

Schriften des Instituts für Dokumentologie und Editorik — Band 12

Digital Scholarly Editions as Interfaces

edited by

Roman Bleier, Martina Bürgermeister, Helmut W. Klug,

Frederike Neuber, Gerlinde Schneider

2018

BoD, Norderstedt

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Digitale Parallelfassung der gedruckten Publikation zur Archivierung im Kölner Universitäts-Publikations-Server (KUPS). Stand 20. November 2018.

2018

Herstellung und Verlag: Books on Demand GmbH, Norderstedt

ISBN: 978-3-7481-0925-9

Einbandgestaltung: Julia Sorouri; Coverbild: "Desktop" von Franz Konrad (www.franzkonrad.com/gallery/desktop-2008-2010) .

Satz: Roman Bleier und Lua \TeX

Critical Editions and the Data Model as Interface

Hugh A. Cayless

Abstract

Critical editions of classical text pose some unique problems that highlight the importance of the editor's contribution to the creation of the edition. The article discusses the issues involved in creating these editions and proposes a method for creating digital critical editions that foregrounds the edition's data model, thereby enabling an intuitive and powerful interface for reading digital critical editions online. It presents the results of experiments with this method that are being undertaken by the Digital Latin Library project, a joint effort of the Society for Classical Studies, the Medieval Academy of America, and the Renaissance Society of America.

There are many interfaces involved in the creation and use of digital critical editions. Not just the ones presented to readers, but those in play in the data formats used and between the layers of software employed to render the "user interface". As a beginning, it is worth enumerating some of the functions of interfaces:

1. They hide implementation details and complexity.
2. They serve as a contract governing the interaction between two parties.
3. They serve as generalizable protocols.

These functions are obviously interrelated. Interfaces define rules for interaction between two parties (parties that may be human or machine). These rules pare down the set of things one party can ask another to do, and because they reduce that set of interactions, they (ideally) make for a reproducible and easy to learn protocol which can be used across a variety of similar systems. Critical editions, as they exist in Classics, have a relatively standard form: a text is presented at the top of the page, while at the foot, there are notes keyed to the line or other numbered section and to the specific word or phrase affected. These notes present variant forms found in the manuscript tradition of the text, the conjectures of previous editors, conjectures by the editor which they are not confident enough to place in the main text, and other notes that elucidate how the editor established the main text. As a very small example, take R.A.B. Mynors' Oxford Classical Text of Vergil's *Eclogues*, poem 1, line 59. The text has:

Ante leves ergo pascentur in aethere cerui
(Sooner then will the agile stags graze in the air)

and the apparatus has:

59 pascuntur P aethere] aequore Ribbeck e recc.

We are given the line number in question, then the variant form *pascuntur* (the present tense instead of the future, which is what the text has). ‘P’ indicates the manuscript *Vaticanus Palatinus lat. 1631*, we are told in the preface. Why we would care that P has the present tense is left unclear – most likely because P and R (*Vaticanus Vat. lat. 3867*, which has *pascentur*) are the only witnesses of the older manuscripts that have this poem, and so it is worth noting when they disagree. The present tense is not impossible, but the future makes better sense, and it continues in the next line. It is obvious in the apparatus what word in the line *pascuntur* would replace, so a lemma is not provided. In the next noted variant, however, we are given one, marked off with a right square bracket. The word ‘aequore’ (sea) was printed by Ribbeck in his 1859 Teubner edition, based on later (*recentiores*) manuscripts. The following line is ‘and the waves will leave the fish naked on the shore’ (*freta destituent nudos in litore pisces*), so one can see the appeal of ‘Sooner will the agile stags feed in the sea’. Even in just this single line of text and single line of apparatus, there is a lot of information. Apart from the fact of the variations, it all points elsewhere: if you do not know what ‘P’ denotes, you have to go and look it up, and likewise ‘Ribbeck’ and *recc.* both require external knowledge: the one of the publication history of the text and the other of how the editor uses the Latin abbreviation *recc.* as a siglum. It is defined in the *Sigla Codicum* of the edition as *codices saec. nono recentiores*, ‘more recent manuscripts from the ninth century’ – P is 4th or 5th century and R, 5th century. This kind of compression is absolutely typical in the critical apparatuses of Classical texts. Capital Latin letters are used for major manuscripts, lowercase Latin letters for less important ones or for families of manuscripts. Greek letters are used for theoretical, lost manuscripts from which extant ones are descended. Sometimes lowercase Greek letters represent families. These symbols are defined by the editor in the preface.¹

The interface of the printed edition gives us the bare facts of textual variation with pointers to fuller details elsewhere. But it also obeys our list of functions: it presents the reader with variants stripped of the underlying details; once the reader understands the apparatus’ “code”, they can simply note the alternate possibilities in passing or dig deeper; the form follows a pattern repeated in most similar editions, and uses conventions and abbreviations the reader will be familiar with (the bracketed lemma, and the single letter *siglum* for example). Printed apparatuses do vary between minimalist and maximalist poles (Tarrant ch. 7), that is between presenting only the notes that the editor considers necessary for the establishment of the text and

¹ Karl Maurer has a nice summary of the kinds of symbols and abbreviations used in critical apparatuses at udallasclassics.org/maurer_files/APPARATUSABBREVIATIONS.pdf. Accessed 16 March 2017.

presenting a fuller view of the tradition (Mynors' apparatus is a minimalist one). But the difference is mainly one of quantity and type of notes rather than form. A maximalist approach might list several variant spellings of a word, for example. A minimalist might argue that while this might be interesting to a student of the manuscript traditions, it does not actually help one read the text. There is a tension in this debate over what is better for the reader: simplicity and ease of use, or better access to the textual tradition with extra complexity. Should the editor work hard to reduce the interpretive burden of the reader, or try to get out of the way between the reader and the history of the text?

Our brief examination of the form of a printed text and apparatus immediately raises the question of what can (or should) then be done in a digital context with this kind of text. Freed from the spatial constraints of print, is there any reason to compress the information therein and to divert readers wanting more information elsewhere? Is there a justification for a minimalist approach to apparatus construction, or should we prefer maximalism, or even "totalism"? Why not record every known variant, whether significant or not? Moreover, we nowadays may have access to images of the manuscripts and to digitized copies of older editions. Why not present them as well, or at least link to them? It quickly becomes clear from this line of thinking that a digital critical edition has the potential to become a complex web application in its own right. And then the question becomes "where do we start?" There is an almost infinite variety to web applications and their expressions online.

The framework for the investigation of digital critical editions discussed here comes from the *Digital Latin Library* project (DLL), a collaborative effort of the Society for Classical Studies, the Medieval Academy of America, and the Renaissance Society of America. The DLL Project has a two-fold mission:

1. To publish and curate critical editions of Latin texts, of all types, from all eras, and to facilitate an ongoing scholarly conversation about these texts through open collaboration and annotation.
2. To facilitate the finding and, where openly available and accessible online, the reading of all texts written in Latin.

The first of these goals will be met by the creation of a Library of Digital Latin Texts, a series of new, born-digital critical editions to be published under the auspices of the aforementioned learned societies.²

The answer to the question of where to start, or at least the answer that the *Digital Latin Library* project has chosen, brings us back around to the topic of interfaces and to the realization that we have to be very careful about what we choose to adopt, copy, and invent. The tension between simplicity and complexity resurfaces again and again. As a pilot edition, the project's Principal Investigator, Sam Huskey, chose

² See digitallatin.org/about-project/mission-and-goals. Accessed 16 March 2017.

T. CALPURNI SICULI

BUCOLICA

I.

[Corydon, Ornytus]

- C. Nondum Solis equos declinis mitigat aestas,
 quamvis et madidis incumbant praela racemis
 et spument rauco ferventia musta susurro.
 cernis ut ecce pater quas tradidit, Ornyte, vaccae
 molle sub hirsuta latus explicuere genista? 5
 nos quoque vicinis cur non succedimus umbris?
 torrida cur solo defendimus ora galero?
- O. Hoc potius, frater Corydon, nemus, antra petamus
 ista patris Fauni, graciles ubi pinea denset
 silva comas rapidoque caput levat obvia soli, 10
 bullantes ubi fagus aquas radice sub ipsa
 protegit et ramis errantibus implicat umbras.

I 1 C. G P A ϕ Ulit. Wernsd. Glaeser sqq., om. N $\pi\chi$ p, O. $\epsilon\beta\gamma\mu\rho$ edd. ante Glaeser. *nundum* G (corr. m¹) P. *declinis* N Heins. Schenkl, *declivis* G V edd., *declivus* Pp, *declives* u. 2 *quatinus* $\phi\pi\eta\theta$ r. *praeda* P. 3 om. $\kappa\chi$. *iniusta* P. 4 C. V plerique, edd. fere omnes ante Glaeser. *ornyte* Heins. Maehly Baehr. Schenkl, *ornite* N G V edd., *ornyce* P, *ornithe* s. *vaccae* $\epsilon\chi\lambda\beta\rho\alpha\gamma\gamma$ edd., *vacoe* N $\nu\phi$ $\pi\kappa\mu$ rp, *bacoe* G, *vaccae* δ . 5 *molle sub* P V edd., *molliter* N G A H Glaeser Keene Jacoby (Woch. f. kl. Phil. III p. 1290), *molle per* Schenkl². *hirsuta* P, *hirsutam* Schenkl². *explicare* $\epsilon\gamma\mu\lambda^1\delta\epsilon$. *genestra* $\lambda^1\delta$, *genesta* $\epsilon\gamma\mu\lambda^2\chi^2$ e Beck, *genistam* Schenkl². 6 *succedimus* G. *ulmis* nonnulli apud Titium. 8 O. om. N (add. m¹). *corydon* $\chi\epsilon\delta\rho^2$ edd., *coridon* N G $\beta\nu\kappa\lambda\gamma\mu\pi\rho^1\phi\alpha\eta\rho$ Barth., *corridon* P. *nemora* G (in mg.). *antra*-9 *ista* N G Glaeser sqq., *ista*-9 *antra* P V edd. ante Glaeser. 9 *gracilis* V (edd. vett. accusativos eiusmodi fere semper in *is* exhibent). *denset* G A P λ^2 Glaeser sqq., *densent* N, *densat* V edd. ante Glaeser. 10 *rabidoque* Baehr. (lect. lat. p. 35). *capud* G. 11 *bullentes* $\epsilon\delta\epsilon\delta^2$, *bullantis* $\eta\nu\phi\phi^2$ rs, *palantes* Heins. *ubi* ut G (et in mg.). 12 *protegis* G (corr. m¹). *errantes* G. 13 *vacae*.

Figure 1: Calpurnius' first eclogue.

Cesar Giarratano's 1910 edition of Calpurnius Siculus' *Bucolica*. A glance at the first page (fig. 1) is enough to show that Giarratano leaned well towards the maximalist end of the spectrum. Twelve lines of text merit sixteen lines of apparatus. This was good, because a more complex apparatus seemed likely to give whatever format we chose a better workout.

One of the first questions the project had to address was just how to format the data for its editions. Several options presented themselves. A relational database might serve well as a store for aligned collation tables of all the sources, for example. There are a number of scholars who have argued for using directed graphs where any version of the text can be constructed by following a particular path through the graph (Schmidt). These approaches both would tend to favor the maximalist approach, as well as more automation in the construction of the text. They should, in theory, scale up to many sources extremely well. To add a new source, the editor would transcribe it, tokenize it (turn it into a structure where each word is an atomic member), and align that tokenization with the main data structure, into which it could then be ingested.³ As Andrews argues, this kind of digital philology holds out a lot of promise as a means for investigating the textual tradition. What it does not necessarily do is point us at an interface for readers to interact with.

In a database or a variant graph, the qualitative aspects of the different witnesses – their worth as witnesses – will tend to be flattened. We can imagine a parallel system of metadata that provides weights or probability measures for variants, but such a system would entail a new and potentially formidable level of complexity. For the study of certain types of texts, this may not matter. If your object of study is how a text was variously presented, then not having, nor attempting to create an authoritative version is an advantage. But in Classics we are usually concerned with what the ancient author wrote, and less in how the text has changed over time except insofar as the patterns of change can hint at the original. This motivation necessarily involves privileging some witnesses over others.

Moreover, in the case of Classical texts, what we get from the textual tradition usually is not sufficient to understand what the ancient author wrote:

In almost all cases those writings have survived, if they have survived at all, only in copies many stages removed from the originals, copies of which not a single one is free from error. Often the errors are so great that it is no longer possible to tell what the author meant to say. (West 7–8)

In the case of Calpurnius, the best manuscripts we have date to the early 15th century. Calpurnius' dates are uncertain. He probably wrote during the reign of Nero (54–68 CE) but might have been as late as the 3rd century CE. Regardless of when he wrote

³ Many of these processes could be automated or crowd-sourced.

the original poems, however, the best evidence we have for his text comes from a time at least twice as far removed from the poet as from us! The best an editor of classical texts can do (in most cases) is to achieve a best-guess approximation of the original source. Conjecture is a necessary part of this process, as the sources may at times not provide any satisfactory readings at all. Because of the state of most classical texts, it is not enough just to produce a comprehensive snapshot of the textual tradition and hand that off to an unprepared reader.

On the other hand, having such data available to the reader of an edition would be a boon if they were interested in exploring the text's history. An interface supporting a multiplicity of varying sources might (in theory) enable the generation of an algorithmically "best" text, where each word displayed is the best-supported choice. It would also permit the reader to trace different threads of the manuscript tradition and see how those threads manifest in the text. It would offer the reader the chance to become more of an expert in that tradition, and to build their own versions of the text to suit their own needs.

We did not want to rule out this kind of approach to edition-making, but at the same time we felt that such work could not actually replace the traditional, editor-curated presentation of a text. Given that such an artifact was still a central requirement, we decided to focus on determining a suitable format for it and testing that format against Giarratano's complex apparatus. Because of the *DLL* project team's expertise with TEI, it was an obvious choice. Though it probably would not permit the representation of unlimited variance,⁴ we would be able to retro-convert existing, out-of-copyright editions (like Giarratano's 1910 edition) to digital form without having to start by breaking them up into their sources.⁵

A random sampling of Calpurnius' first eclogue (ll. 8–9) may provide us with a suitable example to work through:

| | |
|------------|---|
| O. | Hoc potius, frater Corydon, nemus, antra petamus ista patris Fauni, |
| O(rnytus): | Let's rather, brother Corydon, head for this wood those groves of father Faunus, |

⁴ Schmidt's (2009) dismissal of TEI as a suitable format because of the problem of representing structural variance was premature, however, as we will see.

⁵ Even with a maximalist edition, much information from the sources has been discarded. It is impossible to automatically derive a fair representation of, e.g., P from the information given in Mynors' apparatus, and it would be impossible even with a maximalist edition. That being the case, one could not extract the data necessary to perform the workflow described above.

The apparatus in the new edition for these lines has:

8 O.] om. N (a.c.) Corydon] $\chi\epsilon\delta\rho^2$ edd. *coridon* NGβνκλγμπρ¹φαηρ Barth
1613 *corridon* P nemus] *nemora* (in mg.) G¹ antra] NG Glaeser sqq. *ista* PV
edd. ante Glaeser
9 *ista*] NG Glaeser sqq. *antra* PV edd. ante Glaeser

So, first N omits the speaker label (but the omission was corrected by the copyist), then there are a variety of different spellings of Corydon's name, G has *nemora* (the plural form) instead of *nemus*, and P, V, and editions before Glaeser's swap *antra* and *ista*.

The DLL edition models this with the following TEI:

```
<l n="8"><app>
  <lem xml:id="lem1.7-label"><label type="speaker">0.</label></lem>
  <rdg wit="#N" xml:id="rdg1.7-omission" ana="#subtractive"/>
  <witDetail wit="#N" target="#rdg1.7-omission" corresp="#rdg1.7-label"
    type="correction-original"/>
</app>Hoc potius, frater <app>
  <lem wit="#γ #ε #δ #ρ2" source="#edd.">Corydon</lem>
  <rdg wit="#N #G #β #ν #κ #λ #γ #μ #π #ρ1 #φ #α #η #ρ"
    source="#Barth1613" ana="#orthographical">coridon</rdg>
  <rdg wit="#P" ana="#orthographical">corridon</rdg>
</app>, <app>
  <lem>nemus</lem>
  <rdg wit="#G" ana="#morphological"><add place="margin">nemora</add></rdg>
</app>, <app type="transposition">
  <lem wit="#N #G" xml:id="l8a1lem1" require="#l9a1lem1">antra</lem>
  <wit><ref target="#Glaeser">Glaeser</ref> sqq.</wit>
  <rdg wit="#P #V" xml:id="l8a1rdg1" require="#l9a1rdg1"
    ana="#ordinal" copyOf="#l9a1lem1"/>
  <wit>edd. ante <ref target="#Glaeser">Glaeser</ref></wit>
</app> petamus</l>
<l n="9"><app type="transposition">
  <lem wit="#N #G" xml:id="l9a1lem1" require="#l8a1lem1">ista</lem>
  <wit><ref target="#Glaeser">Glaeser</ref> sqq.</wit>
  <rdg wit="#P #V" xml:id="l9a1rdg1" require="#l8a1rdg1"
    ana="#ordinal" copyOf="#l8a1lem1"/>
  <wit>edd. ante <ref target="#Glaeser">Glaeser</ref></wit>
</app> patris <persName ref="#Faunus">Fauni</persName> ...</l>
```

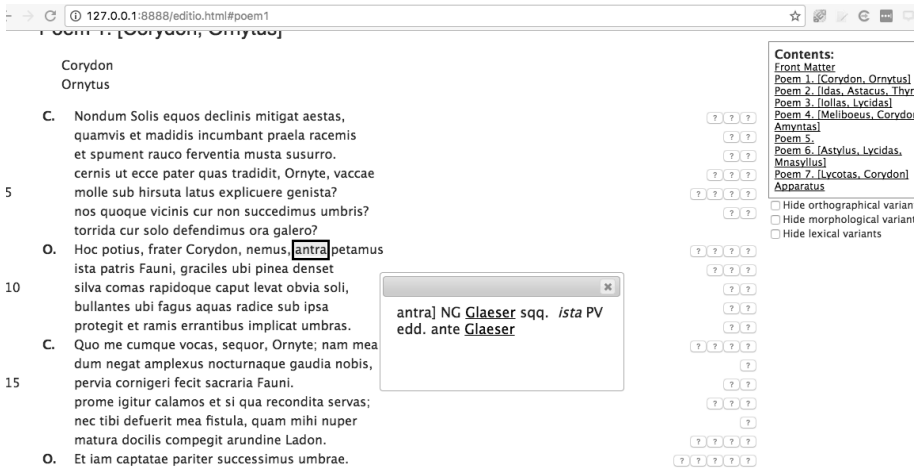
Again, there is a lot happening here. TEI's is actually a different data model than that of the printed edition: variations are placed inline rather than removed into notes, related features are linked rather than being represented in shorthand, and features that are implicit in the formatting of the print edition are made explicit and machine-actionable. TEI uses the app element to represent variation, so each entry in the printed apparatus will have a corresponding app. What is to be printed in the main text (if anything) goes in a child lem element, and variant readings go in

rdgs. So the speaker label that is missing in N goes in the lem, and its absence is represented by an empty rdg. The following witDetail signals that the omission was corrected. It is empty because this is a common feature of the text and the @type="correction-original" signals that when the apparatus is rendered, "(a.c.)" for *ante correctionem* will be printed (meaning the source had this reading before it was corrected). Conceptually then, this data structure places "O." and its absence at the same point in the text. Similarly, in the next entry, "*Corydon*" and the variant spellings "*coridon*" and "*corridon*" occupy the same notional place in the stream of text. In the next variant noted by Giarratano, G has *nemora* added in the margin, which DLL models using the add element. The final variant is the most interesting. NG have *antra...ista* and PV *ista...antra*, the same words, but inverted. This poses a problem of the sort that Schmidt and Colomb assert is more easily solved with a variant graph. But in fact, it poses very little difficulty for TEI, and even better, we can not only avoid the repetition of text with linking attributes,⁶ but we can use @require to note the interdependency between the two. We would not want to prefer the reading of NG and then PV in the next line, nor the reverse, as it would yield nonsense (*antra...antra* or *ista...ista*). If we prefer *ista* in line 8, then we must have *antra* in 9, and vice versa.

The TEI is obviously much more verbose than the compressed printed text and apparatus, but it goes well beyond the capabilities of print as well. Apparatus entries are categorized with @ana, meaning they can be filtered or searched on. Supporting manuscript witnesses and previous editions are linked to in @wit, @source, and wit elements, and other features like add and persName appear. Verbosity is often held up as a criticism of XML-based languages, but modern XML editors with smart autocomplete features make typing it not particularly onerous for an editor, and the fact that it can be parsed directly into a data structure for purposes of display and manipulation makes it extremely powerful. It is also important to remember that TEI's data model is not simply the tree structure that it gets from XML, but also the graph created by the use of its linking attributes (Cayless).

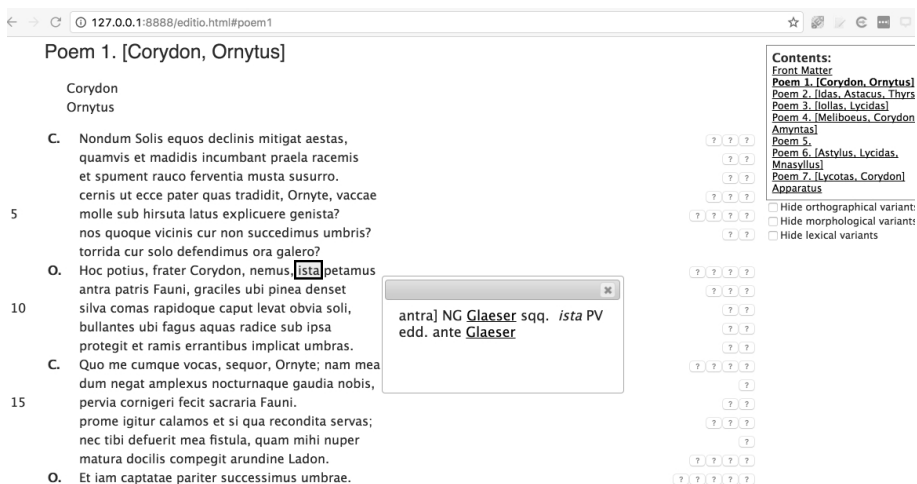
The majority of TEI-based web applications rely on transforming their TEI sources into HTML for display. Compared to TEI, HTML is a semantically impoverished language, so an XSLT transformation risks discarding information in the source. Indeed, the default behavior of an XSLT transform where there is no matching template for an element is to put the element's contents into the result, but discard the element itself, meaning that unless TEI elements are explicitly mapped to HTML, they will emerge from a transformation as plain text. Web browsers have long been happy to render pages with nonstandard elements, but the recently-published W3C *Custom Elements* Working Draft offers us a standard way to define our own

⁶ Schmidt and Colomb call @copyOf a "non-standard" attribute, but as a member of the set of global attributes, available on every TEI element, it could hardly be more standard.

Figure 2: *antra* selected.

elements and add custom appearance and behaviors to them. For our purposes, this means we can use slightly-modified TEI directly in the browser with CSS and JavaScript enhancements. We can leverage the TEI data model directly in the reading environment. The screenshots in figures 2 and 3 illustrate this process in action. Each variance in the text is flagged with a button in the margin. When clicked, the button opens a dialog box which allows the reader to select an alternate reading, which is then swapped into the text in place of the lemma. The reader is thus able to see directly how the text changes if they make a choice different from the editor's. In the first example, we see the default text, with the entry for *antra* in line 1.8 opened (fig. 2).

In the second, *ista* has been selected instead of *antra* (fig. 3). Because the lemmas and readings in lines 8 and 9 are linked with *@require*, when the reading in line 8 is selected, the corresponding reading in line 9 is also changed automatically. Swapping the two words does not much affect the sense of the lines, but it does allow the reader to directly experience the effect. The version Giarratano (and all editions starting with Glaeser's) prefers seems more poetic. The chiasmus, *Hoc ... nemus, antra ... ista*, is a typical poetic figure, and the placement of *antra* next to *nemus* both means the thought of line 8 is complete in itself and tends to intensify the common bucolic trope "Let's get out of the hot sun into the shade", leaving line 9 to expand on the thought. The rejected reading, *ista ... antra*, is a little simpler, and parallel to *Hoc ... nemus*. The ability to read these alternatives in place allows readers to experience both instead of having to mentally construct the version in the apparatus for themselves.

Figure 3: *ista* promoted to the text.

The web interface of the edition uses a JavaScript library called CETELcean,⁷ built by Raffaele Viglianti and myself, which takes a pointer to a TEI file on the web and loads it into the browser, making some changes along the way. The TEI namespace is removed, element names are prefixed with "tei-", as the Custom Elements specification demands, and the document is inserted into the body of the HTML page running CETELcean. Elements are either simply styled with CSS, or in the case of the TEI equivalents to HTML elements with special, predefined behaviors like links, the HTML is inserted alongside the TEI source, which is hidden.⁸ The result is HTML looking like the example below (the speaker label in line 8).

```
<tei-app data-teiname="app" id="id55">
  <tei-lem xml:id="lem1.7-label" id="lem1.7-label" data-teiname="lem">
    <tei-label type="speaker" data-teiname="label">O.</tei-label>
  </tei-lem>
  <tei-rdg wit="#N" xml:id="rdg1.7-omission" id="rdg1.7-omission"
    ana="#subtractive" data-teiname="rdg"></tei-rdg> <tei-witdetail wit="#N"
    target="#rdg1.7-omission" corresp="#rdg1.7-label"
    type="correction-original" data-teiname="witDetail">a.c.</tei-witdetail>
</tei-app>
```

A few additional points to note: elements with @xml:ids have them copied over as @xml:id and as plain @ids. Each element gets a new @data-teiname, which preserves

⁷ github.com/TEIC/CETELcean.

⁸ These include links (TEI ref and ptr) and tables, which have properties that cannot be controlled using CSS.

the name of the original element (crucially its case, which the browser's Document Object Model discards). Elements that will need to be addressed, like `app`, get assigned new `@ids` if they do not already have them. Like `@xml:ids`, `@xml:lang` attributes are copied into `HTML@lang`. All of the element's attributes are preserved. Besides relying on CETelcean, the DLL Viewer uses its own JavaScript to add the marginal apparatus controls, resolve any `@copyOf` references by writing the content of the target into the referring element, and to write out a traditional-style apparatus under the text. In some cases, we will wish to insert HTML content into a TEI element. TEI `ref` elements are one example, which function similarly to HTML hyperlinks. The easiest way to get the expected behavior from them is to insert an `` element inside the `tei-ref`. "Derivative" content like this is distinguished from original in two ways, depending on the browser's capabilities. If the browser supports Shadow DOM, which permits the insertion into an element of content to be displayed instead of the element's regular DOM content, then that is used. Otherwise, the original content (if any) is wrapped in an HTML `span` element with `display:none`; set so it will be invisible, and a copy of that content wrapped in an `` with the `@href` set to the `tei-ref`'s `@target`. Doing this keeps the original content available, in a consistent way, so that it can be serialized back to TEI XML.

Keeping the TEI data model around means that we can operate on it directly in the browser, and it turns out to be very useful. The functions that allow the swapping of apparatus lemmas and readings into the main text rely on using and manipulating the data model of the TEI text. If a reading is selected by clicking, the function simply converts the corresponding `tei-rdg` into a `tei-lem`, and the `tei-lem` into a `tei-rdg`. The page's CSS does the rest: `tei-rdg`'s present in the main text have `display:none`; set, and so are invisible and have no effect on the page's layout. Turning a `tei-lem` into a `tei-rdg` therefore makes it disappear, and the new `tei-lem` appears instead. Any lemmas or readings linked to the changed ones with `@require` or `@exclude` are automatically processed in turn, so that any dependencies are resolved. The DLL Viewer application thus roughly follows the model-view-controller model, where the model is the browser DOM, the view is the browser's view, governed by the application's CSS, and the controller is the DLL Viewer's JavaScript code.

Instead of requiring an XML transform via XSLT to HTML, the DLL Viewer entails the creation of a simple HTML web page, with links to CSS for displaying the transformed document content and a few lines of JavaScript to load the source document. Because it can be run simply from a web page, no server-side code is required at all. The DLL's project workflow for Calpurnius uses an HTML page in the Viewer's GitHub Pages site, which loads the source file from the GitHub repository in which it resides. Whenever changes are pushed to the project repository, those changes are automatically displayed in the corresponding Viewer page. Put simply, the technology enables a variety of collaborative, versioned editing workflows, which require very

little setup and which leverage existing systems to do their work. The workflow's simplicity means that the feedback loop between XML editing and review of the web presentation is very tight, allowing the editor to experiment with the encoding and to check whether it is compatible with the Viewer's interpretation.

This kind of feedback is very useful in helping clarify the decisions the editor has to make in encoding the text. For example,

```
<app>
  <lem>nocturnaue</lem>
  <rdg wit="#G" ana="#orthographical">noturnaue</rdg>
  <rdg wit="#N" ana="#additive"><add place="inRas">no<hi
    rend="superscript">c</hi>turna</add>que</rdg>
</app>
```

Here, the text has *nocturnaue*. The *Codex Gaddianus* (G) has *noturnaue*, and the *Codex Neapolitanus* (N) has an erasure before *-que* into which *nocturna* has been inserted. Giarratano has

noturnaue G, *no^cturnaque* N(*no^cturna* in ras.).

Here we have a case where the encoding can represent the state of the variant more closely than the printed version could. The representation of that variant in the Viewer will be something like

nocturnaue] noturnaue G «no^cturna»que N

But this highlights an issue with the encoding and the functionality in the Viewer. Giarratano adopted a policy of recording every variation in G, as he felt he was the first to do a proper job of collating it. But does that mean *noturnaue* is a viable reading for *nocturnaue*? Not on the face of it, since *noturna* is not a Latin word. It could be argued, however, that having all of the variants from G means the *DLL* edition could display the edition with all of G's changes applied. It is important to recognize that this would not be the same as viewing G, as the text contains readings from other sources too. We might wonder too, whether «no^cturna»que is really a variant at all. It is possibly interesting as signaling that a misspelling like that of G was corrected in N, but it would be silly to promote it to the text. Perhaps, then, it would be better to encode these variants using note rather than *rdg* so that they are visible in the apparatus, but not able to be substituted for the correct reading. A similar case occurs on line 13:

Quo me cumque vocas, sequor, Ornyte; nam mea Leuce,
(Where you call me, I follow, Ornytus; for my Leuce, ...)

In the apparatus, we get

leuce NGAPπεχ² edd., *leute* καχ¹, *lene* φηρνθrs, *lance* γμ, *luce* b, *lauce* λδβ

as variants of the name of Corydon's beloved, Leuce (Λεύκη, literally "White Poplar", the name of one of the daughters of Oceanus). It is not surprising that copyists who did not know Greek might mangle her name in various ways, but it is also not especially helpful to know how it was mangled if you're just trying to read the text, and it is quite clear that 'Leuce' is correct. Again, the editor and encoder has to choose whether these variations are significant enough that they could be promoted to the text, or whether they should be confined to notes.

Tarrant, as a minimalist, would almost certainly argue that none of these variants should be represented in the apparatus at all. But the TEI data model permits us a middle ground between the maximalist and minimalist poles. We can record variants that are interesting because they provide support for an editor's choice of reading for the text, or because they represent previously unpublished information, or make a point about the transmission of the text, but that should not be able to make it into the text itself. Such a distinction could be made by recording the less-textually-significant variants as notes, or by otherwise classifying the apparatus entries and readings. The *LDLT Guidelines* already recommend such classifications, by means of the @ana attribute. Lexical, morphological, and orthographic variants are thus marked, and can be filtered out in the Viewer. We can thus, to an extent, accommodate both a maximalist and minimalist view of the text simultaneously.

The *LDLT Guidelines* have been released and continue to be revised, and the Viewer code is likewise in development in a publicly hosted repository.⁹ The goal for the current phase of Viewer development is to create a JavaScript library which can be plugged into any one of a variety of content management systems or web frameworks. Some issues remain to be resolved: since the current editorial workflow relies on a web page fetching the TEI XML from another site (a GitHub repository in this case), search engine indexing is problematic. Google will in some cases execute JavaScript embedded in a web page that builds page content, and then index the generated content rather than just the plain HTML. But experiments thus far do not hold out much promise for this working with Viewer pages. The alternative is to pre-transform the TEI into HTML Custom Elements and use the Viewer just to style the page. This approach will produce an indexable document, and does not discard any of the source's data model, so it is likely that the final version of the DLL's Library of Digital Latin Texts will take this approach.

Probably the greatest disadvantage of the approach we have taken is one shared with other TEI projects: to produce an edition, one must learn TEI. The unwillingness

⁹ See github.com/DigitalLatin/viewer. Accessed 16 March 2017.

to learn a new technology is frequently raised as an objection to textual scholars doing work like this. There are various ways to deal with it. One being to say that, just as one must learn to type into a word processor, or to format a bibliography, or collate manuscripts in order to produce a critical edition, they must learn to encode their texts to produce a digital critical edition. Handing the work over to an expert encoder is another option. And we may imagine that once the data patterns inherent in these editions have stabilized, interfaces that make their construction a little easier can be constructed. So, both human and technological solutions are possible. At this early stage, when we are still learning what needs to be represented, and how best to model it, we are planning to rely on a hands-on approach.

In the solution presented here, the user interface is constructed directly from the data model used by the editor, essentially acting as a “skin” over it rather than as a separate artifact derived from the source. We can activate or suppress different aspects of the edition as needed. This presents a marked advantage over print editions in which a very full critical apparatus, not to mention other types of apparatus that may be present, can actually compromise readability by breaking up the text too much. Editions where each page consists of a handful of lines of text followed by many lines of notes are not pleasant to use when one’s goal is simply to read the text. Furthermore, we can allow the user directly to manipulate the data model, and to see the effects of their experimentation at once. Besides the user-driven apparatus outlined in this paper, we have the opportunity to produce Linked Data-ready documents, where entities that appear across many digital resources are registered and linked. Persons, places, organizations, events, manuscripts, and other editions are some of the axes across which we can link editions to other web resources. The possibilities are practically endless, but that brings us back to the need for intuitive and well-understood interfaces to help our readers to navigate the editions we will produce.

Bibliography

- Andrews, Tara L. “The Third Way.” *Variants: The Journal of the European Society for Textual Scholarship*, vol. 10, 2013, pp. 61–76.
- Calpurnius Siculus, Titus. *Calpurnii Et Nemesiani Bucolica*. Edited by Ceasar Giarratano. Naples, Detken & Rocholl, 1910.
- Cayless, Hugh A., “Rebooting TEI Pointers”, *Journal of the Text Encoding Initiative*, vol. 6, 2013, doi:10.4000/jtei.907. Accessed 16 March 2017.
- Denicola, Domenic, editor. *Custom Elements*, W3C Working Draft 13 October 2016, www.w3.org/TR/2016/WD-custom-elements-20161013/. Accessed 16 March 2017.
- Huskey, Samuel, et al. *The Digital Latin Library*. digitallatin.org. Accessed 16 March 2017.
- Huskey, Samuel and Hugh Cayless, *LDLT Guidelines: XML Patterns for Encoding Critical Editions*. digitallatin.github.io/guidelines/LDLT-Guidelines.html. Accessed 16 March 2017.

- Schmidt, Desmond, and Robert Colomb. "A Data Structure for Representing Multi-Version Texts Online." *International Journal of Human - Computer Studies*, vol. 67.6, 2009, pp. 497–514.
- Tarrant, R. J. *Texts, Editors, and Readers: Methods and Problems in Latin Textual Criticism*. Cambridge, Cambridge University Press, 2016.
- Virgil. *Opera; Recognovit Brevisque Adnotatione Critica Instruxit* R. A. B. Mynors. Oxonii (Oxford), E Typographeo Clarendoniano (Clarendon P.), 1969.
- West, M. L. *Textual Criticism and Editorial Technique Applicable to Greek and Latin Texts*. Stuttgart, B. G. Teubner, 1973.