The Versioning Machine 3.0: Lessons in Open Source Software [Re]Development

Susan Schreibman *(sschreib@umd.edu)*
University of Maryland Libraries
Ann Hanlon *(ahanlon@umd.edu)*
University of Maryland Libraries
Sean Daugherty *(seancdaug@umd.edu)*
University of Maryland Libraries
Tony Ross *(tonyross@umd.edu)*
University of Maryland Libraries

The Versioning Machine made its debut at the 2002 ALLC/ACH Conference in Tübingen. It is a software tool for displaying and comparing multiple versions of texts designed by a team of programmers, designers, and literary scholars. A primary goal of the software was to create a display environment which provides for features traditionally found in codex-based critical editions – such as annotation and introductory material – that also takes advantage of opportunities offered by editing and displaying witnesses in an electronic environment.

The VM was designed as open source software that would allow literary editors to concentrate on editing and displaying multiple versions of text using the Text Encoding Initiative (TEI), rather than expecting them to build a custom environment. The developers deliberately built the tool with JavaScript, CSS, and XSLT, as these technologies are presumably within the reach of a humanities scholar if she wanted to alter the environment.

The Versioning Machine is now in its third iteration, and has been demoed at various humanities computing conferences over the years. This poster session will not focus on the VM as a tool per se, but will focus on the issues and lessons learned developing open source software for the humanities. As this tool was developed for a very specific goal for a potentially small community of users, this poster will focus on the issues raised by developing software for an emerging practice, the difficulties posed by changing technologies, and the issues raised by moving from fairly standard book-based presentation paradigms for scholarly editions, to the relative vacuum of agreed-upon conventions for web-based editions

Developing Software for an Emerging Practice

At any gathering of digital humanists, there is a plea for software designed specifically for humanities applications. *The Versioning Machine* was developed to fill a fairly narrow niche. As such, its development has had as a central focus the exploration of the possibilities, as well as the limitations, of electronic editions that focus on the presentation of multiple witnesses.

For the first time in the software's development, rigorous and methodical usability testing will be carried out by a team member not directly concerned with software development. This testing will not only assess user-friendliness, accessibility, performance, and overall structure, but will also investigate the larger issue of whether *The Versioning Machine* functions as an asset to the community of scholars it is meant to serve. The test participants will be comprised of 10-15 literary scholars and textual editors. They will be given a series of tasks to perform and will be asked to comment on what they are doing or trying to do. They will also be asked to comment on the overall difficulty or ease of use, as well as the degree to which *The Versioning Machine* has or could enhance their scholarship. Because many of the participants will not be on-site, a web-based survey will be used to capture their impressions. Additionally, a number of participants will be tested on site so that our observations can be compared with the self-reported experiences. The resulting findings will be used as a framework to make further changes to the software.

Difficulties Posed by Changing Technologies

Persistent issues frequently faced in web-based open source tools development include shifting platforms, technologies, and standards on which software is constructed; personnel changes within the development team; and finding enough time to not only do software development, but create the documentation and procedures which allow others access. The difficulties of attempting to adhere to standards, while ensuring cross-browser compatibility, is one issue that will be addressed in the poster as representative of these issues.

During the early stages of the tool's development, it was not possible to replicate the functionality available for browsers on the PC on the Mac OS operating systems. It was decided to go live with limited browser support rather delay the software further. By 2.0, many of the compatibility issues had been worked out, with support for Mozilla-based browsers and Apple's Safari. 3.0 focused both on expanding features of *The*

Versioning Machine as well as addressing several major compatibility issues. This involved a significant reworking of both the XSLT stylesheet and the CSS style rules that power the tool. Earlier versions were lacking in compliance with W3C standards, which accounted for a large part of the cross-browser compatibility issues. A change in personnel meant that several of the people involved in versions 1.0 and 2.0 were no longer on the project team. A lack of both comprehensive documentation and time ensured that, even in its current form, *The Versioning Machine* would not correctly validate as standards-compliant. Thus a pragmatic approach was adopted in which the most obvious discrepancies were addressed, while ensuring that all new additions were coded with respect to the relevant W3C standards.

Of the newly added features, the most significant was the introduction of optional line numbering, drawn from the TEI markup of the document. This proved one of the more difficult features to implement, in large part because of the different ways that various browsers implement CSS. A presentation that looked acceptable in Internet Explorer, for example, did not look acceptable in Firefox or Safari. Toggling line numbers on and off presented another problem. In this case, the JavaScript required worked differently on all three browsers. This was one example of a problem that strict adherence to W3C standards could not solve. There was more than one standards-compliant way of achieving similar functionality, but the difficulty was in finding one method that worked consistently across multiple platforms.

The ability to directly modify the version 2.0 source code certainly expedited the design process for version 3.0. On the other hand, it also meant that we were forced to address many of the shortcomings of the earlier version as well. Thankfully, Amit Kumar, who had worked with the earlier version and implemented the majority of the version 3.0 changes, was able to provide insight into the workings of the application, and to bring his experience working with it to bear on the new compatibility challenges.

As part of the 3.0 redevelopment, a thorough re-evaluation of the interface was undertaken which would move it toward a more finished, professional look. A common pitfall of interface design is that the person designing the interface is often so familiar with the product that they are unable to view it from a new user's perspective. Bringing in a new designer for release 3.0 meant not only a fresh approach to the visual side, but also a new set of eyes completely unfamiliar with *The Versioning Machine*.

Other than the addition of a "backup" menu to the footer, the interface elements of 3.0 are not significantly different from release 2.0 — however, their arrangement has been streamlined for better usability. Even those changes that appear purely cosmetic have a level of thought behind them. For example, the text in the upper left that said "Versioning Machine" now spells out the release version of the software. To the left of that a tagline ("A Tool for Displaying & Comparing Different Versions of Literary Texts") has been added to spell out in broad strokes the software's intended purpose.

What Does an Electronic Scholarly Edition Look Like?

Lastly, this poster will address some of the theoretical issues that the developers of *The Versioning Machine* have faced in designing an environment to present a web-based scholarly edition. There is today, ten years after the development of the World Wide Web, little consensus within the editing community about the features and standards that should be required, or at least desired, in these editions. The user testing on *The Versioning Machine* to be presented at *Digital Humanities 2007*, the ongoing development of the software to keep apace with standards, and changing user expectations, will contribute a small piece to that dialogue.

Bibliography

Better Desktop. Accessed 2006-11-12. <http://www.betterdesktop.org>

Burstein, Cari D. "Viewable with Any Browser Campaign." 2006. Accessed 2006-11-12. <http://www.anybrowser.org/campaign/index.html>

Cockburn, Craig . "Cross Browser Compatibility and Website Design." 2005. <http://www.siliconglen.com/usability/browsers.html>

Kaufman, Joshua. "Practical Usability Testing." 2006. Accessed 2006-11-12. <http://www.digital-web.com/articles/practical_usability_testing/>

Levi, Michael D., and Frederick G. Conrad. "Usability Testing of World Wide Web Sites." U.S. Department of Labor: Office of Survey Methods Research, 2002. Accessed 2006-11-08. <http://stats.bls.gov/ore/htm_papers/st960150.htm>

Nielsen, Jakob. "'Alertbox: Why You Only Need to Test with 5 Users." 2000. Accessed 2006-11-08. <http://www.useit.com/alertbox/20000319.html>

Olson, George. "The State of the Web: Browser Incompabilities Undermine Web's Foundations." 2000. Accessed 2006-11-12. <http://www.webstandards.org/press/releases/2000-state-web>

Petersen, Jeremy. "A Barebones Guide to Usability Testing." . Accessed 2006-11-08. <http://javaboutique.internet.com/articles/Usability/index-3.html>

Schreibman, Susan. "Computer-mediated Texts and Textuality: Theory and Practice." *Computers and the Humanities* 36.2 (2002): 283-293.

Schreibman, Susan. "The Text Ported." *Literary & Linguistic Computing* 17.1 (2002): 77-87.

Smith, David. "Textual Variation and Version Control in the TEI." *Computers and the Humanities* 33.1-2 (1999): 103-112.

Sperberg-McQueen, C. M., and Lou Burnard, eds. *TEI P4: Guidelines for Electronic Text Encoding and Interchange, XML-Compatible Edition.* Oxford, UK: TEI Consortium, 2002.