Creating scholarly tools and resources for the digital ecosystem: Building connections in the Zotero project

Introduction

What does “Web 2.0” mean for universities, libraries, and museums? For most of my colleagues in the professoriate, the mere mention of the trendy term brings to mind a wave of wildly popular sites that have come to haunt the classroom: Facebook, MySpace, Wikipedia. “Social” media, “tagging,” and something called “crowdsourcing” threaten to crowd out serious scholarly pursuits. Unsurprisingly, Web 2.0 has seemed to many faculty, librarians, and curators a barren ground for scholarly presentation and communication, and they have viewed it with a skeptical eye.

Yet if we can move beyond the distractions of buzzwords a more simple description of Web 2.0 is precisely what the term suggests: following an initial phase of using the Web we are finally starting to understand what the medium does well — namely, connect disparate resources and people into a symbiotic whole — and we are now applying those lessons to new resources and tools. If Web 1.0 was about display and access — the process of getting documents online so people could view them anywhere and anytime — Web 2.0 is about the networked communication of the Web, or to use the title of this year’s WebWise Conference: the power of community.

So what should Web 2.0 mean for universities, libraries, and museums? The experience of the Zotero Project at the Center for History and New Media (CHNM) at George Mason University, an open source initiative to provide a high-quality bibliographic, research, and note-taking tool that now serves over a million people in 35 languages, might be helpful as a guide [1]. Although success has many fathers — most importantly the core team led by CHNM’s Director of Research Projects Sean Takats and Lead Developer Dan Stillman — I believe one critical element of the Zotero Project has been the way in which we think of any scholarly tool or resource as existing in an interconnected digital ecosystem — that is, the way in which the Project looks beyond itself.

In a Web 2.0 environment, no application or repository should be an island; to live in this digital realm, applications and repositories must connect with each other, must be able to give to and take from other applications and repositories, and must be able to leverage the combined knowledge and actions of scholars from around the world. A utopian sounding enterprise, to be sure. Yet thinking in this way has serious pragmatic implications for the way in which all of those who serve the scholarly enterprise should orient their online creations for maximal utility.

Breaking Down the Boundaries Between Applications

Research site

Two decades of staring at a screen on which Microsoft Word has been the standard interface for scholarship has clouded our sense of what an academic application might be. Most academics still think of an application as something they download from the Internet or install from a disk, that has its own icon on their computer’s desktop, and its own window once it has launched. The result of this vision has been a balkanization of the research environment into multiple, generally unrelated windows, such as Word, a Web browser, a standalone citation tool like EndNote, and notes written in various digital or even analog forms (see Figure 1). Even so-called Web applications — that is, applications that run on a Web server and that are accessed via the browser — have tended to be disparate silos.
At CHNM, we were not unaffected by this traditional view of applications. Zotero is the result of many years of our experimentation with tool-making. Specifically, it descends directly from two other applications: Scribe, a free EndNote replacement created by Elena Razlogova, and Web Scrapbook, a Web application that I initially coded with Simon Kornblith (who is now a developer on the Zotero Project). Scribe was a stand-alone, downloadable application that managed citations and notes well; Web Scrapbook of course could only be accessed online through the browser and did a good job capturing Web pages, texts, links, and images from that medium.

Both Scribe and Web Scrapbook were decent (if not explosive) successes with thousands of users, but with deficiencies that hindered their usefulness in an era of Web 2.0. As the online resources available from the library and museum community grew exponentially over the last decade, it became clear to us (and everyone else) that the Web browser had become the location for research. It thus no longer made sense to have a research tool exist as a separate application, as Scribe and EndNote did; at the same time, Web-only applications such as Web Scrapbook or RefWorks had limitations, such as the necessity that researchers log into a Web site to access their research (not so great for a historian in an archive without Wi-Fi) and a "walled garden" approach that focused more on retaining objects within the Web application than pushing them out to other locations and software.

In short, we wanted the best of both worlds: the best parts of stand-alone applications and Web applications. We envisioned a tool that lived in the browser and that was very smart about what was going on in the browser, including the recognition of scholarly metadata and objects, and that could interact with elements both on the desktop, such as word processors or other programs, and via standards, services, and communication protocols to other tools and resources across the Web.

One of the most important design decisions we made in the Zotero Project was therefore to focus on the tool as being embedded in the browser (but not as a Web site) — to view the browser not as a window into an array of documents (Web 1.0) but as a location for the exchange of information among scholars and repositories (Web 2.0). The Firefox browser made this all possible since from the start it was geared toward extensibility — over 5,000 extensions attest to the power of that model. Moreover, Firefox extensions, unlike Web applications, can run offline as well as online.

Existing on top of the browser as an extension (rather than within it as a Web application) means that unlike traditional stand-alone applications, Zotero can "read" what is going on in the browser and act on that information. The signature feature of Zotero is indeed its ability to sense scholarly objects in the Web browser — books, articles, manuscripts, and many other forms — and offer to save them to its iTunes-like research interface that slides up from the bottom of the browser window.
This technology of connecting with the “natural scholarly resources” of the digital ecosystem is what we call “translators” — little bits of code that are created by our main development team or, increasingly, from others from outside the Project, to let Zotero know when it is looking at a scholarly object and which actions can be taken on that object. These translators can focus on particular sites, technologies, or software; together they mean that Zotero is compatible with tens of thousands of Web sites from libraries, museums, and other repositories of scholarly materials, and this number grows daily as more translators are added.

Building Bridges, Not Islands

Although part of the success of Zotero probably comes from its easy-to-use interface, it is this unglamorous work of ensuring the tool connects with the digital ecosystem that has been its hidden strength. Translators are the most obvious way that Zotero integrates tightly with the resources of scholarship, but they are only one of the many ways it does so. Behind the scenes the Zotero Project has paid special attention to application programming interfaces (APIs), Web services, standards, and formats for exchanging digital information.

From its inception the Project has had an API so that others might build on top of the tool or connect it to other tools or resources without our intervention. Since the 1960s, computer scientists have used APIs to provide others with robust, direct access to their databases and digital tools. Access via APIs is generally far more powerful than simple Web-based access; indeed APIs are one of defining elements of Web 2.0. APIs enable outside users to develop their own tools or information resources based on the work of others; an API turns a tool or resource into a service that can used to combine and manipulate (in the trendy parlance, “mash-up”) digital objects and data in a free-form and potent way.

The Zotero API allows for communication between the tool and other tools and resources, as well as for enhancements to Zotero by third-party developers. Already software developers have created over a dozen extensions to Zotero to manipulate or reformat personal collections. For example, one popular extension, Vertov, annotates video and audio that has been saved into Zotero. In turn, Zotero takes advantage of external APIs, such as Amazon’s Web services, to help complete partial metadata for some records.

The Zotero Project has also paid attention to supporting many interchange formats. To the uninitiated, the acronym salad of Zotero’s import/export menus and native recognition tools seems overdone — RDF, XML, MODS, MARC, uAPI, COinS, BibTeX, RIS, and many others show up in various contexts — but they mean that Zotero can talk to countless other services and software. Making tools aware of existing standards and other widespread technologies can pay serious dividends, since the tool will end up being more useful — in a sense, extended and placed into a networked community — without the tool-builder taking any action.

A good recent example of this comes from the interaction of Zotero with OCLC’s WorldCat site. From its inception Zotero had a translator for the main WorldCat index, a global catalog of library collections. In addition, we had a generic translator to recognize bibliographic references embedded in Web pages using COinS (ContextObjects in Spans) tags. This past Spring, without our intervention, OCLC added COinS to a new service, WorldCat Lists, that provides user-created bibliographies on various topics. Without CHNM writing a translator for WorldCat Lists the site was instantly Zotero-compatible because both the service and the tool speak the same language (see Figure 3). Other institutions and individuals who maintain scholarly resources can similarly make their site Zotero-compatible by adhering to one of an array of supported standards.

Through such standards, tools and resources are linked together in a way that enhances all participating software and repositories, and that encourages further participation through network effects.

Figure 3: Since WorldCat has added COinS tags to their Lists service, the site has automatically become Zotero-compatible.
Perhaps the most surprising and welcome development of Zotero’s existence as a platform that is both open source and open to external connections and intervention has been the rapid growth in non–English versions of the software. Over the past two years avid Zotero users from around the world have submitted 35 language “localizations” of the Zotero interface, helping boost the adoption of the tool in over a hundred countries. (I refrain from trendily calling this crowdsourcing, but that’s what it is.) To enable this multilingual environment all we had to do was set up the tool to have customizable strings — that is, not hard-code into the software English words — and open this customization to the community. Again here, thinking about one’s tool or resource as existing in a global networked environment has paid major dividends.

Conclusion

Thinking of a tool like Zotero in this networked way makes Web 2.0 tool-building not dissimilar from traditional scholarship. True scholarship does not sit by itself, and is worthless as a disconnected piece of writing; being part of a network of thought and publication is what makes all scholarship active and relevant. We are taught in graduate school to review the existing literature, use citations well, and situate our argument within an interrelated constellation of ideas and sources. Creating scholarly software in the digital ecosystem should be similarly integrative.

About the author

Daniel J. Cohen is an Associate Professor of History and the Director of the Center for History and New Media at George Mason University. He is the co-author, with Roy Rosenzweig, of Digital History: A Guide to Gathering, Preserving, and Presenting the Past on the Web (University of Pennsylvania Press, 2005), author of Equations from God: Pure Mathematics and Victorian Faith (Johns Hopkins University Press, 2007), and has published articles and book chapters on the history of mathematics and religion, the teaching of history, and the future of history in a digital age in journals such as the Journal of American History, the Chronicle of Higher Education, and Rethinking History. He is an inaugural recipient of the American Council of Learned Societies’ Digital Innovation Fellowship. At the Center for History and New Media, Dan has co-directed, among other projects, the September 11 Digital Archive and Echo, and has developed software for scholars, teachers, and students, including the popular Zotero research tool. He received his bachelor’s degree from Princeton, his master’s from Harvard, and his doctorate from Yale.

Blog: http://www.dancohen.org
E–mail: dcohen [at] gmu [dot] edu

Notes

1. Zotero: http://www.zotero.org. The Zotero Project is generously funded by the Institute of Museum and Library Services, the Andrew W. Mellon Foundation, and the Alfred P. Sloan Foundation.

Editorial history

Paper received 21 July 2008.