



Journal of the Text Encoding Initiative

Issue 8 | December 2014 - December 2015
Selected Papers from the 2013 TEI Conference

Edition Visualization Technology: A Simple Tool to Visualize TEI-based Digital Editions

Roberto Rosselli Del Turco, Giancarlo Buomprisco, Chiara Di Pietro, Julia Kenny, Raffaele Masotti and Jacopo Pugliese



Electronic version

URL: <http://journals.openedition.org/jtei/1077>

DOI: 10.4000/jtei.1077

ISSN: 2162-5603

Publisher

TEI Consortium

Electronic reference

Roberto Rosselli Del Turco, Giancarlo Buomprisco, Chiara Di Pietro, Julia Kenny, Raffaele Masotti and Jacopo Pugliese, « Edition Visualization Technology: A Simple Tool to Visualize TEI-based Digital Editions », *Journal of the Text Encoding Initiative* [Online], Issue 8 | December 2014 - December 2015, Online since 29 December 2014, connection on 19 April 2019. URL : <http://journals.openedition.org/jtei/1077> ; DOI : 10.4000/jtei.1077

For this publication a Creative Commons Attribution 4.0 International license has been granted by the author(s) who retain full copyright.

Edition Visualization Technology: A Simple Tool to Visualize TEI-based Digital Editions

Roberto Rosselli Del Turco, Giancarlo Buomprisco, Chiara Di Pietro, Julia Kenny, Raffaele Masotti, and Jacopo Pugliese

AUTHOR'S NOTE

For the purposes of the Italian academy, R. Rosselli Del Turco is responsible for sections 1, 2.1, 3.-3.1, 3.5, 4; G. Buomprisco is responsible for section 3.4; C. Di Pietro is responsible for section 3.3; J. Kenny is responsible for section 2.3; R. Masotti is responsible for sections 2.2 and 2.4; J. Pugliese is responsible for section 3.2. R. Rosselli Del Turco also planned and revised the whole article.

1. The Need for a Digital Facsimile/Edition Browser

1.1 The Digital Vercelli Book Project

- ¹ When the transcription of the Vercelli Book manuscript (Codex CXVII, Archivio e Biblioteca Capitolare di Vercelli) passed the 50% landmark, researchers working on the project¹ started to think about the best way to visualize the edition. Thanks to the openness and support of the

Biblioteca Capitolare, it was decided to abandon the original plan of a CD/DVD publication, largely inspired by digital editions such as the *Electronic Beowulf*,² in favor of a web-based publication. While this decision was critical in that it allowed us to select the most supported and widely-used medium, we soon discovered that it did not make choices any simpler. On the one hand, the XSLT stylesheets provided by TEI are great for HTML rendering, but do not include support for image-related features (such as the text-image linking available thanks to the P5 version of the TEI schema) and tools (including zoom in/out, magnifying lens, and hot spots) that represent a significant part of a digital facsimile and/or diplomatic edition; other features, such as an XML search engine, would have to be integrated separately, in any case. On the other hand, there are powerful frameworks based on CMS³ and other web technologies⁴ which looked far too complex and expensive, particularly when considering future maintenance needs, for our project's purposes. Other solutions, such as the EPPT software⁵ developed by K. Kiernan or the Elwood viewer⁶ created by G. Lyman, either were not yet ready or were unsuitable for other reasons (proprietary software, user interface issues, specific hardware and/or software requirements).

1.2 Standard vs. Fragmentation

- 2 We concluded that, while the TEI schemas and Guidelines are a solid foundation for philological work, this excellent standard is matched by an astounding diversity of publishing tools, particularly when it comes to digital editions, especially editions that include images of manuscripts. As a consequence, a single scholar, or a small group of researchers, can easily encode an image-based edition in TEI XML, but will have to look for further support and resources to publish it, which is a serious obstacle to greater acceptance of digital philology methods and techniques. While continuing to inquire about a good, community-tested solution, and in spite of being aware of the dangers coming from the so-called NIH (Not Invented Here) syndrome, we started looking into a project-specific solution that could fulfill our needs.

1.3 First Experiments

- 3 At first, however, EVT was more an experimental research project for students at the *Informatica Umanistica* course of the University of Pisa⁷ than a real attempt to solve the digital edition viewer problem. We aimed at investigating some user interface-related aspects of such a viewer, in

particular certain usability problems that are often underestimated by similar projects (Rosselli Del Turco 2011), and at encouraging the use of standards to ensure the maximum longevity for the edition; we also considered releasing the source code as free software right from the start, since we firmly support open source and open access within the academic community. The first prototypes implemented on the basis of our concepts were promising, but eventually we reached a dead end: in spite of all our good intentions the User Interface (UI) looked cluttered; the software sported several secondary features, such as a rich text editing widget, but lacked critical ones (text-image linking, for instance), and was not fully stable, possibly as a result of too many widgets of different origins being used at the same time. On the architectural side, data was loaded straight into the web code, which had to be done by hand, with no options for configuration at all. In conclusion: not bad for a first attempt, but falling far short of our original goals.

2. The Current EVT Version

2.1 EVT v. 2.0: Rebooting the Project

- 4 To get out of the impasse we decided to completely reboot the project, removing secondary features and giving priority to fundamental ones. We also found a solution for the data-loading problem: instead of finding a way to load the data into the software we decided to build the edition around the data itself. Making the TEI XML files the starting point means that the editor can focus on his work, marking up the transcription text, with very little configuration needed to create the edition. This approach also allowed us to quickly test XML files belonging to other edition projects, to check if EVT could go beyond being a project-specific tool. The inspiration for these changes came from work done in similar projects developed within the TEI community, namely [TEI Boilerplate](#),⁸ John A. Walsh's collection of [XSLT stylesheets](#),⁹ and Solenne Coutagne's work for the ["Berliner Intellektuelle 1800–1830"](#) project.¹⁰ Through this approach, we achieved two important results: first, usage of EVT is quite simple—the user applies an XSLT stylesheet to their already marked-up file(s), and when the processing is finished they are presented with a web-ready edition; second, the web edition that is produced is based on a client-only architecture and does not require

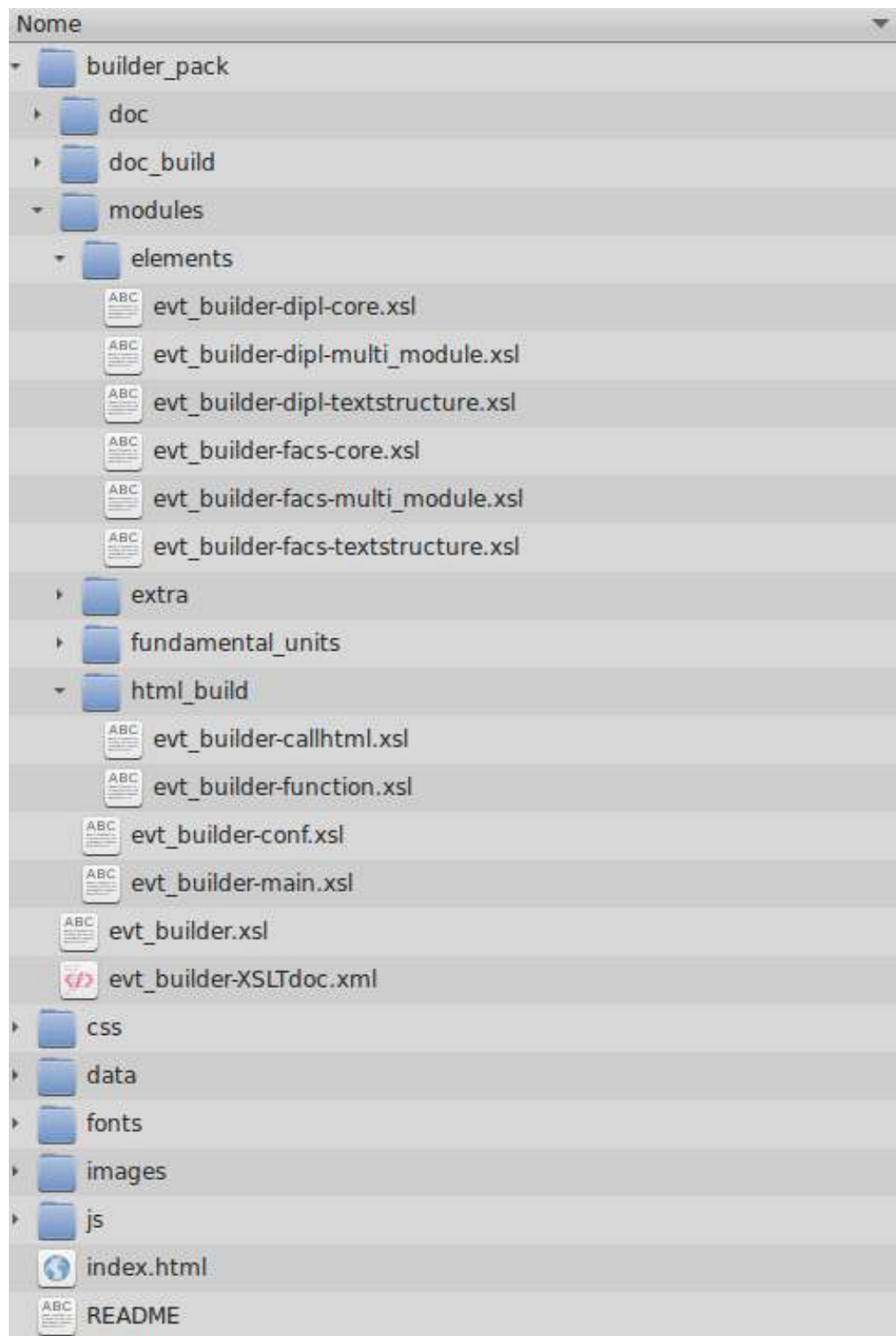
any additional kind of server software, which means that it can be simply copied on a web server to be used at once, or even on a cloud storage service (provided that it is accessible by the general public).

- 5 To ensure that it will be working on all the most recent web browsers, and for as long as possible on the World Wide Web itself, EVT is built on open and standard web technologies such as HTML, CSS, and JavaScript. Specific features, such as the magnifying lens, are entrusted to jQuery plug-ins, again chosen from the best-supported open-source ones to reduce the risk of future incompatibilities. The general architecture of the software, in any case, is modular, so that any component which may cause trouble or turn out to be not completely up to the task can be replaced easily.

2.2 How it Works

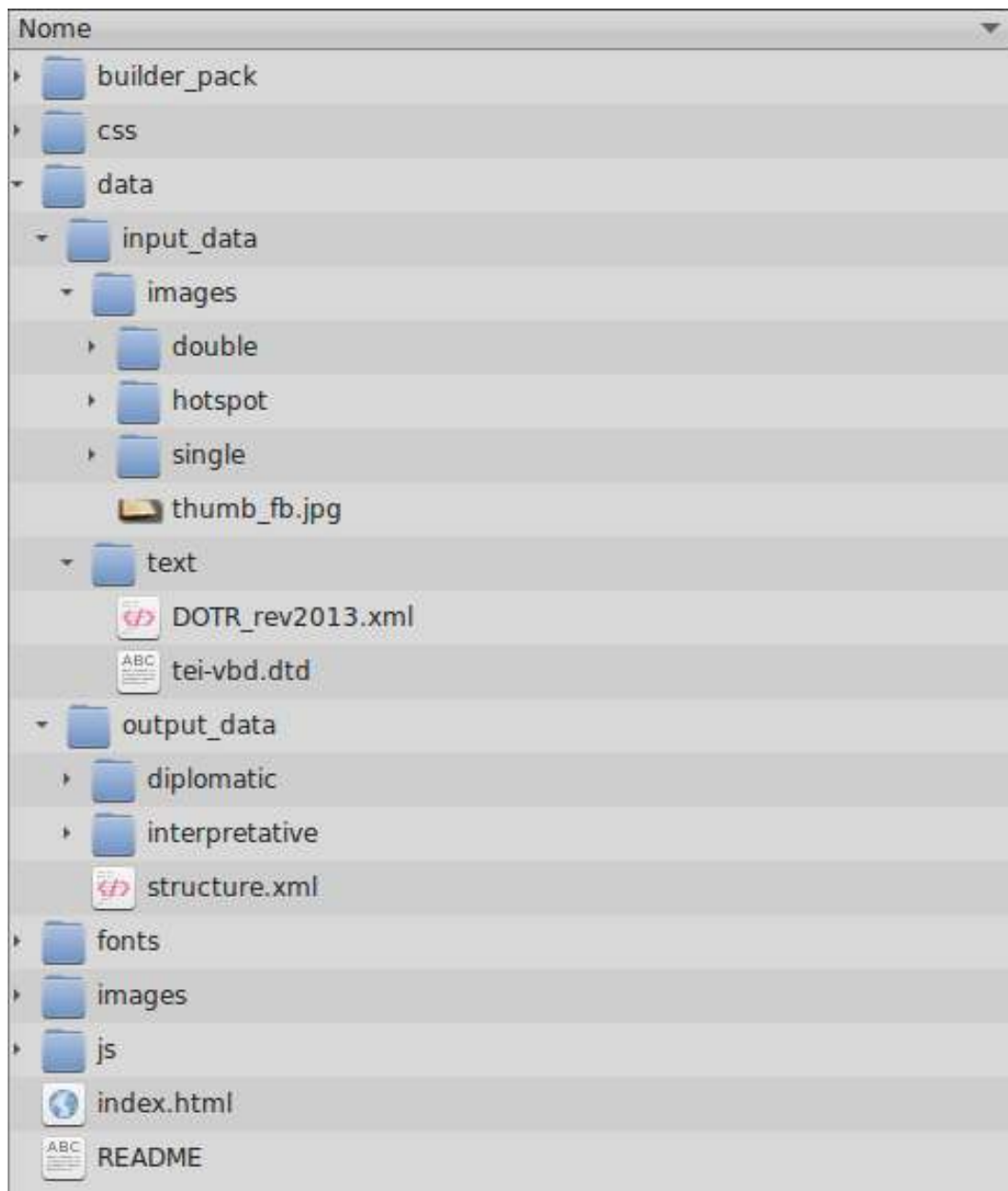
- 6 Our ideal goal was to have a simple, very user-friendly drop-in tool, requiring little work and/or knowledge of anything beyond XML from the editor. To reach this goal, EVT is based on a modular structure where a single stylesheet (`evt_builder.xsl`) starts a chain of XSLT 2.0 transformations calling in turn all the other modules. The latter belong to two general categories: those devoted to building the HTML site, and the XML processing ones, which extract the edition text lying between folios using the `<pb>` element and format it according to the edition level. All XSLT modules live inside the `builder_pack` folder, in order to have a clean and well-organized directory hierarchy.

Figure 1. The EVT builder_pack directory structure.



- 7 Therefore, assuming the available formatting stylesheets meet your project's criteria, to create a digital edition you only have to follow three simple steps:
- copy the edition data into the data/input_data folder: there are different sub-folders for text (TEI XML documents) and images (these have to follow a specific naming convention);
 - optionally you can modify transformation options by editing `evt_builder-conf.xml`, to specify for example the number of edition levels or presence of images;
 - you can then apply the `evt_builder.xml` stylesheet to your TEI XML document using the Oxygen XML editor or another XSLT 2-compliant engine.

Figure 2. The EVT data directory structure.



- 8 When the XSLT processing is finished, the starting point for the edition is the `index.html` file in the root directory, and all the HTML pages resulting from the transformations will be stored in the `output_data` folder. You can delete everything in this latter folder (and the `index.html` file), modify the configuration options, and start again, and everything will be re-created in the assigned places.

2.3 The XSLT stylesheets

- 9 The transformation chain has two main purposes: generate the HTML files containing the edition and create the home page which will dynamically recall the other HTML files.
- 10 The EVT builder's transformation system is composed of a modular collection of XSLT 2.0 stylesheets: these modules are designed to permit scholars to freely add their own stylesheets and to manage the different desired levels of the edition without influencing other parts of the system, for instance the generation of the home page.
- 11 The transformation is performed applying a specific XSLT stylesheet (`evt_builder.xsl`) which includes links to all the other stylesheets that are part of the transformation chain and that will be applied to the TEI XML document containing the transcription.
- 12 EVT can be used to create image-based editions with different edition levels starting from a single encoded text. The text of the transcription must be divided into smaller parts to recreate the physical structure of the manuscript. Therefore, it is essential that paginated XML documents are marked using a TEI page break element (`<pb>`) at the start of each new page or folio side, so that the transformation system will be able to recognize and handle everything that stands between a `<pb>` element and the next one as the content of a single page.
- 13 The system is designed to generate an arbitrary number of edition levels: as a consequence, the user is required to indicate how many (and which) output levels they intend to create by modifying the corresponding parameter in the configuration file.
- 14 The following code segment shows the default configuration with two edition levels:

```
<xsl:variable xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
name="edition_array" as="element()*">
  <edition>Diplomatic</edition>
  <edition>Interpretative</edition>
</xsl:variable>
```

- 15 The user can change the number of output levels simply by adding or removing an <edition> element in the `evt_builder-conf.xml` file, and they can also personalize the edition's name by changing the content of the <edition> element. For example, if they wish to generate a critical level, they are required to add <edition>Critical</edition> to the `edition_array` variable.¹¹
- 16 Once the XML file is ready and the parameters are set, the EVT builder's transformation system uses a collection of stylesheets to divide the XML file containing the text of the transcription into smaller portions, each one corresponding to the content of a folio, recto or verso, of the manuscript. For each of these text fragments it creates as many output files as requested by the file settings.
- 17 In order to create the contents of these files, templates required to handle the edition level are selected by using the value of their @mode attribute. For example, by associating the transformation for the diplomatic edition to the `dipl` mode, the correct content of the pages of this edition is obtained applying the templates that have "dipl" as value for @mode. This requires that each one of the pages selected by the system is processed by an `<xsl:apply-templates select="current-group()" mode="dipl"/>` instruction before its content is inserted into the diplomatic output file.
- 18 Using XSLT modes it is possible to separate the rules for the different transformations of a TEI element and to recall other XSLT stylesheets in order to manage the transformations or send different parts of a document to different parts of the transformation chain. This permits the extraction of different texts for different edition levels (diplomatic, diplomatic-interpretative) processing the same XML file, and to save them in the HTML site structure, which is available as a separate XSLT module.
- 19 The use of modes also allows users to separate template rules for the different transformations of a TEI element and to place them in different XSLT files or in different parts of a single stylesheet. So templates such as the following

```
<xsl:template xmlns:xsl="http://www.w3.org/1999/XSL/Transform" match="tei:abbr" mode="dipl"/>
```

and

```
<xsl:template xmlns:xsl="http://www.w3.org/1999/XSL/Transform" match="tei:abbr" mode="int"/>
```

can be placed in different stylesheets or in different parts of the same stylesheet without generating conflicts.

- 20 If the TEI elements that are processed are placed in an HTML element and these elements' `@class` attribute values are based on the scheme `[edition_level]-[TEI_element's_name]`,¹² it is then possible to retain the semantic information contained in the markup and, if necessary, to associate the element with the corresponding class in the CSS rules so as to specify the visualization and highlighting of the item.
- 21 As already stated above, the editors will be able to freely add their own stylesheets in order to personally manage the different levels of the desired edition. What is required from users is to:
- personalize the edition generation parameter as shown above;
 - copy their own XSLT files containing the template rules to generate the desired edition levels in the directory that contains the stylesheets used for TEI element transformation (`builder_pack/modules/elements`);
 - include the new stylesheets in the file used to start the transformation chain (`builder_pack/evt_builder.xsl`);
 - associate a mode value to the new edition level transformation;
 - add a `@mode` attribute with the corresponding value to all the template rules used for that transformation.
- 22 For the time being, this kind of customization has to be done by hand-editing the configuration files, but in a future version of EVT we plan to add a more user-friendly way to configure the system.

2.4 Features

- 23 At present, EVT can be used to create image-based editions with two possible edition levels: diplomatic and diplomatic-interpretative; this means that a transcription encoded using elements belonging to the appropriate TEI module¹³ should already be compatible with EVT, or require only minor changes to be made compatible. The Vercelli Book transcription schema is based on the standard TEI schema, with no custom elements or attributes added: our tests with similarly encoded texts showed a high grade of compatibility. A critical edition level is currently being researched and it will be added in the future.

24 When the website produced by EVT is loaded in a browser, the viewer will be presented with the manuscript image on the left side, and the corresponding text on the right: this is the default view, but on the main toolbar at the top right corner of the browser window there are icons to access all the available views:

- “Image-Text” view: as mentioned above, this is the default view showing a manuscript folio image and the corresponding text in one or more edition levels;
- “Text-Text” view: conceived to compare different edition levels, which can be chosen by means of the drop-down menu in the text frame toolbar;
- “Bookreader”: this view expands the image frame to show double-side images of the manuscript (that is showing the *verso* side of a folio and the following one’s *recto* side). At the moment it is necessary that these images be double-sided; in a future version of EVT it will be possible to place single folio images side-by-side and to browse them in a synchronized way.

Figure 3. The bookreader view.



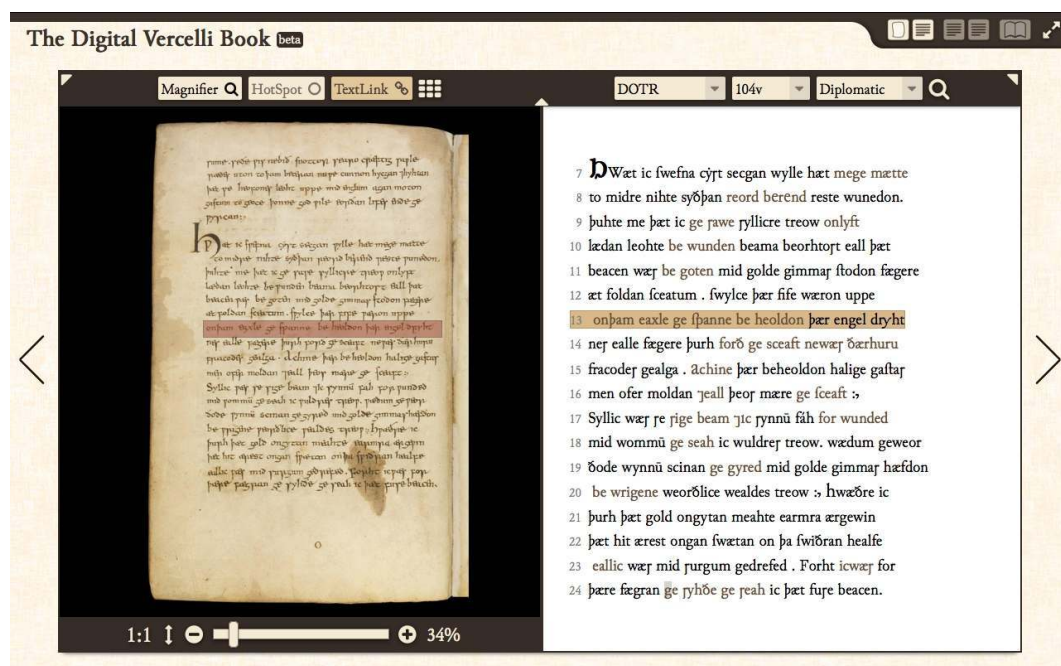
- 25 It is interesting to notice how the “bookreader” view is automatically populated with content during the transformation phase with no action required by the editor. The only necessary requirement at the encoding level, in fact, is that the editor should encode folio numbers by means of the <pb> element including *r* and *v* letters to mark recto and verso pages, respectively. EVT will take care of automatically associating each folio to the images copied in the `input_data/images` folder using a “verso-recto” naming scheme (for example: 104v-105r.png). It is of course possible that in some cases the transformation process is unable to return the correct result: this is why we decided to save this kind of information in an XML file, `structure.xml`, separate and independent from the HTML interface; this file will be updated automatically every time the transformation process is started and can be customized by the editor.

- 26 Although the different views access different kinds of content, such as single side and double side images, the navigation algorithms used by EVT allow the user to move from one view to another without losing the current browsing position.
- 27 All content is shown inside HTML frames designed to be as flexible as possible. No matter what view one is currently in, one can expand the desired frame to focus on its specific content, temporarily hiding the other components of the user interface. It is furthermore possible to collapse the frame toolbars to increase the space devoted to content visualization; it is important to notice, however, that we recommend using EVT in full-screen mode to see images and text at the maximum possible screen resolution. The collapse and restore actions are triggered by icons embedded in the interface, but one can also press the Esc key to instantly return to the default layout.
- 28 On the image side, several tools are available to improve analysis of manuscript images. The zoom feature is always active, so that the user can zoom in or out at any time by means of the mouse scroll wheel or using the appropriate slider in the bottom toolbar. More tools are available on the top toolbar:
- “Magnifier”: a magnifying lens to explore the manuscript images in greater detail; the lens shows an area of a high resolution version of the same folio image, which means a better view compared to the standard zoom feature.
 - “HotSpot”: if activated, areas of the images for which there are specific notes and/or details will be highlighted. Clicking on them will call up a pop-up window with the related information. This feature is quite recent and still needs to be refined; the pop-up window may be replaced by a better solution.
 - “TextLink”: when this feature is active, lines in the manuscript are highlighted and linked to the corresponding lines of the edition text, and vice versa.
 - “Thumbnails”: will show miniature images for all the available manuscript digitized folios.
- 29 The image-text feature is inspired by Martin Holmes’s [Image Markup Tool](#)¹⁴ software and was implemented in XSLT and CSS; all the other features are achieved by using jQuery plug-ins.
- 30 In the text frame tool bar you can see three drop-down menus which are useful for choosing texts, specific folios, and edition levels, and an icon that triggers the search functionality. Again, the editor can modify the `structure.xml` file to change the text-related information, or even to exclude some of the images from the navigation system.

2.5 A First Use Case

- 31 On December 24, 2013, after extensive testing and bug fixing work, the EVT team published a beta version of the [Digital Vercelli Book](#) edition,¹⁵ soliciting feedback from all interested parties. Shortly afterwards, the version of the EVT software we used, improved by more bug fixes and small enhancements, was made available for the academic community on [the project's SourceForge site](#).¹⁶

Figure 4. The Digital Vercelli Book edition based on EVT v. 0.1.48. Image-text linking is active.



3. Future Developments

- 32 EVT development will continue during 2014 to fix bugs and to improve the current set of features, but there are also several important features that will be added or that we are currently considering for inclusion in EVT. Some of the planned features will require fundamental changes to the software architecture to be implemented effectively: this is probably the case for the Digital Lightbox (see section 3.4), which requires a client-server architecture (section 3.5), instead of the current client-only model, to perform some of the existing and planned actions. The currently

developed search engine (section 3.2) may prove too limited when dealing with larger text collections, but thanks to the new architecture it will be possible to consider new, more powerful solutions.

3.1 New Layout

- 33 One important aspect that has been introduced in the current version of EVT is a completely revised layout: the current user interface includes all the features which were deemed necessary for the Digital Vercelli Book beta, but it also is ready to accept the new features planned for the short and medium terms. Note that nontrivial changes to the general appearance and layout of the resulting web edition will be necessary, and this is especially the case for the XML search engine and for the critical edition support. Fortunately the basic framework is flexible enough to be easily expanded by means of new views or a redesign of the current ones.

3.2 Search Engine

- 34 The EVT search engine is already working and being tested in a separate development branch of the software; merging into the main branch is expected as soon as the user interface is finalized. It was implemented with the goal of keeping it simple and usable for both academics and the general public.
- 35 To achieve this goal we began by studying various solutions that could be used as a basis for our efforts. In the first phases of this study we looked at the principal XML databases, such as BaseX, eXist, etc., and we found a solution by envisioning EVT as a distributed application using the client-server architecture. For this test we selected the [eXist](#)¹⁷ open source XML database, and in a relatively short time we created, sometimes by trial-and-error, a prototype that queried the database for keywords and highlighted them in context.
- 36 While this model was a step in the right direction and partially operational, we also felt that it was not sufficiently user-friendly, which is a critical goal of the entire project. In fact, forcing the editor to install and configure specific server software is a cumbersome requirement. Moreover, thanks to its client-only architecture, up to this point EVT could work as a desktop application or an off-line web application that could be accessed anywhere, and possibly distributed in optical formats (CD or DVD). Forcing the prerequisites of an Internet connection and of dependency on a server-based XML database would have undermined our original goal. Going the database route

was no longer an option for a client-only EVT and we immediately felt the need to go back to our original architecture to meet this standard. This sudden turnaround marked another chapter in the research process and brought us to the current implementation of EVT Search.

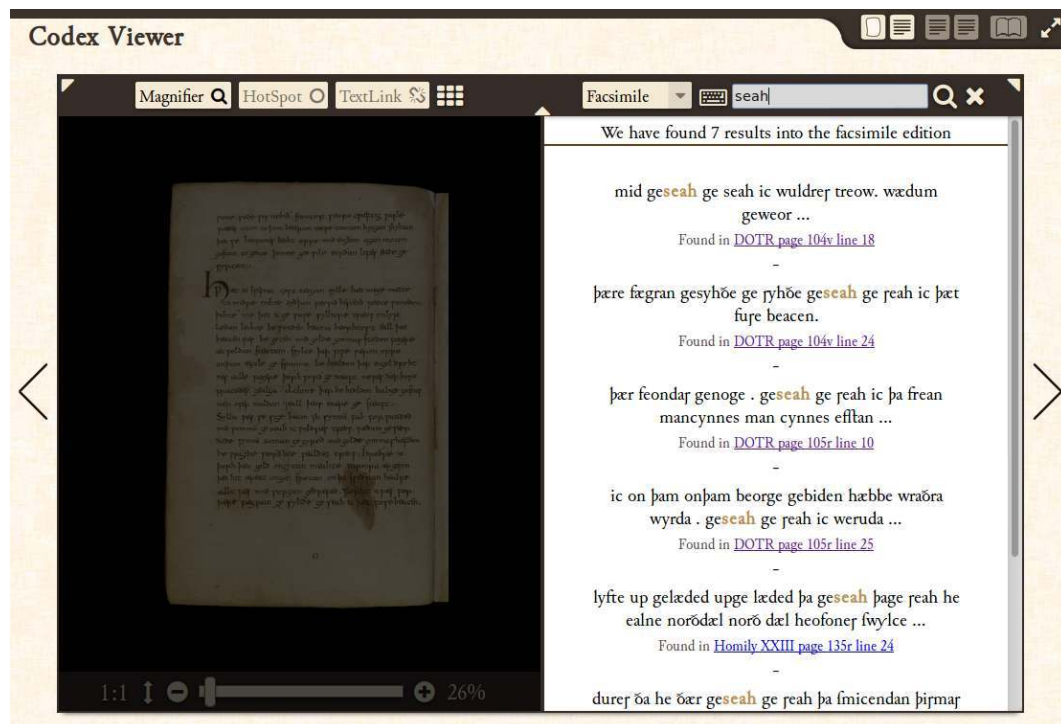
- 37 However, this new vision also had obvious limitations and issues. An XML database could provide us with crucial functionality typical of every information retrieval system: an indexer, a powerful and flexible server-side language (XQuery), and some useful built-in functions for semantic and linguistic analysis (such as keyword in context and concordances). It became clear that our implementation should take those missing functionalities into account and consider alternative technologies.
- 38 To simplify the planning process, we focused on the bare functionalities that would be expected by the user. Essentially, we found that at least two of them were needed in order to make a functional search engine: free-text search and keyword highlighting. To implement them we looked at existing search engines and plug-ins programmed in the most popular client-side web language: JavaScript. In the end, our search produced two answers: *Tipue Search* and *DOM manipulation*.

3.2.1 Tipue Search

- 39 `Tipue search`¹⁸ is a jQuery plug-in search engine released under the MIT license and aimed at indexing and searching large collections of web pages. It can function both offline and online, and it does not necessarily require a web server or a server-side programming/query language (such as SQL, PHP, or Python) in order to work. While technically a plug-in, its architecture is quite interesting and versatile: Tipue uses a combination of client-side JavaScript for the actual bulk of the work, and JSON (or JavaScript object literal) for storing the content. By accessing the data structure, this engine is able to search for a relevant term and bring back the matches.
- 40 Tipue Search operates in three modes:
- in *Static mode*, Tipue Search operates without a web server by accessing the contents stored in a specific file (`tipuedrop_content.js`); these contents are presented in JSON format;
 - in *Live mode*, Tipue Search operates with a web server by indexing the web pages included in a specific file (`tipuesearch_set.js`);
 - in *JSON mode*, Tipue Search operates with a web server by using AJAX to load JSON data stored in specific files (as defined by the user).

- 41 This plug-in suited our needs very well, but had to be modified slightly in order to accommodate the requirements of the entire project. Before using Tipue to handle the search we needed to generate the data structure that was going to be used by the engine to perform the queries. We explored some existing XSL stylesheets aimed at TEI to JSON transformation, but we found them too complex for the task at hand. So we modified our own stylesheets to produce the desired output.
- 42 This output consists of two JSON files:
- `diplomatic.json` contains the text of the diplomatic edition of the Vercelli Book;
 - `facsimile.json` contains the text of the facsimile edition of the Vercelli Book.
- 43 These files are produced by including two templates in the overall flow of XSLT transformations that extract crucial data from the TEI documents and format them with JSON syntax. The procedure complements well the entire logic of automatic self-generation that characterizes EVT.
- 44 After we managed to extract the correct data structure, we began to include the search functionality in EVT. By using the logic behind Tipue JSON mode, we implemented a trigger (under the shape of a select tag) that loaded the desired JSON data structure to handle the search (diplomatic or facsimile, as mentioned above) and a form that managed the query strings and launched the search function. Additionally, we decided to provide the user with a simple virtual keyboard composed of essential keys related to the Anglo-Saxon alphabet used in the Vercelli Book.
- 45 The performance of Tipue Search was deemed acceptable and our tests showed that even large collections of data did not pose any particular problem.

Figure 5. Experimental search interface.



3.2.2 Keyword Highlighting through DOM Manipulation

- 46 The solution to keyword highlighting was found while searching many plug-ins that deal with this very problem. All these plug-ins use JavaScript and DOM manipulation in order to wrap the HTML text nodes that match the query with a specific tag (a span or a user-defined tag) and a CSS class to manage the style of the highlighting. While this implementation was very simple and self-explanatory, making use of simple recursive functions on relevant HTML nodes has proved to be very difficult to apply to the textual contents handled by EVT.
- 47 HTML text within EVT is represented as a combination of text nodes and `` elements. These spans are used to define the characteristics of the current selected edition. They contain both philological information about the inner workings of the text and information about its visual representation. Very often the text is composed of spans that handle different versions of words (such as the sub-elements of the TEI `<choice>` element) or highlight an area of a word (based on the TEI `<hi>` element, for example).

- 48 This type of markup would not have constituted a problem if it had wrapped complete words, since the plug-ins could recursively explore its content and search for a matching term. In certain portions of the text, however, some letters are separated by a span from the rest of the word (for example: `Hello`). Since the plug-ins worked on the level of text nodes, a word split by spans would be seen as two different text nodes (for example: *H* and *ello*) and could not be matched by the query, making the standard transversal algorithm of the plug-ins impossible to use without any further substantial modifications.
- 49 To solve this problem we recreated the transversal algorithm entirely, making it more intelligent and context-sensitive. In order to do so, we explored the text to be highlighted by using a map that kept track of all the spans and text nodes and their starting and ending position in this flow. At the end of the process, we simply inserted the span elements (or the user-defined elements) in the positions marked by the map.

Figure 6. Highlighting of search results in the normal view.

21 egeƿa tome iu icwæƿ geworden wita heardoft . leodum la
 22 ðoƿt ærþan **ichim** lifer weg rihtne gerymde reord be
 23 rendū hwæt me þageweorðode wuldreƿ ealdor oferholm

- 50 We found the performance of this solution acceptable and our text showed that even dynamic highlighting (performed as the user inputs text) posed no problem for this implementation.

3.3 Embedded Transcription and Critical Edition Support

3.3.1 Embedded Transcription

- 51 In some contexts, such as encoding of epigraphical texts, the text transcribed is conceived more as a help for a better understanding of what is written on the object's surface than as a stand-alone edition. This kind of transcription can be encoded by means of the embedded transcription method as described in section 11.2.2 of the [TEI Guidelines](#),¹⁹ in which words and other written traces are not supplied in a `<text>` in parallel with a `<facsimile>`, but are encoded as subcomponents of the elements representing the physical surfaces inside `<sourceDoc>`. The text transcribed, therefore,

is not separated from the information about the image, but is placed inside a <zone> element, which defines two-dimensional areas within a <surface>, and is transcribed using one or more <line> elements.

- 52 Originally EVT could not handle this particular encoding method, since the XSLT stylesheets could only process TEI XML documents encoded according to the traditional transcription method. Since we think that this is a concrete need in many cases of study (mainly epigraphical inscriptions, but also manuscripts, at least in some specific cases), we recently added a new feature that will allow EVT to handle texts encoded according to the embedded transcription method. This work was possible due to a small grant awarded by EADH.²⁰

3.3.2 Support for Critical Edition

- 53 One important feature whose development will start at some point this year is the support for critical editions, since at the present moment EVT allows dealing only with diplomatic and interpretative ones. We aim not only to offer full support for the TEI Critical Apparatus module, but also to find an innovative layout that can take advantage of the digital medium and its dynamic properties to go beyond the traditional, static, printed page: “The layers of footnotes, the multiplicity of textual views, the opportunities for dramatic visualization interweaving the many with each other and offering different modes of viewing the one within the many—all this proclaims ‘I am a hypertext: invent a dynamic device to show me.’ The computer is exactly this dynamic device” (Robinson 2005, § 12).
- 54 A digital edition can, of course, maintain the traditional layout, possibly moving the apparatus from the bottom of the page to a more convenient position, but could and should also explore different ways of organizing and displaying the connection between edition text and the critical apparatus: even if an electronic screen is just as limited by its two-dimensional form factor as a printed book, there are several ways to overcome this limitation, which we intend to research and test until we find an effective solution.
- 55 Furthermore, a real added value granted by the digital medium is the opportunity to transcribe completely every witness and to visualize the different readings of each one in the wider context of the text of the witness itself, and not only as a single, isolated variant inside the critical apparatus. In fact, thanks to hypertextual links, it is possible to connect every single entry to the precise

fragment of text where it belongs, allowing a deeper study and an easier comparison of the main text and the variant texts. A tool based on this approach would really take advantage of the dynamic and hypertextual nature of a web-based edition.

- 56 Some of the problems related to this approach are related to the user interface and the way it should be designed in order to be usable and useful: how to conceive and where to place the graphical widgets holding the critical apparatus, how to integrate these UI elements in EVT, how to contextualize the variants and navigate through the witnesses' texts, and more. There are other problems, for instance scalability issues (how to deal with very big textual traditions that count tens or even hundreds of witnesses?) or the handling of texts produced by collation software, which strictly depend on the current TEI Critical Apparatus module. Considering that there is a subgroup of the TEI's Manuscript Special Interest Group devoted to significantly improving this module, we can only hope that at least some of these problems will be addressed in a future version.

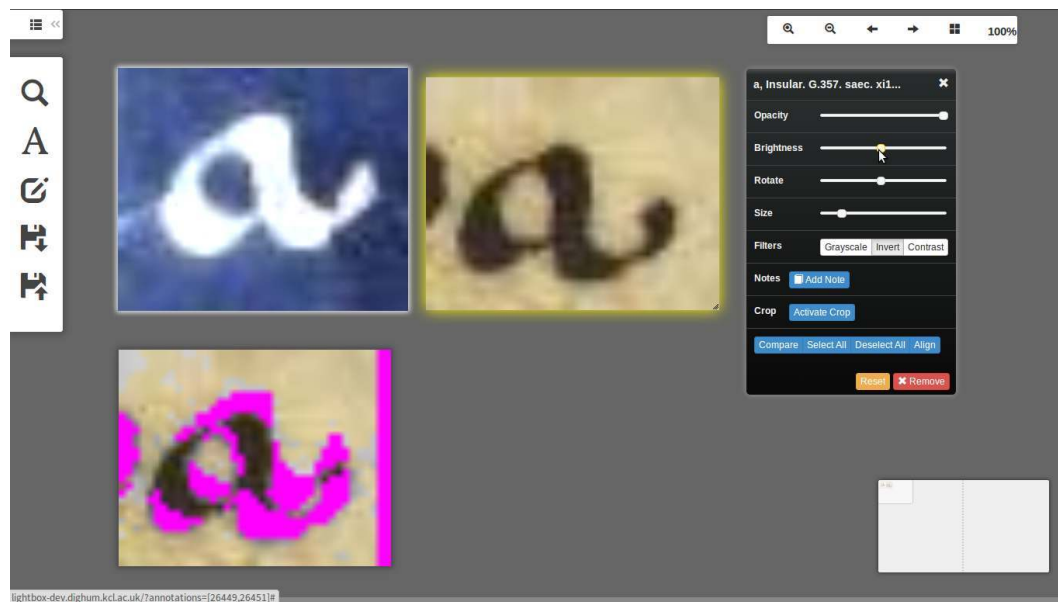
3.4 Digital Lightbox

- 57 Developed first at the University of Pisa, and then at King's College London as part of the [DigiPal](#)²¹ project, the [Digital Lightbox](#)²² is a web-based visualization framework which aims to support historians, paleographers, art historians, and others in analyzing and studying digital reproductions of cultural heritage objects. The methodology of research inspiring development of this tool is to study paleographic elements in a qualitative way, helping scholars' interpretations as much as possible, and therefore to reject any automatic methods such as pattern recognition and clustering which are supposed to return quantitative and objective results. Although ongoing projects making use of these computational methods are very promising, the results that may be obtained at this time are still significantly less precise (with regard to specific image features, at least) than those produced through human interpretation.
- 58 Initially developed exclusively for paleographic research, the Digital Lightbox may be used with any type of image because it includes a set of general graphic tools. Indeed, the application allows a detailed and powerful analysis of one or more images, arranged in up to two available workspaces, providing tools for manipulation, management, comparison, and transformation of images. The

development of this project is consistently tested by paleographers at King's College London working on the DigiPal project, who are using the web application as a support for analyzing and gathering samples of paleographic elements.

- 59 The software offers a rich set of tools: besides basic functions such as resizing, rotation, and dragging, it is possible to use a set of filters—such as opacity, brightness, color inversion, grayscale effect, and contrast—which, used in combination, are able to significantly improve the quality of digital images or specific areas of them. For example, one of the possible applications for filters is the improvement of burned or otherwise damaged manuscripts, which are obviously much more difficult to read and transcribe. Images can also be compared using an automatic tool which allows the user to merge two images, returning a third one made up of the differences between them. A typical application of the combined use of these tools is the disambiguation of different types of the same character (allographs, according to the DigiPal terminology).

Figure 7. Comparison and disambiguation of allographs in the Digital Lightbox.



- 60 Collaboration is a very important characteristic of Digital Lightbox: what makes this tool stand apart from all the image-editing applications available is the possibility of creating and sharing the work done using the software framework. First, you can create collections of images and then export them to the local disk as an XML file; this feature not only serves as a way to save the work, but also to share specific collections with other users. Moreover, it is possible to export (and,

consequently, to import) working sessions, or, in other words, the current status of the work being done using the application: in fact, all the images, letters, and notes present on the workspace will be saved when the user leaves and restored when they log in again. These features have been specifically created to encourage sharing and to make collaborative work more effective and easy. Thanks to a new HTML5 feature, it is possible to support the importing of images from the local disk to the application without any server-side function.

- 61 Digital Lightbox has been developed using some of the latest web technologies available, such as HTML5, CSS3, the front-end framework [Bootstrap](#),²³ and the JavaScript (ECMAScript 6) programming language, in combination with the [jQuery](#) library.²⁴ The code architecture has been designed to be modular and easily extensible by other developers or third parties: indeed, it has been released as open source software on GitHub,²⁵ and is freely available to be downloaded, edited, and tinkered with.
- 62 The Digital Lightbox represents a perfect complementary feature for the EVT project: a graphic-oriented tool to explore, visualize, and analyze digital images of manuscripts. While EVT provides a rich and usable interface to browse and study manuscript texts together with the corresponding images, the tools offered by the Digital Lightbox allow users to identify, gather, and analyze visual details which can be found within the images, and which are important for inquiries relating, for instance, to the style of the handwriting, decorations on manuscript folia, or page layout.
- 63 An effort to adapt and integrate the Digital Lightbox into EVT is already underway, making it available as a separate, image-centered view, but there is a major hurdle to overcome: some of the DL features are only possible within a client-server architecture. Since EVT or, more precisely, a separate version of EVT will migrate to this architecture, at some point in the future it will be possible to integrate a full version of the DL. Plans for the current, client-only version envision implementing all those features that do not depend on server software: even if this means giving up interesting features such as collaborative work and annotation, we believe that even a subset of the available tools will be an invaluable help for manuscript image analysis. Furthermore, as noted above, thanks to HTML5 and CSS3 it will become more and more feasible to implement features in a client-only mode.

3.5 New Architecture

⁶⁴ In September 2013 we met with researchers of the `Clavius on the Web` project²⁶ to discuss a possible use of EVT in order to visualize the documents that they are collecting and encoding; the main goal of the project is to produce a web-based edition of all the correspondence of this important sixteenth–seventeenth-century mathematician.²⁷ The integration of EVT with another web framework used in the project, the eXist XML database, will require a very important change in how the software works: as mentioned above, everything from XSLT processing to browsing of the resulting website has been done on the client side, but the integration with eXist will require a move to the more complex client-server architecture. A version of EVT based on this architecture would present several advantages, not only the integration of a powerful XML database, but also the implementation of a full version of the Digital Lightbox. We will try to make the move as painless as possible and to preserve the basic simplicity and flexibility that has been a major feature of EVT so far. The client-only version will not be abandoned, though for quite some time there will be parallel development with features trickling from one version to the other, with the client-only one being preserved as a subset of the more powerful one.

4. Conclusion

⁶⁵ Started as an experimental, project-based learning initiative to explore issues related to the publishing of TEI-encoded digital editions, this software has grown to the point of being a potentially very useful tool for the TEI community: since it requires little configuration, and no knowledge of programming languages or web frameworks except for what is needed to apply an XSLT stylesheet, it represents a user-friendly method for producing image-based digital editions. Moreover, its client-only architecture makes it very easy to test the edition-building process (one has only to delete the output folders and start anew) and publish preliminary versions on the web (a shared folder on any cloud-based service such as Dropbox is all that is needed).

⁶⁶ While EVT has been under development for 3–4 years, it was thanks to the work and focus required by the Digital Vercelli Book release at end of 2013 that we now have a solid foundation on which to build new features and refine the existing ones. Some of the future expansions also pose important research questions: this is the case with the critical edition support, which touches a sensitive area of the very recent digital philology discipline.²⁸ The collaborative work features of the Digital

Lightbox are also critical to the way modern scholars interact and share their research findings. Finally, designing a user interface capable of hosting all the new features, while remaining effective and user-friendly, will itself be very challenging.

BIBLIOGRAPHY

- Gabler, Hans Walter. 2010. "Theorizing the Digital Scholarly Edition." *Literature Compass* 7(2): 43–56. doi:10.1111/j.1741-4113.2009.00675.x.
- Kiernan, Kevin S., ed. 2013. *The electronic Beowulf*. 3rd edition. London: British Library. CD-ROM.
- Robinson, Peter. 2005. "Current Issues in Making Digital Editions of Medieval Texts—Or, Do Electronic Scholarly Editions Have a Future?" *Digital Medievalist* 1.1. <http://www.digitalmedievalist.org/journal/1.1/robinson/>.
- . 2013. "Towards a Theory of Digital Editions." *Variants: The Journal of the European Society for Textual Scholarship* 10: 105–31.
- Rosselli Del Turco, Roberto. 2011. "After the Editing is Done: Designing a Graphic User Interface for Digital Editions." *Digital Medievalist* 7 <http://www.digitalmedievalist.org/journal/7/rosselliDelTurco/>.
- . Forthcoming. "The Battle We Forgot to Fight: Should We Make a Case for Digital Editions?" In *Digital Scholarly Editing: Theory, Practice and Future Perspectives*, edited by Matthew Driscoll and Elena Pierazzo. Cambridge: Open Book Publishers.
- Siemens, Ray, Meagan Timney, Cara Leitch, Corina Koolen, and Alex Garnett. 2012. "Toward Modeling the Social Edition: An Approach to Understanding the Electronic Scholarly Edition in the Context of New and Emerging Social Media." *Literary and Linguistic Computing* 27(4): 445–61. doi:10.1093/lc/fqs013.
- TEI Consortium. 2014. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Version 2.7.0. Last updated on September 16, 2014. N.p.: TEI Consortium. <http://www.tei-c.org/Vault/P5/2.7.0/doc/tei-p5-doc/en/html/index.html>.

APPENDIXES

Appendix 1. The EVT Team

Project lead:

Roberto Rosselli Del Turco

UI design, HTML programming:

Roberto Rosselli Del Turco, Julia Kenny, and Raffaele Masotti

XSLT framework and graphic plug-ins:

Julia Kenny and Raffaele Masotti

XML search engine:

Jacopo Pugliese

Critical edition and embedded transcription support:

Chiara Di Pietro

Digital Lightbox:

Giancarlo Buomprisco

EVT website: <http://sourceforge.net/projects/evt-project/>

NOTES

- 1 The Digital Vercelli Book project, <http://vbd.humnet.unipi.it/>.
- 2 Support site for the *Electronic Beowulf*, <http://ebeowulf.uky.edu/>.
- 3 The Omeka framework (<http://omeka.org/>) supports publishing TEI documents; see also Drupal (<https://drupal.org/>) and TEICHI (<http://www.teichi.org/>).
- 4 Such as the eXist XML database, <http://exist-db.org/>.
- 5 Edition Production and Presentation Technology, <http://sourceforge.net/projects/eppt/>.
- 6 Elwood Viewer, <http://www3.iath.virginia.edu/seenet/elwoodinfo.htm>.
- 7 BA course, <http://infouma.di.unipi.it/laurea/index.asp>.
- 8 TEI Boilerplate, <http://dcl.slis.indiana.edu/teibp/>.
- 9 `tei2html`, <https://github.com/jawalsh/tei2html>.
- 10 Digitale Edition *Briefe und Texte aus dem intellektuellen Berlin um 1800*, <https://sites.google.com/site/annebaillot/digitale-edition>.

- 11 The available edition levels are described in the software documentation. Adding another edition level requires providing the corresponding stylesheet.
- 12 For example, to process the element `<abbr>` for the diplomatic edition: `dip1-abbrev`.
- 13 See chapter 11, “Representation of Primary Sources,” in the [TEI Guidelines](#).
- 14 The UVic Image Markup Tool Project, http://www.tapor.uvic.ca/~mholmes/image_markup/.
- 15 Full announcement on the project blog, <http://vbd.humnet.unipi.it/?p=2047>. The beta edition is directly accessible at <http://vbd.humnet.unipi.it/beta/>.
- 16 Edition Visualization Technology: Digital edition visualization software, <http://sourceforge.net/projects/evt-project/>.
- 17 eXist-db, <http://exist-db.org/>.
- 18 Tipue Search, <http://www.tipue.com/search/>.
- 19 See “Embedded Transcription,” <http://www.tei-c.org/Vault/P5/2.6.0/doc/tei-p5-doc/en/html/PH.html#PHZLAB>.
- 20 See EADH Small Grant: Call for Proposals, <http://eadh.org/support/eadh-small-grants-call-proposals>.
- 21 DigiPal: Digital Resource and Database of Palaeography, Manuscript Studies and Diplomatic, <http://www.digipal.eu/>.
- 22 A beta version is available at <http://lightbox-dev.dighum.kcl.ac.uk>.
- 23 Bootstrap, <http://getbootstrap.com/>.
- 24 <http://jquery.com/>.
- 25 Digital Lightbox, <https://github.com/Gbuomprisco/Digital-Lightbox/>.
- 26 See <http://claviusontheweb.it/web/index.php>. A preliminary test using a previous version of EVT is available at <http://claviusontheweb.it/EVT/>.
- 27 Currently preserved at the Archivio della Pontificia Università Gregoriana.
- 28 Digital philology makes use of ICT methods and tools, such as text encoding, in the context of textual criticism and philological study of documents to produce digital editions of texts. While many of the first examples of such editions were well received (see for instance [Kiernan 2013](#); also see [Siemens 2012](#) for an example of the new theoretical possibilities allowed by the digital

medium), serious doubts concerning not only their durability and maintainability, but also their methodological effectiveness, have been raised by some scholars. The debate is still ongoing, see [Gabler 2010](#), [Robinson 2005](#) and [2013](#), [Rosselli Del Turco, forthcoming](#).

ABSTRACT

The TEI schemas and guidelines have made it possible for many scholars and researchers to encode texts of any kind for (almost) all kinds of purposes, but this excellent standard is matched by an astounding diversity of publishing tools, which is particularly true when it comes to image-based digital editions. The different needs of scholars, coupled with the constant search for an effective price/result ratio and the local availability of technical skills, have led to a remarkable fragmentation: publishing solutions range from simple HTML pages produced using the TEI stylesheets (or the TEI Boilerplate software) to very complex frameworks based on CMS and SQL search engines. Researchers of the Digital Vercelli Book project started looking into a simple, user-friendly solution and eventually decided to build their own: EVT (Edition Visualization Technology) has been under development since about 2010 and has turned into a flexible tool that can be used to create a web-based digital edition starting from transcription files encoded in TEI XML. This paper describes why this tool was created, how it works, and developments planned for the future.

INDEX

Keywords: manuscript transcription, diplomatic edition, critical edition, digital edition, XSLT, web publication, digital lightbox

AUTHORS

ROBERTO ROSSELLI DEL TURCO

Roberto Rosselli Del Turco is an assistant professor at the University of Torino, where he teaches Germanic philology. He is also an assistant professor of humanities computing at the University of Pisa. He directs the Digital Vercelli Book project and is co-director of the Visionary Cross Project.

GIANCARLO BUOMPISCO

Giancarlo Buomprisco is a graduate student at the University of Pisa currently collaborating with the DigiPal project at King's College London.

CHIARA DI PIETRO

Chiara Di Pietro is a graduate student at the University of Pisa, working toward a MA degree in Informatica Umanistica.

JULIA KENNY

Julia Kenny is a graduate student at the University of Pisa, working toward a MA degree in Informatica Umanistica.

RAFFAELE MASOTTI

Raffaele Masotti is a graduate student at the University of Pisa, working toward a MA degree in Informatica Umanistica.

JACOPO PUGLIESE

Jacopo Pugliese is a graduate student at the University of Pisa, working toward a MA degree in Informatica Umanistica.