Editing tools: transcribing and encoding

Table of Contents
Editing tools: transcribing and encoding.................................................................................................................. Abstract......................................................................................................................................................... 0. Preliminary......................................................................................................................................................
1. A software culture..............................................................................................................................................
2. TEI and DDE, or Textual Encoding Initiative and Documentary Digital Edition........................................
3. Into the tools..................................................................................................................................................
   3.1. Tools.........................................................................................................................................................
      3.1.1. T-Pen...............................................................................................................................................
      3.1.2. CWRC-Writer............................................................................................................................... 3.2. Platforms..................................................................................................................................................
      3.2.1. eLaborate 4...................................................................................................................................
      3.2.2. TextGrid.........................................................................................................................................
      3.2.3. Ecdosis.............................................................................................................................................
3.3. Comparative table.............................................................................................................................................
3.4. Common features of transcription and encoding tools.................................................................................. 4. Towards a modular and complete editing environment?.............................................................................
5. Final notes......................................................................................................................................................... Bibliography............................................................................................................................................

Abstract
This paper deals with editing tools and editing platforms, i.e. programs used by editors in order to
fulfil one or more tasks in the creation of a scholarly digital edition (SDE). Recurring themes in
the field of SDEs are the standardization of XML-TEI markup and the success of documentary
digital editions (DDEs) – as compared with critical and genetic editions. The existing editing
tools reflect these practices: a considerable number of transcribing and encoding tools, mostly
TEI compliant, are developed and used in order to produce DDEs.
A small selection of applications are analysed in detail here: T-Pen, CWRC-Writer, TextGrid,
eLaborate and Ecdosis; the tools are compared in an initial summary table. A second table and
recap reflect the common features of transcribing and encoding applications.
Questions concerning the entire workflow necessary for the creation of a scholarly digital edition
are addressed in brief. In applying Manovich's concept of 'software culture' to the field of digital
ditions, editing tools emerge as crucial, insofar as they may shape investigation and, eventually,
scholarly products.
0. Preliminary.

The goal of this article is to provide an overview of the field of editing tools, and in particular of transcribing and encoding tools; it does not intend to compile a list of existing tools, which will soon be obsolete. It will address digital and non-digital editors as well as the developers of such tools; basic knowledge of XML-TEI principles is required.

For the purpose of this discussion, an editing tool is defined as a web-based or stand-alone application that an editor can use to accomplish one task (or a very restricted number of tasks) during the creation of a scholarly (digital) edition; an editing environment is defined as a platform on which different editing tools work together in order to fulfil more than one task in the creation process and, if possible, to cover the whole workflow.\(^{1}\)

This study focuses on transcribing and encoding tools among other applications used for the preparation of a scholarly digital edition (SDE). The publication phase, for instance, will not be taken into account.\(^{2}\) The article is focused on general-purpose applications, i.e. those editing tools which are not conceived specifically for one project – for one edition, but rather are intended to be used for several projects.\(^{3}\) Also, only those applications specially created for producing SDEs are taken into account; XML editors, for example, which are widely used in this process \([\text{Biblissima 2013}]\), will not be pursued here.

Overall issues concerning editing tools are addressed first; a selection of applications for transcription and encoding is analysed in the central section and their common features are summarized afterwards; finally, broader issues concerning the entire workflow for creating SDEs are briefly pursued.

1. A software culture.

In the past decades, the interaction with computing has led humanists to yield new scholarly products: not only texts, but also things, as images, corpora, software and platforms.\(^{4}\)

Understanding so-called Generative Humanities requires reckoning with a digital humanist not as a scholar dealing with a machine called computer, but rather with a number of digital scholarship practices related to computing, where computing is 'what people wanted computers to do and

---

1 For example, CollateX is a tool, TextGrid is a platform.
2 We are aware of the need for publishing system, especially out-of-the-box and user-friendly ones, in order to 'bring to life' files that lie idle in editors' computers [see Pape-Schöch-Wegner 2012 and its bibliography]. Moreover, considering publication a task for editors, and not for publishers, is one of the novelties of SDEs, and further efforts have to be devoted to the materiality of the representation, i.e. the interface in a digital representation \([\text{cf. Fiormonte 2009}]\).
3 A lot of interesting projects falls outside this study, as I shall repeat below. Among them, the Bentham Transcription Desk \(<\text{http://www.transcribe-bentham.da.ulcc.ac.uk/td/Transcribe_Bentham}>\), Transcribo \(<\text{http://transcribo.org/en/}>\) or the Online Transcription Editor \(<\text{http://wfce-ote.sourceforge.net/}>\). All of them may potentially be used for several editions, but so far has been strongly customized on one project. All URLs accessed 2015-03.
4 'The advent of Digital Humanities implies a reinterpretation of the humanities as a generative enterprise: one in which students and faculty alike are making things as they study and perform research, generating not just texts (in the form of analysis, commentary, narration, critique) but also images, interactions, cross-media corpora, software, and platforms' Burdick 2012, p. 10.
how people designed computers to do it' [McCarty 2005, p. 14]. Modelling, prototyping and testing developments have therefore become humanist scholarly activities, even as scholars struggle to have them recognized as legitimate activities in the Humanities [Schreibman and Halon 2010, Juola 2008].

What is a digital editing tool? It is, first of all, a program, a stand-alone or web application. Software is pervasive in the digital era, hence Manovich’s notion of *software society* or *software culture*. In his words, 'software has become our interface to the world, to others, to our memory and our imagination' [Manovich 2013, p. 2]. Manovich's examples vary, ranging from hospital and transport management to email providers. In order to better understand what software is, we need to analyse 'its genealogy (where it comes from), its anatomy (interfaces and operations), and its practical and theoretical effects' [*ibid.*, p. 124]. This paper will provide some targeted (and by no means comprehensive) insight into these issues.

As already mentioned, an editing tool is here defined as a piece of software used in the creation of scholarly digital editions. Word processors, concordancers or lemmatizers are programs, i.e. digital tools; dictionaries and glossaries can be computer applications or appear in print, i.e. digital or non-digital tools; a set of paper forms for bibliographical resources is a non-digital tool. 'Extending the toolkit of traditional scholarship' [Burdick 2012, p. 8] is doubtless one of the aims of Digital Humanities. The core of each discipline (and profession) can be found in its toolkit and its applicability: a tailor might not anticipate the next dress to be commissioned by her client, but with the right measuring tape, thimble, shears (etc.), s/he can produce it. Within digital media, scholars have created digital equivalents to non-digital media. However, the centrality of interpretation in the Humanities presents a challenge, as does the fact that practices are not always clearly documented and shared. How does one process irregular metre? How does one detect figures of speech? How have humanists been collating texts throughout centuries? Creating tables, using a base-text, selecting *loci critici*? As Andrews points out, the question of digital publication has attracted a lot of attention; but 'the method of production, rather than the published form that the resulting edition take, is the practice wherein lies most of the promised revolution within textual scholarship' [Andrews 2013]5.

The development of ad hoc tools for the creation of a scholarly edition is not novel. The history of editing tools remains to be written, but some pioneering projects in the field are well known: TUSTEP, for instance: a toolbox for scholarly processing textual data, designed at the Computing Center of the University of Tübingen, first implemented in the seventies and constantly upgraded until today; or Collate, a collation tool developed by Peter Robinson in the early nineties, which has only recently encountered a successor in CollateX [Dekker and Middell 2010-2014].

Theoretical reflection on digital tools for the humanities has been on the increase in past decades,

---

5 Cf. Pierazzo 2011: 'Perhaps we should just stop trying to map digital editions to printed ones and instead recognize that we are producing a different type of object, one that we can perhaps call a *documentary digital edition*. This new object necessarily comprises all three components of a digital publication—the source, the output and the tools to produce and display it—and it is worth emphasizing again that all three are scholarly products that result from editorial practice' (p. 474-5) and Van Zundert-Boot 2011: 'We argued that digital scholarly editions will be composites of three types of digital web services: data sources, processing services, and interfaces' (p. 150).
fostered by landmark writings [e.g., McCarty 2005, Bradley 2002, Andrews 2013, Unsworth 2003], projects\(^6\) and conferences\(^7\). On a practical note, a number of resources are available today; two main repositories of humanities applications, not focused only on editing but more generally on research tools for scholarly use, are DIRT (Digital Research Tools Directory)\(^8\) and TAPOR (Research Tools for Textual Studies)\(^9\). The spreadsheet Collaborative Transcription Tools set by Ben Brumfield is a useful in-progress and collaborative resource\(^10\).

The history of text-oriented applications swings between tools developed for specific uses and general-purpose ones\(^11\), producing what has been called the 'tool paradox' [Pape, Schöch, Wegner 2012]: 'the conclusion would be that either “complexity is the price paid for generality” or that “limitation is the price paid for ease-of-use,” with both standing in the way of a widespread user base'. The authors continue: 'the challenge to combine expressive power with simplicity of use is faced by any developer in the domains of application software or operating systems, and it marks the long-term evolution of such endeavours'. The application of software development models to text analysis software development offers a promising path, but it does not seem to be a priority of humanists or even of digital humanists. This is changing; increasing interest is funnelled into, for example, user-centric design or web standards.

On the matter of SDEs, the importance of software is overwhelming. In Schmidt’ somewhat provocative definition, 'SDE is software' and should therefore 'be governed by the procedures and principles of software development'. As this paper is focused on tools for preparing SDEs, it is worth clarifying that here the user is the editor, and not the reader of the edition.

2. **TEI and DDE, or Textual Encoding Initiative and Documentary Digital Edition.**

This article focuses on transcribing and encoding tools because they constitute a significant portion (at least in terms of quantity) of the existing editing tools. In this section I argue that the flourishing of transcribing and encoding tools is related to the acceptance of XML-TEI as a standard for the encoding of texts and to the predominance of documentary editions in the SDEs panorama.

---


\(^8\) <http://dirtdirectory.org/>.

\(^9\) <http://www.tapor.ca/>.

\(^10\) <https://docs.google.com/spreadsheets/d/1MFsRSZRgy3RRB4AU6AFp7IQsecqcauJLyZLGVzJFWs/>.

\(^11\) Unsworth distinguish four generations, from ad hoc programs, through reusable libraries of text-processing routines, to interactive general-purpose programs, which move over the network in the last stage [Unsworth 2003].
A recurring theme surrounding SDEs is the overwhelming presence of documentary digital editions (DDE)\(^\text{12}\), as compared to critical and genetic editions. To better understand the relevance of DDEs in the wider SDEs panorama, we should remember that many editions that have been called Electronic Archives in fact contain documentary editions of a number of 'single documents'. This is clearly exemplified in one of the earliest SDE projects, the *Piers Plowman Electronic Archive*: on the 'The texts of *Piers Plowman*’ page, the reader can find both the *Critical Edition* (for instance *The B-Version Archetype*) and the *Documentary Editions* of the manuscripts. Several scholarly, disciplinary and technological factors have influenced DDE’s development. Among the latter, the availability of digital facsimiles, for instance, has been significant. Moreover, as Pierazzo makes clear, 'critical editions require considerable effort on many fronts, starting from the development of tools to support the many operations that are part of producing such an edition, such as transcription, collation, stemma generation, and web publishing, among others. [...] DDEs, on the other hand, require less investment'. The available editing tools reflect current practices in the creation of SDEs: transcribing tools are overwhelmingly present, if compared to, for example, collation or phylogenetic tools, just as much as documentary editions are endemic if compared to critical or genetic editions.

Observing the editing tools panorama it seems that the majority of these applications takes XML-TEI markup into account. Why do they do it? Encoding is a fundamental step: a way of putting ‘intelligence in the text’ [Hockey 2000, p. VI] – a kind of intelligence that computational tools can process. It therefore carries surplus value specific to digital texts. Nevertheless, there is no common procedure to follow in determining what should be encoded. A rather basic, practical and economic proposal is to encode only what will be processed\(^\text{13}\). XML-TEI is not the only existing data format for creating SDEs: other XML languages, markup (such as LaTex) and markdown syntax, or the code of web pages, HTML, can be used. However, XML-TEI markup, extremely rich and explicitly devoted to text encoding, has become a standard for transcribing and editing literary texts and historical sources, which are normally the objects of scholarly editions. Thus existing editing tools reflect common practices in the creation of SDEs; the development of tools follows the demand for tools; and scholars seem to demand XML-TEI compliant tools.

To summarize, available editing software reflect common practices of digital editors’ work: a considerable number of transcribing and encoding tools, compliant with the XML-TEI framework, have been developed and used to produce Digital Documentary Editions, a kind of edition that is particularly successful in the SDEs' panorama.

\(^{12}\) 'For the purposes of this discussion, I define a documentary edition as an edition of a text based on a single document [...] It can assume different formats, by presenting the textual content of the document as semi-diplomatic, diplomatic, ultra-diplomatic, or even facsimile editions' Pierazzo 2011.

\(^{13}\) I.e., not encoding persons, if there is no plan to have an index of persons or to process in any other way the person-like tags.
3. Into the tools.

A number of editing tools and environments will be analysed here, focusing on encoding and transcribing while also considering collation for one of the environments. This selection follows a strictly empirical criterion of 'user-friendliness'. Only tools that require minimal computer literacy\textsuperscript{14} and not to consult complex manual; only browser or portable applications, for which no installation is needed, have been selected\textsuperscript{15}. Though this is a small selection of the available applications\textsuperscript{16}, it is worth engaging with these examples.

The tools discussed here are T-Pen and CWRC-Writer; the environments are eLaborate, TextGrid and Ecdosis. Each of these will be described and then compared in a summary table.

A variety of materials have been used to test the tools. These include modern and medieval material: a manuscript of the ancient French prose \textit{Lancelot} (Paris, BnF fr. 1430); the three witnesses of the anonymous \textit{reverdie Volez vos que je vous chant} (Paris, Arsenal, 5198, BnF fr. 845; BnF nouv. acq. fr. 1050), a manuscript of Giacomo da Lentini \textit{Madonna dir vo’ voglio} (Firenze, BNC, ms. Banco Rari 217) and a selection from the correspondence between Ernesto Monaci and Francisco Adolpho Coelho, textual scholars of international relevance between the 19th and 20th centuries.

3.1. Tools.

3.1.1. T-Pen.

T-Pen\textsuperscript{17} is an open source tool developed at the Center for Digital Theology of Saint Louis University.

Its main function is managing manuscript images and transcriptions. The T-Pen Repository contains images of 4177 manuscripts; they are not the property of T-Pen, but are linked from hosting repositories: Parker Library on the Web, e-codices, Codices Electronici Ecclesiae Coloniensis, Harvard Houghton Library, SISF - Assisi et Stanford University Libraries. Furthermore, users can upload their images.

It is possible to link an already completed transcription to one of the images or to use T-Pen as a transcribing tool.

To upload an image, one can address the 'Tools for advanced users' section. A new project is created while uploading an image, and metadata can be added in the 'Active projects' page.

\begin{footnotesize}
\begin{itemize}
\item[14] As said in the Preliminary, we include in the 'minimal computer literacy' basic knowledge of XML-TEI, as it is rather common in textual scholars who are interested in Digital Humanities and more specifically in producing SDE.
\item[15] A tool that works in a browser is made up on web pages. Within an editor, the user interaction (typing of the text, pressing buttons, etc.) modifies the HTML source and the corresponding XML, if any, calling the proper JavaScript libraries. When the user gives the command, or automatically in time, the editor interacts with a server (or a web-worker) for saving and validating.
\item[16] A lot of interesting projects falls outside this presentation, for instance Tustep \url{<http://www.tustep.unittuebingen.de/tustep_eng.html>}, the oXygen plug-in Ediarum \url{<http://www.bbaw.de/en/telota/software/ediarum>}, the Islandora TEI Editor \url{<https://jtei.revues.org/790>}, just to mention some of them.
\item[17] \url{<http://t-pen.org/TPEN/>}. Further developments announced at \url{<http://lib.slu.edu/digital-humanities/projects>.
\end{itemize}
\end{footnotesize}
T-Pen automatically parses the image, recognizing the manuscript columns and lines; users can then merge, delete, add or adjust them.

Figure 1: T-Pen. Parser (1)
The transcription is line by line; the text of the previous line is shown on the top and the text of the whole page can be shown in a right window. Users can move from one line to another using the dedicated button, the Tab key (the most time-saving option) or by clicking on the corresponding line on the image.

---

18 Cf. Lowe 2015: ‘Although this tool is undoubtedly useful for students beginning to transcribe script, reducing as it does the likelihood of eyeskip by focusing attention on each line in turn, it proved frustrating for those practiced at transcription who found that the interface slowed the progress for touch-typers’ (p. 195).
Figure 3: T-Pen. Transcription area

Several buttons are available next to the transcription area. In the Options section of the Transcript project management page, users can determine which buttons will be displayed.

Figure 4: T-Pen. Transcription project management
In the central column there is a preview of the selected buttons; by clicking on one of them in the left column, users can access the management page in order to customize buttons for inserting XML tags and special characters (keyboard shortcuts are automatically created).

Figure 5: T-Pen. Button management (1).

Figure 6: T-Pen. Button management (2).
Figure 7: T-Pen. Transcription area.
Working on the whole page transcription and not on the line by line transcription will make it easier to insert tags that include several lines of text. Notes can be added on each line by clicking on the corresponding button.
A schema can be uploaded, pointing to a URL (no internal path is admitted).
The import and export functionalities are remarkable, allowing the user to upload and download using several formats, and offering a number of options for the HTML export.

Figure 8: T-Pen. Export (1).
To summarize, T-Pen is mainly a tool for managing manuscript images and transcriptions. Editing tasks such as marking abbreviations, shift of hands, normalizations, can be fulfilled through the encoding; the customizable buttons can facilitate this task, while the line by line structure may slow down it when multiple lines are included in one tag.

### 3.1.2. CWRC-Writer

CWRC-Writer (CWRC-W)\(^\text{19}\) is a web-based transcribing tool created by the Canadian Writing Research Collaboratory (Collaboratoire scientifique des écrits du Canada). It is under development and a prototype is available online for demonstration and testing purposes. CWRC-W is a WYSIWYG XML editor that allows users to easily markup the text with TEI tags and to create RDF\(^\text{20}\).

The prototype offers three templates: for letters, poems and prose texts. The first one will be used here.

\(^{19}\) [http://www.cwrc.ca/projects/infrastructure-projects/technical-projects/cwrc-writer/]. For testing the tool, see [https://sites.google.com/site/cwrcwriterhelp/].

\(^{20}\) RDF (Resource Description Framework) is a Semantic Web Standard for data interchange. ‘RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a “triple”)’ [http://www.w3.org/RDF/].
The main elements of CWRC-W are the tags, the entities and the relations. Entities are specific tags, which can be linked in order to create a relation, i.e. a RDF triple.

The text is in the central window; on the left column the user finds the XML structure, the entities list and the relations list; on the top is a toolbar with buttons for handling tags, entities and relations, for showing or modifying the XML source and for validating, saving or uploading documents.

The letter template is suggestive of XML structure, with <opener>, text and <closer>. Each carriage return automatically creates a paragraph (<p>.

To insert a tag, the user can select a portion of the text and choose from the drop-down menu which can be opened from the toolbar or with a right click.

Figure 10: CWRC-Writer. Letter template.

Figure 11: CWRC-Writer. Insert a tag.
In the toolbar, there are buttons to create entities: person, place, date, event, organization, citation, note, text/title, correction, keyword, link. Users can select a part of the text and press the corresponding button: the tag will be added and the attributes defined in the dialogue box. Once an entity has been created, it appears in the Entities column on the left.

Figure 12: CWRC-Writer. Date entity.

Authority lists are available while creating an entity: internal lists, VIAF\textsuperscript{21} and GeoNames\textsuperscript{22}. Any name that is not in the lists can be added to the internal lists, creating a new authority.

\textsuperscript{21} The Virtual International Authority File <http://viaf.org/>.
\textsuperscript{22} The GeoNames geographical database <http://www.geonames.org/>.
Figure 13: CWRC-Writer. Steps for the creation of a Place entity.

Figure 14: CWRC-Writer. Creation of a new Title authority.
Figure 15: CWRC-Writer. Add RDF.

Figure 16: CWRC-Writer. Edit the Header.
The wizard has some disadvantages: for a new title, for instance, only a limited selection of resource types are available (audio, book, correspondence, journal, manuscript, video, web resource), and an Editor field cannot replace or be added to the Author field. Once entities have been created, users can establish relations that will be stored as RDF triples.

A dedicated dialogue box is used for editing the Header of the document.

An important functionality for advanced users is that the XML source can be shown and modified. It is possible to add an XML schema and a CSS style-sheet. One drawback is that pointing to a local resource is not allowed: like the field for associating a XML schema or a CSS style-sheet, the attribute @target for the <ref> tag, only accepts URLs.

Different views are available, with or without markup and with or without coloured brackets for entities.

Figure 17: CWRC-Writer. Document with tags, entities in the left column.

CWRC-W does not offer an Undo button or export services so far. Facsimiles management and view are absent. Nevertheless, it stands out among the selected tools because of its powerful and user-friendly functionalities for the encoding and the semantic enrichment of the text.
3.2. Platforms.

3.2.1. eLaborate 4.

eLaborate\textsuperscript{23} is a web-based editing environment developed at Huygens ING – KNAW since 2003; the fourth version was released in April 2014. It allows editors with little computer literacy to create digital scholarly editions and enables collaborative editing. In the workbench, users can upload images, transcribe and annotate the text and publish the results.

In eLaborate, the text is organised in entries; each entry corresponds to the page of a manuscript or to a column or to a document (e.g. a letter).

Once an entry has been created, users can add one or more facsimiles to it, choose the number and the names of the layers (e.g. diplomatic, normalised, translation), manage metadata and insert transcriptions. Three windows for transcribing are open in the work environment: a column for the image on the left, one column for editing the transcription in the centre and, on the right, one column for a preview in which annotations can be inserted. The latter two windows are synchronized while scrolling down the text.

Figure 18: eLaborate. Transcription area.

A toolbar is available in the edit window for styles (italic, bold, underline, strike-through, subscript, superscript) and special characters, for undoing or redoing, and for wrapping the text (without changing the line breaks that the editor may have inserted manually). A double click on a word in the right column will add a note. Styles and notes allow the user to record abbreviations, deletion, shift of hands and similar features.

\textsuperscript{23} <http://elaborate.huygens.knaw.nl/>.
In the Settings page, the core of the application, the project's leader inserts metadata and criteria for the publication, manages collaborators' rights and edits the parameters for the entries' metadata (name and order of the fields to fill).
Specific metadata for each entry can be very useful for the organization (and retrieval) of different materials, as shown by the *Rembrandt Documents Project*, powered by eLaborate3.
Again in the Settings page, users can edit the number and names of the text layers. Annotations can be organized in categories. The weak point of this functionality is that different kinds of annotations (e.g. structural, semantic, internal) are mixed on the same level.

In the History page, users will easily find all past revisions. The application uses a WordPress environment for adding editorial materials as a forward and introduction. Users will appreciate the functionality Publish Draft, which shows a preview of the publication. No export functionality is available through the interface.

Multiple editorial tasks can be accomplished using eLaborate, from managing the facsimile to publication, though encoding is not supported. It allows editors with little technical skill to create digital editions. Its strong points are the management of metadata and collaborative work. Nevertheless, eLaborate is not suitable in certain cases: multiple witnesses and aligned synoptic edition, complex layers of annotations, need to insert non-textual objects into the annotations.

3.2.2. TextGrid.

TextGrid\textsuperscript{24} is a joint project of ten partners, funded by the German Federal Ministry of Education and Research (BMBF) for the period between June 2012 and May 2015.

It provides an infrastructure with a strong focus on text-oriented research; it has a grid-capable architecture that assures interoperability and homogeneity throughout the complete workflow, including editing, storing and publishing; one of its strengths is that it enables collaborative work, associating roles and rights with users.

TextGrid is an open source software, developed in Eclipse. It is portable, and thus does not require any modification to the operating system; connection with the server is needed for the authentication.

The two main components are the Repository and the Laboratory. The first is a long-term archive; users can store, publish and research in the server-based Repository. It also includes an extensive collection of German texts from the Digital Library at zeno.org. The Laboratory is a work environment that combines several tools and services, such as an XML Editor, a Metadata Editor, a Text-Image-Link Editor, dictionaries, a German Lemmatizer, a Workflow Tool, the DigiLib viewer, the SADE Publish Tool, import and export facilities.

Text Grid’s main elements are the objects and the relations between them (stored in RDF triples), which constitute objects too. Objects are Items, Aggregations, Works, Editions and Collections. All of them must belong to a Project, which serves as a container for the management rights. An Item is the smallest unit; it can be, for instance, a plain text or XML document, a DTD, a CSS style-sheet or an image. Some examples of the relations between objects are: a Collection may contain Editions or gather Items; an Edition is a Work's manifestation.

\textsuperscript{24} <https://textgrid.de/>.
The XML Editor presents several functionalities: visualization of the XML structure, schema association and validation, content completion, WYSIWYM view (standard style-sheet) and preview (customized style-sheet). A Unicode table is available on the left column. Metadata can be inserted in the TeiHeader, or in the Metadata Editor and automatically transferred in the TeiHeader.

I will briefly describe the Text-Image-Link Editor. The chosen image is on the top part of the screen with the text below. The user selects a portion of both in order to create a link. If the selected text does not correspond to a tag, two empty <anchor> tags will be automatically created, with xml:id attributes like @xml:id="a6_start" and @xml:id="a6_end"; if the selected text does correspond to a tag, the xml:id attribute will be automatically added; if there already is an identifier, the image will be linked without any additions. The selected part of the image can have different shapes: the editor may decide to select lines, stanzas, etc. The XML document, the image and the links between them will be included while exporting a Text-Image-Link Object.

![Figure 23: TextGrid. Text-Image-Link (1).](image-url)
Overall, TextGrid is the most complete application among those analysed here: it includes a well-structured editing environment, linked to the storing service of the repository; export is available in plain text and XML and a publishing tool is integrated. As regards the transcription and the encoding, it merges the functionalities of a XML editor with those of the Text-Image-Link.

**Beyond transcription and encoding: use CollateX in TextGrid**

In addition to the basic TextGrid tools, others can be connected using the Eclipse Marketplace. CollateX is a recently released and already popular collation tool, the designated successor of Peter Robinson's Collate. It was developed within the EU-funded initiative Interedition and under the leadership of Ronald Haentjens Dekker and Gregor Middell. It is an open source program, available in Java and Python versions.

In TextGrid, CollateX runs on a collation set, which gathers plain text documents. The results can be shown as an alignment table, a variant graph or in XML (following the guidelines of the TEI module Critical Apparatus).
Figure 25: TextGrid. CollateX variant graph. Data from Inf., III, 3 (Dante Alighieri: *Commedia. A Digital Edition*, ed. P. Shaw).

Figure 26: TextGrid. CollateX alignment table.

Users can establish an equivalence set, which determines the items that should be treated as identical; variance between them will be ignored by the tool.
3.2.3. Ecdosis.

Ecdosis is an application developed by Desmond Schmidt (AustEse), in collaboration with the University of Sassari\textsuperscript{25}. It is intended as a complete editing and publishing environment. Given Ecdosis is still under development\textsuperscript{26}, it was not possible to test it; thus I will only present its structure and some of its features. Because of its peculiarities, it is nevertheless worth including it in this overview.

The internal format of Ecdosis is the MultiVersion Document (MVD), developed in order to overcome the limitations of XML technologies together with, for instance, the overlapping problem. Schmidt and Colomb affirm: 'it will not be necessary to seek a solution to the overlapping hierarchies problem if a solution can be found to the textual variation problem. Any solution to the latter is also a solution to the former' [Schmidt-Colomb 2009]. The solution is the MVD format, which uses 'the list form of the variant graph as the basis for an encoding of a single work, in all its versions or markup perspectives, as a single digital entity. [...] It enables a work to be viewed and searched, its versions compared and edited as one file'. Therefore no additional tool for collation is needed, as in the file, 'the relationships between various parts of each version, the what-is-a-variant-of-what information, is also recorded' (ibid.). One of the weak points of the MVD format seems to be that there are no versions for sections of the texts (words, characters, paragraphs, etc.), but only for the whole text. When dealing, for example, with a genetic edition, it is difficult to decide whether two deletions at different points of the text belong to the same version, or layer, as this may assume that the two deletions were made at the same time. Though this might present a problem for the edition of modern manuscripts, the format

\begin{figure}
\centering
\includegraphics[width=\textwidth]{TextGrid.png}
\caption{TextGrid. CollateX results after regularisation according to the equivalence set. Variant graph.}
\end{figure}

\textsuperscript{25} <http://www.ecdosis.net/main/>.
\textsuperscript{26} As for March 2015.
seems very suitable for recording, processing and editing works preserved in many witnesses and characterized by widespread variation, such as a number of medieval texts. A counterpart is that the usual distinction between a diplomatic and a normalized edition is hard to handle with MVD, because there is no difference between two versions of the same manuscript and two versions of the same work.

The MVD stands as the internal format of the editing platform, which accepts several input formats and provides output ones. When importing an XML document into Ecdosis, the program processes the encoding in order to create autonomous versions: each layer of nested editorial (but not structural) markup corresponds to a version.

The Ecdosis platform consist of two websites: one for the editors and one for the readers; as already mentioned, it is also a publishing platform.

In the back-ends (the editors’ side), there is a MML editor, an Event editor, a TILT editor, the Doc/Image management, the Project management and a Plain Doc editor.

The Minimal Markup Editor (MML) is a Markdown editor customizable for each project; because of the separation into layers (versions), less markup is needed. The MML handles hyphenation at line-end intelligently, converts quote-marks automatically, allows user to create ‘dividers’ between sections and multiple indentation for poetry.

Figure 28: Ecdosis. MML editor.

Events are biographical or publication events in the life of the author, which can be displayed in an interactive timeline or as a readable biography.
TILT provides text-image links, connecting words to word-shapes semi-automatically (rather than manually as is done by other tools, e.g. EPT, TILE, TextGrid Link Editor). Further developments are expected for TILT in the British Library Labs.

Figure 29: Ecdosis. Event creator.

Figure 30: Ecdosis. TILT editor.

27 <http://labs.bl.uk/>.
The Plain Doc editor is needed for writing Introductions and other paratextual materials. Overall, Ecdosis is a complete editing environment, created with a user-oriented approach and based on an innovative data format. Its strengths and weaknesses will be better evaluated once it is officially released and tested by the community.

3.3. Comparative table
In the following table, the main features of the tools that have been analysed are summarized and compared.
<table>
<thead>
<tr>
<th></th>
<th>Steps of the editing process</th>
<th>Annotation and markup</th>
<th>Import and export</th>
<th>Search functionaliti es</th>
<th>Image management</th>
<th>Online collaboration</th>
<th>Web-based or standalone</th>
<th>Licence</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-PEN</td>
<td>Metadata management, transcription, encoding, annotation</td>
<td>Annotation. Possible TEI encoding.</td>
<td>Import: TXT and XML. Export: PDF, XML/PlainText, HTML</td>
<td>No</td>
<td>Advanced. Text/Image link (line level).</td>
<td>Yes. Leader project and users with different rights.</td>
<td>Web-based</td>
<td>Open source ECL-2.0</td>
<td>Users documentation</td>
</tr>
<tr>
<td>CWRC</td>
<td>Metadata management, transcription, encoding, semantic enrichment</td>
<td>TEI encoding (embedded) and RDF encoding (standoff)</td>
<td>Import: plain text</td>
<td>No</td>
<td>No</td>
<td>No.</td>
<td>Web-based</td>
<td>Open source</td>
<td>Users documentation</td>
</tr>
<tr>
<td>eLaborate 4</td>
<td>Metadata management, transcription, annotation, publication</td>
<td>Annotation.</td>
<td>Import: plain text. Export: plain text, XML (not available on the normal user interface)</td>
<td>In metadata: advanced and friendly. In text: full text search</td>
<td>Advanced. Text/Image link (page level).</td>
<td>Yes. Leader project and users with different rights.</td>
<td>Web-based</td>
<td>Open source GNU GPLv3</td>
<td>Users and developers documentation</td>
</tr>
<tr>
<td>TextGrid</td>
<td>Metadata management, transcription, encoding, collation, lexicon creation, paratext creation, publication (+ Lemmatizer for German texts)</td>
<td>TEI encoding.</td>
<td>Import: plain text, XML. Export: plain text, XML.</td>
<td>Full text search into the project metadata and into TextGrid Repository.</td>
<td>Advanced. Text/Image link (manual).</td>
<td>Yes. Leader project and users with different rights.</td>
<td>Portable (connection with the server needed)</td>
<td>Open source See policy at <a href="https://www.textgrid.de/en/registration/download/tou/">https://www.textgrid.de/en/registration/download/tou/</a></td>
<td>Users and developers documentation</td>
</tr>
<tr>
<td>Ecdosis</td>
<td>Metadata management, transcription, encoding, paratext creation, event editor, publication</td>
<td>Markdown encoding.</td>
<td>Import: plain text, XML. Export: plain text, XML.</td>
<td>No</td>
<td>Advanced. Text/Image link (word level).</td>
<td>Yes. Leader project and users with different rights.</td>
<td>Web-based</td>
<td>Open source</td>
<td>/</td>
</tr>
</tbody>
</table>
### 3.4. Common features of transcription and encoding tools.

All the tools mentioned here are or include transcription and/or encoding facilities. The following table lists the common functionalities for transcription and encoding; other features are discussed below.

<table>
<thead>
<tr>
<th>Object</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>add / edit / delete, insert special characters, copy / paste, search, save, import / export</td>
</tr>
<tr>
<td>Image</td>
<td>add / edit / delete, zoom in / out, link text-image (image parsing or shape recognition, link mechanism, etc.), copy / paste, save, import / export</td>
</tr>
<tr>
<td>Metadata</td>
<td>add / edit / delete, may be done through a form, copy / paste, search, save, import / export</td>
</tr>
<tr>
<td>Markup</td>
<td>add / edit / delete, code completion, copy / paste, search, save, import / export</td>
</tr>
</tbody>
</table>

The tools enable different views of the content: XML (and XML structure), HTML, text, text and markup.

Certain applications, such as T-Pen, offer friendly functionalities for working with images; others, like CWRC-W, are more oriented towards easy ways to markup and enrich the text.
In the case of a side-by-side view (image and text, annotated text and preview, normalised or diplomatic editions, etc.), synchronization on scroll-down is implemented.

The encoding structure may be free or fixed. In applications for archives, using for examples the EAD or CEI Guidelines, or in filling in a TeiHeader, the encoding structure is, more or less, fixed. A form can therefore be used to markup the text.

When there is no fixed encoding structure, it is more difficult to provide simple access to the high number of available tags, for instance TEI tags. The tool will propose the most common, using buttons or other intuitive mechanisms; it is worth remembering that the more frequent tags do not correspond to the Bold, Italic and Underline buttons of most text editors: in a TEI editor, the tags <p> for paragraphs, <lb> for line breaks or <hi> for highlighted portion of text will probably be among the most used. The rest of the available tags may be accessible through menus or other graphical elements, referring to the modular structure of the TEI Guidelines. Having tools that process the schema and customize the interface accordingly (for instance buttons and menus for accessing elements and attributes), may aid easier encoding mechanisms.

4. Towards a modular and complete editing environment?

Transcribing and encoding tools do not, of course, cover the entire editing process. What are the editing tasks, in particular in a digital context? In other words, what functionalities should an editing environment cover?

Considering several formalisations of editing tasks and the peculiarities of a digital project, we can list: collection of witnesses (doc/image management and metadata), transcription, encoding, named-entity recognition, semantic enrichment, collation, analysis, constitution of the critical (or copy) text, compilation of apparatuses, compilation of indexes, preparation of paratextual material, data visualization. This list, which primarily reflects a traditional workflow transposed into a digital environment, is not comprehensive. Should an editing environment offer the possibility of automatic lemmatization, of consulting dictionaries or text repositories inside it? And what does analysis mean? As Andrews points out, beyond cladistic analysis based on

---

28 Similar tools are LIME <http://lime.cirsfid.unibo.it/>, under development at the University of Bologna and mostly oriented towards the encoding of manuscript descriptions; and Doctored.js <http://holloway.co.nz/doctored/>, an application under development as well, which deal so far with the encoding of bibliographical references.

29 Cf. the customizable Author View of the XML editor oXygen.

30 See for example the developments at the Center for Textual Studies and Digital Humanities at the Loyola University Chicago <http://www.luc.edu/ctsdh/researchprojects/hrit-catt/>; demos are accessible at <http://hritwiki.ctsdh.luc.edu/demos>; the editor concept at <http://hritwiki.ctsdh.luc.edu/galleries/hrit-php-demos/tei-editor-concept>.

31 See for example Wed, an online schema-aware XML editor, developed at the Mangalam Research Center for Buddhist Languages <http://mangalam-research.github.io/wed/>.

32 To mention only the most influential for the present discussion, Maas 1957, West 1973, Andrews 2014, Ott 2000, Zundert and Boot 2011. I will not enter into the formalisation of different kind of scholarly editions and the related (national or trans-national) traditions.

33 Once again, we leave outside publication.

34 This is what TextGrid does, for German language and literature.
methods borrowed from phylogenetic to group the manuscripts, analysis may include 'any form of stylistic analysis such as authorship attribution, or even inclusion in a corpus for large-scale data mining or the application of distant reading techniques' [Andrews 2013].

As already stated, many tools handle the management of documents and images (including metadata), the transcription and the encoding. Several tools for automatic collation are available [see Andrews 2014, pp. 181-184]; the more complete editing environments deal with the comparison of versions in different ways: for instance, TextGrid integrates CollateX; Ecdosis uses the MultiVersion Document format.

The compilation of paratextual material – such as prefaces, titles, introductions – is a basic task that can be achieved with any text editor; one option is to include a plain text editor in the working environment, as in TextGrid and Ecdosis; eLaborate uses WordPress instead. More specific applications or modules may be suitable for the creation of, for instance, a glossary. If supported by consistent markup, the compilation of indexes is a rather straightforward task that can be fulfilled during the publication stage.

One of the frequently ignored tasks in editing programs is the constitution of a critical (or copy or reading) text [see Andrews 2014, pp. 190-191]; a digital tool may suggest alternative readings and ensure the consistency of the editor’s choices.\(^{35}\)

Ideally, we will have different complete editing environments and the possibility to choose between them. If none of the (so far, only few) editing environments meets the editor’s requirements, interoperability is needed between tools in order to cover the whole workflow. So long as the number of complete editing environments which reflect different approaches, traditions and disciplines remains low, a "toolflow" or personalized workflow which includes several tools is likely the most suitable environment in which to preserve the diversity of final products, i.e. the richness and variety of textual scholarship.\(^{36}\)

The first conference on tools for textual scholarship was probably organized by Susan Hockey and held at Princeton in 1996. The critical features identified at the Text Analysis Software Planning Meeting are still valid: modularity, professionality, integration, portability [Sperberg-McQueen 1996]. In order to achieve modularity (the system should be a 'collection of relatively independent programs, each of which offers a well-defined subset of basic operations for processing textual data'), interoperability is needed. In order to achieve interoperability, standards

---

\(^{35}\) The TEI Critical Edition Toolbox (<http://ciham-digital.huma-num.fr/teitoolbox/index.php>) checks the consistency of the encoding, and not of the critical text. It would, for instance, control that all the witnesses have been mentioned in each <app>. This is fundamental for the constitution of the critical text.

\(^{36}\) More than 70 years ago Barbi wrote about Rajna's teaching: (translation is mine) 'At Rajna's school one would not learn a system [...] There was always exercises on concrete cases, and the solution was always: - so you can see that if you go ahead rationally the problems are put in the right terms, and a satisfactory answer, more or less perfect, depending on the available data, cannot miss. - We leave with the right idea that every text has its critical problem, each problem its solution, and so that editions cannot be produced following a model and, as to say, with a machine' [Barbi 1938, p. X]. In the digital context, see e.g. Régnier 2014, p. 57: 'one is inclined to imagine that in the future that could be not just one model of scholarly edition but a whole constellation of models according to the types of objects to be edited'. Cf. Leonardi 2007, Andrews 2014, Pierazzo 2014.
for data formats and communication protocols are created and implemented\(^{37}\). As regards data formats, XML is one of the most common in Digital Humanities, in spite of the many death warrants it has received\(^{38}\). TEI is the standard XML language for textual scholars. But TEI interoperability is limited because each project uses a different subset of tags (or the same tag for different purposes). It is therefore possible to build tools that can process a fixed and restricted tag set (e.g., TeiLite), but nearly impossible to build tools that can process 'all' TEI encoded texts. Schmidt's sensible proposition for ensuring interoperability consists of an effective distinction between metadata, annotation, markup and plain text (Schmidt 2014); plain text assures interoperability. This requires at least a stand-off model for the markup, which lacks widespread implementations\(^{39}\). Another basic requirement for several tools, and possibly linked to a stand-off model, is the tokenization of texts. Also, one of the difficulty is that if we conceive of a system as a 'collection of relatively independent programs', managing the updates of each program and integrating new releases in a more complete working environment can be difficult. In all cases a long-term sustainability plan is needed.

In short, modular and comprehensive editing environments are a good objective, even if comprehensiveness can never be totally achieved (new tools will emerge and very specific tasks may be needed for particular projects) and modularity is possible only under the condition of interoperability, which is something DH has to work for.

### 5. Final notes.

This study provides an overview of transcribing and encoding tools in the field of editing applications. As pieces of software, the development of such tools should take into account technical standards, operability and user experience. As scholarly products, they reflect common practices in SDEs, i.e. the use of XML-TEI for text encoding and the production of DDEs among other kind of digital editions. The compared analysis of a number of applications shows how they deal with metadata, markup and images. This overview may be of use for digital editors and developers of such tools.

Editing tools and environments are flourishing around the world, from small projects developing an ad hoc application, to a consortium of universities aiming to build the ‘final’ tool. As McCarty points out, the effort to build effective applications must be paralleled by an effort to learn how to master them. As said in the beginning, modelling, prototyping and testing developments are becoming humanist scholarly activities. In order to complete these complex tasks, the

\(^{37}\) The reference is to technical standards. The aim is not, as already said, a standardized way of editing a text. In this respect, flexibility and customizability, which Andrews points out to be fundamental to textual scholars, are not, as she suggests, necessary opposed to standards [Andrews 2013].

\(^{38}\) See for example James Clark, *XML vs the Web* 

\(^{39}\) Markup is said to be standoff, when the markup data is placed outside the text it is meant to tag.
collaborative approach of Digital Humanities is essential in gathering expertise in technologies, standards, software development and, last but not least, textual scholarship.

**Bibliography**


