Umanistica Digitale – ISSN: 2532-8816 – n. 10, 2021
D. Schmidt – *True interoperability for digital scholarly editions*
DOI: http://doi.org/10.6092/issn.2532-8816/12604

True interoperability for digital scholarly editions

Desmond Schmidt

Charles Harpur Critical Archive
desmond.schmidt@protonmail.com

Abstract

If the digital scholarly edition (DSE) is ever to replace the print scholarly edition it must be made truly interoperable so it can be easily secured, moved, published, aggregated, distributed and sold. Current DSEs are customised for particular projects, and must be maintained by their creators. Their contents are also not easily reusable by others. However, digital editions can be made truly interoperable by writing them directly for the Web rather than first in XML, then converting them to HTML. For this to work, some changes to the organisation of the software and data of a DSE are needed. Instead of dividing the software into two parts that run on the server and the client, all software can be moved to the client. In this way, a DSE can become portable, durable and directly usable in any web-browser. Instead of XML, the textual data can use a simplified form of HTML consisting of only two elements: <P> and <SPAN>, controlled and customised by a standard CSS stylesheet. The current practice of encoding alternatives such as variants can be replaced by versions and layers: versions are complete texts written by the author and layers are notional transcripts of local changes ordered chronologically. In this way textual data can express most of the information formerly specified by the complex TEI-XML Guidelines, and the rest via other technologies, and reorganise it in a form that allows easy comparison, editing, searching and textual analysis using standard software tools.

Se l'edizione accademica digitale (DSE) è destinata a sostituire l'edizione accademica cartacea, deve essere resa veramente interoperabile in modo che possa essere resa sicura, facilmente trasportabile, pubblicabile e commercializzabile. Le attuali DSE sono create per progetti limitati o particolari e devono essere gestite dai loro creatori e il loro contenuto non è facilmente riutilizzabile. Tuttavia, le edizioni digitali possono essere rese veramente interoperabili creandole direttamente per il Web anziché prima in XML e poi convertendole in HTML. Affinché ciò funzioni sono necessarie alcune modifiche all'organizzazione del software e dei dati di un DSE. Invece di dividere il software in due parti che vengono eseguite sul server e sul client, tutto il software può essere spostato sul client. In questo modo, una DSE può diventare portatile, durevole e utilizzabile direttamente in qualsiasi browser. Al posto dell'XML, i dati testuali possono utilizzare una forma semplificata di HTML composta da due soli elementi: <P> e <SPAN>, controllati e personalizzati da un foglio di stile CSS standard. L'attuale pratica di codificare alternative come le varianti può essere sostituita da versioni e livelli: le versioni sono testi completi scritti dall'autore o autrice e gli strati o livelli sono trascrizioni interpretative di

modifiche locali ordinate cronologicamente. In questo modo i dati testuali possono esprimere la maggior parte delle informazioni precedentemente specificate dalle complesse Linee Guida TEI-XML, e il resto tramite altre tecnologie, e riorganizzarle in una forma che consente un facile confronto, modifica, ricerca e analisi testuale utilizzando strumenti software standard.

Introduction

Interoperability is usually defined as 'the ability of computer systems or software to exchange and make use of information' ([34]). When applied to the digital scholarly edition (DSE), interoperability is most useful when it allows digital editions produced by one research group to be read and fully utilised without modification in several applications by others ([47]). To avoid confusion I will call this 'true interoperability', or 'interoperability' for short.

Since software consists of functions that manipulate data, for example, that change its presentation, or which allow access to data through searching, comparing, editing, annotation, etc. print scholarly editions can also be regarded in a non-digital sense to be broadly interoperable. A printed book may be static but it does not require custom software to operate. All it needs are human eyes and fingers: 'we are born with the hardware to use books and we learn the software' ([20]). The user of a printed book can read text and glean information from its layout, for example bold text in a margin might indicate the speaker of a speech, or italics might indicate a stage direction in a play. An index can help the reader quickly locate passages for comparison or research. An apparatus criticus can provide a way to compare versions or to describe the state of the text in the source documents. Notes at the foot of the page can elucidate cryptic passages or provide information about people or places. The margins can serve as a medium for recording (for the owner of a book) personal observations about the text. In short, a print scholarly edition contains many built-in functions that work well without any need for software maintenance, security, or the expense of running a server, and will continue to work correctly for perhaps hundreds of years.

The point here is not that digital scholarly editions have no advantages over printed ones – of course they have many. The point is rather that humanists have come to expect interoperability in their use of any scholarly edition: that it will work for its user-readers not just for its creators, and will continue to do so. Although any given DSE may work fine right now, and be readily accessible to people all over the world, what happens when its creators can no longer afford to maintain it? For example, if the grant used to produce it runs out? For it to survive in future someone else must eventually take over responsibility for it, such as a publisher, or another research group, who will likely use different types of computers, different software, or employ researchers with different technical skills and ideas. By the time it comes to that, technology will also likely have moved on, so that the archivability and reusability of its data and software will become major problems. The interoperability of the software and data of DSEs also implies durability, since the multiple programs and billions of documents that support true interoperability will resist change simply by the force of their own momentum. It should therefore be clear that interoperability is a crucial requirement for digital scholarly editions, if they are ever to replace print editions.

The solution described below proposes general software (rather than specific applications) and techniques for creating digital scholarly editions independently of the desired approach of the scholarly editor. Any of the various formal orientations of the editor outlined by Shillingsburg ([54]: Ch.2) for producing critical editions of works or versions using eclectic methods, or documentary or genetic editions, or editorial strategies arising from Eggert's sliding model of the archival vs editorial impulse ([19]) can use the same basic approach described here.

History of the interoperability problem

Digital scholarly editions have been produced since the 1970s, and the software for creating them, such as collation programs, goes back to the early 1960s ([13]). After an initial period of experimentation in which idiosyncratic encoding systems were developed for individual projects, the need for interoperability of textual transcriptions and the software for processing them was eventually recognised. The Text Encoding Initiative (TEI) proposal for funding ([26]) envisaged that establishing a common encoding scheme for digital texts would facilitate interoperability roughly in the sense defined above:

> ... the materials created by projects over the next decade could serve as input to as-yet undeveloped software designed for any number of text analytic tasks. If both the creators of textual scholarly materials and software developers utilize a common encoding format, the texts may be used with any software package.

Unfortunately, this admirable goal proved hard to achieve in practice. Although this commitment was postponed in versions 3 and 4 of the TEI Guidelines as 'future work' ([55]; [57]: 1.3), it was eventually abandoned in version 5, the authors noting instead that: 'no predefined encoding scheme can possibly serve all research purposes' ([56], iv).

Bauman ([2]), who had worked on version 5 of the TEI Guidelines, tried to move the goal back to something more achievable, which he called 'interchange', defined as the transfer of texts between research groups through previous negotiation on a common format, or through changes to the encoding. He suggested that true interoperability, or 'blind interchange', that is, the free exchange of texts and tools for processing, was probably impossible to achieve in practice due to variations in the way similar features were encoded by different research groups.

Cummings ([9]) also implies that interoperable DSEs are an unrealistic goal:

> ... interchange is not and should not be confused with true interoperability. I would argue that being able to seamlessly integrate highly complex and changing digital structures from a variety of heterogeneous sources through interoperable methods without either significant conditions or intermediary agents is a deluded fantasy.

A similar conclusion was also reached in the German TextGrid project, which ran from 2006 to 2015. In the project's final report Aschenbrenner ([1]) complains that texts were encoded in idiosyncratic and conflicting ways, reflecting the fluidity of humanistic research. Even when

projects adopted common standards for data, metadata and interfaces, interoperability was not guaranteed. As a result, TextGrid enforced no interoperability between different projects, although it did encourage it through the use of standards and tools. He remarked that there appeared to be a conflict between flexibility in data modelling versus interoperability between projects.

Beshero-Bondar and Viglianti ([4]) suggest that true interoperability of editions is not necessary: 'We do not mean that designing for interoperability is never worth the trouble, but rather that all-or-nothing arguments about it are not especially helpful to the interests of textual scholarship.' They go on to argue that negotiated interchange is sufficient since it uses an agreed vocabulary of tags that can be understood and reused as required.

Pierazzo ([39]) sees the lack of interoperability as a vicious circle: the customisation of source transcriptions leads to the development of customised software. This in turn leads to a lack of generally available tools of sufficient power, which could be reused. Hence each new editing project is compelled to develop its own textual encoding model and customised software for it.

But that hasn't stopped people wanting truly interoperable DSEs. Brown ([6]), for example, writes:

> ... it certainly is the case that project silos are in large part responsible for what Rebecca Frost Davis and Quinn Dombrowski characterize as the 'Multivarious Isolation' that is 'hobbling' digital humanities and preventing mainstream scholars from embracing digital resource use and production ... So, it seems crucial for digital editing infrastructures aimed at reader-oriented editions ..., if not at enabling text mining or visualization across texts or collections, to promote greater interoperability.

Robinson also expressed some years ago ([43]) the need for 'fluid, co-operative and distributed editions'. These would be maintained by a community of scholars and readers working together, and freely exchanged. His later Textual Communities project sought to tackle the interoperability problem this raised by focusing on the creation of DSEs on a single server, to which collaborators from around the world would contribute. The distribution aspect relied on open-source licensing of all the transcriptions, and on the interchangeability of the TEI format. But the project ended with only a partial realisation of its goals ([44]).

There have been some large projects that tried to tackle the interoperability problem. The Interedition project, which ran from 2010-2012 had as its objective 'to encourage the creators of tools for textual scholarship to make their functionality available to others.' ([27]). Similarly, the TextGrid project developed a virtual research environment (VRE) to provide open source tools especially for the development of DSEs that would promote collaboration and open access ([33]). Although both projects set out to tackle aspects of the interoperability problem, and in spite of the fact that each developed useful tools, especially CollateX and the TextGrid VRE, neither project solved the interoperability problem.

And so, years after these initiatives and calls for action, there is still no close agreement on ways to create and share DSEs. The point here is not so much the failure to recognise that such interoperability is desirable, but rather that there is general scepticism that it is even possible.

The problem with accepting that digital editions can only be shared via negotiated interchange is that this stifles the development of sophisticated shared tools that work out of the box, or which do much more than format simple texts with an index. 'Sophisticated' refers to those tools found consistently in the most advanced DSEs such as 1) text and facsimile together 2) interactive timeline 3) side-by side comparison of versions 4) variant table 5) full text search 6) paginating manuscript viewer and 7) external annotations ([49]). As Cummings ([10]: i71) admits: 'Most tools that projects create are for their own bespoke purposes rather than generalized tools for any TEI document.' If most features of the modern DSE are bespoke then that means starting more or less from scratch with each new digital edition.

A consequence of this bespoke nature of the DSE is that editions themselves can't be easily shared, and must be maintained and secured on a server at the expense of their creators. It would be far better if, on completion, editions could be handed over to publishers for future maintenance, aggregation, distribution and sale. But this is unlikely to happen on any significant scale unless true interoperability can be achieved.

A possible solution

The question then becomes, what can be done about it? In fact, the solution to this whole problem may be hiding in plain sight. If true interoperability means using the same transcriptions and software in multiple programs without modification, and if true interoperability for the DSE is indeed possible, then those very programs must already exist. There is no need to make them. Nor is there any need to invent complex new data formats and try (often unsuccessfully) to persuade other researchers to use them exactly as was specified. The formats would be those already used by existing interoperable software. And there is no need to look far to find software and data with these very characteristics. The answer is simply the Web.

Of course, editions designed for the Web will need to be customised, as every website is already. But the underlying functionality and data of the Web is based mostly on three technologies: HTML, CSS and Javascript, which have all been around for decades. HTML is used to structure text. CSS is used to present it in a formatted way, and Javascript provides interactivity that can make editions come to life. That is all that is needed. These three technologies have now reached a point of stability that will be hard to subvert. Around 55 billion web-pages all over the world use them (WorldWideWebSize.com 2021). Changing them would cause chaos. And multiple browsers can read and understand them in an increasingly consistent way. The past 30 years have seen a gradual development of these technologies through backwards compatibility, so avoiding changes to existing data and software.

HTML/CSS/Javascript in the DSE

It is worth looking at these three technologies in more detail, since they form the basis of the proposed solution. Even though they are well-known, the way in which they can be leveraged to enable truly interoperable DSEs requires explanation.

HTML

HTML was first formulated by Tim Berners Lee at CERN in 1992 ([3]), and from that moment its popularity exploded. The current specification of HTML5 contains 106 tags ([60]). But for a digital scholarly edition only two are really essential: <P> and <SPAN>. The P-tag provides paragraph-level structure, and the SPAN-tag formats runs of characters. P-tags cannot nest, although SPANs can. Like XML tags, HTML tags can also take attributes. The only attribute needed for our Ps and SPANs is the class attribute. This will be used to specify the *type* of the paragraph or span.

One of the main reasons why TEI-XML is not interoperable is because the names for elements and attributes have to be agreed by researchers – hence the need for the TEI itself. However, as explained above, complete standardisation of those names is impossible to achieve in practice. In HTML, the names for tags and their attributes are already fixed by international convention, so interoperation on their basis is automatic.

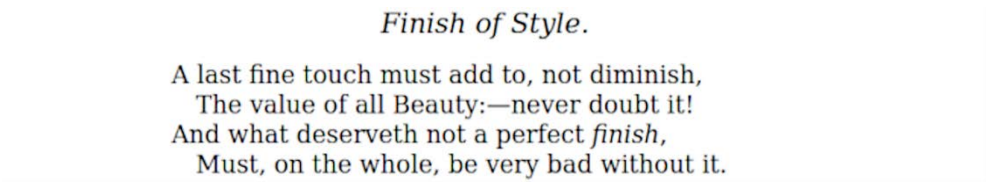
The HTML specification requires at least a <!DOCTYPE>, <HTML>, <HEAD> and a <BODY> tag to give the page structure. In practice, all of these can be dispensed with. All major browsers will load a page containing only P- and SPAN-tags successfully. Also, when serving a page containing only P and SPAN a website could be programmed to add the required extra tags as needed. So there is no need to store DOCTYPE, HTML, HEAD and BODY tags in the transcriptions of a DSE.

The <DIV> tag (division) is often used to provide additional structure for HTML, particularly to lay out the text in blocks. However, if the text gets packaged up into DIVs for presentation this fact need not be included in the transcriptions either. It may be objected that DIVs, which (unlike paragraphs) may nest, are needed to represent structured documents like plays. Although plays are composed of acts, scenes, speeches and lines, which nest at least conceptually, they don't have to be represented in that way explicitly. A much flatter representation consisting of a succession of paragraphs can do the job just as well and users cannot tell the difference. They don't see the DIV-tags. All they see is the vertical white-space that may be attached to them. This white-space can instead be attached to different types of P-tags.

Another objection might be that there are many different types of paragraphs. Some are headings, some are indented, some have significant line-endings and some do not. Also there are many different types of formats for runs of characters. There is italics for emphasis or foreign phrases, for example. All of this is formatting, which can be added to the paragraphs and character spans using the class-attribute. What is, for example, the conceptual difference between the TEI XML ([56]) tag:

<hi rend="italics">
and its HTML equivalent:
?
or between the TEI-XML tag:
<sp>
and the HTML
<p class="speech">?

Poetry can be precisely rendered by encoding each line as a with a particular class-attribute such as , and then enclosed in a <p class="stanza"> element formatted to honour line-breaks. In the Charles Harpur Critical Archive ([18]) and in the Giacomo Leopardi *Idilli* website ([29]), poems of various lengths and in different styles are represented this way. Figure 1 shows an example, followed by its underlying HTML markup:


Finish of Style.

A last fine touch must add to, not diminish,
The value of all Beauty:—never doubt it!
And what deserveth not a perfect *finish*,
Must, on the whole, be very bad without it.

Figure 1: HTML poetry example.

```
<p class="title"><span class="italics">Finish of Style.</span></p>
<p class="stanza"><span class="line">A last fine touch must add to, not
diminish,</span>
<span class="line-indent1">The value of all Beauty:—never doubt it!</span>
<span class="line">And what deserveth not a perfect <span
class="italics">finish</span>,</span>
<span class="line-indent1">Must, on the whole, be very bad without
it.</span></p>
```

The proof that HTML is adequate to encode *any* DSE is that all DSEs on the Web already use HTML for presentation, either directly or indirectly. So why not use HTML in our transcriptions too?

CSS (Cascading stylesheets)

The CSS standard was first published in 1996 ([30]). Adoption was initially slow, and it was not until 2000 that browsers began to support it fully. Since then both the number and standardisation of CSS features across browsers have increased steadily. CSS separates formatting specifications from the HTML tags and expresses these formats externally via a stylesheet. This is convenient for making a DSE because the stylesheet can then be used to customise the DSE by attaching formats to different types of paragraphs or spans distinguished by the class-attribute.

However, DSEs using XML have to undergo additional steps before their transcriptions can be displayed. First, the XML is (usually) transformed into HTML, then the HTML is rendered using a CSS stylesheet. The HTML transformation is usually performed first on the server via custom software, as shown in Figure 2. It is possible to transform XML in the browser using tools like TEI Boilerplate (Walsh, Simpson, and Moaddeli n.d.), but this method is not as reliable or as flexible as converting it to HTML first on the server. In contrast, storing the transcription directly in HTML omits this step and hence simplifies rendering.

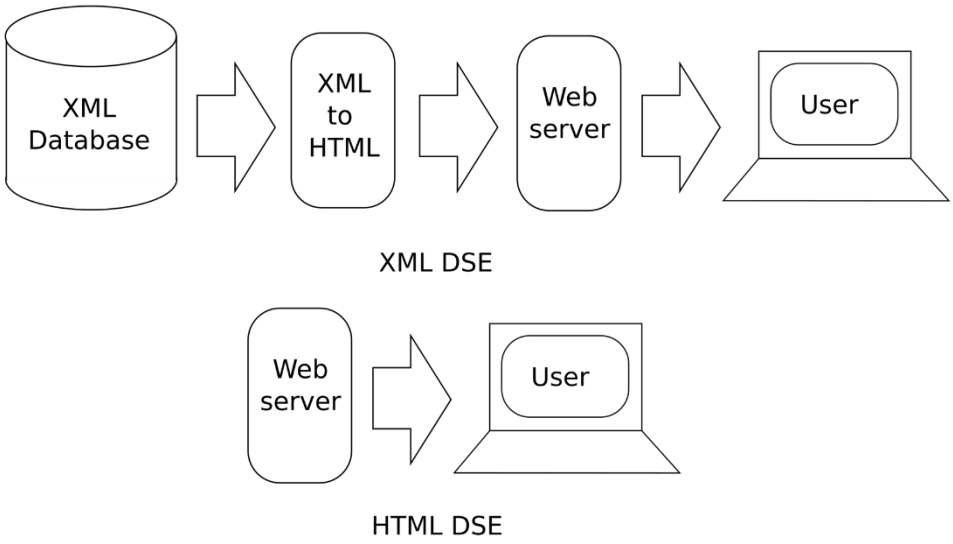


Figure 2: XML vs HTML edition.

Javascript

A DSE is not just static data. It can have many functions. For example, searching, comparison, interaction with the user, animation, navigation, synchronised scrolling, annotation, etc. Javascript allows the modern DSE to implement the functions of the print scholarly edition enumerated above in an interactive digital edition.

Javascript is a computer language ([17]) that can provide *all* of this functionality. It was originally developed at Netscape in 1995, and was first released as part of their Navigator browser. In 2008 Google developed a faster version of Javascript called V8 for its Chrome browser, which greatly increased Javascript's popularity and ensured its standardisation across the Web ([24]). A server version of javascript called Node.js, also using Google's V8 Javascript, was first released by Joyent Inc in 2009 ([11]). This allowed Javascript to be used for both server-side and client-side functionality. Javascript is now the most popular programming language in the world ([35]).

These developments make it possible to build a truly interoperable DSE. Consider all the functions performed in a modern DSE. Some of them take place in the client-side web browser and some on the server. The server-side functions are either 1) simple, such as fetching an image or a version of a document, or 2) complex, such as computing the differences between two versions or searching for something. To make a DSE that is fully interoperable all of these functions must be moved from the server to the client. Once there, every format, structure and function of the DSE will be processed directly in the browser. And as explained above, HTML, CSS and Javascript run equally well without modification in a wide range of browsers – hence this type of DSE will be truly interoperable.

Moving the simple server-side functions to the client is easy. The data they retrieve can simply be placed inside the web-folder and the web-server will return them on request. The usual

database can thus be replaced by a 'filebase' – a folder of files stored directly on the web-server. In testing this proved to be ten times faster than the best databases (Figure 3).

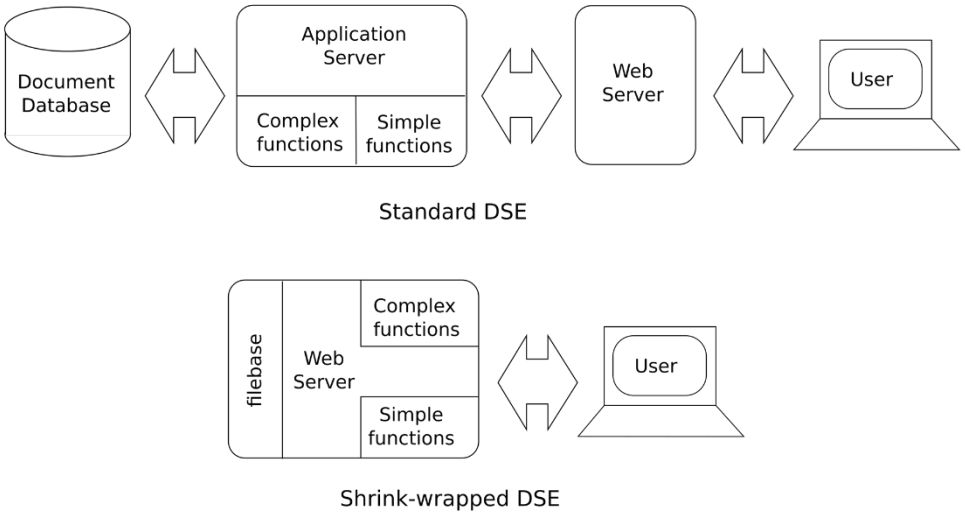


Figure 3: Standard vs shrink-wrapped DSE.

The complex functions like comparison and searching are harder to run in the client, but it can now be done since the server-side functions can be re-written in Javascript, and moved to the web-server. Then they will run on the client, since the user's browser loads and runs any Javascript it finds there.

Using this new design it becomes possible to produce a self-contained, fully featured and interoperable DSE, which only needs an ordinary web-server on which to run, and which can be read and interacted with through any web-browser. Such a DSE consists of a single folder of files that can be archived or copied from one machine to another easily. Since the only technology it needs to run is standard HTML, CSS and Javascript, no software beyond a browser and an ordinary web-server needs to be installed. Most people have these already.

A DSE created in this way has the additional advantage that it will likely continue working for years without maintenance, since the web-server software will be automatically updated as part of the operating system. It will also be secure from attack, since in its published form it is read-only.

I call this a 'shrink-wrapped DSE'. This is similar to the way EVT editions are published ([45]). The new Charles Harpur Critical Archive is a fully-featured shrink-wrapped DSE. It will be released as soon as the final functions are moved to the client. At the time of writing the current version still uses the standard server design.

An analysis of the TEI-XML format

Most DSEs are currently made in TEI-XML and then converted into HTML for viewing. However, this leads to the interoperability problem described above. Writing them instead directly in HTML solves this problem, but people will argue that TEI-XML contains more information than can be represented in HTML, or that XML can contain a superset of information that can be transformed into more than one type of output. However, this advantage is largely theoretical, as there is rarely more than one output ever generated from the XML files in a DSE. This argument has more to do with the XML way of doing things than any practical advantage. HTML can also be transformed into other outputs, such as RTF for print publication.

The next two sections explain how all the information in the TEI Guidelines pertaining to digital scholarly editions can in practice be transferred either to HTML or to other technologies that complement it.

Pierazzo ([38]) in fact raised this possibility a few years ago at the 2015 TEI Conference. As several people have also noted, XML is now threatened by obsolescence ([10]; [7]). Usage of XML both in web-services and as a document format have fallen dramatically in recent years ([53]). Converting DSEs currently stored in XML to HTML will avert this and ensure the longevity of new DSEs into the foreseeable future.

Analysis of the 589 tags in the TEI Guidelines P5 version 4.2.1 shows that only 35% are usable within a DSE. Figure 4 gives an overview of the various classes of markup tags in the TEI schema ([52]).

Several attempts have been made over the years to reduce the number of tags in the TEI schema to improve interoperability, such as TEI Lite ([8]), TEI Tite ([59]) and TEI Simple ([32]), but they have not addressed the underlying problem of semantic variation: the idiosyncratic ways in which tags are understood, selected and used by transcribers ([15]). This is how non-interoperability of manually encoded transcriptions arose in the first place, as Renear ([42]) saw. These reductions of the TEI schema omit most of the variant tags, and are thus usually inadequate for making digital scholarly editions.

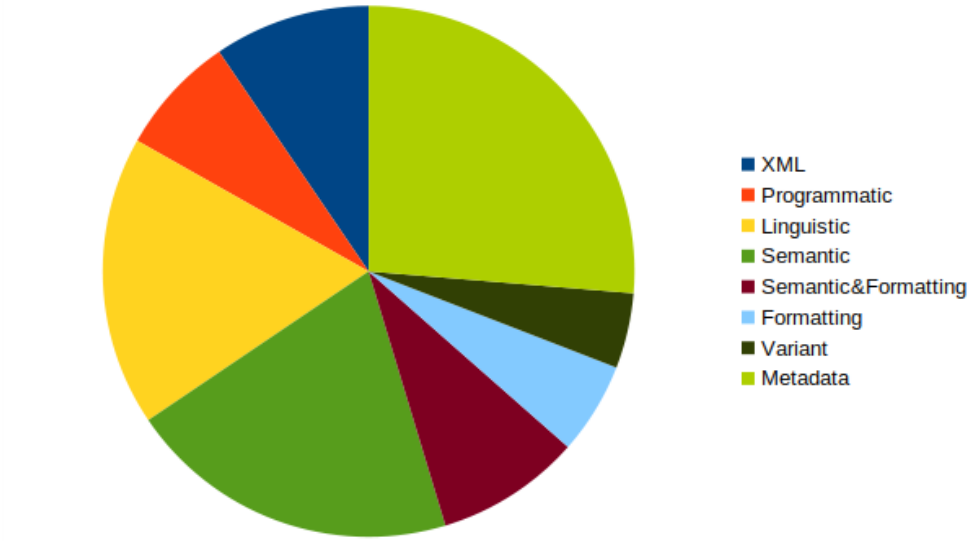


Figure 4: TEI P5 v4.2.1 tag types.

In TEI 155 tags or 26% are used for document-wide metadata. As suggested in an earlier paper (Schmidt 2014), it is now common practice to store metadata external to the document, rather than embed it in a header. There are many formats suitable for this purpose including EAD (2020), which is a close approximation of TEI metadata. Other simpler formats like Dublin Core (2021) or a custom format tailored to an edition's needs would reduce complexity. So none of these tags are needed for transcriptions in DSEs.

56 tags or 9.5% are for formatting actual XML tags in a book like the TEI Guidelines. Clearly none of these are needed in a DSE.

104 tags or 17.7% are designed for corpus linguistics applications and are unlikely to be used in a DSE.

119 or 20% are semantic tags like <date> which have little or no expression as a format. In HTML semantic information can be represented in a variety of ways: either externally as annotation using Web Annotation, RDF or OWL, or less advantageously embedded in the transcription as RDFa or microformats. Alternatively, semantic markup can be omitted entirely since it is tedious to encode, and restricts the information that can be retrieved to what has already been encoded. A more efficient approach is to use text-mining tools on lightly marked-up texts to discover meaning serendipitously ([58]). So semantic markup does not seem to be essential to making a DSE.

52 tags or 8.8% are a mixture of semantic and formatting information. An example is the <foreign> tag – indicating a foreign word or phrase. This tells us something about the content but also usually needs to be formatted in italics. TEI mixes these two aspects of markup – semantic and formatting – whereas HTML focuses on formatting markup only. Any valuable semantic information in these tags can be represented externally using annotation or discovered via text-mining. Those with useful formatting functions can be retained.

33 tags or 5.6% are purely formatting tags like <head>.

43 tags or 7.3% are for storing some kind of programming data, and don't seem to be very useful in a DSE.

27 tags or 4.6% are used to express some kind of variant, including abbreviations and their expansions, deletions and insertions, variant readings etc. In the proposed scheme the content of all variant tags will be moved into versions and layers ([50]), which are essentially separate documents. This is explained in the next section.

So in summary, the only tags that are needed in transcriptions in an interoperable DSE are those that contain formatting information. Metadata, variant and semantic information, and tags relating to non-DSE data can be omitted or represented using other technologies. In addition, only a tiny fraction of the remaining 204 tags that might possibly be used in a DSE will be needed for individual projects, and all can be represented via the class attribute used in combination with P- and SPAN-tags in HTML. The names used in the class-attributes do not have to conform to those used in the TEI schema, since all interoperation between editions will be on the higher levels of HTML, CSS and Javascript. The names of all formats used can be contained in a CSS stylesheet, included in the edition's folder. So there is no need to interoperate on the basis of those names.

As suggested above, any DIV-like tags that provide deep structure in TEI-XML can simply be removed. If they have any effect on the formatting of their contents, these properties can be transferred to the P-tags they enclose. Deep structure is mostly semantic. It was originally designed to facilitate information retrieval through tools like XQuery and XPath, but inconsistencies in the way these deep structures are recorded in markup make this approach impractical for querying documents held in heterogeneous collections, without a significant investment in homogenising markup practice ([25]). Text mining, which operates on unstructured or lightly structured texts, has exploded in interest over the past ten years and offers a way to make sense of large bodies of text with less effort ([58]).

Representing alternatives

The main difference between TEI-XML and HTML lies not in the tags but in the way textual information is encoded. Text is normally linear: a succession of character codes – letters, spaces and punctuation, regardless of whether it is presented on a two-dimensional medium like paper or stored in the memory of a computer. Virtually every comparison tool, search engine and text-mining tool assumes that this linear order reflects the order of the information. But TEI-XML records alternatives to the main text stream inline, so that words which follow one another may in fact be conceptually in parallel. This creates serious problems in text processing: 'the non-linearity of the textual and other data in question and its temporal as well as interpretive dimension appears to be a non-trivial problem' ([41]).

An example from an early draft of Valerio Magrelli's poem 'Campagna romana' ([31]; [22]) will make this point clearer. The whole page is shown in Figure 5. Figure 6 shows two small sections

of it representing revisions to the second half of line 4 and all of line 5 in the finished poem, which are used here as an example.

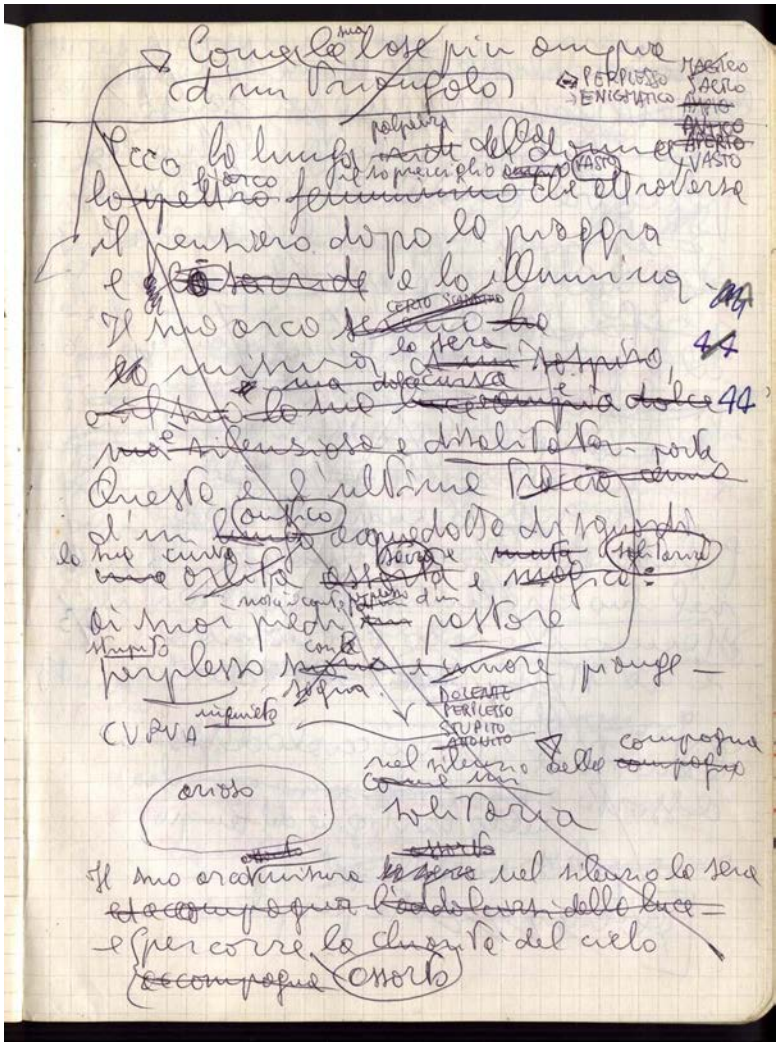


Figure 5: First draft of 'Campagna romana'.

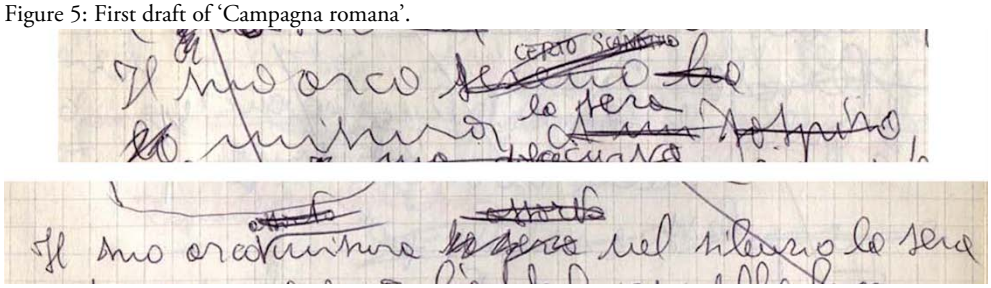


Figure 6: Lines 4 and 5 from 'campagna romana'.

A possible TEI-XML encoding (among many) of the three lines in Figure 6 might be:

```
<l>il suo arco <subst><del>sereno</del><del>certo</del>
<del>scandito</del></subst> <del>ha</del></l>
<l><del>la </del>misura <subst><del>d'un sospiro</del><add>la
sera</add></subst></l>
<l>il suo arco <del>assorto</del> misura <subst><del>la
sera</del><del>assorto</del></subst> nel silenzio la sera</l>
```

Listing 1: TEI-XML encoding of draft poetry example.

Copying out only the text as it was transcribed produces:

> il suo arco sereno certo scandito ha
> la misura d'un sospiro la sera
> il suo arco assorto misura la sera assorto nel silenzio la sera

which is clearly nonsense. It is surely not something that Magrelli ever intended to be read in this way. However, that is how the text will appear to any collation tool or full text search engine. This will stop the text from being successfully collated with other versions of the same poem, and also prevent literal searches from succeeding. A survey of 30 DSEs revealed that only 10% of editions could find literal expressions across internal variants ([48]). Keyword searching will still work, but literal searching is a useful tool of research. So why stop it from working? Text mining tools will also be confused by the nonsensical word-order.

Collating non-linear text in TEI-XML

Probably the most characteristic function of the digital scholarly edition is comparing or collating versions. Various attempts have been made to overcome the problems that this unusual structuring of textual data in TEI-XML causes. There are three basic strategies.

The first is to ignore all deletions and then collate the uncancelled text of each transcription. Any deletions in the transcription are passed through unchanged and are not collated. Dekker et al. ([14]) and Beshero-Bondar and Viglianti ([4]) follow this route. This has the advantage of being a general method, but it has the disadvantage of ignoring internal variants in each transcription. The full evolution of the text, and not just the changes between the final state of each draft or version must be visualised. Also this method doesn't take account of uncancelled variants, as can be seen in the Magrelli example above and frequently elsewhere ([50]).

The second approach, followed by Schäuble and Gabler ([46]), is to manually re-encode the transcriptions temporally to facilitate collation. This strategy was also followed when importing old TEI-XML transcriptions of Charles Harpur into the versions and layers model currently in use ([18]), and also for the digital edition of Giacomo Leopardi's *Idilli* ([23]).

Most TEI-XML transcriptions are coded topographically, that is, they record the textual changes (deletion and insertion) and their relative or absolute position on the page. A temporal encoding, on the other hand, records what the resultant states of the text were, disregarding position and

the graphical status of the text. Temporal and topographic encodings are similar for light levels of change, but quickly diverge as the alterations become more complex. For comparison, Listing 2 shows a temporal encoding of the two lines of the Magrelli poem, which is quite different from the shorter topographic encoding in Listing 1. Once the transcription has been temporally re-encoded each stage of revision can be extracted as a linear text and then collated.

```
<app><rdg><l>il suo arco sereno ha</l>
<l>la misura d'un sospiro</l></rdg>
<rdg><l>il suo arco certo</l>
<l>misura la sera,</l></rdg>
<rdg><l>il suo arco scandito</l>
<l>misura la sera,</l></rdg>
<rdg><l>il suo arco</l>
<l>misura la sera,</l></rdg>
<rdg><l>il suo arco misura la sera nel silenzio</l></rdg>
<rdg><l>il suo arco assorto misura la sera nel silenzio</l></rdg>
<rdg><l>il suo arco misura assorto nel silenzio la sera</l></rdg>
<rdg><l>il suo arco misura nel silenzio la sera</l></rdg></app>
```

Listing 2: Temporal encoding of draft poetry example.

The problem with this approach is twofold: firstly, it involves a lot of manual work re-encoding the transcription, and secondly the re-encoded transcription can't replace the original one because it contains numerous repetitions of similar textual states, which makes updating it difficult. It no longer corresponds to what the transcriber sees 'on the page', and it is also typically much longer. Schäuble and Gabler (2018) create one additional difficulty by including additions and deletions that occur between physical drafts, which do not occur in the source document. So following this path will probably mean maintaining two transcriptions of the same artefact.

A third strategy is outlined by Bleeker, Buitendijk, and Dekker ([5]). They suggest that a 'hypergraph', a combination of a tree representing the parsed textual structure and a variant graph, can be generated from each document witnessing a work. Then the hypergraphs are combined to represent the differences between all versions, including areas of local revision. This approach is still in its early stages and hasn't yet been proven to work on cases more complex than simple deletions or substitutions. Topographic encoding of separate blocks of text rotated or on different pages, or of more complex nested changes, e.g. in the Magrelli example below, would not contain enough information to place each local change in its temporal context, and so the construction of a variant graph and its subsequent collation would not seem to be possible. The transcription would first have to be re-encoded temporally.

In summary the first strategy is incomplete. It doesn't allow collation of all of the text. The second strategy is not interoperable and is impractical for anything other than short texts or for those with very few versions. The third strategy is still incomplete and unproven.

Permanently rearranging the textual data to prioritise its temporal evolution over its topography, and removing all the variant tags is currently the only practical way to facilitate comparison, searching, and editing, and to ensure full interoperability. The next two sections describe how this can be achieved.

Layers

In the example above it is possible to determine the temporal sequence of changes with a high degree of certainty. It is clear that 'd'un sospiro' was crossed out and replaced by 'la sera'. It is also clear from sense and grammar that 'la' at the start of line 2 and 'ha' at the end of line 1 must have been deleted at the same time as 'la misura d'un sospiro' was changed to 'misura la sera'. Similarly, in line 1 Magrelli first wrote 'sereno' then 'certo' then 'scandito', crossing them all out successively. The rewriting of these lines at the foot of the page follows all of these revisions. First 'nel silenzio' was added at the end. Then 'assorto' was inserted before 'misura', then replaced 'la sera' before it was moved to the next line. 'la sera' was first written before 'nel silenzio' then after it. So here, even in this complex sequence of changes, it is possible to determine the temporal sequence with a high degree of confidence.

Though it is not always the case, as in parts of the text in Figure 5, most changes in manuscripts are simple deletions and replacements, where the temporal sequence is not in doubt. In rare cases where the sequence of changes cannot be established with confidence, layers still provide a way to record the most likely scenario, while allowing re-interpretation at a later date.

Once the sequence of changes at a given location is known, level numbers can be assigned to each alteration. The first level will be the unchanged base text. The second level will be a replacement for the first level text and so on. Once each change has been assigned to a local temporal level it is a simple matter to write out the text of each layer. Layer 1 will be a copy of all text assigned to level 1, ignoring higher levels. Layer 2 will be all text assigned to level 2, plus any intervening text assigned to layer 1. Layer 3 will be the level 3 text plus all intervening text assigned to level 2 or lower. And so on. In fact this formula is very similar to that used in the Hypernietzsche Markup Language (HNML) some 15 years ago ([63]). It is also similar to the method of Schäuble and Gabler (2018), who also use layers. Taking the Magrelli example again, the TEI encoding can be rewritten as eight linear layers:

1. il suo arco sereno ha
   la misura d'un sospiro
2. il suo arco certo
   misura la sera,
3. il suo arco scandito
   misura la sera,
4. il suo arco
   misura la sera,
5. il suo arco misura la sera nel silenzio
6. il suo arco assorto misura la sera nel silenzio
7. il suo arco misura assorto nel silenzio la sera
8. il suo arco misura nel silenzio la sera

This representation of the manuscript text replaces all the variant tags with layers. It is far simpler to represent than the TEI-XML and is entirely interoperable, since each layer represents the text in its conceptual order without repetition. So it can be compared and searched using standard tools and techniques. The position of the differences in each layer can easily be recovered by comparing each layer against the others.

Bleeker, Buitendijk, and Dekker ([5]) object that layers cannot be used because they are interpretations. But topographical encoding in XML is at least equally interpretative, if not more so. For example, in the common case where some inserted text is later deleted, the transcriber must decide whether to encode it as a deleted insertion or as an inserted deletion, i.e. as <add>~~...~~</add> or as ~~<add>... </add>~~. Admittedly, layers are not created without some interpretation, but they are based on a clearer temporal logic than the vagaries of arbitrarily nested markup tags. Markup and layers are alternative tools the editor can use to realise an edition, which has always been an interpretation, even in print.

Splitter tool

To demonstrate the practicality of splitting existing TEI-XML encodings into separate layers, the Splitter tool ([51]) is included as part of this paper. This loads any TEI-XML file and looks for a limited set of variant tags: <app>, <rdg>, <lem>, <subst>, <mod>, <choice>, <add> and <del>. Other variant tags that use linking like <addSpan> or variants encoded using the double-endpoint attachment method or any other form of linking are ignored, since they cannot be converted into layers. Splitter looks for patterns in the way they are used, and assigns their contents to layers. Each source transcription is thus split into several separate XML files, each representing one layer – a copy of the original file minus the variant tags that were matched. Novel patterns of variant tags are flagged and must be manually assigned to layers, but these new patterns are remembered on subsequent runs.

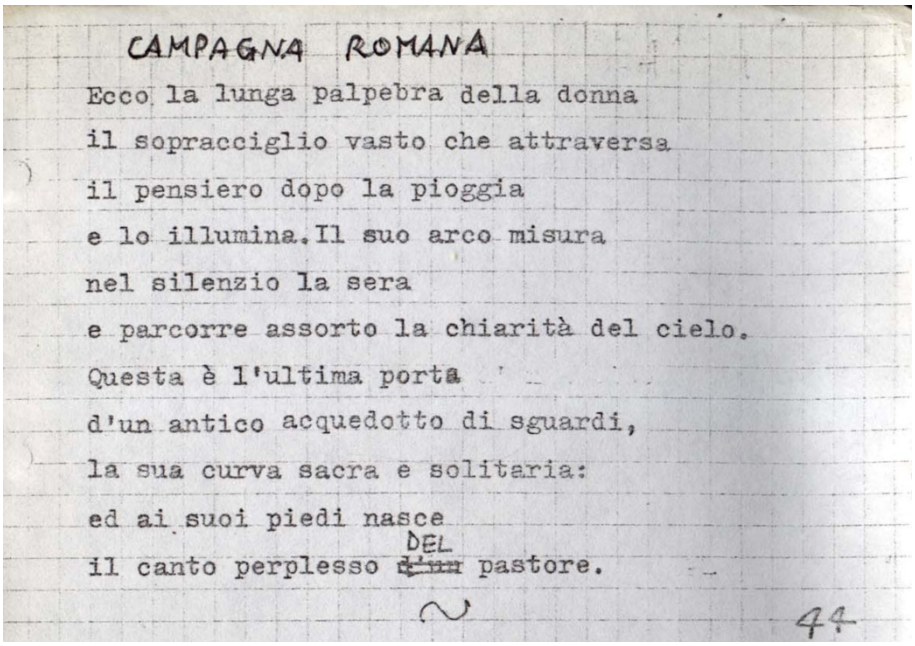
The layers produced can be examined by the user and, if not satisfactory, the original transcription must be modified and the Splitter run again. In complex cases the TEI-XML must be re-encoded temporally to obtain a satisfactory split.

Versions

It is important to note that layers are not texts the author ever wrote. They are rather *notional* transcriptions of the text intended to store local changes in their temporal order. Layers do not attempt to represent all possible combinations of independent changes, which for a given document could run into the billions. Layers only attempt to store all local variants in their immediate context by recording *one* such possibility for each level of correction.

Versions, however, *are* complete texts once written by the author at some point in time. Usually there is only one such version per document. In the above example layer-8 represents a complete version, since this records the text of the example as it was left by the author when he abandoned this particular draft.

Figure 7 shows another version of the same poem, the typescript draft 2a of Campagna romana. This is actually a (slightly altered) photocopy of an earlier state of Figure 8, the finished document, which represents draft 2b. So here are two versions in the one document preserved by the photocopying process. There are plenty of other cases where multiple versions can be discerned in the one document. Italia ([28]) for example, identified four pens used to write and revise the Neapolitan Notebook manuscript of Giacomo Leopardi's *Idilli*, each of which is a version. And within each version there are also layers of correction in the same pen.

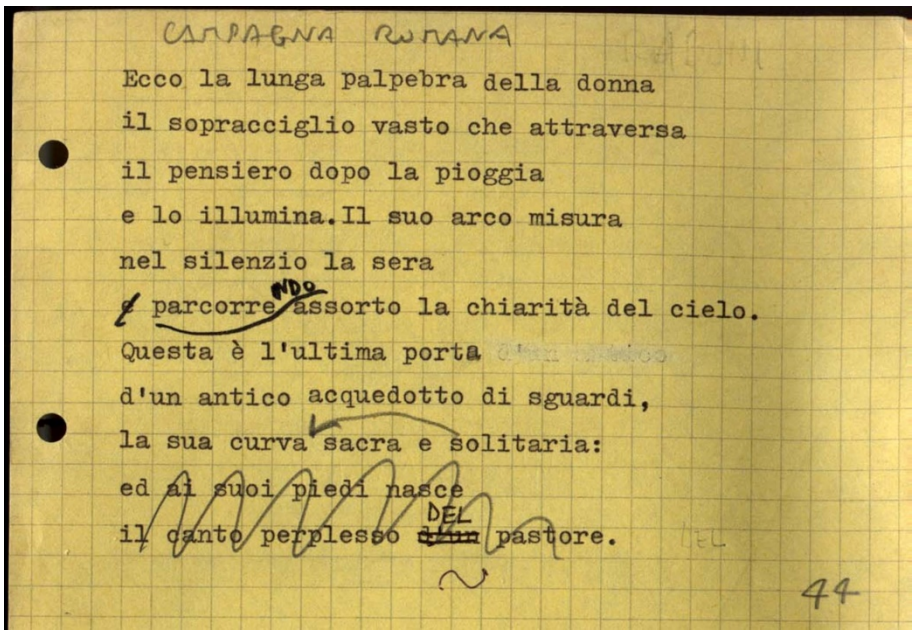


CAMPAGNA ROMANA

Ecco la lunga palpebra della donna
il sopracciglio vasto che attraversa
il pensiero dopo la pioggia
e lo illumina. Il suo arco misura
nel silenzio la sera
e parcorre assorto la chiarità del cielo.
Questa è l'ultima porta
d'un antico acquedotto di sguardi,
la sua curva sacra e solitaria:
ed ai suoi piedi nasce
il canto perplesso DEL pastore.

44

Figure 7: Modified photocopy of second draft.



CAMPAGNA ROMANA

Ecco la lunga palpebra della donna
il sopracciglio vasto che attraversa
il pensiero dopo la pioggia
e lo illumina. Il suo arco misura
nel silenzio la sera
e parcorre NDO assorto la chiarità del cielo.
Questa è l'ultima porta
d'un antico acquedotto di sguardi,
la sua curva sacra e solitaria:
ed ai suoi piedi nasce
il canto perplesso DEL pastore.

44

Figure 8: Second draft (final).

Comparison of TEI-XML and versions and layers

The TEI-XML encoding normally prioritises topography over temporal development of the text. Schäuble and Gabler ([46]) expend considerable effort in establishing a temporal encoding in TEI-XML of Virginia Woolf's essay 'A Sketch of the Past', in order to develop a 'diachronic slider'. They wouldn't make this effort unless they thought the temporal encoding was valuable. Pierazzo ([37]) also notes the coolness with which complex editions based on topographic encoding have been received, and affirms the importance of temporal encoding.

There appear to be only two things you can do with a topographic encoding.

The first is to display it diplomatically, showing inserted text above the line, deleted text as struck out, and diacritical signs. Although this type of display works satisfactorily for lightly altered texts, it is not useful for heavily edited sources like the Magrelli poem above. And it also must be asked whether users want a diplomatic display, since it is not easy to read and interpret for the purpose of analysing the text's genesis. Or is it just something that can be done with the transcription once it has been encoded in that way?

The second is suited to more complex documents. A topographic display attempts to recreate the layout of the text in the source document by redrawing it using digital type. Blocks of text that appear in the margins will be drawn in the margins. If they are upside-down or rotated they will be displayed upside-down or rotated. The various blocks can then be animated or activated by the user to highlight the sequence of changes on the page. However, the increase in clarity for the user is marginal. The layout of the text is already captured more accurately in the page image, and a temporal transcription that could flow over page-breaks would more clearly represent the evolution of the text. Another problem is that it is very labour-intensive to produce. Pierazzo ([36]) admits that this kind of display 'will not be taking off on the web any time soon'. It would appear more suited to heavily edited short texts of great importance, rather than as a general method of display in digital scholarly editions.

The versions and layers approach replaces topographic encoding with facsimile images of the source document and a temporal encoding of the sources. The diplomatic and topographic displays would then be difficult to achieve, but this is not much of a disadvantage given their shortcomings as already explained. But versions and layers as a method has one big advantage over TEI-XML: it is fully interoperable. Other advantages of the versions-and-layers approach are that the text of editions can be more easily edited, compared, searched and displayed in a clearly readable form.

Conclusion

True interoperability is a desirable and achievable goal for digital scholarly editions, although it does require the replacement of currently used XML techologies by modern web technologies such as HTML, CSS and Javascript. This change would allow editions to be shared and aggregated, and mitigate the threat of obsolescence. There are always gains and losses in any change to practice. But increasing interoperability of the DSE has to be the way forward.

References

[1] Aschenbrenner, Andreas. 2015. "Share It - Kollaboratives Arbeiten in TextGrid." In *TextGrid: Von Der Community – Für Die Community Eine Virtuelle Forschungsumgebung Für Die Geisteswissenschaften*, edited by Heike Neuroth and Sibylle Söring, 201–10. Glückstadt: Werner Hülsbusch. https://docplayer.org/5233636-Textgrid-von-der-community-fuer-die-community.html.

[2] Bauman, Syd. 2011. "Interchange vs. Interoperability." In *Proceedings of Balisage*. Montréal, Canada. https://doi.org/10.4242/BalisageVol7.Bauman01.

[3] Berners-Lee, Tim, and Mark Fischetti. 1999. *Weaving the Web*. San Francisco: Harper-Collins.

[4] Beshero-Bondar, Elisa, and Raffaele Viglianti. 2018. "Stand-off Bridges in the Frankenstein Variorum Project Interchange and Interoperability within TEI Markup Ecosystems." In *Proceedings of Balisage*. Washington, DC. https://doi.org/10.4242/BalisageVol21.Beshero-Bondar01.

[5] Bleeker, Elli, Bram Buitendijk, and Ronald Dekker. 2020. "Marking up Microrevisions with Major Implications: Non-Linear Text in TAG." In *Proceedings of Balisage: The Markup Conference*. Washington, DC. https://doi.org/10.4242/BalisageVol25.Bleeker01.

[6] Brown, Susan. 2016. "Tensions and Tenets of Socialized Scholarship." *Digital Scholarship in the Humanities* 31 (2): 283–300.

[7] Burnard, Lou. 2013. "Resolving the Durand Conundrum." *Journal of the Text Encoding Initiative* 6. https://doi.org/10.4000/jtei.842.

[8] Burnard, Lou, and Michael Sperberg-McQueen. 2012. "TEI Lite: Encoding for Interchange: An Introduction to the TEI." 2012. https://tei-c.org/release/doc/tei-p5-exemplars/html/tei_lite.doc.html#index.xml-back.1_div.1.

[9] Cummings, James. 2012. "The Compromises and Flexibility of TEI Customisation." In *Proceedings of the Digital Humanities Congress*, edited by Claire Mills, Michael Pidd, and Esther Ward. Sheffield, UK: The Digital Humanities Institute. https://www.dhi.ac.uk/openbook/chapter/dhc2012-cummings.

[10] Cummings, James. 2019. "A World of Difference: Myths and Misconceptions about the TEI." *Digital Scholarship in the Humanities* 34 (Suppl. 1): 158–79.

[11] Dahl, Ryan. 2009. "Ryan Dahl: Original Node.Js Presentation - YouTube." 2009. https://www.youtube.com/watch?v=ztspvPYybIY.

[12] "DCMI: Home." 2021. 2021. https://dublincore.org/.

[13] Dearing, Vinton. 1962. *Methods of Textual Editing*. Los Angeles: William Andrews Clark Memorial Library, University of California.

[14] Dekker, Ronald, Dirk Van Hulle, Gregor Middell, Vincent Neyt, and Joris Van Zundert. 2015. "Computer-Supported Collation of Modern Manuscripts: CollateX and the Beckett Digital Manuscript Project." *Digital Scholarship in the Humanities* 30 (3): 452–70.

[15] Durusau, Patrick. 2006. "Why and How to Document Your Markup Choices." In *Electronic Textual Editing*, edited by Lou Burnard, Katherine O'Brien O'Keefe, and John Unsworth, 299–309. Modern Language Association of America.

[16] "EAD: Encoded Archival Description (EAD Official Site, Library of Congress)." 2020. 2020. https://www.loc.gov/ead/.

[17] ECMA. 2020. "ECMA-262 - Ecma International." 2020. https://www.ecma-international.org/publications-and-standards/standards/ecma-262/.

[18] Eggert, Paul. 2019a. "Charles Harpur Critical Archive." 2019. https://charles-harpur.org/Home/Site/.

[19] Eggert, Paul. 2019b. "The Work and the Reader." In *Literary Studies: Scholarly Editing and Book History*, 80–92. Cambridge: Cambridge University Press.

[20] Eggert, Paul. 2021. "Personal Communication (from an Unknown Article ca. 2005)," 2021.

[21] Etamad, Erika, and Simon Sapin. 2018. CSS Paged Media Module Level 3. 2018. https://www.w3.org/TR/css-page-3/.

[22] Fiormonte, Domenicoco, Valentina Matiradonna, and Desmond Schmidt. 2010. "Digital Encoding as a Hermeneutic and Semiotic Act: The Case of Valerio Magrelli." *Digital Humanities Quarterly* 4 (1). http://www.digitalhumanities.org/dhq/vol/4/1/000082/000082.html.

[23] Giuffrida, Milena, Paola Italia, Simone Nieddu, and Desmond Schmidt. 2020. "From Print to Digital: A Web Edition of Giacomo Leopardi's Idilli." *Digital Scholarship in the Humanities* Advance Access. https://academic.oup.com/dsh/advance-article-abstract/doi/10.1093/llc/fqaa022/5828480.

[24] Google. 2008. "Google on Google Chrome - Comic Book." 2008. http://blogoscoped.com/google-chrome/.

[25] Haaf, Susanne, Alexander Geyken, and Frank Wiegand. 2015. "The DTA 'Base Format': A TEI Subset for the Compilation of a Large Reference Corpus of Printed Text from Multiple Sources." *Journal of the Text Encoding Initiative* 8. https://journals.openedition.org/jtei/1114.

[26] Ide, Nancy, Michael Sperberg-McQueen, Robert Amsler, Donald Walker, Susan Hockey, and Antonio Zampolli. 1988. "Proposal for Funding for An Initiative to Formulate Guidelines for the Encoding and Interchange of Machine-Readable Texts." Text Encoding Initiative. https://tei-c.org/Vault/SC/scg02.html.

[27] Interedition. 2013. "IntereditionWiki." 2013. http://www.interedition.eu/wiki/index.php/Main_Page.

[28] Italia, Paola. 2016. *Il Metodo Di Leopardi. Varianti e Stile Nella Formazione Delle «Canzoni»*. Roma: Carocci.

[29] Italia, Paola, Simone Nieddu, and Desmond Schmidt. 2019. "Idilli Di Giacomo Leopardi." 2019. http://leopardi.ecdosys.org/en/Home/.

[30] Lie, Håkon Wium, and Bert Bos. 1996. "Cascading Style Sheets, Level 1." 1996. https://www.w3.org/TR/REC-CSS1/.

[31] Magrelli, Valerio. 1980. *Ora Serrata Retinae*. Milano: Fetrinelli.

[32] Mueller, Martin, Sebastian Rahtz, Brian Pytlik Zillig, James Cummings, and Magdalena Turska. 2015. "TEI Simple: An Introduction." 2015. https://github.com/TEIC/TEI-Simple/blob/master/teisimple.odd.html.

[33] Neuroth, Heike, Andrea Rapp, and Sibylle Söring, eds. 2015. *TextGrid: Von Der Community – Für Die Community Eine Virtuelle Forschungsumgebung Für Die Geisteswissenschaften*. Glückstadt: Werner Hülsbusch. https://docplayer.org/5233636-Textgrid-von-der-community-fuer-die-community.html.

[34] OED. 2021. "Oxford English Dictionary." In , edited by Justin Stevenson, 3rd ed.

[35] O'Grady, Stephen. 2020. "The RedMonk Programming Language Rankings: June 2020." Tecosystems. July 27, 2020. https://redmonk.com/sogrady/2020/07/27/language-rankings-6-20/.

[36] Pierazzo, Elena. 2014a. "Digital Documentary Editions and the Others." *Scholarly Editing* 35. http://www.scholarlyediting.org/2014/essays/essay.pierazzo.html.

[37] Pierazzo, Elena. 2014b. "Of Time and Space: Unpacking the Draft Page: A New Framework for Digital Editionsof Draft Manuscripts." HAL archives-ouvertes.fr. https://hal.univ-grenoble-alpes.fr/hal-01182133/document.

[38] Pierazzo, Elena. 2015. "TEI: XML and Beyond." In *Papers of the Text Encoding Initiative Conference*. Lyon, France. https://tei-c.org/Vault/MembersMeetings/2015/en/index.html%3Fp=50.html.

[39] Pierazzo, Elena. 2019. "What Future for Digital Scholarly Editions? From Haute Couture to Prêt-à-Porter." *International Journal of Digital Humanities* 1: 209–20.

[40] Quin, Liam. 2013. "Xsl 2.0?" 2013. http://www.biglist.com/lists/lists.mulberrytech.com/xsl-list/archives/201311/msg00012.html.

[41] Rehbein, Malte, and Hans Walter Gabler. 2013. "On Reading Environments for Genetic Editions." *Scholarly Research and Communication* 4 (3). http://src-online.ca/index.php/src/article/viewFile/123/260.

[42] Renear, Alan. 2000. "The Descriptive/Procedural Distinction Is Flawed." *Markup Languages: Theory & Practice* 2 (4): 411–20.

[43] Robinson, Peter. 2002. "Where We Are with Electronic Scholarly Editions, and Where We Want to Be." *Jahrbuch Für Computerphilologie* 4. http://computerphilologie.digital-humanities.de/jg03/robinson.html.

[44] Robinson, Peter. 2021. "Creating and Implementing an Ontology of Texts, Documents and Works in Complex Textual Traditions." https://zenodo.org/record/4006582.

[45] Roselli Del Turco, Roberto, Chiara Di Pietro, and Chiara Martignano. 2019. "Progettazione e Implementazione Di Nuove Funzionalità per EVT 2: Lo Stato Attuale Dello Sviluppo." *Umanistica Digitale* 7. https://doi.org/10.6092/issn.2532-8816/9322.

[46] Schäuble, Joshua, and Hans Walter Gabler. 2018. "Visualising Processes of Text Composition and Revision across Document Borders." In *Digital Scholarly Editions as Interfaces*, edited by Roman Bleier, Martina Bürgermeister, Helmut Klug, Frederike Neuber, and Gerlinde Schneider, Digital Scholarly Editions as Interfaces:165–91.

Schriften Des Instituts Für Dokumentologie Und Editorik 12. Norderstedt: BoD. https://kups.ub.uni-koeln.de/9116/.

[47] Schmidt, Desmond. 2014. "Towards an Interoperable Digital Scholarly Edition." *Journal of the Text Encoding Initiative* 7. https://doi.org/10.4000/jtei.979.

[48] Schmidt, Desmond. 2016. "Enhancing Search for Complex Historical Texts." In *Digital Humanities Australia Conference*. Hobart, Australia. https://drive.google.com/file/d/1qb6hivk7JW-eJTzGlHvVlGjpYA-OcgFc/view?usp=sharing.

[49] Schmidt, Desmond. 2018. "The Current State of the Digital Scholarly Edition and Three Challenges." In *Per Una Critica Del Testo Digitale*, edited by Domenico Fiormonte. Roma: Bulzoni.

[50] Schmidt, Desmond. 2019. "A Model of Versions and Layers." *Digital Humanities Quarterly* 13 (3). http://www.digitalhumanities.org/dhq/vol/13/3/000430/000430.html.

[51] Schmidt, Desmond. 2021a. *Splitter*. https://bitbucket.org/chasharpur/splitter.

[52] Schmidt, Desmond. 2021b. "TEI." https://docs.google.com/spreadsheets/d/1_U8hjXaUG8TckSPaiLMOZhg27sz8Sr1jcfWyjYgjJwE/edit?usp=sharing.

[53] Schmidt, Desmond, and Paul Eggert. 2019. "The Charles Harpur Critical Archive. A History and Technical Report." *International Journal of Digital Humanities* 1.

[54] Shillingsburg, Peter. 2004. *Scholarly Editing in the Computer Age Theory and Practice*. Third. Ann Arbor: University of Michigan Press.

[55] Sperberg-McQueen, Michael, and Lou Burnard. 1999. "Guidelines for Electronic Text Encoding and Interchange." 1999. https://tei-c.org/Vault/GL/P3/index.htm.

[56] TEI. 2021. "The TEI Guidelines." 2021. https://tei-c.org/release/doc/tei-p5-doc/en/html/index.html.

[57] TEI, Oxford University Computing Services. 2004. "https://tei-c.org/Vault/P4/doc/html/." 2004. https://tei-c.org/Vault/P4/doc/html/.

[58] Tonkin. 2016. "Working with Text." In *Working with Text Tools, Techniques and Approaches for Text Mining*, edited by Emma Tonkin and Gregory Tourte, 1–19. Amsterdam: Chandos.

[59] Trolard, Percy. 2001. "TEI Tite — A Recommendation for off-Site Text Encoding." 2001. https://tei-c.org/Vault/P5//2.3.0/doc/tei-p5-exemplars/html/tei_tite.doc.html#index.xml-back.1_div.2.

[60] W3C. 2021. "HTML 5.2." 2021. https://www.w3.org/TR/html52/.

[61] Walsch, John, Grant Simpson, and Saeed Moaddeli. n.d. "TEI Boilerplate." Accessed March 25, 2021. https://dcl.ils.indiana.edu/teibp/.

[62] WorldWideWebSize.com. 2021. "The Size of the World Wide Web (The Internet)." 2021. https://www.worldwidewebsize.com/.

[63] Zapf, Volker. 2006. "HNML: The Hypernietzsche Markup Language." https://www.yumpu.com/en/document/read/22399002/slides-and-texts-files-hnmlpdf.