

# **THE BRAVE NEW WORLD OF ONLINE MAPPING**

Michael P. Peterson  
Chair, ICA Maps and the Internet Commission  
Professor, Department of Geography/Geology  
University of Nebraska at Omaha  
Omaha, NE 68182

## **Abstract**

A remarkable change took place in cartography with the introduction of the Mosaic World Wide Web browser in 1993. No longer tied to paper, millions of maps began to be distributed through computer networks. Another remarkable change has taken place since 2000 as new web services have become available for online mapping. These new methods facilitate expanded access to spatial information, increased levels of user input, real-time locational information, and greater integration of multimedia content with pictures, sound, and video. The relative ease of overlaying all types of information with online maps has further transformed cartography. For better or worse, the future of cartography is largely tied to the development of these tools for online mapping. A brave new world of online cartography is envisioned here in which web map services are used for all online maps, and spatial data providers integrate scale dependent spatial information as either vector or raster overlays.

## **1.0 Introduction**

Despite the increased availability of a variety of mobile navigation devices, the use of online maps is still growing. The four leading online map providers, AOL MapQuest, Google Maps, Yahoo! Maps, and MSN Bing Maps, report continued growth in usage. There is also a strong competition between these sites. MapQuest, the leading online street map provider in the US since 1996, still held a slight edge over Google Maps in early 2009 (Hopkins 2009). Introduced in 2005, and helped by the dominance of the Google search engine, Google Maps has become a major player in online mapping. Yahoo! Maps and MSN Bing Maps are in third and fourth place respectively. OpenStreetMap provides an open source web services option. All of these web services offer an Application Programmer Interface (API) for integration into web pages.

An API implements a web map service. Both are major aspects of the new Web 2.0, defined as a variety of innovative resources, and ways of interacting with, or combining web content. As with any new development, the long-term viability of these methods is based on the perceived functionality, user-interface, and sustainability. This paper examines the current state of development in online distributed mapping, particularly in reference to commercial web map services.

## **2.0 Web Services**

The World Wide Web consortium (W3C) defines a web service as "a software system designed to support interoperable machine-to-machine interaction over a network" (W3C 2004). Guah (2006, p. 9) refers to these services as the "small software components that are available over the Internet." They are normally implemented as an Application Programming Interface (API) that are accessed through the Internet and executed on a remote system that hosts the requested service (Wikipedia Web Services 2009).

In this service-oriented architecture (SOA), software is organized into independent and distributed service modules (Papazoglou 2003). Popular transport protocols for Web services include HTTP GET/POST and SOAP (Hugentobler, et.al, 2007). The Simple Object Access Protocol (SOAP) is the communication protocol that helps in transporting the Extensible Markup Language (XML) messages between the client and server. When the Web service client makes a request, the SOAP client application programming interface (API) constructs a corresponding XML message containing the remote method name and its parameters and sends the XML message over HTTP to the server hosting the Web services.

There are many advantages to web services. Rather than duplicating the efforts of others, a software developer simply needs to access the web service that performs the necessary function. The services promote the longevity of software by making components useful for multiple applications. Ultimately, web services help to reduce costs associated with the development, integration, and maintenance of software applications (Guah 2006).

Unsustainable software development is a major problem in the software industry. Most developers place too much emphasis on feature development and do not pay enough attention to the health of the underlying software. The result is "software with a high cost of change, that is increasingly unmaintainable, and where every change has the risk of destabilizing the product" (Tate 2005). A reliance on web services standardizes software development and distributes the cost of software maintenance across a wider user base.

Web services are available for many different applications. The OpenGIS Web Map Service Interface Standard (WMS) is a web service for requesting geo-registered map images from one or more distributed geospatial databases. A WMS request defines the geographic layer(s) and area of interest. Iosfescu-Enescu et. al. (2007) propose a number of cartographic extensions to OGC Web Map Services.

## **3.0 Map Mashups**

The application of web services is commonly referred to as a mashup. The most common type of mashup is the mapping of data. Like all web services, mashups are

based on Application Programming Interfaces (APIs), online libraries of functions that are made available at no cost to website designers. In a strict sense, a *map mashup* combines data from one website and displays it with a mapping API. The term has come to be used for any mapping of data using an API, even data supplied by the user.

There are a number of different, commercial-based APIs that are designed for online mapping, and even third party APIs that are built on top of these APIs and expand their functionality. A list of available mapping APIs is provided in Table 1. Most of these APIs are free to the developer and end user.

The major mapping APIs are Google Maps, Yahoo! Maps, MapQuest, and Microsoft Bing Maps. They all offer similar functionality and performance, although tests have shown that Yahoo! has slightly faster response times. The maps produced by all three services have different appearances. The imagery offered with Microsoft Bing Maps is acknowledged to be the most detailed. Bing Maps also includes an oblique view option for many areas that allows a point to be viewed from all four cardinal directions. New views added since the introduction of Google Maps include traffic maps, street views (for larger cities) and terrain maps. Of all of the mapping sites, Google Maps is the only service that has seen an appreciable gain in the number of users.

**Table 1.** Available mapping APIs

AOL MapQuest API	<a href="http://developer.mapquest.com/">http://developer.mapquest.com/</a>
Bing Maps (Microsoft)	<a href="http://www.bing.com/maps/">http://www.bing.com/maps/</a>
deCarta	<a href="http://developer.decarta.com/web/guest/home">http://developer.decarta.com/web/guest/home</a>
ESRI Arcweb Services geocoder API	<a href="http://www2.arcwebservices.com/v2006/index.jsp">http://www2.arcwebservices.com/v2006/index.jsp</a>
GlobeExplorer API	<a href="http://geocoder.us/help/">http://geocoder.us/help/</a>
GlobeXplorer API	<a href="http://www.digitalglobe.com/index.php">http://www.digitalglobe.com/index.php</a>
Google Maps API	<a href="http://code.google.com/apis/maps/index.html">http://code.google.com/apis/maps/index.html</a>
HeatMap API	<a href="http://www.heatmapapi.com/">http://www.heatmapapi.com/</a>
HopStop (mass transit)	<a href="http://www.hopstop.com/freeapi/">http://www.hopstop.com/freeapi/</a>
Mapstraction API	<a href="http://mapstraction.com/doc/">http://mapstraction.com/doc/</a>
MapTP (Map24)	<a href="http://developer.navteq.com/">http://developer.navteq.com/</a>
Microsoft Virtual Earth API	<a href="http://www.microsoft.com/maps/developers/">http://www.microsoft.com/maps/developers/</a>
MM Map API	<a href="http://www.mm.co.th/mmmmap/api">http://www.mm.co.th/mmmmap/api</a>
Multimap API	<a href="http://www.multimap.com/">http://www.multimap.com/</a>
NASA Imagery API	<a href="http://wms.jpl.nasa.gov/">http://wms.jpl.nasa.gov/</a>
Naver (South Korea)	<a href="http://dev.naver.com/">http://dev.naver.com/</a>
OpenLayers API	<a href="http://www.openlayers.org/">http://www.openlayers.org/</a>
PlaceBase Pushpin Le	<a href="http://www.pushpin.com/api/1.3/docs/">http://www.pushpin.com/api/1.3/docs/</a>
Plazes (social web)	<a href="http://plazes.com/">http://plazes.com/</a>
Poly9 Globe API	<a href="http://freeearth.poly9.com/api/documentation.php">http://freeearth.poly9.com/api/documentation.php</a>
Thematic Mapping API	<a href="http://thematicmapping.org/api/">http://thematicmapping.org/api/</a>
uMaps ActionScript API	<a href="http://www.afcomponents.com/">http://www.afcomponents.com/</a>

Urban Mapping API	<a href="http://developer.urbanmapping.com/">http://developer.urbanmapping.com/</a>
where2get>it	<a href="http://www.where2getit.com/">http://www.where2getit.com/</a>
Whereis API (AU & NZ)	<a href="http://www.whereis.com/">http://www.whereis.com/</a>
Yahoo! Maps API	<a href="http://developer.yahoo.com/maps/">http://developer.yahoo.com/maps/</a>

The advantage of using major online mapping sites as a base map for all maps that are produced is that these online maps represent a common and recognizable representation of the world. Overlaying features on top of these maps provides a common frame of reference for the map user. A particular advantage of making thematic maps with these online sites is the ability to spatially reference thematic data.

### 3.1 A Comparison of Google and Yahoo! Maps APIs

The major problem with current web map services is that the implementation varies between different services. Figure 1 shows a line-by-line comparison between a Yahoo! and Google Maps JavaScript implementation. The maps produced are identical – a map centered at a point with a clickable marker that opens a pop-up window (see Figure 2).

Both scripts start with the identical document type and html designations. After the title, the scripts define the user key. The key is free and is required if the script is executed through a website. It is not required for individual usage so that these scripts will work locally without the key. The numeric key allows Yahoo! and Google to keep track of the number of maps generated per user and stop any illegal use.

Both scripts then create a map object that stores all of the parameters associated with the map. Yahoo uses the YMap function while Google uses a function called GMaps2. Yahoo! defines the center of the map with YGeoPoint, while Google uses GLatLng. Following this, both scripts create the map type control (map, satellite, hybrid buttons in the upper-right of the map), the default map type (map, satellite or hybrid), and the zoom control. Yahoo! adds a pan function. This feature is automatic with Google. Following this, the marker is defined and the text associated with the pop-up bubble. Finally, the map is created in the body of the document by calling the initialize function. Google defines the size of the map in pixels (600 x 300) in the body. With Yahoo!, the div layer is defined in the head part of the script.

### 3.2 Generic Map Web Services APIs

While the Yahoo! and Google function calls are very similar, it would be difficult to make all of the changes required to switch between them. Fortunately, generic map web services APIs are available that make it possible to use a single script to access a number of different map web services. These are essentially API libraries that make calls to other API libraries.

One of these generic APIs is Mapstraction (Mapstraction 2009), an open source project that accesses multiple mapping APIs. Mapstraction makes it possible to dynamically switch between eleven different map providers (see partial list in Table 2). Differences between the mapping API implementations make it impossible for Mapstraction to translate all of the functions. But, for basic applications, such as the mapping of points (addmarker) or the addition of lines, it would be possible to easily switch between the different APIs. Another advantage of Mapstraction is the ability to take advantage of the functionality and speed of each API.

Yahoo! Maps API Implementation	Google Maps API Implementation
<pre> &lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"&gt; &lt;html xmlns=http://www.w3.org/1999/xhtml&gt;   &lt;head&gt;     &lt;title&gt;Yahoo! Maps API Example&lt;/title&gt;     &lt;script type="text/javascript" src="http://api.maps.yahoo.com/ajaxymap?v=3.8&amp;appid=YOUR _API_KEY"&gt;&lt;/script&gt;     &lt;style&gt; div#ymap {       width: 600px;       height: 300px;} &lt;/style&gt;     &lt;script type="text/javascript"&gt; function initialize_ymap() {   // Create a map object   var map = new YMap(document.getElementById('ymap'));    // Create latitude/longitude point   var pt=new YGeoPoint(41.258682,-96.012935);    // Add map type control   map.addTypeControl();    // Default map to hybrid (YAHOO_MAP_HYB) </pre>	<pre> &lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"&gt; &lt;html xmlns="http://www.w3.org/1999/xhtml"&gt;   &lt;head&gt;     &lt;title&gt;Google Maps API Example&lt;/title&gt;     &lt;script src=http://maps.google.com/maps?file=api&amp;v=2 &amp;amp;key= YOUR_API_KEY type="text/javascript"&gt; &lt;/script&gt;     &lt;script type="text/javascript"&gt;  function initialize() {   if (GBrowserIsCompatible())      // Create a map object     {       var map = new GMap2(document.getElementById("map"));      // Create latitude/longitude point     var pt = new GLatLng(41.258682,-96.012935);     map.setCenter(pt, 17);      // Add map type control     map.addControl(new GMapTypeControl());      // Default map to hybrid (G_HYBRID_MAP)     map.setMapType(G_HYBRID_MAP); </pre>

```

map.setMapType(YAHOO_MAP_SAT);

// Add zoom control
map.addZoomLong();

// Add the pan control
map.addPanControl();

// Display the map centered on the lat/long point
map.drawZoomAndCenter(pt, 1);

// Create a new marker at this point
var myMarker = new YMarker(pt);

// Content for pop-up window
var html = "Cartography and GIS
Laboratory<br>University of Nebraska at Omaha";

// Show pop-up when marker clicked
YEvent.Capture(myMarker, EventsList.MouseClick,
    function() {
        myMarker.openSmartWindow(html);
    });
// Add a label to the marker
myMarker.addAutoExpand("Click for more!");

// Put the marker on the map
map.addOverlay(myMarker);
}
</script>

```

```

// Add zoom control
map.addControl(new GLargeMapControl());

// Create a new marker at this point
var marker = new GMarker(pt);

// Content for pop-up window
var html = "Durham Science Center<br>University of
Nebraska at Omaha<br>Cartography and GIS Laboratory!";

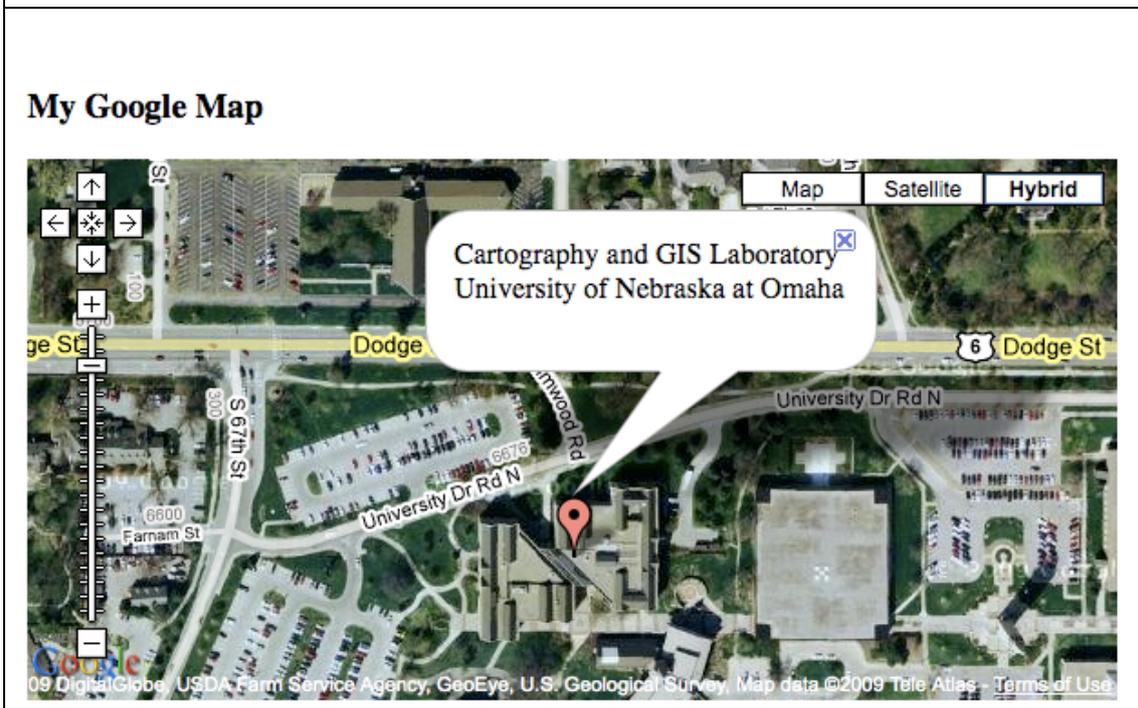
// Add a label to the marker
GEvent.addListener(marker, "click", function() {
    // Show pop-up when marker clicked
    marker.openInfoWindow(html);
})

// Put the marker on the map
map.addOverlay(marker); }
</script>
</head>
<h3>My Google Map</h3>

```

<pre>&lt;/head&gt; &lt;body onload="initialize_ymap()"&gt;   &lt;h3&gt;My Yahoo! Map&lt;/h3&gt;   &lt;div id="ymap"&gt;&lt;/div&gt; &lt;/body&gt; &lt;/html&gt;</pre>	<pre>&lt;body onload="initialize()" onunload="GUnload()"&gt;   &lt;div id="map" style="width: 600px; height: 300px"&gt;&lt;/div&gt; &lt;/body&gt; &lt;/html&gt;</pre>
---	---

**Figure 1.** Comparison between Yahoo! and Google Maps API Implementations. Both display a marker with an info bubble as shown in Figure 2.



**Figure 2.** Maps produced by Yahoo! Maps (top) and Google Maps (bottom) of the same location. Both use the same air photo, and the map is based on the same NAVTEQ vector database. The rendering of the database varies between the two companies, as can be seen with the different symbols and lettering.

**Table 2.** Cross-Mapping API Mapstraction functionality.

Method	Google	Yahoo!	Bing	OpenLayers	MultiMap	MapTP	MapQuest	Poly9
addAPI	√	√	√	√	√	√	√	√
resizeTo	√	√	√	√	√	√	√	
addControls	√	√		√	√		√	
dragging	√	√			√		√	
setCenterAndZoom	√	√	√	√	√	√	√	√
addMarker	√	√	√	√	√	√	√	√
removeMarker	√	√	√	√	√	√	√	
removeAllMarkers	√	√	√	√	√	√	√	
declutterMarkers					√			
addPolyline	√	√	√	√	√	√	√	
removePolyline	√	√	√	√	√	√	√	
removeAllPolylines	√	√	√	√	√	√	√	
getCenter	√	√	√	√	√	√	√	
setCenter	√	√	√	√	√	√	√	√
setZoom	√	√	√	√	√	√	√	
getZoom	√	√	√	√	√	√	√	
getZoomLevelForBoundingBox	√			√	√	√		
setMapType	√	√	√		√		√	
getMapType	√	√	√		√		√	
getBounds	√	√	√	√	√	√	√	
setBounds	√	√	√	√	√	√	√	√
addImageOverlay	√			√	√			
setImageOpacity	n/a							
setImagePosition	√				√			

addOverlay	√(G/K)	√ (G)	√ (G)					√ (G)
addTileLayer	√							
toggleTileLayer	√							
showRoute							√	
<b>Marker</b>								
Label	√		√		√		√	
Info Bubble	√	√	√	√	√	√	√	√
Info Div	√	√			√		√	√
Icon	√	√	√	√	√	√	√	√
Icon Size	√	√		√	√			
Shadow	√							
Hover Bubble	√					√		
Hover Icon	√							
Draggable	√							
<b>Polyline</b>								
Color	√	√	√	√	√	√	√	√
Width	√	√	√	√	√	√	√	√
Opacity	√	√	√	√	√		√	√
Polygon	√			√	√			
Fill Color	√			√	√			
Encoded	√							

<b>Notes</b>	* n/a - indicates this feature isn't provider specific
* G – GeoRSS; * K - KML	Source: <a href="http://mapstraction.com/features.php">http://mapstraction.com/features.php</a>

## Summary and Discussion

Web services represent a brave new world in software development. The services create a strong interdependence between software providers, developers and users. It requires a working Internet connection, and this connection must be relatively fast. The use of web services requires confidence that the tools will remain available, and free, and that they will not change. APIs that are built upon other APIs require yet another level of confidence that they will remain available in the future. It is indeed a brave new world.

Those who make data available in the form of maps now have more tools at their disposal. One can envision that such agencies will implement systems that make their data available as overlays on any existing web mapping service. In fact, there will likely be an expectation among map users that spatial data be made available in such a form because these maps will be viewed as “standard” maps of the world. It is therefore likely that any provider of information that uses online maps will need to create a background database that can interact with multiple web map services. The advantage, of course, is that providers of spatial information will not need to generate their own maps. The disadvantage is that the developer may have little control over the resultant map.

## References

- Guah, Matthew. (2006) Web Services. In Guah, Matthew W. and Currie, Wendy L. (2006) Internet Strategy: The Road to Web Services Solutions. Hershey, PA: IRM Press.
- Hopkins, Heather (2009) Google Maps Edges Closer To MapQuest. Hitwise Intelligence. Feb. 11, 2009. [[http://weblogs.hitwise.com/us-heather-hopkins/2009/02/google\\_maps\\_edges\\_closer\\_to\\_ma\\_1.html](http://weblogs.hitwise.com/us-heather-hopkins/2009/02/google_maps_edges_closer_to_ma_1.html)]
- Hugentobler, M, Iosifescu-Enescu, I., Hurni, L., (2007) A Design Concept for Implementing Interoperable Cartographic Services based on reusable GIS Components. Proceedings of ICC 2007. Moscow.
- Mapstraction (2009). JavaScript Mapping Abstraction Library. [<http://www.mapstraction.com/>]
- Open Geospatial Consortium (OGC). (2009) Web Map Service. [<http://www.opengeospatial.org/standards/wms>]
- Papazoglou M. (2003) Service-Oriented Computing: Concepts, Characteristics and Directions. Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE 2003).
- Tate, Kevin. (2005) Sustainable Software Development: An Agile Perspective. Indianapolis: Addison-Wesley Professional.
- W3C (2004) Web Services Glossary. [<http://www.w3.org/TR/ws-gloss/>]
- Wikipedia (2009) Web Services. [[http://en.wikipedia.org/wiki/Web\\_services](http://en.wikipedia.org/wiki/Web_services)]