

TEI BOILERPLATE

John A. Walsh, Indiana University
Grant Simpson, Indiana University

Introduction

TEI Boilerplate (<http://tei.boilerplate.org>) is a lightweight solution for publishing styled TEI (Text Encoding Initiative) P5 content directly in modern browsers. With TEI Boilerplate, TEI XML files can be served directly to the web without server-side processing or wholesale translation to HTML.

TEI is an XML-based language for describing and analyzing literary texts and other documents of interest to humanities scholars. Although TEI provides mechanisms for describing the design, presentational, and material features of the source document, projects and individual scholars that use TEI are responsible for developing their own methods—or implementing existing solutions—for converting the TEI to a presentation-ready state for the web or print (Rahtz, 2012). Two potential paths to reach this goal are:

1. Transforming TEI to HTML using XSLT and styling the HTML output with CSS.
2. Styling the TEI directly with CSS by referencing a CSS stylesheet from within the TEI document.

Both of these approaches have advantages and disadvantages. Although HTML is the language of the web and, as such, is well supported by browsers, HTML's descriptive capabilities are much less expressive than TEI's. When TEI is transformed to HTML, much of the richness of the TEI is lost or obscured in the resulting HTML. However, the browser understands HTML very well and knows, for example, when to initiate retrieval of a document based on certain user events, such as clicking a link. The second option, CSS-styled TEI, delivers the TEI document directly to the browser. However, while the browser may apply CSS to format and style a TEI document, the browser does not understand the semantics of TEI. For instance, the browser does not understand that TEI's `<ptr>` and `<ref>` elements are linking elements.

TEI Boilerplate bridges the gap between these two approaches by making use of the built-in XSLT (1.0) capabilities of browsers to embed the TEI XML, with minimal modifications, within an HTML5 shell document. Features expected of web documents, such as clickable links and display of linked images, are enabled through *selective transformation* of a very small number of TEI elements and attributes. Both the HTML5 shell and the embedded TEI are styled using CSS.

TEI Boilerplate is not intended to be a replacement for the many excellent XSLT solutions for publishing and displaying TEI/XML on the web. It is intended to be a simple and lightweight alternative to more complex XSLT solutions. There are both practical and theoretical advantages to this lightweight approach.

TEI Boilerplate requires a robust, modern browser to do its work. It is compatible with current versions of Firefox, Chrome, Safari, and Internet Explorer (IE 9).

Practical Benefits

- Heavy-lifting is done by CSS and Javascript
- XSLT expertise is not required
- System provides a simple solution for teaching and training contexts
- TEI semantics are delivered directly to the browser
- Design and processing are targeted at semantically rich TEI data rather than less-expressive HTML surrogates

XSLT is far more powerful than CSS and can manipulate XML documents in ways that CSS cannot. XSLT can radically transform documents and can combine multiple XML data sources into a single output file or multiple output files. But the power of XSLT comes at a cost. XSLT is a far more complex language than CSS and has a far steeper learning curve. There are relatively few expert XSLT programmers compared to the numbers of expert Web developers with extensive knowledge of HTML/CSS/Javascript.

On the other hand, JavaScript is an extremely capable language for doing fancy things with XML content. JavaScript frameworks like JQuery and EXT JS and the major browsers provide excellent support for manipulating the DOM (Document Object Model) of the web page. With TEI Boilerplate, the TEI document becomes part of the DOM, and JavaScript may then be used to perform more complex transformations, movements, and other manipulations of the TEI content than can be achieved with CSS alone.

TEI Boilerplate gives HTML/CSS/Javascript documents direct access to original TEI content, and it gives TEI documents direct access to the substantial capabilities of HTML, CSS, and JavaScript—the dominant document format, styling language, and (client-side) programming language of the web. TEI Boilerplate aims for simplicity and elegance, but it also facilitates complexity and innovation by exposing TEI content directly to the capabilities of JavaScript, the many powerful JavaScript frameworks, and CSS.

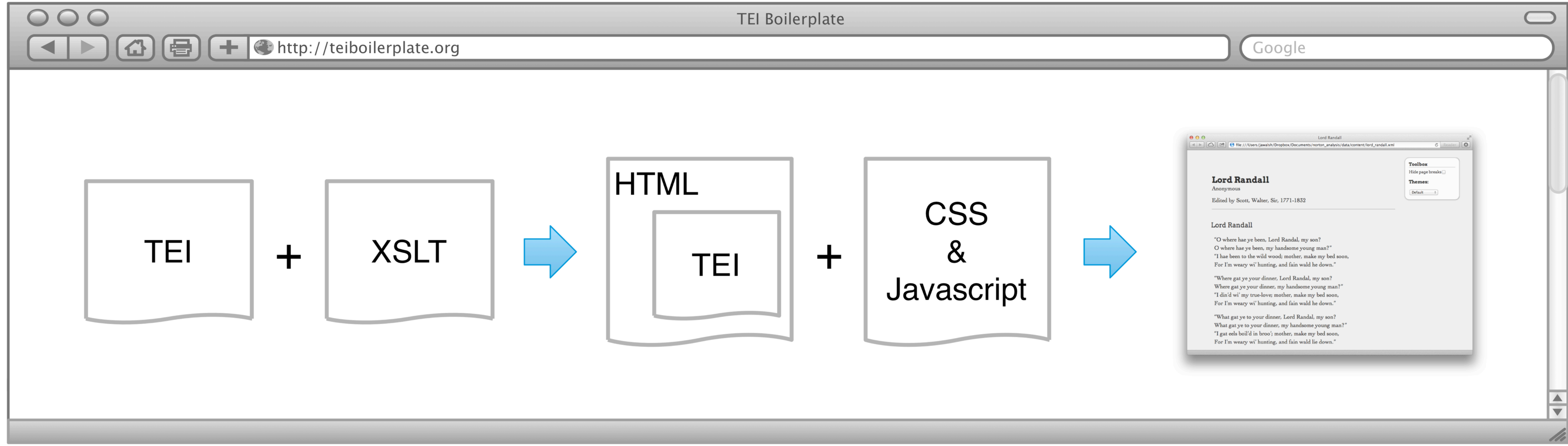


Figure 1. TEI Boilerplate Processing Flow. The TEI document is processed by a relatively compact and simple XSLT stylesheet. The result of the XSLT transformation is a sparse HTML5 document, within which is embedded, with minimal modifications, the full original TEI document. The resulting HTML/TEI document is styled and manipulated by CSS and Javascript. All of this is handled by the browser, without the need for any additional server software.

In teaching contexts, TEI Boilerplate is a useful solution. Students and scholars new to TEI are rightly excited and enthused as they encode their first documents and experience the expressive power of TEI markup. But too often that initial excitement and enthusiasm are dampened when new TEI users inquire about publishing their documents on the web and are confronted with sometimes overwhelming details about an unfamiliar programming language, XSLT, and other information about server-side processing and transformation engines. With TEI Boilerplate, users can simply add a single line of code to their documents to instruct modern browsers to produce formatted output of their TEI content. And that default TEI Boilerplate output may be further tweaked and customized using CSS.

Theoretical Motivations

The power of TEI lies in the richness and expressiveness of its vocabulary. But much of that richness and expressiveness is lost in the translation to HTML. TEI Boilerplate largely preserves the integrity of the TEI document. Because the TEI document is delivered directly to the browser, that source TEI document—unchanged by any XSLT transformation—can be easily accessed and saved to the desktop or mobile device.

Scholars labor over the intricate encoding of TEI documents, encoding that may represent sophisticated readings and analysis. But with the typical XSLT publishing solution, which transforms the TEI to HTML, much or all of the richness of the TEI content is lost. Furthermore, the presentation of the document is targeted at the HTML surrogate rather than the intricately encoded TEI document. This results in a conceptual disconnect between the design of the document and the original TEI encoding. By exposing the TEI directly to the browser, one may format the TEI directly, applying intentional design to a sophisticated model of a source document. If one wishes to manipulate the document using JavaScript, one may manipulate the TEI directly. Scholars are likely to be intimately familiar with their TEI documents, and their



Figure. Through a simple XSLT transformation, the original TEI document is embedded within an HTML5 document. The resulting HTML/TEI document can then be styled with CSS and manipulated with JavaScript as one would an more familiar HTML document.

engagement with and understanding of these documents may benefit when the formatting and processing is targeted at the TEI itself, rather than at an HTML surrogate.

TEI Boilerplate respects the integrity of the TEI document, and keeps the TEI document central throughout the publication process. TEI Boilerplate takes advantage of the separation of form and content inherent in XML, XSLT, CSS frameworks. However, like Liu (2004), Galey (2010), and others, the authors of TEI Boilerplate view that separation with suspicion. TEI Boilerplate attempts to weaken that separation of form and content in the typical TEI-to-web design and delivery model by largely removing the HTML layer, exposing the TEI-encoded text directly to the browser, and providing scholars with more immediate access to the readings, models, and analysis embedded in the TEI-encoded document.

How It Works

When a TEI document using TEI Boilerplate is accessed by a browser, the browser reads a line (the xml-stylesheet processing instruction) in the file that indicates that the XML should be processed by the TEI Boilerplate XSLT stylesheet. The browser then processes the TEI according to the rules of the stylesheet, which results in an HTML5 shell document within which the TEI document is embedded. A very small number of TEI elements and attributes (e.g. `<ptr>`, `<ref>`, `<figure>`, and `@rend`) are converted to their closest HTML equivalents to enable features such as clickable links, display of images, and support for inline CSS. The HTML shell contains a link to the TEI Boilerplate CSS file, which tells the browser how to style the full document.

Using it in Your Project

Download the TEI Boilerplate files, and host the `teibp` directory on a web server.

The simplest way to use TEI Boilerplate (TEIBP) is simply to add your TEI files to the `teibp/content` directory of TEI Boilerplate and include the following xml-stylesheet processing instruction at the top of your TEI documents, after the XML declaration and before the root `<TEI>` element:

```
<?xml-stylesheet type="text/xsl" href="teibp.xsl"?
```

You may then access your TEI files from a modern browser and see the resulting styled document.

TEI Boilerplate and CSS

TEI Boilerplate makes extensive use of CSS to generate styled documents in the browser. The CSS is drawn from three possible sources:

External css stylesheets:

1. `teibp.css` includes default styles for the HTML shell and TEI document.
2. `custom.css` is an empty stylesheet provided as a placeholder for user styles.
3. `<rendition>` elements with `@scheme="css"` within the TEI document are used to generate CSS declarations understood by the browser.
4. The global `@style` attribute may be used to include inline CSS in TEI documents.

Customizing the Look

TEI Boilerplate includes a default "theme" and some alternative themes. These are provided mostly as an illustration of TEI Boilerplate's capabilities. Users may modify these supplied themes in the following ways:

By supplying `<rendition>` elements with `@scheme="css"` within the `<tagsDecl>` section of your TEI document. TEI Boilerplate will automatically convert the `<rendition>` elements into CSS declarations in a `<style>` element of the HTML5 shell that surrounds the TEI document after the simple XSLT transformation. TEI elements that use the `@rendition` attribute to point to `<rendition>` styles will be styled accordingly. This mechanism is similar

to using HTML's `@class` to point to CSS classes. See the TEI Guidelines for more details on `<rendition>` and `@rendition`. The TEI Boilerplate TEI template (`teibp.xml`) also includes examples.

By adding custom CSS to the `custom.css` file located in the `teibp/css` directory. The HTML shell created by TEI Boilerplate includes a link to the `custom.css` file. This file is empty by default. You may add user styles to this file.

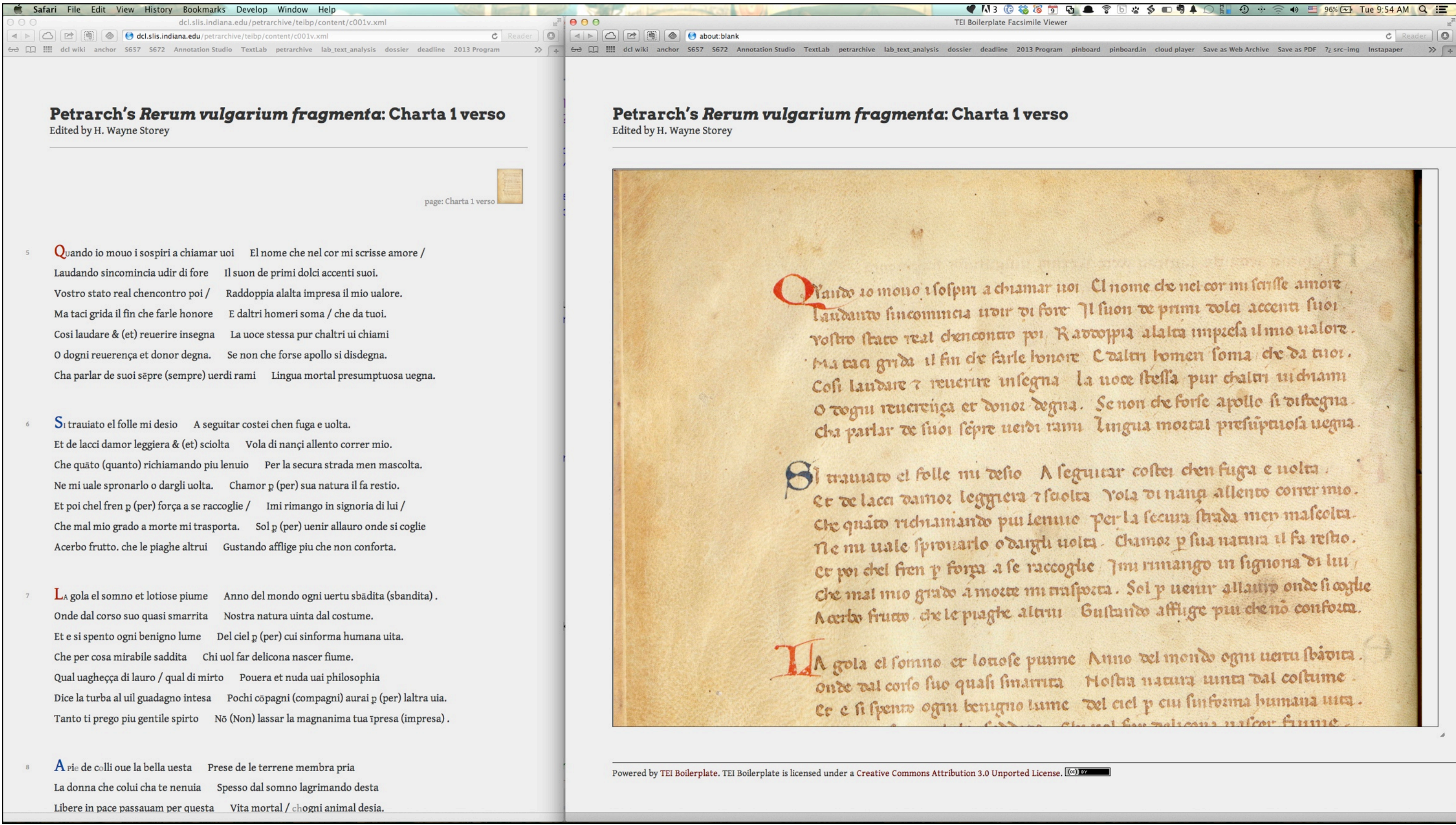


Figure. TEI Boilerplate supports display of facsimile page images recorded in the `@fac` attribute.

Facsimile Page Images

Since version 1.1, TEI Boilerplate includes a simple built-in page viewer for facsimile page images. The page viewer looks for `@fac` attributes on page break `<pb/>` tags, e.g.:

```
<pb n="42" fac=".. /images/page057.jpg" />
```

The TEI Boilerplate Template

An important component of TEI Boilerplate is a "boilerplate" TEI template (`teibp.xml`), a very simple TEI file, with some XML comments about where one might typically put basic information (e.g., title, author, bibliographic details of any source material). The most important features of the TEI Boilerplate template are:

The `xml-stylesheet` processing instructions with links to the TEI Boilerplate XSLT and CSS files.

A pre-defined set of `<rendition>` elements with generally useful styles for describing a source document or for use in a born-digital TEI document. For instance, the template includes styles for bold, italicized, underlined, and "normal" text; for left-aligned, right-aligned, centered, and fully justified text; for indentation of verse lines; block quotes; and so on.

Users of TEI Boilerplate may choose to use the styles provided in the template, and add to, modify, or replace them. As explained above, `<rendition>` elements, along with `@rendition` and `@rend` are all used by TEI Boilerplate to produced styled documents in the browser. Thus, TEI authors may achieve substantial customization by using these standard TEI features, without the need to edit any of the TEI Boilerplate code.

Downloads

Download the latest release of TEI Boilerplate from GitHub: <http://github.com/GrantLS/TEI-Boilerplate>.

People

The TEI Boilerplate team is John A. Walsh, Grant Simpson, and Saeed Moaddeli. Indiana University-Bloomington.

Please send general feedback to John Walsh at jawalsh@indiana.edu. Bug reports and feature requests may be submitted through GitHub: <http://github.com/GrantLS/TEI-Boilerplate/issues>.

References

Galey, A. (2010). The human presence in digital artifacts. In W. McCarty (Ed.), Text and genre in reconstruction: effects of digitization on ideas, behaviours, products, and institutions (pp. 93–117). Oxford: Open Book.

Liu, A. (2004). Transcendental data: Towards a cultural history and aesthetics of the new encoded discourse. *Critical Inquiry*, 31, 49–84.

Rahtz, S. (2006). Storage, retrieval, and rendering. In L. Burnard, K. O'Brien O'Keefe, & J. Unsworth (Eds.), *Electronic textual editing* (pp. 310–333). New York, NY: Modern Language Association of America.

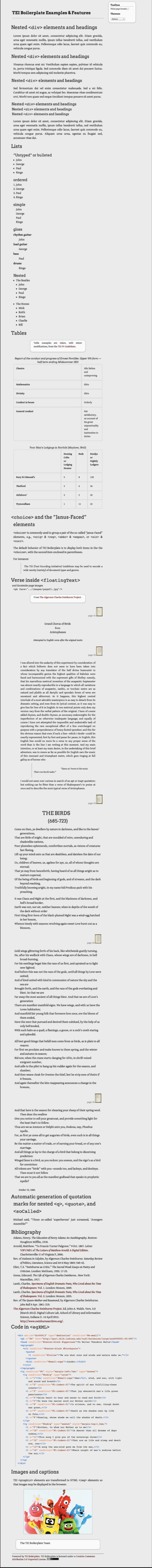


Figure. TEI Boilerplate demo page illustrates formatting of common TEI structures: lists, poetry, tables, etc.