Edition Production Technology: an Eclipse based platform for building Image-based Electronic Editions

Software demo

We demo the Edition Production Technology (EPT) of the Electronic Boethius[1] and ARCHway Projects[2] at the University of Kentucky, an integrated development environment for building Image-based Electronic Editions (IBEE)[3]. EPT is developed in Java on Eclipse platform and benefits from the Eclipse's plugins open architecture and portability (currently EPT runs on Windows XP, Linux, and Mac OS X).

The goal of EPT is to provide software support for building digital libraries of image-based cultural manuscripts: starting with images and text, EPT tools assist the electronic edition editor for creating electronic edition encodings, searching the electronic edition, linking text and images, and presenting the electronic edition (using filters and XSLT). A snapshot of EPT (for few component tools) is given in Figure 1.

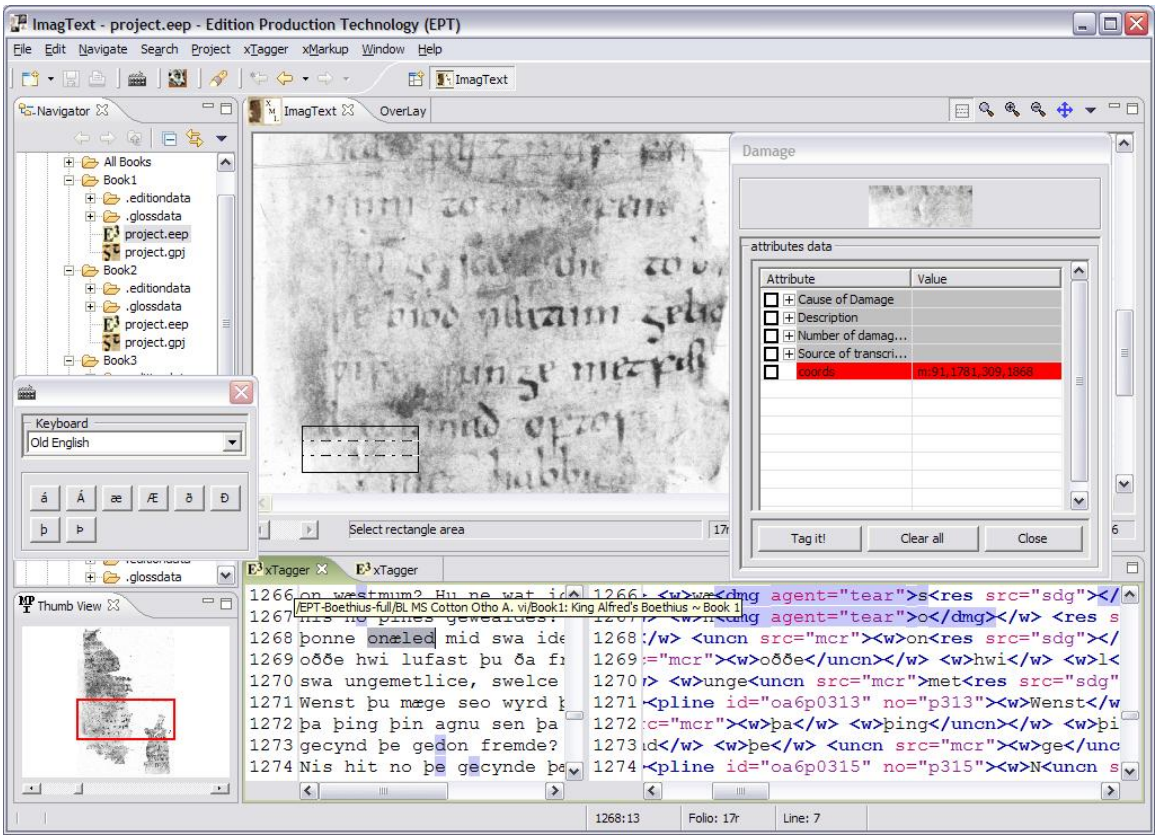

Figure 1. A snapshoot of EPT including few component tools: xTagger and ImagText, Keyboard, Markup interface, and an XML view

The component tools of the EPT are briefly presented in the following.

Editorial Tools on EPT Platform

The core component tools of the EPT are:

- ImagText and xTagger: an image-based XML editor for document centric XML. This tool is the core component of image-based text encodings: any markup that is to be introduced in text (using xTagger) is associated with an image region based on the user selection (using ImagText). In summary, tagging various features work as follows: select text and image (in any order), select the feature to be markup up (that is tag and attributes values) and insert markup (using a tag template interface). The xTagger makes sure that the new markup is well-formed and *potentially valid*[4]. Moreover, multiple markup DTDs are supported in the same edition.
- DucType: provides specialized interface for marking up paleographical information for manuscript letters.
- Overlay: provides images manipulation support. It is often the case that, for highly damaged manuscripts, one image (in normal lightening conditions) is not good enough for an editor to distinguish all preserved manuscript characters (cite Kiernan). This tool overlays two images of the same folio (using, for instance ultraviolet and normal lightening conditions) and allows changing transparency of the upper layer: a useful comparison can be performed.
- SaMod: a specialized tool for creating manuscript text collations with text from other sources. This tool assists the editor in finding differences and marking up these differences as variants of the original manuscript.
- Statistics: provides manuscript statistics (number of folios, lines per folio, characters per line, etc.). Based on these statistics the tool provides support for folio reconstruction: text splitting in lines and folios and virtual folio images build character by character using letters templates.
- Quires: a specialized interface for quires markup (insertion/deletion) and visualization.
- Advanced Search GUI: is a user friendly interface for searching the edition. It hides from the user the intricacies of the query language (an extension of XPath that supports multiple hierarchies).
- Datalayer: API for data access in EPT. Each tool requests and delivers the data through Datalayer API. The advantage is the interchangeability of data support back-end: a database, file system, remote server.
- Glossary: a data-centric XML editor for creating the glossary information for each edition word. It provides customizable templates for each part of speech and tools for saving the information in XML format (used later on for searching purposes) and HTML format (used for display glossed information).
- HTML browser: provides HTML display and, in general, browser support in EPT. The advantage of having a browser integrated in the platform is that the XSL transformations can be dynamically directed to browser.
- Keyboard panel: due to variety of text encoding projects, it might be the case that a configurable character toolbox is desirable.

In addition, EPT provides support for project management such as:

- Project wizard: initializes an electronic edition project. The input data consists in image files, text content (or partially encoded text) and one or more DTDs (EPT provides support for concurrent markup).
- Project properties editor: GUI for various settings related to the project, such as fonts, encoding, title, etc. Provide support for adding/removing images to a project and for customizing markup tags: grouping tags in meaningful use categories, assigning aliases to tags and attributes, adding/removing DTDs, adding text encodings to the current project by merging certain XML encodings of the project text with the current project encodings. This merging operation in conjunction with filter operation provides a basic concurrency control over the electronic edition: many editors (or students) may work on different text encodings (without changing the content) which eventually are to be merged into the same electronic edition.
- XML filter editor: a GUI to define encodings filter. The output of a filter can be used for visualization, XSL transformation, or data interchange.
- Letter Template: creates and maintains a repository of letter images for a manuscript. The letter images are then used by the DucType tool as base of comparison with any letter image in manuscript image.
- Extended XPath search: a search GUI using extended XPath language (an extension of XPath that applies to concurrent markup structures).

From the Eclipse platform, EPT inherits three important features for a project development: versioning control (CVS), automatic updates, and help content support. The first feature is to be used by editor to share work in progress projects or just as projects repository. The second feature is useful in the context of providing tools updates as well as bug fixes: an EPT use has just to check for updates and download them if available. While developing the tools the greatly valuate the automatic update feature as the programmers could provide new features and bug fixes while editors were working on electronic edition projects. Finally, Eclipse provides an open help architecture (and therefore in EPT), so that the application help information can be added independently of the application program.

Demo overview

We demo the EPT as an electronic edition by creating a project and going through the usual operations for preparing an image-based electronic edition: markup content (using only text projections or filtered XML views), automatic linking of images and text, and text updates. We demo the support for overlapping markup structures by adding/removing DTDs and markup encodings from external files. In addition we show how a project can be customized, starting from user interfaces (toolbox, fonts, encodings, etc.) and ending with markup customization: associating aliases to tag elements and attributes, group tag element by functionality, displaying status bar information based on XPath queries. We demonstrate that our document-centric XML editor (xTagger) can significantly simplify and speed up the encoding process. At any moment, the editor can search for the information, visualize the encodings using customizable filters, or change project properties. We demo the specialized GUI for marking up: DucType (and we show

how to customize the tool), Quires, and Overlay. Statistical information for the project encodings can be obtained dynamically and we show how this information can be used in folio reconstruction (text and image) for missing manuscript part. We also show manuscript text collation with other text source, using SaMod tool.
The demo also includes automatic generation of HTML content from edition data (glossaries, manuscript edition and manuscript transcription).

We emphasize during the demonstration how the Eclipse's open architecture turns to be an excellent platform choice for implementing the EPT.

References

[1] Kevin Kiernan and Dorothy C. Porter, "Edition Production Technology (EPT) and the *Electronic Boethius* Project," forthcoming in *DigiCULT* 2004/5.
[2] Kevin Kiernan et al., "Edition Production Technology (EPT) and the ARCHway Project," forthcoming in *DigiCULT* 2004.
[3] Kevin S. Kiernan, "Digital Facsimiles in Editing: Some Guidelines for Editors of Image-based Scholarly Editions," forthcoming in *Electronic Textual Editing*, a volume of essays jointly sponsored by the Modern Language Association and the TEI Consortium, funded by the Mellon Foundation, and co-edited by John Unsworth, Katherine O'Brien O'Keeffe, and Lou Burnard, 2005.
[4] Ionut Emil Iacob, Alex Dekhtyar, Michael I. Dekhtyar, "Checking Potential Validity of XML Documents," Proceedings, Seventh International Workshop on the Web and Databases, WebDB@SIGMOD/PODS, 2004.