Journal of the Text Encoding Initiative

Issue 2 | February 2012
Selected Papers from the 2010 TEI Conference

A TEI-based Application for Editing Manuscript Descriptions

Cristina Vertan and Stefanie Reimers

Electronic version
URL: http://journals.openedition.org/jtei/392
DOI: 10.4000/jtei.392
ISSN: 2162-5603

Publisher
TEI Consortium

Electronic reference
Cristina Vertan and Stefanie Reimers, « A TEI-based Application for Editing Manuscript Descriptions », *Journal of the Text Encoding Initiative* [Online], Issue 2 | February 2012, Online since 03 February 2012, connection on 30 April 2019. URL : http://journals.openedition.org/jtei/392 ; DOI : 10.4000/jtei.392

A TEI-based Application for Editing Manuscript Descriptions

Cristina Vertan and Stefanie Reimers

AUTHOR'S NOTE

This work was carried out within the DFG-funded Project "Teuchos: A Virtual Research Environment for Classical Philology". The authors would like to thank our colleagues who contributed valuable ideas to the project and spent time checking and correcting errors in annotations.

1. Introduction

1 The mass digitisation process that has taken place during the last 10 years has made valuable historical manuscripts available both to the broad public and to the research community. Digital libraries often integrate high-resolution viewers which allow codicological and paleographical analysis of manuscript onlines. Therefore, there is an increased demand that the results of such analysis (e.g., manuscript descriptions) also be available online. Manuscript descriptions are of particular importance as they often refer to other manuscripts, collections, or scribes. Therefore an adequate system for annotating content of manuscripts is needed.

2 The P5 Manuscript Description module[1] offers the entire apparatus necessary for content annotation of such documents. Since ENRICH[2] adopted it for encoding manuscripts, it has become a standard in the research community and in large European initiatives. For computer scientists and scholars trained in TEI, direct use of the standard is not problematic. For untrained users, the wide spectrum of possibilities in TEI-P5 frequently seems too complicated. Use of the module is time-consuming and error-prone, and even XML editors do not sufficiently simplify the workflow.

3 Within the Teuchos Group[3] at the University of Hamburg we developed a software package that integrates a graphical user interface (GUI) as well as modules for automatic annotation of manuscripts. The main advantage of such a system is that the user works entirely in plain text, not needing to learn markup, with the system only saving in TEI P5 once the manuscript description is complete. This paper presents our software package, which includes a module for generating P5-conformant manuscript descriptions as well as several modules for automatic deep annotation of the manuscript transcription at the phrase level.

2. User Scenario

4 Manuscript description is typically performed by codicologists in order to summarize the main physical features of a manuscript (type of binding, type and number of folios, type of material, etc.) and its transmission path (authors, place where the manuscript was found, persons who copied it from older manuscripts, etc.). A manuscript description is essentially metadata associated with a manuscript. Since the manuscript itself is usually difficult to access, the metadata is of particular importance for research.

5 The manuscript cataloguing guidelines provided by the DFG[4] provide a standard framework for description of ancient and medieval manuscripts in several sections corresponding to the manuscript type: there are different guidelines for ancient, medieval, and illuminated manuscripts. These DFG guidelines are completely conformant with the Manuscript Description module in P5, so TEI documents automatically conform to the DFG guidelines. This is of particular importance because TEI offers a straightforward mechanism for re-encoding pre-existing manuscript descriptions. Furthermore, XML offers also the possibility of linking a manuscript description with a digitized manuscript or with other manuscript descriptions. The more detailed the manuscript descriptions created, the larger are the possibilities for online research and semantic interconnection of manuscripts.

6 However, without a dedicated tool, a codicologist has to use an XML editor to annotate his text. He has to be aware of the correspondences between the TEI modules and the different sections of the DFG Guidelines. For a non-specialist, this is quite difficult and leads often to errors. Additionally these non-specialists tend to make a rather flat description and annotation in order to avoid a deep understanding of the Manuscript Description module, creating XML documents that are not as useful as they might be.

7 We developed a GUI which allows the user to concentrate on the codicological information. The interface contains tabs for the top-level components of the manuscript description. Each tab contains several fields with the corresponding subsections. The user inputs only text, and the TEI markup is generated in the background. At any time the user can interrupt his work, save, and reload the document to resume work where he left off.

3. System Description

3.1. The Annotation System

8 The application consists of two separate components: the GUI and the annotation tool. In figure 1 we present an overview of the system.

9 The user inserts his manuscript description into the GUI by filling out the fields corresponding to sections of the manuscript description, such as general information, watermarks, and physical description. Data is stored as a structured text file, and the GUI can load these structured text files in order to modify their contents later.

10 The structured text file serves as input for the annotation tool. The annotation of the manuscript description takes place in two steps. First, the text file is annotated by structure—i.e., the contents of each field are tagged with XML. In a second step, deeper TEI encoding is carried out, inserting links to objects within the XML document and to other documents. The final output is a TEI P5-conformant XML file.

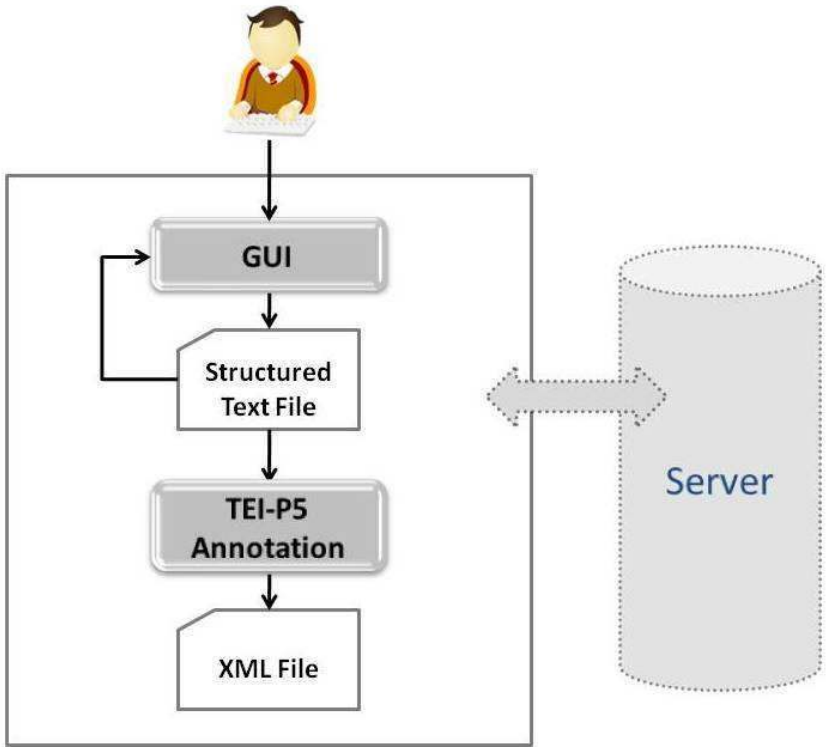

Figure 1: The annotation system

3.2. The Graphical User Interface

11 The GUI allows the user to record data in ten different sections. Most of them include subsections.

12 The Basic Data section asks for the name and place of the holding library, the signatory, editor, collection, number of lines, dimensions, material, date and editor of the edition. The indications of name, place, signatory and editor are obligatory. The Layer section consists of the subtopics layer such as “Greek Custos”, “Claimant”, “urther ayer ignatures”.

13 The Physical Description section includes form fields for information about foliation, lining, the cover, and remarks concerning the material. In the Annotation section, notes about annotations, empty folios, and appendices to the textual inventory can be entered.

14 Historical data is divided into two main sections. The first is about the origin, the original state, and historical notes of the text of the manuscript. The second describes

provenance, illumination, copyist and condition. Bibliographic data consists of facsimile information, catalogue data, text, and unclassified notes.

15 The Watermarks, Content and Reproductions sections provide possibilities for exactly the purpose mentioned in their titles and are not divided into subsections.

16 There is no predefined order for editing the sections. The user can switch between them as he likes. Data is stored in Unicode so that all special characters can be processed.

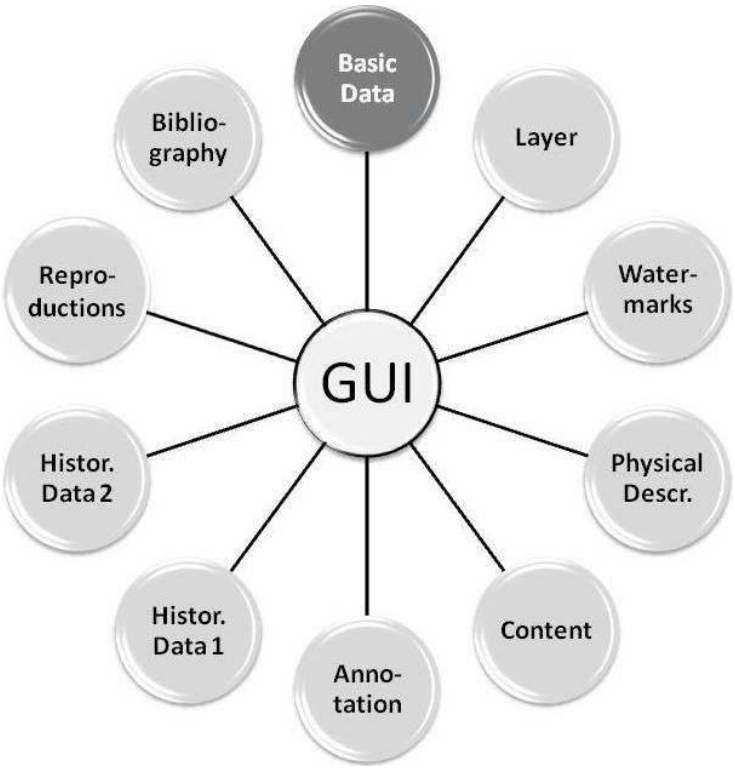

Figure 2: GUI sections

17 The screenshot of the editor in figure 3 shows that each section has a tab, allowing the user to easily navigate between sections. Each tab contains one or more fields corresponding (when applicable) to the subsections. The user can create more than one text box for each subsection and rearrange or delete them using the buttons to the right of each field. The contents of each field will be wrapped in a <p> element in the final XML.

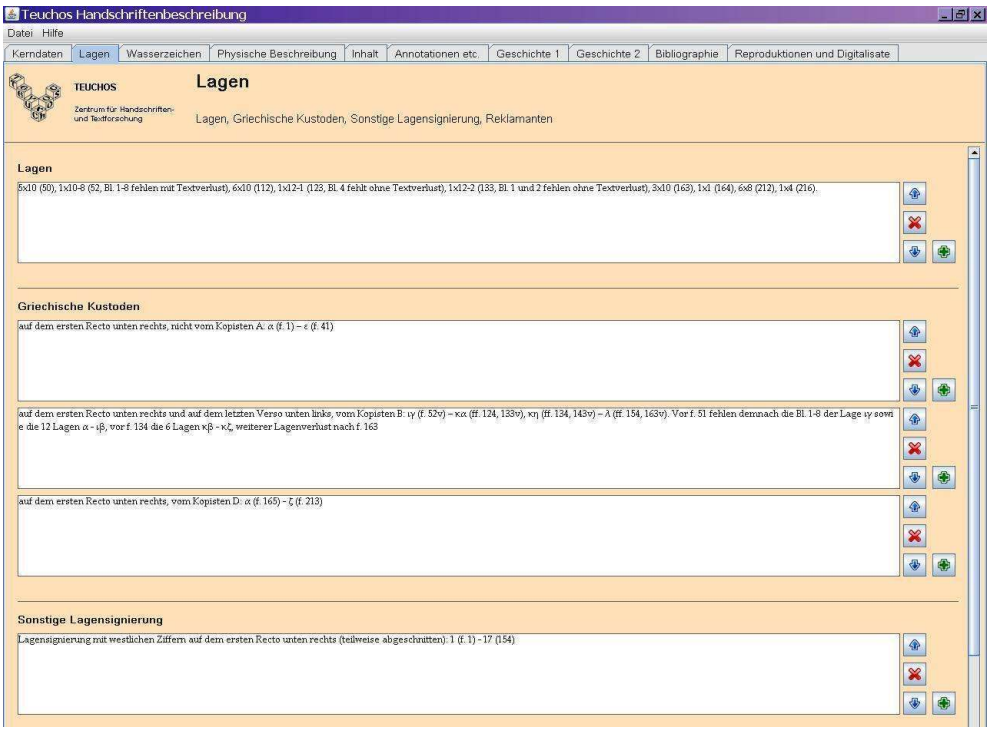

Figure 3: Screenshot of the user interface

Here is a sample manuscript description output by the GUI:

```
PARIS, BIBLIOTHÈQUE NATIONALE ANCIEN FONDS GREC

1856

15. Jh. 1. H.

Papier

mm 293x213

ff. VII, 138, V'

Lin. 26

Lagen.

7x10 (70), 1x10-2 (78, Bl. 3 und 8 fehlen mit Textverlust), 6x10 (138).

Wasserzeichen.

ff. 1-88, 89-91/ 96-98, 111/116, 122/125, 129-138: Krone,
entfernt ähnlich Br. 4708 (Bologna 1413), Pi. ohne Beleg.
ff. 93/94, 109-110/117-118, 113/114, 121/126: Dreiberg mit
Kreuz, sehr ähnlich Br. 11689 (Florenz 1411-1421, mit
Varianten Braunschweig 1412 und Pisa 1416)...
```

20 The DLG guidelines require a fixed notation at the top of the manuscript description:

- place and name of the library in the first row, divided by a comma;
- a blank line;
- dating, material, dimensions, and number of folios and lines.

21 Following this, each section is labelled with a title followed by a full stop (period), like each corresponding paragraph of the description. Sections are separated by blank lines. Optional fields don't appear in the text file.

22 The entire application is written in Javawith GPL-licensed open source. Its modular architecture allows for the addition of new sections.

23 Figure 4 shows the general architecture of the most important Java classes. The whole GUI implementation is mainly divided into two packages: *model* for the data model (on the left) and *gui* for the graphical realization (on the right). The class *Hauptseite* manages the instantiation of all components as well as the composition of the complete GUI. The interface *Konstanten* serves as storage for constants like names and titles, which are displayed on the user interface, or are internally used for functionality such as buttons, menu items etc.

24 Within the package *gui*, the class *Seite* provides all methods for the composition and functionality of every section tab—i.e., there are methods for creating text areas with integrated labels in addition to plus, minus, up, and down buttons. Additionally, it

implements the interface *Konstanten*. Each section of the manuscript description has a corresponding subclass of *Seite*, allowing for shared code among the sections of the bibliographic description.

25 The package *model* contains a data class for each corresponding section class in the package *gui*. If the user saves his manuscript description, the data entered by the user is transformed according to the corresponding model classes. Just as the package *gui* has a class containing shared code, all classes in the package *model* inherit from the central class *Daten_Seite*, which offers methods for converting section entries into the required structured output format for the text file.

26 The mediator class *Hauptseite*—together with some utility classes not illustrated in figure 4—concentrates the key functionalities of the interface. It includes the logic for the load, preview, save, open, and exit commands, plus a mechanism for checking that required fields have content.

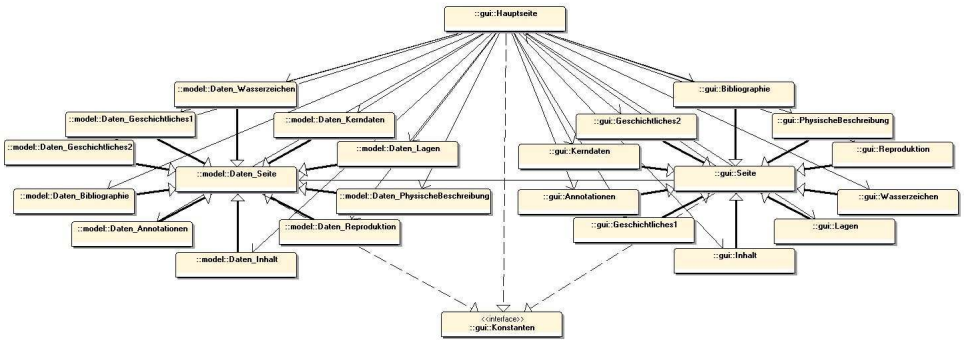

Figure 4: Overview of the most important Java classes of the GUI

4. Conclusions and Discussion

27 By using the annotation system and GUI, a user is able to create a manuscript description that is as granular as that allowed by the tabs and text fields of the GUI. While the manuscript is described at a fairly surface level, using mainly the top-level components of a manuscript description, the results are both interoperable with the DFG guidelines for manuscript description and can be directly uploaded in any platform without human correction.

28 Such a tool for rapid generation of machine-readable manuscript descriptions allows existing manuscript descriptions to be annotated in short time. So far we have annotated about 250 manuscript descriptions in this way.

29 Although our annotation system and GUI are at first glance similar to the application developed in the ENRICH project, there are several caveats:

- Our tool is designed particularly for classical philology, following the textual structure used in this research area;
- The user interface is language customizable; all elements related to the textual structure are stored in a separate Java interface. The GUI reads the Java interface;
- Being scaled for the particular needs of a classical philologist, our tool avoids parts of the TEI's Manuscript Description module which are not of interest to classical philology;
- The tool is written in Java SDK, is open source and GPL-licensed, and can be run locally on the client side.

30 Several separate modules can be run from the GUI in order to refine the annotation—i.e., to carry out a deeper annotation process. For the moment we implement functions that automatically recognize work titles, watermark motifs, authors, and references to certain folios. The automatic recognition of work titles, authors and watermark motifs rely on corresponding dictionaries; for the automatic recognition of references to folios we implemented a pattern-based approach. The precision rate at present is 80%. We have to mention that this precision is obtained in manuscript descriptions edited without the GUI. The current version of the GUI also includes a syntax component which checks text input against the implemented patterns.

31 For these pattern-based recognition of references to folios we first manually collected such references from about 70 manuscript descriptions. We cluster references in five clusters which are applied iteratively on each line of the input. The module works with either pure text or with TEI-encoded data. In order to be able to distinguish digits related to folios from other digits in the manuscript description, the module only recognizes sequences that start with "f." or "ff." So while a sequence like "ff 1, 22v, II, 45-67rv" will be recognized, an isolated sequence like "23-44" will be skipped.

32 We plan to implement additional modules for automatic annotation as well as improvement of the existing ones to make better use of machine learning techniques. Data already annotated using the tool are undergoing manual correction and will be used as training material.

NOTES

1. http://www.tei-c.org/release/doc/tei-p5-doc/en/html/MS.html

2. http://beta.manuscriptorium.com/apps/m-tool/m-tool.php

3. http://beta.teuchos.uni-hamburg.de

4. http://www.manuscripta-mediaevalia.de/hs/kataloge/HSKRICH.htm

ABSTRACTS

Within the Teuchos Group at the University of Hamburg we developed a software package that integrates a graphical user interface (GUI) as well as modules for automatic annotation of manuscripts. The main advantage of such a system is that the user works entirely in plain text, not needing to learn markup, with the system only saving in TEI P5 once the manuscript description is complete. This paper presents our software package, which includes a module for

generating P5-conformant manuscript descriptions as well as several modules for automatic deep annotation of the manuscript transcription at the phrase level.

INDEX

Keywords: codicology, graphical user interface, manuscript description

AUTHORS

CRISTINA VERTAN

Cristina Vertan is senior researcher at the University of Hamburg, Research Group "Computerphilologie". Her research interests include multlingual language technology, the semantic web, and digital humanities. She has lead several international research projects and has been working for the DFG-funded project TEUCHOS from 2007 to 2010.

STEFANIE REIMERS

Stefanie Reimers is a University of Hamburg computer science student with a focus on computational linguistics. She is currently working on her diploma thesis on ontology extraction. From 2009 to 2010, she was working for the DFG-funded project TEUCHOS.