[CV-05-015] Web Mapping

Abstract

As internet use has grown, many paper maps have been scanned and published online, and new maps have increasingly been designed for viewing in a web browser or mobile app. Web maps may be static or dynamic, and dynamic maps may either be animated or interactive. Tiled web maps are interactive maps that use tiled images to allow for fast data loading and smooth interaction, while vector web maps support rendering a wide variety of map designs on the client. Web maps follow a client-server architecture, with specialized map servers sometimes used to publish data and maps as geospatial web services. Web maps are composed of data from a database or file on the server, style information rendered on either server or client, and optionally animation or interaction instructions executed on the client. Several graphic web platforms provide user-friendly web mapping solutions, while greater customization is possible through the user of commercial or open source web mapping APIs. When designing web maps, cartographers should consider the map's purpose on a continuum from exploratory and highly interactive to thematic and less interactive or static, the constraints of desktop and/or mobile web contexts, and accessibility for disabled, elderly, and poorly connected users.

Keywords: animated maps, geospatial web services, interactive design techniques, interactive maps, static maps, web mapping

Author & citation

Sack, C. (2017). Web Mapping. The Geographic Information Science & Technology Body of Knowledge (4th Quarter 2017 Edition), John P. Wilson (ed.). DOI: 10.22224/gistbok/2017.4.11

This Topic is also available in the following editions: DiBiase, D., DeMers, M., Johnson, A., Kemp, K., Luck, A. T., Plewe, B., and Wentz, E. (2006). Web mapping and visualizations. The Geographic Information Science & Technology Body of Knowledge. Washington, DC: Association of American Geographers.

Explanation

1. Definitions
2. What is a web map?
3. Web map construction
4. Design considerations

1. Definitions

web map: A map that is published and accessed via the internet, usually as part of a web page. Web maps fall into one of two categories:

static web maps are map images rendered in the browser that do not change given user input. These include map images that the user can increase or decrease in scale via zoom functionality in the browser without changing the image itself.

dynamic web maps are web maps that change appearance as they are viewed by the user. There are two sub-types of dynamic maps:

animated web maps change frequently and automatically, using time to represent one or more data attributes. Animated maps may or may not include interactivity, but often at least include controls allowing the user to pause, play, and adjust the starting point of the animation.

interactive web maps change in response to user input. Because of the ubiquity of this type of map on the internet today, many people casually think of “web maps” as synonymous with interactive web maps.

tiled web map (**slippy map**): An interactive web map that uses image or data tiles sent to the user's browser on demand, allowing for smooth, uninterrupted zooming and panning.

mashup: the combination of a set of map tiles with one or more data overlays from a different source or sources than the tileset.

application programming interface (API): A set of instructions that programs use to communicate with one another, often published in human-readable, online documentation.

commercial web mapping API: A library of code instructions that is primarily or exclusively useful for accessing a proprietary web mapping service.

open source web mapping API: A library of code instructions that is not designed with a specific service in mind and which any developer can freely alter to suit their purposes.

rich internet application (applet): A third-party application designed to compile and run binary programs in the browser. Rich internet applications are quickly disappearing in favor of open web technologies.

client-server architecture: A system allowing computers to communicate with each other across a network, in which one computer program (the server) stores and delivers content to a receiving program (the client).

server: A piece of software installed on a computer or group of computers that sends information stored on the local machine to a remote device.

client: A piece of software installed on a computer that sends requests to and receives information from a server.

Document Object Model (DOM): The collection of elements, attributes, styles, data, and script objects and procedures used to view a web page in the browser.

map server: A specialized web server used to publish geospatial web services.

geospatial web services: Data, map images, and processing services made available in real-time by a map server, usually conforming to standards set forth by the Open Geospatial Consortium (OGC).

vector tiles. Small chunks of vector data each covering the geographic area of a single map tile delivered with a separate set of style instructions that is used to render each tile in

the browser.

stop-frame animation: Animation created by displaying many individual images (or frames) in rapid succession.

tweening: A smooth transition between map states executed programmatically (i.e., through code).

interaction: The conversation between the user and the map mediated by the client.

user interface (UI): The set of elements on the page that the user can see and/or manipulate.

AJAX (Asynchronous Javascript And Xml): A set of procedures that allows a client to send requests to a server and receive data in response to user input without having to reload the whole web page.

visual hierarchy: The relative visual dominance of objects on a map, with more important objects being more visually dominant.

strong visual hierarchy: Objects on the map contrast distinctly and have a very recognizable order of importance.

weak/flat visual hierarchy: Objects on the map are visually comparable and difficult to order in terms of relative importance.

overdesign: The inclusion of too much information, too many visual elements, or too much interactivity given the map's purpose, resulting in clutter and reduced usability.

map layout: The visual arrangement of user interface (UI) elements on and around a web map.

map composition: The way in which web maps are visually structured to communicate certain information to the user.

responsive web design: A set of techniques for modifying the layout of web page content to conform to the user's display size and context.

accessibility: The principle of ensuring that people who have disabilities, are older, or live in rural areas or developing countries can use websites and web applications.

2. What is a web map?

2.1 Static vs. dynamic web maps

The first web maps were printed maps that were simply scanned and posted to the internet as early as 1993, when the introduction of the Mosaic web browser allowed early internet users to view images for the first time (Peterson, 2008). Scanned map images are still a common type of web map; often these are delivered as Adobe PDF documents or placed within a viewing frame that allows the user to zoom in to view parts of the map in greater

detail (Peterson, 2014). In recent years, though, increasing attention has been paid to the design constraints of computer displays and web browsers, and there are now a host of design guidelines for digital maps that depart from those for print maps (e.g., Jenny et al., 2008; Muehlenhaus, 2014; Tolochko, 2016). Whether printed first and then scanned, or created on a computer for digital consumption, a web map that does not change its appearance is classified as a **static web map** (Figure 1).

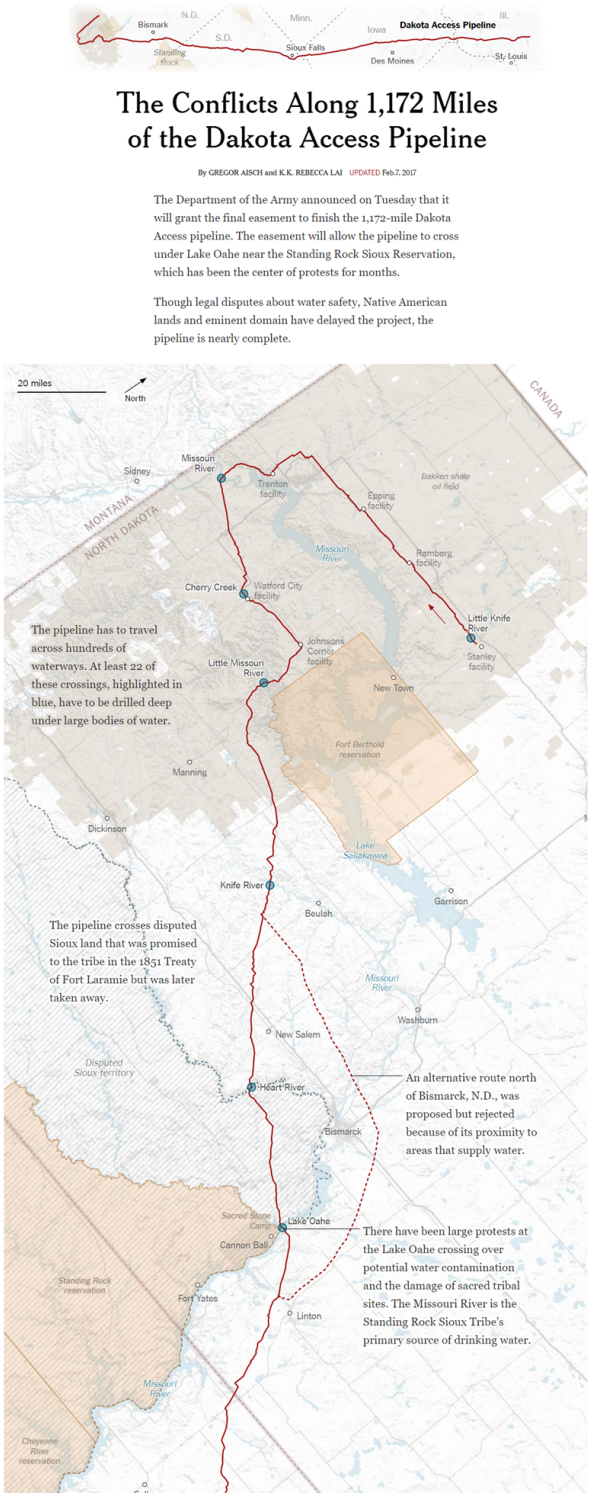


Figure 1: A portion of a static map formatted for the web. The full map is very long (1097x8032 pixels) to facilitate vertical scrolling on a web page. Map by Gregor Aisch and K.K. Rebecca Lai, from The New York Times, last updated February 7, 2017. Reprinted with

permission.

The development of dynamic web maps has taken place parallel to the proliferation of static web maps. **Dynamic web maps** change their appearance, either frequently and automatically to show change in a phenomenon (**animated web maps**; Figure 2) or arbitrarily in response to user input (**interactive web maps**) (Roth, 2013). The Map Viewer website, developed in 1994, allowed the user to request a unique map centered on a specific point (Peterson, 2014). In 1996, MapQuest introduced a map that users could zoom in and out of and pan side to side, with each request loading a new web page with the new map view. About a decade later, MapQuest was eclipsed by Google, which introduced the tiled web map in 2005.

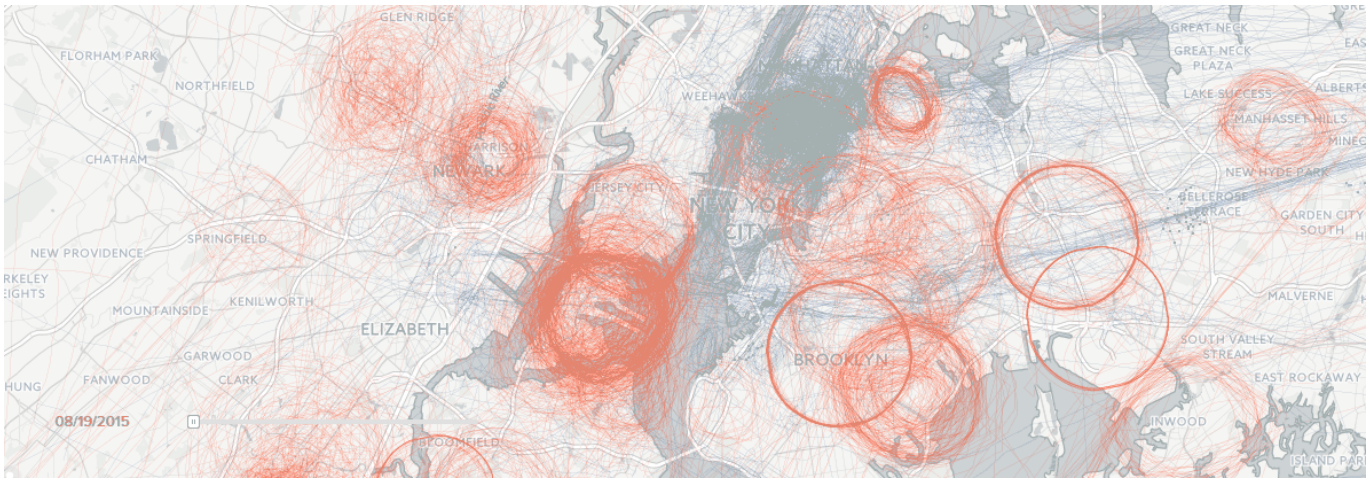


Figure 2: An animated web map of flight tracks taken by government surveillance planes. The original map also provides some interactions. Map by Peter Aldhous, from BuzzFeed News, April 6, 2016. Reprinted with permission.

2.2 Tiled web maps and mashups

Tiled web maps, also termed **slippy maps**, allow for uninterrupted panning and zooming by cutting up the data or map image into 256-by-256-pixel tiles and sending tiles to the user's browser as the browser requests them (Peterson, 2012; Figure 3). These live requests are performed via **AJAX (Asynchronous JavaScript and XML)**, a set of computational routines that allows the client to get data from the server without reloading the web page, making interacting with the map smooth and intuitive. Tiled maps remain among the most popular kinds of web maps.



Figure 3. A map tile from OpenStreetMap, a tiled reference map composed of data from hundreds of thousands of contributors.

Along with tiled web maps, Google invented and standardized the Web Mercator (or Spherical Mercator) projection, which is still used by almost all tilesets. This projection is useful because of its simple computation, rectangularity (allowing the map to be continuous east-to-west), and conformality, which reduces shape distortion at large scales (Muehlenhaus, 2014). However, it is also less accurate than ellipsoidal projections, and it greatly distorts high-latitude areas at the global scale, leading to concerns over its impact on how people imagine the planet's shape and interpret areal data (Battersby et al., 2014).

The advent of the tiled web map was shortly followed by the first mashup. By reverse-engineering the Google Maps website, 'map hackers' figured out how to put data from elsewhere on top of Google's image tiles (Crampton, 2010). Google responded by openly publishing its **application programming interface (API)**, a set of instructions programs use to communicate with one another (Muehlenhaus, 2014). While OpenLayers was soon developed as a non-commercial alternative, Google Maps API remained the most used mashup generator until Google began limiting its unpaid usage in 2011. At that point, open source alternatives rapidly gained in popularity.

2.3 Rich internet applications and vector web maps

Prior to the advent of tiled web maps, cartographers who sought smooth animation and interaction began making maps using **rich internet applications**, or third-party binary applications embedded in the browser as an extension (often called an **applet**). The most common of these for mapping was Flash, developed by Macromedia in 1997 and later purchased by Adobe (Peterson, 2008). By downloading and compiling as a single executable package, programs built with Flash could overcome the bandwidth limitations of 56K modems to provide a smooth user experience with high-quality rendering of custom graphics (Roth et al., 2014). Flash maps were common until the early 2010s, when mobile operating system makers announced they would no longer support third-party applications in mobile browsers, and the web development community moved toward sole reliance on open web technologies implemented natively by browsers (Adobe Corporate Communications, 2015).

As applets went away, cartographers both turned to tiled web maps and began seeking

alternatives that did not rely on Web Mercator-based tilesets. The SVG (Scalable Vector Graphics) standard provided a way to draw and style maps that could still be dynamic and interactive directly in the browser. Although its adoption by some browsers took several years and its rendering speed was initially poor, SVG is now supported by all major browsers, and improved bandwidths and graphics processors have made its user experience competitive with that of the tiled web map (Peterson, 2008). The newer HTML Canvas element supports faster rendering of vector linework, but treats the graphic as a raster image once rendered (Lienert et al., 2012). Interaction and 3D scenes can be added to Canvas using WebGL, a JavaScript API that uses the computer's graphics processor to further accelerate rendering (Caballero, 2011).

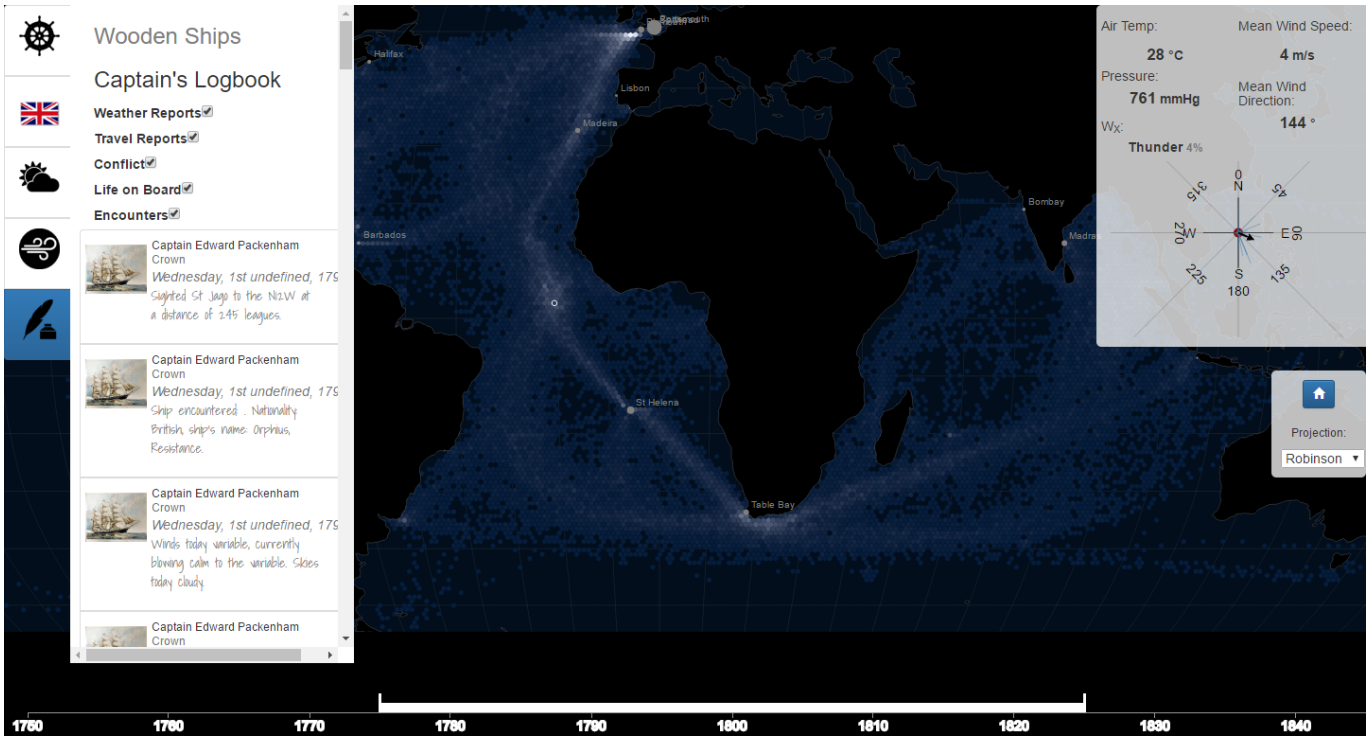


Figure 4. An interactive vector web map allowing exploration of historic ship captains' logs. Map by Scott Farley, Starr Moss, and Meghan Kelly. Reprinted with permission.

3. Web Map Construction

3.1 Client-Server Architecture

Like all web content, web maps follow a client-server architecture model (Peterson, 2014). The server is a piece of software installed on a computer or group of computers that sends information stored on the local machine to a remote device, or **client**. The server also receives requests and data from the client, performs computations, and writes data to storage on its machine. Each server has its own static IP (Internet Protocol) address, a string of numbers by which the client locates it. The client—a browser or app stored on the user's device—makes requests for the information stored on the server's machine by making a call to the server's IP address (usually routed through a domain name server (DNS) address, e.g., www-dot-example-dot-com). The server returns to the client the HTML documents, CSS stylesheets, JavaScript program instructions, and other necessary data and images stored in the website's directory. The client then compiles these data and

instructions into the web page viewed by the user.

The word “server” has become shorthand for a high-power computer, often one that sits in a data center with hundreds of other similar machines (Peterson, 2014). More precisely, a server is a piece of software that can be located on any computer, even a home PC or laptop, that makes the contents of a certain directory on that computer visible to a network. The most popular servers currently are Apache, NGINX, Google Web Server, and Microsoft’s IIS (Netcraft, 2017).

On the client side, the browser renders pages and executes programs sent to it using the **Document Object Model (DOM)**, which organizes all the elements, attributes, styles, data, and script objects and procedures needed to view the web page (Sack and Roth, 2017). In addition to the current set of fully-featured web browsers (Firefox, Chrome, Internet Explorer/Edge, Safari, and Opera), web maps are increasingly being viewed on mobile devices using apps developed to tap into a particular internet service. Google Maps and Apple Maps are the best-known map apps, but there are many others, and cartographers with basic coding skills can build their own for distribution through OS makers’ app stores (Muehlenhaus, 2014).

The basic ingredients of a web map are data, **representation** of the data using style information, and, if a dynamic map, **animation** or **interaction** (Donohue, 2014). Each will be covered in turn below.

3.2 Data

Data is hosted on the server’s machine. Vector data used to dynamically render web maps is often stored in a database. The most common database management systems used for the web (MySQL, PostgreSQL, Microsoft SQL, and Oracle) all have geospatial extensions that allow for the storage of geographic vector data, as well as a varying ability to do server-side geoprocessing with the data (Lienert et al., 2012). Vector data may also be stored in a file on the server, or converted from a database to a file format for transmission. It is important to note that vector data used in web maps typically does not include projection information, as most web mapping APIs project the data in the client application, and pre-projected data can generate incomprehensible results.

Raster map images (including raster tiles) are almost always stored as individual files in a simple directory structure. Table 1 lists common vector and raster file formats for the web (see also Raster Formats and Sources and Vector Formats and Sources).

Table 1. Common web map data file formats

Format	File Extension	Description
GML	.gml	Geographic Markup Language; an XML-based Open Geospatial Consortium standard for storing spatial vector data and transmitting web feature services (see section 2.3) (Stefanakis and Patroumpas, 2008)
KML	.kml	Keyhole Markup Language; an XML-based format used by Google Earth and Google Maps API. KMZ files are zipped KMLs. (Stefanakis and Patroumpas, 2008)
SVG	.svg	Scalable Vector Graphics; an XML-based W3C open web standard vector graphics format that seamlessly integrates with HTML. SVG can be drawn from georeferenced data, but cannot store geospatial coordinates. (Lienert et al., 2012)
GeoJSON	.geojson, .json	Geospatial extension to JavaScript-based JSON specification. Easily translated into JavaScript in the browser. (Bostock and Davies, 2013)
TopoJSON	.topojson, .json	JSON-based format that encodes geospatial topology. Greatly reduces file size over GeoJSON and enables line simplification. (Bostock and Davies, 2013)

Esri Shapefile	several (.shp, .shx. .dbf, .prj, etc.)	The most common data format for desktop GIS, but does not integrate well with open web standards. Upload and download of zipped shapefiles are supported by many web mapping applications. (Esri, 2017)
JPEG	.jpg, .jpeg	Joint Photographic Experts Group; raster image format with large color range and robust compression. Best suited for photographs and remotely sensed imagery. (Peterson, 2014)
PNG	.png	Portable Network Graphics; raster image format with lossless compression and good edge rendering. Best suited for linework. (Peterson, 2014)
GIF	.gif	Graphics Interchange Format; an early, compact web graphics format limited to 206 colors that supports brief, non-interactive animations.
PDF, GeoPDF	.pdf	Portable Document Format; print-optimized raster and/or vector graphics rendered by browser plug-ins as well as widely available desktop applications and mobile apps. Allows image layers to be turned on and off, creating limited interactivity. GeoPDF is an extension of PDF that allows georeferencing information to be included. (Muehlenhaus, 2014).

3.3 Representation

The representation aspect of a web map is how the mapped information is seen and interpreted by the user (Donohue, 2014). It involves the application of style information to the data to render a visible map image. This rendering may take place on either the server or client side.

In the case of raster map images, the style information is included in the image provided by the server. The map images may either be preexisting files or rendered by a specialized **map server** using a Styled Layer Descriptor (SLD) stylesheet (Lienert et al., 2012). Map servers can be set up to render images on-the-fly as requested by the client, or render each image once and store it in a server-side cache. Raster tilesets for tiled maps are typically created using the latter approach. MapServer, Geoserver, and QGIS Server are popular open source map servers, while ArcGIS Server is the most widely used commercial map server.

Map servers publish data and images as **geospatial web services**. The Open Geospatial Consortium (OGC) maintains a set of standards for geospatial web services, listed in Table 2 with the name of each service linked to its documentation on the OGC website. These standards allow clients to query the map server's data and operations to produce a custom result, and they immediately update with changes to the underlying data, allowing for real-time cartographic representation of changing phenomena (Cerba & Cepicky, 2012).

Table 2. OGC geospatial web services.

OCG Standard	Description
Web Map Service	Publishes static, whole raster map images rendered by the map server using an SLD stylesheet.
Web Map Tile Service	Publishes raster image tilesets rendered by the map server for use in tiled web maps.
Web Feature Service	Publishes vector data in GML format.
Web Coverage Service	Publishes coverages, or geospatial information representing space/time varying phenomena.
Web Processing Service	Specifies rules for client requests and inputs for geoprocessing on a server and for server responses.

For both static and dynamic vector maps, the style information is transmitted separately from the data and rendered by the client. In the case of SVG graphics, the style information

may be included within the SVG file, or applied by separate instructions written in CSS or JavaScript, or a combination (Muehlenhaus, 2014). An SVG may also be built from scratch on the client using separate data sources and instructions in the JavaScript code, as are all maps that are drawn using Canvas (Lienert et al., 2012). Tiled maps may use **vector tiles**, which consist of many small chunks of data covering the geographic area of a single tile and one separate set of style instructions that is used to render each tile in the browser (Turner, 2015).

3.4 Animation and Interaction

The final component of a dynamic web map is the map's animation or interactions, which are executed on the client side. There are two basic types of animated web maps (Muehlenhaus, 2014). **Stop-Frame Animation** is provided through a video or GIF image file, which is composed of many individual images (or frames) that appear in rapid succession. Time-series data typically uses this kind of animation. **Tweening**, or a smooth transition between two map states, is created by program instructions in SVG or JavaScript code and is useful for showing continuous spatial-temporal data (see Mapping Time).

Interaction is the conversation between the user and the map mediated by the client (Roth, 2012). With the demise of rich internet applications referenced in Section 1, almost all web map interactions are implemented using JavaScript, the programming language of the web (Peterson, 2014). User input is collected through the **user interface (UI)**, the set of elements on the page that the user can see and/or manipulate, and is interpreted by the script to generate a change in the map state (see User Interface and User Experience Design).

Tiled web maps create smooth map interaction through the use of AJAX (see Section 1.2). While AJAX revolutionized how web pages are programmed, its major drawback is the requirement for continuous network connectivity, making most tiled web maps unusable where internet service is spotty or unavailable (Tsou, 2011).

3.5 Web Mapping Platforms and APIs

Increasingly user-friendly, cloud-based web applications have brought web mapping within the reach of technical novices (Muehlenhaus, 2014). These services allow users to add data to preexisting base maps, style the data, add interactive components, and host the final product for direct sharing or embedding in a web page. Several popular web mapping services are listed in Table 3.

Table 3. Graphic, cloud-based web mapping applications

Web Application	Description
Mapbox Studio	Styles and hosts custom vector tilesets using OpenStreetMap and user-imported datasets.
CARTO Builder	Hosts geospatial data in PostgreSQL/PostGIS database, maps data on a wide range of tilesets, styles data layers, creates visualizations, allows data queries, and performs spatial analysis.
Google My Maps	Allows drawing or adding data layers on Google tilesets and provides limited data styling.
Google Fusion Tables	Georeferences and maps tabular data on Google tilesets, allows limited data styling, and creates visualizations.
Esri's ArcGIS Online Map Viewer	Allows drawing or adding and styling data layers on ESRI and open-licensed tilesets. Allows spatial analysis and access to curated datasets with enterprise subscription.

More advanced cartographers who wish to customize the representation and user interface

of a web map can do so on the client side using a JavaScript API (Application Programming Interface; see Section 2) (Peterson, 2014). **Commercial APIs** tap into proprietary map services, while **open source APIs** have no specific parent service (Roth et al., 2014). Both consist of code libraries that provide methods to simplify the use of the DOM to create web maps. Popular APIs are listed in Table 4.

Table 4. Web Mapping APIs

API	Description
Google Maps API	Provides access to Google Maps tilesets and services.
Bing Maps Control	Provides access to Bing Maps tilesets and services.
HERE Maps API	Provides access to HERE Maps tilesets and services.
ArcGIS API	Allows mashups, spatial analysis, and visualization of Esri tilesets.
CARTO.js	Provides access to data and visualization stored on CARTO.
Mapbox GL JS	Renders Mapbox vector tiles with UI controls and interactions.
Mapzen Tangram	Renders Mapzen vector tiles, using Leaflet for controls and interactions.
OpenLayers	Robust open source library for tiled map mashups on any tilesets.
Leaflet	Lightweight open source library for tiled map mashups on any tilesets.
D3	Open source library for dynamic vector mapping and data visualization.

4. Design Considerations

The purpose of this section is outline the unique design constraints of web maps without rehashing the cartographic principles presented in other CV topics, which remain relevant to web maps.

4.1 Map Purpose

Following the Swoopy diagram (DiBiase, 1990; see Cartography and Technology), web maps can be placed along a continuum from highly exploratory to highly thematic (Figure 5; see User Interface and User Experience (UI/UX) Design).

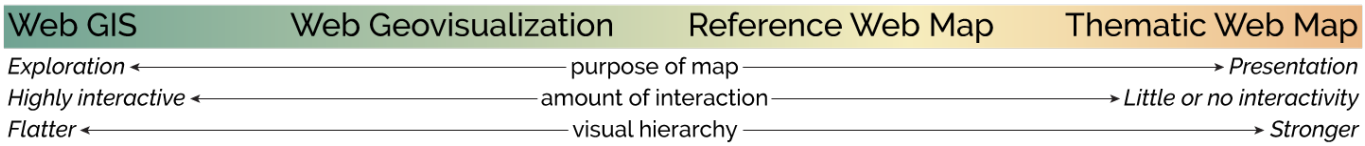


Figure 5. The web map continuum.

Visual hierarchy is the relative visual dominance of objects on the map, such that more important objects are more visually dominant (Dent, 1990). Because they display many features that may be of interest, more exploratory web maps should have a flatter visual hierarchy to allow users greater freedom to draw their own conclusions. They should also allow for more interaction to enable the user to create more unique map views (MacEachren, 1994). By contrast, thematic maps should have a strong visual hierarchy and few if any interactions to keep the user's focus on the key information being presented (Figure 6).

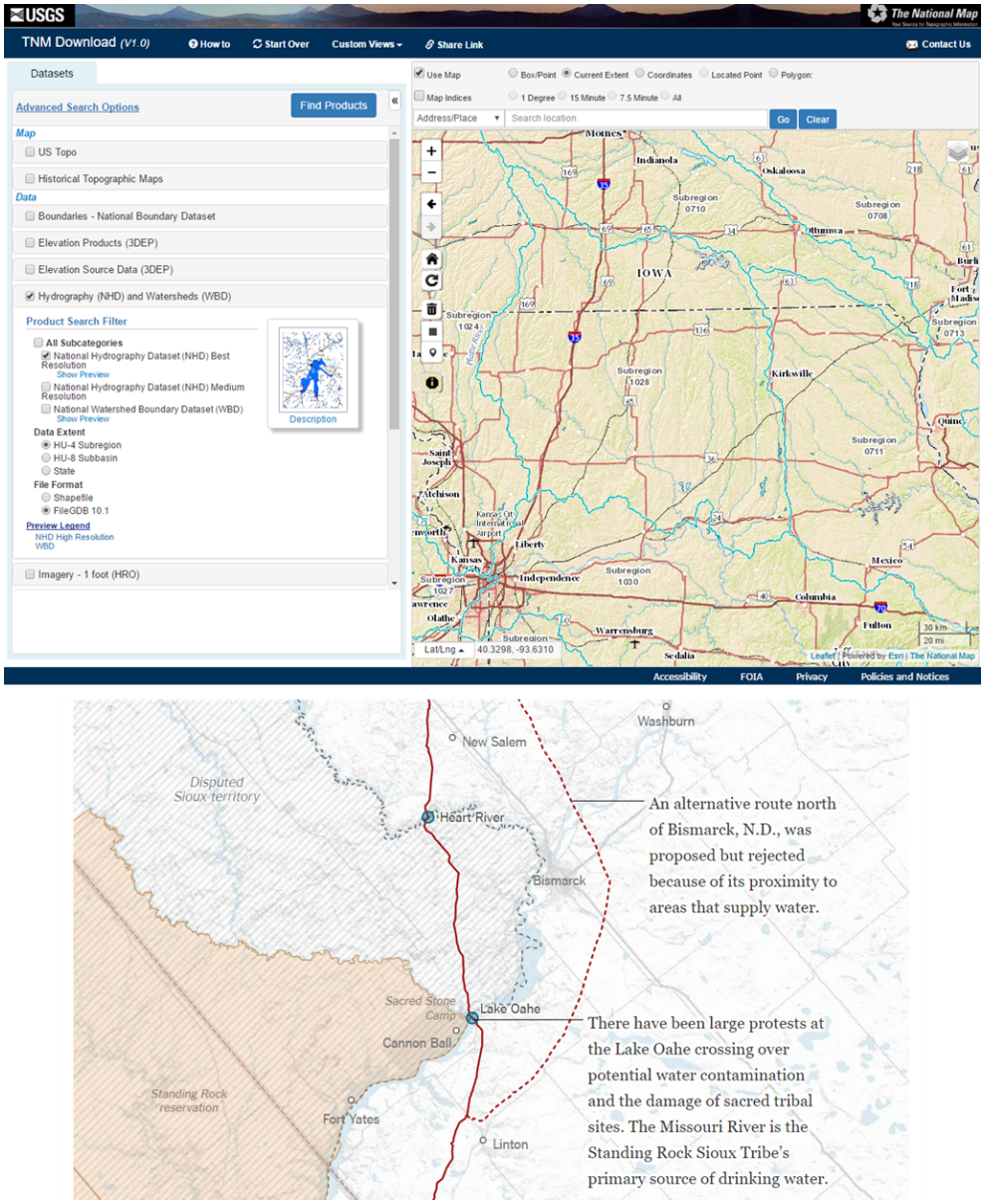


Figure 6. The National Map download tool (top) provides a large amount of data for exploration, includes many possible interactions, and symbolizes map layers to create a flattened visual hierarchy. In contrast, the thematic New York Times map shown in Figure 1 (bottom) contains no interaction and highlights important information with bold symbolization, labels, and callouts atop a washed-out base map, creating a strong visual hierarchy.

Web maps should avoid **overdesign**. To maximize its clarity and usability, a web map should only include the information, map elements, and interactions necessary to accomplish the map's purpose (Tolochko, 2016).

4.2 Map Elements

Web map elements should have a consistent look across the base map, data, and user interface layers of the map, and fit with the overall page or app design (Tolochko, 2016). The sizes and resolutions of device screens vary widely and affect how web maps appear to the user (Muehlenhaus, 2014). A web map should adjust its layout dynamically to fit the available display space of any device it is intended for use on, following the strategies of

responsive web design (Jenny et al., 2008; Marcotte, 2011). Additionally, different browsers and operating systems have slight differences in how they display web pages, so web maps should be tested on several different platforms before release. Table 5 presents a summary of map-based UI elements and common implementation schemes.

Table 5. Web map elements based on descriptions in Muehlenhaus (2014) and Tolochko (2016).

Map Element	Web map implementation(s)
Map title	Page header or temporary splash screen.
Map area	May be dynamically adjusted through panning/zooming, changes to screen size, and device orientation.
Base map	Provides geographic context. Use muted colors in thematic web maps to create strong visual hierarchy (Figure 5). Including a satellite imagery layer is appropriate for exploratory and reference web maps, but distracting in thematic maps.
Data layers	Exploratory maps include many data layers with a flatter visual hierarchy; thematic web maps include a few styled to stand out against base map (Figure 6). Point symbols should be highly iconic so they can be distinguished when small. Thematic layers should be easy to interpret and avoid possible hidden meanings (see Common Thematic Map Types).
Map scale	May be displayed through a scale bar that changes dynamically if the map can be zoomed or panned. Do not display a ratio scale, as absolute scale changes with display size and projection centering.
Supplemental information	Embedded text, links, images, graphics, and videos that may be presented as information windows, tooltips atop the map, and/or elsewhere on the web page. Provides a point of entry for the map user.
Labels	Quantity and size of labels can change at different zoom levels. Labels should be large enough for pixel densities of intended users' displays and use sans-serif, common web fonts.
Inset/Locator map	If included in an interactive map, changes dynamically to match the current view.
Map metadata	At minimum, include the cartographer(s)' name(s), data sources, and license information. It can be shown in small font directly on or below the map or made easily available in a splash screen or on a linked webpage.
North arrow	Do not include in a static web map oriented with north up. If a web map can rotate, it should include an arrow that also rotates dynamically.
Neat lines / frame lines	Not included if map takes up the full size of the screen or is well-integrated with the rest of the page content.
Supplemental graphs	Graphs or charts may be linked to a thematic map through simultaneous highlighting or brushing.
Legend	For dynamic maps, may update dynamically or allow interactions that affect the map.
Menus	May be included to provide the map user with additional options and interactions.
Help	May be provided as a link or tutorial pop-ups to orient the user to the map.
Animation controls	May be included to give the user control over playback. They should integrate a temporal legend. Keep animations short and simple (see Mapping Time).
Interaction controls	Should be included to facilitate interactions or when mouse/touch/ keyboard interactions are not immediately obvious. They should provide feedback to the user when activated (see User Experience and User Experience (UIUX) Design). On highly interactive maps, include a mechanism to reset the default view and parameters. Avoid interactions that interfere with web page scrolling.

4.3 Accessibility

Accessibility for the web means that people who have disabilities, are older, or live in rural areas or developing countries can use websites and web applications (W3C, 2016). Web maps should follow the Web Content Accessibility Guidelines published by the World Wide Web Consortium (W3C) to the greatest extent possible. For example, web maps can provide alt tags for images to be read by screen readers, use proper heading levels for titles, and allow keyboard input options for interactions. Styling should take colorblind users into account. Interactive web maps that rely on newer technologies or those with large

datasets should provide alternative versions that can be viewed with slower connection speeds and older, non-conforming browsers (Tolochko, 2016).

References

Adobe Corporate Communications. (2015, November 30). Flash, HTML5 and open web standards [Blog].

Battersby, S. E., Finn, M. P. Usery, E. L., & Yamamoto, K. H. (2014). Implications of Web Mercator and its use in online mapping. Cartographica, 49(2), 85-101.

Bostock, M., & Davies, J. (2013). Code as Cartography. The Cartographic Journal, 50(2), 129-135.

Caballero, L. (2011, October 13). An Introduction to WebGL - Part 1. Dev-Opera.

Cerba, O., & Cepicky, J. (2012). Web Services for Thematic Maps. In M. P. Peterson (Ed.), Online Maps with APIs and WebServices (pp. 141-155). Berlin, Heidelberg: Springer.

Crampton, J. (2010). Mapping: A Critical Introduction to Cartography and GIS. Chichester, UK: Wiley-Blackwell.

Dent, B. D. (1990). Cartography: Thematic Map Design. 2nd Edition. Dubuque, IA: W.C. Brown.

DiBiase, D. (1990). Visualization in the Earth Sciences. Earth and Mineral Sciences, Bulletin of the College of Earth and Mineral Sciences, Pennsylvania State University, 59, 13-18.

Donohue, R. G. (2014). Web Cartography with Web Standards: Teaching, Learning, and Using Open Source Web Mapping Technologies (Dissertation). University of Wisconsin-Madison, Madison, WI.

ESRI. (n.d). Shapefiles. ArcGIS Online Help. Retrieved March 13, 2017

Jenny, B., Jenny, H., & Räber, S. (2008). Map design for the Internet. In M. P. Peterson (Ed.), International Perspectives on Maps and the Internet (pp. 31-48). Berlin-Heidelberg: Springer.

Lienert, C., Jenny, B., Schnabel, O., and Hurni, L. (2012). Current Trends in Vector-Based Internet Mapping: A Technical Review. In: Peterson, M. (eds) Online Maps with APIs and WebServices. Lecture Notes in Geoinformation and Cartography(). Springer, Berlin, Heidelberg.

MacEachren, A. M. (1994). Visualization in Modern Cartography: Setting the Agenda. In: A. M. MacEachren & D. R. F. Taylor (Eds.), Visualization in Modern Cartography (pp. 1-12). Oxford, England: Pergamon.

Muehlenhaus, I. (2013). Web Cartography: Map Design for Interactive and Mobile Devices.

1st Edition. Boca Raton, FL: CRC Press.

Netcraft. (2017). February 2017 Web Server Survey. Retrieved June 2, 2017.

Peterson, M. P. (2008). International Perspectives on Maps and the Internet: An Introduction. In M. P. Peterson (Ed.), International Perspectives on Maps and the Internet (pp. 3-10). Berlin, Heidelberg: Springer.

Peterson, M. P. (2012). The Tile-Based Mapping Transition in Cartography. In L. Zentai & J. Reyes Nunez (Eds.), Maps for the Future: Children, Education and Internet (pp. 151-163). Berlin, Heidelberg: Springer.

Peterson, M. P. (2014). Mapping in the Cloud. New York and London: Guilford Press.

Peterson, M. P. (ed.). (2003). Maps and the internet (1st ed). Amsterdam and London: Elsevier.

Roth, R. E. (2012). Cartographic Interaction Primitives: Framework and Synthesis. The Cartographic Journal, 49(4), 376-395.

Roth, R. E. (2013). Interactive Maps: What we know and what we need to know. Journal of Spatial Information Science, 6: 59-115.

Roth, R. E., Donohue, R. G., Sack, C. M., Wallace, T. R., & Buckingham, T. M. A. (2014). A Process for Keeping Pace with Evolving Web Mapping Technologies. Cartographic Perspectives, 2014(78), 25-52.

Sack, C. M. (2013). Mapmaking for Change: Online Participatory Mapping Tools for Revealing Landscape Values in the Bad River Watershed (Master's Thesis). University of Wisconsin-Madison, Madison, WI.

Sack, C. M., & Roth, R. E. (2017). Design and evaluation of an Open Web Platform cartography lab curriculum. Journal of Geography in Higher Education, 41(1), 1-23.

Stefanakis, E., & Patroumpas, K. (2008). Google Earth and XML: Advanced Visualization and Publishing of Geographic Information. In M. P. Peterson (Ed.), International Perspectives on Maps and the Internet (pp. 143-152). Berlin, Heidelberg: Springer.

Tolochko, R. C. (2016). Contemporary Professional Practices in Interactive Web Map Design (Thesis). University of Wisconsin-Madison, Madison, WI.

Tsou, M.-H. (2011). Revisiting Web Cartography in the United States: the Rise of User-Centered Design. Cartography and Geographic Information Science, 38(3), 250-257.

Turner, A. (2015, July 20). Vector Tiles preview. ArcGIS Blog.

World Wide Web Consortium (W3C). (2016). Accessibility. Retrieved March 17, 2017.