UCL

WORKING PAPERS SERIES

Paper 153 - Feb 10

Developing Efficient Web-based GIS Applications

Developing Efficient Web-based GIS Applications

Adnan, M., Singleton, A.D., and Longley, P.A.

Department of Geography and Centre for Advanced Spatial Analysis
University College London, Gower Street, London, WC1E 6BT.

Tel: +44 (0)20 7679 0510 Fax: +44 (0)20 7679 0565
Email m.adnan@ucl.ac.uk, a.singleton@ucl.ac.uk, plongley@geog.ucl.ac.uk

February 2010

ABSTRACT

There is an increase in the number of web-based GIS applications over the recent years. This paper describes different mapping technologies, database standards, and web application development standards that are relevant to the development of web-based GIS applications. Different mapping technologies for displaying geo-referenced data are available and can be used in different situations. This paper also explains why Oracle is the system of choice for geospatial applications that need to handle large amounts of data. Wireframing and design patterns have been shown to be useful in making GIS web applications efficient, scalable and usable, and should be an important part of every web-based GIS application. A range of different development technologies are available, and their use in different operating environments has been discussed here in some detail.

1. INTRODUCTION

Recent years have seen an enormous increase in the number of web-based applications that use techniques derived from geographic information systems (GIS). Thus, for example, Haklay et al. (2008) describe how, by mid-2005, Multimap had attracted 7.3 million visitors in UK and Mapquest was used by 47 million in US; and how, by mid-2007, there were an estimated 50,000 websites that took feeds from the Google Maps (http://maps.google.com) site. web-based GIS applications can be as simple as presenting a simple map of the world to front-ending complex spatial analyses of spatial distributions and processes. GIS-enabled web applications can utilise any of a number of technologies and database platforms, the choice of which can impact on the performance of the web application. MS-Access is not the software of choice for a GIS application built around millions of records because the software has not been developed to handle very large amounts of geographically referenced data efficiently. An important consideration in software selection is speed of response as users may expect the result of GIS analysis to be produced in real time. These requirements make the choice of mapping, database, and development technologies and standards very important. This paper provides GIS application developers and programmers with insights into the mapping, database, and development technologies that are being used to develop high volume web-based GIS applications. This further provides an assessment of between different technologies and their usefulness in different scenarios in order to perform in a range of circumstances.

1. WEB-BASED GIS MAPPING TECHNOLOGIES

An important part of every web-based GIS application is its mapping or visualisation technology, which makes it possible to show data in the form of maps. Visualisation of data as maps has become increasingly popular, with hundreds of websites presenting geographic data. The popularity of web-based mapping applications arises in large part through the wide dissemination of software that makes it easy for users and developers to publish map data. Improvements in usability through improved user interfaces also account for the increased popularity of visualisation techniques (Aoidh et al. 2008). In similar vein, Cashera et al. (2008) have demonstrated that growing interest in visualisation and analysis of social networks has led to the development of several methods of structural analysis in order to analyse individual and group behaviour. This visualisation is not limited to the display of raw data in maps but is increasingly widely applied in the representation of large spatial databases. One such example is the Worldnames website (http://www.publicprofiler.org/worldnames), which accesses a database of 1 billion geo-referenced names around 26 different countries of the world. This website generates flash maps to show the concentration of surnames around different countries of the world. And this explains the extent of technology which is used now a days in web-based GIS applications.

There are three ways in which visualisation functionality is implemented; static map renderers, slippy maps, and flash mapping. The following is a discussion of each of these techniques with discussion of their use in different scenarios.

1.1. STATIC MAP RENDERERS

Static map renderers have been heavily used in the creation of maps. They render maps in the form of images. They work by rendering the map to an image (such as a PNG image), and then send the image

back to the user who can view it using a web or desktop application. Static map renderers do not allow any user interaction with the maps because they are just static images, so they do not allow users to pan or zoom in, out or around the map. On the one hand, the resulting map is very simple and easy to interpret, because there is nothing is required on the client side, but on the other hand the maps produced are very in flexible. Because, a user cannot interact with the maps they are not suitable for rich Internet applications, or for the new Web 2.0 applications.

Figure 1 shows an example of static map renderer, with a map produced by a UK National Trust website that allows users to view the spatial distribution of their family names (http://www.nationaltrustnames.org.uk).
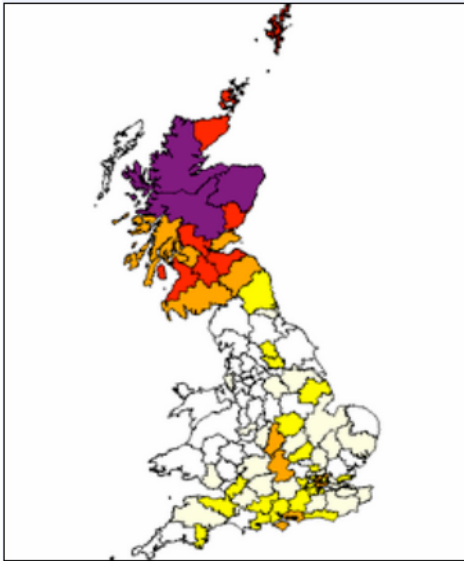

Figure 1: A map produced by a static map renderer

Static map renderers are fast to compute but they do not present a flexible solution for representing maps because of their non-interactive nature. This is, however, offered by 'slippy' or tile-based maps.

1.2 SLIPPY (TILE-BASED) MAPS

One of the important and widely used geo-visualisation techniques is tile-based or slippy maps. This technique works by dividing the map into a discrete number of zoom levels, so that each zoom level has an identical number of tiles. Instead of accessing the whole map at once, a tile-based client builds the map by accessing individual tiles and then assembling them to form a map. Google Maps, Microsoft Virtual Earth, Yahoo Maps, Mapnik, Open Layers, and ArcGiS as well as derivative products such as MapTube are based on tile based or slippy maps. Slippy maps were introduced as an alternative to static maps because they are dynamic and faster to load and the tiles of specific parts of the map only load on user requests. Figure 2 shows the division of tiles into a discrete number of small tiles.
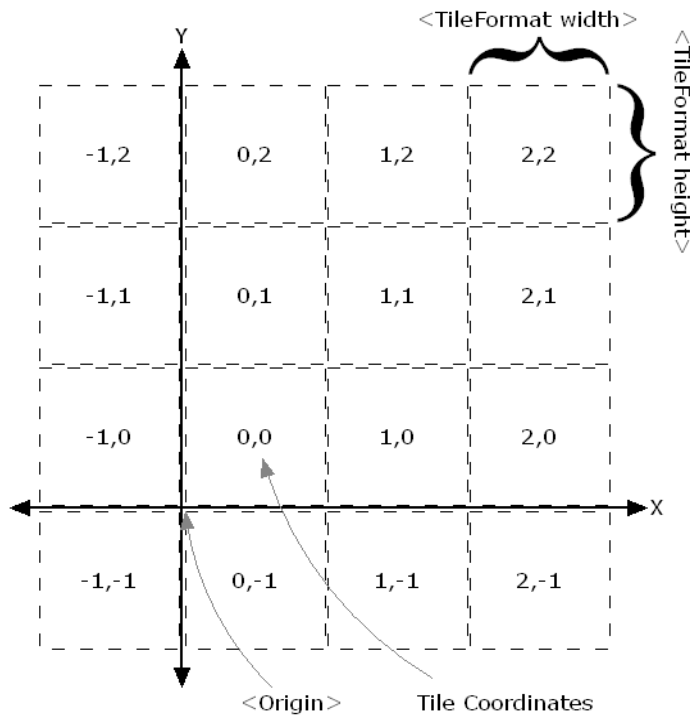
Figure 2: Tiling in slippy maps

The tiling process works by dividing the whole map into small equally sized tiles. The coordinates identified when a user clicks to view the finest level of detail acts as the origin of the tiling divisions and all the tiles are numbered as shown in the above figure. The map display is centred upon the user's 'origin tile'. This process of dividing a map into further tiles proceeds recursively if user clicks on the map again.

1.3 FLASH MAPPING

Flash maps are based on vector images. Busselle (2008) describes how a raster image is a rectangular array of elements called pixels, and a vector image is a collection of connected lines and curves that can be used to delineate objects. Raster images are resolution dependent, so their shape distorts if the image is made bigger than its actual resolution. Vector images can be thought of as having an advantage than raster images in that, in the absence of point weeding or any other line generalisation technique, they are resolution independent. Thus their shape does not distort if the granularity of the image changes. This is because a vector image is drawn dynamically by using mathematical calculations which do not allow the map to become distorted and keeps it in the right shape.

Flash (http://www.adobe.com/products/flash/) has become popular as a mapping solution during the last few years. It has been incorporated as an API into ArcGIS (ArcGIS Flex API) and it is now also possible to integrate Flash with the Google Maps API. This has made it possible to create flash mash ups on top of Google Maps. The increasing popularity of Flash arises because of it makes possible to develop rich internet applications, and it is easy to integrate flash with other technologies and thus making it possible the use of Flash as a mapping solution in a range of applications.

The Google Maps API for Flash and the Flex API for ArcGIS Server allow users to use Flex to embed the functionality of Google Maps or ArcGIS maps into web applications. Both APIs manage complex

4

details of connectivity using the GIS package and they provide an easy to use interface for developers in order that Flash can be used to develop more interactive and rich Internet applications by users who do not need to understand Flash Action Scripts in much detail. This opens up a new way of developing GIS applications whereby the GIS server produces the map and then Flash is used to add some more information to it. This information might entail adding a point, line or polygon, or equally might entail adding some more complex graphics to the map, e.g. custom navigation Flash controls.

2. CHOOSING AN APPROPRIATE MAPPING TECHNOLOGY

The choice of an appropriate mapping technology is based on the requirement of the time it takes for the user to submit a query and to receive the results. Static map renderers are an old style mapping technology, which produces map in the form of an image before display of the image by the web application. Static map renderers are nonetheless useful in situations where the number of geographically referenced areas is moderate. This can be described as any number of geographically referenced areas less than 1000. Because each user request generates a new image, the application becomes slow if there are a great number of geographically referenced areas to be displayed in the image produced. However, static map renderers are fast for small number of geographically referenced areas, although they are not a flexible solution for representing maps because of the non-interactive nature of the final maps produced.

Today, the tile based or slippy mapping technique is the most widely used mapping technique. Slippy maps are better than static maps because tiles are served to users based on requests rather than serving the whole map at once. Slippy maps are dynamic and faster to load because the tiles that a user wants to see only load up and it is not necessary to load all the tiles at once. This saves computational processing power and memory resources, both on the client and the server side. In addition, it provides a powerful architecture for the creation of rich Internet GIS web applications. Rich Internet GIS web applications are defined as defined as web-based applications having characteristics of desktop applications and are delivered by way of a standard web browser. However, slippy maps do not provide a good solution for the development of on the fly rendering applications. On the fly rendering requires a lot of memory and computational resources and this makes the slippy maps solution inflexible to overlay data on top of all the tiles. Taking the example of mapping surname data on top of Google Maps, if a user searches for a name 'Longley' and this needs to be visualised on 3 zoom levels of geography (Country, Region, and County), this requires the need to overlay the data for all levels on the tiles of Google Maps. It is fine for one level, but to do it for all the tiles of all zoom levels is time consuming. Data could be overlaid on Google Maps in form of KML as well, but KML files have restriction on the size of the data they can store. Thus in an environment where data change dynamically based on user requests and on-the-fly rendering is required, slippy maps are not a suitable option. One option, as described by Gibin et al. (2008), is based on pre-rendering the tiles using a tool called GMAPCreator, storing the pre-rendered tiles on disk, and serving the tiles to users from disk rather than rendering them on the fly. However, while this technique solves the problem of computation time, it presents another problem of storing the tiles on disk. There could be millions of tiles and so considerable disk space is required to store all the tiles, which is not efficient.

Flash mapping is good for producing maps that are rendered on-the-fly. Flash files are easy to load in the browser and they do not need to be stored as pre-rendered files in anticipation of any user request. Thus once a user sends a request, the only function of the software is to paint the objects (for example,

the polygons that make up a choropleth map) of the flash file with a particular colour value, which makes it really fast. However, the user's machine must have flash player installed on it.

Static map renderers, slippy or tile based maps, and flash mapping are the three possible mapping solutions, and their deployment in appropriate situations can enhance the performance of the web-based GIS application.

3. DATABASE STANDARDS

The database is an important part of web applications. In the case of GIS web-based applications, it becomes more important because of the storage requirements of geographically referenced data. Once the geographically referenced data are in a database, they can be used and displayed in the form of maps by the web-based application. Thus the performance of a database has a direct impact upon the performance of the web application. There are a number of different database management systems available; in addition there are a number of database tuning techniques available to enhance the performance of a database. This section covers these techniques in detail.

3.1 CHOOSING AN APPROPRIATE DBMS (DATABASE MANAGEMENT SYSTEM)

There are many database systems which can store the data in their data structures in the form of database tables. The two most popular relational database management systems are Oracle (http://www.oracle.com) and SQL Server (http://www.microsoft.com/sqlserver). These solutions each have their own characteristic properties regarding the storage, management, retrieval, and display of data. Other relational database management systems include MySQL (http://www.mysql.com) and MS-Access (http://office.microsoft.com/access), although these database management systems are unsuited to handling very large volumes of spatial data, and so Oracle and SQL Server are the two database management systems of choioce for storing large amount of geographically referenced data.

Oracle (2005) compares Oracle and SQL Server and suggests that Oracle outperforms SQL Server in performance trials based on its concurrency model, involving indexing, partitioning, and parallel execution of processes. In addition, Oracle claims better security of the database as it encrypts and stores the data using better encryption algorithms and thus makes it possible for the maintenance of confidentiality of spatial records and protects the web application from 'scraping' of large amount of data. This makes Oracle a better database management system for use in web-based applications of GIS. Oracle, currently in its 11g release, offers different versions of its DBMS which differs in the functionality. Oracle 10g was launched for grid computing, Oracle 11i is an e-business suite, and Oracle 11g has the support of grid computing and it is also an e-business suite.

3.2 DATABASE ISSUES

Database performance has a major impact upon the performance of the web application. Database performance indicates the response time of the database management system for retrieving/storing records, it also represents how efficiently database handles multiple read/write requests to different database tables. Because GIS web-based applications are normally required to serve large amounts of geographically referenced data from a database, it is important that the database perform efficiently in response to different user requests. A number of tuning procedures have been devised for the enhancement of database performance. Appropriate use of normalisation, stored procedures, and indexes can improve the performance of web-based GIS applications. These are discussed in turn here.

A) NORMALISATION

Fotache (2006) has illustrated how a relational database can take the form of a single huge relation with thousands of records, although any such relation may create data redundancy and accommodate a lot of insert, update, and delete anomalies. Normalisation is a technique used to reduce data redundancy while maintaining integrity of the data in the database. When data are loaded into the database it is possible that there are many redundant fields within a database table. Normalisation works by dividing database table of redundant fields into smaller tables by maintaining the integrity of data. There are four forms by which data can be Normalised: first, second and third Normal form, and the Boyce-Codd Normal Form (BCNF).

The Normalisation procedure starts by converting the raw data to the first Normal form, after which the data are checked for redundancy. If data are still redundant, then the process of Normalisation continues to the next level and further checks for redundancy and integrity are performed. It is conventional to assume that the third Normal form reduces data redundancy of the data to a sufficient extent, whilst also maintaining the integrity of the data. However, the BCNF procedure may be used by specialist database programmers to further reduce the redundancy of data.

Normalisation reduces data redundancy by dividing a single database table into multiple tables and retaining unique field values across different tables. However, successful use of Normalisation in web-based GIS applications depends on the volume of data which any given application has to handle. Where the amount of data that the application has to handle is small, then Normalisation can improve the performance of the web application by removing redundant records and thus makes searching of data fast. However, Fotache (2006) has described how, for large databases (i.e. where there are millions of records), the web application may work better without Normalisation. The reason for this is that Normalisation separates the data into multiple tables, thereby increasing the number of joins between the tables that must be resolved by SQL queries when accessing the data. Retrieving data from a query containing joins between tables takes more time than retrieving the data from a single table.

B) STORED PROCEDURES

Stored Procedures are similar to normal PL/SQL procedures except that they are stored in database dictionary as database objects. Stored procedures can enhance the speed of data retrieval drastically. Microsoft (2009) have described how stored procedures as similar to procedures that can access input parameters and can return a value back to the calling program. They are stored in the data base dictionary.

Stored procedures offer a number of important benefits over queries. Because stored procedures are stored physically in the database dictionary, a database management system creates the parse plan for their execution when a procedure is called for the first time. Each subsequent calls use the same parse plan and thus reduce the overhead of the database management system. Because they are stored physically in the database dictionary, the DBMS compiles them when they are called for the first time. On each subsequent call, the DBMS uses the old parse plan to delegate requests to stored procedures. This improves the speed of access to the database and enhances the performance of the web application using stored procedures.

Stored procedures also make it possible to reduce network traffic. In a three-tier web application it is possible that the database is located remotely from the web application. Each request from the web application to the database server consumes network bandwidth. When thousands of users are

accessing the web application simultaneously, effective management of network traffic becomes crucial. If a web application has to send queries to the database server following each individual request, this consumes memory and network bandwidth space. One solution is to use a stored procedure which encapsulates the queries that fulfil a particular web request, thereby obviating the need for the web application to send queries in response to each request. This makes it possible to reduce network traffic and also improves the performance of the web application.

C) DATABASE INDEX

The database index is an important concept in enhancing the data retrieval speed from a database. Smith (2006) has described how a database index helps to speed up data retrieval while simultaneously reducing database server load.

Database indexes are created in one or more columns of a table and enable a database system to perform rapid random lookups on the database table. Such database indexes are part of the database structure, and are stored in the database physically. However, the database index is much more compact than the database table itself because it only references the key-fields rather than the whole table. The architecture of a database index can be classified as clustered or un-clustered. An un-clustered index is not based on the ordering of the actual records, and works in a similar way to the index at the back of any book. A clustered index, by contrast, is based on the ordering of the actual data records, and works in the same way as a phone book ordered by last name.

While on the one hand a database index makes searching through the table efficient, on the other hand it makes the process of amending the table through insertions and deletions slow and laborious. If a column in a table is indexed, a new index number is created while a record is inserted into the table and, in the same way, an index number is removed while a record is deleted from the table. Thus if an application involves a lot of insertions and deletions and less searches, it is better not to use indexes.

4. WEB APPLICATION DEVELOPEMENT STANDARDS

Web applications development standards are the third important part of any web-based application. They provide the standards concerned with the actual development of the web application and entail different techniques and technologies that, once used efficiently, can enhance the performance of a web application. They can be divided into design standards and development standards.

A) DESIGN STANDARDS

Design standards are concerned with the pre development structure of a web application, and are usually decided prior to commencing development of the web application.

WEB WIREFRAMING

Web wireframing is a unique concept in web site design and development. Miller (2008) has described web wireframes as 'simple line drawing' to show the placement of elements in the web page. Web wireframing is used to provide a visual guide to the website and demonstrates the links between different pages in the website. Web wireframing helps in creating a final design of the

website before its actual development. This saves a lot of time if there are any changes needed in the design and it helps in making a final design decision before development of the web application.

Wireframing has been used for a number of years in web site design and development and can help in enhancing the usability of a GIS web application. Because developers can see the web application design in prototype form and give feedback, it is possible to develop an application which marries user expectations with requirements. Several tools are available to design web wireframes, of which Adobe Illustrator (http://www.**adobe**.com/products/**illustrator**), Microsoft Visio (http://office.microsoft.com/visio), and ProtoShare (http://www.protoshare.com) are some of the popular.

Figure 3 shows an example of a web wireframe, which demonstrates how the web page will look like after development.
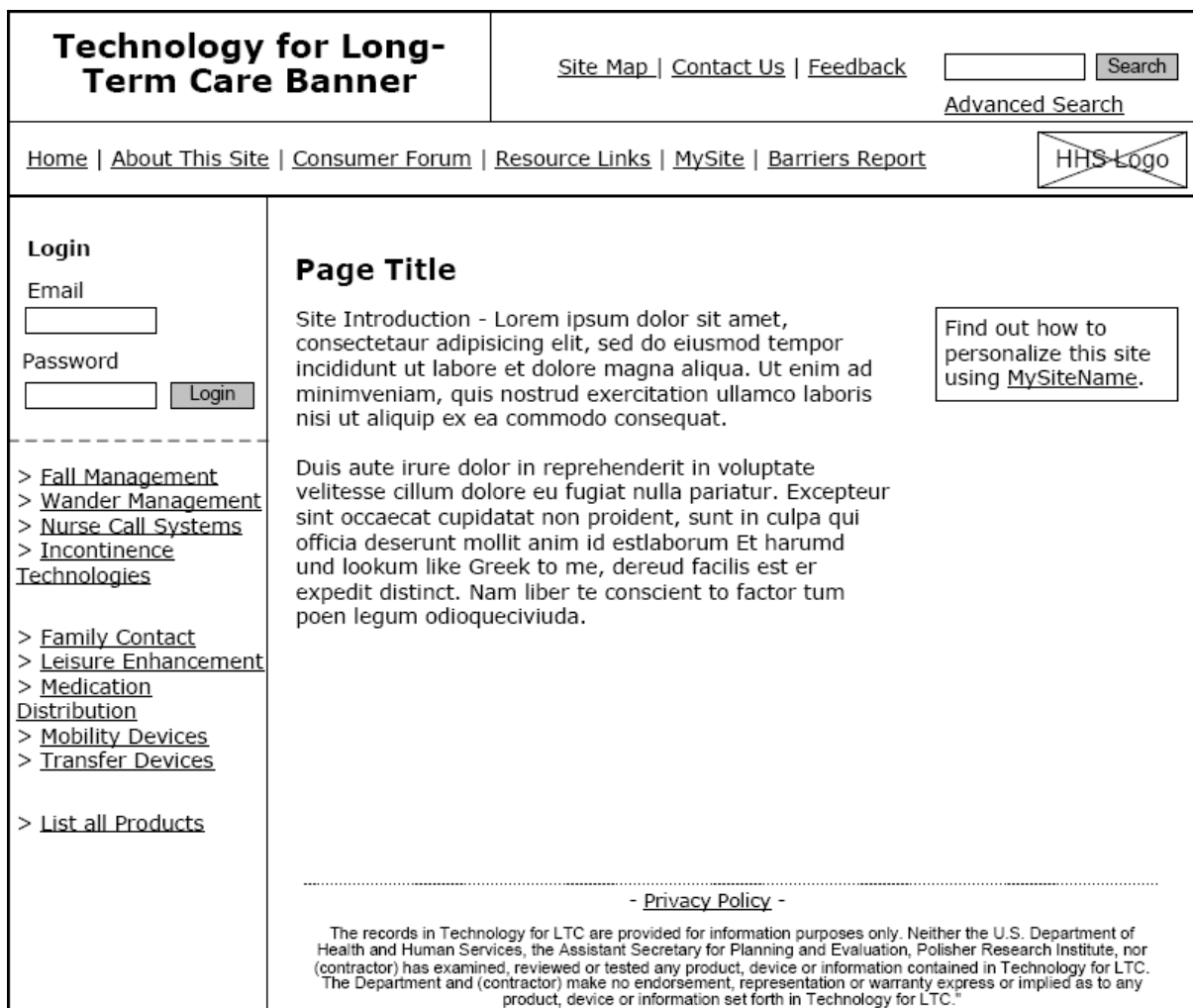

Figure 3: A sample wireframe of a web page

DESIGN PATTERNS

Chambers et al. (2000) have described design patterns as a recent software engineering innovation that explains common problems and solutions in object-oriented development. Design patterns have become very popular because they can ease the development of software engineering projects. They provide general solutions to commonly occurring problems by using a template to clarify the relationship between different entities of a software or web application. Abstract factory, Factory method, Builder, Lazy initialization, Singleton, and Prototype are examples of software engineering design patterns. They can be used in web-based GIS applications to standardise the application according to the problem domain and hence tuning the web applications to work better.

B) DEVELOPEMENT STANDARDS

Development standards describe the choice of an appropriate development technology for the web application. For example, .NET is a development technology used to develop software or web applications and Oracle & SQL Server are database servers which can be used as the back end of web applications. Most of the development standards remain same for different development technologies, and these describe a way to use different development technologies in efficient way. In addition they also describe the efficient use of memory and physical storage, and provide a common standardised protocol (e.g. SOAP for XML) for communication with other web applications.

CHOOSING THE APPROPRIATE DEVELOPMENT TECHNOLOGY

A range of development technologies is available for different operating platforms in order to develop GIS web-based applications. Important development technologies include ASP (Active Server Pages), JSP (Java Server Pages), ASP.NET (Active Server Page dot Net), and PHP (Hypertext Preprocessor), the choice between which depending upon the developer's operating environment.

ASP (Active Server Pages) is a Microsoft's technology used to develop web applications for deployment on Windows. Users of an ASP Web application can access it from any platform. ASP.NET (Active Server Pages Dot Net) is the enhanced extension of Microsoft's ASP technology which offers object oriented support. Web applications developed in ASP.NET can also only be deployed on Windows platforms, but can be accessed from any platform. JSP (Java Server Pages) is a Sun Microsystems technology used to develop web applications. They can be deployed or accessed on any operating platform because java technologies are platform independent. The same is true of PHP (Hypertext Preprocessor), which is an open source technology used to develop web applications.

USING COMMON DATA STORAGE AND COMMUNICATION PROTOCOLS

A web application often stores temporary data in files or communicates with other web applications over the web. Common communication protocols can be used to enhance the performance of the web application. One such common protocol is SOAP (Simple Object Access Protocol): Kyrnin (2008) describes how a group of vendors, including Microsoft, IBN and Lotus created this XML-based protocol for standardised communication, and how SOAP uses XML (Extensible Markup Language) and HTTP (Hypertext Transfer Protocol) for communication.

The XML-based SOAP has become the *de facto* communications standard for linking two or more web applications or web services. Use of a common communications protocol enhances the performance of the web application because the information contained in the application need not be converted into the application data format. Data sent by the other web application or service can be used directly by the caller web application and this enhances the response time to users.

In addition it is suggested that if a web application stores information locally in files, these should be formatted in XML, since parsing XML files is quicker that parsing raw data files. Thus if a web application is required to write data to temporary files, use of XML can improve the performance of local read/write functions.

USE OF OBJECT ORIENTED DEVELOPMENT STANDARDS

The object oriented programming paradigm uses objects and their interactions to design and develop software or web applications. The use of object orientation in web applications has increased in the recent years since applications are well structured, scalable to new changes, and offer perform better operating performance. Examples of object-orientated development frameworks include Microsoft Visual Studio (2005 and 2008) and JSP (Java Server Pages). From a web-based GIS application point of view, object oriented programming presents a better choice for developing a web-based GIS application and copes with the new challenges involving performance and scalability of applications in a real world environment.

5. GENERAL ARCHITECHTURE OF WEB-BASED GIS APPLICATIONS

Different architectures can be used to develop web-based GIS applications. 2-tier architectures retain the user interface and functional part of the web application on the first tier and deploy the database and data storage functions on the second tier. However, a more efficient architecture is a 3-tier architecture in which the user interface and functional parts are separated on different tiers. In such circumstances, if any part of the application needs to be changed, other parts do not get affected by that change, and the web application remains scalable to changes in different development environments.

Figure 4 illustrates two-tier architecture, and Figure 5 shows one with three tiers.
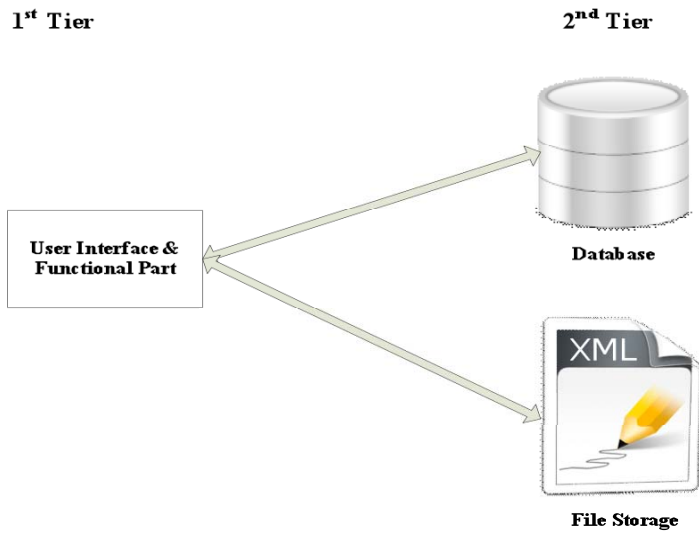
1ˢᵗ Tier   2ⁿᵈ Tier

User Interface &
Functional Part

Database

XML

File Storage

Figure 4: A two tier Architecture

1ˢᵗ Tier   2ⁿᵈ Tier   3ʳᵈ Tier

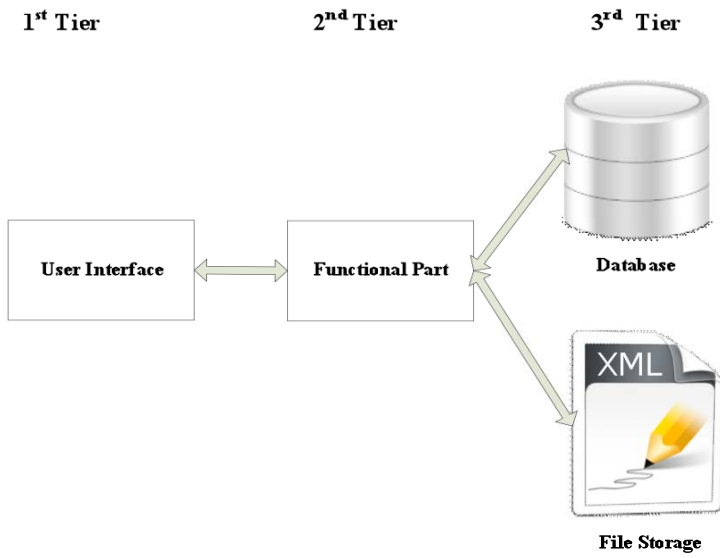User Interface   Functional Part

Database

XML

File Storage

Figure 5: Three tier Architecture

Based on 3-tier architecture and the discussion in other sections, the following architecture is suggested when developing web-based GIS applications.
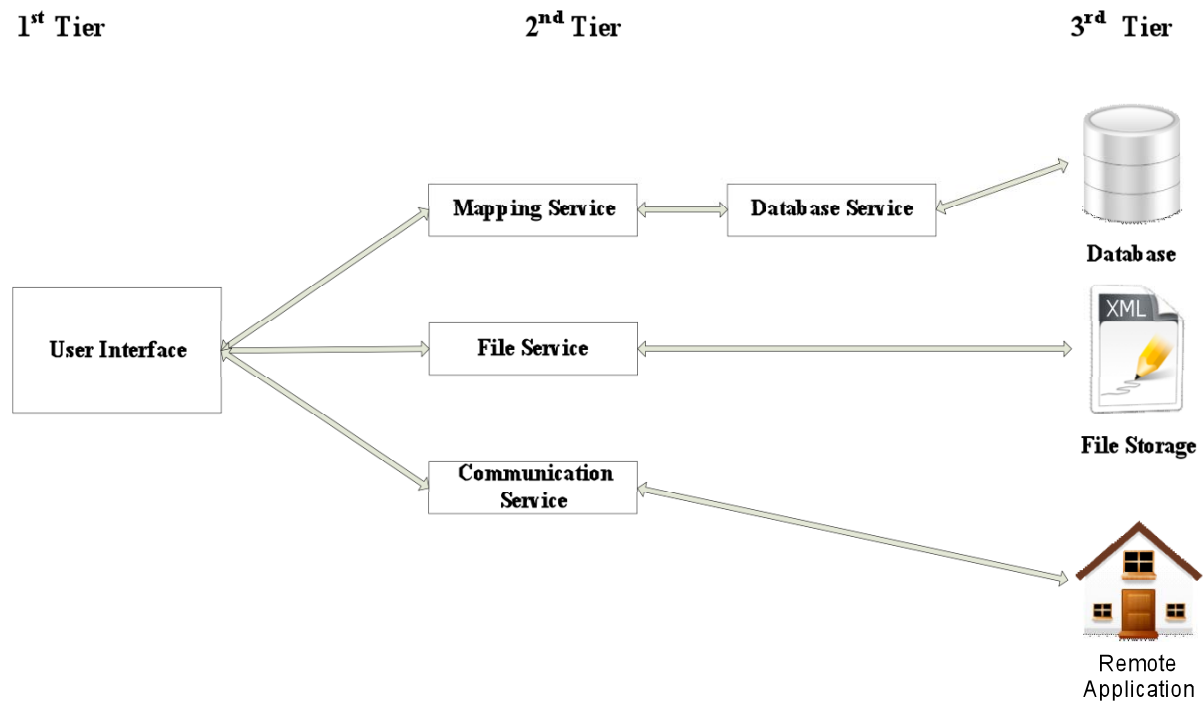

Figure 6: Three-tier web-based GIS applications Architecture

The user interface is on the 1st tier, while all of the business logic functions lie on the 2nd tier. A business logic part is defined as the part of the web application containing all the functional parts. For example, if a web application is accessing data from a geo-referenced database, the part of the web application which reads data from the database is said to be the business logic. In the Figure 6, the functional part includes a mapping service which produces and displays maps to the users. The mapping service interacts with the database service in order to retrieve data from the database, as required. File and communications services are only required in particular scenarios. The file service writes data into XML files, while the communications service interacts with other remote applications in order to receive live feeds of data, as required.

5. CONCLUSION

This paper has described the different mapping technologies, database standards, and web application development standards that are relevant to the development of web GIS applications, and it is important to match any given application to the appropriate technology. Oracle is the system of choice for geospatial applications that need to handle large amounts of data. Normalisation, stored procedures, and database indexes are important when working with any database, but they are not useful in every situation. Database indexes are not useful when size of database tables is very large as it can increase insertion and deletion times. Normalisation is only useful when web application does not need to join many database tables as it can increase joining time. In the same way, stored procedures will increase performance time of a web application if the size of database tables is large.

Special care is needed when using them in web-based GIS applications. Wireframing and design patterns have been shown to be useful in making GIS web applications efficient, scalable and usable, and should be an important part of every web-based GIS application. A range of different development technologies are available, and their use in different operating environments has been discussed here in some detail. Use of a common communication protocol (SOAP) alongside the object oriented programming paradigm is strongly recommended for the development of efficient web-based GIS applications.

6. REFERENCES

Aoidh, E. M., Bertolotto, Wilson, D.C. (2008). "Understand geospatial interests visualising map interaction behaviour". GeoVisualization of Dynamics, Movement and Change, 7(3), 275-286.

Burnard, T. (2000). "Slave Naming Patterns: Onomastics and the Taxonomy of Race in Eighteenth-Century Jamaica." Journal of Interdisciplinary History, XXXI(3), 325-346.

Chambers, C., Harrison, B., Vlissides, J. (2000). "A Debate on Language and Tool Support for Design Patterns". Proceedings of the 27th ACM SIGPLAN-SIGACT symposium on Principles of Programming Languages (pp. 277-289).

Fotache, M. (2006). "Why Normalisation failed to become the ultimate guide for database designers?". Social Science Research Network Working Paper Series. [online]. Retrieved on 1 June 2009 from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=905060.

Gibin, M., Singleton, A., Milton, R., Mateos, P., Longley, P. (2008). "An Exploratory Cartographic Visualisation of London through the Google Maps API." Applied Spatial Analysis and Policy, Vol. 1, 85-97.

Haklay, M., Singleton, A., Parker, C. (2008). "Web Mapping 2.0: The Neogeography of the GeoWeb." Geography Compass, 2(6), 2011- 2039.

Kyrnin, J. (2006). Simple Object Access Protocol - SOAP. [online]. Retrieved on 2 June 2009 from http://webdesign.about.com/library/weekly/aa031802a.htm.

Microsoft (2009). Stored Procedures. [online]. Retrieved on 1 June 2009 from http://msdn.microsoft.com/en-us/library/aa214299(SQL.80).aspx.

Miller, E. (2008). How to create website wireframes. [online]. Retrieved on 1 June 2009 from http://graphicdesign.about.com/od/effectivewebsites/ss/wireframes.htm.

Oracle (2005). "Technical Comparison of Oracle Database 10g vs. SQL Server 2005: Focus on Performance.". Oracle White Paper. [online]. Retrieved on 4 June 2009 from http://www.oracle.com/technology/deploy/performance/pdf/twp_perf_oracle%20database%2010gr2%20vs%20ss2005.pdf.

Smith, C. (2006). Introduction to Database Indexes. [online]. Retrieved on 1 June 2009 from http://www.interspire.com/content/articles/34/1/Introduction-to-Database-Indexes.