“Do I Live in a Flood Basin?” Synthesizing Ten Thousand Maps

Miguel Elias
Department of Computer Science
University of Chile
Santiago, Chile

Jeremy Elson, Danyel Fisher, Jon Howell
Microsoft Research
1 Microsoft Way
Redmond, Washington USA 98052
{jelson, danyelf, howell}@microsoft.com

ABSTRACT

The recent introduction of simple, web-based geographic visualization interfaces has unleashed a tidal wave of new geographic content now available on the Internet. There has been enormous attention on the development of data interchange standards and programming interfaces that make all this content interoperable, but far less thought about how the user experience should change when users have their choice of 10,000 maps.

To inform the design of online mapping systems, we investigate the case of queries that require correlation of multiple maps---that is, discovery and synthesis of several map layers. We based our study on interviews with expert users of maps: archivists and librarians. This paper describes our user-task taxonomy distilled from these interviews, and presents MapSynthesizer, a prototype system that allows users to efficiently query, discover, and integrate many maps from a corpus of thousands.

Author Keywords

Online maps; GIS; mashups

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

The past few years have seen an explosion of interest in using web-based tools for visualizing geographic data. While computerized Geographic Information Systems (GIS) tools have been available for decades, their price and sophistication have limited them to a specialist market. Online map portals targeted at consumers (such as Google Maps, Microsoft Virtual Earth, Yahoo! Maps) have begun to offer simple, public programming interfaces, putting a core set of useful GIS tools into the hands of a much larger audience of non-specialists. Geographic “mash-ups” have become commonplace, typically relating third-party content with the mapping portal’s standard road maps and aerial photography. Web sites abound in stunning variety: interactive maps annotated with local gasoline prices, reported crimes, real estate prices, amateur weather observations, bus routes, bicycle trails, and so forth.

Most existing web applications place a single form of data (e.g., gasoline prices) over a single standard map layer (e.g., road map). However, there is obvious utility to combining multiple types of data in a single view: Which bicycle routes are near a bus stop? Which houses are for sale in areas of low crime? As a result, many interfaces have begun to introduce *layers*: annotations that can be turned on and off individually, and synthesized into a single display. Current interfaces—both in consumer tools and professional GIS products—can support the selection of a few layers from among several, by utilizing the file system or a checkmark list of prepared overlays. However, they are targeted toward users who have a small set of layers that they are familiar with, such as maps that they have created or based on familiar data.

The existing interfaces work well when there are just a few layers to choose from. But we now have the ability to share *thousands* of maps from distributed sources on the internet, as well as from broad libraries coming online. Map archives are being scanned and a variety of organizations are making their own maps available. While these maps may be individually useful, they are even more powerful when combined. Standards such as GeoRSS [4] and KML [8], services such as Microsoft’s Virtual Earth Collections [17], and application interfaces such as Google Maps API [6] have made it easier to share geographic data across the web.

The wide availability of interoperable map data raises interesting new user interface challenges. How does a user discover and explore the multiplicity of maps that are now available? What cues will help her find the maps she needs? How can the search space most effectively be narrowed? Existing user interfaces fail in many of these tasks at large scales.

Map librarians and archivists repeatedly encounter—and solve—these problems manually. Their experience lets us understand how users are likely to address and use

CHI 2008, April 5–10, 2008, Florence, Italy.

multiple-layer maps. To understand how to build a better mapping interface, we interviewed eight map librarians and archivists, asking them to describe how they answer questions about the world using both paper and digital map collections. We found significant commonalities in their experiences, which we distilled into a simple taxonomy that covers common uses of map data.

This paper makes two main contributions to the question of user-centric map discovery and exploration. First, we attempt to define the problem space of correlating maps through our interviews with experts, and we present the framework derived from the interviews. Second, we present a prototype map browser that can scale to tens of thousands of maps. It embodies several design features motivated by our taxonomy of map collection uses. We describe our map browser and report our experiences in using it to browse maps from among the ten thousand of the Washington State Digital Archives.

While our subject is maps, this paper also has important implications for information retrieval interfaces. In general, we have good methods for handling both highly structured data (such as Flamenco [7]) and entirely unstructured (such as standard search engines). However, we are still learning how to handle situations when we have a mix of structure. In the case of maps, we have a well-defined coordinate system but more poorly-defined metadata. Our discussion is relevant to navigation techniques for these systems.

The organization of this paper is as follows. First, we describe our interviews and the taxonomy of map use. Next, we describe the design principles that we derived from the interviews. We then describe MapSynthesizer, our scalable map browser. The first component of MapSynthesizer is MapCruncher, which georeferences and displays individual high-resolution image-based maps, including paper maps. MapSynthesizer's browser-based user interface allows georeferenced maps to be quickly discovered, layered, and compared. Finally, we discuss our conclusions.

HOW PEOPLE CORRELATE MAPS

In building an interface that scales to a large number of layers, we began by building a clear picture of the task users would expect the interface to perform. This was difficult: while computerized tools that enable viewing of multiple simultaneous map layers have long been available, there seem to be little data available on how typical users interact with such systems. (However, some usage tasks for government GIS users are available [15].)

To understand this task, we interviewed eight experts: librarians, archivists, and GIS professionals, many of whom specialize in fielding map-related inquiries from the public. The experts are listed in Table 1. We asked each of them to describe examples of research queries they had received from patrons that were correlating multiple maps together.

Our experts described a total of 27 unique user scenarios. There was significant overlap and repetition in the replies we received from each expert, giving us confidence that the scenarios cover a reasonably complete cross-section of map correlation. We begin with some reflections on the notation of base information, and the differences between online maps and digital sources.

Table 1: Experts we interviewed on how users correlate maps.

E1	Map Librarian, university library
E2	Map Library technician, university library
E3	Public services librarian, university library
E4	GIS Staffer, university
E5	GIS student
E6	Archivist, state archives
E7	Chief of Archives, state archives
E8	Map Librarian, federal agency

Base Information

In the cartography literature, maps are discussed as portraying a central feature over *base information* [13]. Base information is the geographic frame of reference for the map: well-known boundaries or landmarks that are understood by both the mapmaker and the reader. For example, a map of the population of 1770 Europe might show contemporary cities and borders as base information. Base information is critical to orient the user to an unfamiliar map: topographic maps show roads and state boundaries; demographic maps show county lines; bathymetric (water depth) maps show shore towns and bridges.

Expert discussion of maps

We describe three important distinctions that the experts drew for us: between presentation maps and shapefiles; between map locations; and between the ways that users accessed those maps. We then present our taxonomy.

Shapefiles versus Presentation Maps

The production of a computer-based map can be divided into two steps: First, one collects a *shapefile*, which both describes the geographic location of features and includes machine-readable semantically-meaningful attributes: the altitude of a contour line, or the name of a lake. Shapefiles may also include digital metadata descriptions of what they show. The shapefile is a collection of machine-readable facts, but by itself, not yet a map.

The second step is making a *presentation map*, in which a cartographer makes decisions about how to present the shapefile data in context. The cartographer usually has both an output presentation modality (e.g. a 24x36" paper map) and the intended application (hiking) in mind. The cartographer selects a projection suitable to the scale, locale, and terrain of the map. He makes decisions including how to use color, texture, and thickness to emphasize important features or to defocus base layers. The cartographer also decides when to elide data to reduce clutter or improve usability of the map.

Presentation maps may be available digitally in either raster (for example, scanned paper maps or maps rendered to JPG) or vector (rendered to PDF) digital formats; however, in either format, the content of the map is generally no longer machine-readable. Therefore, unlike a shapefile, tasks such as "hide all the water features" are not generally practical with a presentation map.

The content of most web map mashups are transmitted as simple shapefiles, drawn with basic presentation in the browser. Because shapefiles are machine readable, they afford the opportunity to optimally overlay layers, since the computer can still distinguish between "river" and "class B airspace boundary;" in presentation form, both are just blue lines.

Our experts indicated that they sometimes satisfied patron requests using both kinds of maps, including engaging in crude cartography to turn shapefiles into maps (E4). The majority of the data they accessed, however, was only available as a presentation map. Presentation maps are not only widespread, but they contain a wealth of information not found elsewhere.

Even when new maps are drawn by computer, presentation maps are published far more often than shapefiles. This is necessarily true for the many maps designed to be printed on paper. More generally, however, maps are designed to be *used*. The zoomable road map layers in Virtual Earth, for example, are images that reflect explicit design decisions about which details appear at what scale and where location labels appear. In addition, countless historical maps were drawn before computerized GIS systems existed. Although these maps are often scanned and made available online, they are still presentation maps; reconstructing the semantically-equivalent shapefile is very difficult (and, therefore, expensive).

Thus, we have the linked observations that presentation maps are very common, contain data that is very important, and are unlikely to be anything but presentation maps in the foreseeable future. These observations have an important impact on the design of MapSynthesizer. Clearly, any system intended as a large-scale resource for geographic data must contend with presentation maps, not just shapefiles. Fortunately, tools like tools like MapCruncher [3] aid tremendously in the process of making presentation maps interoperable: each can be re-projected into a common coordinate system, and can be downloaded incrementally as the user pans and zooms through a large map.

How Maps are Accessed
The experts we spoke to brought up several interesting notes about how users accessed their maps. They noted that users' information needs were non-specific: a user would come in with a desire to resolve a general problem, but not with the knowledge to know which maps would address their problem. For example, one user came in wanting to trace the Cowlitz trail, which passes through the Pacific Northwest (E7). The archivist was able to suggest which historical maps were likely to show the Cowlitz trail, and then help the user locate the trail by comparing against contemporary topographic maps. The experts consistently emphasized that maps are only part of the story: that external information sources help make sense of the maps.

Locating Maps
The particular area of interest indicated on a map—its *location*—is crucial to a librarian or archivist; they consider extent of a single map to be defined by the boundaries of the paper on which it is drawn. In their model, a road map of Seattle and one of Portland are distinct. Each is separately archived, indexed, searched for, retrieved, and viewed.

Map Classification Axes
Based on our discussions with the map experts, we decided on a simple, two-axis classification of maps in a collection: *category*, and *time*. These two axes are depicted in Figure 1.

The continuous time axis describes the point in time whose features the map depicts, regardless of when the map was actually drawn.

The discrete category axis characterizes the content of the map: the types of features depicted. Some maps depict categories of natural phenomena: water depth, tree canopy type, bedrock composition. Others depict categories of man-made phenomena: structures, political boundaries, or demographics. A map, then, represents at least one point on this two dimensional space; a second point would represent the base information. In Figure 1, we label the base information as living in the "familiar" category.

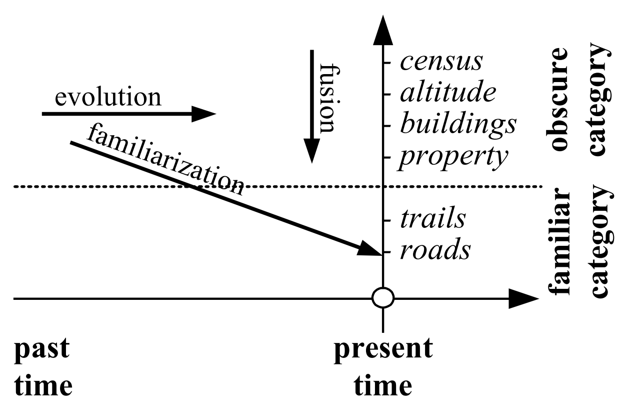

Figure 1: The conceptual axes we use to classify maps. The vectors indicate common user map tasks as reported by the experts we interviewed.

Our classification differs from the experts', however, in that we do not place "location" on these axes. Computer-based map viewers have done away with some of the artificial map gutters—the seams between paper maps—that exist only as a technical limitation of paper. Microsoft Virtual Earth, for example, has a layer called "Roads" which implicitly covers both Seattle and Portland—a single "map" that an archivist might have considered two distinct maps. Thus, in our model, a set of maps that can be stitched together at the seams are no longer "maps that are closely related;" they are now a *single* map.

Some gutters are, of course, social: even two different state highway maps may not align well at the edges where they disagree about town names, highway labels, or color schemes. But some maps—such as US Geological Survey (USGS) topological quadrants—are parts of a larger, continuous view of data.

Moving away from the notion of location-specific maps helps us build a large scale map synthesizer because it can dramatically reduce the number of "maps" in a collection. All of the USGS maps of Washington State for a given year, each depicting a small area, can be merged, and thus thought of, as a single continuous map layer. This can reduce the number of items that show up in a map collection index by an order of magnitude.

Task Classification Axes

We now turn to the main subject of our expert interviews: what questions are people typically trying to answer when they compare maps to each other? From the 27 distinct scenarios the experts described, we distilled three major task types: *familiarization*, *evolution*, and *fusion*.

Familiarization

We refer to the first task performed in most applications of maps is *familiarization*: relating the primary features of a map to a context understood by the map user. That is, gaining an understanding of the spatial context of the map in familiar terms. This is accomplished by placing the data of the map over the socially-understood base information. A familiar map is one that depicts a recent time and shows features from a category familiar to the user. Thus, whether a synthesis task is familiarization depends on the user's existing knowledge. Most of the tasks turned up in our interviews were familiarization tasks, relating a map of an unfamiliar category or a past time to a recent, familiar map. This task is aided by the base information provided by the map, which gives a point of reference to the map.

One example of familiarization came from expert E7, an archivist, who was visited by homeowners trying to determine whether their property was inside a protected watershed. The expert found a map of protected watersheds in their area. For familiarization, it also depicted coastline contours and city names, but did not show streets. The expert then found a street map of the same area and of a similar scale and helped the homeowners to synthesize the two, matching points on the unfamiliar watershed map to points on the familiar street map.

As in this example, the preponderance of familiarization reported by our experts occurred even though the origin map contained some familiarization features of its own. In some cases, the map of interest was old and its own base information was no longer appropriate for a modern audience: E1 reported that a geologist found a bedrock map with state boundaries, but wanted to relate interesting features to familiar hiking trails so she could arrange a field trip. In some cases, the familiarization features were insufficiently detailed.

Additional examples of familiarization included:

- relating water rights, property lines, and Indian reservations to present property occupancy; (E7)
- discovering prior uses of property; (E1)
- planning a journey in an unfamiliar location or mode; (E1)
- planning a geology field trip, relating geographic map data to topographic maps showing hiking access; (E1)
- discovering historical or logging roads and railroads; (E3, E6, E7)
- traveling to historical Indian trails; (E6)
- learning where a grandparent lived by translating from historical landmarks to contemporary labels; (E3, E6)
- finding cemeteries; (E3, E6)
- studying ghost towns; (E3, E6)
- finding a shipwreck; (E3)
- and finding a geocache site near an airstrip.[1]

These applications of familiarization demonstrate that the "familiar" map varies by user: the familiar context ranges from roads to hiking trails to buildings to aeronautical charts. Familiarization enables the map user to take abstract data about an unfamiliar category and act on it, by navigating roads or trails, or by connecting to non-map data sources. The features on familiar maps, because they are both temporally modern and belong to widely-used categories, are well-connected to other non-spatial data. ("Does that hotel next to the archaeological dig have a hot tub?")

We note that a side effect of a large-scale map-synthesizing interface is to make familiarization features unnecessary in most presentation maps. Once any familiar map can be visually synthesized with new maps of interest, built-in familiarization becomes redundant. Thus, the map need *only* feature its novel information.

Evolution

The theme of studying "change across time" was ubiquitous among our experts. We describe as *evolution* the broad category of synthesizing maps in a common category across time periods.

In simple cases, such as matching a single historical map with the present, the task can really be classified as familiarization. The evolution category refers more to

[1] This "aerocaching" scenario is a hobby of two of the authors; it did not appear in our expert interviews.

complex queries that require a series of maps to be compared, each drawing fundamentally the same data but from a different time. For example, three experts (E1, E5, and E7) reported interest in studying erosion of a shoreline by comparing a series of maps or photos that depict it.

In some cases, the evolution task is opportunistic rather than planned. That is, users of our experts' map archives sometimes came to find a single map and were surprised (and delighted) to discover a time-series of interesting maps. Thus, we suggest that a large-scale map synthesis system should both make it easy to find content related along the time axis, and easy to view an evolving time series of maps that are properly overlaid.

This is an area particularly badly served by consumer-oriented map tools: they typically focus only on having the *latest* data. The major online map sites (MapQuest, Microsoft, Google, Yahoo!, etc.) all compete to have the most up-to-date road maps and aerial imagery, yet none of them have an interface for accessing the enormous volume of useful, out-of-date (historical) data.

Other real scenarios reported by our experts included:

- studying the natural meandering and human modification of streambeds, riverbeds, and canals ; (E1, E5, E7)
- daylighting a stream to match its appearance from before the Army Corps buried it ; (E1)
- studying the erosion and modification of shorelines ; (E3, E5, E7)
- relating the original plat ; (land division) plan for a community to its eventual development ; (E5, E7)
- studying the changes of counties or other political boundaries ; (E1, E7)
- and studying changes in road and railroad networks. (E3, E7)

Fusion

We call our last user-task category *fusion*, defined as the synthesis of two maps from categories that are both unfamiliar to the user. We consider fusion distinct from familiarization, in that the latter is the special case of placing a new map into a user's mental model of their world's geography. Fusion, on the other hand, refers to tasks whose focus is to compare two distinct datasets to each other. With fusion, the important feature is that the maps be correctly aligned relative to *each other* for comparison, with a lesser emphasis on familiarization's requirement that that the user understand where the aligned maps are in an absolute sense.

Perhaps the best-known historical example of a fusion map is epidemiologist John Snow's 1854 depiction of London, which he created to illustrate his hypothesis that a Cholera epidemic was due to a water-borne contaminant [9]. The map correlated the position of Cholera patients with areas of the city served by different water sources. We would characterize this as fusion because the relationship between the illness and the water was relevant, not the absolute location of either in the world.

Our modern map experts reported that users primarily wanted to fuse two or more maps drawn from the present-day—that is, using the most up-to-date possible information. However, in some cases, users perform fusion of historical maps. For example, experts E1 and E6 both reported serving users who wanted to identify features seen in historical (e.g., 1930s-era) aerial photography by correlating them with maps from the same era. However, whether fusing maps from the past or present, fusion was nearly always described as across maps from the *same* time.

Fusion tasks reported by our experts included:

- relating ocean temperature to shoreline features for an environmental study; (E1)
- relating community size to the capacity of the roads that connect to it for a demographic study; (E7)
- relating census demographics to the location of hospitals and business parcels for an ethnographic study; (E1)
- relating wetland habitats, soil composition, bedrock composition, and critical areas as part of an ecological study; (E1)
- relating bathymetric data to shoreline topography (E8).
- interpreting decades-old aerial photographs using plat maps from the same period; (E1, E6)
- linking low-resolution navigational bathymetric data to high-resolution seafloor scans for a bridge engineering project; (E8)
- and tracking pollution flows from a shipwreck to shore using water current maps (E3).

Summary of synthesis tasks

Each map synthesis task can be characterized as a straight line within our two-axis characterization of maps. Familiarization relates any map to a map near the "origin", evolution relates maps along the time axis holding category constant, and fusion relates maps along the category axis holding time constant. Although other connections would be possible, they did not occur in the interviews, to our surprise. Our interviews with experts suggest that these three categories well characterize most tasks required by actual users.

A system that facilitates these three task modalities, then, should facilitate most of the tasks encountered by the real users our experts described. Just as online maps provide continuous pan and zoom that eliminate the spatial gutters between map pages, a fluid user interface to the set of all maps should eliminate the temporal and category "gutters" that separate maps.

DESIGN PRINCIPLES

We derived a series of design lessons from our interviews with experts, reflecting on both which tasks they needed, and what obstacles now block their way.

- The interface should support both layers that are presentation images and modern shapefile formats. There are a large number of valuable maps that have no shapefile representation.
- Layers should be discoverable based on *location*: if a user is looking at a particular part of the world (e.g., Seattle), it should be easy to discover other layers that have detail at that location.
- Layers should be selectable based on *scale*: a user looking at the city of Seattle probably is not interested in a world map that shows per-capita income by country. Even though a world map covers Seattle, and thus matches a simple location filter, scale mismatches mean the maps are unlikely to be synthesized usefully. (Toyama *et al* [16] take a similar approach in handling the precision of image locations.)
- Layers should be *text-searchable.* Most layers have associated textual meta-data; in addition to the geography and scale filters, filtering based on keyword can further narrow the search.
- Layers should be quickly *previewable*: when a user is exploring a large corpus, many maps will often still match the geography and scale filters. Therefore, seeing a map preview should be very fast and fluid, allowing a user to make a quick decision about relevance. Map previews serve a purpose analogous to thumbnails in an image search engine.
- Maps should be easily *comparable*: once maps of interest have been identified, the user should be able to visually manipulate them to achieve the best possible synthesis.

We note that existing software designed for exploration of map layers often lacks many of these features. For example, Google Earth [5] is currently in wide use by enthusiasts to share geographic data. Overlays are not discoverable from within the application; rather, users must find new layers using text search in an external search engine. These queries are based on textual metadata; there is no automatic way for the search engine to restrict its results based on the user's viewing location or the current map scale. Quick previews are not possible: the entire file describing the layer must be web-downloaded and imported into Google Earth before it can be seen, requiring the user to switch back and forth between the web browser and map application. While Google Earth does have limited support for image overlays, they must be smaller than 2000x2000 pixels and drawn in a cylindrical projection. In contrast, presentation maps often have detail requiring scans of 10 and even 100 times that size, and are drawn in dozens of different non-cylindrical projections.

MAPSYNTHESIZER

In this section, we describe the design and implementation of MapSynthesizer[2], an interface that allows users to explore and compare many thousands of maps. MapSynthesizer is an early prototype that explores some of these design ideas.

In the following sections, we will explore various aspects of MapSynthesizer, and describe how it strives to meet the requirements motivated by the map experts. We first briefly describe MapCruncher[3] [3], a tool that MapSynthesizer uses to correlate map layers; we then discuss the main MapSynthesizer interface.

MapCruncher

Geographic data sharing formats, such as KML [8] and ESRI's Shapefile [2] are primarily geared for transmitting shapefile data. However, as we described earlier, there are millions of presentation maps that already exist, and the thousands more that are being produced each day: both new maps and scans of historical paper maps. The data they contain are unlikely to be translated into a shapefile format. As a result, when they are shared online, presentation maps are typically treated as any other bitmap on the Web; their geospatial properties are not yet well exploited. This makes synthesis difficult.

MapCruncher [3] is a tool that allows integration between presentation maps. It dramatically reduces the cost of posting maps online in a format that is rapidly accessible and preserves the geospatial relationship between each newly-posted map and *every other online map*. After a user manually indicates several correspondence points between features on the input map and a reference to ground-truth, MapCruncher reprojects the input map into the (cylindrical) Mercator projection and slices it into 256x256 pixel image tiles, making it compatible with web-based mapping systems such as Microsoft Virtual Earth and Google Maps.

Our initial corpus for MapSynthesizer came from the Washington State Digital Archives (WSDA), which is in the process of digitizing, geo-referencing, and publishing to the web some ten thousand maps using MapCruncher. WSDA's maps span dates from the late eighteenth century through the present, and cover varying geographic regions, scales, and content. When complete, each of these maps will be available in a layer format that can be overlaid and displayed over all other maps. At the time of writing, 350 maps were available in this shareable format.

User Interface Design

Following Shneiderman's visual information seeking mantra—"Overview first, zoom and filter, then details-on-demand" [12], we created a workflow in which users select a map, add the maps to the working set, and then scroll through their working set to bring them up individually or to overlay multiple maps.

The browser-based interface works left-to-right. The Search Panel, top left, enables the user to discover maps by search

---

[2] http://research.microsoft.com/mapcruncher/MapSynthesizer

[3] http://research.microsoft.com/mapcruncher

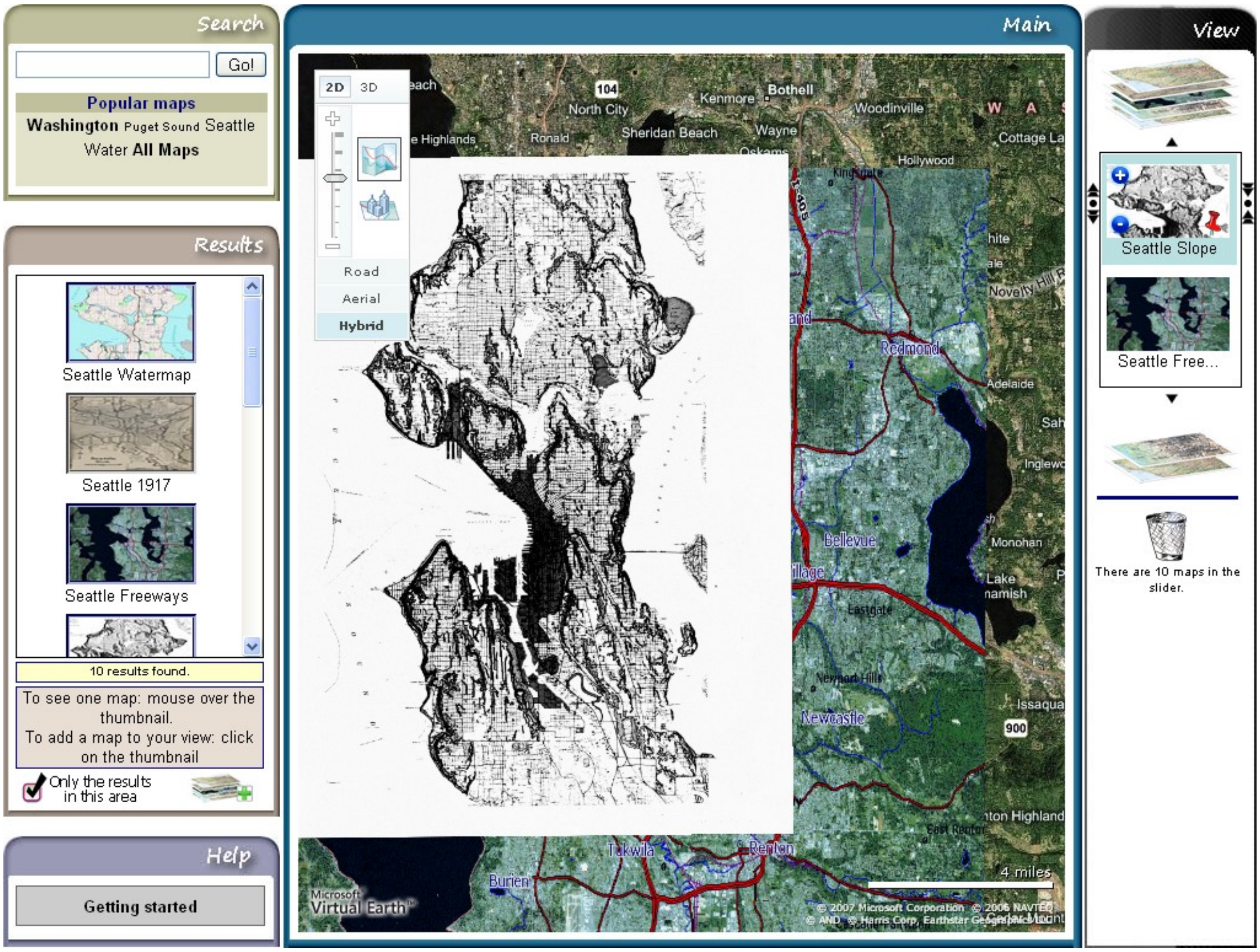

Figure 2: A screenshot of MapSynthesizer. New maps are found on the left, viewed in the center, and then manipulated on the right. See also the associated video demonstration.

keyword or popular tags. In the current implementation, the tags are hand-created; as our collection grows, however, they will be automatically collected from the names of the maps. Maps that match the search criteria appear in the Results Window, center left, as a scrollable list of map thumbnails. Each thumbnail is captioned with the map's name. Searches can be limited to current location and scale of the map view in the Map Window (center). If keywords are left blank, the Results Window contains all maps that match the current view's location and scale.

The Map Window is based on Microsoft's Virtual Earth developer interface, and so allows the user to fluidly pan and zoom around the map, showing whatever layers are currently visible. The tile model means that map access is rapid: previews and images can be loaded in moments—rather than the usual time that traditional map distributions (such as PDF) take to render and re-orient.

Hovering the mouse over a thumbnail in the Results Window displays a preview. The selected map is overlaid over the main Map Window, which is re-centered over the selected map. When the mouse leaves the thumbnail, the Map Window snaps back to the user's last context. This feature makes it easy for a user to quickly preview each map in the Results Window as she is scrolling through the results list.

When the user clicks on a map from the Results Window, that map is pushed into the View Window, right. The user can continue searching in the Search Window, and selecting maps in the Results Window, to push as many maps as desired into the View.

The maps in View can be further manipulated to make synthesis easier. The order of the thumbnails in the View window reflects the z-order of the maps in the Map Window. The user can reorder the maps in the Map Window by dragging the thumbnails up or down in the View. Controls on each map in the View also allow its transparency to be adjusted, allowing maps deeper in the stack to be visible behind maps that are above. The View

Window also has an adjustable-size view port. That is, the user can choose to view all maps in the View simultaneously, or only two or three at a time.

Synthesis Tasks with the View Window

MapSynthesizer's View facilitates all three of the synthesis tasks described in the previous sections. For *familiarization*, the user places an obscure map over a familiar map in the View. Layered maps are shown in the same coordinate system; the user can readily peek under the topmost layer by pushing the map out of the context window onto the top pile, or see through any layer by reducing a layer's opacity. Both techniques enable the user to relate features between two maps by their spatial positions.

Fusion works similarly: the user finds one map first, and adds it to his working context View. A second map in a different category can be discovered using the "search only results in this area" combined with a keyword or tag search. Fusing many categories can lead to a cluttered display; to tidy it up, the View has room for only a few maps. When the stack is scrolled down, maps fall off the bottom onto a "bottom pile," where their order is preserved, but the maps are hidden from the Map Window until they are scrolled back onto the stack. When the stack is scrolled up, a map flips up onto a symmetric "top pile" (hidden but still ordered), and the top map on the bottom pile takes the empty space at the bottom of the stack.

The same View also makes it easy to view map *evolution*. The user can discover a group of results in the same category but from different periods. That group of results can be added en masse to the stack and ordered temporally. The user can drag a knob to reduce the visible region of the View to just a single map, pushing all others into the top or bottom piles. Now, as the user scrolls up and down through the stack, the Main map view flips forward and backward through time. The spatial context does not change until the user explicitly pans or zooms the main map.

The working context View has a single dimension, but works well for all three common task modalities because each modality explores the space of maps along a one-dimensional vector: category, time, or the special case of familiarization that connects just two maps. Thus the MapSynthesizer interface directly represents the exploration space: two spatial dimensions in the Main map window, plus an additional dimension, which varies according to task, represented by the working context View and the z-order of the maps in the Main map window.

Technical Details

MapSynthesizer is implemented entirely in JavaScript, allowing users from any platform to quickly get started without having to install any client software.

In order to generate a MapSynthesizer view, a set of maps are pre-processed through MapCruncher. This step generates—for each map—a series of map overlay tiles, a thumbnail image, and a metadata file that describes the location of the map and its description. In a full implementation of MapSynthesizer, these maps could be located anywhere on the internet and could be indexed with available search engines. (In our prototype, we provide a simple textual search across the limited set of maps that we provide to MapSynthesizer).

The search collects the series of relevant metadata and thumbnails and arranges the thumbnails as small images. When the user brushes over any thumbnail, MapSynthesizer uses the metadata to move the map to an appropriate vantage, and then overlays the tiles. The tile overlay natively supported under the Virtual Earth API is rapid, as each tile is small; as such, a preview of the map in real time is nearly instant.

The system overlays all the maps in the map stack over the underlying image. Because each tile is small, this scales well (from a performance perspective) to even a large number of overlaid maps: MapSynthesizer only loads tiles if they are within the scope of the current view, and often only needs to load one or two additional tiles during a pan operation. We designed this interface to facilitate searches over a Web-scale corpus of maps. Our current implementation has a corpus of about seventy maps from Washington State Digital Archives' collection and another ten maps we added by hand to study specific layering interactions. As MapCruncher and shapefile layers tend to be readily discoverable using standard internet search engines, we hope to dramatically increase the corpus available to users within our interface.

Evaluating MapSynthesizer

The design principles above can used to create reference tasks for user tests of mapping systems. Users would simply be given queries that require familiarization, evolution, or fusion. For example, a user might be presented with questions such as "is the house at 221B Baker Street in a flood basin?" and measured on time and accuracy. The control condition would be a close analogue to the way that users handle the situation now: present them with a set of maps that can be opened individually, and panned and scrolled easily.

User Feedback

A user test of MapSynthesizer is reserved for future work: we are currently in process of testing the current prototype following the experimental design above.

However, in creating the MapSynthesizer prototype, we were able to present it to some of our expert users. While the user interface still needs some work, they were able to overlay and explore the maps easily. E7, an archivist from the state archives, wrote, "The more I think about it, the more I really like the concept!" After experimenting with a set, E5, a student archivist, wrote to us, "This thing can be addictive." We feel that this positive feedback suggests that our design is working in the right direction.

RELATED LITERATURE

The literature on map *integration* has addressed the notion of layering data. Integration is the problem of linking multiple data sources together. Research on integration has centered on trying to find ways to link datasets (e.g. [1]). Integration researchers are interested in reconciling databases that are inconsistently or incompletely coded between coders or sources. MapCruncher sidesteps the problem of integration by merging presentation map versions as data layers. These images have some disadvantages—for example, they cannot be queried as relational database entries—but they do provide an efficient approach to making the data available, no matter what form it originally came in.

As mentioned above, a broad variety of tools support a notion of data layers that can be added to a base map, and can be rendered in a variety of different ways. While this use of layers is common, there appears to be less research in the space of manipulating layers. Past research has noted that map experts showed greater skill than beginners in selecting multiple data layers [11]. Beginners added more layers to their maps and had more difficulty finding their desired information on the maps. But the interfaces themselves seem less well-studied

Cutting between multiple layers was a primary aspect of the Magic Lenses toolkit [14]: visual "magic lens" would show a second view, metadata, or a zoomed view of a specific region of an image. Magic lenses, however, assume that the data is meaningfully available an entire area, and that there would be a reasonably small number of lenses available.

The classic way of handling multiple maps is exactly that managed by the archivists and librarians we interviewed: users spread out a variety of maps on a large table surface, comparing areas and looking for correlations. While effective, this does not take advantage of the increasing numbers of maps available online, nor of precise correlation: archivists are forced to rediscover how maps relate to each other over and over.

CONCLUSION

In this paper, we have considered the design of a user interface that is centered on the task of correlating maps from a large corpus. Although many extant systems support layering of map data, few of them have interfaces that make discovery and manipulation of layers easy, especially as the corpus grows large.

The design of our synthesis-centric interface was informed by interviews with experts (map librarians, archivists, and GIS professionals) who regularly help customers answer real questions using correlated maps. We distilled the scenarios they described into a simple taxonomy of user tasks: *familiarization*, *evolution* and *fusion*. The task categories suggested specific requirements for a user interface: maps should be easy to overlay using a common coordinate system; "legacy" paper maps must be supported to avoid losing centuries of historical data; maps search results should be filtered based on both structured data (map location and scale) and unstructured data (free text search) quick visual previews are often more intuitive than textual metadata to evaluate search results. Each of these three individual tasks can be supported by navigating through a linear array of maps.

We built a prototype system, MapSynthesizer, whose interface incorporates these design elements. Though our experience with it is still limited, we believe it is a simple and intuitive system for discovering new maps, and correlating them to answer familiarization, evolution, and fusion questions.

While this paper has focused on maps, we believe that its insights are useful for a variety of semi-structured search and browse tasks across broad data sets. MapSynthesizer takes advantage of the structure we had—map location and scale—to construct a framework to accelerate navigation through the unstructured map metadata. This model may well be useful in other domains.

ACKNOWLEDGMENTS

We thank the experts for their insights, and the archivists at Washington State Digital Archive, whose technology needs helped motivate the inquiry behind this project.

REFERENCES

1. Beeri, C., Doytsher, Y., Kanza, Y., Safra, E., and Sagiv, Y. 2005. Finding corresponding objects when integrating several geo-spatial datasets. In *Proceedings of the 13th Annual ACM international Workshop on Geographic information Systems* (Bremen, Germany, November 04 - 05, 2005). GIS '05. ACM Press, New York, NY, 87-96.
2. ESRI. 1998. *ESRI Shapefile Technical Description.* ESRI White Paper J-7855. Retrieved from www.esri.com/library/whitepapers/pdfs/shapefile.pdf Sept 19, 2007
3. Elson, J., Howell, J., and Douceur, J. R. 2007. MapCruncher: integrating the world's geographic information. *SIGOPS Oper. Syst. Rev.* 41, 2 (Apr. 2007), 50-59.
4. GeoRSS. http://georss.org
5. Google Earth version 4.2 from http://earth.google.com
6. Google Maps API. http://www.google.com/apis/maps/documentation/
7. Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., and Yee, K. 2002. Finding the flow in web site search. *Commun. ACM* 45, 9 (Sep. 2002), 42-49.
8. KML. http://code.google.com/apis/kml/documentation/
9. Koch, T. 2004. The Map as Intent: Variations on the Theme of John Snow. *Cartographica.* 39:4. Winter, 2004.

10. MacEachren, A. M. 2004. *How Maps Work: Representation, Visualization, and Design*. New York: Guilford Press. Paperback edition.

11. McGuiness, C., van Wersch, A. ,and Stringer, P. 1993 User Differences in a GIS environment: A protocol study. *Proceedings of the 16th Conference of the International Cartographic Association.* May 3-9, Cologne, Germany. 478-485. Cited in MacEachren (2004).

12. Shneiderman, B. 1997. *Designing the User Interface*. Addison-Wesley.

13. Slocum, T., McMaster, R., Kessler, F., and Howard, H. 2003. *Thematic Cartography and Geographic Visualization.* New York:Prentice Hall.

14. Stone, M. C., Fishkin, K., and Bier, E. A. 1994. The movable filter as a user interface tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems:* (Boston, Massachusetts, United States, April 24 - 28, 1994). CHI '94. ACM Press, New York, NY, 306-312.

15. Suchan, T. 2002. Usability Studies of Geovisualization Software in the Workplace.” *Proc. of Annual Nat’l Conf. on Digital Government Research (DGO ’02)*. ACM Int’l Conf. Proc. vol 129.

16. Toyama, K., Logan, R., and Roseway, A. 2003. Geographic location tags on digital images. In *Proceedings of the Eleventh ACM international Conference on Multimedia* (Berkeley, CA, USA, November 02 - 08, 2003). MULTIMEDIA '03. ACM Press, New York, NY, 156-166.

17. Virtual Earth Collections. http://collections.live.com