Edition Production Technology: an *Eclipse*-Based Platform for Building Image-Based Electronic Editions

Ionut Emil Iacob *(ionut@ms.uky.edu)*

Department of Computer Science, University of Kentucky

Kevin Kiernan *(kiernan@uky.edu)*

Department of English, University of Kentucky

Alex Dekhtyar *(dekhtyar@cs.uky.edu)*

Department of Computer Science, University of Kentucky

We are developing the *Edition Production Technology* (*EPT*), an integrated development environment for building Image-based Electronic Editions (IBEE) (Kiernan 2005), through the *Electronic Boethius* (Kiernan and Porter 2005) and ARCHway Projects (Kiernan et al. 2004; Kiernan et al. 2005) at the University of Kentucky. We built the EPT using *Java*, and it operates through the *Eclipse* platform, benefiting from *Eclipse's* open architecture and portability. Currently the *EPT* runs on *Windows XP*, *Linux*, and *Mac OS X*.

The goal of the *EPT* is to provide software support for building image-based electronic editions of cultural manuscripts. Starting with images and text, the *EPT* enables the editor to create an electronic edition with complex, pervasive XML encodings, search the electronic edition, link text and images, and deploy the completed electronic edition using filters and XSLT.

A fully functional demo version of the *EPT* software suite for PC, including sample projects, is available for download at <http://rch01.rch.uky.edu/~ept/download>.

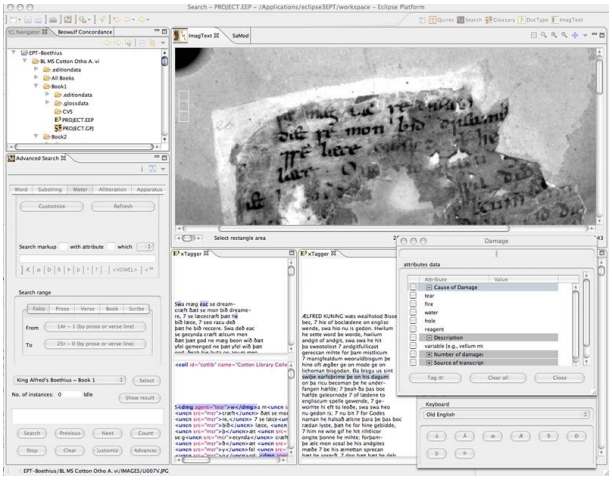

Figure 1. A snapshot of EPT illustrating image-based encoding through ImagText, xMarkup, and xTagger (including an XML view). The figure also shows the Keyboard panel and Search Tool.

Editorial Tools on EPT Platform

- Project wizard initializes an electronic edition project. The input data consists of image files, text content (or partially encoded text) and one or more DTDs (*EPT* provides support for concurrent markup using multiple DTDs).
- *xMarkup*, *xTagger* and *ImagText* form the core component for encoding image-based, document-centric XML, working together to link text and image. Through *xMarkup*, the editor selects edition markup through a series of simple, configurable templates. *xTagger* introduces markup into the text and provides filtered XML views, while *ImagText* associates that text section with the corresponding image region, selected by the editor. In summary, the tagging works as follows: the editor selects text and image (in any order), describes the manuscript or textual feature, choosing tags and attribute values through *xMarkup*, and inserts the markup. See Figure 1 for an illustration of the cooperation between *xMarkup*, *xTagger*, and *ImagText*. The *xTagger* ensures that the new markup is well-formed and potentially valid (Iacob, Dekhtyar, and Dekhtyar).
- *DucType* provides a specialized interface for describing individual manuscript letters. The editor configures *DucType* through the Letter Template, which also 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.
- *Overlay* provides image manipulation support. An editor may find that multiple images of the same folio are required for a complete view of the manuscript. Using this tool the editor lays one image over another image of the same folio (using, for instance, ultraviolet and normal lightening

conditions) and changes the transparency of the upper layer, enabling a useful comparison of the two images.

- *SaMod* is a specialized tool for creating manuscript text collations with text from multiple sources (for example, the same text found in different manuscripts). This tool recognizes differences between transcripts and marks up these differences as variants of the text the editor identifies as the base text.
- *StaTend*: Using a transcript marked with basic navigational markup – folio and folio line tags – this tool calculates manuscript statistical tendencies (number of folios, lines per folio, characters per line, etc.). Based on these statistical tendencies the tool reconstructs missing folios for which we can supply the text from another source, based on these statistics. The *StaTend* tool also includes functionality, called *RamSome*, for taking these textual 'virtual folios' and translating the text into image, built character by character using letters taken from the manuscript.
- *Quires* is a specialized interface for the edition and visualization of codicological markup. It allows the editor to build a virtual map of the physical object. We used this tool in the *Electronic Boethius* project to reconstruct the gatherings of a manuscript whose binding was destroyed by fire.
- The Search GUI is an interface for searching the edition. The editor can configure it to search any combination of XML markup, while hiding the intricacies of the query language (an extension of *XPath* that supports multiple hierarchies).
- Datalayer is the API for data access in *EPT*. Tools request and deliver edition data (image and text files, DTDs, etc.) through the Datalayer API, which can interface with a variety of data storage devices, whether a database, file system, or remote server.
- *Glossary* is a data-centric XML editor for creating a glossary including each word from the edition text. It automatically generates a complete word list from a transcript file encoded with basic formatting information (folio and folio line markup. The glossary links its entries to the text through the `<word>` tag – changes made within the edition text are automatically reflected in the glossary. It provides customizable templates for parts 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).
- The HTML browser provides HTML display and general browser support in *EPT*. Having a browser integrated in the platform enables the *EPT* to direct XSL transformations dynamically to the browser.
- The Keyboard panel enables the editor to configure keyboards containing special characters (Old English *æ*,*ð*, and *þ*, Greek characters, etc.).

In addition to editorial tools, the *EPT* provides support for project management such as: Project properties editor is a GUI for various settings related to the project, such as fonts, encoding, title, etc. It provides support for adding and removing project images and for customizing markup tags, grouping tags in meaningful use categories, assigning aliases to tags and attributes, and adding and removing DTDs from a project. XML filter allows the editor to create encoding filters for viewing different combinations of elements from the entire set. The output of a filter can be used for visualization, XSL transformation, or data interchange. Extended XPath search is 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 project development: versioning control (CVS), automatic updates, and help content support. The editing team uses CVS to share project work-in-progress and as projects repository. Updates are useful for providing tools updates as well as bug fixes: an *EPT* user need only check for updates and download them if available. Finally, the open help architecture enables the editor to create and use help files in such a way that the application help information is added independently of the application program.

Demo overview

Our demonstration will begin with examples of the most basic *EPT* functionality, and depending on time we will demonstrate any tool or function. We will begin by creating a project and going through the usual operations for preparing an image-based electronic edition: content markup (using only text projections or filtered XML views), automatic linking of images and text, and text updates. We will demonstrate that our document-centric XML editor (*xTagger*) can significantly simplify and speed up the encoding process. The editor can search for the information, visualize the encodings using customizable filters, or change project properties at any point in the editorial process. We will demonstrate the support for overlapping markup structures by adding/removing DTDs and markup encodings from external files. Depending on the interests of the audience, we can also show how a project can be customized, starting with user interfaces (toolbox, fonts, encodings, etc.) and ending with markup customization: associating aliases to tag elements and attributes, grouping tag elements by functionality, and displaying status bar information based on XPath queries. We will also be prepared to demonstrate *Quires*, *Overlay*, and *DucType*, and show how to customize *DucType*. Statistical information for the project encodings can be obtained dynamically and we can

show how this information can be used in folio reconstruction (text and image) for missing manuscript part. We can also demonstrate *SaMod*, showing how it collates several different texts.

The demo may also include 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 is an excellent platform choice for implementing the *EPT*.

Bibliography

Iacob, Ionut Emil, Alex Dekhtyar, and Michael I. Dekhtyar. "Checking Potential Validity of XML Documents." *Proceedings, Seventh International Workshop on the Web and Databases, WebDB@SIGMOD/PODS.* 2004. 91-96.

Kiernan, Kevin S. "Digital Facsimiles in Editing: Some Guidelines for Editors of Image-based Scholarly Editions." *Electronic Textual Editing.* Forthcoming. 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.

Kiernan, Kevin S., Alex Dekhtyar, Jurek Jaromczyk, Dorothy C. Porter, and Ionut Emil Iacob. "Edition Production Technology (EPT) and the ARCHway Project." *DigiCULT.Info* (August 2004): 36-38.

Kiernan, Kevin S., Jurek Jaromczyk, Alex Dekhtyar, Dorothy C. Porter, Kenneth Hawley, Sandeep Bodapati, and Ionut Emil Iacob. "The ARCHway Project: Architecture for Research in Computing for Humanities through Research, Teaching, and Learning." *Literary and Linguistic Computing* (Forthcoming). Special issue, papers from Joint International Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing, 2003

Kiernan, Kevin S., and Dorothy C. Porter. "Edition Production Technology (EPT) and the Electronic Boethius Project." *DigiCULT* (Forthcoming).