

MapHappy: A User-Centered Interface to Library Map Collections Via a Google Maps “Mashup”

Lisa R. Johnston and Kristi L. Jensen

Lisa R. Johnston is the Geology, Physics, and Astronomy Librarian at the Science & Engineering Library at the University of Minnesota, 108 Walter Library, 117 Pleasant St. SE, Minneapolis, MN 55455 (e-mail: ljohnsto@umn.edu). Kristi L. Jensen is the Head of the John R. Borchert Map Library and Map Librarian at the University of Minnesota, S-76 Wilson Library, 309 19th Ave. South, Minneapolis, MN 55455 (e-mail: kjensen@umn.edu).

Abstract

Providing users the best possible access to the unique cartographic materials found in libraries has been a constant challenge for Map Librarians. In a Web 2.0 world, existing mapping APIs make it possible to extend the library OPAC past the text-based search and enable users to locate maps using a familiar interface, such as Google Maps. This article describes how librarians at the University of Minnesota developed MapHappy, a “mashup” of their existing map MARC records into a geospatial web-application providing unique access to the print maps in their collections. As expected, this project raised many questions and produced a new set of challenges and opportunities; two such problems, the issue of missing or faulty map coordinates and aligning the interface design with user-expectations, are discussed. And while still in beta, a variety of future plans are considered for the further development and improvement of MapHappy. This project demonstrates that the representation of library records in a geospatial format provides a more intuitive and streamlined method of identifying maps and makes accessible a wide range of data previously meaningless or invisible to users.

Key terms: maps, libraries, map libraries, mashup, Google, access, user-centered web development

Introduction

Providing users the best possible access to the unique cartographic materials found in libraries has been a constant challenge for Map Librarians. Up to now, the evolution of descriptive cataloging and access of maps has largely followed the same path as books and other library

materials. Their information was filed in the historic card catalog and physically they were arranged for browsing purposes. More recent technological advances, including the advent of online catalogs and the Internet, have provided both new opportunities and new challenges. Online finding aids and complete retrospective cataloging projects respectively offer varying degrees of breadth and depth of the map collections held by a particular library. But they come with the limitation of textual-based descriptions of geospatial objects.

The advent of searchable, online systems greatly increased the likelihood that cataloged cartographic materials might be discovered by the savvy searcher. Library users could perform complex keyword searching across a range of fields. Yet, regardless of how descriptive the catalog information, text-based representations of cartographic materials still requires users to translate what is often primarily a geographic inquiry into a text-based search. In addition, these systems require users to understand and interpret variables like scale and map size with an assumed familiarity. While library staff and frequent users of a collection may easily make these interpretations, this text-only form of searching is a barrier for a majority of map users.

In a Web 2.0 world, the evolution of cartographic cataloging can now extend past the text-based system. Map-based application programming interfaces (APIs) allow anyone to download mapping source code and reprogram it for their own customized purpose. There are several map API's available, with the two popular being Google Maps (Google, 2008) and Yahoo Maps (Yahoo! Inc., 2008). Libraries can use map API's to "mashup" existing data from their MARC records into a geospatial representation to form a single integrated web-application providing access to both print and digital maps in their collections.

A handful of University of Minnesota librarians, with an interest in improving access to maps held across the Libraries' collections, worked together to produce a pilot project exploring the opportunities and issues surrounding a geographical interface facilitating map access. The result was a web-application named MapHappy (available in beta at <http://www-dev.lib.umn.edu/scieng>). This geographical interface allows users to search and browse maps held in the library using a simple web tool they are accustomed to, like Google Maps. MapHappy returns search results found in the library catalog and displays them geographically using their center point coordinates and four-corner spatial coverage onto the map. MapHappy search results also include additional information such as bibliographic details, scale, and a link back to the OPAC for library availability. This representation of library records in a geospatial format provides a more intuitive and streamlined method of identifying maps that cover an area of interest and makes accessible a wide range of data previously meaningless or invisible to users.

Literature Review

When the myriad of data about cartographic items available in library records or in the databases of government and private entities is combined with the recently available web-based mapping technologies, a plethora of collection-based map mashups have begun to emerge on the web. In addition, a variety of publications containing a geographic component have been geocoded, or given coordinates based on key term, and presented in a variety of mashup environments. Successful examples of this approach include:

- Texas A&M University Libraries has provided a mashup interface (<http://txspace.tamu.edu/handle/1969.1/2490>) allowing users to identify items from their collection of digitized versions of the *Geologic Atlas of the United States* series. Rather than solely relying on text-based searches, their alternative interface quickly and easily displays all 227 folios on a map of the United States (Weimer, 2006).
- The USGS Store Map Locator ([http://store.usgs.gov/b2c_usgs/usgs/maplocator/\(xcm=r3standardpitrex_prd&layout=6_1_61_48&uiarea=2&ctype=areaDetails&carearea=%24ROOT\)/do](http://store.usgs.gov/b2c_usgs/usgs/maplocator/(xcm=r3standardpitrex_prd&layout=6_1_61_48&uiarea=2&ctype=areaDetails&carearea=%24ROOT)/do)) allows users to search for or browse to a location using a Google Maps API interface. Users can readily identify USGS topographic maps at a variety of scales available for a particular location, order a print copy of a map, or download a digital version to view onscreen.
- The David Rumsey Historical Map Collection (<http://rumsey.geogarage.com/>) provides access to over 100 maps in a Google Maps interface. These georeferenced maps can also be selected and added as overlays in the Google Earth interface.
- The department of Anthropology at the University of California Berkeley created a mashup, Mapping Anthropology at UC Berkeley (<http://anthromap.lib.berkeley.edu/map.php>) to geographically represent the entire collection of dissertations completed there since 1901. The geographic area represented on the map is focused on the location of the culture or group under study in the dissertation. Users can do a search for a location or zoom in to a geographic region and see all available items for that area.
- The OCLC WorldMap prototype (<http://www.oclc.org/research/projects/worldmap/default.htm>) provides an interactive visual tool for selecting and displaying international library holdings represented in WorldCat, as well as data from other sources related to publishing, cultural heritage, libraries, and collection data.
- Aquifer American Social History Online (<http://www.dlfaquifer.org/home>) allows users to visualize the the first 100 records from their archive search on a map of the United States so that they can view the geographic distribution of the search results.
- The New York City Map (<http://gis.nyc.gov/doitt/cm/CityMap.htm>), discussed in an editorial in the *New York Times* (Klinkenberg 2008), creates a complex mashup offering users a wide range of information from the location of schools, daycares, subway stations, and libraries to online property, building, statistics, and census information. Users can search for a location or zoom to an area of interest.
- The Biodiversity Heritage Library Project took mapping one step further. Their map geocodes books based on their LCSH subject headings (Freeland, Kalfatovic, Paige, & Crozier, 2008). The article explaining the project was the inspiration of the MapHappy team to ask the question: How hard could it be to map the library map holdings into a searchable Google Maps interface?

Each of these examples demonstrates that geospatial interfaces for map and literature search and discovery are a welcome addition to the library tools arsenal. MapHappy, building on these previous examples, dynamically searches and displays maps held in the library catalog to meet emerging user expectations.

Method

Though still in development, the initial planning to create the Library Catalog / Google Maps mashup was relatively linear and three main phases emerged in the project:

1. **Data Extraction:** MARC records were extracted from the library OPAC and converted to XML for ingest into a stand-alone database.
2. **Database Design:** A custom SQL database was built to control the organization of the data. The database was built for speed and programming agility using only essential MARC record data fields and then fully keyword indexed for relevancy-ranked results.
3. **Interface Programming:** The Google Maps API was utilized to mashup the custom-built database with a geospatial interface to create a unique web-application.

1. Data Extraction

In order to kickoff the project, the team first examined potential methods for accessing or extracting existing descriptive information about our map holdings. As a pilot we identified a subset of records to download from the OPAC which represented a sample of maps held in the University libraries; given our collection strengths, maps covering the state of Minnesota were an obvious choice. Manually extracting the data for the pilot project was a fairly simple, straightforward process. Using a command language search in the OPAC we queried for Minnesota maps held in multiple libraries within the Twin Cities campus (subject heading=Minnesota, format=mp, campus library=T*). We attempted to exclude maps contained on microfiche (not format=mic), though this attempt was not entirely successful because some maps are held in multiple locations as both print items and microfiche. Not wanting to limit access to those with alternative formats, some fiche maps were ingested. These nearly 2000 records (see Table 1) became an excellent test-case during the interface programming stage to explore the problems of map rendering, given the small subset of maps with relatively well-known locations. For example, Minnesota maps displayed by the API in Mongolia signaled a coding problem.

Table 1: Pilot "Minnesota" Maps and Library Locations

Library Location	Number of Maps
Main Library	1196

Science & Engineering	201
Agriculture	125
Duluth Campus	119
Duluth Science	97
Forestry	45
Archives	22
Fisheries/Water	6
Heath Science	1
Vet Med	1

Moving beyond a pilot, the team was able to work with the OPAC programmers to extract all map records in the system, over 30,000 as of the printing of this article. Naturally a one-time manual data extraction would not be sufficient to keep the map record data current and up-to-date. An automated script would extract all maps and provide the necessary maintenance to the data set by refreshing newly acquired and recently cataloged map records to the MapHappy database. The final data extraction step was to convert the MARC data into XML. To do this we used an open source software program, *Marc Edit 5.0* (Reese, 2008) to easily convert a tab-delimited MARC file of all the map records from the OPAC into XML format. The XML was then more easily malleable for ingest into an SQL database, the second step of the process.

2. Database Design

The team considered several options for delivering the MARC data to the Google Maps API mashup. The simplest approach includes a flat file of information, typically in XML-tagged format, that would generically display every item on the map. The most advanced option required a custom-built program that queried the library catalog directly, on the fly, as each user searched (like a public interface to the catalog). The former is an excellent choice for smaller holdings, say 100 objects, not a collection in the thousands which would slow down the web application noticeably. The latter requires advanced knowledge of the OPAC's data structure and unrestricted access to its residing server. Since our project had too many records for a static flat-file approach and not enough control over the OPAC structure for direct access, the team took a middle-of-the-road approach. Our solution was to load the library's map holdings into a stand-alone SQL database. Creating a separate database gives full control over how the data is structured so that our mashup can search our records easily and achieve faster results than attempting to search the OPAC on the fly.

Another key consideration for building the standalone SQL database was the essential calculation of center point coordinates for each map. Map MARC records typically record the four corner points of the item's geospatial coverage. The Google Maps API, on the other hand, requires center point locations to interpret where each marker will display. This simple, yet important calculation was done using a Microsoft Visual Basic script that formats each map's center point location into the database as a new field. The script does double-duty as it "cleans" the XML file of any odd characters from the map records before the data is converted into SQL to build the standalone database.

It is also important to note that the coordinate notation used for the Google Maps API must be delivered in decimal degrees. Therefore the MARC records, whose coordinates are in arc degrees must be transformed; while not forgetting that the cardinal direction for arc degree converts the decimal number to a positive or negative (ie. latitude as N="+" and S="-" and longitude W="-" and E="+"). Therefore, the MARC record listing "W930306" in arc degree coordinates, becomes decimal degree "-93.1008" for the API.

Next, when building the database structure, each MARC record field was carefully examined to determine which components were necessary for the project. Of course, pulling the entire MARC record is an easy option, yet, redefining the fields for our own purposes was the only way to simplify the records to a state that eventually would provide a clear programming path for querying it. The data elements selected (see Table 2) represent the table structure of our relational database (table names are in bold) along with the MARC field and an example map record.

Table 2: MapHappy Data Elements

Database Field	MARC Field	Example Map Record	Notes
Titles			"search" parameter
TitlePrimary	245	a Mower County, Minnesota, land atlas & plat book, 2003	
TitleAlt	246	a Land atlas & plat book, Mower County, Minnesota, 2003	
Places			"search" parameter
Place	650	z Minnesota z Mower County	
PlaceAlt	651	a Mower County (Minn.)	note the alternative formatting
Topics			"search" and "refine" parameter
Topic	650	a Real property a Landowners	

Maps			
SysID	001	0114608010	MARC control field used to build link back to catalog
LongCenter	n/a	-93.1008	Average of LongCorner1 and LongCorner2 converted to decimal degrees
LatCenter	n/a	45.4167	Average of LatCorner1 and LatCorner2 converted to decimal degree
LongCorner1	034	d W930306	Coordinates represented in arc degrees, ie. -93° 3' 6"
LongCorner2	034	e W922655	
LatCorner1	034	f N435106	
LatCorner2	034	g N433000	
Scale	255	a Scale [ca. 1:56,320]	Note the formatting
ScaleAlt	034 1	b 56320	Scale fields with and without formatting could be useful in the future.
Author	110 2	a Rockford Map Publishers.	
PublicationPlace	260	a Rockford, Ill.	
PublicationDate	260	c [c2003]	Non-numeric characters are "cleaned" for the database.
Description	300	a 1 atlas (52 p.) : b maps (some col.) ; c 28 cm.	
DescriptionAlt	500	a Shows state lands located within parks and wildlife areas.	
CallNumber	852	h G1428.M7 i R58 2003	The call number might be used for refining search results by LC topic
Libraries			"refine" parameter
Library	852	b TWILS c MAPM 4 TC Wilson Library 5 Borchert Map	May include multiple library locations
URLs			
URL	856	u "Digital Map"	Digital URL of the map on the web, if available (less than 10% of our collection)

** The database table names are in bold.*

With the database structure in place, the next step was to load the map record SQL into the database. Here we analyzed the completeness of our data and discovered that nearly half of the map records were missing coordinates. This problem, later discussed in the results section, might seem discouraging at first, however there are solutions and a future project will hopefully correct any faulty coordinates in the OPAC to update the MapHappy database on our regular rebuild cycle.

3. Interface Programming

From the start, the team wanted an interface to work similar to the current Google Maps experience. First several wire frames were drawn to envision the user experience prior to coding. Ideally, a user might search the map by keyword and the results are displayed both on the map geographically and textually in a results box. The results are delivered based on relevancy using an "or" anticipated search string expanding the breadth of the results (like a Google search) rather than limiting by "and" like the library catalog does. The interface also includes additional information such as map coverage area (represented by a box outlining the four corners of the map), hyperlinks back to the library catalog, and a browseable map that dynamically displays all items in a particular region of the map. Given the flexibility of the Google Maps API, building the user-interface was a great project for any programming novice. And by soliciting the help of a web developer for the more complex AJAX and mySQL languages we were able to put our ideas to life and MapHappy was born (see Figure #1). Our web-application components are:

- **Map Interface:** Programming inherited from the javascript Google Maps API controls how the map renders. Requires an API key and javascript libraries.
- **Search and Browse functionality:** Uses AJAX, a programming language that enables dynamic PHP to search, retrieve, and load data “asynchronously” in the browser without reloading the page.
- **SQL Database:** Allows unrestricted access to the MARC records (see Table 2 for our Database structure).
- **Queries and Results:** MySQL queries the database, retrieves relevancy-based search results, and outputs them as JSON files, a file-format faster than XML.
- **Web page:** PHTML/CSS web page design gives the map a simple look and feel consistent with the University of Minnesota style guidelines.

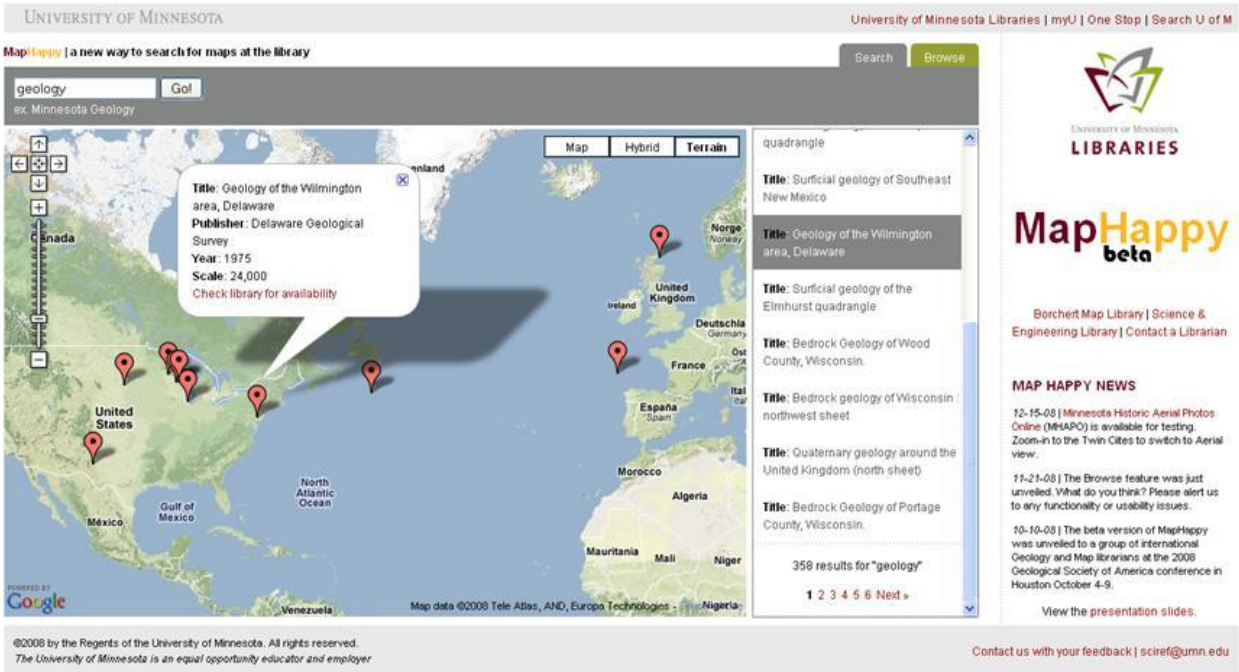


Figure #1: MapHappy screen shot displaying the search tab functionality and results column.

Our interface priorities (see Table 3) were customized to how the user would interact with the map. To accomplish these goals, some major programming hurdles were tackled.

Table 3: MapHappy Interface Wish list

Required Feature	Description
Simple “google like” search box	Search form queries our fully-indexed database to return results based on relevancy, or "best match" and not based on complex search form
Markers convey map area coverage	Each marker "pin point" represents the center of a map with a graphical representation of the map's content.
Results link back to OPAC	An information window should open with relevant details including title, publisher, publication date, URL (if available), and a link to the catalog holdings.
Search results "Refine by" capability	Results list includes user-control over meta-data and refinement of facets, such as Topic and Library location.

Firstly, the requirement of a simple "Google like" search box. MapHappy's search feature was built to interact with our data using MySQL queries, expedited by the fully indexed nature of our database. The search box returns records based on the relevancy of search terms from tables "titles," "places," and "topics." Optimizing and scaling for large search results was a priority. For example, due to the display restriction of the Google Maps marker size, the pin points representing each map can potentially crowd and overlap one another. Therefore MapHappy was designed to display only ten results at a time. Results beyond ten are then "paginated" where additional results appear on the subsequent pages allowing the user to select the next page of ten results and so on. Although the API allows "clustering" of large numbers of results in a small location, this was rejected as it would add an additional layer of complexity and require the user to zoom-in to reveal individual records. Our search results box also alleviates, partially, the issue with missing map coordinates by returning those map results textually, rather than not at all. In this case the search is at least equal to traditional library OPAC, until a more permanent fix can be made.

Furthermore, *how* the search box functions was a major concern. An assumed "or" boolean operator is built-in to our search to always broaden a complex, multi-term query. Rather than being precise (ie. only return results that include exactly both terms) our search relies on the fully-indexed database to provide the best possible results and rank them highly based on relevancy and word proximity. Partial matches are also returned (potentially including only one of the search terms) thereby expanding the interface as a discovery tool. When no results are returned, the user is prompted to check their spelling or to try the "Browse" feature.

The Browse tab allows users to zoom to a particular area and see the maps available for that region. This is particularly useful when keyword searching fails or when a user wishes to have *all* maps covering a region. A couple of issues with this alternative make it a less than ideal solution, as well, however. Given the number of items that can be represented on the Browse map, the markers representing the center points must be limited to a maximum number, in our case 200 maps display once a user has zoomed in to a particular level. Due to this limitation, not all maps covering a particular location might be readily apparent. The center point of each map determines where the marker displays and this will not always be within view once they have zoomed-in. The other limitation of the Browse component is that only maps with coordinate data in the MARC records can be displayed here. Other maps covering this area may be available, and may appear in the search interface results box, but will not show up in the Browse interface.

The next two required features, displaying map area coverage and linking back to the catalog, were relatively simple interface elements to program, but make a big impact for the user (see Figure #2). These developments are possible from the API's built-in interactions which "listen" for user interactions that "trigger" certain events. First, as the user hovers over a particular pin-point a semi-transparent box appears approximating the coverage of that particular item. This information is generated by the MARC record coordinate fields of each corner of the map. Moving the mouse away from the pin-point will remove the box. Clicking each pin-point creates a different action that opens an Information Window for that map. The window displays bibliographic information, including title, publisher, year published, and scale, when these data are available. Perhaps more importantly a link appears in the information window hyperlinking the pin point map to the public catalog map record, allowing the user to "Check library for availability" and then proceed to access the item.

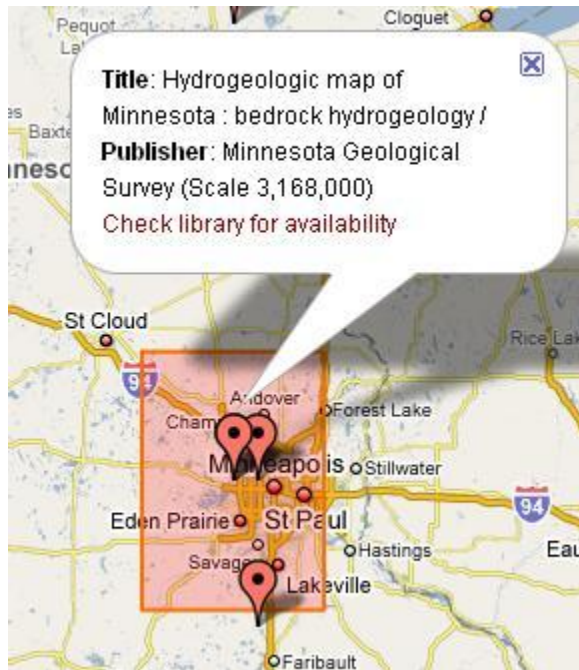


Figure #2: Search results in MapHappy display each map's coverage area as a red semi-transparent box. The markers are fixed at the center point of the map that, when clicked, open an information window linking back to the library catalog.

A final required interface component of MapHappy was its refinery tool limiting the results by either Topic or Library Location (see Figure #3). Refine by Library Location is pretty straightforward and shows the map results listed by how many items appear in their respective library branches. The Refine by Topic is probably a more user-centered feature. The "Topics" in this case are the map's MARC record subject headings. Each map record may have multiple subject headings, all helpful key terms for users attempting to locate a library holding. By displaying these additional keywords, MapHappy enables the user to refine their search based on terms previously unknown. Plus, since our search returns results based on the broadened "OR" assumption, it was necessary to provide users with the ability to specify exact topics to meet their need.



Figure #3: The Refine Results tools was integrated into the search results box allowing users to refine by Subject headings and by the library locations of MapHappy's search tab.

Results and Discussion

As expected, this project raised many questions and produced a new set of challenges and opportunities to be addressed when considering Web 2.0 applications as discovery tools for map libraries. Two problems we encountered, the issue of missing or faulty map coordinates and aligning the interface design with user-expectations, are discussed.

Problem 1: Missing or Faulty Coordinates

Two issues related to coordinates became apparent early in the pilot project phase as we examined the MARC records for Minnesota items extracted from our OPAC. First, not all records contained coordinates. Since not all records contained basic coordinate information which would allow us to represent the items on our geographic interface, mapped results only represent a limited subset of items available in our collections. However, all MARC records for cartographic items were extracted from the OPAC, even those that did not contain coordinate information, allowing users to produce search results in MapHappy that are similar to and as

complete as a search executed in MNCAT, the library catalog. The search results, contained in the sidebar, contain all relevant results including those without coordinates.

Approximately 66% of the MARC records for Minnesota contained coordinates which could be converted to a center point and represented geographically in MapHappy. Providing users an alternative means for accessing approximately two-thirds of the Libraries' maps via this innovative interface was still deemed a valuable proposition, despite the limitations of the data. Entering coordinate information for the records lacking this information was set aside as a potential future project that will be evaluated at a later date.

Second, not all coordinates had been entered in the MARC records in one standard format. In order to convert coordinates to center points as described above, the coordinates contained in MARC records had to be standardized. The most common problem identified was coordinates missing leading zeros. Corrections for these items were manually entered in our OPAC to allow for conversion and use in MapHappy. Other coordinates were simply incorrect, such as "East" rather than "West" arc degree coordinates, that when mapped indicate a map covering Minnesota could be found off the shore of Great Britain or elsewhere in the world. These outliers were easy to identify when dealing only with Minnesota maps and manual corrections were made in the original MARC records. Outliers for the complete data set will not be as easy to identify and corrections will continue to be made in the future when these items are brought to our attention by users and library staff.

Unfortunately the coordinate problem doesn't end with the data. There were additional concerns with map record coordinates on the interface side of things. First, the Google Maps API only allows the display of one pin for each coordinate location. When more than one map is represented by the same coordinates, such as the center of a state, the only indication is the shadow for that pin is darker. This is a very subtle visual representation that may not be intuitive to users, or the programmers for that matter! Currently only information about one of the maps can be displayed in the Information Window when clicking directly on a pin that represents multiple maps. Otherwise the user may only access the lower, hidden maps by clicking on the titles appearing in the right-hand results column. This action, fortunately, will uncover the descriptive information for that map and display it in the pop-up window. There are some alternatives or workarounds to this problem including an information window that has tabs containing information about all maps associated with a pin. This alternative requires significantly more programming and might be explored in the future.

Finally, dealing with map records that contained more than one set of coordinates was another idiosyncrasy. One center point was calculated for each MARC record, stored in the database and geographically rendered to represent the location of an item. However, given our system for calculating center points, a cartographic item that contains more than one map, for example multiple sheets in a set, will display on the interface for only one of the possible locations. While this might seem to reflect "faulty" data, this may not always be the case, but merely reflects how the item was cataloged. Again, several alternatives for dealing with this problem may exist but at this time only a minor number of items appear to fall into this category.

Problem 2: Aligning with user expectations

A number of interface challenges led our team to question users' interpretation of search and browse results available in MapHappy. Some of the most critical issues emerged from our efforts to align with user expectations of a Web 2.0 mashup tool. Whether users do a keyword search or use the Browse map to zoom-in to an area, both options provide users with numerous pins representing the center point for each maps whose coordinate data was sufficient in it's corresponding MARC record. One question that was raised early in development was whether or not users would readily understand what each pin represents. If a user wants a map covering the entire state of Minnesota, how do they know exactly where to go to find a pin representing such a map? It was concluded that the addition of an overlay indicating the coverage area for each map was a fairly simple solution that provided users with the exact information needed to easily interpret their search results.

Another concern arose though presenting users with a Search box on the primary MapHappy interface. When faced with a search box on a page containing a Google map, many users choose to search for a specific location or address (for example, a town or city). Since MapHappy is searching titles, subject headings, and place terms from the MARC records harvested from our OPAC, an "address query," no matter how exceptional in Google Maps, may present unexpected or no results here. Also, because MapHappy provides search results for all terms (by performing an "OR" search on keyword terms) and then provides the most relevant items first, the results may not match up with what a typical user expects to see. For example, searching MapHappy for an address "California Avenue in Saint Paul, MN" may result in a mixture of California and Minnesota state maps, not a Google map zoomed into an address location. The questions of how accustomed users are to searching Google Maps and expecting a geocoded result for a specific address or location is key when building off the API. How best can we indicate that the most successful searches in MapHappy contain keywords describing the topic of a map? Or will users easily figure out how to use the tool after a couple of attempts? These are questions we have yet to thoroughly answer.

Conclusion

The creation and development of the MapHappy project was an exciting learning experience. A number of informal user tests were carried out during the early phases of development, but a more thorough investigation is planned to answer questions about user expectations and their interpretation of search results. And while still in Beta, a variety of future plans are already under consideration related to further development and improvements to MapHappy.

One obvious expectation expressed by users in our early user tests was the desire to find digital maps available in MapHappy. Linking to digital records is already a part of the programming which generates each map's Information Window. When available, the link the digital URL appears, however the current cataloging policy at the University of Minnesota restricts use of the MARC 856 field (digital URL) and therefore only a limited number of map records contained a web link.

Additional efforts are being considered to increase the number of MARC records representing digital maps within our OPAC. For example, the library system routinely digitizes historic map collections that represent the legacy of the institution and these scans should be consistently

made available through library tools, including MapHappy. Also, we might easily bring in a large batch of records representing digital maps already cataloged by other institutions, for example the David Rumsey Historical Map Collection (<http://www.davidrumsey.com/>). There are many valuable pools of digitized and digital maps available on the web, such as those archived at ALA's MAGERT Map Scanning Registry (<http://mapregistry.library.arizona.edu>) and the Western Association of Map Libraries Scanning Projects Clearinghouse (<http://www.waml.org/clearinghouse.html>).

A next necessary step is a large scale "clean up" of map MARC records that currently cannot be displayed geographically due to a lack of coordinates in the MARC records. Correcting this issue is possible, but labor intensive. A variety of approaches are being considered including programmatically geocoding records based on their Library of Congress subject headings, similar to the efforts carried out in the Biodiversity Heritage Library (Freeland, Kalfatovic, Paige, and Crozier 2008). Otherwise re-cataloging the maps might involve manually locating and recording the information contained on the physical map or checking relevant databases, such as GeoBase or OCLC, to see if the data is available elsewhere for these items.

Finally, there is the issue of implementation. How do we introduce MapHappy to our library users and streamline it into their information seeking behavior? Ideally, a link to MapHappy would be readily available to users while searching for maps or other cartographic materials in the public library catalog. Examples of this sort of mapping capability are beginning to appear in a variety of vendor databases and, with the proper programming support, could be inserted as a component of our OPAC. In addition, multiple promotion strategies will be geared toward our target audiences in the Map Library and the relevant university departments, like Geography and Geology.

The Web 2.0 world and the map mashup options offer a world of possibilities and our "future plans" list could be never ending. MapHappy has been well received by our Minnesota Colleagues and thus far the initial user group feedback demonstrates that the representation of library records in a geospatial format provides a more intuitive and streamlined method of identifying maps. We look forward to the many future opportunities and challenges MapHappy affords our libraries with hopes of making accessible a wide range of collections previously meaningless or invisible to users.

Acknowledgments: The authors would like to thank Gary Fouty, Jan Fransen, Blagovest Dachev and Heather Hessel for their invaluable assistance in the development, implementation, and support of this project. MapHappy is available for comment and review at <http://www-dev.lib.umn.edu/scieng>.

Works Cited

Freeland, C., Kalfatovic, M., Paige, J., & Crozier, M. (2008, March 24). *Geocoding LCSH in the Biodiversity Heritage Library*. Retrieved November 3, 2008, from The Code4Lib Journal: <http://journal.code4lib.org/articles/52>

Google. (2008). *Google Maps API*. Retrieved November 3, 2008, from <http://code.google.com/apis/maps/>

Klinkenborg, Verlyn (2008). *Map Upon Map: New Dimensions in What Maps Can Do* :[Editorial]. New York Times, November 12, Late Edition (east Coast).

Reese, Terry (2008). *MarcEdit version 5.1* last updated April 17, 2008. Retrieved from <http://oregonstate.edu/~reese/marcedit/html/downloads.html>

Weimer, Katherine H., Kimble, R. (2006). "Access and Preservation of Scientific and Cartographic Literature Using an Institutional Repository; USGS's Geologic Atlas of the United States in DSpace. *Documents to the People*, 20-24.

Yahoo! Inc. (2008). *Yahoo! Maps Developer APIs*. Retrieved November 3, 2008, from <http://developer.yahoo.com/maps/>