#ur-leminskate"></g>, where ur-leminskate is a defined XML ID elsewhere in the project's code (perhaps in a text's header within a <charDecl>) and/or its documentation. Alternately, a project might opt to use named entities and entity declarations—or some combination of all of the above—in such cases.

34 In summary, *TEI P5* does not appear to prefer one methodology over the other for "representation of non-standard characters and glyphs" (or for representation of characters and glyphs used in non-standard ways). It does not offer clear use cases for the gaiji-character declaration or named entity-entity declaration, both of which seem to have been devised at a time when broad support for PUA characters was not possible (especially on the Web). We have demonstrated that this limitation is no longer insoluble. A project can, in fact, now embed fonts supporting such characters directly in its Web pages with very high degrees of cross-browser support to display PUA characters. Wrapping or replacing a character reference processable in both XML and HTML with TEI's <gaiji> for later processing through XSLT, which will restore the element to a character reference, makes little sense to us—especially when alternatives exist for defining characters and character references in XML without necessitating <gaiji> or elaborate header declarations.

35 While the case can be made that XML and HTML are descriptive markup languages with differing goals, as the Web becomes a more centralized social and cultural technology (not to mention the preferred way that scholarly editions are presented to audiences worldwide), the interchange between these two languages, as well as the opportunities and the consequences that arise from that interchange, must become more central to TEI. Digital editions that have embraced TEI and XML are far too often limited by print models for textual representation in the absence of guidelines, recommendations, or even exemplars for display. Since XML and HTML are frequently partnered in achieving a common goal, and since a fundamental goal of HTML is display, the problems of and solutions for display must become more central to TEI. It is our hope that TEI will soon embrace encoding solutions that will make possible new levels of accuracy and transparency in presenting the graphic features of texts as they are witnessed in their material artefacts—coding that respects the original purposes and meanings of the thousands of characters for which print and Unicode have never offered equivalents.

Bibliography

Burnard, Lou, Katherine O'Brien O'Keeffe, and John Unsworth. 2006. *Electronic Textual Editing*. New York: MLA.

Capelli, Adriano. 1996. *Dizionario di abbreviature latine ed italiane: usate nelle carte e codici specialmente del medio-evo riprodotte con oltre 14000 segni incisi*. Milan: Ulrico Hoepli.

Cummings, James. 2008. "The Text Encoding Initiative and the Study of Literature." *A Companion to Digital Literary Studies*, edited by Susan Schreibman and Ray Siemens. Oxford: Blackwell, 2008. http://www.digitalhumanities.org/companion/view?docId=blackwell/9781405148641/9781405148641.xml&doc.view=content&chunk.id=ss1-6-6&toc.depth=1&brand=9781405148641_b

Deegan, Marilyn. 2006. "Collection and Preservation of an Electronic Edition." In *Electronic Textual Editing*, edited by Lou Burnard, Katherine O'Brien O'Keeffe, and John Unsworth, 358–70. New York: MLA. Preview version accessed February 8, 2012. http://www.tei-c.org/About/Archive_new/ETE/Preview/mcgovern.xml.

Duggan, Hoyt. n.d. "Piers Plowman Electronic Archive." *TEI Projects*. http://www.tei-c.org/Activities/Projects/pi01.xml

Kiernan, Kevin. 2006. "Digital Facsimiles in Editing." In *Electronic Textual Editing*, edited by Lou Burnard, Katherine O'Brien O'Keeffe, and John Unsworth, 262–68. New York: MLA. Preview version accessed February 8, 2012. http://www.tei-c.org/About/Archive_new/ETE/Preview/kiernan.xml.

McGann, Jerome. 2001. "The Rationale of Hypertext." In *Radiant Textuality: Literature after the World Wide Web*, 53–74. New York: Palgrave. First published in *TEXT* 9 (1996):11–32; reprinted in *Electronic Text: Investigations in Method and Theory*, edited by Kathryn Sutherland, 19–46 (Oxford: Clarendon Press, 1997).

Meech, Sanford B., and Hope Emily Allen, eds. 1940. *The Book of Margery Kempe*. EETS 212. London: Oxford University Press.

O'Donnell, Daniel Paul. 2008. "Disciplinary Impact and Technological Obsolescence in Digital Medieval Studies." In *A Companion to Digital Literary Studies*, edited by Susan Schreibman and Ray Siemens, 65–81. Oxford: Blackwell. http://www.digitalhumanities.org/companion/view?docId=blackwell/9781405148641/9781405148641.xml&chunk.id=ss1-4-2.

Parkes, Malcolm B. 1992. *Pause and Effect: An Introduction to the History of Punctuation in the West*. Aldershot, England: Ashgate.

Pierazzo, Elena. 2011. "A Rationale of Digital and Documentary Editions." *Literary and Linguistic Computing* 26(4):463–77. doi:10.1093/llc/fqr033.

Renear, Allen, Elli Mylonas, and David Durand. 1993. "Refining Our Notion of What Text Really Is: The Problem of Overlapping Hierarchies." http://www.stg.brown.edu/resources/stg/monographs/ohco.html.

Robinson, Peter. 2005. "Current Issues in Making Digital Editions of Medieval Texts —or, Do Electronic Scholarly Editions Have a Future?" *Digital Medievalist* 1. http://www.digitalmedievalist.org/article.cfm?RecID 6.

———. 2006. "*The Canterbury Tales* and Other Medieval Texts." In *Electronic Textual Editing*, edited by Lou Burnard, Katherine O'Brien O'Keeffe, and John Unsworth, 74–91. New York: MLA. Preview version accessed February 8, 2012. http://www.tei-c.org/About/Archive_new/ETE/Preview/robinson.xml.

———. 2009. "What text really is not, and why editors have to learn to swim." *Literary and Linguistic Computing* 24(1):41–52. doi:10.1093/llc/fqn030.

Simpson, Grant Leyton, and Dot Porter. 2012. "Transforming Backward: HTML and HTML+RDFa to TEI." *Journal of the Text Encoding Initiative* 2. http://jtei.revues.org/407.

Stubbs, Estelle. 2000. *The Hengwrt Chaucer Digital Facsimile*. Birmingham, England: Scholarly Digital Editions.

Sutherland, Kathryn. 2009. "Being Critical: Paper-based Editing and the Digital Environment." In *Text Editing, Print and the Digital World*, edited by Marilyn Deegan and Kathryn Sutherland, 13–25. Aldershot, England: Ashgate.

TEI Consortium. 2012. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*, edited by Lou Burnard and Syd Bauman. Version 2.1.0. Last updated June 17. N.p.: TEI Consortium. http://www.tei-c.org/release/doc/tei-p5-doc/en/html/.

Vertan, Cristina, and Stefanie Reimers. 2012. "A TEI-based Application for Editing Manuscript Descriptions." *Journal of the Text Encoding Initiative* 2. http://jtei.revues.org/392.

Wittern, Christian. 2006. "Writing Systems and Character Representation." In *Electronic Textual Editing*, edited by Lou Burnard, Katherine O'Brien O'Keeffe, and John Unsworth,

291–98. New York: MLA. Preview version accessed February 8, 2012. http://www.tei-c.org/About/Archive_new/ETE/Preview/wittern.xml.

———, Arianna Ciula, and Conal Tuohy. 2009. "The Making of TEI P5." *Literary and Linguistic Computing* 24(3):281–96. doi:10.1093/llc/fqp017.

For Further Reading: Digital Editions

The Auchinleck Manuscript. David Burnley and Alison Wiggins, Project Directors. http://auchinleck.nls.uk/.

The Blake Editions of the Canterbury Tales. Orietta Da Rold, Simon Horobin, Estelle Stubbs, and Claire Thomson, Editors. http://www.shef.ac.uk/hri/projects/projectpages/blakeeditions.

The Book of Margery Kempe. Joel Fredell, Project Director. http://english.selu.edu/kempe/.

The Canterbury Tales Project. Peter Robinson, Project Director. http://www.canterburytalesproject.org/index.html.

The Cotton Nero A.x Project. Murray McGillivray, Project Director. http://people.ucalgary.ca/~scriptor/cotton/.

The Malory Project. Takako Kato, Project Director. http://www.maloryproject.com/index.php.

The Newton Project. Rob Iliffe, Project Director. http://www.newtonproject.sussex.ac.uk/prism.php?id=46.

The Piers Plowman Electronic Archive. Hoyt Duggan, Project Director. http://www3.iath.virginia.edu/seenet/piers/.

Notes

1 Envisioned as an extension of our existing XML codebase, such editions are already planned for our project, *The Book of Margery Kempe*. In fact, our web app currently features extensive built-in support for the visually impaired.

2 For this stated goal of TEI see the TEI home page, http://www.tei-c.org/index.xml. As Grant Leyton Simpson and Dot Porter have noted recently, "TEI, even coupled with CSS, is not, for the most part, designed for direct presentation [...] TEI XML is just one of many steps along the path toward generating the final product. That is, TEI must be transformed 'forward' to another format" (Simpson and Porter 2012). XSLT alone will not solve the problem of displaying special characters. For an overview of changes in *TEI P5* see Wittern, Ciula, and Tuohy 2009.

3 The TEI overview of transcription, "Representation of Primary Sources," 11.3.1.2, simply points to a section of the *Guidelines* discussing the use of <gaiji>s: "In cases where [Unicode] does not [support a particular glyph], these Guidelines recommend use of the <g> element provided by the **gaiji** module described in chapter 5 Representation of Non-standard Characters and Glyphs" (TEI Consortium 2012). On problems with the <gaiji> strategy see below.

4 Of the 150-plus projects listed on the "TEI Projects Page", http://www.tei-c.org/Activities/Projects/, we count seventeen directly concerned with texts from pre-print culture. Several projects span manuscript and print cultures as well, and arguments can be made for difficulties in addressing issues in handwritten documents from the age of print such as authorial holographs; in fact, our team has several members also working on the Ruskin Project (not yet online), trying to decide how to display characters from the notebooks of the young John Ruskin wherein he tries to imitate print effects in his handwriting. Also see our discussion of the Newton manuscripts below for the strategy of using image files for non-standard characters (in this case alchemical symbols within texts otherwise

consisting of standard characters). Nonethless, the vast preponderance of the work represented on the "TEI Projects Page" represents print-to-digital conversion.

5 See, for instance, the MLA-authorized discussion of electronic textual editing emphasizing interoperability and preservation—with no discussion of display—by Deegan (2006). A recent utility for editing manuscript descriptions, described by Vertan and Reimers (2012), also illustrates this priority. We acknowledge that many challenges remain in print, particularly for characters in early print and non-western languages. However, much more progress in these areas has been made than in the case of early scripts (aside from some ancient languages), as the Unicode Character Code Charts indicate; see http://www.unicode.org/charts/.

6 On these many punctuation characters and their uses in medieval England, see Parkes 1992.

7 One example of this trend is the Canterbury Tales Project. This ambitious project, overseen by Peter Robinson and Norman Blake and now moribund, intended to create a new critical edition out of the 84 surviving manuscripts and several early print editions of the *Canterbury Tales*. A few editions for individual tales have been completed, but out of that project instead came a new online initiative dedicated to facsimile/diplomatic editions for individual manuscripts called the Blake Editions. In 2003 the harbinger of this new initiative was the publication on CD of the Hengwrt Manuscript in facsimile edited by Estelle Stubbs (Stubbs 2000), one of the editors of the Blake Editions. It should be noted also that Peter Robinson's important contributions to early humanities computing (Robinson was an early member of TEI) were predominantly concerned with complex textual editing (as in his Cladistics software) rather than the "new bibliographic" focus on individual material witnesses to medieval texts. Robinson's recent publications show no signs of interest in the specific problems of digital facsimiles and diplomatic transcriptions, remaining firmly fixed on problems of creating critical editions in digital forms; see most recently Robinson 2005, Robinson 2006, and Robinson 2009. Also see the TEI statement (http://www.tei-c.org/Activities/Projects/pi01.xml) and Web site (http://www3.iath.virginia.edu/seenet/piers/) for the *Piers Plowman Electronic Archive*, which has evolved markedly since its inception in 1994. For an overview of problems with various early adopters, see O'Donnell 2008.

8 Kevin Kiernan (2006) all but admits this, but does not explore the issue of display beyond a stated need for description and illustration: "Focused, comprehensive access to scribal letterforms might be mediated through the glossary, by linking all head-letters to salient examples in the manuscript. However it is accomplished, examples of all letterforms should be described and illustrated [...] XML markup is good at distinguishing different letterforms, such as insular, caroline, and uncials, for searching of text, but to be of real value, the editor and the researcher should be able to link any search results to the specific instances in the manuscript images." For a similar avoidance of display problems see Wittern 2006.

9 The site, http://english.selu.edu/kempe/, currently has a full working facsimile and some text.

10 *The Book of Margery Kempe* has become a major text in medieval studies, now second only to Chaucer in frequency of teaching for college-level courses in the literature of late-medieval England. It survives in a single manuscript now housed in the British Library, MS Add. 61823. A number of teaching editions have been published in recent years, but all are based on a problematic critical edition from 1940 by Meech and Allan.

11 Use of the abbreviation marker element (<am>) in medieval manuscripts is frequently problematized by the irregularity of scribal abbreviations, many of which occur as strokes, collectively called scribal sigla, above or through a single character. Contextually, this type of sigla may be interpreted as the abbreviation

of several letters before and/or after the letter to which it has been applied. In many cases, this type of sigla is also best represented in XML/HTML in a precomposed Unicode character for which only one of the underlying glyphs (i.e. the *siglum*) may actually represent an abbreviation. The *Latin small letter p with stroke through descender* (U+A751), for example, may be decomposed into a *Latin small letter p* (U+0070) paired with a diacritic *stroke through descender*, which medievalists might interpret as an abbreviation of *er*, *ar*, *or*, or *ri*—resulting in *per*, *par*, *por*, or *pri* (*p* + *er*, *p* + *ar*, *p* + *or*, *p* + *ri*) based upon context. Given then 1) that it would be inappropriate to mark a precomposed character with an <am> and 2) the sheer number and irregularity of scribal abbreviations of the single character + siglum type (for which precomposed characters are best suited) in *The Book of Margery Kempe*, we have elected, for accuracy and consistency, not to use the <am> in our encoding of <choice> and, instead, to treat the contents of the abbreviation (<abbr>) as a single, abbreviated unit.

12 Here the @n records the size of the drop capital *D*, as well as of the original space reserved for it by the prompt *d*. Within *The Book of Margery Kempe*, drop capitals tend to span three to four lines, represented by n="3" or n="4". The <hi> "marks a word or phrase as graphically distinct from the surrounding text" (*TEI P5* at 3.3.1). Coupled with the @rend, the element may be used to mark italicized, bolded, underlined, colored, or virtually any other variation of text—including text that has actually been highlighted. Within *The Book of Margery Kempe*, a hand that we identify as the "Red Ink Annotator" frequently highlights text ascribed to the manuscript's primary hand, "Salthows." Marking major divisions within the manuscript, denoting agreement or disagreement with Salthows's grammar, or suggesting emphasis in places where Salthows has indicated none, this highlighting has important text-critical implications that can be fully understood only when responsibility for it is appropriately documented. Unfortunately, *TEI P5*'s <hi> does not natively support the @hand, used "to signal the person responsible (the hand) for the writing of a whole document, a stretch of text within a document, or a particular feature within [a] document" (*TEI P5* at 11.3.2.1). Consequently, another early challenge that our project team faced was separating responsibility for highlighting from text. Briefly, our solution makes use of the @rendition—which is supported by the <hi>—and an abbreviated XML ID that corresponds to the XML ID for each of the hands that we have identified within *The Book of Margery Kempe*: *RIA*, thus, corresponds to *RED_INK_ANNOTATOR*. The abbreviated XML IDs are documented within a <tagsDecl>—the XML IDs within a <handDesc>—within our <teiHeader>. This solution is detailed in full on our project site.

13 We routinely employ Digital Volcano's freeware tool TextCrawler (http://www.digitalvolcano.co.uk/content/textcrawler) for find-and-replace operations across multiple files (not all of them XML) simultaneously. Our use of the tool required an analysis of word frequencies to determine which words abbreviated or marked by annotators were best coded with automation. For this analysis we have relied on the web-based Voyeur Tools (http://hermeneuti.ca/voyeur).

14 The problem of such semantic disjunctions is raised in Wittern 2006, though without specific display solutions.

15 See note 10 above for one aspect of this problem for the Kempe site. Another approach to this problem can be found on the *Cotton Nero A.x Project*; see their transcription for *Cleanness* and discussion of transcription policies.

16 A number of other projects incorporate MUFI's character recommendations in their XML encodings: The Corpus of Early Hungarian Printed Books (http://korpusz.ektf.hu); ENRICH (http://enrich.manuscriptorium.eu); and Medingen Manuscripts (http://research.ncl.ac.uk/medingen). The ENRICH gBank application http://www.manuscriptorium.com/apps/gbank/ is particularly

valuable as a tool for browsing MUFI characters and adding TEI code to one's transcriptions, though it wraps each character in a <gaiji> tag; see below on the <gaiji> strategy.

17 While support for CSS is user-definable—a built-in feature, it can be modified and even disabled within the Web browser—the fact that CSS is built-in and now widely used on the Web (in controlling the overall appearance of Web sites) means that it is enabled by default in the current versions of all major Web browsers. Our strategies for display assume (and have been tested to work with) the default settings of these browsers.

18 This issue changed with the release of IE 9, which does now support embedded TTFs. Unfortunately, this latest version of the Web browser currently runs only on Windows Vista and 7—leaving XP users stuck at version 8.

19 We have developed a tool to test—or "prototype"—special font characters and abbreviations, called PROTOtypeR, available on our site for general use.

20 Tested through version 10.0.

21 Andron Scriptor Web (http://www.signographie.de/cms/front_content.php?idart=69&changelang=2) was developed by Andreas Stötzner for MUFI; the font contains all of the PUA characters approved by that international group of medieval scholars and every special character used by our project.

22 The marginalia contain drawings and other elements which will call for other strategies. Although a set of medieval dingbat-like characters is tempting, widespread and consistent use must be a precondition for the creation of a character in a PUA just as for Unicode itself.

23 A new version of an embedded font (e.g. one with enhanced character support and/or features) can be uploaded to the server as soon as it is available—making it immediately usable by site visitors.

24 Mandell, in an e-mail message to the authors (January 25, 2012), notes that she uses the decimal character reference *£* to encode the pound sign, but that Oxygen displays the symbol instead of the character reference. We would use the hexadecimal character reference *£* to encode the pound sign since hexadecimal character references are directly relatable to Unicode points—i.e. *£ = 00A3*.

Endnotes

i Editor's note: because of the PUA character present in this example, the editors have opted to provide this example as an image. For users interested in the textual form of the encoding, this is rendered here (note that there's little chance the UAP character in line 7 will display properly):

Cite this article

Electronic reference

Joel Fredell, Charles Borchers IV and Terri Ilgen, « TEI P5 and Special Characters Outside Unicode », *Journal of the Text Encoding Initiative* [Online], Issue 4 | March 2013, Online since 11 March 2013, connection on 08 August 2013. URL : http://jtei.revues.org/727 ; DOI : 10.4000/jtei.727

Authors

Joel Fredell
Joel Fredell is Professor of English at Southeastern Louisiana University and project director for the online documentary edition of *The Book of Margery Kempe*.
Charles Borchers IV

Charles Borchers IV recently earned his M.A. in English from Southeastern Louisiana University, where he remains a Consultant for Digital Humanities Projects and continues to supervise encoding and Web development for the online documentary edition of *The Book of Margery Kempe*.

Terri Ilgen

Terri Jo Ilgen is an M.A. candidate in English at Southeastern Louisiana University and encoder for the online documentary edition of *The Book of Margery Kempe*.

Abstract

Index terms

Keywords : diplomatic editions, documentary editions, special characters, font embedding, Private Use Area (PUA), Medieval Unicode Font Initiative (MUFI), Unicode, gaiji