Journal of the Text Encoding Initiative

Issue 10 | December 2016 - July 2019
Selected Papers from the 2015 TEI Conference

Why TEI Stand-off Markup Authoring Needs Simplification

Raffaele Viglianti

Electronic version
URL: http://journals.openedition.org/jtei/1838
DOI: 10.4000/jtei.1838
ISSN: 2162-5603

Publisher
TEI Consortium

Electronic reference
Raffaele Viglianti, « Why TEI Stand-off Markup Authoring Needs Simplification », *Journal of the Text Encoding Initiative* [Online], Issue 10 | December 2016 - July 2019, Online since 16 June 2019, connection on 04 July 2019. URL : http://journals.openedition.org/jtei/1838 ; DOI : 10.4000/jtei.1838

Why TEI Stand-off Markup Authoring Needs Simplification

Raffaele Viglianti

1. Introduction and Background

1 It is now commonly acknowledged that a text can be encoded according to multiple and competing modeling principles, despite the longstanding influence of the OHCO model (Ordered Hierarchy of Content Objects: see DeRose et al. 1990), which identifies nesting units of writing (such as chapters, sections, and paragraphs) as the representation of "what text really is." The TEI Guidelines generally follow this principle (with some important recent changes in stance), but provide mechanisms for encoding other textual features as secondary hierarchies, such as syntactic or rhetorical structures, or structures pertaining to physical text-bearing objects such as pages, openings (e.g., a pair of facing pages in a book), or other surfaces.

2 Secondary hierarchies can be encoded in TEI in a few ways, covered in Chapter 20 of the Guidelines "Non-hierarchical Structures." A commonplace solution is to use empty elements through the text to indicate the beginning of textual units ("milestones" in TEI jargon, such as `<pb>` for page breaks). When milestones are not sufficient, some inline elements can use pointing mechanisms to connect the starting and ending points of a textual feature. The elements `<addSpan>` and `<delSpan>`, for

example, can be used in conjunction with <anchor> to identify textual revisions across units of writing such as a paragraphs. Similarly to milestones, their placement indicates the starting point of the added or deleted text.

3 Finally, with stand-off markup one can encode secondary hierarchies by addressing arbitrary portions of an encoded text. To do this, a stand-off element links together one or more elements, or even identifies ranges within text nodes. The TEI Guidelines define a large number of stand-off elements that, arguably, have equivalent importance to elements forming the OHCO-inspired primary hierarchy. Hugh Cayless makes the helpful suggestion of regarding TEI documents as structures consisting of three equally important layers: the text (as in strings of characters), the tree (or the nesting text-containing elements), and the metastructures (forming a graph) that require stand-off markup and pointing mechanisms (Cayless 2013 ¶2).

4 Stand-off techniques are well documented in the TEI Guidelines, but they are not widely adopted, except in their simplest forms. Often TEI projects shy away from stand-off techniques because they introduce considerable managerial overhead: even with the simplest form of stand-off, the linking between elements happens through string identifiers, so encoders need to make sure that the references are correctly entered. Linguistic corpora in TEI are a notable exception because they are particularly dependent on stand-off techniques, as they enable linguistic analysis according to more than one principle (e.g., morphological, orthographic, syntactic; see Bański 2010). Often, these stand-off elements are automatically created by tokenization and part-of-speech identification algorithms. When the encoding is done by hand, which is by far the most common way of creating TEI for most projects, stand-off markup becomes a challenge, particularly in order to avoid misspellings and to track changes to identifiers.[1]

5 Typical TEI stand-off elements delimit a portion of text by using pointing mechanisms, with the aim of applying the semantics of the element over a span of text. These elements can be placed within a TEI document, but can also exist in separate TEI documents (what Bański calls "*remote* stand-off"; Bański 2010, sec. 4.1, emphasis added). To give three examples: (1) chapter 16 of the TEI Guidelines describes elements and attributes for temporally synchronizing transcriptions of spoken text, or placing textual elements on a timeline; (2) linguistic annotation, as mentioned earlier, chiefly relies on stand-off, supported by elements such as <interpr> defined in chapter 17;

finally, (3) the TEI provides a catchall stand-off structure, the "Feature Structures" (`<fs>`) defined in chapter 18, for annotating any kind of textual feature (often linguistic) that is not represented by other TEI means.[2]

6 Beyond these already established, fairly domain-specific TEI patterns, the need for stand-off markup is going to become gradually more pressing in a wider range of TEI projects, provides a technical answer to recent developments in the theory of textual modelling that have been strongly acknowledging the multifaceted nature of text. Patrick Sahle's theory of pluralistic text (Sahle 2013), notably claims that text can be viewed in different ways, for example as an idea, a work, a linguistic expression, as well as a sign and a material document.[3] Elena Pierazzo has proposed a multidimensional model of text, document, and work where each dimension is the function of a reader's selection of observable features and the context in which they are found (Pierazzo 2015, 40–54). Peter Robinson, as a reaction to a recent proliferation of digital editorial projects focusing on single sources, has remarked that editions should "illuminate both aspects of the text, both text-as-work and text-as-document" (Robinson 2013, 123). Invoking these positions, which one can hardly hope to summarize exhaustively in a few sentences, serves as a reminder that scholarly text encoding is determined by fundamental decisions about one's idea of text, and that competing views will emerge and require encoding. Indeed, the recent push for a multidimensional modeling of text is already generating projects that attempt to integrate competing representations of the same text.

7 The TEI has recently adopted a set of elements in direct competition with the OHCO-inspired representation of units of writing. The Guidelines chapter "Representation of Primary Sources" was substantially expanded in TEI P5 version 2.0.0 (in December 2011), following the proposal of a working group focused on genetic editing.[4] The resulting new elements and encoding strategies allow for a document-focused form of encoding, where concepts of page, opening, zone of writing, and lines are of foremost importance. These structures are useful to formalize an encoder's understanding of how text takes form on a writing surface, as well as tracking an author's actions on the page to identify textual revisions and the order in which they occurred. This has already sparked a debate on how to coordinate document-focused encoding with the more traditional (in TEI terms) text-focused encoding, often using text duplication or stand-off markup (see Brüning, Henzel, and Pravida 2013 and Muñoz and Viglianti 2015).[5]

8 The two case studies that follow will exemplify this tension between competing representations of text and how some balance was achieved with stand-off markup. The first case study describes the digital edition of the libretto of Carl Maria von Weber's opera *Der Freischütz* in the context of the now-completed *Freischütz Digital* project. The need for a stand-off critical apparatus of variants emerged because the encoding of each source included typographical and orthographical detail that made the use of embedded apparatus entries (also known as parallel segmentation) impossible. The second case is a report of work in progress at the Shelley-Godwin Archive (S-GA), an early adopter of the new document-focused encoding strategies for the representation of primary sources. In the S-GA, units of writing were initially encoded with milestones (empty elements between the units), but this course of action has proven to be unsatisfactory and too complex to maintain, and the project is currently experimenting with remote stand-off markup. Both projects described make use of *coreBuilder*, a web application originally designed for the *Freischütz Digital* project to facilitate the authoring of stand-off markup in a visual environment. The tool will be presented as an example (as opposed to an outright solution) of making the application of stand-off techniques more approachable in the context of projects dealing with multidimensional representations of text, without substantially disrupting workflows already familiar to TEI encoders.

2. The Libretto Edition for the *Freischütz Digital* Project[6]

9 *Freischütz Digital* (FreiDi) was a project funded by the Bundesministerium für Bildung und Forschung (Federal Ministry for Education and Research - BMBF) to create a digital critical edition of Carl Maria von Weber's romantic opera *Der Freischütz*.[7] The project set ambitious goals, including the digitization of all manuscript sources for both the score and the libretto, a new recording, and experimental audio-score alignment features. As a general principle for the digital edition, encoded diplomatic transcriptions were to be provided for all sources. For the musical ones, encoded using the Music Encoding Initiative (MEI) format, a diplomatic transcription meant dealing with source-specific aspects of: (1) textual components of music notation, such as performance instructions, dynamics, tempo marks, and lyrics; and (2) music notation symbols and conventions, including the use of brackets, tuplets, and articulation signs. The sources of the libretto, written by Friedrich Kind with some substantial interventions by Weber (see Weber

2007), were encoded in TEI "with a focus on the dramatic and lyrical structure of the text: the content is organized by scenes and verses; prose texts are distinguished and attributed to each actor; and stage directions and other descriptive text are identified as such. The transcriptions preserve original spelling and emphasis such as italicized and underlined text. Revised, deleted, and added passages are identified and marked up" (Viglianti 2016, 735). Since all sources were transcribed and encoded independently, creating a collation of variants required remote stand-off techniques.[8] *FreiDi*'s approach to sources implemented a pluralistic view of text: the digital model needed to account for both the genetic and codicological dimensions of source-specific details, as well as textual variance and, more generally, the work.

10 The "parallel segmentation" method for encoding variants described in the TEI Guidelines (chapter 12) could not be applied to the TEI-encoded libretto sources of *FreiDi* because, according to this method, textual variants are encoded directly in the text at phrase level. The TEI also offers a stand-off method for encoding variants, called "double-endpoint attachment," in which variants can be encoded away from the base text by specifying the starting and ending points of the lemma of which they are a variant. This allows encoders to refer to overlapping areas on the base text.

```
<l n="1">
 <seg xml:id="WBP-so.1.1">Experience</seg> though noon Auctoritee </l>
<!-- In another file -->
<app from="example.xml#WBP-so.1.1">
 <rdg wit="#La">Experiment</rdg>
 <rdg wit="#Ra2">Eryment</rdg>
</app>
```

(TEI Consortium 2019, chapter 12.2.4)

11 While more flexible, this method was not ideal for *FreiDi*'s model because it assumes the existence of a base text, in terms of which the editor records variants from other sources; this makes it impossible to preserve textual aspects of the various sources such as spelling, abbreviations, or

emphasis without identifying these aspects as substantial variants themselves. For example, if the character name *Agathe* were underlined in one source and not in another, one would have to create a new apparatus entry to record such a difference:

```
<speaker>
 <app>
  <rdg wit="#W1">Agathe</rdg>
  <rdg wit="#W2"><hi rend="underline">Agathe</hi></rdg>
 </app>
</speaker>
```

12 The solution for *FreiDi*'s libretto edition involved the creation of a separate apparatus file that encodes textual variance with `<rdg>` elements containing pointers to markup in the encoding of the sources. This approach has a lot in common with collation files generated after an alignment step in software such as Juxta and CollateX,[9] but it is designed to operate at more than one level of tokenization, so that statements about variation can be attached to any element in the TEI-encoded sources. To briefly illustrate this model, let us consider the following verses from two libretto sources and the corresponding apparatus file entry.[10]

Source KA-tx15.xml	Source A-pt.xml

`<l xml:id="KA-tx15_l1">Sie erquicke,</l>`
`<l xml:id="KA-tx15_l2">Und <w xml:id="KA-tx15_w1">bestricke</w></l>`
`<l xml:id="KA-tx15_l3">Und beglücke,</l>`
	`<l xml:id="A-pt_l1">Sie erquicke,</l>`
	`<l xml:id="A-pt_l2">und beglükke</l>`
	`<l xml:id="A-pt_l3">und <w xml:id="A-pt_w1">bestrikke.</w></l>`

Apparatus file

```
<app>
 <rdg wit="#KA-tx15">
  <ptr target="KA-tx15.xml#KA-tx15_l2"/>
  <ptr target="KA-tx15.xml#KA-tx15_l3"/>
 </rdg>
```

```
 <rdg wit="#A-pt">
  <ptr target="A-pt.xml#A-pt_l2"/>
  <ptr target="A-pt.xml#A-pt_l3"/>
 </rdg>
</app>
<app>
 <rdg wit="#KA-tx15">
  <ptr target="KA-tx15.xml#KA-tx15_w1"/>
 </rdg>
 <rdg wit="#A-pt">
  <ptr target="A-pt.xml#A-pt_w1"/>
 </rdg>
</app>
```

13 In this example, the apparatus file records two variants: the first identifies the inversion of verses by pointing to <l> elements, while the second identifies a variant in spelling. Differences in capitalization and punctuation, on the other hand, are not directly targeted or identified as variants. This solution has allowed *FreiDi* editors to model two competing representations: the one pertaining to variance and the one pertaining to the peculiar features of a specific source.

3. A New Possible Encoding Strategy for the Shelley-Godwin Archive

14 The Shelley-Godwin Archive (S-GA) began as a virtual reunification project to bring together high-quality digital images of manuscripts of Mary Wollstonecraft, William Godwin, Percy Bysshe Shelley, and Mary Wollstonecraft Shelley. The archive has now completed two phases of development (the latest in February 2016) and its focus has expanded to include semi-diplomatic transcriptions of a growing portion of the archive's holdings (Muñoz and Viglianti 2015). To encode these transcriptions, the S-GA adopted document-focused encoding strategies for the representation of primary sources introduced in TEI P5 2.0.0. As a result, S-GA TEI files model the *mise-en-page* of writing on the manuscripts and include "rigorous description of often complicated sets of additions, deletions, and emendations" (Muñoz and Viglianti 2015).

15 The editorial team thought it necessary to also represent, as much as possible, units of writing more typical of text-focused TEI encoding, both to provide users of the S-GA with a less noisy "reading view" as well as to facilitate interchange with other TEI documents.[11] As in the *FreiDi* example above, there is a tension between different and competing views of the text that need to be reconciled. In this case, too, stand-off offers a way forward.

16 Units of writing in S-GA are currently encoded as a secondary hierarchy using `<milestone>` and `<anchor>` elements that mark the beginning and the end of the unit. The `@unit` attribute is used to indicate the corresponding TEI element:

```
<milestone unit="tei:p" spanTo="#ap1"/>
<!-- document-focused encoding -->
<anchor xml:id="ap1"/>
```

17 This method worked well for relatively simple units of writing such as paragraphs, chapters, and headers, and S-GA was able to successfully deploy it for the publication of the *Frankenstein* manuscripts. While encoding Percy Shelley's *Prometheus Unbound* manuscripts, however, some limitations of this approach became apparent. Both poetic and dramatic structures can be encoded in Shelley's lyrical drama, which made the milestone-based solution too complex to create and maintain. In particular, while milestone `@unit` attributes invoke TEI element names, the secondary hierarchy that they establish cannot be validated as easily and effectively as an explicit hierarchy of text-focused TEI elements.

18 The solution that is currently being developed promotes the text-focused secondary hierarchy back into the <text> element. Instead of duplicating text content, however, stand-off references are used to include content from the document-focused encoding, which remains the primary form of encoding, as shown in the following example.

```
<sourceDoc>
 <!-- . . . -->
 <zone>
  <line xml:id="l1">
   <del rend="strikethrough">also</del>
   <metamark> [</metamark>
   <mod>
    <del rend="overwritten">w</del>
    <add place="intralinear">W</add>
   </mod> hen I was about <mod>
    <del rend="strikethrough">twelve</del>
    <add place="superlinear">fourteen</add>
   </mod> years </line>
  <line xml:id="l2"> old we were at our house near</line>
 </zone>
 <!-- . . . -->
</sourceDoc>
<text>

 <div type="chapter">
  <p>
   <ptr target="#l1" type="includeText"/>
   <ptr target="#l2" type="includeText"/>
   <!-- . . . -->
  </p>
 </div>

</text>
```

19 This approach will simplify the encoding of more complex textual structures without overcrowding the document-focused hierarchy; validation will also be simpler and more effective. Arguably, the document-focused encoding could similarly include textual content with stand-

off elements from a "plain" text document containing just strings of characters. This approach has been suggested in a number of publications (e.g., Schmidt and Colomb 2009), but we are deliberately avoiding it because we argue that plain text is not sufficiently devoid of interpretation or markup. Rather, we designate our document-focused encoding as the primary encoding that other representations can target with stand-off elements.

20 While being more expressive and potentially easier to use than the previous milestone-based model, creating and managing the pointers introduces complications typical of managing stand-off encoding.

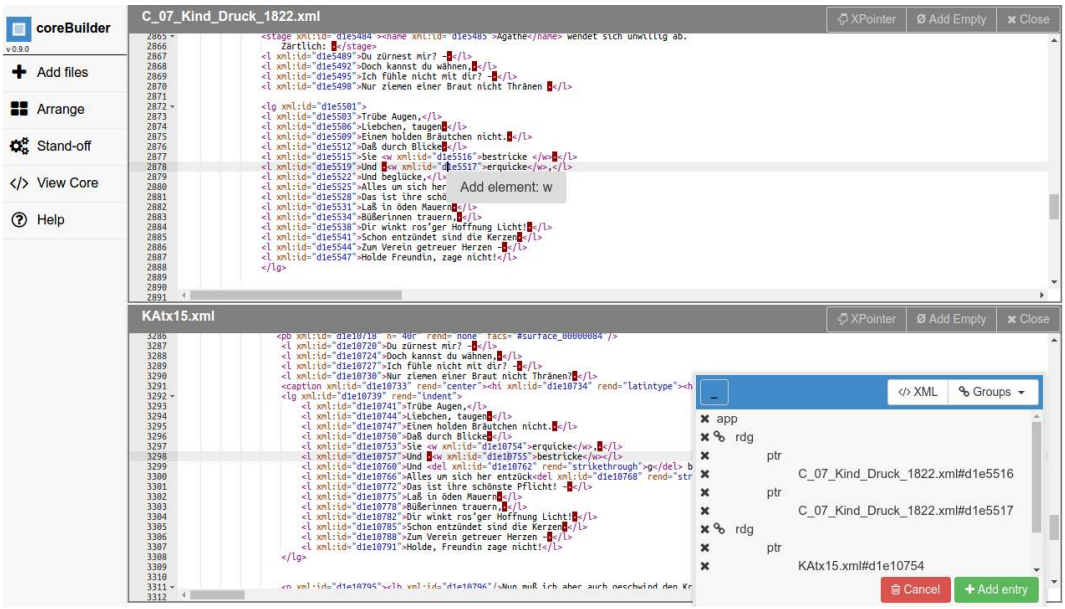
4. Facilitating Stand-off Markup Creation with *coreBuilder*

21 These two case studies show that stand-off techniques can be effective for encoding text according to a pluralistic, multidimensional modeling approach. They also highlight at least two challenges for the manual creation of stand-off markup: (1) entering pointers as string identifiers is error-prone; and (2) copy-pasting identifiers can be a slow process, particularly when there are multiple documents involved, exemplified in the case of the libretto edition in *FreiDi*. Good authoring tools may help produce solid stand-off markup, but tools that are customizable, non-domain-specific, or offering TEI support, are not readily available.[12]

22 The *coreBuilder* tool was developed to address this gap.[13] Originally developed for *FreiDi*'s libretto edition (Viglianti 2016, 738–39), the tool is a web application for creating stand-off markup in a visual environment. Encoders can open XML files from their local machine or from the web,[14] set the stand-off elements and attributes that the tool should create, and finally click on XML elements with identifiers (`@xml:id`) to create stand-off entries. The tool automatically creates the links using the identifiers, thus reducing human error. The visual interface is designed to eliminate the need

for scrolling between different parts of the same file, or moving between different windows while locating target elements and creating the stand-off elements. The stand-off entries are stored in an XML file (or "core") that can be downloaded and integrated into a TEI encoding.

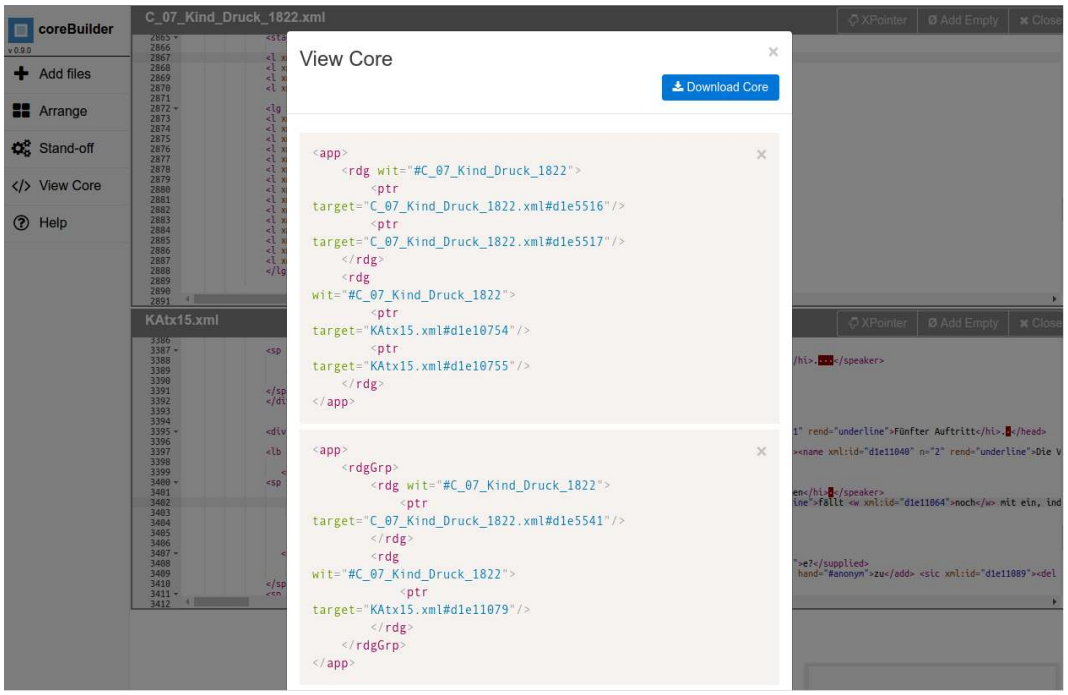
Figure 1. A screenshot of *coreBuilder* version 0.9.0.



23 Figure 1 shows the use of *coreBuilder* for creating a stand-off apparatus entry. There are two TEI files opened in the tool, each encoding a different source document of the same work. Users can create stand-off entries by clicking on elements with `@xml:id` attributes, or by selecting text to create XPointer expressions.[15] The apparatus entry created in the example identifies a variant

with two readings that invert the words “bestricke” and “erquicke” in two consecutive verses. The remote stand-off approach allows encoders to preserve differences in punctuation between the two sources without encroaching on the critical apparatus.

Figure 2. A list of stand-off entries ready to be downloaded as XML.



24 The earliest version of *coreBuilder* that was used for *FreiDi* relied exclusively on element identifiers: *FreiDi* encoders, therefore, were also tasked with introducing elements such as `<seg>` and `<w>` to allow the stand-off apparatus to refer to the text at the right point (see example described in section 2). The latest version of the tool also supports the creation of XPointer expressions by highlighting and selecting ranges of text directly on the XML document. These make it possible to refer to arbitrary ranges of text in a TEI document, as defined in chapter 16 of the Guidelines. The XPointer functionality is adapted from *tei-xpointer.js*,[16] a JavaScript library by Hugh Cayless, who also contributed to a general overhaul of the TEI XPointer scheme definition (see Cayless 2013).

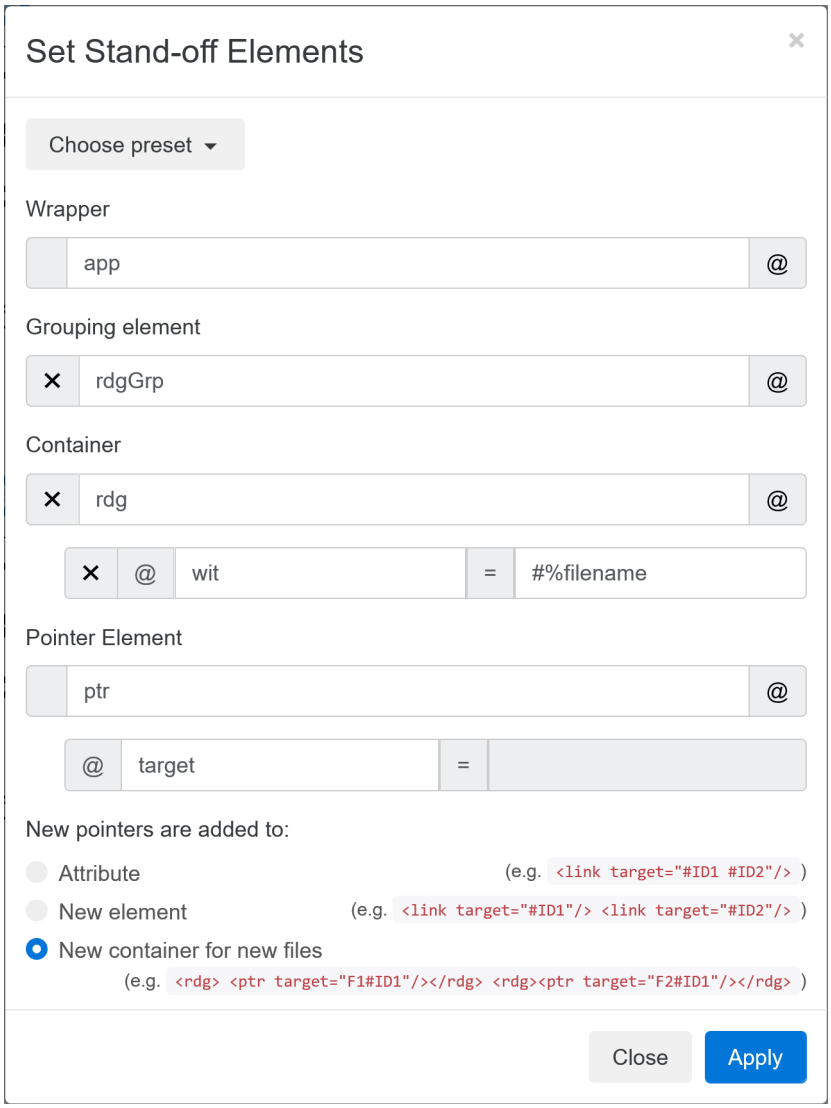
25 Using XPointer avoids cluttering TEI documents with unnecessary markup and provides a great level of flexibility and granularity. This approach has one important drawback, however: any small change in the target encoding will likely “break” the XPointer reference. Consider, for example, correcting the typo in the word “exmple” to “example” and consider also that this text is marked

up with a <w> element that is targeted by stand-off markup. Correcting the typo would not require an update of the stand-off entry because it targets the <w> element as a whole. If the target were expressed with an XPointer string range, however, the extra "a" would cause a shift in the character count and introduce an error in the stand-off element that could very well go unnoticed. Practically speaking, even though a tool such as the *coreBuilder* is able to simplify the creation of stand-off markup, the managerial overhead remains somewhat problematic, particularly when references involve XPointer expressions with string ranges that cross hierarchies.

26 The *coreBuilder* may not be a complete solution to all the issues related to stand-off markup, but hopefully it can serve as an example of an application for simplifying stand-off markup authoring.[17] The tool provides the following features that we argue are essential for a stand-off markup tool to be useful and effective in a workflow where TEI is created by hand.

- It provides a visual environment to create references to XML identifiers by pointing and clicking (or equivalent touch-enabled operations).
- Users are able to set the stand-off elements that they need to create (see figure 3).
- The XML is not hidden from the user; the tool strives to be useful to TEI encoders at all levels, including experts who may find working with hidden or rendered XML a complication rather than an aid.
- Users can create TEI XPointer expressions instead of ID references when required.

Figure 3. The dialog box where users can set up stand-off elements. The image shows the elements for building a stand-off critical apparatus, one of the preset combinations from which the user can choose.



27 It is also tempting to argue for a deep integration with the web and a browser-based implementation; however, browsers are not currently the most scalable environment for working with large, or a large number of, XML files. Moreover, desktop XML editors are the most common means of creating TEI documents. Is it reasonable to step out of a desktop application to a browser-based one, and then potentially go back to the desktop? Possibly. There certainly are advantages to working in the browser, including the potential for online collaboration and editing, or even user-

generated content production (such as crowdsourcing). Indeed, popular XML editor enterprises, such as Syncro Soft's Oxygen XML editor, are beginning to create browser-based counterparts of their tool suites.[18]

28 Whether browser-based, or part of a desktop application, the complexities surrounding the use of stand-off markup in a TEI project will not be solved by one simple tool; parsing and processing stand-off are also less straightforward than other operations in XML. The main obstacle to greater adoption, however, is likely to be authoring because, unlike other steps, it needs to be done by hand, which is still the expected way of authoring documents in many TEI projects. Solving the problem of creating stand-off markup may simplify and aid projects that already make use of stand-off markup for analysis and annotation, but more importantly it has the potential to lead to a new wave of experimentation in TEI encoding. Modeling texts according to a pluralistic, multidimensional ontology of text is already largely possible with TEI, particularly after the expansion of the "Representation of Primary Sources" chapter and the introduction of document-focused elements. The limitations on doing so at this point are more practical than conceptual.

BIBLIOGRAPHY

Bański, Piotr. 2010. "Why TEI Stand-off Annotation Doesn't Quite Work: And Why You Might Want to Use It Nevertheless." In *Proceedings of Balisage: The Markup Conference*, 2010, Balisage Series on Markup Technologies, vol. 5. https://doi.org/10.4242/BalisageVol5.Banski01.

Brüning, Gerrit, Katrin Henzel, and Dietmar Pravida. 2013. "Multiple Encoding in Genetic Editions: The Case of 'Faust.'" *Journal of the Text Encoding Initiative* 4. http://journals.openedition.org/jtei/697; doi:10.4000/jtei.697.

Cayless, Hugh A. 2013. "Rebooting TEI Pointers." *Journal of the Text Encoding Initiative* 6. http://journals.openedition.org/jtei/907; doi:10.4000/jtei.907.

DeRose, Steven J., David G. Durand, Elli Mylonas, and Allen H. Renear. 1990. "What Is Text, Really?" *Journal of Computing in Higher Education* 1(2): 3–26. doi:10.1007/BF02941632.

Muñoz, Trevor, and Raffaele Viglianti. 2015. "Texts and Documents: New Challenges for TEI Interchange and Lessons from the Shelley-Godwin Archive." *Journal of the Text Encoding Initiative* 8. http://journals.openedition.org/jtei/1270; doi:10.4000/jtei.1270.

Pierazzo, Elena. 2015. *Digital Scholarly Editing: Theories, Models and Methods*. Farnham, Surrey, UK: Ashgate.

Piez, Wendell. 2015. "TEI in LMNL: Implications for Modeling." *Journal of the Text Encoding Initiative* 8. http://journals.openedition.org/jtei/1337; doi:10.4000/jtei.1337.

Robinson, Peter. 2013. "Towards a Theory of Digital Editions." *Variants* 10:105–31.

Sahle, Patrick. 2013. *Digitale Editionsformen. Zum Umgang mit der Überlieferung unter den Bedingungen des Medienwandels. Teil 3: Textbegriffe und Recodierung.* Schriften des Instituts für Dokumentologie und Editorik, 9. Norderstedt: Books on Demand. https://kups.ub.uni-koeln.de/5353/1/DigEditionen_3.pdf.

Schmidt, Desmond, and Robert Colomb. 2009. "A Data Structure for Representing Multi-version Texts Online." *International Journal of Human-Computer Studies* 67(6): 497–514. doi:10.1016/j.ijhcs.2009.02.001.

TEI Consortium. 2019. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Version 3.5.0. Last updated January 29. N.p.: TEI Consortium. http://www.tei-c.org/Vault/P5/3.5.0/doc/tei-p5-doc/en/html/.

Viglianti, Raffaele. 2016. "Music and Words: Reconciling Libretto and Score Editions in the Digital Medium." In *Ei, dem alten Herrn zoll' ich Achtung gern': Festschrift für Joachim Veit zum 60. Geburtstag*, edited by Kristina Richts and Peter Stadler, 727–46. München: Allitera Verlag. doi:10.25366/2018.46.

Weber, Carl Maria von. 2007. *Der Freischütz: Romantische Oper in drei Aufzügen: Kritische Textbuch-Edition*. Text by Friedrich Kind; edited by Solveig Schreiter in collaboration with the Carl-Maria-von-Weber-Gesamtausgabe. Munich: Allitera Verlag.

NOTES

1 There are non-XML solutions to deal with multiple encoding hierarchies, such as LMNL (Layered Markup and Annotation Language), a data model that can be expressed with a syntax that, unlike XML, allows overlapping elements. Seemingly, stand-off would not be necessary in a TEI document expressed using LMNL: "to be unburdened of the chore of assigning and maintaining pointers or references … comes as a considerable relief: even in a plain text editor, one can simply *tag*, focusing on the tasks of tagging" (Piez 2015). The main obstacle to adopting this kind of approach is ultimately the lack of software support, as XML is supported by a large number of tools and languages. The relief of "simply tagging" offered by LMNL is also likely proportional to the number of hierarchies one wants or needs to represent: the proliferation of overlapping elements may burden the readability of the text enough to prompt encoders to move one or more hierarchies "away" from the text, in some stand-off form.

2 Some inline elements also use pointing mechanisms to connect the starting and ending points of a textual feature. The elements `<addSpan>` and `<delSpan>`, for example, can be used to identify textual revisions across units of writing such as a paragraph. Unlike stand-off markup, however, these elements cannot be extrapolated from the text flow because their placement indicates the starting and ending point of the encoded text.

3 Sahle organizes these different notions of text as spokes on a wheel, which effectively positions them as supporting components of a whole.

4 "Preliminary Updates for Genetic Editing in P5," October 19, 2011, https://tei-c.org/Vault/TC/tcw21.html.

5 The Guidelines of SGML versions of the TEI included elements for encoding volumes, pages, columns, and lines as concurring hierarchies (see TEI P3 Guidelines, part 6, chap. 31). This involved a technique known as CONCUR that was unfortunately poorly supported. These elements were removed in TEI P4 in favor of their milestone counterparts (e.g., `<pb>`, `<cb>`, `<lb>`). The new elements introduced in P5 2.0.0, on the other hand, are designed to encode a topographical representation of text—a representation of how it was laid out on a physical surface.

6 A more detailed overview of the encoding model described in this section is given in Viglianti 2016, where the need for the model is contextualized in more musicological terms by exploring the relationship between textual and musical sources and its impact on digital editions of operatic works.

7 Accessed April 26, 2019, http://freischuetz-digital.de.

8 The model for encoding variation across musical sources is described in "Das *FreiDi*-Datenmodell," Freischütz Digital, July 2014, accessed April 26, 2019, http://www.freischuetz-digital.de/datamodel.html.

9 See, for example, "Textual Variance" on the "TEI Wiki": http://wiki.tei-c.org/index.php/Textual_Variance#Aligner, last modified April 8, 2011.

10 This example is adapted from Viglianti (2016, 737–38).

11 See Muñoz and Viglianti 2015 for a detailed discussion, particularly §6, "The Need for Interchange Between Document-Focused and Text-Focused Models."

12 This was certainly the case when work began on the encoding of the libretto edition in *FreiDi*, which prompted the creation of the *coreBuilder* tool discussed in this section. See also Pierazzo 2015, 121, for a similar complaint.

13 Accessed April 26, 2019, http://raffazizzi.github.io/coreBuilder.

14 XML files can be loaded from the web only if they are in the same domain as the *coreBuilder*, or if the content providers allow cross-domain requests. This is because of JavaScript's same-origin policy.

15 For an overview of the functionalities offered by *coreBuilder* and how to use them, see the project wiki, last updated August 7, 2016, https://github.com/raffazizzi/coreBuilder/wiki.

16 Available at https://github.com/hcayless/tei-xpointer.js.

17 Suggestions for features and bug reports, as well as developer contributions, are welcome at the "Issues" section of the *coreBuilder* GitHub site, accessed April 26, 2019, https://github.com/raffazizzi/corebuilder/issues.

18 Syncro Soft, Oxygen XML Web Author, accessed April 26, 2019, https://www.oxygenxml.com/webauthor/.

ABSTRACT

Stand-off techniques are well documented in the TEI Guidelines and their use is encouraged. They are not widely adopted, however, except in their simplest forms, because they introduce considerable managerial overhead to a project's workflow. This article argues that stand-off markup is becoming increasingly relevant to a wider range of TEI projects, partly due to recent developments in the theory of textual modeling that strongly acknowledge the multidimensional nature of text. As a consequence, there is a need to considerably simplify the authoring, management, and processing of stand-off markup; but to what extent is this possible? Can tools make stand-off techniques more approachable without substantially disrupting workflows familiar to TEI encoders? This article focuses on the complexities of stand-off markup authoring through two examples, one from the now completed project *Freischütz Digital* and one from ongoing development at the Shelley-Godwin Archive. More specifically, these examples present situations in which stand-off has been essential for bridging competing editorial representations of text, an operation that is becoming fundamental in the multidimensional modeling of text. The *coreBuilder*, an open source web-based visual environment for

authoring stand-off markup, was created to aid the encoding workflow for these TEI projects. *coreBuilder* was awarded the first *Rahtz Prize for TEI Ingenuity* in 2017. The tool is presented here as an example of simplifying the creation of stand-off markup without hiding the XML from the expert encoder.

INDEX

Keywords: stand-off markup, textual modeling, multiple encoding, tools

AUTHOR

RAFFAELE VIGLIANTI

Raffaele Viglianti is a research programmer at the Maryland Institute for Technology in the Humanities (MITH) at the University of Maryland and a current member of the TEI Council. Before MITH he worked at the Department of Digital Humanities at King's College London, where he contributed to several digitization and text encoding projects while also completing a PhD in digital musicology. Raffaele's research revolves around digital editions and textual scholarship, with a focus on editions of music scores. His work also focuses on the shaping of music performance practice by the performance of a music score from a digital device.