Abstract

Purpose – The paper aims to describe a proof of concept web application designed to allow users to search for library materials with geographic subject headings using Google Maps as the primary interface for navigation. The purpose of the paper is to describe the development of an innovative tool that one library has created to provide users with a new way to access bibliographic records.

Design/methodology/approach – The approach taken is descriptive, with the development process for the proof of concept laid out in detail and placed it within the broader contexts of web application development at the host library and the emergence of Web 2.0 tools.

Findings – The paper shows that unique and valuable new methods of accessing bibliographic data can be created through the use of application programming interfaces (APIs), one of the more powerful tools available to web application developers today.

Research limitations/implications – Challenges of working with geographic information in subject headings of bibliographic records are discussed, and potential methods for dealing with these challenges are described.

Originality/value – The paper shows that it is possible to use APIs provided by large internet entities such as Google to create map based navigational tools for accessing bibliographic information. It also shows the value of allowing library systems staff to explore new technologies.

Keywords Information retrieval, Bibliographic systems, Libraries, Geography

Paper type Conceptual paper

One of the more interesting developments in what has come to be referred to as Web 2.0 has been the release of publicly-accessible application programming interfaces (APIs) by large internet entities like Amazon and Google. Libraries were quick to see the potential of these APIs, and were certainly among the first to reap the benefits of access to Amazon’s data. Amazon book covers found their way into library catalogue result lists as early as 2003[1], and soon after the release of the Google Maps API in early 2005, libraries were making use of it on their branch location pages.

Having followed the emerging body of literature encouraging libraries to investigate and participate in Web 2.0 technologies (Levine and Stephens, 2005), in May 2005 systems staff at the Kingston Frontenac Public Library (KFPL) decided to make use of the Amazon API in an online request for purchase (RFP) web application being developed in house to replace a paper-based system. This RFP application allowed users to perform a simple search in Amazon for the title they were requesting, instead of requiring them to fill in close to a dozen fields of bibliographic data in a standard web form. By using the API, they were able to present Amazon search results to users directly within the library’s web site with a button beside each result that the user could click on to request that a copy of the title be purchased (see Figures 1-3). Not only did this result in greater ease of use for users, it provided collection development staff who process these requests with more accurate information, which has helped to
Figure 1. RFP application screen requiring user login

Figure 2. RFP application screen displaying user information and Amazon search box
reduce the turn around time for RFPs. Development of KFPL’s RFP application was
done using Python and the Zope application development platform. With the Python
code needed to access the Amazon API readily available on the internet from web sites
like O’Reilly’s (Bausch, n.d.), adding this functionality to KFPL’s RFP application did
not substantially increase the time it took to complete development of the project. The
application was released in early 2006, and is used to process approximately 700 RFPs
per month, showing a high level of adoption by library users.

Bolstered by early success in the development process for our RFP application,
systems staff decided to investigate Google Maps as well. The Google Maps API is well
documented, and in May 2006 using examples provided in the documentation they
were able to create interactive maps of the library system’s catchment area with
markers over each branch location that display hours and contact information when
clicked. Although use of the Google Maps API has become common and tools have
proliferated across the internet to aid users in the creation of their own Google Maps
mashups[2], this early experience seemed close to magic and raised questions of how
the API could be used in other interesting ways.

One of the earlier and more successful Google Maps mashups was HousingMaps[3],
which allows users to browse real estate listings from the popular Craigslist classified
ad web site[4] using Google Maps as the primary tool for navigation. The map in this
mashup provides a powerful and intuitive interface for real estate searchers, and soon
after it was posted the web site was held up as the very embodiment of the future of the
internet (Singel, 2005). It was this mashup that provided staff at KFPL with the
inspiration to attempt the creation of a library mashup, mixing Google Maps with
bibliographic data from the library’s catalogue.
With a substantial portion of the metadata in bibliographic records referring to geographical locations, the idea of being able to click on a point in a map to retrieve a set of titles related to that location seemed useful, and at first blush even simple to implement. Unfortunately, despite the potential of the idea, the library’s development priorities lay elsewhere. As a result, the project was undertaken by a single staff member at KFPL as a personal challenge, but also as a proof of concept with the ultimate goal of making it available on the library’s public web site.

As the project was developed almost entirely in the free time of the database librarian, it was approached in an *ad hoc* manner. The first stage of development consisted of conceptualising the final product and identifying the technical hurdles that would have to be crossed in order to create it. At its simplest, the concept consisted of a web site modelled on HousingMaps, with a Google Map on the left side of the screen and bibliographic records loading on the right side.

The HousingMaps web site and other Google Maps mashups use AJAX[5] to present these search results to users in a dynamic fashion without requiring a page to be reloaded. KFPL had made use of Mike West’s DataRequestor JavaScript wrapper[6] in an earlier project, in order to load new data into one portion of the screen without reloading the page, making this an obvious choice of tools to perform this task.

For retrieval and display of bibliographic records, it was hoped that another earlier project developed by KFPL could be repurposed for the task. In 2002, KFPL’s database librarian developed an application to allow staff to create and maintain booklists on the library’s web site. This project was developed with the idea that replicating portions of the library’s bibliographic database on the web site would be a mistake, and that it would be preferable to pull this information directly from the library’s database as needed, thereby reducing data redundancy. The application was developed on a LAMP[7] platform, using Index Data’s YAZ server[8] to retrieve records from the database via Z39.50. For the backend of the web site, KFPL staff are presented with a set of tools allowing them to create a booklist, and search the library’s database for records to populate it. Search results are accompanied by links to add individual records to the booklist. Only the unique identifier for a selected record is stored in the booklist database, and when a booklist is retrieved on the public web site, a series of Z39.50 requests are made in order to present users with the bibliographic records needed to display the list. It was hoped that, with minor modifications, the staff search tool could be used to display bibliographic records for the Google Maps mashup as well. Instead of users entering a search term in a form to retrieve results, they would click on the map.

Before actual development on the project began, an opportunity to get help from the library IT community presented itself. Prior to the Canadian Library Association’s Access 2006 conference in Ottawa[9], organisers issued a request for submissions of problems or projects to be worked on in the event’s Hackfest. The Access Hackfest is a one-day pre-conference event that is typically attended by 30 to 40 people with varying levels of technical expertise. Attendees are presented with a list of these submissions and asked to self-organise into groups to work on them. The idea for the Google Maps OPAC mashup was submitted to Hackfest organisers and accepted.

Two of KFPL’s staff attended Hackfest on October 11, 2006, but unfortunately no groups were formed to tackle the Google Maps OPAC project. Failure to have the project worked on at Access 2006 was not noted except in the Hackfest results, which
were presented to conference attendees, where it was reported that although this project was interesting, for a variety of reasons it would be impossible to create[10]. This served as the impetus to stop conceptualising the project and to begin work on development.

The project was developed incrementally and began with the coding necessary to display the Google Map. With the map in place, the challenge became one of creating a search term to send to the library’s bibliographic database following a mouse click on the map by a user. The decision was made to begin at the country level, and not to handle searches for smaller geographic areas such as states and provinces or cities. At the time of development, the Google Maps API did not have the functionality to return a country name for a point that a user clicked on, so a further decision was made to place a marker on the map for each country. Clicking on one of these markers would trigger a search for bibliographic records related to that country.

To place a marker on a Google Map, the API requires the latitude and longitude for the point where it will be placed. These coordinates were found on the CIA’s World Factbook (CIA World Factbook, n.d.) web site, where a page is maintained listing the average latitude and longitude for 284 countries. For the purposes of building a working prototype, it was decided that this was a sufficiently accurate and comprehensive source of data. The coordinates were loaded into an XML file that is read by the Google Maps API to place a marker at the centre of each of the 284 countries. Clicking on the marker initially displayed a text box overlay on the map containing the name of the country. It was a small step from here to convert this into a search string that could be sent to the Z39.50 server to retrieve a set of records (see Figures 4 and 5).

**Figure 4.**
Google Maps OPAC mashup screen showing a marker on each country
A brief survey of the library’s subject heading index led to the conclusion that it would be a mistake to retrieve all records containing a given country’s name, and that it would make much more sense to limit the search to records related to the subjects of history and travel and description. It was decided that the text box overlays for markers on the map would present the user with the option to search the libraries catalogue for records on either of these subjects as the cataloguing for these headings is applied fairly consistently. One of the limits involved in this approach is that for many records in KFPL’s catalogue, the country name in the subject heading does not match the country name provided by the CIA. This is a historical and political problem for which no solution has been applied during the course of development, but in theory it would not be difficult to create a list of country names that over time have been associated with the coordinates for a map marker and to use Boolean logic to retrieve records from the catalogue for all of those names when the marker is clicked. Failing this solution, an “Any Keyword” search was created for both history and travel description that includes only the name of the country supplied by the CIA.

Mike West’s DataRequestor AJAX tool made it relatively simple to take search strings from the text box overlays, and use them to load a set of catalogue results onto the right side of the screen without causing the entire page to reload (see Figure 6). With this last technical challenge solved, the prototype was considered complete in December 2006. The application was demonstrated at the Ontario Library Association’s Superconference in February 2007 (Ryhno and Vandenburg, 2007), and again in an Education Institute session in March 2007,(Vandenburg, 2007) but has not yet been posted on KFPL’s public web site. If development time becomes available, it
will be used to work on the issues surrounding country names before the application is made public.

The development of this application has served as more than simply a personal challenge for the database librarian at KFPL, as all of the skills required to complete it have been useful in other projects that being undertaken at the library. Libraries that encourage their employees to “play” with technology and provide them with the time and tools to do so stand to benefit in both the quality of the services they provide to users, and the health and creativity of their employees.

Notes
1. The Richmond Public Library in British Columbia began displaying book cover images on their web site in 2003 through an associate account that they have with Amazon.
2. Initially web sites such as MapBuilder (www.mapbuilder.net/) provided tools aiding users in the creation of customised Google Maps, but more recently Google has added the “My Maps” feature to Google Maps to allows users to create these maps directly on the Google web site.
3. Housing Maps web site – www.housingmaps.com
5. AJAX is the acronym for Asynchronous JavaScript and XML, a technique used to make web sites more responsive to user input, allowing portions of a page to be loaded with new information without requiring a complete reload of the page.
6. Mike West’s web site – http://mikewest.org/archive/datarequestor
7. LAMP is the acronym for Linux Apache MySQL PHP, a set of open source software tools commonly used to create web applications.
8. Index Data’s YAZ server web site – www.indexdata.dk/yaz/

References


Vandenburg, M. (2007), Mashups, a New Opportunity in This 2.0 World, Education Institute, London, 6 March.

About the author

Michael Vandenburg received his MLIS from Dalhousie University in 1998, and has worked as a systems librarian in Halifax, Nova Scotia, Philadelphia, Pennsylvania and Kingston, Ontario. Michael currently holds the positions of Vice-President/President-Elect of the Ontario Library and Information Technology Association and Past-President of the Geac Library User Group.