Best practices for the development and serving of web map services

by Hermanus Brynard, Geological/GIS Consultant

Abstract

Numerous organisations in South Africa and elsewhere have been gathering valuable spatial and attribute data over decades, mostly for their own use, but much of this data has a multidisciplinary application and constitutes an essential component in the scientific pursuits of other organisations. However, much of this digital data is still only directly available to users within the custodian organisation. This restricted access to data leads to ineffective use of resources and is a retardant for growth in a country where well-trained human resources, and funding, are in short supply. The need exists in many organisations for user-friendly web map services systems to be put in place to make this data available on the internet.

With the advent of scientific and technological development robust computing and information systems and satellite imagery have been developed that did not exist before. Modern software systems provide the facilities to readily serve spatial data on the web where it can be made readily accessible either to selected clients or to a wider audience.

An essential component for the serving of spatial data is the availability of an on-line high-quality base map service in the form of a street map and/or imagery that provides a geographic frame of reference for orientation and navigation. This obviates the need for developers and cartographers to spend valuable time to compile accurate, up to date and presentable maps. The use of these maps is usually free when they are not applied for commercial purposes. The one draw-back of some on-line maps is that they are not regularly updated to keep pace with the rapid process of name changes in South Africa.

Furthermore, an application programming interface (API) is required that offers the functionality to overlay the base map with the organisation’s operational (or thematic) map service layers or integrate map service layers from other sources. Controls are also required to navigate around the map or for changing the layer display sequence and adjusting layer transparency. It may even be a requirement for the overlay of ASCII-based geographic point data from a file on the user’s computer onto the base map. The use of pure DOM and HTML elements such as drop-down menus may be required to facilitate rapid navigation to a predefined listed location such as a province, local municipality, town or a cadastral boundary or provide links to peripheral sources of information.

The author has for several years been involved in the development of web map services and web applications, mostly in the geological field. This paper discusses some of the systems available to develop web services. Some of the more popular APIs to integrate these web services into a web application are discussed as well as some open source APIs, for which varying computer and programming skills are required. The paper also addresses best practices available and the pitfalls that exist in the development of both web services and applications, for example the influence of the web browser on the application design.

The paper is concluded with some examples of good and useful existing web applications in the South African and international context.

Keywords

web mapping, web map services, application programming interface

Introduction

Many organisations in South Africa and abroad that are involved in scientific and other pursuits have been gathering valuable spatial and attribute data over decades, mostly for their own use. Few scientific disciplines function independently in the sense that they do not require inputs from other disciplines. Much of the data generated by the custodian organisations require inputs from other disciplines or their data constitutes an essential component in the scientific pursuits of other organisations. As an example we may consider the compilation of the map of the vegetation of South Africa, Lesotho and Swaziland [7] that was published by the South African National Biodiversity Institute (SANBI). For this map considerable input of detailed geology and climatic data was required due to the dependence of vegetation on the underlying rock strata, but also on the macro- and microclimate, and other factors. The information contained in this map may again be applied by geologists for mapping due to the association of certain rock types with certain vegetation, or by biologists for the study of mammal distribution which may have a relationship to vegetation.

Much of the digital data within organisations is still only directly available to users within the custodian organisation. This restricted access to data leads to ineffective use of resources and is a retardant for growth in a country where well-
trained human resources and funding are in short supply. The need exists in many organisations for user-friendly web map services systems to be implemented to make this data available on the internet. Serving data in this way ensures that the data is always current and up to date.

This paper outlines the concept of web mapping, discusses some tools to set up these map services and the best practises to integrate map services from various sources into a single web application. Some examples are presented of various web map services originating from South Africa and elsewhere.

Web mapping

Addressing the 24th International Cartographic Conference in Santiago, Chile during January 2010, Mike Peterson said, "For better or worse, the future of cartography is largely tied to the development of tools for online mapping".

The essential components to enable web based mapping are:

- A web map server – either in an intranet environment or connected to the internet through which the web map services can be published
- Web map services – comprising base and operational map layers
- An application programming interface or viewer where the layers are integrated
- Guidelines for the most efficient methods to serve the data

The organisation’s decision as to what technology should be embraced for the provision of web map services depends on a number of factors:

- The adopted technology within the corporate environment
- Communication infrastructure
- Availability of data (raster or vector)
- Funding
- Know-how or technical expertise available within the organisation

With the advent of scientific and technological development, robust computing and information systems and satellite imagery have been developed that did not exist before. Modern software systems provide the facilities to serve spatial data on the web where it can be made readily accessible either to selected clients or to a wider audience. What is really required is corporate commitment, and champions within the organisation to drive the process.

Web map servers

Basically two approaches to web map services exist; the use of proprietary or commercial-off-the-shelf and open source software. It is not the purpose of the author to discuss the merits of the various approaches, since either approach will require some technological skills and an internet infrastructure and the decision regarding the best implementation will be largely dictated by the factors listed above.

Commercial map servers

Several commercial systems are available today of which a few are listed below (Source: [http://en.wikipedia.org/wiki/Web_Map_Service](http://en.wikipedia.org/wiki/Web_Map_Service)), but no in-depth discussion will be given of any of these:

- ArcObjectFX Web Mapping Tools
- ArcGIS Server from Esri
- ArcIMS(ArcIMS 10 was the last release, and its functionality has been replaced by ArcGIS Server)
- VT MAK's VR-TheWorld
- GeoWebPublisher from Bentley Systems
- GeognoSIS from Cadeorp
- GeoMedia
- OracleMapViewer
- LizardTech's Express Server
- SIAS (Smallworld Internet Application Server) from GE Energy
- Autodesk's Infrastructure Map Server
- Erdas, Inc's Apollo Suite of products
- GIS Server
Open source web map servers

Several open source map servers are available for different operating system platforms, and they support the most popular geospatial file formats, databases, and standards. Some servers also include map viewers and an application programming interface (API) for the development of applications to integrate the map services with other layers. Owusu-Banahene and Coetzee [10] published an excellent evaluation of three open source map servers for setting up a geospatial thematic web service. The systems they evaluated include MapServer, GeoServer and QGIS Server. Their discussion addresses criteria such as ease of installation and cartographic workflow, level of support for displaying web maps, output formats supported, level of quality of web maps, types of data that can be served, level of support for styling, adherence to OGC standards and quality of the user documentation. An account of QGIS Map Server, MapServer and GeoServer was also published by Lienert et. al. [6]. Furthermore, the OneGeology initiative provides an excellent downloadable cookbook for the implementation of Web Map Services (WMS) using ArcGIS Server and MapServer with links to downloadable data for testing the implementation [15].

Web map services

The essential components for the serving of spatial data is the availability of an on-line high-quality base map service in the form of a street map and/or imagery that provides a geographic frame of reference for orientation and navigation and operational layers containing the data of interest. Base maps and operational layers often require separate strategies for effective maintenance and display in web maps [11] and it is good practice to separate these.

- **Base maps**

  Many of the available application programming interfaces (API) available today provide a base map as well as supporting imagery of reasonable resolution e.g. ArcGIS Javascript API, Google Maps API and Streetmaps. This obviates the need for developers and cartographers to spend valuable time to compile accurate and up to date presentable maps.

  The use of these maps is usually free when they are not applied for commercial purposes. The one draw-back of some on-line maps is that they are not regularly updated to keep pace with the rapid process of name changes in South Africa. Even though there are base maps made available by commercial organisations such as Esri and Google, there exists a need for an official national map service in South Africa.

- **Operational layers**

  In addition to the base map there are the actual map layers that constitute the essence of the data being served. The advantage of developing web map services lies in the fact that the data provided is authoritative and up-to-date so whenever changes are made to the data they are immediately available.

Application programming interfaces (API)

Commercial software offers the functionality to integrate web map services with the user’s data, but more often a web based application programming interface (API) is required that offers the functionality to overlay the base map with the organisation’s operational (or thematic) map service layers or layers from other sources in order to display it in a web environment. Controls are also required to navigate around the map, zoom in or out, query, changing the layer display sequence and adjusting layer transparency or generating a legend or scale bar. It may even be a requirement to overlay ASCII-based geographic point data from a file on the user’s computer onto the base map. The use of pure DOM and HTML elements such as drop-down menus may be required to facilitate rapid navigation to a predefined listed location such as a feature or geographical boundary or to provide a link to peripheral sources of information. These elements may also be used to generate on-line reports from data in a database.

The scope of the present paper prohibits an extensive discussion of all technologies available and only some of the more generally used interfaces are discussed below. Lienert et. al. [6] discuss additional technologies such as Adobe Flash, Scalable Vector Graphics (SVG), JavaFX, Canvas, and WebGL.

ArcGIS API for Javascript

The ArcGIS API for Javascript is an API provided by Esri, the developers and distributors of ACGIS software. The ArcGIS Resource Center offers a host of resources in the way of blogs, example code, help files, forums, videos, a knowledge base, and online maps and imagery. The author found that a drawback of some of the Esri online streetmaps is that the South African place names have not recently been updated to reflect the current name changes of many streets and towns which seem to change at a pace that is difficult to match. Much use is made of the DoJo toolkit and
the JSON specification in the ArcGIS API. The former is an open source Javascript modular library for the web
development environment. JSON (JavaScript Object Notation) was developed in 2005 for data exchange and is now
more widely used than XML containers that tend to have a large file size and in view of the time-consuming parsing of
the XML structure.

Google Maps Javascript API

Version 3 is the current version of the Google Maps API, with version 2 being deprecated in May 2010. The original
deprecation date has been extended from 19 May 2013 to 19 November 2013. As of this date, all applications
requesting version 2 will be served with a special, wrapped version of the version 3 API instead. It is expected that the
wrapped version of the API will work for most simple maps, but Google strongly encourages users to migrate their code
to version 3 of the API before this date [14].

The earlier version of Google Maps Javascript API was supported by Esri through the ArcGIS Javascript Extension for
Google Maps and allowed the use of ArcGIS Server services with the Google Maps API, however, Esri have decided to
no longer support the API at version 3. There is, however, unofficial support by Esri for the Google Maps API through
the use of ArcGIS ServerLink for Google Maps API version 3.

ArcGIS Flex Viewer

The Flex Viewer is available in both a compiled and un-compiled version, the former version is intended for users who
plan to build custom web applications through changes to the viewer’s XML files. The latter version allows for a more
customised approach by developers who wish to create custom widgets and who would like to extend the core viewer
application using the ArcGIS Viewer for Flex Application Builder. This is a WYSIWYG application that allows easy
creation and modification of the Flex Viewer applications in a GUI-driven approach.

Both these options use the ArcGIS API for Flex and can both provide a solution for creating customised GIS-enabled
web-mapping applications that do not require programming experience. The ArcGIS API for Flex in turn is built on the
Apache Flex framework. Functionality in the above applications is based on an extensible widget programming model.
Widgets are portable code blocks that provide functionality in a modular fashion; they can be easily added to or
removed as needed. Additional widgets for new custom functionality can be developed using ArcGIS API for Flex.

A distinction needs to be drawn between the ArcGIS Flex applications, ArcGIS Flex Viewer and the ArcGIS Viewer for
Flex Application Builder and the ArcGIS API for Flex. All these options are designed to work with ArcGIS Server and
ArcGIS Online web services.

Silverlight

The ArcGIS API for Silverlight enables the user to create rich internet and desktop applications that utilise the powerful
mapping, geocoding, and geoprocessing capabilities provided by ArcGIS Server and Bing services. The current version
includes a viewer, application builder and an extensibility kit. When the application builder is used no special
programming skills are required or any editing of configuration files. The extensibility kit allows for extension of the
core viewer and can utilise the extensive libraries available [15].

ArcGIS Server applications

ArcGIS Server allows the development of applications with GIS capabilities using the Web Application Developer
Framework for .Net or Java, depending on which installation of ArcGIS Server is on the users system. Both application
frameworks include server-side controls and client-side behaviour accessible via JavaScript libraries.

OpenLayers

OpenLayers is a GUI and a customisation tool for combining raster and vector data sources. It consists of a JavaScript
library for displaying map data in web browsers, without any server-side dependencies. It is often used in combination
with OpenStreetMap, a freely editable map of the world [6]. OpenLayers is useful for the embedding of web map
services (WMS) but other geometries cannot be easily integrated.

Streetmaps

With the free version of this API the user is entitled to full use of the mapping API, but is subject to a limit of 1000 hits
per month which includes geocode and map requests. For most APIs one needs to first sign up for an API key. This is
usually provided free of charge, subject to the terms and conditions which are clearly outlined on the website of the provider [13].

Most APIs work in conjunction with Javascript, and a thorough knowledge of this programming language is a prerequisite, but providers often provide clear examples to use the API.

An important aspect to consider is the provider’s right to allow advertising on the web map. Most providers are obliged to have copyright branding on the map in terms of agreements with their data partners, but they may also reserve the right to populate the map with points of interest from their advertisers, which show as small, usually unobtrusive logos on the map.

The author has no particular preference for any of the APIs, and it is not the intention of this paper to draw a comparison between the merits of the various APIs.

Best practices

In general, the quality of web map design and the measure of what constitutes a good web map can be measured only with difficulty [5]. A number of papers and web based articles on good web map design, and pitfalls that exist in the development of both web services and applications have been written. Some excellent accounts on best practices in web and map application design exist [3, 8], but there is to date no comprehensive compilation that addresses all the issues of both web application design and map services.

The author has for several years been involved in the development of web map services and web applications, mostly in the geological field. Based on this experience the discussion below addresses the more important issues to consider in developing web map services and designing applications. The present compilation by no means claims to be complete, but hopefully it will serve as a basis on which future compilations can build.

User requirements

A pre-requisite for the design of any map is to consider the purpose of the map. In the absence of a proper user requirements analysis the map designer has to pre-empt what the user of the map may require from the map which in turn will determine the map content. This aspect determines the usefulness of the map and may prove to be more difficult than the programming involved in compiling the map. Naegeli [8] lists a number of elements that are necessary to know about the intended users of a map viewer. These include:

- The type of equipment available to the user
- Their level of computer literacy
- Their educational or professional background
- The frequency of use of the application
- Their reason for using the application

With applications that serve data of a scientific nature to users in a corporate environment the answers to these questions may be surmised, and it may be safely assumed that the user’s computer proficiency will be adequate, however, when the user community is not well-defined the answers to the above list may not be so obvious and the process becomes more challenging. This applies especially where the users are expected to be less computer literate and their equipment to access the web services more antiquated.

It is not the purpose of this paper to discuss how the design of the web application should respond to each of the above user requirements, but rather to provide general guidelines to minimise their negative influence. Irrespective of the user community’s requirements, there are some basic principles that will apply to all users and which will contribute to the usefulness of the web application.

Map design and work flow

Four activities in the workflow for map design can be identified [3]:

- Designing the map information – with due consideration to completeness, currency, relevance and source.
- Designing the map – with consideration to the web interfaces’ ability to communicate the map’s message and its aesthetic appeal.
- Designing the user experience with attention to the interaction between the user and the map and its information
- Promoting the completed web map in order to maximise its value.
The use of unnecessary elements that have the effect of cluttering the page should be avoided. Many web pages are cluttered with unnecessary content that, while they may create a favourable visual impression, serve no purpose other than slowing down the download time of content and distract the user from the purpose of the web site. Also avoid features in the web application that may require downloads of software or plug-ins. If they are essential, then links should be provided to the download site.

*Look and feel*

The visual quality and functionality of the application interface determine the “feel” of an application, which describes the comfortable feeling of the user that he or she understands the controls and is able to obtain the desired results. The user should be left in little doubt about what the computer is doing [8].

Furthermore visual information should be provided that helps the user to understand the function of controls and warns not to use radio buttons and check boxes interchangeably [8]. It is also to be encouraged to use visual clues to distinguish icon buttons that invoke a different mode or state of the application and change its behaviour from buttons that execute a single narrowly defined function. It is also suggested to use icons that relate to familiar real life objects to make user interfaces more intuitive and memorable e.g. the recycle bin, but warns against the fact that these may be objectionable in some cultures [8].

Jenny et. al.[5] warn against the use of unfamiliar and complex tools to interact with the map and recommend the use of interface elements that are well known from other applications or web maps, such as underlined hyperlinks, buttons, sliders or other standardised interface elements. They propose the placing of such tools onto the map instead of on the periphery e.g. having a magnifying glass tool on the map for zooming and placing drawn rectangles on areas of interest for which more detailed maps can be loaded.

*Separation of structure and design*

An important aspect in the development of efficient web pages is the principle of separating structure and design.

- **Structure**

This comprises the text, images, tables, map viewers and other visual or audio elements that make up the content of a web page. The HTML specification describes the structure of these elements.

- **Presentation**

This is the way in which the above elements are displayed and is described by other specifications of which Cascading Style Sheets (CSS) is one of the most important. The structural elements of the web page are identified by an `<id>`- or a `<name>`-element which is referred to in the CSS-specification which can be used inline or better in a separate file and refers to a single page or multiple pages constituting the entire web site. This obviates the need to specify the structure for each separate element, and provides a means to change the appearance of an element through a single parameter in one place. It also constitutes a time saving measure in that the CSS file is loaded and cached once after which it can be applied to the entire web site. Style sheets should also be linked to the web document which results in it being downloaded once, and then used for the rest of the site without being reloaded. All the style specifications should preferably be loaded into a single stylesheet rather than in separate files.

*Page layout*

There are two ways to plan the layout of a web page’s width i.e. fixed or liquid.

- **Fixed width web pages**

This is the more difficult and more rigid approach to design a web page layout where the web page elements are allocated fixed dimensions and positions. The developer needs to know what the lowest common screen resolution is for the intended audience.

- **Flexible web pages**

This approach is also known as liquid design which can adjust to the size of the browser window and different screen resolutions. For a flexible design the dimensions and position of the web elements and containers of the web page elements are allocated in percentages of the screen dimensions. A container can be a `<div>`-element or a table. All the
Page dimensions: 595.3x841.9

web page elements rearrange themselves to suit the browser window or user’s screen resolution if the design is done correctly. Just like a fixed width design, all the widths have to add up to the maximum width of the screen. A ruinous aspect of a flexible design is using images that are too wide.

Although frames have certain benefits like providing a constant position on a web page such as a navigation page or table of contents or for hiding a deep link in the site from users, the use of <Frameset>-elements in web design is generally discouraged and in HTML5 their use has become obsolete. The use of <iframe>- or <div>-elements are proposed instead. (http://webdesign.about.com/od/framesprosandcons/a/aaframesyuck.htm)

Screen resolution and size

Screen resolution is an important aspect of web page design and requires the developer to test the web page on computers with different screen resolutions. If the map is the essential content of a page, it should be allowed to occupy appropriate space, however, after the other elements of the web page have been allocated only limited space may remain for the map. An important factor to keep in mind in this respect is to specify the size of web elements in percentages rather than in absolute screen pixels. The map should be designed to allow for its size to be adjusted dynamically to the size and proportion of the available space. According to a survey the trend in screen resolution has been increasing rapidly since 2010 as shown in Table 1.

<table>
<thead>
<tr>
<th>Date</th>
<th>Higher</th>
<th>1024 x 768</th>
<th>800 x 600</th>
<th>640 x 480</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 2013</td>
<td>90%</td>
<td>9%</td>
<td>0,5%</td>
<td>0%</td>
<td>0,5%</td>
</tr>
<tr>
<td>January 2012</td>
<td>85%</td>
<td>13%</td>
<td>1%</td>
<td>0%</td>
<td>1%</td>
</tr>
<tr>
<td>January 2011</td>
<td>85%</td>
<td>14%</td>
<td>0%</td>
<td>0%</td>
<td>1%</td>
</tr>
<tr>
<td>January 2010</td>
<td>76%</td>
<td>20%</td>
<td>1%</td>
<td>0%</td>
<td>3%</td>
</tr>
</tbody>
</table>

Table 1: The trend of computer screen resolution between 2010 and 2013.
(Source: www.w3schools.com/browsers/browsers_display.asp)

While the old recommendation was to design for a web site at 1024 × 768, the new guideline is to optimise for widescreen monitors around 1440 pixels wide [9], which is borne out by a survey of the market share of some screens arranged according to screen resolution shown in Table 2.

<table>
<thead>
<tr>
<th>Screen resolution</th>
<th>Market share % (Jan 2010 to June 2013)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024 x 768</td>
<td>22,07</td>
</tr>
<tr>
<td>1366 x 768</td>
<td>15,82</td>
</tr>
<tr>
<td>2560 x 1440</td>
<td>0,28</td>
</tr>
<tr>
<td>1920 x 1200</td>
<td>1,11</td>
</tr>
<tr>
<td>1920 x 1080</td>
<td>4,19</td>
</tr>
<tr>
<td>1680 x 1050</td>
<td>3,8</td>
</tr>
<tr>
<td>1600 x 900</td>
<td>3,24</td>
</tr>
<tr>
<td>1440 x 900</td>
<td>6,92</td>
</tr>
<tr>
<td>1360 x 768</td>
<td>2,16</td>
</tr>
<tr>
<td>1280 x 800</td>
<td>14,3</td>
</tr>
<tr>
<td>1280 x 1024</td>
<td>8,01</td>
</tr>
<tr>
<td>1024 x 600</td>
<td>2,02</td>
</tr>
<tr>
<td>800 x 600</td>
<td>1,39</td>
</tr>
<tr>
<td>Other</td>
<td>6,61</td>
</tr>
</tbody>
</table>

Table 2: Results of a survey of the market share of some screens arranged according to screen resolution.
(Source: http://gs.statcounter.com)
Influence of the web browser on the application design

It is essential for the developer to note that browsers behave differently to programming code due to different rendering algorithms, rendering and scripting capabilities, font availability and a varying degree of adherence to official web standards [5].

It is, therefore, useful for the developer to be informed on the frequency of use of different web-browsers and plug-ins and to evaluate the graphical appearance and interactive behaviour of the map with these browsers.

The popularity of browsers according to a survey [19] is shown in Table 3.

<table>
<thead>
<tr>
<th>Date</th>
<th>Chrome</th>
<th>IE</th>
<th>Firefox</th>
<th>Safari</th>
<th>Opera</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jun-12</td>
<td>32,76</td>
<td>32,31</td>
<td>24,56</td>
<td>7</td>
<td>1,77</td>
<td>1,61</td>
</tr>
<tr>
<td>Jul-12</td>
<td>33,81</td>
<td>32,04</td>
<td>23,73</td>
<td>7,12</td>
<td>1,72</td>
<td>1,58</td>
</tr>
<tr>
<td>Aug-12</td>
<td>33,59</td>
<td>32,85</td>
<td>22,85</td>
<td>7,39</td>
<td>1,63</td>
<td>1,69</td>
</tr>
<tr>
<td>Sep-12</td>
<td>34,21</td>
<td>32,7</td>
<td>22,4</td>
<td>7,7</td>
<td>1,61</td>
<td>1,38</td>
</tr>
<tr>
<td>Oct-12</td>
<td>34,77</td>
<td>32,08</td>
<td>22,32</td>
<td>7,81</td>
<td>1,63</td>
<td>1,39</td>
</tr>
<tr>
<td>Nov-12</td>
<td>35,72</td>
<td>31,23</td>
<td>22,37</td>
<td>7,83</td>
<td>1,39</td>
<td>1,46</td>
</tr>
<tr>
<td>Dec-12</td>
<td>36,42</td>
<td>30,78</td>
<td>21,89</td>
<td>7,92</td>
<td>1,26</td>
<td>1,72</td>
</tr>
<tr>
<td>Jan-13</td>
<td>36,52</td>
<td>30,71</td>
<td>21,42</td>
<td>8,29</td>
<td>1,19</td>
<td>1,87</td>
</tr>
<tr>
<td>Feb-13</td>
<td>37,09</td>
<td>29,82</td>
<td>21,34</td>
<td>8,6</td>
<td>1,22</td>
<td>1,93</td>
</tr>
<tr>
<td>Mar-13</td>
<td>38,07</td>
<td>29,3</td>
<td>20,87</td>
<td>8,5</td>
<td>1,17</td>
<td>2,09</td>
</tr>
<tr>
<td>Apr-13</td>
<td>39,15</td>
<td>29,71</td>
<td>20,06</td>
<td>8</td>
<td>1,01</td>
<td>2,08</td>
</tr>
<tr>
<td>May-13</td>
<td>41,38</td>
<td>27,72</td>
<td>19,76</td>
<td>7,96</td>
<td>1</td>
<td>2,18</td>
</tr>
</tbody>
</table>

Table 3: Comparison of the popularity of browsers. (Source: [http://gs.statcounter.com](http://gs.statcounter.com))

• **Compatibility view**

It may occasionally happen that a web site does not appear as it should in the browser with the divisions becoming jumbled or images not displaying correctly or at all. This may be due to incompatibility issues between the web site and the browser. Internet Explorer from version 9 on has a compatibility option in the form of a button on the address bar that can be toggled to allow the site to be viewed in compatibility view mode. In Internet Explorer 10 this feature may activated through the Tools option on the Menu or Command bars.

• **Cache**

Very often a map service may not function correctly due to cache issues and this may be rectified by clearing the cache on the internet browser. The user should be alerted to this possibility and guided as to the correct procedure to go about.

**Speed of download and page size**

Internet bandwidth no longer constitutes the limiting factor in the web streaming of data which was the case ten years ago, and even home users have access to bandwidths in excess of 500 kbps at affordable cost. However, within organisations where large bandwidth is mostly needed this is still a limiting factor since the bandwidth needs to be shared by numerous users, many of whom have other objectives than scientific pursuits.

The time allowed for download of a web page should be in the order of 1 to 2 seconds, which at an internet speed of 28 kbps means the page size should not exceed around 50-60 kB. To achieve this the text in a web page should be kept to a minimum and unnecessary functions in Javascript- or CSS files of an application that govern the functionality and appearance should be avoided. When static images are used these should be compressed. The user should always be able to see the status of the application and of a task, what is expected, or what has gone wrong and how it can be corrected. If the application has multiple modes, it should be ensured that the current mode is indicated conspicuously.
Visual as well as functional controls that are unusable in the current context should be deactivated. User input should be acknowledged and the progress of tasks that take more than a couple of seconds should be shown. The user should never be left wondering whether the input was accepted, the last click made any change, or an operation has completed [8]. The use of an indicator that the page is loading or that a background process is in progress contributes to patience on the side of the user.

**Multiple versus single map layers**

The use of a series of ready-made static maps is recommended rather than requiring users to manipulate multiple layers by themselves [8]. The use of multiple map layers in logical groups such as base map features and operational layers which can be published as separate map services and for which separate display strategies can be formulated is recommended [11].

The following process of selecting layers is proposed [16]:

- Meet with project staff to determine:
  - Essential data layers
  - Possible additional data layers
  - Data display order
- This is an iterative process
- Avoid the temptation to include layers that may possibly be required
- The overall goal of the application should be to guide web visitors to data quickly and logically

**Dynamic map layers**

This is the more general approach to displaying map layers. The server draws the map at the extent specified by the user and has the advantage that they may be switched on or off by the user. Furthermore the minimum and maximum scales at which the layer displays may be specified at the design stage.

The principle of progressive downloading is promoted [12] which hinges on the idea of providing the user with some web content while the rest of the page is loading in the background. This has the effect of holding the user’s interest. This especially applies when tables are used since they only appear on the browser once the entire table has been downloaded. It entails splitting up the page into many smaller tables as opposed to using one large table.

**Caching**

In order to provide the rapid display of map layers where the data does not change frequently it is good practice to publish map services as cached layers which consist of a pre-generated set of map images in tiles that the server can distribute to clients. As the user zooms into the map layer more detail in the data becomes visible. Base maps and their accompanying map caches are generally easy to maintain and when the data needs to be changed, the cache can be updated. Caching strategies should be designed to cache only areas of interest and, thereby, avoiding caching areas of little interest such as the ocean or vast extents containing no data.

Some software systems provide the tools that allow the use of the boundaries of feature classes to define the area that needs to be cached [11]. Caches can also be created on demand which obviates to pre-cache the entire map layer, however, this incurs a performance penalty to the user.

**Use of map elements**

The following is a list of elements for performing basic tasks that could be incorporated in a web application, but not all need to be present.

- Support for various map projections: The projection is determined by the other map layers that may be used in conjunction with the map is or to a large extent by the nationally accepted projections
• Geographic extent – this is largely determined by the purpose of the map i.e. whether the data pertaining to the map is provincial, national, continental or global
• Support for dynamic and cached (tiled) map services
• Graphics – allowing users to draw graphics
• Pop-up windows and map or tool tips when users click the mouse or hover the pointer
• Info window
• Querying and identifying features
• Attribute table
• Locating addresses
• Finding attributes
• Generating renderers and providing pre-designed templates
• Printing features
• Geoprocessing- e.g. for buffering
• Scale bar
• Legend
• Icon indicating progress
• Transparency slider

Accessibility

While considering the use of the above map elements, the developer should also take cognisance of the issue of accessibility to users with some impairment such as visual or a physical handicap when designing an application. This may imply the implementation of technologies that can compromise download speed. In this respect Jennyet. al. [5] propose that “Web technology can support accessible map design: the user should be able to enlarge maps with very detailed information; vector-based symbols should change dynamically at the user’s request; the user should be able to increase the size of text; and colour schemes should be exchangeable according to the needs of the visually impaired. All of these aids should be activated with as little user interaction as possible, preferably with a single button click.”

Fonts

Excellent accounts on the optimal use of fonts have been written [2,3,4,5].

• Fonts should show clear contrast on coloured backgrounds [3].
• Symbols and text must be large enough to enhance legibility. A rule of thumb for text size is that text should be 10 pixels high amounting to font size of 7 points or larger on a PC and 9 points on an Apple Macintosh [3].
• Serifs are the finishing strokes at the bottom and top of certain typefaces that theoretically aid the flow of the eye and enhance reading [3] and should be avoided in text.
• Anti-aliasing is a technique to improve the graphical appearance of digital imagery by smoothing jagged edges of map elements. Along borders with visually high contrast, intermediate colours are assigned to the pixels creating a blurry image. Anti-aliasing is applied when converting vector objects to a raster image for display on the screen [5]. Most graphic software allows the implementation of anti-aliasing techniques, and GIS software also allow for anti-aliasing of map graphics.

The fonts best suitable to use for web pages are sans serif with open shapes, wide punch, wide letter spacing and tall x-height (Fig. 1). The recommended font types for web pages conforming to these characteristics are: Verdana, Lucida Grande, Frutiger, Stone Sans ITC, Cisalpin and Myriad [5]. To this list can be added Arial, Helvetica, Georgia, Trebuchet, Century Gothic and Palatino [3]. It is, however, important to consider whether the font to be used is installed on the user’s computer to render the map.

Fig. 1: Web font characteristics [4].
Image types

In the discussion of image types a distinction needs to be made between the map image itself and supporting images in the web page. With regard to the latter a good comparison of different image compression formats is given by Aguilera [1]. In general the following guidelines apply:

- **BMP** is an uncompressed format, but with large size.
- **PNG** was originally designed for internet based imagery and not for professional graphics. It employs lossless compression, and is generally suitable for even-coloured images, but not for images in which there is great colour variation. It is in general better suited for images containing text or line art such as scanned documents or drawings. Different formats of PNG are available and a distinction needs to be made between these.
- Regarding the map image itself, it is recommended to use PNG 8 image format for the caching of any layer that will be displayed in the map. PNG 24 is not a good choice because Internet Explorer version 6, and earlier have limitations with displaying transparency in this image format.
- **TIFF** is a lossless compression format, but characterised by large image sizes and generally not suitable for internet applications.
- **JPEG** is a lossy compression format and has generally small image size, but the loss of image quality is proportional to the compression ratio. JPEG is the most commonly used format for storing and transmitting images in Internet Explorer. JPEG does not support transparency and is, therefore, not a good choice for map images.

Querying and client-side graphics

Quinn et. al. [11] advocate the use of client-side graphics to display the results of features without necessarily displaying the entire feature layer at once or at all. By clicking on the map to query a point or area, the area of interest shows up as a client-side graphic. This approach has the following advantages:

- Only the features that are needed are loaded.
- The load on the server CPU can be lightened due to the fact that advantage is taken of the CPUs available on the users’ machine.
- When edits to the data have been made, they will show up right away the next time the user navigates the map.
- Display options and symbology can be easily exposed to users through the use of JavaScript.

With regard to querying a map the number of records returned may be huge and may slow down performance on the user’s side. Some software systems allow the limit on the number of records returned upon a query to be specified. This applies especially where high resolution features such as cadastral boundaries are returned.

Editing

Finally it is recommended that for the development of a web application use is made of an editing environment such as Aptana Studio to facilitate the editing of programming code, and products such as Fiddler or Firebug for the debugging and testing of the behaviour of web processes by recording the traffic between the web and the user’s computer.

Examples

The paper is concluded with some examples of good and useful web applications in the South African and international context.

OneGeology International Project

The OneGeology project has the objective of making global digital geological data available on the internet at a scale of 1:1-million or better. This data could be in the form of either vector or raster data as provided by the countries. No initial harmonisation of data is required at this stage of the project.

At the outset of the project Open Geospatial Consortium (OGC) Web Mapping Service (WMS 1.1.1) standards were adopted for data serving. It was also decided that the preferred open source software for data serving would be Mapserver, however, this is not prescriptive and participants are at liberty to use any WMS compliant map service.
However, a pre-configured Mapserver package was made available to prospective users who do not already have WMS compliant map services.

The OneGeology map data are served from a centralised portal (Fig. 2) maintained and managed by the Bureau de Recherches Géologiques et Minières (BRGM) in France. The portal offers an interactive global map with navigational and querying tools. Drop-down lists are available where the country of interest can be selected and the specific layers required (if available).

![Interactive global map on the OneGeology portal.](www.onegeology.org)

The OneGeology project makes provision for countries that do not have the technological know-how to be assisted by other countries that do have the technological infrastructure through the so-called "Buddy" system. This assistance can be either through technological assistance or by serving that country's data on their behalf.

At this point in time the Council for Geoscience (CGS) in South Africa is already serving the 1:1-million scale geological data of South Africa, including Lesotho and Swaziland as well as the geological map of the SADC countries at a scale of 1:2,5-million and the national 1:1-million scale geological data sets of both Botswana and Mozambique at their request.

**Kentucky Geologic Map Information Service**

The Kentucky geological map information service [17] is a good example of a focused map service that provides access to a vast amount of geological and related information. The page layout appears complex, but the navigation tools are well explained and simple to use. Drop-down menus provide quick access to supplemental information and user tutorials are downloadable. Features such as layer transparency, pre-defined layouts, querying tools and different base maps are some of the attractive features of this service.

**Namibia Atlas**

This web site, though not a web map service in the strict sense, was developed by the Acacia Project E1 in Germany. The site presents a simple clean and clutter-free example of good web page design and clarity of purpose. Digital data can be downloaded from the site in Portable Map File (PMF), shape file and JPEG format as well as metadata.

**British Geological Survey (BGS)**

This is one of the most advanced web map sites in the opinion of the author with geological web map services available in a host of viewers for different themes, including a mobile map service.

**South African National Biodiversity Institute (SANBI)**

The South African National Biodiversity Institute’s web map service provides access to the abovementioned vegetation map of Mucina and Rutherford [7]. The layers available include the national land cover, vegetation, biomes and soils. Additional layers include the national parks, national protected areas, free flowing rivers, water management areas, catchment areas, forest patches, wetlands, fish sanctuaries, and ecosystems in addition to the layers for geographic orientation.
Department of Water Affairs

The Directorate: Spatial and Land Information Management offers the following map services to view:

- Water Management Areas (including drainage areas)
- Vaal River flow lines
- RDM Surface water reserve status (Dec 2010)
- RDM Ground water reserve status (Dec 2010)
- RDM Ground water: General authorisations

In addition there are data downloads for the primary, secondary, tertiary and quaternary catchment boundaries and the water management areas as at 2004. Even on computers with access to broadband internet this web site was found to be extremely slow.

**Dam contours**

This map service (www.damcontours.co.za/default.asp), powered by StreetMaps shows the elevation contours of a number of the major dams in South Africa. This is, in the opinion of the author, an excellent example of a web map service that shows simplicity in design and clarity of purpose. The audience is well defined as being fishermen who want to pre-plan a fishing trip, but the information has other applications. The map displays rapidly and the navigation controls are simple and easy to understand. Reasonable resolution supporting imagery is one of the display options.

**Conclusion**

The subject of web map services is a vast topic and resources outlining good design are legion. In this paper the author has made an attempt to synthesise the salient aspects of serving and developing web map services, but it remains an almost impossible endeavour to present all the aspects within the constraints of a single paper, and all that can be hoped for is that this paper will encourage further discussion on this vast topic.

**References**


[13] www.streetmaps.co.za


Contact Dr. Hermanus J Brynard, Geological/GIS Consultant, Tel 082 820-9499, mbrynard47@gmail.com