

A Process for Keeping Pace with Evolving Web Mapping Technologies

Robert E. Roth
University of Wisconsin–Madison
reroth@wisc.edu

Richard G. Donohue
University of Kentucky
rgdonohue@uky.edu

Carl M. Sack
University of Wisconsin–Madison
cmsack@wisc.edu

Timothy R. Wallace
University of Wisconsin–Madison
twallace2@wisc.edu

Tanya M. A. Buckingham
University of Wisconsin–Madison
tbuckingham@wisc.edu

The current pace of technological innovation in web mapping offers new opportunities and creates new challenges for web cartographers. The continual development of new technological solutions produces a fundamental tension: the more flexible and expansive web mapping options become, the more difficult it is to maintain fluency in the teaching and application of these technologies. We addressed this tension by completing a three-stage, empirical process for understanding how best to learn and implement contemporary web mapping technologies. To narrow our investigation, we focused upon education at the university level, rather than a professional production environment, and upon open source client-side web mapping technologies, rather than complementary server-side or cloud-based technologies. The process comprised three studies: (1) a competitive analysis study of contemporary web mapping technologies, (2) a needs assessment survey of web map designers/developers regarding past experiences with these technologies, and (3) a diary study charting the implementation of a subset of potentially viable technologies, as identified through the first two studies. The process successfully achieved the practical goal of identifying a candidate set of web mapping technologies for teaching web mapping, and also revealed broader insights into web map design and education generally as well as ways to cope with evolving web mapping technologies.

KEYWORDS: web mapping; UI/UX design; open web standards; interactive cartography; cartographic education; D3; Leaflet; Google Maps API; OpenLayers

INTRODUCTION

THE CURRENT PACE of innovation in web cartography is spectacular, with new releases of or substantial updates to web mapping technologies occurring almost daily (Haklay et al. 2008; Harrower 2008). However, the ever-evolving nature of web mapping technologies results in a fundamental tension for cartographers. On one hand, the increasing flexibility and interoperability of web mapping technologies create new opportunities for cartographers; we can do more now than ever. On the other hand, as technology evolves, so does the solution space from which cartographers can draw; it is growing ever more difficult to establish and maintain one's bearings within this increasingly complex array of technologies. In the following, the term *web mapping technologies* is used broadly to describe

the compilation of APIs, frameworks, libraries, services, etc., that altogether enable the creation and dissemination of web maps (Kraak and Brown 2001; Peterson 2003).

The research reported here is motivated specifically by (at the time of this writing) a broad transition in client-side web mapping away from standalone, proprietary technologies (e.g., Adobe Flash) and towards open technologies that leverage the HTML, CSS, SVG, and XML web standards (the Open Web Platform) and the JavaScript programming language. As a result, professional cartographers have needed to update their skillsets in response to shifting client requests, while educators have needed to rethink their approach to teaching on the Open Web



© by the author(s). This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

Platform. To this end, our research was designed to generate initial insight into the following four questions, ranging from practical questions approaching the current technological landscape to longer-term conceptual questions working towards a deeper understanding of web cartography:

1. What technologies currently are available for web mapping and how do they vary?
2. What are the important characteristics of web maps that should inform the selection of web mapping technologies?
3. How should web mapping be taught in higher education?
4. How can we better cope with continued evolution in web mapping technologies?

To address these research questions, we designed and executed a repeatable process following the discount, convergent approach recommended in the fields of usability engineering and user-centered design (Buttenfield 1999). The process was completed in three stages: (1) an initial competitive analysis study of contemporary web mapping technologies to evaluate said technologies according to

their relative strengths and weaknesses, (2) a needs assessment survey with web map designers and developers to solicit past experiences with and existing opinions of contemporary web mapping technologies, and (3) a diary study charting the design and development of prototype applications implemented with a selected subset of candidate web mapping technologies deemed potentially viable. The process itself was calibrated to address educational and learning objectives for beginning users, rather than all professional web mapping contexts, and to address open source, client-side technologies built on the Open Web Platform, rather than proprietary server-side or cloud-based technologies. However, the three-stage process is offered as a generic approach to keeping pace with evolving web mapping technologies than can be applied in both industry and government contexts, as well as repurposed for other forms of mapping technology.

We proceed with four additional sections. In the following section, we introduce important concepts related to historical and contemporary web cartography. We then describe parameters of each of the studies included in the three-stage process. We report the results of each study in the fourth section. We reflect on the meaning of these results with regards to our four research questions and offer concluding remarks in the fifth and final section.

BACKGROUND

ARGUABLY, WEB MAPPING IS as old as the web itself, a technological innovation that often is dated to 1991, with the public release of the World Wide Web, or 1993, with the launch of the first browser supporting a graphical user interface (Peterson 2008). The actual progression of technological developments required for the web as we know it today has a much longer history coinciding with the history of computing (Leiner et al. 1997; Leiner et al. 2009). The use of the term ‘web’ is intentionally distinct from the terms ‘Internet’ or ‘World Wide Web.’ The *Internet* refers to the series of interconnected computer networks facilitating the transfer of files, while the *World Wide Web (WWW)* refers to the corpus of interconnected documents shared over the Internet in a web browser (Tsou 2011); such a technical characterization of the Internet and WWW solely as a file or document sharing mechanism is commonly described as *Web 1.0*. In contrast, the term *web* is used today to describe the Internet as a platform upon

which otherwise disparate data and services are integrated for customized use (O’Reilly 2007); the characterization of the Internet as a virtual space with activities occurring within it is described as *Web 2.0*.¹

The design and use of web maps has evolved profoundly from the early days of the Internet (for a comprehensive review, see Donohue 2014). Following a Web 1.0 model, early attempts to disseminate maps online primarily were limited to digitally scanned, static maps (Cartwright 2008). Today, web maps commonly are *adaptive* in response to the use and user context (e.g., Reichenbacher 2003; Friedmannová et al. 2006), *interactive* to respond to user requests (e.g., Andrienko and Andrienko 1999; Roth 2013b), *mobile*, indicating the user’s location on the

1. The distinction between Web 1.0 and Web 2.0 is more about how the Internet is used, and not necessarily the technologies needed to support these uses.

map as they move through the depicted area (e.g., Clarke 2004; Meng et al. 2005), *multiscale* to view the world at every geographic location and extent (e.g., Brewer and Buttenfield 2007; Roth et al. 2011), and/or updated in *real time* to respond to geographic events and processes as they unfold (e.g., Boulos and Burden 2007; Goldsberry 2007). The suite of Web 2.0 tools and techniques facilitating design and development of online, dynamic maps collectively are described as *web mapping technologies* and the corpus of geographic information (and associated maps of these information) contributed online through web mapping technologies collectively is described as the *GeoWeb* (Leclerc et al. 2001; Haklay et al. 2008). While there are a growing number of important and timely treatments of the latter topic—particularly with regards to the social and ethical implications of volunteered contributions to the GeoWeb (e.g., Goodchild 2007; Crampton 2009; Elwood 2010; Harvey 2012; Wilson 2012; Sack 2013)—this research treats challenges related to the former: adapting to ever-evolving web mapping technologies.

Contemporary web mapping technologies, and web technologies generally, can be organized into one of three broad categories: (1) *server-side technologies* used to index and query geographic information from a centralized source or, increasingly, distributed sources (e.g., the cloud), (2) *client-side technologies* used to render and manipulate web maps of these geographic information in the user's browser, and (3) *web services* or similar intermediary scripts used to relay information requests between the client and server (Roth et al. 2008). Careful design and development of all three technologies are essential architecturally for an effective web map. When it comes to web mapping, however, we contend that the cartographer may be separated from the GIS technician by an increased contribution to client-side technology, as it is the client implementation that includes design decisions regarding the map representation itself (choices of projection, generalization, symbolization, typography, etc.), the *user interface* (UI) provided for manipulating this map, and the overall *user experience* (UX) achieved with the web map. The following discussion therefore is constrained to client-side technology, given the overall motivation of this research on the transition in client-side technologies (see details below) and this increasingly important role of UI/UX designer for the cartographer. As stated in the introduction, we expect our proposed method to remain useful when applied for evaluation of server-side and cloud-based technologies as well (for a broad review, see Peterson 2014).

Until recently, a tension existed in client-side web mapping between technologies built upon *open web standards*, which can be interpreted natively by web browsers and viewed for reuse and extension, and technologies leveraging browser *plugins*, which require installation of an additional software component into the browser to run a stand-alone executable embedded in the webpage (Hu 2008). Starting with the former, the World Wide Web Consortium (*W3C*: www.w3.org) governs specification of the majority of these web standards, a suite referred to as the *Open Web Platform*. W3C client-side standards commonly leveraged in web mapping include:

- Hypertext Markup Language (*HTML*), providing the structure of a webpage and allowing for the organization, layout, and styling of its content, as well as the interconnection of webpages through hyperlinks.
- Extensible Markup Language (*XML*), a more flexible text-based markup language than HTML typically used as a data format for loading and manipulating information in the browser; proprietary variants of XML include Esri's ArcXML and Google's KML.
- Scalable Vector Graphics (*SVG*), an open variant of XML for defining and rendering vector shapes and text in the browser.
- Cascading Style Sheets (*CSS*), affording separation of content and styling in web design through definition of hierarchical and reusable style rules, which can be defined to account for varying viewing contexts, allowing the webpage design to be responsive to display devices.
- the Document Object Model (*DOM*), a platform-independent specification for creating and manipulating HTML and XML objects within the browser; the DOM enables manipulation of objects in the browser using *JavaScript*², a popular browser programming language used for defining client-side business logic, and the associated JavaScript Object Notation (*JSON*).

2. JavaScript was originally developed by Netscape and continues to be maintained as part of the open source Mozilla project, although is officially trademarked by Oracle. Although not a standard maintained by the W3C, JavaScript typically is considered part of the Open Web Platform.

While the earliest web maps made use of the first-generation versions of these open web standards (Peterson 2008), the use of plugin technologies with compiled scripts for web mapping grew in popularity in the late 1990s and into the early 2000s, in part due to their perceived performance and stability. Notable plugin solutions for web mapping included Tcl/Tk applications (e.g., Dykes 1996; Dykes 1997; MacEachren et al. 1999; Masters and Edsall 2000), enhanced QuickTime videos (e.g., Al-Kodmany 2001; Cammack 2003; Schwertley 2003), and Java applets (e.g., Herzog 2003; Tsou 2004; Hardisty and Robinson 2011). Flash Player was the most popular plugin for web mapping in the mid-to-late 2000s (Muehlenhaus 2013), which ran Shockwave Flash³ (*SWF*) executables produced first from Macromedia's Director (raster) or Flash (vector) authoring environments and later from Adobe's Flash, Flex, or Flash Catalyst authoring environments.⁴ At one point in time, 98% of all personal computing devices had Flash Player installed (Jenny et al. 2008). Web applications, maps or otherwise, relying on plugin client technologies commonly are described as *Rich Internet Applications* (Tsou 2011).

There were several important advantages to browser plugins that justified their popularity for web mapping at this time. First, plugin technologies afforded greater consistency across browsers and across platforms (if a plugin version was available for the platform). Cross-browser and cross-platform compatibility were particularly important during the 'browser wars' of 1990s (Peterson 2005), but remain frustrating aspects of client-side web map development today.⁵ Second, the Shockwave Flash format, and other plugin executables, compiled vector graphics into relatively small binary files, greatly expanding the potential for high-quality, vector-based web mapping at a time when open web standards like SVG were cumbersome to load, render, and manipulate due to bandwidth and hardware limitations (Hu 2008; Lienert et al. 2012). Finally, use of a plugin-based technology typically meant development in a single authoring environment using a single scripting language, allowing for greater ease in learning the web mapping technology and maintaining source code across projects. Comparatively, open web standards are considered to have a steeper learning curve for be-

ginners given the number of markup/scripting languages and file formats used across a single web mapping project (Woodruff 2011). In this regard, Adobe Flash was particularly kind to cartographers, as it allowed for tight integration with Adobe Illustrator and Photoshop (i.e., what cartographers already use: graphic design software) and afforded an inherently visual design environment (i.e., how cartographers already think: visually).

Despite their advantages, use of plugin-based technologies for web mapping waned in the late 2000s in favor of open web standards (Pulsifer et al. 2008), a transition that has been realized fully over the past several years. There were at least three drivers for this wholesale transition. First, the introduction of Google Maps in 2005 pioneered the use of AJAX for web mapping. Asynchronous JavaScript and XML, or *AJAX*, is an approach to using JavaScript and open web standards that allows for client-server communication without requiring a webpage refresh (Garrett 2005; Tsou 2005); such communication required plugin-based technology prior to the advent of AJAX. The use of AJAX in Google Maps and other web map services afforded continuous panning and zooming of map tiles that were pre-processed and loaded on-demand, giving rise to the now ubiquitous *slippy map* (Haklay et al. 2008; Haklay and Weber 2008). While rendering and serving custom tiles was prohibitively expensive until only recently (Peterson 2011), cartographers could create *mash-ups* of their own geographic information and tilesets from Google Maps and other commercial map services using a provided application programming interface (*API*), which exposes a subset of a proprietary map service's functionality for open use (Plewe 2007; Tsou 2011). At the time of this writing, most popular client-side web mapping APIs are provided in the JavaScript language and leverage open web standards.⁶

The second driver towards open web standards was the recent improvement of telecommunications bandwidth and hardware, and the associated increased consumption of maps on mobile devices such as smartphones and tablets, making continuous network connectivity more accessible (Meng et al. 2005). In April 2010, Apple Inc. announced that it would not support the Flash Player plugin on its iPhone and iPad mobile devices, citing increased openness, improvement of web standards, the interconnectedness

3. Later rebranded as 'Small Web Format.'

4. Adobe acquired Macromedia in 2005.

5. In particular, many web clients still rely on older versions of Internet Explorer, which did not comply with many open web standards until version 9.0.

6. Google Maps had separate APIs written for JavaScript and ActionScript, the scripting language used for Flash development. The Google Maps API for Flash was deprecated in 2011.

of reliability/security/performance, improved battery life, support for touchscreens, and the politics surrounding third-party control over app development as reasons for the move towards HTML5 (Jobs 2010). As a result, development of mobile maps using Flash Player waned to the point of extinction (Muehlenhaus 2013) at the same time that a vibrant research and development initiative emerged around responsive web design using the Open Web Platform (Gardner 2011). *Responsive web design* describes an approach to using open web standards that modifies the layout and styling of content according to the display device and user context (Marcotte 2010). Despite relevant work on the topic of adaptive cartography (Reichenbacher 2003), research and practice on *responsive cartographic design* remains in its infancy.

The third driver away from plugin-based technologies was the formation of open source interest groups across the web mapping community, such as the Open Geospatial Consortium (OGC; www.opengeospatial.org) and Open Source Geospatial Foundation (OSGeo; www.osgeo.org/home). The OGC was founded in 1994, just one year after formalization of the W3C, with the mission of promoting standards across the spectrum of geospatial technologies. Notable contributions of the OGC to web mapping include the web mapping services (WMS) and web feature services (WFS) technical specifications, which leverage

the SVG open web standard (Peterson 2012). While the OGC primarily focuses on standardization, OSGeo plays a broader role in the promotion and advancement of open source geospatial data, software, and tools. Among the most important outreach activities of OSGeo is the Free and Open Source Software for Geospatial (FOSS4G; foss4g.org) conference, meeting annually since 2006. Finally, there is an informal community of web map designers and developers—arguably numbering in the thousands if not tens of thousands—contributing to the missions of the OGC and OSGeo through the sharing and maintenance of source code, web map examples, and tutorials. Due in large part to the efforts of this open web mapping community, there are now a multitude of open source web mapping technologies that afford a sophistication in cartographic design previously only possible through use of proprietary plugin technologies (Pulsifer et al. 2008). These open source technologies offer the promise of rapid advancement due to the scale of developer collaboration and their ability to be creatively manipulated to suit specific application needs. Because of their current dynamism and increased relevance of open source on contemporary web map design, we have constrained the following treatment of client-side web mapping to open source technologies, reviewing several proprietary technologies for comparison only.

METHODS

WE DESIGNED AND EXECUTED a three-stage process in order to characterize and push our way into the current landscape of open source web mapping technologies. Design of the process followed the *convergent methods paradigm*, which prescribes administration of multiple, often qualitative methods (Buttenfield 1999). Each study then is conducted in a *discount* manner (e.g., leveraging secondary sources, recruiting only a small number of participants) to ascertain input and feedback quickly (Nielsen 1993). Reliability of the project as a whole is maintained through triangulation of insights generated across the studies.

There is a growing volume of research in the context of user-centered design that follows a discount, convergent approach to include target end users in the design and development of interactive maps and map-based visualizations (e.g., Slocum et al. 2003; Fuhrmann et al. 2005; Robinson et al. 2005). Here, we leverage a discount, convergent

process to understand the experience of web map designers and developers, rather than the ultimate users of these maps. We triangulated insights across three studies in total: (1) a competitive analysis of existing web mapping technologies, (2) a needs assessment survey with web map designers and developers, and (3) a diary study tracking the implementation of the same web map using a candidate subset of technologies identified from the first two studies. The following subsections describe the method design of each of the three studies included in the process.

COMPETITIVE ANALYSIS STUDY

We began the process by completing a competitive analysis of contemporary open source web mapping technologies. A *competitive analysis study* is a systematic, critical comparison of a suite of related tools or technologies based on their relative merits (Nielsen 1992). The competitive

analysis method is appropriate when evaluating new and emerging tools/technologies, in formative stages of evaluation, and in situations when there are a large number of competing technological alternatives. When the tools or technologies are compared according to established theoretical frameworks, a competitive analysis study is effectively a content analysis of secondary sources, common to archival research in social science (Roth et al. 2015). The competitive analysis represented the widest scoping stage in the process, as it assumed little or no existing knowledge of contemporary technologies and sought to characterize the complete solution space for open source web mapping. Even with existing knowledge, however, we recommend completing a new competitive analysis study at the start of each process cycle, given the pace of technological change in web mapping.

We collected the primary webpages (i.e., the secondary sources included in the content analysis) for open source web mapping technologies over a two-week period in the spring of 2012, making use of keyword searches, popular blogs, and social media for webpage collection. In total, thirty-five (n=35) web mapping technologies with some degree of openness were identified for the competitive analysis during this timeframe.⁷ Two project members then independently ‘coded’⁸ the technologies according to the supported *representation* techniques for graphically encoding information and the supported *interaction* techniques for building user interfaces around the representation. Representation techniques included support for different basemaps, vector overlays, and linked graphics/charts, as well as support for common thematic map types (Slocum et al. 2009). Interaction techniques included support of interaction operators, or generic kinds of interactive functionality available for manipulating maps and other visualizations (Roth 2012; Roth 2013a), as well as support for mobile and location-aware web maps. Twenty-seven (n=27) ‘codes’ were used in total between the representation and interaction categories; Table 1 lists and defines the representation and interaction codes used for the competitive analysis.

7. Over a dozen additional technologies have been released since the initial coding in the spring of 2012. In addition, several of the technologies evaluated have undergone significant changes and upgrades, and some have been deprecated.

8. ‘Code’ in this sense refers to a qualitative topic by which each web mapping technology is assessed, rather than in the sense of the ‘source code’ comprising these web mapping technologies that is manipulated and appended by the developer.

We instructed the project team coders to apply codes based solely on the documentation included in the collected webpages (i.e., what the webpage promised of the technology) without experimenting with the technology itself. We also instructed the coders to follow a four point ordinal scale in their coding: (1) supported, (2) known work-around, (3) requires hack, and (4) not possible, with the average score between the coders ultimately used for a reliable comparison across technologies. These coding instructions followed the discount, convergent approach in the process, producing a broad understanding of the solution space in the first stage of the process.

ONLINE NEEDS ASSESSMENT SURVEY

Following the competitive analysis study, we administered an online needs assessment survey with web map designers and developers. We included the survey as the second step in the overall process in order to acquire rapid feedback about technologies collected in the competitive analysis from designers and developers outside of the project team. The online survey acted as a *needs assessment study*, as the purpose of the survey was to elicit past experiences with the collected technologies as well as to identify future or currently unmet web mapping needs (Wiggins and French 1991). We chose the online survey format over interviews or focus groups given the discount, convergent approach to the overall process, enabling rapid feedback from a diverse set of designers/developers in a distributed manner.

Twenty-one (n=21) web map designers and developers participated in the online needs assessment survey in the spring of 2012. Participation was limited to individuals who either develop web maps as part of their work responsibilities or supervise individuals who develop web maps. Table 2 describes the frequency with which the participants used geographic information, made print maps, and developed web maps as part of their daily work.

We divided the survey question protocol into three sections: (1) current use of the web mapping technologies identified in the competitive analysis, (2) important qualities of web mapping technologies we should consider when selecting a technology, and (3) approaches to keeping pace with evolving web mapping technologies. The non-biographical portion of the survey included 12 questions in total, with four categories of Likert scale questions (each category including multiple Likert scales) and eight free

REPRESENTATION		
1	Map vs. Imagery	load different basemap tiles, such as road map, satellite imagery, etc.
2	Basemap Styling	adjust the styling of the basemap
3	Tile Rendering	generate and serve custom maps as tiles
4	Vector Overlays	draw and overlay additional vectors, including points, lines, and polygons
5	Choropleth	generate a choropleth map
6	Proportional Symbol	generate a proportional symbol map
7	Dot Density	generate a dot density map
8	Isoline/Surface	generate an isoline or surface map
9	Flow	generate a flow map
10	Cartogram	generate a cartogram
11	Bivariate/Multivariate	depict two or more statistical variables on the map
12	Animation	animate the map over a time series
13	Graphics/Charts	add additional information graphics or charts to the map
INTERACTION		
1	Arrange/Linked Views	manipulate the layout of the map and linked views
2	Reexpress	change the displayed map type
3	Sequence	generate an ordered set of related maps or change the map from the sequence that is shown
4	Resymbolize	change the design parameters of a map without changing the map type
5	Overlay/Toggle	adjust the feature types included in the map
6	Reproject	change the map projection
7	Pan	change the geographic center of the map
8	Zoom	change the scale or resolution of the map
9	Filter	alter the map to remove map features that do not meet one or a set of user-defined conditions/constraints
10	Search	alter the map to add/indicate a particular location or map feature of interest
11	Retrieve	request specific details about a map feature of interest
12	Calculate	derive new information about a map feature of interest
13	Mobile Support	support for viewing and interacting with the map on a mobile device
14	Location Aware	support for collecting and mapping information about the user's location

Table 1. The representation and interaction codes used to compare the collected suite of open web mapping technologies.



Table 2. The frequency with which participants in the online needs-assessment survey use geographic information, make print maps, and develop web maps as part of their daily work.

response. We designed the online needs assessment survey to take no longer than 15 minutes to complete.

DIARY STUDY

We combined insights from the competitive analysis study and online needs assessment survey to identify a final

subset of four candidate technologies that we suspected could meet the needs of a contemporary curriculum in web cartography. We then evaluated the four candidate technologies using a *diary study*, a variation of participant observation that requires participants to ‘self-observe’ as they complete an activity (Marsh and Haklay 2010). In the diary study, participants developed a case study web

map using one of the four candidate technologies and recorded their progress in an online journal. We selected a diary study for the third step in the process because it provided the deep development experience with a given technology needed to properly assess its advantages and limitations, but did so in a convergent, discount manner by relying on prior methods to reduce the total number of technologies under consideration. As described in the next section, we selected D3, the Google Maps API, Leaflet, and OpenLayers for inclusion in the diary study.

Four students representative of the targeted user group were recruited to complete an example web mapping scenario during the summer of 2012, each using a different candidate technology. A fifth student completed the same web mapping scenario with all four technologies to improve reliability across participants. As a result, there were eight (n=8) diaries total, two for each of the candidate technologies. We did not allow participants to integrate multiple technologies into their web map solutions, nor allow collaboration across participants, in order to identify the limitations of each technology in isolation. While the limitation on group work unrealistically constrains an important aspect of classroom learning, it was further justified by our interest in generating insight across the four research questions enumerated above, rather than focusing solely on the third research question (understanding how students best learn a new web mapping technology). All participants had taken one introductory course on cartographic design and one advanced course on web mapping, and thus were familiar with the requirements included in the web mapping scenario from a conceptual statement. Participants were required to complete lynda.com's video tutorials on HTML, CSS, and JavaScript before beginning the diary session, resulting in a one day engagement with the Open Web Platform before introduction to the assigned web mapping technology. No further training on the assigned web mapping technology was given, allowing us to identify important web resources (e.g., documentation, examples, forums, etc.) for each candidate technology, and to learn how beginners integrate such learning materials into their design and development workflows.

At the start of the diary study, we introduced participants to a web mapping scenario, presented as a hypothetical client request for a web map depicting energy consumption by country over the past 30 years and included a requirements document outlining the project scope. We provided the energy time series dataset to the participants as a CSV

file. As with the competitive analysis, requirements for the web mapping scenario were split between representation and interaction techniques. Representation requirements covered elements of effective design for classed choropleth and graduated symbol maps, including traditional cartographic design topics such as aesthetics, classification, typography, and visual hierarchy as well as emerging cartographic design topics enabled by digital media, such as animation, linked information graphics, and visual storytelling. The specific representation requirements included in the diary study deviated from the representation codes included in the competitive analysis due to the narrowed focus upon only two thematic map types. Interaction requirements were specific to the interaction operators included in the competitive analysis, and included one additional requirement for interface design aesthetics. Twenty-four (n=24) requirements in total were included in the web mapping scenario; Table 3 lists and defines the representation and interaction requirements used for the diary study.

We gave participants a total of 40 hours of development time to complete as many of the twenty-four requirements as possible, mimicking constraints of a standard work week. Given the lack of familiarity with the assigned technology, we did not expect participants to implement all requirements within the provided time period and instead instructed participants to implement what they considered to be 'easy' requirements before moving onto more difficult ones. Therefore, the requirements that ultimately were implemented by the participants indicate functionality that likely is natively supported by the technology, rather than functionality needing a work-around or hack.

We required participants to log a diary entry every hour across the 40-hour period. Within each diary entry, participants were asked to describe: (1) the requirement(s) they implemented in the past hour, (2) key frustrations or breakthroughs in the past hour, and (3) their current satisfaction with the web mapping technology drawing from a provided list of 125 emotions, derived from the list of moods available at github.com/hazbo/mood-swing2. Regarding the latter, we selected the larger list of moods—rather than more terse taxonomies of affective or emotional experiences (e.g., Plutchik 1980; Feldman-Barrett and Russell 1998)—to give participants greater flexibility and precision in describing their emotional state; the moods were coded according to their valence (positive, neutral, and negative) for subsequent analysis. It is

REPRESENTATION		
1	Classed Choropleth	generate a classed choropleth map
2	Graduated Symbol	generate a graduated symbol map
3	Animation	animate the map over the included time series
4	Typography	label map features following typographic conventions
5	Classification	use an equal interval classification scheme for both maps
6	Legend	dynamically redraw a map legend to match the displayed map type
7	Highlighting	include a highlighted variant of each map feature to indicate selection
8	Information Graphic	include a line graph showing the signature of the selected country in comparison to the United States and the median value for the year
9	Visual Hierarchy	style the basemap to produce a strong visual hierarchy
10	Storytelling	provide a title and supplementary text to introduce the map subject and purpose
11	Cartographic Design Aesthetics	customize the look and feel of the map itself to fit the scenario
INTERACTION		
1	Reexpress	change the displayed map type between classed choropleth and graduated symbol
2	Sequence	include standard VCR controls (play, stop, step, back) to control the animation
3	Resymbolize	change the number of classes used for the classed choropleth or graduate symbol map
4	Overlay/Toggle	toggle between a road map and aerial image for the basemap
5	Reproject	set the map projection to equal area for the classed choropleth map and conformal for the graduated symbol map
6	Pan	change the geographic center of the map
7	Zoom	change the scale and/or resolution of the map
8	Filter	filter the map according to the attribute range using a two-thumb slider
9	Search	search for a specific country
10	Retrieve	highlight a probed map feature and activate an associated information window with details about the feature
11	Calculate	dynamically calculate deviation of the map feature from median (i.e., percentile)
12	Link	coordinate retrieve on the map with the line graph to show the selected feature on both graphics
13	Interface Design Aesthetics	customize the look and feel of the interface to the map to fit the scenario

Table 3. *The representation and interaction requirements of the diary study.*

important to note that insight from the field of curriculum and instructional design suggests that students learn best when pushed towards the point of frustration, but without reaching this point (Bampton 2011). Thus, an

overwhelming positive valence may not reflect an optimal learning experience; however, an overwhelming negative valence likely does indicate a suboptimal learning experience.

RESULTS

COMPETITIVE ANALYSIS STUDY

RESULTS OF THE COMPETITIVE ANALYSIS study are illustrated in Table 4. In the matrix, the darkest blue shading indicates a representation or interaction technique that was coded as ‘supported’ by both coders and the white shading indicates a technique that was coded as ‘not possible’ by both coders.

When interpreted horizontally, the matrix summarizes the supported functionality for each of the 35 reviewed web mapping technologies. The competitive analysis revealed a basic distinction between specialist web mapping technologies designed to support a small subset of specific functions (e.g., Cloudmade Editor, Mapnik, Modest Maps) and multi-purpose web mapping technologies designed

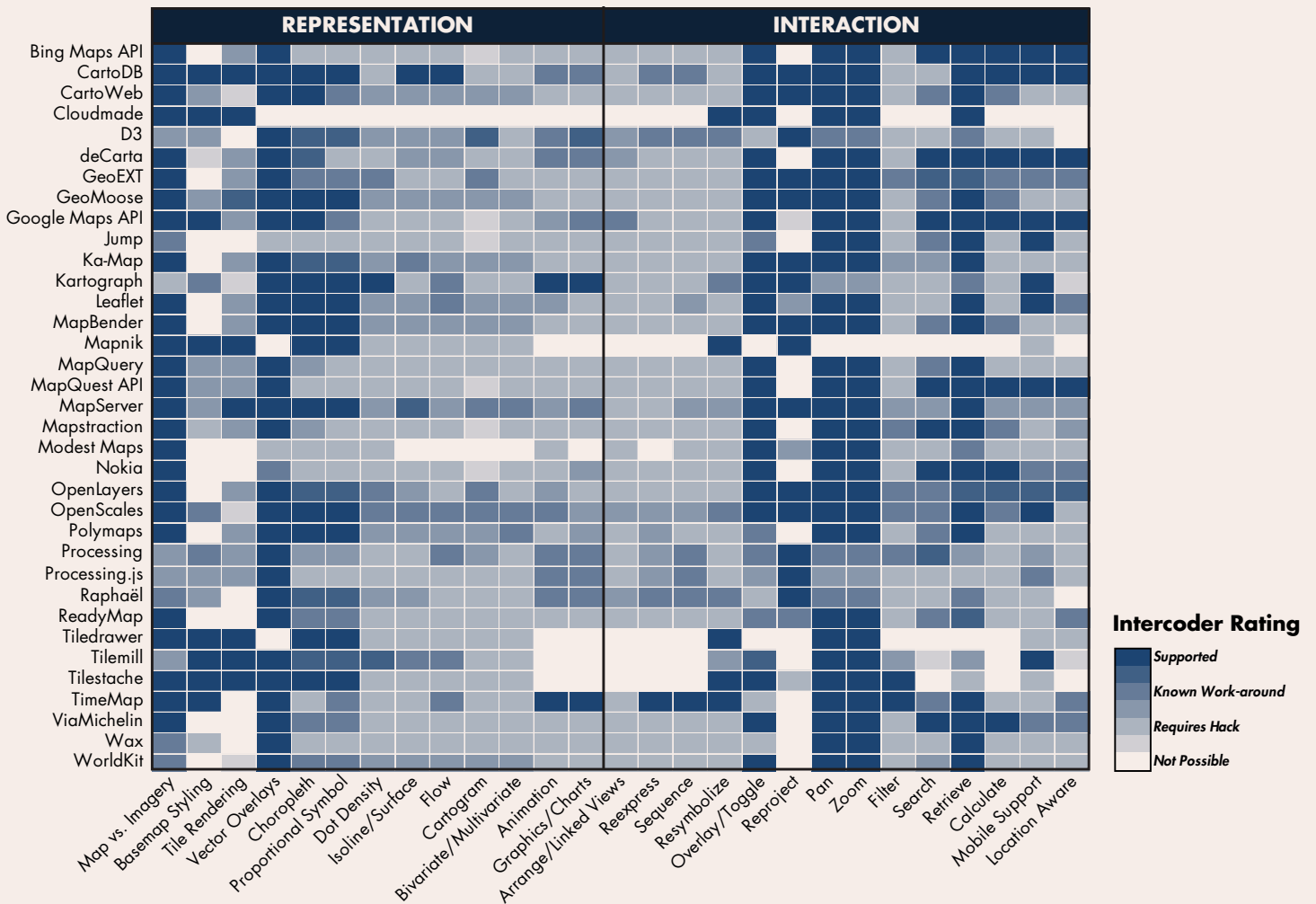


Table 4. Results of the competitive analysis study. Collection and coding was completed in the spring of 2012; therefore, the matrix is no longer complete nor accurate, although arguably it never can be, given the speed of technological advancements in web mapping. The matrix does provide a snapshot in time of web mapping technology that is useful for understanding general patterns and emerging trends in web map design.

to support numerous functions (e.g., CartoDB, D3, the Google Maps API, Leaflet, MapServer, OpenLayers/OpenScales). Individual technologies generally fell into one of the following categories: (1) *frameworks* (10/35; 28.6%) providing a full stack of client- and server-side technologies (e.g., GeoMoose, MapServer, Processing), (2) *open libraries* (14/35; 40.0%) supporting client-side map rendering (e.g., D3, Leaflet, OpenLayers), (3) *closed APIs* (6/35; 17.1%) exposing a subset of functionality for creation of web map mashups (e.g., the Bing Maps API, the Google Maps API, the MapQuest API), and (4) *tile rendering services* (5/35; 14.3%) facilitating the rendering and serving of basemap tiles (e.g., Cloudmade Editor, TileMill, TileStache). The large majority of the reviewed technologies (28/35; 80.0%) leveraged JavaScript as the base programming language, with four (4/35; 11.4%) exclusively leveraging CSS or the CartoCSS variant used for

tile rendering, one (1/35; 2.9%) leveraging Java, one (1/35; 2.9%) leveraging PHP, and one (1/35; 2.9%) leveraging ActionScript.

From the competitive analysis by technology, we identified open libraries implemented in JavaScript as the most suitable technological form for teaching web mapping in the context of higher education. Open libraries can be combined flexibly with other non-mapping JavaScript libraries (e.g., jQuery) and can be extended more easily to implement non-natively supported representation and interaction functionality, two advantages that open libraries hold over closed APIs. Full stack frameworks, while more powerful and feature-complete than open libraries, are less approachable for a single, semester-long course and require background on server-side databases outside the scope of a course on interactive cartography and geovisualization.

We instead recommend instruction of frameworks that integrate client- and server-side technologies in an advanced course on geocomputing and web GIS, taken towards the end of a program in cartography and GIS after students have been exposed to foundational concepts and skills. Finally, the opportunity to teach and practice interaction design is limited with tile rendering services in comparison to the other forms of technologies. As an auxiliary outcome of the competitive analysis, we now teach the TileMill tile rendering service as a laboratory exercise in an advanced graphic design course in cartography, rather than our course on interactive and web-based mapping. It is important to note that our decision to prioritize open libraries was specific to our emphasis on education rather than production, and should not be interpreted as a statement of superiority of open libraries over other forms of web mapping technology.

When interpreted vertically, Table 4 provided us with a snapshot of trends in contemporary web map design. Widely supported representation functionality included custom vector overlays (29/35 supported; 82.9%), loading of map versus imagery basemaps (26/35; 74.3%), and choropleth (19/35; 54.3%) or proportional symbol (16/35; 45.7%) thematic maps. Overall, the competitive analysis suggested a general focus on reference mapping over thematic mapping in existing web map technologies, as most of the reviewed technologies required an advanced, custom solution to implement advanced thematic map types beyond the choropleth and proportional symbol techniques. The lack of support for advanced thematic mapping is a real and significant gap between contemporary web mapping practice and traditional cartographic scholarship that should be addressed as web design and cartographic design continue to collide. Basemap styling and tile rendering exhibited the greatest variation in support across technologies; both were supported by eight (8/35; 22.9%) technologies, but not possible in thirteen (13/35; 37.1%) technologies. This variation was explained by inclusion of tile rendering services in the competitive analysis, rather than restriction to frameworks, libraries, and APIs.

Widely supported interaction functionality included panning (29/35; 82.9%), zooming (29/35; 82.9%), retrieval of details using an information window (25/35; 71.4%), and overlay of context layers (24/35; 68.6%). Arguably, these four interaction operators (overlay, pan, retrieve, zoom) along with a multiscale reference basemap have coalesced to define the prototypical web map, an extension

to the combination of panning and zooming explicit in the colloquial use of ‘slippy map.’ Tracking the evolution of the prototypical web map is useful for cartographers, as it exposes the expectations of non-specialist web map users and reveals potential gaps in contemporary design solutions due to technology constraints. Such gaps included support for reexpress (not supported natively by any technology; 0.0%), filter (2/35; 5.7%), and calculate (8/35; 22.9%). Reexpress and filter are considered important for exploratory visualization, while calculate is essential for advanced WebGIS. Dynamic reprojection exhibited the greatest variation across technologies, which was supported by sixteen (16/35; 45.7%) technologies but not possible in fifteen (15/35; 42.9%). Many of the technologies supporting reprojection were limited to a small set of cylindrical projections, further defining the prototypical web map as a reference map served as raster tiles. Finally, eleven (11/35; 31.4%) of the technologies natively included responsive mobile support, but only six (6/35; 17.1%) were location aware. Such a finding suggested that cartographic design for mobile has garnered some attention in client-side web map design, but the implementation of location-based services using the Open Web Platform has been limited to date.

ONLINE NEEDS ASSESSMENT SURVEY

As described in the third section, we organized the online needs assessment survey around three topics related to web mapping. The first section of questions addressed existing use of the web mapping technologies that we collected through the competitive analysis. Table 5 presents the frequency that survey participants were aware of or had used the collected set of web mapping technologies. We listed three proprietary technologies in this section of the online needs assessment survey—ArcServer, Adobe Flash, and Adobe Flex—as a baseline against which to compare the collected set of open source web mapping technologies, resulting in evaluation of 38 technologies in total.

Survey participants had used just a subset of the collected technologies. Only the Google Maps API was used by a majority of participants in the past year (11/21; 52.4%), with OpenLayers (9/21; 42.9%), ArcGIS Server (8/21; 38.1%), and Adobe Flash (6/21; 28.6%) used in the past year by a large minority. There were several technologies that numerous participants were aware existed, but had never used themselves, most notably the MapQuest (17/21; 81.0%) and Bing Maps (15/21; 71.4%) APIs, the

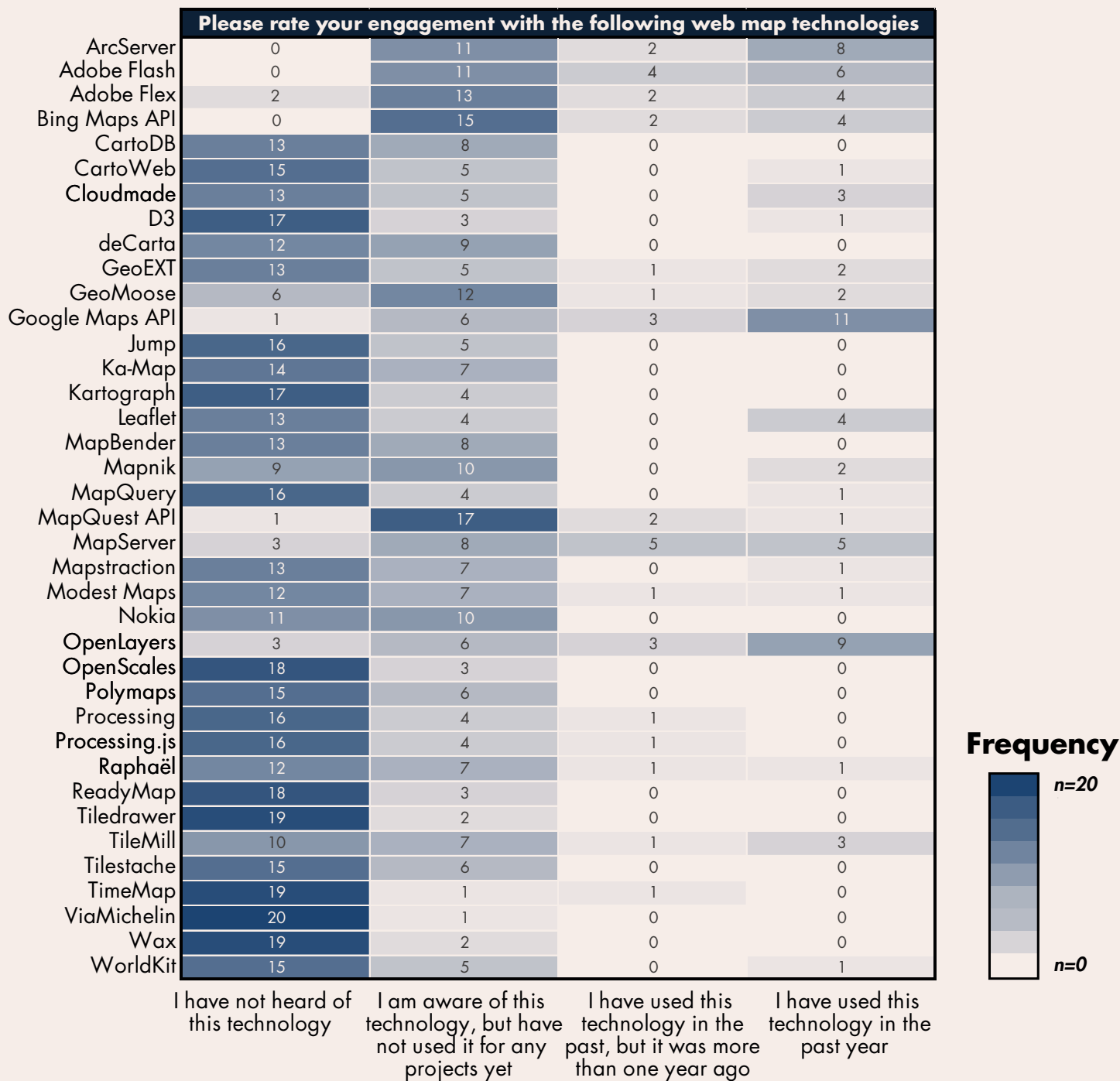


Table 5. The level of engagement with the set of web mapping technologies gathered through the competitive analysis study. The proprietary technologies ArcServer, Adobe Flash, and Adobe Flex are added to the top of the table to provide a comparison against open-source technologies.

GeoMoose framework (12/21; 57.1%), and the ArcServer (11/21; 52.4%), Adobe Flash (11/21; 52.4%), and Adobe Flex (13/21; 62.0%) proprietary technologies. Overall, participants had not heard of the majority of the surveyed technologies, including technologies that have gained in popularity across the cartographic community since administering the survey, such as D3 (17/21; 81.0%),

Processing/Processing.js (16/21; 76.2%), CartoDB (13/21; 62.0%), and Leaflet (13/21; 62.0%). Such a shift in awareness signals the fast pace of technological change in web mapping. This finding also justified our pairing of the competitive analysis study with the online needs assessment study, as reliance on an internal survey alone would have limited discussion to a small subset of available

technologies (e.g., the Google Maps API, Open Layers, ArcServer) not fully representative of the trajectory of web mapping at the time.

Open-ended comments regarding the technologies that participants continued to leverage versus those they completely abandoned revealed broad awareness of the technology transition in web mapping underway at the time. Overall, participants acknowledged the move towards the Open Web Platform and JavaScript, with one participant stating, “In testing technologies for next generation of web apps, we’re quickly moving toward primarily JavaScript-based frameworks” and a second adding, “I am going to transition to JavaScript.” This discussion provided further justification for narrowing our focus to JavaScript-based technologies implemented on the Open Web Platform. Looking towards the future, several participants predicted a move away from closed APIs and towards full-stack frameworks or client-side mapping libraries. Following such sentiment, one participant indicated that his or her institution does not “employ programs like Bing and Google Maps API unless students are working on navigational aids,” and a second stated, “I suspect the Google Maps API is on its way out.” A third participant gave justification for the move away from closed APIs, stating that the “advancement of many of these libraries/frameworks [provides] highly-customizable standard mapping interface components and interaction behaviors.” Thus, responses to the first section of questioning revealed a general preference for openness and extensibility, but an overall poor awareness of the emerging frameworks and libraries that could be used in place of closed APIs and proprietary technologies.

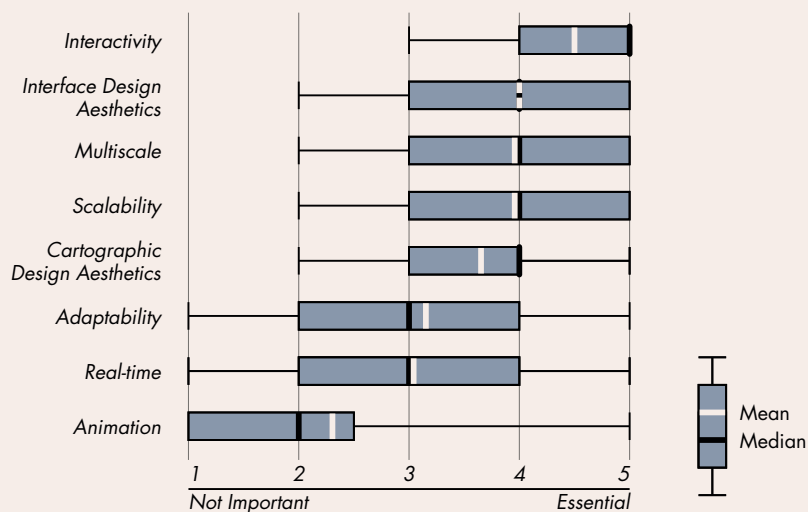
The second section of questions solicited feedback about the qualities of web mapping technologies that should be considered when selecting an appropriate technology or set of technologies. Figure 1 presents a series of box plots depicting participant responses to a series of five-point Likert ratings ranging from ‘not important’ to ‘essential.’ The box plots are organized according to three qualities of web mapping technologies: (1) design characteristics of the resulting web map, (2) technical considerations, emphasizing constraints in applying the technology associated with hardware or software, and (3) practical considerations, including other non-functional constraints when applying the technology.

Participants rated interactivity as the most essential characteristic of web maps that should be supported by a web mapping technology (mean=4.50; median=5), with no participant rating interactivity lower than a ‘3’ (‘important’). Such a finding justified inclusion of both representation and interaction functionality in the competitive analysis coding (Table 4), and reflected the growing importance of UI and UX design to web mapping specifically, and the discipline of cartography broadly. Participants also listed interface design aesthetics (mean=4.00; median=4), multiscale (mean=3.95; median=4), and scalability (mean=3.95; median=4) as important aspects of web map design that must be supported in the underlying technology. Participants rated animation as the least essential property of web maps to consider when selecting a technology (mean=2.30; median=2), a surprising finding given the substantial body of research on animation in the cartographic literature. Overall, Likert scale ratings on web map characteristics suggested an increase in importance on user-driven display changes (i.e., interactivity) and a decrease in importance on system-driven display changes (i.e., animation and real-time updates) in contemporary web map design.

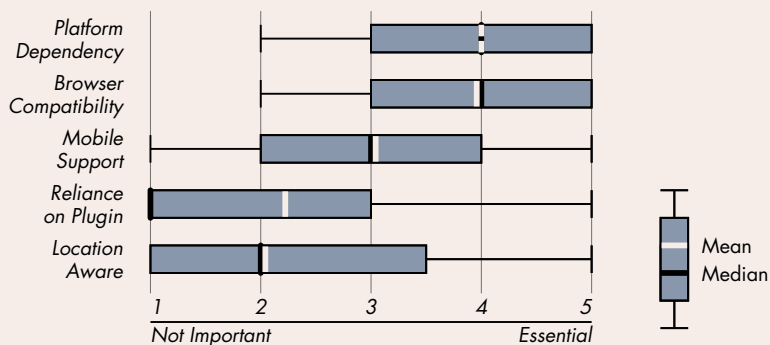
Participants rated platform dependency as the most important technical consideration for web mapping (mean=4.00; median=4), directly followed by browser compatibility (mean=3.95; median=4). As reviewed above, cross-browser and cross-platform compatibility were major advantages to using plugin-based technologies for web mapping through the mid-2000s, and the sharp decline in cross-platform compatibility, specifically mobile support, was an important driver away from plugins like Flash Player in the early 2010s. Participant responses regarding technical considerations indicated that cross-browser and cross-platform compatibility remain a high priority in web mapping, and provides further justification for leveraging frameworks and libraries that can be used in combination with other open libraries that enable cross-browser and cross-platform compatibility. Location awareness was rated as the least important technical consideration (mean=2.05; median=2), providing further evidence that implementation of in-browser, location-based services was not common at the time of conducting the survey.

Finally, participants rated maintenance/stability as the most important practical consideration when selecting a web mapping technology (mean=4.05; median=4). Poor long-term maintenance and source code instability

A. Please rate the importance of the following characteristics of web maps:



B. Please rate the importance of the following technical considerations:



C. Please rate the importance of the following practical considerations:

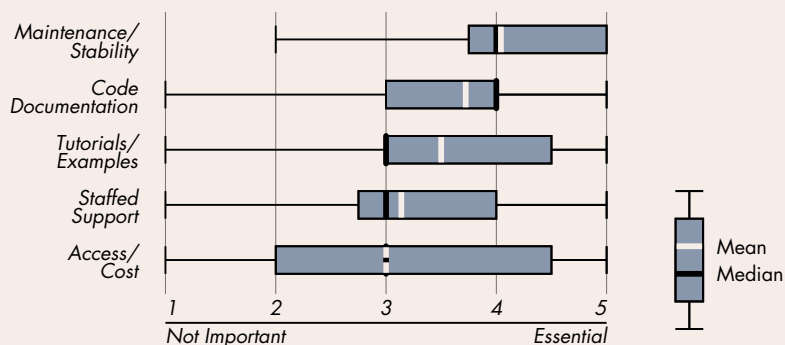


Figure 1. The importance of different qualities of web mapping technologies when choosing an appropriate technology or set of technologies: (A) web map characteristics, (B) technical considerations, and (C) practical considerations. The qualities are listed vertically in the order of descending mean value.

historically have been criticisms of open source technologies, but are improving as the open source web mapping community strengthens and matures, adding clout to the enforcement of open web standards. High quality code documentation (mean=3.70; median=4) and tutorials/

examples (mean=3.50; median=3) also were listed as important practical considerations, both of which aid in learning a new technology as well as keeping one’s skills up-to-date as the technology evolves.

Opinion was split across surveyed participants regarding the value of open source technology. Access/cost was rated as the least important practical consideration (mean=3.00; median=3), a finding that contrasts with the above participant comments about transitioning to open source frameworks and libraries. Responses to the access/cost Likert scale revealed a divergence in opinion regarding open source web mapping technology, with nine (9/21; 42.9%) participants listing access/cost as ‘not important’ or only ‘somewhat important’ and eight (8/21; 38.1%) participants listing access/cost as ‘very important’ or ‘essential.’ One participant shed light on this bimodal distribution in an open-ended response, stating, “I personally think open source is a great ideal... but not as important as people make it sound.” This participant went on to state that “there are many good open source products...there are many good closed products too. I will use whatever software is most user-friendly and easily adaptable. I don’t care if it is open or closed.” A second participant stated, “Increasingly it is a blurry line between commercial, open source, [and] cloud-based options and hybrid applications utilizing all of these are a growing trend.” Therefore, it is important to remember that good

design matters more than novel tools, and that a robust cartography curriculum should introduce students to a representative portfolio of industry-standard technologies, open and proprietary. Because of this feedback, we decided to include one closed-API in the diary study to enable consideration across different degrees of openness.

The third and final set of questions in the online needs assessment survey solicited approaches used by the participants to keep pace with evolving web mapping technologies. The most common strategy listed for experimenting with new web mapping technologies was the completion of a pilot study or proof-of-concept prototype (6 participants), followed by working through posted examples and tutorials (3), reading other developers' experiences on forums and web blogs (2), and directly reviewing the available documentation (2). In particular, completion of a pilot study falls in line with the process we describe in this paper, as the process represents a structured, repeatable approach to prototyping. Participants indicated that before they are willing to experiment with a technology, they need details about its cost (3 participants), documentation (3), examples and tutorials (3), server requirements (3), development environment (3), functional capabilities (2), base programming language (2), security (1), stability (1), and supported data formats (1).

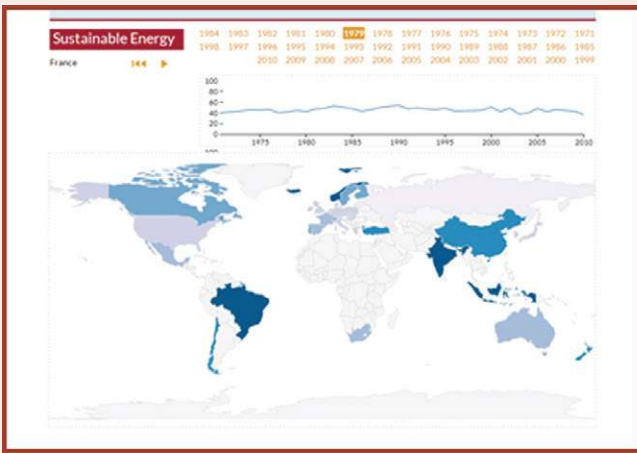
Several participants (3/21; 14.3%) indicated they rarely experiment with new technologies, with one participant stating "experimentation does not occur too much unless someone requests the change" and a second stating "we know what we know and use it and tweak it to the utmost...we only really evolve if we learn of a new software or plugin that fits with our current ecosystem." A third participant indicated that experimentation is limited because "resources [are] committed to existing projects...and [we] don't usually pick technologies on a project-by-project basis," and went on to say that too much experimentation may lead the team to "become novices in many technologies instead of proficient in a few." Therefore, constraints on resources and time may lead to path dependencies, with a program or firm leveraging the same web mapping technology long beyond its functional viability. Again, these comments fall in line with our recommended process for running a pilot study, as the discount, convergent approach enables effective use of resources and time. Additional barriers to learning new web mapping technologies listed by participants included poor or incomplete code documentation (4 participants), poor or incomplete examples

and tutorials (4), difficulty in knowing where or how to get started (2), limited awareness of available technologies (1), difficulty in working across a stack of technologies (1), and the prerequisite of learning a new programming language, such as JavaScript (1).

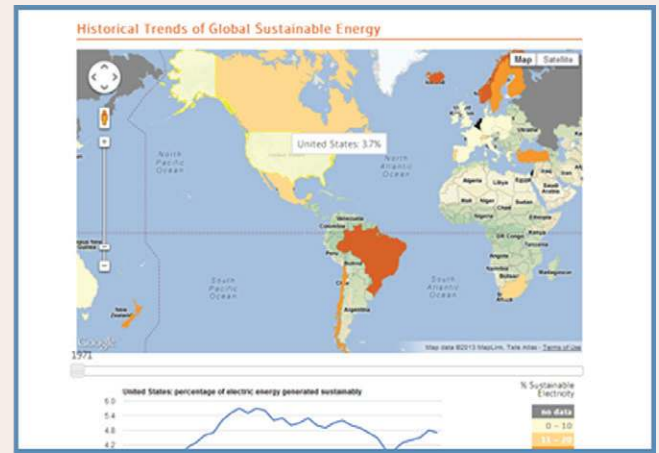
DIARY STUDY

Insights from the competitive analysis and needs assessment were triangulated to identify four candidate technologies for inclusion in the diary study. As described above, we placed an emphasis on open libraries implemented in JavaScript, given the needs of a contemporary curriculum in web cartography, yet maintained one closed API for comparison.

- *The Google Maps API* is a JavaScript API made available by Google for the creation of slippy map mashups. As reviewed above, the AJAX-based Google Maps, and its subsequent API release, was an important innovation in web mapping, giving rise to the multiscale, slippy map mashup. We selected the Google Maps API (Version 3.0) because it was the most robust in terms of supported functionality across the closed APIs reviewed in the competitive analysis study (Table 4) and was the most commonly used technology (11/21; 52.4%) by participants in the online needs assessment study. The source code of the Google Maps API is closed and therefore not available for modification or compiling on non-Google web servers, foreclosing possibilities of customization beyond methods afforded by the API itself. Further, the Google Maps API has several usage restrictions, including a maximum number of website visits before Google charges for use of its service and a requirement that web maps using the API must be freely and publicly accessible. The Google Maps API therefore served as a baseline in the diary study against which to compare the open libraries without usage restrictions.
- *OpenLayers* is an open library based in JavaScript supported by the OSGeo community. We selected OpenLayers (Version 2.12) for the diary study because it was the most robust in terms of supported functionality across the reviewed open libraries (Table 4) and was the most frequently used open library (9/21; 42.9%) by participants in the needs assessment survey, and second most frequently used technology



D3



Google Maps API

Leaflet

OpenLayers

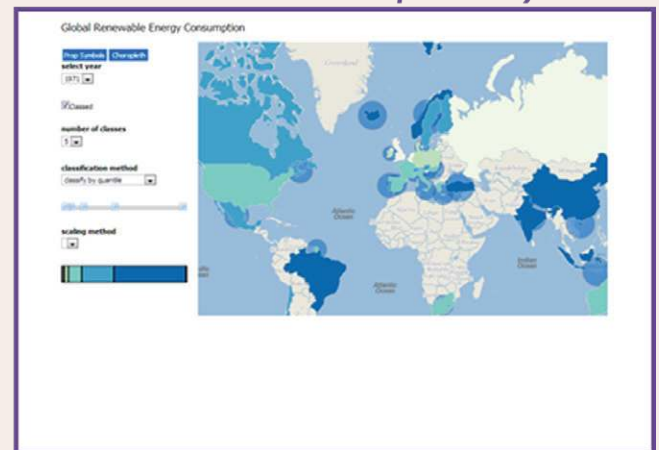
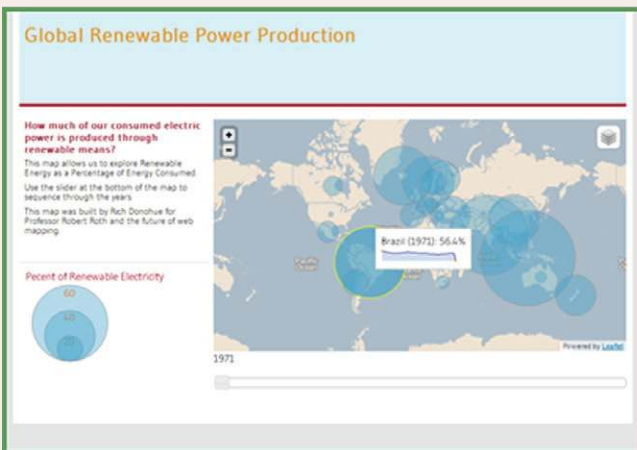


Figure 2. Example solutions for the energy web mapping scenario resulting from the diary study.

overall, behind the Google Maps API. With its initial release in 2006, OpenLayers also had a level of long-term maintenance and stability uncommon to other open web mapping libraries reviewed in the competitive analysis study, an important consideration identified through the needs assessment survey. OpenLayers is released under a 2-clause BSD License, which requires a copyright notice under code redistribution and absolves the creators of any liability; otherwise, the library is open for customization and reuse.

- *Leaflet* is an open JavaScript library pioneered and maintained by Vladimir Agafonkin (agafonkin.com/en). Leaflet supports SVG rendering within Internet Explorer 7 and 8, one advantage over most other open libraries using the SVG specification for client-side rendering. At the time of writing, Leaflet

was considered among the best web mapping libraries when designing for mobile devices because of a small file size (28MB in Version 0.4) and support of touch-based interactions. However, Leaflet was among the newest technologies included in the competitive analysis study, and was not a commonly used technology among the needs assessment participants (Table 5). We included Leaflet in the diary study due to the above advantages, and because it was the second most robust open library in terms of supported functionality, following the OpenLayers/OpenScales combination (Table 4). Leaflet is released under the same 2-clause BSD License as OpenLayers.

- *D3* (Data Driven Documents) is an open JavaScript library pioneered and maintained by Mike Bostock (bost.ocks.org/mike). We selected D3 (Version 2.0) over other open libraries that natively supported a

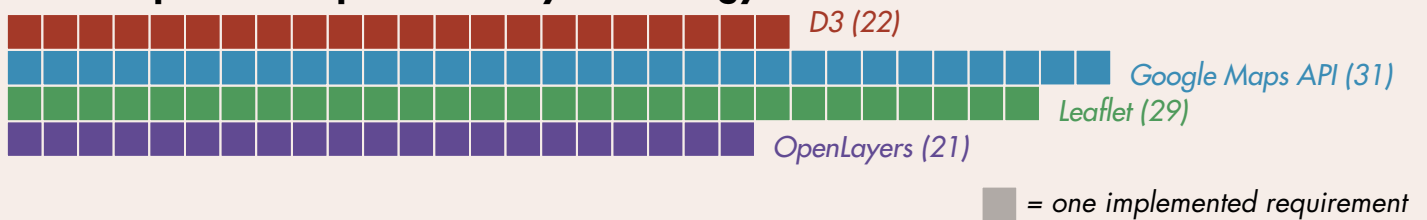
similar amount of functionality because of its unique approach to client-side rendering and interaction. Unlike tile-based technologies, D3 explicitly supports dynamic projection of linework into a wide array of map projections, using SVG to draw the projected vectors in-browser. This SVG then can be exported for manipulation in graphic design software, a unique advantage supporting other cartographic design workflows. Finally, D3 was designed to support rendering of any interactive visualization, not just maps, and therefore offers potential for multiview, coordinated geovisualization unavailable by alternative web mapping technologies. D3 is released under a 3-clause BSD License, which adds that the name of the creator (Mike Bostock) may not be used to promote or endorse any product made with D3 without written consent. D3 has no usage restrictions beyond this, and thus may be customized to the same comprehensive degree as OpenLayers and Leaflet.

Figure 2 presents example solutions to the energy web map scenario completed within the 40-hour time limit of the diary study, illustrating the relative affordances and constraints in web map design of the four candidate technologies.

Figure 3 presents an overview of the diary study results. Figure 3a illustrates the total number of diary study requirements by candidate technology. Again, each candidate technology had a pooled sample size of two, resulting in a maximum of 48 requirements per technology (2x24). On average, participants completed the most scenario requirements using the Google Maps API (31/48; 64.6%), with Leaflet a close second (29/48; 60.4%). Fewer requirements were accomplished with D3 (22/48; 45.8%) and Open Layers (21/48; 43.8%).

Figure 3b reorganizes the diary study results by individual scenario requirements. There was substantial variation in the final maps by individual requirement, with

A. Accomplished Requirements by Technology



B. Frequency by Individual Requirement

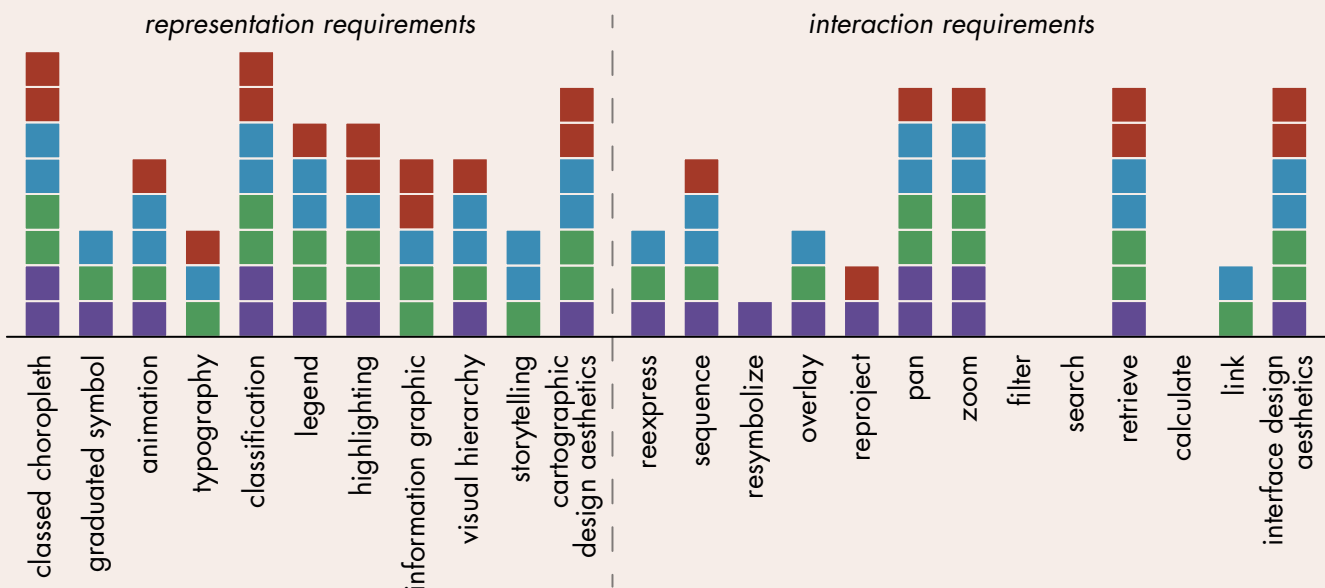


Figure 3. Overview of the diary study results: (a) total frequency of accomplished requirements by technology; (b) frequency of individual accomplished requirements from Table 3.

the choropleth map and dynamic classification the only features implemented in all eight (n=8) diary sessions. Overall, many more representation requirements (59 total, or 7.4 per diary session) were implemented compared to interaction requirements (44 total, or 5.5 per diary session), despite the energy scenario including 11 representation requirements and 13 interaction requirements. Such a finding reflects the primacy of representation over interaction in the web development workflow, and potentially decreased native support for interaction versus representation across the candidate technologies. The operators pan (7/8; 87.5%), zoom (7/8; 87.5%), and retrieve (7/8; 87.5%) were implemented in the large majority of diary sessions, but the fourth common operator, overlay (3/8; 37.5%), was implemented less frequently, likely in part due to its decreased relevance to the thematic mapping energy scenario. Looking at the absences, calculate, filter, and search were not implemented in any web map (0/8; 0.0%), suggesting increased difficulty in implementing these operators across the candidate technologies (and perhaps in all web mapping technologies broadly).

There were several differences across candidate technologies that suggest their relative affordances and limitations. Visual storytelling and live linkage between graphics were

implemented using the Google Maps API and Leaflet, but not D3 or OpenLayers. The reproject operator was implemented in D3 and OpenLayers, but not the Google Maps API or Leaflet. The resymbolize operator was implemented using D3 only. There also were several gaps in which a requirement was implemented in three of the four technologies, suggesting a limitation of the absent technology. Such gap requirements included the graduated symbol map and the reexpress operator for D3 and typography, a linked information graphic, and the overlay operator for OpenLayers.

Table 6 provides an overview of the participants' emotional experience while working with their candidate technologies. Overall, participants used 65 of the 125 (52.0%) unique terms to describe their emotional status across the eight diaries, entering their moods a total of 320 times across the diary study (8 diaries, each with 40 entries). The most commonly supplied mood was the neutral 'Okay' (31/320; 9.7% of all entries), followed closely by the negative 'Frustrated' (30/320; 9.4%). Other frequently supplied moods across the eight diaries included 'Blank' (17/320; 5.3%), 'Confused' (15/320; 4.7%), 'Content' (14/320; 4.4%), 'Excited' (14/320; 4.4%), 'Anxious' (13/320;

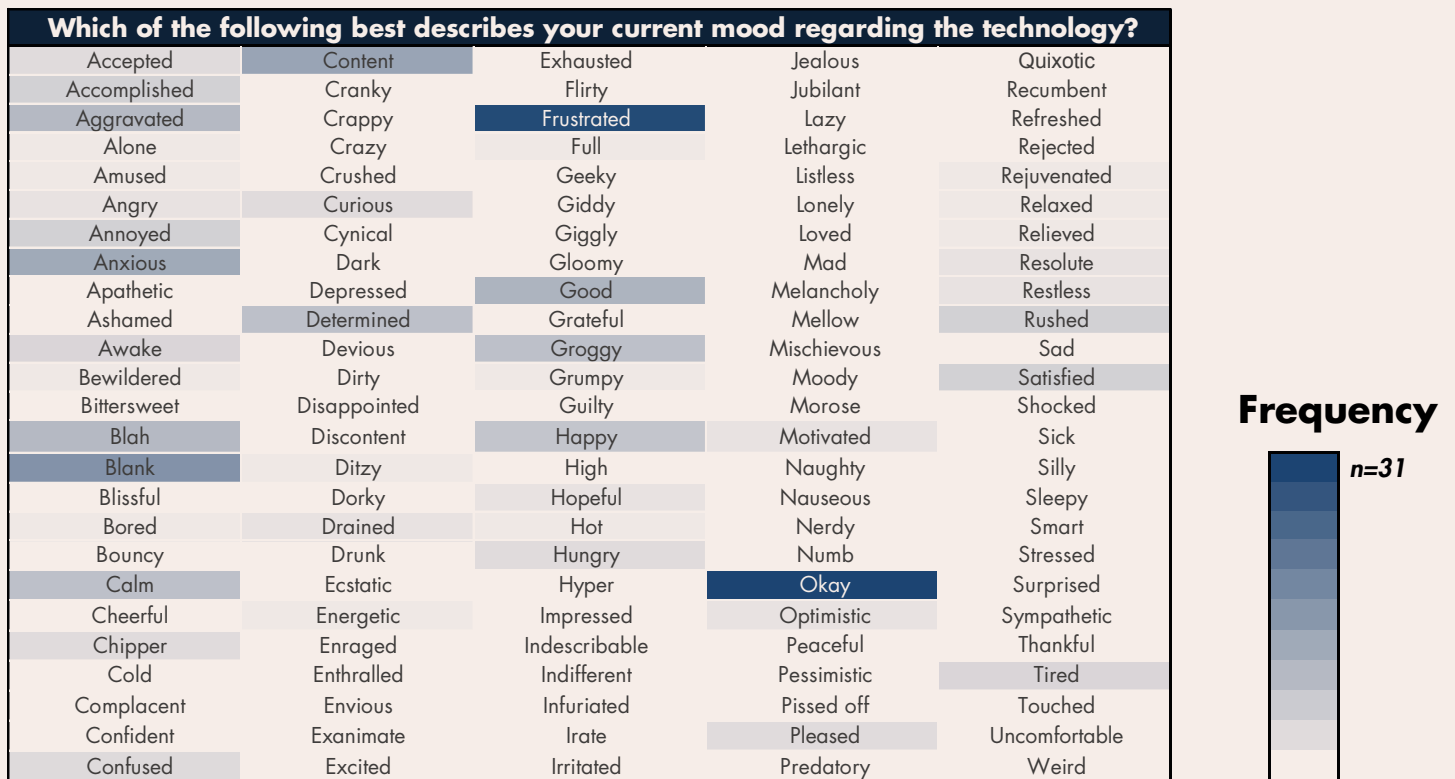


Table 6. The participants' overall emotional experiences during the diary study. Participants used 65 of the provided 125 moods across the eight diaries. A single mood was supplied for each diary entry, totaling 320 moods across the diary study (8 diaries by 40 work hours).

4.1%), 'Good' (11/320; 3.4%), 'Blah' (10/320; 3.1%), and 'Aggravated' (10/320; 3.1%).

Figure 4 breaks down the participants' emotional experiences according to the four candidate technologies. Figure 4a compares the overall valence of supplied moods across technologies. Across all eight diaries, participants were balanced nearly perfectly in their valence, supplying 125 negative moods (39.1%), 74 neutral moods (23.1%), and 121 positive moods (38.8%). In comparison to the overall average, the valence was more positive with both the Google Maps API and Leaflet. The pair of participants working with the Google Maps API supplied 26 negative moods (32.5%), 23 neutral moods (28.8%), and 31 positive moods (38.8%), while the pair of participants working with Leaflet supplied only 23 negative moods (28.8%), 19 neutral moods (23.8%), and 38 positive moods (47.5%). This means that, when working with Leaflet, participants were in a positive emotional state nearly half of the 40-hour work sessions, while in a negative emotional state just over one-quarter of the session. Therefore, participant experiences with Leaflet were slightly more positive (+8.7%) compared to participant experiences with the Google Maps API, despite the Google Maps API sessions resulting in two more implemented requirements than the Leaflet sessions.

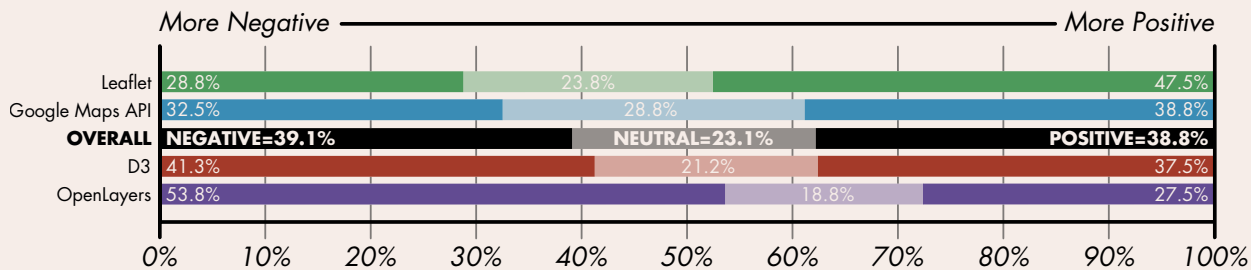
In contrast, the valence of supplied moods was more negative with D3 and OpenLayers compared to the overall average. The emotional experience with D3 was only slightly more negative than average, with the pair of participants supplying 33 negative moods (41.3%), 17 neutral moods (21.2%), and 30 positive moods (37.5%). However, the emotional experience with OpenLayers was considerably more negative than the overall average, as well as any of the other three evaluated web mapping technologies. The pair of participants working with OpenLayers supplied 43 negative emotions (53.8%), 15 neutral emotions (18.8%), and only 22 positive emotions (27.5%). Thus, the experience with OpenLayers was the opposite of Leaflet, with participants working in a negative emotional state over half of the 40-hour work session and working in a positive state just over one-quarter of the time. Inspection of the most commonly supplied moods by technology provides further evidence of this emotional disconnect between OpenLayers and the other technologies (Figure 4b), as no positive mood was supplied more than three times by participants using OpenLayers, whereas participants using

the other technologies supplied an even mixture of negative, neutral, and positive emotions.

The above summaries of implemented functionality and emotional experience are specific to the four candidate technologies included in the diary study. By analyzing the individual diary entries themselves, we were able to expose broader characteristics of web mapping technologies—and the overall web mapping process—that helped to explain the variation in implemented functionality and emotional experience. We first coded the diary entries according to the technical (Figure 1b) and practical (Figure 1c) considerations surveyed in the needs assessment. Across the 320 diary entries, 79 discussed available tutorials or examples (24.7%), 40 discussed code documentation (12.5%), 1 discussed browser compatibility issues (0.3%), and 1 discussed staffed support (0.3%); there was no discussion in the diary entries of the other technical or practical considerations listed in Figure 1.

The discussion on tutorials/examples versus code documentation revealed confusion among the participants over the best approach for getting started with their assigned technology. One participant working with Leaflet noted, "It's difficult right away to figure out what I need to read first, what is most important, and where to begin" while a second participant working with OpenLayers was "a little overwhelmed by the number of examples and possible directions." The balance in discussion between tutorials/examples and code documentation indicated opposing strategies for getting started, with six diaries starting with code examples and two starting with a multi-hour review of the code documentation. Interestingly, the participants starting with code examples ultimately second-guessed this approach. A participant working with D3 stated that "I took one of the examples and decided to manipulate it to meet my needs...This seemed like a good idea until I realized I have no idea what the code does, which led to me spending inordinate amounts of time trying to understand my own code," and went on to say "I can't decide if starting from scratch would have been more efficient or not." Similarly, a participant working with the Google Maps API stated that "the development practice of pulling from various examples, specific examples that may not work well together, [requires] thorough knowledge of the code to modify accordingly." Finally, a participant working with OpenLayers stated "one problem [is] the lack of ownership/authorship of the code I've been horsing around with the past 40-hours...[it is] perhaps better as a learner

A. Valence of Moods by Technology



B. Most Common Moods by Technology

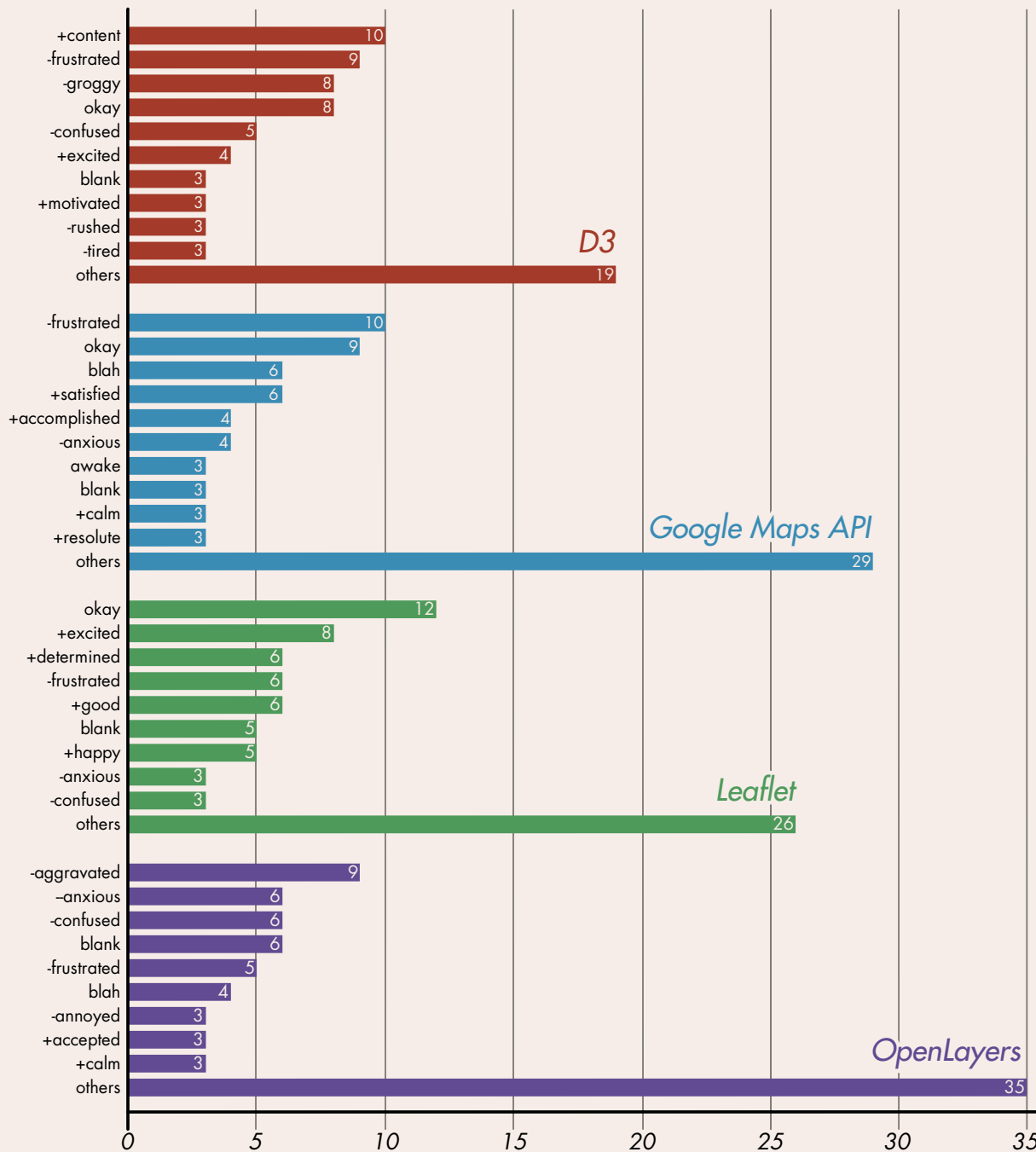


Figure 4. The participants' emotional experience with each of the four candidate technologies: (a) valence of emotional descriptions by technology, organized by negative, neutral, and positive moods (percentages out of 80 for individual technologies and 320 for the overall summary); (b) all moods provided at least three separate times for each technology (maximum frequency of 80).

and early developer to struggle through simple, less-than-ideal solutions of my own making, rather than [to be] lost debugging the more ideal examples.” Thus, while all four candidate technologies had a plethora of code examples, most of these examples were targeted toward seasoned developers and thus may not be appropriate for classroom education. Comments in the diary study instead suggested that beginning students would benefit from an initial, condensed overview of code documentation combined with simplified code examples to improve their emotional experience and encourage active learning.

Notably, there was one example of ‘staffed’ support regarding D3, with the participant writing “After my frustration earlier in the day, venting on Twitter got the attention of Mike Bostock...I spent the hour uploading data for him to look at and troubleshoot,” with the conversation resulting in the participant having a much deeper understanding of how to implement D3. While anecdotal, this interaction between a student and the creator of D3 was indicative of the positive atmosphere of the open web mapping community. It also provided an example of how online collaboration—through social media such as Twitter and forums such as Google Groups and Stack Overflow—has become part of the support process for open source web mapping technologies. As such, the developers of D3, Leaflet, and OpenLayers explicitly state that users are free to contact them with questions through these collaborative outlets.

Discussion around the best way to get started with a technology ultimately led to comments about the optimal web mapping workflow, with consensus being to first format and load the dataset, then implement the representation requirements (e.g., symbolize the dataset), and finally implement the interaction requirements (i.e., build the user interface). While the scenario requirements focused on representation and interaction design, participants spent the largest portion of their time formatting and loading the energy dataset. Across the 320 diary entries, 125 (39.1%) primarily referenced the data, 122 (38.1%) the representation requirements, and 73 (22.8%) the interaction requirements. Such a work distribution in the diary sessions signaled an importance of teaching to the *data* → *representation* → *interaction* workflow, and also revealed a blind spot in many of the existing code examples that limited their utility. A participant working with OpenLayers stated, “All examples use online data sources for the base-map, making it very difficult to figure out how to use my own data...documentation and examples for importing

data is very weak,” while one working with Leaflet observed, “There just doesn’t seem to be much information telling me how to get to the point of taking data out of [the dataset].” Therefore, we need to teach data formats and loading first—even if they are not the primary goals of the course—before teaching representation and interaction techniques.

Interestingly, one of the key advantages of the Google Maps API, leading to the high level of implemented functionality, was the documentation for loading data through Google Fusion Tables, with one participant “able to get started much more quickly on the representation, given how the [Google Maps] API makes loading and styling tiles straightforward” and the second stating that he or she “plugged the data into Fusion Tables, from which point it was pretty easy to merge with the KML and get it to render as a layer.” However, the initial decision to use Fusion Tables for the dataset led to constraints in both representation and interaction design later in both diary sessions using the Google Maps API. A first participant noted a problem with representation, stating, “Each map is limited to five layers, and each layer is limited to five styles...this means I can only have five classes (including a null value class) and won’t be able to allow users to reclassify the map,” while a second noted a problem with interaction, stating “tooltips fusion library has its own constraints... unsure if Fusion Tables are cool or not.” Ultimately, the initial decision to use Fusion Tables led to a large amount of code refactoring midway through the diary sessions to better support the representation and interaction requirements, and ultimately to some of the dissatisfaction with the proprietary Google Maps API in comparison with the fully open and more flexible Leaflet (Figure 4a).

The importance of initial data formatting and loading for subsequent representation and interaction development was not specific to the Google Maps API, however. One participant working with Leaflet reformatted the dataset numerous times throughout the diary session, stating, “working on building stats and all of a sudden I realized my data is actually completely incorrect...time to rebuild AGAIN!” This participant was unable to start on the representation and interaction requirements until the second half of the diary study due to issues with data formatting, but was able to implement the requirements quickly once the data were formatted and loaded correctly. In one extreme case, a participant working with OpenLayers struggled to implement any of the requirements due to issues

with loading the dataset until finding a set of symbolization examples that included code for data loading, which in turn led to a period of rapid development of the representation requirements. However, this participant then was unable to extend these examples to implement the interaction requirements, leading to a plateau in development over the final third of the work period. Thus, proper formatting of geospatial data remains paramount in web mapping, and should be processed not just to optimize the initial loading, but to enable the representation and interaction design as well.

Participant issues with data formatting and loading reflected broader issues with transitioning to the Open Web Platform. While the process focused on JavaScript web mapping libraries, the diaries revealed a multiplicity of competencies required to develop on the Open Web Platform. Related to the above discussion on data processing, participants struggled to manipulate data objects through the DOM. In total, 61 (19.1%) entries noted problems with manipulating GeoJSON or TopoJSON, the geospatial variants of JSON. One participant working with OpenLayers stated, “Accessing feature properties of a JSON has proven to be difficult,” while a second, working with Leaflet, was in the second half of the 40-hour period before “really understanding how to dig into the GeoJSON now, which I think is one of the most important parts of this entire exercise.” In addition to issues related to JSON and the DOM, 19 (5.9%) of the entries noted problems with SVG, 15 (4.7%) noted problems with HTML, 7 (2.1%) noted problems with CSS, and 7 (2.1%) noted problems with XML. Thus, participants spent approximately one third of their time (109 of 320 entries; 34.1%) working on development tasks unrelated to the writing of JavaScript code, an indication that students require competency in both basic website design and data formats before implementing web map behavior.

Interestingly, new versions were released during the diary study for two of the four candidate technologies: D3 and Leaflet. As discussed above, maintenance/stability was identified as the most important practical consideration from the needs assessment survey (Figure 1c). In both cases, the update was positively received by participants and aided development, rather than hindering it. One participant working with D3 stated, “A new version of D3 was released this morning which allows a thresholded

color ramp...it means I can use ColorBrewer to pick out colors, and then feed exact hex codes into D3 for my classification!” Similarly, a participant working with Leaflet stated, “With the new version of Leaflet coming out, there’s some new additions that make some of the work [simpler] so I decided to recode the map in the updated version...the language took me 3 lines to get my basemap and GeoJSON in and styled...super concise!” While only two examples, the ease participants had with integrating new releases of the D3 and Leaflet core libraries pointed to the broader trend of improved stability in open web mapping technology.

Finally, the diary entries suggested two limitations of the diary method design that could be improved in subsequent applications of the process. The most common complaint was about the rigid, hour-long structure imposed for each diary entry. Articulating this issue well, one participant working with D3 stated, “[I] had a hard time with the format of this project, mostly because I’m not very good at staying focused but also because I like to work in small chunks of time,” and went on to say “If I have only half an hour to do some work, I feel like it’s not worth it because, on average, I need about an hour and 15 minutes for each hour of this project.” While the 40-hour structure promoted consistency across the diary sessions—allowing for a more reliable comparison across technologies—it would be more practical in a professional setting to instruct participants to log a diary only after making a significant breakthrough or running up against a difficult challenge. Several participants also felt the constraint of using only a single web mapping technology was counterproductive. One participant working with Leaflet stated, “I feel like if I want to bring in something like a graph or chart to complement this I would use D3...Leaflet doesn’t support that and isn’t intended to.” A separate participant working with OpenLayers noted both of the aforementioned limitations of the diary study, stating, “Two of the constraints of the experiment that must be considered in terms of how they impact the practice of development: the [interrupted] 40-hours and mutual exclusivity of technologies.” Because most firms are likely to combine web mapping technologies based on their relative affordances and limitations—rather than relying on only one ‘winner’ technology—allowing the mixing of technologies in the diary study would have better mimicked real-world development.

CONCLUSION AND OUTLOOK

THE RESEARCH PRESENTED HERE describes a case study process for keeping pace with emerging web mapping technologies. The process supports selection of an appropriate technological solution—or combinations of solutions—for a specific web mapping context. The purpose of the process is not to identify an overall winner for all web mapping contexts, but rather to shed light on the dynamic landscape of web mapping technologies as it shifts and to select viable options from this solution space for a specific project. Following a discount, convergent approach, the process is designed to efficiently make use of project resources by iteratively narrowing available technologies across three stages: (1) a competitive analysis of available technologies, (2) a needs assessment survey of past experiences with and current opinions about these technologies, and (3) a diary study following the application of a small subset of candidate technologies.

The process successfully achieved the practical goal of identifying a candidate set of web mapping technologies for teaching web mapping, and also revealed broader insights into web map design and education generally as well as ways to cope with evolving web mapping technologies. In the following, we summarize our findings about each of these four research questions introduced earlier:

1. *What technologies currently are available for web mapping and how do they vary?* The technologies surveyed in the competitive analysis took one of four forms: frameworks, open libraries, closed APIs, and tile rendering services. The large majority leveraged JavaScript as the base programming language, and thus integrated with the Open Web Platform broadly. Multiple participants in the needs assessment survey noted the growing importance of learning and applying JavaScript-based technologies. Survey participants had used a small subset of the technologies identified through the competitive analysis study, and were unaware of a large majority of these technologies. We ultimately identified open libraries as most appropriate for higher education, as frameworks required too many additional competencies, closed APIs were limited in their openness and extensibility, and tile rendering services provided limited opportunity to teach interaction design. Four candidate technologies were selected for the diary study: three

open libraries (D3, Leaflet, and OpenLayers) and one closed API (Google Maps). While participants accomplished slightly more scenario requirements with the closed Google Maps API, they had a slightly more positive emotional experience using the open Leaflet library, in part due to the larger amount of code refactoring necessary when working around a closed API. Overall, the insight generated through the process reminded us that good web map design is more important than using novel tools—open or proprietary—and that university programs, government agencies, and cartography firms actively should combat path dependencies on one technology that lead to its use beyond its functional utility.

2. *What are the important characteristics of web maps that should inform the selection of web mapping technologies?* The competitive analysis revealed notable patterns in supported representation and interaction functionality across open web mapping technologies. The majority of technologies natively supported reference maps served as a set of raster tiles in a cylindrical projection, as well as panning, zooming, retrieval of details, and overlay of context information. Altogether, this representation and interaction functionality defines the prototypical web map. The competitive analysis also revealed gaps between contemporary web mapping practice and traditional cartographic scholarship, including thematic mapping beyond choropleth and proportional symbol representation techniques and the calculate, filter, reexpress, and reproject interaction operators. Participants in the needs assessment survey placed an emphasis on support for (user-driven) interactivity when choosing a technology, and placed less importance on (system-driven) animation and real-time updates. Despite this emphasis on interactivity, participants in the diary study accomplished nearly two more representation requirements than interaction requirements per diary session, suggesting the primacy of representation over interaction in the web mapping workflow. Only the choropleth map and dynamic classification were implemented in all eight diary sessions, with the common interaction operators pan, zoom, and retrieve implemented in

all but one diary session. The interaction operators calculate, filter, and search were not implementing in any diary session, further identifying gaps between practice and theory in web cartography. Finally, the diary study identified nuanced differences in supported functionality across the four candidate technologies, suggesting their relative affordances and limitations. Such insight can be used to derive preliminary recommendations for pairing web map requirements with potentially viable technology solutions, allowing for design to precede development (rather allowing technology to constrain design).

3. *How should web mapping be taught in higher education?* The diary study yielded multiple insights that inform the way web mapping should be taught in a university setting. First, the collection of participant moods across the diary sessions highlighted the role of emotional experience when learning a new technology. The valence of moods overall was balanced nearly perfectly across all 320 diary entries, but varied considerably by technology. When choosing a web mapping technology, therefore, it is equally important to understand both what the student can do with it and how the student feels while doing it. Second, discussion in the diary entries suggested confusion about how best to get started, with six diary sessions beginning with manipulation of code examples and two with a multi-hour review of documentation. Participants in the needs assessment identified good tutorials/examples and code documentation as practical considerations of near equal importance. However, participants in the diary study who started with code examples ultimately came to question this decision due to their complexity, with comments suggesting that an initial, condensed overview of code documentation combined with simplified code examples would be a better approach to reducing the initial learning curve. Having simple ‘beginner’ exercises early in the learning process also would provide students with early ‘victories,’ improving their emotional experience and promoting active learning. Finally, the diary studies revealed the complete set of competencies required for web mapping. This included the importance of teaching across the *data* → *representation* → *interaction* workflow, with an emphasis on data formats and loading that enable future

representation and interaction development. They also highlighted the need to teach across the Open Web Platform, with a sequence of modules on manipulating JSON in the DOM and the HTML, CSS, SVG, and XML specifications.

4. *How can we better cope with continued evolution in web mapping technologies?* The proposed three-stage process was successful in identifying viable client-side web mapping technologies for a contemporary curriculum in web cartography. Importantly, the process aligned with the pilot study approach to exploring emerging technologies already completed by several of the participants in the needs assessment survey. The process proceeded in a discount, convergent manner, which should allay concern over the resource investment of technological experimentation, and ultimately act to combat path dependencies in one technology. The case study revealed two ways to improve the diary stage of the process: requiring entries only following critical incidents (rather than hourly for 40 hours), and allowing the flexible combination of technologies using one as a base (rather than artificially restricting development to a single technology). We also put several structures in place to improve the scientific reliability of the generated insights, such as employing a pair of coders in the competitive analysis, having a fifth participant complete a diary session with all candidate technologies, and restricting group work; such structures may be removed when applying the process in a non-research context. Finally, the process suggested additional mechanisms for coping with the continued evolution in web mapping technology centered upon the active and growing open source web mapping community. From the academic world, this includes translating the development features of technologies into the *lingua franca* of cartographic design as well as the sharing of both conceptual and technical learning materials as open educational resources. From the development world, this includes crowdsourcing the organization and synthesis of web mapping technologies across the community to support young developers as they forge careers in web cartography.

As a result of the case study process, we began using Leaflet in the fall of 2012 as the base JavaScript library

for the advanced course, “Interactive Cartography and Geovisualization” at the University of Wisconsin–Madison, but included one advanced lab introducing D3 after JavaScript and Leaflet are learned, given its broad potential for thematic mapping and coordinated visualization

(see Donohue et al. 2013; Sack et al. 2014). We anticipate administering the process at approximately three-year intervals through the UW Cartography Lab—with the laboratory curriculum revised accordingly—to enable us to evolve along with emergent web mapping technologies.

ACKNOWLEDGEMENTS

WE WISH TO THANK John Czaplewski, Isaac Dorsch, and Sam Matthews for their contribution to this project.

REFERENCES

- Al-Kodmany, K. 2001. “Supporting imageability of the World Wide Web: Lynch’s five elements of the city in community planning.” *Environment and Planning B* 28: 805–832. doi: [10.1068/b2746](https://doi.org/10.1068/b2746).
- Andrienko, G. L., and N. V. Andrienko. 1999. “Interactive maps for visual data exploration.” *International Journal of Geographical Information Science* 13 (4): 355–374. doi: [10.1080/136588199241247](https://doi.org/10.1080/136588199241247).
- Bampton, M. 2011. “Addressing misconceptions, threshold concepts, and troublesome knowledge in GIScience education.” In *Teaching Geographic Information Science and Technology in Higher Education*, edited by D. J. Unwin, K. E. Foote, N. J. Tate, and D. DiBiase, 117–132. Chichester: John Wiley & Sons. doi: [10.1002/9781119950592.ch8](https://doi.org/10.1002/9781119950592.ch8).
- Boulous, M. N. K., and D. Burden. 2007. “Web GIS in practice V: 3-D Interactive and real-time mapping in Second Life.” *International Journal of Health Geographics* 6: 51. doi: [10.1186/1476-072X-6-51](https://doi.org/10.1186/1476-072X-6-51).
- Brewer, C. A., and B. P. Buttenfield. 2007. “Framing guidelines for multi-scale map design using databases at multiple resolutions.” *Cartography and Geographic Information Science* 34 (1): 3–15. doi: [10.1559/152304007780279078](https://doi.org/10.1559/152304007780279078).
- Buttenfield, B. 1999. “Usability evaluation of digital libraries.” *Science & Technology Libraries* 17 (3/4): 39–59. doi: [10.1300/J122v17n03_04](https://doi.org/10.1300/J122v17n03_04).
- Cammack, R. G. 2003. “Cartography, virtual reality, and the Internet: Integrating abstract models of the environment via the Internet.” In *Maps and the Internet*, edited by M. P. Peterson, 359–370. Amsterdam: Elsevier. doi: [10.1016/B978-008044201-3/50024-4](https://doi.org/10.1016/B978-008044201-3/50024-4).
- Cartwright, W. 2008. “Delivering geospatial information with Web 2.0.” In *International Perspectives on Maps and the Internet*, edited by M. P. Peterson, 11–30. Berlin-Heidelberg: Springer. doi: [10.1007/978-3-540-72029-4_2](https://doi.org/10.1007/978-3-540-72029-4_2).
- Clarke, K. C. 2004. “Mobile mapping and geographic information systems.” *Cartography and Geographic Information Science* 31 (3): 131–136. doi: [10.1559/1523040042246043](https://doi.org/10.1559/1523040042246043).
- Crampton, J. 2009. “Cartography: maps 2.0. Progress in Human Geography.” 33 (1): 91–100. doi: [10.1177/0309132508094074](https://doi.org/10.1177/0309132508094074).
- Donohue, R. G. 2014. *Web Cartography with Web Standards: Teaching, Learning, and Using Open Source Web Mapping Technologies*. PhD Diss., University of Wisconsin–Madison.
- Donohue, R. G., C. M. Sack, and R. E. Roth. 2013. “Time series proportional symbol maps with Leaflet and JQuery.” *Cartographic Perspectives* 76: 43–66. doi: [10.14714/CP76.1248](https://doi.org/10.14714/CP76.1248).
- Dykes, J. 1996. “Dynamic maps for spatial science: A unified approach to cartographic visualization.” In *Innovations in GIS 3*, edited by D. Parker, 177–188. London: Taylor & Francis.

- Dykes, J. A. 1997. "Exploring spatial data representation with dynamic graphics." *Computers & Geosciences* 23 (4): 345–370. doi: [10.1016/S0098-3004\(97\)00009-5](https://doi.org/10.1016/S0098-3004(97)00009-5).
- Elwood, S. 2010. "Geographic information science: Emerging research on the societal implications of the geospatial web." *Progress in Human Geography* 34 (3): 349–357. doi: [10.1177/0309132509340711](https://doi.org/10.1177/0309132509340711).
- Feldman-Barrett, L., and J. A. Russell. 1998. "Independence and bipolarity in the structure of current affect." *Journal of Personality and Social Psychology* 74 (4): 967–984. doi: [10.1037/0022-3514.74.4.967](https://doi.org/10.1037/0022-3514.74.4.967).
- Friedmannová, L., M. Konečný, and K. Staněk. 2006. "An adaptive cartographic visualization for support of the crisis management." In *Proceedings of AutoCarto*, 100–105. Vancouver, WA: Cartography and Geographic Information Society.
- Fuhrmann, S., P. Ahonen-Rainio, R. M. Edsall., S. I. Fabrikant, E. L. Koua, C. Tobon, C. Ware, and S. Wilson. 2005. "Making useful and useable geovisualization: Design and evaluation issues." In *Exploring Geovisualization*, edited by J. Dykes, A. M. MacEachren, and M.-J. Kraak, 553–566. Amsterdam: Elsevier.
- Gardner, B. S. 2011. "Responsive web design: Enriching the user experience." *Sigma Journal: Inside the Digital Ecosystem* 11 (1): 13–19.
- Garrett, J. J. 2005. "Ajax: A new approach to web applications." *Adaptive Path*. Accessed September 30 2013. <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>.
- Goldsberry, K. 2007. *Real-Time Traffic Maps for the Internet*. PhD Diss., University of California, Santa Barbara.
- Goodchild, M. F. 2007. "Citizens as sensors: The world of volunteered geography." *GeoJournal* 69 (4): 211–221. doi: [10.1007/s10708-007-9111-y](https://doi.org/10.1007/s10708-007-9111-y).
- Haklay, M., A. Singleton, and C. Parker. 2008. "Web mapping 2.0: The Neogeography of the GeoWeb." *Geography Compass* 2 (6): 2011–2039. doi: [10.1111/j.1749-8198.2008.00167.x](https://doi.org/10.1111/j.1749-8198.2008.00167.x).
- Haklay, M., and P. Weber. 2008. "OpenStreetMap: User-generated street maps." *Pervasive Computing* 7 (4): 12–18. doi: [10.1109/MPRV.2008.80](https://doi.org/10.1109/MPRV.2008.80).
- Hardisty, F., and A. C. Robinson. 2011. "The GeoViz Toolkit: Using component-oriented coordination methods for geographic visualization and analysis." *International Journal of Geographical Information Science* 25 (2): 191–210. doi: [10.1080/13658810903214203](https://doi.org/10.1080/13658810903214203).
- Harrower, M. 2008. "The Golden Age of Cartography is now." *Axis Maps*. Accessed September 30 2013. <http://www.axismaps.com/blog/2008/10/the-golden-age-of-cartography-is-now>.
- Harvey, F. 2012. "To volunteer or contribute locational information? Toward truth in labeling for crowdsourced geographic information." In *Crowdsourcing Geographic Knowledge: Volunteered Geographic Information (VGI) in Theory and Practice*, edited by D. Sui, S. Elwood, and M. Goodchild, 31–42. Amsterdam: Springer. doi: [10.1007/978-94-007-4587-2_3](https://doi.org/10.1007/978-94-007-4587-2_3).
- Herzog, A. 2003. "Developing cartographic applets for the Internet." In *Maps and the Internet*, edited by M. P. Peterson, 117–130. Amsterdam: Elsevier.
- Hu, S. 2008. "Advances of web standards and techniques for developing hypermedia maps on the Internet." In: *International Perspectives on Maps and the Internet*, edited by M. P. Peterson, 115–124. Berlin-Heidelberg: Springer. doi: [10.1007/978-3-540-72029-4_8](https://doi.org/10.1007/978-3-540-72029-4_8).
- Jenny, B., H. Jenny, and S. Räber. 2008. "Map design for the Internet." In *International Perspectives on Maps and the Internet*, edited by M. P. Peterson, 31–48. Berlin-Heidelberg: Springer.
- Jobs, S. 2010. "Thoughts on Flash." *Apple, Inc.* Accessed September 30 2013. <http://www.apple.com/hotnews/thoughts-on-flash>.
- Kraak, M.-J., and A. Brown. 2001. *Web Cartography: Developments and Perspectives*. London: Taylor & Francis.

- Leclerc, Y. G., M. Reddy, L. Iverson, and M. Eriksen. 2001. "The GeoWeb: A new paradigm for finding data on the web." In: *Proceedings of the International Cartographic Conference*. Beijing, China: International Cartographic Association.
- Leiner, B. M., V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. S. Wolff. 1997. "The past and future history of the Internet." *Communications of the ACM* 40 (2): 102–108. doi: [10.1145/253671.253741](https://doi.org/10.1145/253671.253741).
- Leiner, B. M., V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. S. Wolff. 2009. "A brief history of the Internet." *ACM SIGCOMM Computer Communication Review* 39 (5): 22–31. doi: [10.1145/1629607.1629613](https://doi.org/10.1145/1629607.1629613).
- Lienert, C., B. Jenny, O. Schnabel, and L. Hurni. 2012. "Current trends in vector-based Internet mapping: A technical review." In *Online Maps with APIs and Web Services*, edited by M. P. Peterson, 23–36. Berlin-Heidelberg: Springer-Verlag. doi: [10.1007/978-3-642-27485-5_3](https://doi.org/10.1007/978-3-642-27485-5_3).
- MacEachren, A. M., M. Wachowicz, R. Edsall, D. Haug, and R. Masters. 1999. "Constructing knowledge from multivariate spatiotemporal data: Integrating geographical visualization with knowledge discovery in database methods." *International Journal of Geographical Information Science* 13 (4): 311–334. doi: [10.1080/136588199241229](https://doi.org/10.1080/136588199241229).
- Marcotte, E. 2010. "Responsive web design." *A List Apart*. Accessed September 30 2013. <http://alistapart.com/article/responsive-web-design>.
- Marsh, S. L., and M. Haklay. 2010. "Evaluation and deployment." In *Interacting with Geospatial Technologies*, edited by M. Haklay, 199–221. West Sussex: Wiley-Blackwell.
- Masters, R., and R. Edsall. 2000. "Interaction tools to support knowledge discovery: A case study using Data Explorer and Tcl/Tk." In: *Proceedings of Visualization Development Environments*. Princeton, NJ.
- Meng, L., T. Reichenbacher, and A. Zipf. 2005. *Map-based Mobile Services*. Berlin-Heidelberg: Springer-Verlag. doi: [10.1007/b138407](https://doi.org/10.1007/b138407).
- Muehlenhaus, I. 2013. *Web Cartography: Map Design for Interactive and Mobile Devices*. Boca Raton: CRC Press.
- Nielsen, J. 1992. "The usability engineering life cycle." *Computer* 25 (3): 12–22. doi: [10.1109/2.121503](https://doi.org/10.1109/2.121503).
- . 1993. *Usability Engineering*. San Francisco: Morgan Kaufmann.
- O'Reilly, T. 2007. "What Is Web 2.0: Design patterns and business models for the next generation of software." *Communications & Strategies* 1: 17.
- Peterson, M. P. 2003. *Maps and the Internet*. Amsterdam: Elsevier.
- . 2005. "A decade of Maps and the Internet." In: *Proceedings of International Cartographic Conference*. A Coruna: International Cartographic Association.
- . 2008. "International Perspectives on Maps and the Internet: An Introduction." In *International Perspectives on Maps and the Internet*, edited by M. P. Peterson, 3–10. Berlin-Heidelberg: Springer.
- . 2011. "Travel log: Travels with iPad maps." *Cartographic Perspectives* 68: 75–82. doi: [10.14714/CP68.9](https://doi.org/10.14714/CP68.9).
- . 2014. *Mapping in the Cloud*. New York: Guilford Press.
- Plewe, B. 2007. "Web Cartography in the United States." *Cartography and Geographic Information Science* 34 (2): 133–136. doi: [10.1559/152304007781002235](https://doi.org/10.1559/152304007781002235).
- Plutchik, R. 1980. *Emotion: A Psychoevolutionary Synthesis*. New York: Harper & Row.
- Pulsifer, P., A. Hayes, J. P. Fiset, and D. Taylor. 2008. "An open source development framework in support of cartographic integration." In *International Perspectives on Maps and the Internet*, edited by M. P. Peterson, 165–185. Berlin-Heidelberg: Springer.
- Reichenbacher, T. 2003. "Adaptive methods for mobile cartography." In *Proceedings of the International Cartographic Conference*. Durban: International Cartographic Association.

- Robinson, A. C., J. Chen, E. J. Lengerich, H. G. Meyer, and A. M. MacEachren. 2005. "Combining usability techniques to design geovisualization tools for epidemiology." *Cartography and Geographic Information Science* 32 (4): 243–255. doi: [10.1559/152304005775194700](https://doi.org/10.1559/152304005775194700).
- Roth, R. E. 2012. "Cartographic interaction primitives: Framework and synthesis." *The Cartographic Journal* 49 (4): 376–395. doi: [10.1179/1743277412Y.0000000019](https://doi.org/10.1179/1743277412Y.0000000019).
- . 2013a. "An empirically-derived taxonomy of interaction primitives for Interactive Cartography and Geovisualization." *Transactions on Visualization & Computer Graphics* 19 (12): 2356–2365. doi: [10.1109/TVCG.2013.130](https://doi.org/10.1109/TVCG.2013.130).
- . 2013b. "Interactive maps: What we know and what we need to know." *Journal of Spatial Information Science* 6: 59–115.
- Roth, R. E., C. A. Brewer, and M. S. Stryker. 2011. "A typology of operators for maintaining legible map designs at multiple scales." *Cartographic Perspectives* 68: 29–64. doi: [10.14714/CP68.7](https://doi.org/10.14714/CP68.7).
- Roth, R. E., C. Quinn, and D. Hart. 2015. "The competitive analysis method for evaluating water level visualization tools." In: *Modern Trends in Cartography*, edited by A. Vondrakova, J. Brus, and V. Vozenilek, 241–256. Lecture Notes in Geoinformation and Cartography. Switzerland: Springer.
- Roth, R. E., A. Robinson, M. Stryker, A. M. MacEachren, E. J. Lengerich, and E. Koua. 2008. "Web-based geovisualization and geocollaboration: Applications to public health." In *Proceedings of the Joint Statistical Meeting, Invited Session on Web Mapping*. Denver: American Statistical Association.
- Sack, C. 2013. "Online participatory mapping: Volunteered geographic information tools for local empowerment over land use." In *Proceedings of the International Cartographic Conference*. Dresden: International Cartographic Association.
- Sack, C. M., R. G. Donohue, and R. E. Roth. 2014. "Interactive and multivariate choropleth maps with D3." *Cartographic Perspectives* 78: 57–76. doi: [10.14714/CP78.1278](https://doi.org/10.14714/CP78.1278).
- Schwertley, W. 2003. "QuickTime virtual reality maps for the web." In *Maps and the Internet*, edited by M. P. Peterson, 371–384. Amsterdam: Elsevier.
- Slocum, T. A., D. Cliburn, J. Feddema, and J. Miller. 2003. "Evaluating the usability of a tool for visualizing the uncertainty of the future global water balance." *Cartography and Geographic Information Science* 30 (4): 299–317. doi: [10.1559/152304003322606210](https://doi.org/10.1559/152304003322606210).
- Slocum, T. A., R. B. McMaster, F. C. Kessler, and H. H. Howard. 2009. *Thematic Cartography and Geographic Visualization*. Upper Saddle River: Pearson Prentice Hall.
- Tsou, M. H. 2004. "Integrating web-based GIS and image processing tools for environmental and natural resource management." *Journal of Geographical Information Systems* 6: 155–174.
- . 2005. "Recent development of Internet GIS." *GISdevelopment*. Accessed September 30 2013. http://www.gisdevelopment.net/technology/gis/techgis_002pf.htm.
- . 2011. "Revising Web Cartography in the United States: The rise of user-centered design." *Cartography and Geographic Information Science* 38 (3): 250–257. doi: [10.1559/15230406382250](https://doi.org/10.1559/15230406382250).
- Wiggins, L. L., and S. P. French. 1991. *GIS: Assessing your needs and choosing a system*. Chicago: American Planning Association.
- Wilson, M. W. 2012. "Location-based services, conspicuous mobility, and the location-aware future." *Geoforum* 43 (6): 1266–1275. doi: [10.1016/j.geoforum.2012.03.014](https://doi.org/10.1016/j.geoforum.2012.03.014).
- Woodruff, A. 2011. "Introducing 'On the Horizon.'" *Cartographic Perspectives* 68: 83–86. doi: [10.14714/CP68.10](https://doi.org/10.14714/CP68.10).