

# CARTOGRAPHIC WEB APPLICATIONS – DEVELOPMENTS AND TRENDS

Dr. Olaf SCHNABEL  
[oschnabel@ethz.ch](mailto:oschnabel@ethz.ch)

Prof. Dr. Lorenz HURNI  
[lhurni@ethz.ch](mailto:lhurni@ethz.ch)

Institute of Cartography, ETH Zurich  
Wolfgang-Pauli-Str. 15  
8093 Zurich, Switzerland

## **Abstract**

In the last 5 years, new techniques are conquering the internet and the web mapping domain. New input devices such as multitouch devices with an interesting potential for cartographic navigation applications appeared, while mobile devices are used for Location Based Services and to access the web. Generally, web cartography shifted to a distributed and service-oriented cartography, serving individual maps on-demand for each single purpose. Since in this process the demands of map users on web maps changed from more general to individual information visualizations, also the web techniques in the background changed. What can cartographers expect from new techniques such as spatial databases, JSON, Web Map Services or Rich Internet Applications (RIA)? What are the potential and limits of the new techniques for cartographic web applications? In this paper, an overview of the current technical developments is given and their influence on web cartography and their applications as well as advantages and disadvantages are discussed.

## **Introduction**

In the last 5 years, new techniques, terms and acronyms are conquering the internet and therefore the field of web mapping. On the hardware side, new input devices such as multitouch devices with interesting potential for cartographic navigation applications appeared, while more and more mobile devices are used to access the web, to navigate and to use Location Based Services (Peterson 2008). Generally, web cartography shifted to a distributed and service-oriented cartography, serving individual maps on-demand for each single purpose. Since in this process the demands of a map user on a web map changed from a more general to a more individual information visualization, also the web techniques behind need to be and already do change. This evolution of web mapping techniques started in the early 90s of the last century where static raster maps were presented via internet. Few years later, interactive raster maps evolved. Areas in the ras-

ter graphic, the so called "image maps", pointed to other sources in the internet. Additionally, raster layers with transparent background were combined in a web page and could be made interactive via JavaScript. Around the year 2000, the first vector-based web maps made with SVG were introduced (Neumann and Winter 2000). In the years 2003 and 2004, it became common knowledge to separate the vector-based geometry of the map from attribute data stored in a database. Two years later even the map geometries were usually stored in a spatial database such as PostgreSQL with PostGIS extension. Today, Web Map Services dominate the market since the combination of different data sources requires a technical standardization.

Since it is difficult for cartographers to evaluate each new technique regarding its use for the cartographic production, in the following chapters an overview of the current technical developments in the fields of geodata storage, representation with web graphics, dissemination of geocontent, user interfaces as well as 3D cartography is given.

### **Geodata Storage: GIS and Spatial Databases**

Since the cartographic web applications serve maps for each single purpose but nevertheless need to be up-to-date, relational or object-oriented databases will be used in the background to feed the applications with data because of their easy updatability. A tendency to use databases with spatial extension even for small projects can be observed. Typical commercial databases are Oracle with the spatial extension Oracle Spatial and DB2 with the Spatial Extender. But also open source databases such as PostgreSQL in combination with the spatial extension PostGIS are used more and more in a productive environment. These spatial databases have two main advantages: they can store attribute data as well as the geometry and they are capable to do extremely fast spatial analysis, the classical GIS domain, by using simple SQL commands. The range of spatial functions is limited but covers buffer, intersection and union functions.

On the other hand, also GIS moves towards the web and to spatial databases. While a few years ago geodata were stored in Shapefiles, today geodatabases (e.g. the file geodatabase of ArcGIS) are used or the data are directly stored in the above mentioned spatial databases.

Even in small projects, whether located in the web or offline on CD-ROM, DVD or memory stick, small local offline databases such as SQLite with spatial extension SpatiaLite are used to do fast searches and serve data on demand. Currently, SQLite is used by major browsers such as Mozilla Firefox and Safari as well as in operating systems for mobile phones such as Google Android, Apple iPhone or Symbian (SQLite 2009). Additionally, user interfaces such as Adobe AIR use the database to react fast to user inputs.

## Geodata Representation: Vector Graphics for the Web

With the exception of Web Map Services, in the web cartography a general shift from raster to vector graphic formats can be observed to improve the graphical quality of web mapping applications. This development started around 1995 with the first ideas of using 2D vector graphic formats in browsers by Chris Lilley (Ferraiolo 2008). In 1996, Macromedia Flash became a popular but proprietary and binary vector graphic format with support for animation and later with interactivity via ActionScript, an ECMAScript derivative. In 1998, the "eXtensible Markup Language" (XML) was specified and standardized. Based on this language, different companies and consortiums tried to standardize their XML based vector graphic formats, for example the "Vector Markup Language" (VML) by Microsoft, Autodesc, HP, Macromedia, Visio Corporation or the "Precision Graphics Markup Language" (PGML) by Adobe, IBM, Netscape, Sun (Ferraiolo 2008). But none of the formats could penetrate the whole market and reach a significant market share. Meanwhile, the W3C developed on basis of the two proposals the "Scalable Vector Graphics" (SVG) format, which was standardized in 2001, can use geographical coordinates and included parts of other specifications such as DOM, XML, CSS, SMIL and XLink for linking, animation and access via scripting languages (W3C 2009). This format is natively implemented in nearly all major browsers except the Internet Explorer (Table 1).

|  |   |        |
|--|---|--------|
| <b>Browser</b>                             | Opera 9.50  | 94.16% |
|  | Mozilla Firefox 3.0                                 | 60.40% |
|  | Mozilla Firefox 3.5                                 | 66.42% |
|  | Google Chrome 2.0                                   | 81.39% |
|  | Apple Safari 4                                      | 81.39% |
| <b>Browser Plugins<br/>(mainly for IE)</b> | ReneSis 1.1   | 58.73% |
|  | GPAC 0.4.5  | 64.96% |
|  | Corel SVG Viewer 2.1 (discontinued support)         | 61.27% |
|  | Adobe SVG Viewer 3.03 (discontinued support)        | 83.03% |
|  | Adobe SVG Viewer 6 pre-alpha (discontinued support) | 86.13% |
| <b>Stand-alone Viewers</b>                 | Batik 1.7   | 93.61% |
| <b>Mobile Phone<br/>OS and Viewers</b>     | Ikivo SVG Viewer                                    | yes    |
|  | Opera Mini  | yes    |
|  | BitFlash SVG Viewer                                 | yes    |
|  | Flash Lite 3.1 Viewer                               | yes    |
|  | iPhone  | 81.39% |
|  | Android   | 81.39% |

Table 1. SVG support in browsers, viewers and mobile phones, state: March 2009 (Schiller 2009)

Since a few years, Microsoft tries to push a new XML based vector graphic format on the market, the XAML (eXtensible Application Markup Language) format. XAML is a 2D vector graphic subset of the graphic library WPF (Windows Presentation Foundation). The markup code has nearly the same syntax as SVG but extends it with form elements. XAML does not support geographical coordinate systems (Hauser 2008). A browser plugin is necessary to view the XAML content (Silverlight for Windows and Mac platforms, Moonlight for Linux platforms). An additional option for web cartographers is the use of the Canvas format for simple drawings via JavaScript which is integrated in the upcoming HTML 5 standard. Even now, Canvas is natively supported by all major browsers except Internet Explorer. Since also Flash and the Java 2D library can be used to visualize web maps, none of the formats dominate the market (Table 2). Nevertheless, SVG as vector graphic standard is integrated in most of the web workflows as intermediate format or output and can therefore be recommended for cartographers.

|        |                   |     |
|--------|-------------------|-----|
| SVG    | - (plugin for IE) | yes |
| XAML   | plugin            | no  |
| Canvas | - (plugin for IE) | no  |
| Java2D | Java runtime      | no  |
| Flash  | plugin            | no  |

Table 2. 2D vector graphic formats in the web

### **Geodata Dissemination: Standardization versus Individualization**

One of the main tasks of a cartographer is the use of different techniques to visualize geospatial data and bring the map to the user. In the last decades, a tendency to use distributed data sources and to act and collaborate globally could be observed (Sykora et al. 2007). This is mainly caused by the will of governments to solve environmental problems (see INSPIRE directive of the European Union) and the will of companies to stay competitive in the global market. But even single users share their knowledge globally with the help of social networks such as Facebook or mapping portals such as Open Street Map. Therefore, a standardization was necessary to bring the different data sources together and visualize them in the way the user want. As a result, Web Services such as Web Map Services (WMS) were developed and standardized by the OGC in the year 2000 (OGC 2009a). A Web Service can publish meta information of existing data, visualize maps on demand and answer requests for underlying data (Figure 1). Typical Map Server products for the publication of maps via Web Services are ESRI ArcGIS Server, UMN Map Server, GeoServer or QGIS Map Server.

Nevertheless, current web services have their limits. For example, they lack the capability to visualize multiple data values as diagrams (Schnabel and Hurni 2007). In the project SANY, this problem is getting to be solved with a new "Map and Diagram Service", an extension of the current OGC WMS standard (SANY 2009). With the further extension of standards, a development to a general cartographic description language can be foreseen. Meanwhile, cartographic web services will be used in "Service Chains" to model human mapping workflows.

As side effect of the standardization, also an individualization process of web maps takes place. Since several years a tendency to build web maps for a specific purpose could be observed but standardization makes it easier for users to build individual mashups (combinations) of different Web Services (e.g. Google Maps with overlaying data). Therefore, the individualization process is increased by these techniques.

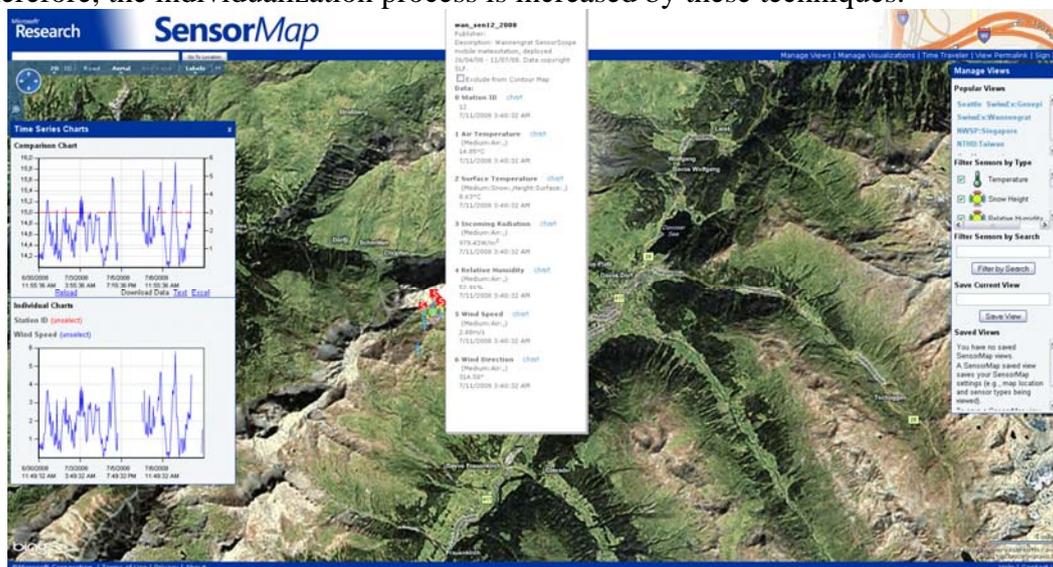


Figure 1. Example of a mashup of WMS from different sources (background map, sensor data)

Furthermore, the duration of validity of the map decreased rapidly. Especially in the environmental sector, web maps tend to be used in time-critical situations. A typical example of such an application is the real-time flow map of the Swiss river network Lienert et al. (2008). Nevertheless, the updating of Web Map Services and Web Feature Services may cost some time, since the whole Web Service is loaded again during the update. But in most of the cases, only small parts of the maps need to be updated. Furthermore, in a modern web application a continuous interaction between the web map and the server (e.g. the database) takes place to react on user inputs. Therefore, other techniques have to be used to exchange the data. This exchange is typically done with XML based containers, for example with the Simple Object Access Protocol (SOAP). Table 3 shows typical XML based techniques for the exchange of geodata: GML (Geography Markup Language, OGC and ISO standard since 2007), KML (Keyhole Markup Language, OGC standard since 2008 and widely used in Google products) and GeorSS

(Really Simple Syndication, Simple and GML version) (OGC 2009b, OGC 2009c, GeoRSS 2009).

Nevertheless, the XML containers tend to have a large file size and the parsing of the XML structure is very time-consuming. To overcome the problem, in 2005 JSON (JavaScript Object Notation) was developed and is now widely used. A derivative is GeoJSON for the geodata dissemination and storage for small web applications (Butler et al. 2008). Table 4 shows an exemplary GeoJSON notation of the same example from Table 3. This ECMAScript based containers store data in arrays which massively decreases the geodata transfer rate. This development is very important for the cartography, since map layers such as rivers or borders contain huge amounts of geographical data. A further option is the use of the WKT format (Well-Known Text, part of the OGC Simple Feature Access Specification) which is often used in spatial databases such as PostgreSQL/PostGIS for the description of 2D geometries.

GML



```
<gml:featureMember xmlns:gml="http://www.opengis.net/gml"
xsi:schemaLocation="http://www.opengis.net/gml
http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0
/gmlsf.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <feature:feature xmlns:feature="http://example.com/feature">
    <feature:geometry>
      <gml:Polygon><gml:exterior><gml:LinearRing>
        <gml:posList>40.078125 8.0859375 22.5 14.4140625 -
14.765625 35.5078125 29.53125 49.5703125 74.53125
34.8046875 40.078125 8.0859375</gml:posList>
      </gml:LinearRing></gml:exterior></gml:Polygon>
    </feature:geometry>
  </feature:feature>
</gml:featureMember>
```

KML



```
<kml xmlns="http://earth.google.com/kml/2.0">
  <Folder>
    <name>OpenLayers export</name>
    <description>Exported on Mon Jun 22 2009 15:33:03
GMT+0200</description>
    <Placemark>
      <name>OpenLayers.Feature.Vector_128</name>
      <description>No description available</description>
      <Polygon><outerBoundaryIs><LinearRing>
        <coordinates>40.078125, 8.0859375 22.5, 14.4140625 -
14.765625, 35.5078125 29.53125, 49.5703125 74.53125,
34.8046875 40.078125, 8.0859375</coordinates>
      </LinearRing></outerBoundaryIs></Polygon>
    </Placemark>
  </Folder>
</kml>
```

|               |   |   |
|---------------|---|---|
| GeoRSS Simple |  | <pre>&lt;item xmlns="http://backend.userland.com/rss2"&gt;   &lt;title&gt;&lt;/title&gt;   &lt;description&gt;&lt;/description&gt;   &lt;georss:polygon     xmlns:georss="http://www.georss.org/georss"&gt;8.0859375     40.078125 14.4140625 22.5 35.5078125 -14.765625     49.5703125 29.53125 34.8046875 74.53125 8.0859375     40.078125&lt;/georss:polygon&gt; &lt;/item&gt;</pre> |
|---------------|---|---|

Table 3. GML, KML, GeoRSS - the different XML based formats for the dissemination of geodata

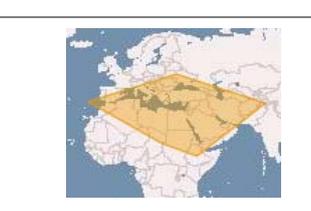
|          |   |  |
|----------|---|--|
| Geo-JSON |    | <pre>{ "type": "Feature",   "id": "OpenLayers.Feature.Vector_128",   "properties": {},   "geometry": { "type": "Polygon", "coordinates": [[[40.078125,     8.0859375], [22.5, 14.4140625], [-14.765625, 35.5078125],     [29.53125, 49.5703125], [74.53125, 34.8046875], [40.078125,     8.0859375]]]],   "crs": { "type": "OGC",     "properties": { "urn": "urn:ogc:def:crs:OGC:1.3:CRS84" } } }</pre> |
| WKT      |  | <pre>POLYGON((40.078125 8.0859375, 22.5 14.4140625, - 14.765625 35.5078125, 29.53125 49.5703125, 74.53125 34.8046875, 40.078125 8.0859375))</pre>  |

Table 4. GeoJSON and WKT code samples

Additionally, these techniques are applied in the field of mobile cartography to reduce data transfer rates. Here, a strong tendency to use mobile devices for the web access and for car, bike or pedestrian navigation could be observed (Peterson 2008). With the use of individual mobile devices, more individual inquiries on cartographic services such as Location Based Services are requested, e.g. to navigate to a railway station or find a restaurant in the surrounding area (Reichenbacher 2004). To successfully provide such a service, four problems need to be solved. The first problem is the determination of the users position which is usually done by an integrated GPS module. Since only a fractional amount of mobile devices has GPS (Peterson 2008), the user location can be triangulated from nearby cell towers by using the signal strength. In urban areas, large improvements were made through the use of WLAN hotspots and mobile networks such as UMTS to determine the position of the user even in buildings (Retscher et al. 2006). Until now, only one of the above mentioned techniques can be used per mobile device. Therefore, a future task is the combination of these different navigation techniques in a single device. The second problem is the transfer of geodata (e.g. the position, the

nearest restaurants and the background map) from the server to the mobile device, since the cellular network has only a limited band width. This problem is solved with the help of the above mentioned dissemination techniques. Thirdly, the geodata have to be displayed on the mobile device screen. This can be done by using the integrated browser (usually a light-version of a standard browser) or with separate viewers. Even if mobile devices have only a small processing power, vector data such as SVG can be displayed very fast. Table 1 lists some SVG viewers which are designed for the use in mobile devices. The fourth problem is mainly caused by the mobile device hardware. The small screens of the devices (usually 160x120 pixels, see Peterson 2008) lead to generalization problems, environmental influences such as sun light disturbs the perception of the map. These problems are open tasks.

### **User Interfaces for Web Maps: Rich Internet Applications**

Until 2005, the whole web application must be downloaded to the client computer and executed afterwards, or all calculations and interactions were done on the server. But in 2005, a new term conquered the market: AJAX (asynchronous JavaScript and XML). AJAX is basically an ECMAScript which loads only the necessary parts of a web application from the server. Parts of the application, e.g. map layers or tooltip information, can be included or updated on-demand. Additionally, XML based descriptions of Graphical User Interfaces (e.g. buttons and other form elements) were introduced in 2006. The combination of these two techniques allowed the development of flexible web mapping applications. These applications provided through their on-demand behaviour an immediate feedback for the user and more possibilities to interact between map and GUI elements (e.g. pressing the zoom button extends both map and reference map extend). Furthermore, video, audio, 2D and 3D graphics and animation could easily be added via scripting to the web application. Therefore, they were named Rich Internet Applications. Currently, three big companies provide such a framework for the creation of Rich Internet Applications: Microsoft Silverlight, Adobe Flex / AIR and Sun JavaFX.

|  |  |  |   |
|--|--|--|---|
| <b>XML based GUI descr.</b>                  | XAML   | MXML   | JavaFX ML   |
| <b>scripting language</b>                    | C#, Visual Basic, JavaScript, ...  | ActionScript (ECMAScript derivative)   | JavaFX Script (ECMAScript derivative)   |
| <b>prerequisites for 1. web, 2. desktop</b>  | 1. Silverlight or Moonlight plugin<br>2. Silverlight   | 1. Flash plugin (version 9+),<br>2. Adobe Integrated Runtime - AIR   | 1. Java plugin (version 6+),<br>Java Web Start<br>2. JavaFX Desktop runtime   |
| <b>tools for 1. developers, 2. designers</b> | 1. XML or text editor, MS Visual Studio<br>2. MS Expression Blend                                      | 1. Flex Builder (new: Flash Builder)<br>2. Flash Professional, Flash Catalyst  | 1. JavaFX SDK, Netbeans, Eclipse<br>2. JavaFX Production Suite, JavaFX Graphics Viewer  |
| <b>framework</b>                             | .NET Framework   | Flex SDK   | JavaFX SDK  |
| <b>workflow</b>                              | creation of XAML files, compilation as binary XAP file, presentation via plugin or desktop application | creation of FLA files and/or ActionScript classes or conversion of Illustrator or Photoshop images to ActionScript classes, compilation as binary SWF file, presentation via plugin or desktop application | creation of JavaFX code or conversion of Illustrator or Photoshop images to JavaFX classes, compilation as JNLP or JAR file, presentation via plugin or desktop application |

Table 5. Features of Rich Internet Applications

Usually, a browser plugin is needed and JavaScript should be enabled in the browser to show Rich Internet Applications. But in the last months all three companies presented platform-independent stand-alone desktop applications such as Adobe AIR which allows to run Rich Internet Applications on the desktop. Therefore, developers can reduce their programming effort (one code for both desktop and browser) and get access to the local file system. Table 5 shows several features of the three frameworks.

For cartographers, the design tools are interesting since with no or less programming effort web mapping applications with rich features such as video or audio can be created. But since all of the frameworks need plugins, the use of specialized toolkits are more common for cartographic web applications. These toolkits usually work only in the browser and use AJAX, HTML with embedded Web Map Services (raster) or SVG for map and GUI. Common toolkits for web cartography are

- Google Maps API (GUI + Map)
- MS Virtual Earth API (GUI + Map) (new: bing maps)
- Yahoo! Maps API (GUI (YUI) + Map)
- OpenStreetMap API (Map)
- OpenLayers API (GUI)
- carto.net SVG API (GUI)

Other toolkits such as prototype, dojo or jQuery provide also a wide spectrum of GUI elements but use the above mentioned APIs for the presentation of maps. Typical toolkits which are adapted to the cartographers needs are the Google Maps API or the carto.net SVG/ECMAScript framework of Andreas Neumann (Neumann 2009, see example in Figure 2). The OpenLayers API are useful for the embedding of WMS but other geometries cannot be easily integrated. Since the used API or background map respectively from one data source (e.g. Google) and the overlaying data from another data source (the user) are mixed, the resulting web map application is a "Mashup". Mashups are combinations of different data sources and are widely used in the web, e.g. in the customer relationship management or resource planning (Lörracher 2009).

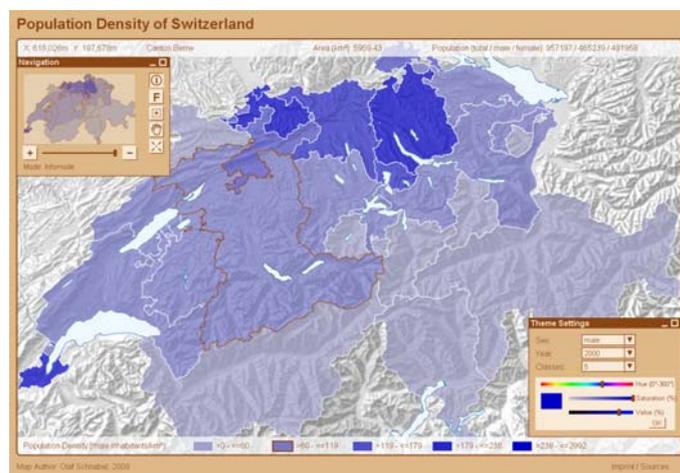


Figure 2. Map "Population Density of Switzerland" using the carto.net SVG API

### Web Cartography goes 3D

The development and use of new in- and output devices has a huge impact on the manner how the user navigates through geospatial data. At the end of 2006, the first game controller with motion and acceleration sensors was developed by Nintendo, the Wii control. This kind of device can be seen as a kind of three-dimensional mouse, which allows to navigate in 3D environments, e.g. in Google Earth or other 3D globes such as the globe in the Swiss World Atlas. These new input devices also require new navigation concepts. In the meantime, other devices such as 3D cameras with integrated face and speech detection and infrared sensors which detects finger movements will be developed (Gieselmann et al. 2009). Together with the development of the first 3D screen by the "Dresden 3D Display" GmbH and the TU Dresden (Kirschenbauer and Buchroithner 1999) or the 3D displays of the big display manufacturer AUO, the prerequisites for the use of 3D web maps are given. These 3D displays serve stereoscopic images without glasses by using lenticular lens technology. But also a description format and a rendering component were necessary to complete the 3D visualization workflow. Even here, a standardization process can be observed. In 1995, the "Virtual Reality Modeling Language" (VRML) as text-based format was developed by the company Silicon Graphics.

Since the visualization of 3D objects needed very high hardware prerequisites, the format never penetrated the market but was established as exchange format between the programs. In 2004, the "eXtensible 3D" (X3D) format was standardized, several companies such as Cortona or Octaga provided browser plugins. Since the same performance problems occurred, the market share stayed very low. As side effect, 3D effects in 2D web maps rised, e.g. the so called bump mapping which uses the visual perception effects to simulate surface roughness and therefore improve the readability of the web maps. With the OGC and ISO standardization of the Geographic Markup Language (GML) in the year 2000 and its derivatives CityGML, a global exchange format was established which is widely used in the field of geodata exchange and management. But not until Google with their Keyhole Markup Language (KML) and Google Earth viewer in 2004 penetrated the market, 3D maps were used by a wide range of cartographers and users. Examples of 3D city models can be found all over the web. With the development of Microsoft's Virtual Earth 3D plugin as well as the Google Earth plugin in 2008 and with the help of scripting APIs for interactive 3D maps, the 3D cartography is finally arrived in the web.

### **Conclusion: The Future of Web Cartography**

In the future, multitouch devices such as Apple iPhone and motion sensors will serve as human-computer-interface for web maps, allowing the navigation with fingers and several "touches" at the same time. Also, the fusion of desktop, browser and even web server will become reality (example: Opera Unite) which will simplify the production of web maps. Additionally, web standards such as HTML and SVG will be mixed with each other and geodata will be stored in local or remote databases. An increasing number of cartographic web services and web applications with service character will be available in the internet, e.g. the Color Brewer or the Map Symbol Brewer (Brewer and Harrower 2002, Schnabel 2007). They will be combined in a specific order in immediately responding Rich Internet Applications. These composite web applications will connect the human mapping workflow processes with the processes or services in web applications.

### **References**

- Brewer, C.A., Harrower, M., 2002. Color Brewer. Available at: <http://www.colorbrewer.org> [Accessed June 2009].
- Butler, H., Daly, M., Doyle, A., Gillies, S., Schaub, T., Schmidt, Ch., 2008. The GeoJSON Format Specification. Available at: <http://geojson.org/geojson-spec.html> [Accessed June 2009].
- Ferraiolo, J., 2008. How Ajax Changes the Game for SVG. SVG Open Conference, Nuremberg, 2008. Available at: [http://www.svgopen.org/2008/papers/63-How\\_Ajax\\_Changes\\_the\\_Game\\_for\\_SVG/](http://www.svgopen.org/2008/papers/63-How_Ajax_Changes_the_Game_for_SVG/) [Accessed June 2009].
- GeoRSS, 2009. GeoRSS. Available at: <http://www.georss.org> [Accessed June 2009].

- Gieselmann, H., Austinat, R., Bonnert, E., 2009. Der Minderheiten-Bericht: Träume von der Zukunft der Videospiele auf der E3. In *c't*, 14, pp. 38-39.
- Neumann, A., Winter, A.M., 2000. Vector-based Web Cartography: Enabler SVG. Available at: [http://www.carto.net/papers/svg/index\\_e.shtml](http://www.carto.net/papers/svg/index_e.shtml) [Accessed June 2009].
- Neumann, A., 2009. Carto.net SVG Framework. Available at: <http://www.carto.net/papers/svg/gui> [Accessed June 2009].
- Hauser, T., 2008. SVG, Silverlight, and Flex: Similarities and Differences. SVG Open Conference, Nuremberg, 2008.
- Kirschenbauer, S., Buchroithner, M.F., 1999. "Real" 3D Technologies for Relief Depiction. In *Proceedings of the 19th International Cartographic Conference*, Ottawa. Available at: [http://www.mountaintopography.org/publications/papers/ica\\_cmc\\_sessions/1\\_Ottawa\\_Session\\_Relief/01\\_Ottawa\\_Kirschen\\_Buchroithner.pdf](http://www.mountaintopography.org/publications/papers/ica_cmc_sessions/1_Ottawa_Session_Relief/01_Ottawa_Kirschen_Buchroithner.pdf) [Accessed June 2009].
- Lienert, Ch., Schnabel, O., Hutzler, E., Hurni, L., 2008. A Real-Time Flow Map of the Swiss 1:200,000 River Network. In *Proceedings of the 6th ICA Mountain Cartography Workshop*, Lenk, pp. 147-153.
- Lörracher, T., 2009. Decision Support Using Map-Based Enterprise Mashups. In *Kartographische Nachrichten*, 2, pp. 70-77.
- OGC, 2009a. Web Map Service. Available at: <http://www.opengeospatial.org/standards/wms> [Accessed June 2009].
- OGC, 2009b. Geography Markup Language. Available at: <http://www.opengeospatial.org/standards/gml> [Accessed June 2009].
- OGC, 2009c. KML. Available at: <http://www.opengeospatial.org/standards/kml> [Accessed June 2009].
- Peterson, M.P., 2008. Trends in Internet and Ubiquitous Cartography. In *cartographic perspectives*, 61, pp. 36-49.
- Reichenbacher, T., 2004. Mobile Cartography – Adaptive Visualisation of Geographic Information on Mobile Devices. Ph. D. Institute of Photogrammetry and Cartography: TU Munich.
- Retscher, G., Moser, E., Vredeveld, D., Heberling, D., 2006. Performance and Accuracy Test of the WLAN Indoor Positioning System "ipos". In *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*, Hannover. Available at: [http://www.wpnc.net/fileadmin/WPNC06/Proceedings/6\\_Performance\\_and\\_Accuracy\\_Test\\_of\\_the\\_WLAN\\_Indoor\\_Positioning\\_System\\_IPOS.pdf](http://www.wpnc.net/fileadmin/WPNC06/Proceedings/6_Performance_and_Accuracy_Test_of_the_WLAN_Indoor_Positioning_System_IPOS.pdf) [Accessed June 2009].
- SANY, 2009. Sensors Anywhere. Available at: <http://www.sany-ip.eu> [Accessed June 2009].
- Schiller, J., 2009. Codedread. Available at: <http://www.codedread.com/svg-support.php> [Accessed June 2009].
- Schnabel, O., 2007. Map Symbol Brewer. Available at: <http://www.carto.net/schnabel/mapsymbolbrewer/> [Accessed June 2009].
- Schnabel, O., Hurni, L., 2007. Diagram Markup Language - A new Model for

Symbolization in Internet Maps. In *Proceedings of the 23rd International Cartographic Conference*, Moscow.

SQLite, 2009. Well-Known Users of SQLite. Available at:

<http://www.sqlite.org/famous.html> [Accessed June 2009].

Sykora, P., Iosifescu Enescu, I., Schnabel, O., Hurni, L., 2007. Extended Cartographic Interfaces for Open Distributed Processing. In *Cartographica*, 42(3), pp. 209-218.

W3C, 2009. Scalable Vector Graphics. Available at: <http://www.w3.org/Graphics/SVG/> [Accessed June 2009].