OPEN-SOURCE TECHNOLOGIES IN WEB-BASED GIS AND MAPPING

A THESIS PRESENTED TO
THE DEPARTMENT OF GEOLOGY AND GEOGRAPHY
IN CANDIDACY FOR THE DEGREE OF
MASTER OF SCIENCE

By
ERIK HARPER

NORTHWEST MISSOURI STATE UNIVERSITY
MARYVILLE, MISSOURI
APRIL, 2006

Open-Source Technologies in Web-Based GIS and Mapping

Abstract

This project is aimed to provide a scientific and thorough analysis of alternative options to building spatial websites through the use of open-source technologies. In comparison to costly software licenses, restrictions on use, interoperability, extensibility, ease of use, training resources and performance, open-source technologies have been proven in recent years to meet and in some cases surpass the abilities of proprietary software to produce effective and robust web-based spatial applications. The goal of this project is to analyze the industry-leading software tools and packages to provide a comprehensive cost-benefit analysis and gain a better understanding of the available alternatives to building spatial websites. This study proves that an open source web-based GIS configuration provides a viable alternative to using proprietary software in terms of cost, ease of use, performance and conformance to open standards.

Table of Contents

List of Figures

List of Tables

Acknowledgements

The author would like to thank Whit Sours from the Central Shenandoah Planning District Commission for providing the ArcIMS 9.0 license and the GIScience faculty at Northwest Missouri State University, including Dr. Hung, Dr. Drews and Dr. Wu, for guiding me through this research.

List of Abbreviations

API – Application Programming Interface
GML – Geographic Markup Language
GRASS - Geographical Resources Analysis Support System
IIS – Internet Information Services (Microsoft)
LGPL – GNU Lesser General Public License
OGC – Open Geospatial Consortium
OS – Open Specifications or Operating System
OSGeo – Open Source Geospatial Foundation
OSS – Open Source Software
OSWBGIS – Open Source Web-Based Geographic Information System
PPGIS – Public Participation in Geographic Information Systems
RDBMS – Relational Database Management System
SVG – Scalable Vector Graphics
TCO – Total Cost of Ownership
uDIG – User-Friendly Desktop Internet GIS
WBGIS – Web-Based Geographic Information System
W3C – World Wide Web Consortium
XML – Extensible Markup Language
XSL(T) – Extensible Stylesheet Language

Chapter 1: Introduction

Web-based Geographic Information Systems (WBGIS) facilitate the widespread use and dissemination of spatial information and services and promote the technology to a much greater audience than it has ever been introduced before. The utility of the Internet allows information to be exchanged in a rapid and efficient manner, thereby helping individuals make important decisions quicker. The applications running on the Internet, known as the World Wide Web (WWW) give Internet users countless powers for obtaining and disseminating information and services. In the field of Geographic Information Systems, the Internet has played a significant role in the development of new facets of the technology that open many doors for expanding the options for building spatially-enabled web applications. The software developed for building these types of systems varies in terms of cost, efficiency, scalability, robustness, security, support and ease of use. This study will analyze a special sub-field of Web-based Geographic Information Systems known as Open-Source Web-based GIS (OSWBGIS).

Open Source Software (OSS) is a topic of study all in its own, but for the purposes of this research, it will involve particular OSS applications used to serve spatial data over the Internet. The term Open Source Software can be defined as software that is freely available to the general public for free use, alteration of source code and dissemination of altered source code without imposed restrictions on the use of the modified software and profit to the modifying user. Many organizations and individuals involved with open source software feel that allowing a program to be altered by the general public is a more efficient way to develop software compared to an isolated group of programmers on the payroll of a corporation accomplishing the same task. Many successful open source software projects exist, most notably the Linux platform, the Apache Software Foundation, the MySQL database application, and countless others. In some cases, an open source software option may even hold a higher market share in a particular market, for example the Apache Web Server product. According to the Netcraft Web Server Survey (Netcraft, Ltd 2006), the Apache Web Server is used to power roughly 68% of the 76,184,000 surveyed websites on the Internet, compared to Microsoft's Internet Information Server, holding at 20%, as illustrated in Figure 1. Table 1 also shows a breakdown of the top web server developers by percent change from January 2006 to February 2006. So, there are definite cases where open source is worth considering when deciding on deploying mission-critical systems on the Internet.

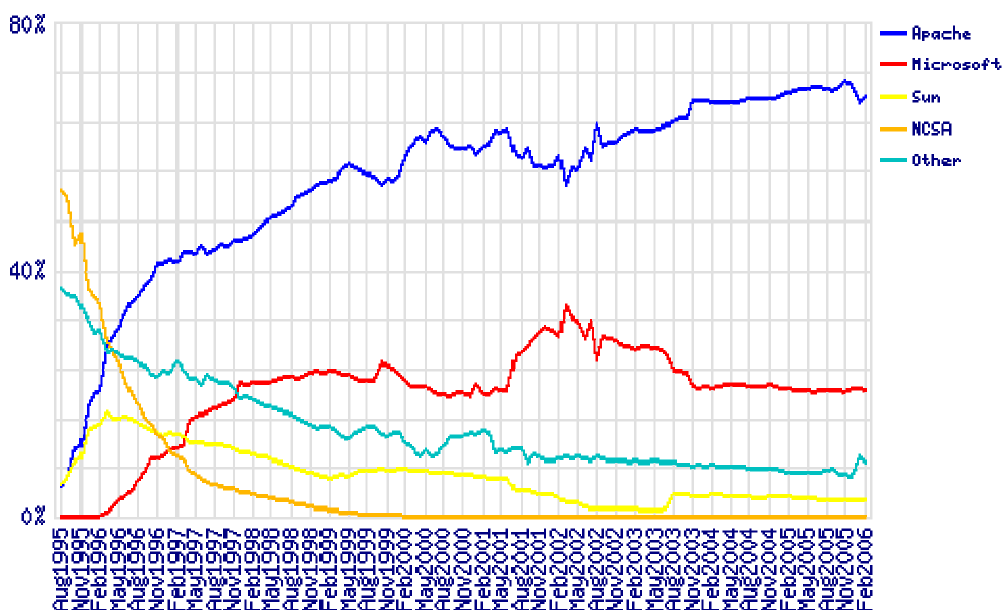

Figure 1. Market Share for Top Web Server Products Across All Domains August 1995 – February 2006 (source: http://news.netcraft.com/archives/web_server_survey.html)

Table 1. Number of Sites and % Market Share for Top Web Server Products (source: http://news.netcraft.com/archives/web_server_survey.html)

Developer	January 2006	Percent	February 2006	Percent	Change
Apache	50502840	67.11	51810676	68.01	0.90
Microsoft	15510953	20.61	15666702	20.56	-0.05
Sun	1879856	2.50	1880313	2.47	-0.03
Zeus	561524	0.75	579198	0.76	0.01

As opposed to proprietary or commercially produced software, a trend towards utilizing open-source technologies has recently come into the forefront of WBGIS for many reasons not limited solely to the difference in financial savings. It can be argued that proprietary software options for implementing spatial websites have been built to an unwieldy level of complexity, making time-to-market an issue in the process of producing a spatial website. Lowe (2002) argues that while much time and money have gone into producing spatial server products, they can become too complex but also more difficult to install, more expensive, harder to maintain, and slower to serve maps. He also feels that the justification for most proprietary software becoming too complex is so that the new products are more scalable, secure, robust, and full-featured than their forebears.

Organizations that wish to implement spatial website services for their own benefit should weigh the pros and cons to choosing proprietary or open-source software. Making quick decisions without weighing the benefits could result in the loss of many cost, time and resource saving opportunities. Sometimes a proprietary software solution could prove more valuable than an open source one, and vice versa depending on the problem it is intended to address. A comparison between a proprietary software solution and an open source solution could provide critical information in such a decision making process.

The rationale for performing this type of research stems from the lack of concrete performance analysis in existing literature related to the subject of OSWBGIS products. Also, with the growing trend, popularity and commercial support of open source projects in the web-based technology market, to focus on the impact of open source is noteworthy for in depth study and research.

Research Objectives

The research objectives for this study include how OSWBGIS technologies impact the field of WBGIS and how OSWBGIS options compare with proprietary software options in terms of specific factors used for judging overall performance. The overall goal of this research is to promote the awareness of these technologies and evaluate them experimentally as they relate to the field of geographic information science.

Chapter 2: Literature Review

The growing trend of the adoption of open source technologies for WBGIS is largely due to the fact that many successful open source software projects have proven under many circumstances to perform at acceptable and sometimes exceptional levels compared to proprietary products. The trend can be seen more and more through commercial/organizational support for open source projects and widespread adoption of open source technologies. This literature review will examine many key pieces of literature that back up the argument that open source technologies impact the field of WBGIS in more ways everyday as well as mention important examples of successful proprietary WBGIS systems and products for comparison.

Open Source WBGIS

Many will hold different opinions about the open source vs. proprietary software debate, from users who feel OSS products can't even compare with their proprietary counterparts to users who feel just the opposite. Despite what opinions either side holds on the subject, OSS is a force to be reckoned with and deserves the attention of the GIS community as a subject of discussion, research and experimentation. There is a growing argument within the field of WBGIS advocating for more efficient standardizations of GIS data transfer across multiple platforms, in universally-understood languages and protocols that is a huge driving force behind most open source projects. It is these Open Standards (OS) that provide a guide and well-defined structure for how GIS data can efficiently be shared across a heterogeneous global network of computers. And, it is these same Open Standards that many successful open source products adopt, thereby increasing their likelihood for widespread adoption. The Open Geospatial Consortium (OGC) is an independent organization for the development of standards for geospatial applications and services. Among the 311 compliant or implementing products as of January 29, 2005 (Netcraft 2006), the amount of open source products is quite extensive, including The University of Minnesota's (UMN) MapServer and PostGIS. These open source products exist alongside many proprietary software applications by the leading GIS vendors, including Environmental Systems Research Institute (ESRI), Autodesk and General Electric (GE). This is clear evidence that the open source GIS movement is in full force and not slowing down anytime soon.

Tyler Mitchell (2005), author of the book *Web Mapping Illustrated,* is the first to publish a book that focuses solely on web mapping as an Information Technology (IT) practice within the context of open source technologies. Mitchell points out that a new world of open source collaboration along with a healthy commitment from industry is filling the need and demand for spatial technologies at a growth rate unmatched in the industry. He writes this book not as a GIS professional, but assuming the reader has never heard of the term. His focus on the technologies that are readily available to the general public through open source projects such as MapServer and PostGIS in a way that seeks to, as he puts it, "fill in the gaps of existing documentation and answer the new user's common questions". Mitchell takes the point of view that open source technologies are viable alternatives to proprietary GIS software especially in the WBGIS and mapping realm. He supports the open source philosophy of software development, where a community of developers contributing to the project is a successful model for the evolution of a software product and feels that most projects promote interoperability by implementing open standards, and in many cases implement them faster than their proprietary counterparts. He also provides detailed installation instructions for MapServer and PostGIS and provides helpful advice for using the products. His lessons on map files supported by MapServer also help a smooth transition into using the product that anyone with a background in computer science and GIS could pick up easily. Mitchell's work is a significant contribution to the field of OSWBGIS, and is currently the largest publication on this particular topic.

Financial Benefits of Open Source WBGIS

In 2003, the U.S. Department of the Interior sponsored a GIS technology survey in which 1,156 local government jurisdictions responded to questions relating to the organization's use of GIS technology (Center for Disease Control (CDC) 2004). Overall, the survey found that of the organizations that had concrete GIS infrastructures, GIS is recognized as an essential tool for accomplishing many of the organization's goals. The survey found, however, that there are many barriers for some to adequately use GIS technology to their benefit. Some of the largest barriers to using GIS more effectively from most common to least common are funding, technical expertise and awareness. Of the 1,156 surveyed organizations, over 64% of them are interested in GIS but do not have the funding to implement them. It seems that of the organizations that know enough about GIS that they see a clear benefit from it, costs are the biggest issue. The survey concluded by stating various calls to action for federal, state and local governments. To summarize, some of these calls to action include outreach programs to increase awareness of GIS, provide GIS tools over the Internet, support cost sharing programs, lead and participate in regional cost and information sharing initiatives and promote the development of enterprise GIS and shared infrastructure across separate government organizations. Based on these survey results, it is clear that because cost and awareness are large issues that hinder the use of GIS technology, the open source movement can effectively meet some of these challenges by providing cost-effective, intuitive, industry supported and community supported solutions.

Looking at open source software from a philosophical and economical standpoint, Ramsey (2002) clearly supports the open source standpoint, feeling that the more successful an open source project becomes, the more developers are attracted to it, thereby furthering its own growth and success. Most of the time, the developers interested in improving the product end up benefiting personally because they are making their own jobs easier through the improvements to the software. Ramsey quotes Bob Young, co-founder of Red Hat, a successful Linux distribution company, on his view of the ideal software market:

> *"The software market should be one in which consumers don't purchase software per se, but instead purchase whatever services they need to effectively use the software they choose. Rather than purchase a proprietary database system and then purchase support from the proprietary database company, customers instead choose an open-source database and purchase support from an array of support companies with expertise in the chosen database. The net effect is the same--customers have functioning and supported products--but the balance of power is shifted in favor of customers"* (Ramsey 2002).

With this viewpoint in mind, Ramsey solidifies this philosophy and feels that the open source movement can promote a stable software economy. The trend for private companies to offer support services for open source software products is expanding rapidly. DM Solutions Group and Refractions Research, Inc. are two examples of companies that develop software in support of open source projects and make their own efforts free to the public, but also charge at minimal costs their knowledge and expertise at implementing GIS with these open source tools. In the consumer's eye, this is definitely a win-win situation.

Barr (2005) more clearly delineates the difference between open source and proprietary software in a way that critiques the philosophy of closed source software development. He argues that most commercial software producers develop software with a more narrow range of expertise as opposed to open source teams and that their primary motivation is their salary. In contrast, open source developers are more motivated by personal interests or need of a particular application for themselves and usually include a much wider range of technical backgrounds. It seems also that for many open source projects, particularly the ones that really succeed, like Apache, the sheer amount of collaboration on such a project would definitely be larger than a single software company like Microsoft working on a comparable software product. With the entire world at the open source project's disposal, development is sped up, bugs are fixed quicker and the overall quality of the product exceeds at sometimes a higher rate than a commercial counterpart. Another interesting argument that Barr makes is his explanation of Total Cost of Ownership (TCO) between open source and proprietary. TCO should encompass both up-front hardware and software costs, but also the ongoing maintenance, upgrade and training costs associated with the software and hardware. He compares both sides to the argument by citing that first, Microsoft feels that Windows results in a lower TCO in terms of IT staffing costs due to its out-of-the-box functionality and its familiar, consistent-looking Graphical User Interface (GUI). Still, others feel that considerable savings can exist with the adoption of open source software. Barr cites a UK government report that suggests a typical hardware refresh period for Linux systems was 6-8 years as opposed to 3-4 years for Windows-based systems. Also, with the corporate support of open source projects, most notably Red Hat, International Business Machines Corporation (IBM) and Novell, software consumers sometimes choose service over licensing.

Lowe (2002) talks about the impact of open source software in the web-based GIS and mapping market by actually testing some of the popular open source products for producing web-based mapping applications. He first argues that in the spatial industry OSS has a barely measurable market share compared with proprietary vendors like ESRI, Autodesk and Intergraph, but that when focused on web-based mapping applications, these large complex products offer scalability, security and robustness that in some cases does not outweigh the benefit of a lightweight, easy to use solution for a WBGIS with basic functionality. Within Lowe's experiment he found that within an hour of installing MapServer a website was up and running with basic map tool functionality, such as zoom, pan and identify. Lowe then goes into the discussion for the financial implications of open source software development. He cites specific examples where financial support from private and public sector organizations help fund the continuing development of OSS products like MapServer. Organizations like the National Aeronautics and Space Administration (NASA), DM Solutions, Canadian government organizations, Brazilian government and the Minnesota Department of Natural Resources are just some of the examples of financial support for MapServer he cites. Lowe also cites a case with the government of British Columbia and Refractions Research, Inc., where an interesting relationship occurs between a company that uses OSS products to produce customized software for the government, charging only for consulting services. This is an example of how many organizations use OSS to keep alive in the software market by developing customized software solutions for their clients without strict software licensing costs, economic TCO cost-savings through consulting fees only and a greater focus on consumer requirements as a result.

From a more general open source topic standpoint, Camara (2004) discusses the challenges developing GIS in open source. He explains that OSS projects reach a critical size where many benefits come about where the application offers rich functionality, robustness, cooperation from contributing developers and continuous improvement. He identifies common misconceptions taken towards open source projects, including the idea that OSS is developed by an individual or small group as opposed to a global network of professionals, that OSS systems are complex and innovative and therefore are standalone and noninteroperable and that there is a single repository for development. His idealized model of OSS development is a network of committed individuals that take advantage of previous conceptual designs and high problem granularity through effective peer production. This means that the more individuals committed to production of software, the more granular a problem becomes and thus easier to manage the pieces of the problem. The most interesting theory that Camara formulates is his four types of open source software: 1) high reverse engineering with high distribution potential, 2) high reverse engineering with low distribution potential, 3) low reverse engineering with high distribution potential and 4) low reverse engineering with low distribution potential. The software products in the first category are the ones with the highest success rate because they make use of existing technologies the most and there is a high distribution potential because of community-led support. Software products falling into this category are projects that are community and commercially supported, such as Linux. Camara also summarizes the balance of four powers relating to this 4-way OSS model: the balance between innovation (low-low), collaboration (low-high), corporation (high-low) and community (high-high) where the first value refers to reverse engineering and the second refers to distribution potential. The ideal category for a successful OSS product would balance between the corporate and community categories based on the varying degrees of community and corporate support.

Example Open Source WBGIS Solutions

Just as Hypertext Transfer Protocol (HTTP) and Hypertext Markup Language (HTML) have helped computers communicate through virtually every platform, OGC standards have strived to accomplish a similar task in the spatial industry. Reichardt (2004) outlines the impact of the OGC in terms of system interoperability in the exchange of geographic information over the Internet. For almost all open source GIS projects, the OGC specifications are the foundation. With all of the example WBGIS mentioned in this study, OGC standards for interoperability are shown to be the cornerstone to success.

Smotritsky (2004) advocates for the open source community of GIS products by summarizing his key players in this market, including Geographical Resources Analysis Support System (GRASS), PostgreSQL and PostGIS, MapServer, Thuban, QGIS and GRASS Server. He focuses on pertinent details of each of the software products that would help end users and developers in their decision to choose one over any of the proprietary solutions in the GIS software market. He first focuses on the desktop GIS solution GRASS powered by a PostgreSQL/PostGIS backend for storing data since the native version of GRASS only supports a single attribute for each spatial object (vector or raster). He then discusses GIS on the web using MapServer and lists supporting packages and helpful hints for installing and troubleshooting as well as referencing web links to give access to the abundance of documentation available for the product.

Turner (2005) in a project evaluating GIS software options argues that there should be a very clear justification for choosing a closed source option based on certain factors he outlines. The factors are that there are numerous options for open source solutions, it is difficult to evaluate many of the commercial options because of restrictive licensing, and that open standards approaches that comply with OGC are a first priority. He outlines other reasons that closed source options limit the developer including that the Applications Programming Interface (API), if available, is not enough because the underlying code is still hidden. On his discussion of closed source options, he concludes that open source is generally better because the code reveals exactly what is going on, but he also cites ESRI as a good candidate for a closed source option for a state-of-the-art GIS. In the end, he makes a recommendation to use uDIG (User-Friendly Desktop Internet GIS), a product developed by Refractions Research that is a desktop application that can import local data and data from the Internet for many GIS operations. Based on the goal for his project, which is to utilize a user-friendly Internet GIS application, Turner felt that uDIG, with its richness of features out of the box would bode better than developing an application with MapServer. Turner's support of uDIG is yet more evidence of an open source option chosen over a proprietary one.

Neumann (2003) explores the use of OSS to build an enterprise GIS database feeding into a map server that allows for the extraction of subsets of a large database through server and client-side scripting. He first lists requirements that should be desired in a database system, a web server system and a client viewer system. His recommendations for requirements for a database system include that the system should be reliable, support geometry and spatial data types, be standard compliant (SQL92 and OpenGIS), and able to handle large datasets and support major operating systems and programming languages, including open source technologies. For map server requirements he feels the product needs to be reliable, customizable, extensible, support all major operating systems and languages and should be scalable and allow for load balancing. For his particular implementation, he felt that the use of a Scalable Vector Graphics (SVG) viewer would offer superior graphical quality, interactivity, W3C (World Wide Web Consortium) conformance and a free-to-download plug-in that would support major operating systems and allow for scripting, among other factors. For the client viewer system he recommends that it should provide superior graphical quality and interactivity, should be available for free or at a cheap cost, should support W3C-compliant data formats (preferably eXtensible Markup Language (XML)-based), should allow for scripting, extensibility and animation and should be available for all major operating systems and browsers. While he does not go into much technical detail about the map server in this document, he does provide some guidelines for good database, map server software, client viewer software and recommendations and experience migrating data into PostGIS from other vector and raster data formats.

Anderson and Moreno-Sanchez (2003) offer a formal treatment of the topic of Open Specifications (OS) applied to OSS for the construction of web-based spatial information solutions. They cite motivation for employing open source software for constructing WBGIS due to the high costs of proprietary solutions, complexity and special requirements. They introduce the concept of the Web Services Model of distributed applications in contrast with the traditional client-server model, where data and tools for geo-processing functionality are wrapped in interoperable software components that process on the server side to be delivered to clients. The Web Services Model promotes interoperability and a seamless web application environment through the use of server-side processing of application data. By integrating the application logic on the server instead of relying solely on client-side script execution, this model strives to provide GIS web services to virtually any type of client on any platform as long as the web browser can display HTML and understand XML (which many current browsers support). For more information on the Web Services model as it relates to GIS, Gonzales (2003) provides a thorough discussion. Anderson and Moreno-Sanchez also outline five concrete issues with proprietary web-based GIS software, including the lack of out-of-the-box geo-processing functionality, cost, steep learning curves, requiring IT personnel to become specialists in the software and difficulty integrating into existing IT infrastructures. They implement a system with many parts working together, including a backend database built with PostgreSQL/PostGIS, MapServer map viewer, SVG processing for overlaying interactivity to the map and the use of XML in the forms of XSLT (eXtensible Stylesheet Language) and GML (Geographic Markup Language) for interoperable data exchange. They conclude by stating that while the open source technologies they employed can offer state of the art geo-processing functionality, there is still a lot to build upon. They considered the OSS products used in their experimentation to be the most powerful, widespread, accessible, easy to learn and with a good level of user support in the form of software documentation, books and user-group forums, which helps reinforce the decision for choosing particular products to use in this research. Lastly, they cite a group of helpful advantages of OS and OSS tools for constructing WBGIS including no software costs, tools were easily learned by personnel with general IT backgrounds, small software footprints, no need to commit to proprietary web GIS or DBMS (Database Management System), ease of compatibility with existing IT infrastructure, flexibility to allow for implementation of geo-processing functionality, principles to implement the technologies are straightforward and accessible to a broad audience of GIS developers and the developed system has the potential to interoperate with other systems using the same open specifications.

Raghavan et al. (2002) provide a concise explanation for the implementation of a WBGIS using purely open source technologies for the management of landslide data in eastern countries. They employ GRASS as the GIS server for implementing geo-processing functionality, PostgreSQL/PostGIS for storing the data, MapServer for generating a map viewer and an Apache Web Server to respond to HTTP requests, as illustrated in Figure 2. The setup they use for this project is simple, intuitive and a good model for a successful OSWBGIS. They praise the software's ability to create rich Internet applications and reinforce the fact that taking advantage of open source software benefits organizations that cannot afford proprietary solutions.

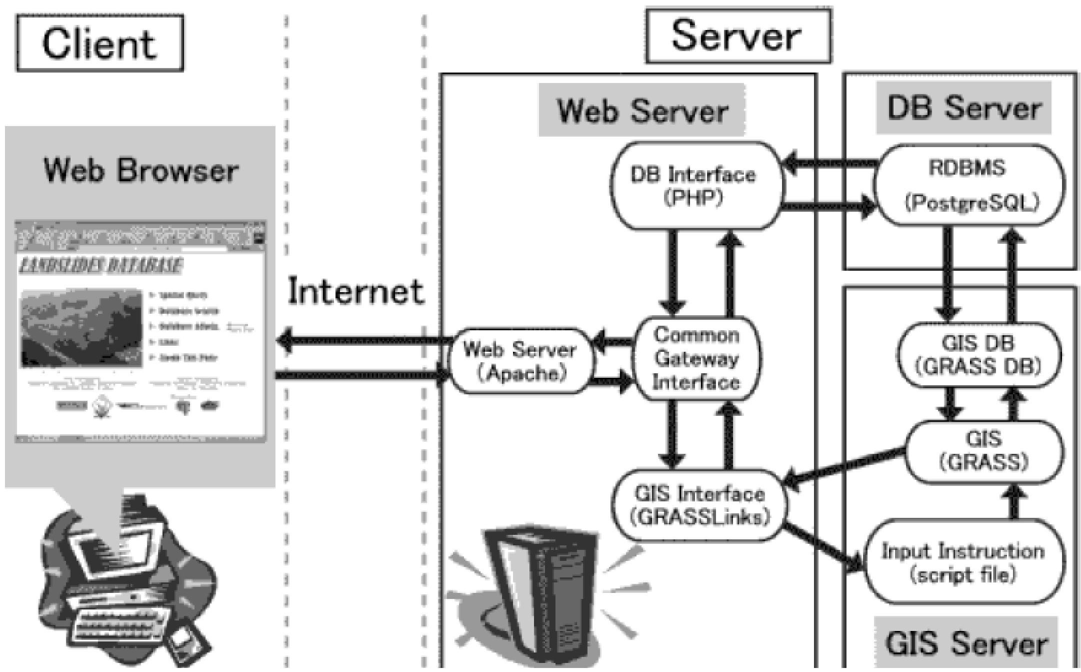

Figure 2. Components and Information Flow within Prototype System (source: Raghavan, et al. 2002)

They also mention that the basic tools for building WBGIS exist in open source, but existing OSS projects have the potential and community support to allow for widespread development of Spatial Data Infrastructures (SDI).

Proprietary WBGIS Solutions and Example Implementations

There are a number of successful software vendors in the spatial website market, including ESRI, Autodesk and Intergraph, to name a few. Geospatial Solutions Magazine provides an annual Buyer's Guide report that outlines GIS products and services that commercial vendors offer to the geospatial market. Among these types of products and services are tools for implementing spatial websites and for services related to spatial website implementation. Among the most successful web-mapping software solutions, ESRI, Intergraph and MapInfo stand out as being comparable options for utilizing a proprietary WBGIS software solution.

ESRI leads the spatial mapping industry with ArcIMS (ESRI 2006a). It provides a full-featured web-based GIS application out of the box with support for standard data file formats. ESRI is also becoming more OS-compliant, based on the Open Geospatial Consortium's listing of implementing and compliant software products (OGC 2006). This listing shows many other proprietary and open source software providers and the OGC specifications that the products are implementing. ESRI catalogs successful implementations of their ArcIMS product with a site that provides links to many live ArcIMS sites particular clients have implemented (ESRI 2006b). The ArcIMS products make use of a number of other software applications, including enterprise DBMS such as Oracle, Structured Query Language (SQL) Server and Informix using the ArcSDE product (Spatial Database Engine) that connects spatial data into a SQL-based DBMS system (much like what PostGIS does for the PostgreSQL DBMS). ArcIMS also makes use of multiple web server products, including Microsoft's Internet Information Services (IIS) and Apache Web Server as well as Java Servlet Engines such as Apache Tomcat (Apache Software Foundation 2006a) and New Atlanta's ServletExec (New Atlanta 2006) (which both make use of the Java Runtime Environment) for serving out custom applications. Apache Web Server and Apache Tomcat are both open source products, while ServletExec and IIS are not. ESRI's pattern of success in conventional desktop GIS applications has translated into the top WBGIS software solution out in the market today and will continue to be a forerunner in the industry.

Intergraph has the second-largest market share in the geospatial software industry and is a leading competitor in the WBGIS software market industry with their product GeoMedia WebMap (Intergraph 2006a). WebMap boasts raster and vector support, SVG output support, spatial and attribute querying, buffering, spatial intersecting, joining, geocoding and measuring, among a multitude of other functions pertaining to spatial data manipulation. Intergraph also hosts a number of example implementations from some of their clients (Intergraph 2006c). It seems that WebMap supports only the IIS server versions 4.0 to 5.0 and utilizes Open Database Connectivity (ODBC) drivers to connect their internal proprietary database format to other DBMS such as Oracle, SQL Server and Microsoft Access (Intergraph 2006b). The software will only run on certain Windows operating systems, unlike ArcIMS, that can run on Windows, Linux, Hewlett-Packard UniX (HP-UX) and Sun Solaris operating systems.

MapInfo Corporation, with their Desktop/Web mapping application development environment, MapXtreme (MapInfo Corporation 2005) combines GIS analysis capabilities into a single integrated environment for deploying desktop and web applications to support a number of rich GIS functions. MapInfo takes the web GIS approach more from a software developer's standpoint and makes the Microsoft .NET environment the primary technology behind MapXtreme's object model. Since .NET is the primary environment, most of the other technologies the software supports are Microsoft-based. The software only includes support on any Windows operating system, version 2000 and above, supports SQL Server, Access, ActiveX Data Objects for .NET (ADO.NET) and the Visual Studio development environment. MapXtreme allows for database connectivity through ODBC and ADO.NET through backend DBMS such as SQL Server, Oracle Spatial, Microsoft Access, Informix IDS 9.3 and Oracle 9i and 8.1.7. So, again unlike ArcIMS, MapXtreme only supports the Windows family of operating systems, but does support some OpenLS (Location Services) standards and other OGC standards as well as provide a feature-rich environment for developing web-based GIS applications.

Chris Bradshaw (2006), Vice President of Autodesk, Inc., recently announced a unique move in the geospatial industy: a move to support and partner with open source organizations, particularly those in the web mapping industry. As of January 2006, Autodesk is an official founding member of the MapServer Foundation, or as it will become to be known, the Open Source Geospatial Foundation (OSGeo) (Open Source Geospatial Foundation 2005). In addition to being a founding member of OSGeo, Autodesk is now offering the source code to its own proprierary WBGIS solution, MapGuide! The new open source project is titled MapGuide Open Source and is the first ever proprietary product that has been released to the open source community. The product is released under the LGPL (GNU Lesser General Public License) that provides for the best balance of business-friendly use and widespread adoption of the software through open source collaboration. Autodesk has strategically taken advantage of the open source community and philosophy to further the use of their MapGuide product.

MapGuide Open Source employs Hypertext Pre-Processor (PHP), .NET and Java tools for building web applications on Windows and Linux platforms using DWF plug-in viewers or AJAX-based viewers (much like ArcIMS HTML viewer). MapGuide also supports Apache Web Server 2.0, IIS 5.0/6.0 and Tomcat 5.5.9. However, at the time of this writing it appears that MapGuide only supports Windows 2000 Server and Windows 2003 Server (SP1 and above), which could be a drawback if only XP was available. Autodesk's Feature Data Objects (FDO) technology provides the API for accessing and manipulating data from multiple data stores including SDF, SHP, ArcSDE, WFS/WMS, ODBC, PostGIS and MySQL. Figure 3 shows a conceptual drawing of the components that work together to serve out geospatial data through MapGuide Server. This unique product is like a fusion between the proprietary and open source worlds and is yet another sign of the changing times in the GIS software industry.

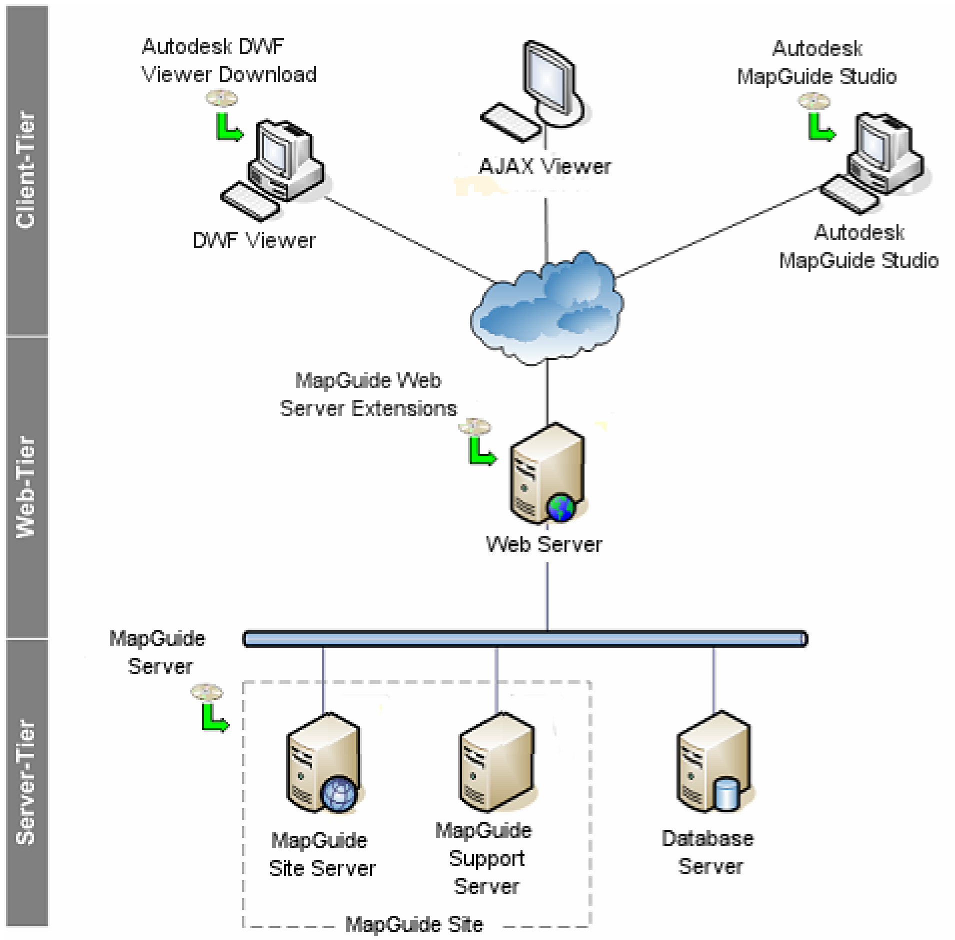

Figure 3. Autodesk MapGuide Open Source Components (source: http://mapserverfoundation.org/mapguide/overview.html)

Chapter 3: Conceptual Framework and Methodology

The analysis in this study for comparing open source and proprietary software options for deploying spatial websites was based on pre-defined criteria defined in this section that describe the components of an ideal WBGIS software solution. The criteria apply to a complete system that likely requires the use of more than one software product. A WBGIS, like most web-based applications, requires many software components to function. It is these software components that were evaluated based on the special criteria the framework sets forth. This study tested the most common software products in both the open source and commercial GIS software markets to get the most meaningful comparison. Choosing the most commonly-used products was reasonable because readers will be able to apply the knowledge gained from the analysis much easier with familiar products in widespread use. Also, adequate documentation exists for implementing the products and they conform to open and industry standards for interoperability.

Conceptual Framework

The conceptual framework for this research was based on a general model for a successful WBGIS implementation. The model encompassed performance considerations as well as accessibility and economy considerations. Performance considerations included basic GIS functionality that is most common to any implementation irrespective of domain application. The following criteria defined the performance considerations for a successful WBGIS implementation. Each product evaluated in this study was checked against these criteria to determine how well they performed and was assigned a rating on a scale of 1 to 5, where:

- ➢ 5 is the best performance possible with very little drawbacks.
- ➢ 4 is great performance with a few minor drawbacks.
- ➢ 3 is good performance with a few major drawbacks.
- ➢ 2 is fair with many major drawbacks.
- ➢ 1 is poor performance with too many major drawbacks for efficient use.

- Criterion 1: All web application software must conform to industry-accepted data and protocol standards, such as applicable OGC standards and W3C standards.
- Criterion 2: All software should be able to run on common system platforms, such as Windows and Linux and be supported in common browser products such as Internet Explorer and Mozilla Firefox.
- Criterion 3: All software should make as much use of server-side processing as possible to provide a rich experience to thin clients.

In addition to the above three general criteria, the following criteria were applied to the Database Server, Web Server, Application Server and Map Server components.

Database Server

- Criterion 4: The database server must support a Relational Database Management System (RDBMS) or data file structure that enforces data integrity, security and reliability, provides adequate support and documentation, allows for storage of industry-standard spatial data formats and/or conforms to industry-standards (e.g. SQL).

Web Server

- Criterion 5: Web server software must provide for efficient process and memory management for responding to HTTP requests, ease of setup and configuration, reliable security and have adequate support and documentation.

Application Server

- Criterion 6: Application server software must provide support for common high-level programming and/or scripting languages for extensibility and customization, efficient compilation/interpretation of program code, provide adequate documentation and be able to communicate efficiently between the web server and the database server (if any).

Map Server

- Criterion 7: Map server software should provide quality output, including clean raster and vector graphics, labeling/annotation, feature identification/querying, support for industry-standard map projections and coordinate systems, basic viewing functionality (zooming, panning), multiple data layer support, basic spatial analysis functionality between layers (overlay, etc.), allow for extensibility and customization through programming/scripting languages, easy to learn, setup and maintain, support for common spatial data formats and able to communicate efficiently with other middleware software (e.g. backend DBMS, application server, web server).

The following criteria were used to evaluate each software product in terms of accessibility and economy. Each product evaluated in this study was checked against these criteria as well.

1) Restrictions imposed on use of software such as:
    a. Minimum system requirements (hardware, operating system, etc.)
2) Total Cost of Ownership (TCO):
    a. Up-front costs for hardware, software licensing and materials
    b. On-going maintenance and licensing fees
3) How accessible it is to acquire support help either from a user community that is responsible for the project or from a corporate sponsor or corporation that manufactures the product
4) Costs associated with obtaining help and documentation for the product (if any)

The philosophy behind the methodology was to obtain an objective viewpoint about the overall performance of particular software components used to construct a fully-functional WBGIS with open source and proprietary methods. Based on this philosophy, the performance of each particular software product was evaluated as it fits into the entire system as a whole. This was important to understanding how the software component fits into the system and how best to judge its performance.

For each individual software component, the following steps were taken in order to completely evaluate it within the context of its role in a WBGIS, from setup to implementation:

1) Average time (in hours) taken to install the product, including requirements gathering and the physical installation of the product
2) Average time (in hours) taken to make full use of the product once the entire WBGIS is configured and deployed based on its function in the system

Once each of the software products had been tested within a hypothetical WBGIS configuration, a matrix was developed that rates both complete systems on the following items:

1) Total time taken to learn, setup and deploy the WBGIS software solution (all times for each software component added up)
2) Total Cost of Ownership (includes a hypothetical hardware configuration, maintenance fee calculation and software licensing fees)
3) Number of Open Standards the total solution conforms to
4) Number of Operating Systems the solution can operate in
5) Whether the solution allows for server-side scripting or not
6) DBMS evaluation ratings based on Performance Criterion#4
7) Web Server evaluation ratings based on Performance Criterion #5
8) Application Server evaluation ratings based on Performance Criterion #6
9) Map Server evaluation ratings based on Performance Criterion #7

The results of the analysis show a breakdown of many criteria that provide a good broad overview of the potential value both the open source and proprietary methods can offer based on the total cost needed to implement the system. The results provide a thorough picture of the two methods.

Methodology

Based on the above considerations for evaluating software products, this study implemented two WBGIS configurations: one using at least one proprietary software product and another using all open source products (excluding the operating system).

Proprietary Configuration

The Proprietary WBGIS configuration utilized ESRI's ArcIMS 9.0 product release, Apache HTTP Server 2.0.48, Tomcat 4.1.29 with J2SDK 1.4.2 on Windows XP platform (Figure 4).

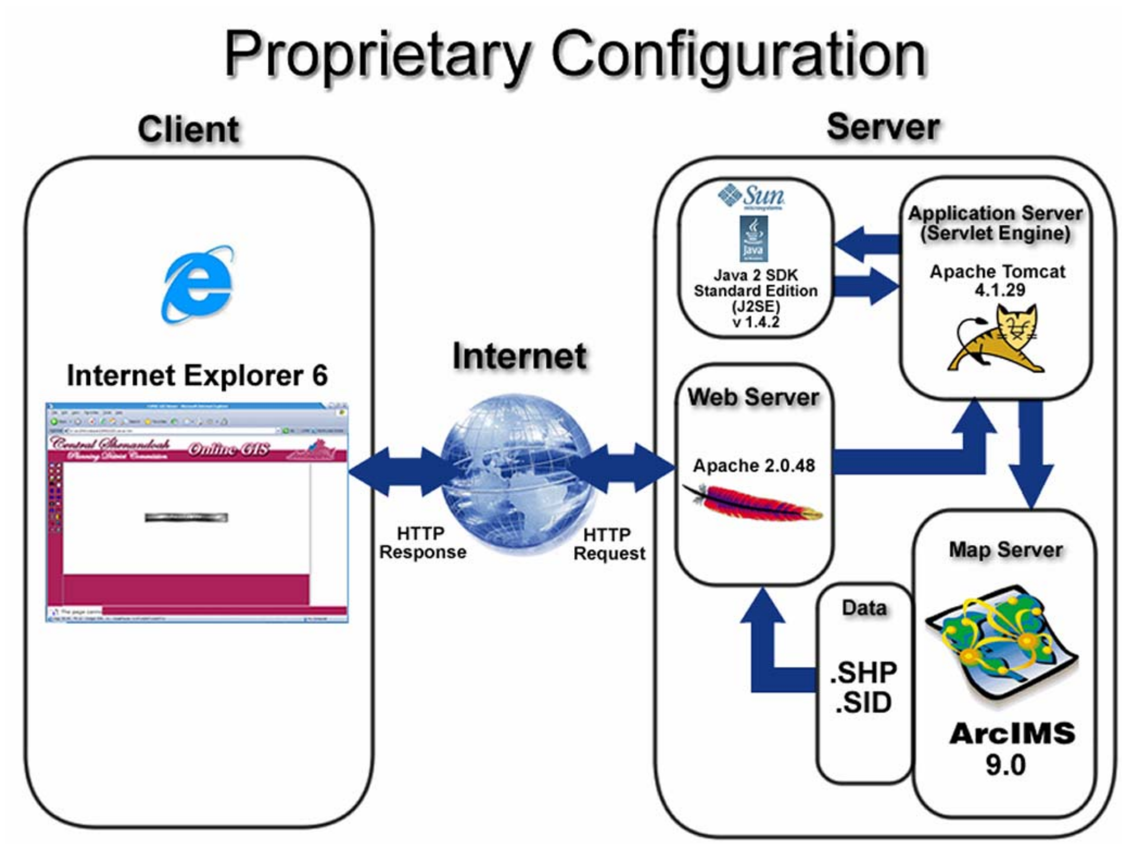

Figure 4. Proprietary WBGIS Configuration implemented in this study.[1]

[1] See Appendix A for copyright notices.

This configuration utilized shapefiles as the primary data store and Internet Explorer 6.x as the browser product. Alternate DBMS were not available and ArcSDE was also not available at the time of setup for testing.

Open Source Configuration

The Open Source WBGIS configuration utilized Apache HTTP Server 2.0.55, PHP 4.4.3, MapServer 4.4/4.5 and employed shapefiles for the data store as well as made an attempt to access the same data from the shapefiles stored in the backend relational database system PostgreSQL, utilizing the PostGIS extension for MapServer to access, (Figure 5). Internet Explorer 6 and Mozilla Firefox 1.5.0.1 were both used to test for comparison, since Internet Explorer is a proprietary browser product and Firefox is open source.

For both of these configurations, the steps previously mentioned in the conceptual framework were taken to rate each of the products. For a realistic financial analysis, research was done to find out real costs for associated hardware and software for deploying these systems based on the combined minimum system requirements of the software components together. After these two pieces of information were obtained, the final analysis showed a performance matrix rating the two configurations based on the performance criteria and the financial criteria. This provided the most comprehensive analysis possible.

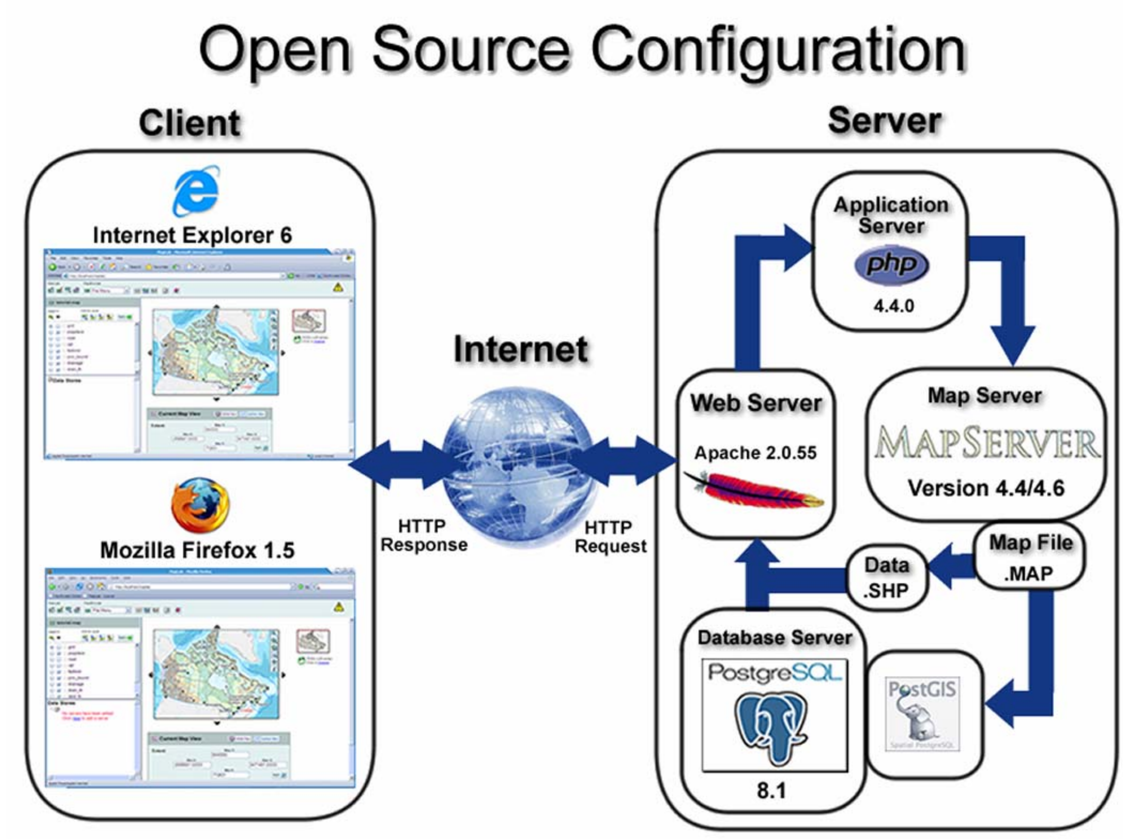

Figure 5. Open Source WBGIS Configuration implemented in this study.[2]

[2] See Appendix A for copyright notices.

Chapter 4: Analysis Results and Discussion

Proprietary Configuration Results

For the proprietary configuration experiment, ArcIMS offers a robust and fully-featured web-based GIS, but does have a somewhat steep learning curve. Gathering a configuration took quite some time because there are only certain middleware products that ArcIMS 9.0 supports. For this step, an installation and setup log was recorded in order to document the time in hours each step took for comparison with the open source configuration setup. Table 2 shows the complete list of tasks taken in the entire setup process and the time in hours each step took on average.

Table 2. ArcIMS Installation and Setup Log

Dates	***Hours***	***Task***	***Description***
3/18/2005 - 3/24/2005	4	Install complete ArcIMS system	Installed Java 2 SDK, Standard Edition Version 1.4.2, Apache Web Server 2.0.48, ServletExec 4.1.1, ArcIMS 9.0, and then configure all components to work together
3/24/2005 - 3/26/2005	3	Run ArcIMS Post Installation	Ran Post Installation multiple times to configure ArcIMS to work right with all the other components. First few post installation runs failed so had to repeat until the configuration worked
3/28/2005 - 4/15/2005	3.5	Re-Install complete ArcIMS system	Removed everything from previous install and reinstalled Java 2 SDK 1.4.2, Apache 2.0.48, Tomcat 4.1.29, ArcIMS 9.0 and then configure all components to work together
4/15/2005 - 5/6/2005	1.5	Run ArcIMS Post Installation	Ran Post Installation to configure ArcIMS to work right with all the other components. Worked the first time
8/3/2005 - 8/15/2005	2.5	Create AXL File	Created the main AXL File through Author and imported data layers, defined layer symbology and labeling, performed data editing tasks to prepare for importing into Author
8/20/2005 - 10/8/2005	2	Test AXL File and HTML Viewer Application	Ran completed AXL File in a basic ArcIMS HTML Viewer Application with basic tools for zoom, pan, identify, buffer, query
Total Hours	16.5		

ESRI's documentation site contains detailed installation instructions for ArcIMS using a number of different component configurations. Since the test machine ArcIMS was to be installed and run on was Windows XP Home Edition SP2, the Microsoft web server product Internet Information Services was not available to be installed as a windows component (IIS 4.x/5.x is available only for Windows 2000 Server, NT, XP Professional and Server 2003 products). Because of this limitation, an alternative web server product had to be used. Fortunately, ESRI built support in for ArcIMS to work with the Apache Web Server. At the time of this writing, ESRI offers 21 unique configurations of ArcIMS and Apache products, so navigating through all the options took some time as well.

A servlet engine was the next product to determine. At first, based on the contents of the ArcIMS installation CD, the product ServletExec by the company NewAtlanta seemed to be a good choice for a servlet engine. Unfortunately, after going through the installation of ServletExec and the post installation configuration of ArcIMS, it was realized that the full version of ServletExec had to be purchased separately. The version on the ArcIMS Installation CD was only a trial version of 30 days. After realizing this, a complete uninstall of the entire system was performed so that Tomcat could be installed in the configuration to replace ServletExec. Tomcat was chosen because it was the only other servlet engine product ArcIMS supported, and it happened to be open source and specially configured to work with Apache, since they are both open source products that are developed to be compatible with one another. At the time of researching installation options, ArcIMS offered only one configuration for ArcIMS 9.0 on a Windows XP platform, and that just happened to utilize Apache Web Server v.

2.0.48 with Tomcat 4.1.29 and J2SDK 1.4.2 (ESRI Support 2005). So, after realizing this the installation was redone with these particular components followed by the post installation to configure and test every component. After this specific technical configuration was nailed down, the next step was to start creating the map file for the data, in this case an .AXL file.

ArcIMS utilizes XML to store information about the data and the environment for the data that displays in the resulting map. It stores projection and coordinate system information as well as which layers are to be displayed at what extents and custom labeling and symbology as well. ArcIMS Author, a Java-based application, provides an intuitive interface for creating AXL files which is similar to conventional GIS desktop applications like ArcView, so familiarity with any other GIS software packages would provide enough knowledge to effectively utilize the product. ArcIMS Administrator is a Java-based application that manages different types of spatial data services, mainly in the form of feature and image services. The main type of spatial data service is an image service which listens to requests for maps to produce images to the client from the data specified in the AXL file. A feature service streams feature data that is not in the form of an image. ArcIMS Designer is the third main application of ArcIMS that allows for the creation of a customized GIS website. It is a wizard that takes the user through a basic setup of core website components, such as the toolbar functionality, the type of viewer to use (HTML or Java) and what spatial data services the website will utilize. It is a quick and easy way to create a basic skeleton website that can be expanded upon later. Overall, the three applications were relatively straightforward but did require some research to figure out how to use them effectively.

Performance Criteria Results for Individual Components

The individual software components that make up the configured proprietary WBGIS system were evaluated according to the factors listed in the conceptual framework.

Apache Web Server 2.0.48

The Apache Web Server product, as an open source application, had a straightforward and user-friendly setup wizard that required some knowledge of basic website concepts, such as root directory and ports, that were easy enough to configure and supplied default values to aid in the installation. Based on ESRI's setup notes, setting up Apache to configure with ArcIMS did take some time due to the complexity of the procedure, but the instructions were clear and easy to understand. The performance criteria ratings for Apache Web Server 2.0.48 are shown in Table 3.

Table 3. Performance Criteria Ratings for Apache Web Server 2.0.48

Criterion	***Comments***	***Rating***
1	The Apache Web Server conforms to basic W3C standards for data transfer over the web (most notably HTTP) and many more standards that would be too many to list here.	**5**
2	Apache also runs on 31 operating systems (Apache Software Foundation 2005).	**5**
3	Apache supports any kind of application server, so depending on the application server used, server-side processing may be available	**5**
5 - Web Server	Since stress testing measures of the website could not be adequately performed, the processing and memory management considerations for Apache can only be based on minimal client requests. Apache did perform well in a non-stressed HTTP environment. Setup was easy and did not take long, but the configuration with ArcIMS did take some time. Security features could not be tested, but Apache allows for HTTP security. Apache also provided a detailed help file and a user community for support	**4**

The accessibility and economy criteria results are as follows.

1) For minimum system requirements as it relates to this configuration, Apache supports the NT kernel, which includes Windows NT, 2000, XP and Server 2003 operating systems and also supports 9x family of Windows operating systems to a limited degree. It supports x86 family of processors, which are the de facto standard processor of any machine running a version of the Windows NT kernel (Apache Software Foundation 2005). Apache also supports 31 different operating systems as of March 2, 2006, both open source and proprietary (Apache Software Foundation 2005). As far as licensing goes, it operates under the General Public License (GPL) which allows for redistribution of modified source code, but without profit to the modifying user or with its own restrictions on the use of said modified source code. Overall, Apache supports a wide variety of operating systems and has a flexible license agreement that gives it an advantage in the web server market.
2) Total Cost of Ownership
    a. Hardware costs for minimal system configuration: $729.00
        i. Costs are based on a Dell system configuration with minimum required features (Dell, Inc. 2006).
    b. Software Licensing: $0
    c. On-going maintenance: $0
3) Apache has a detailed help file for reference on any issue with the product and provides mailing lists, bug reports and a host of third-party websites that provide support for the Apache Web Server product.

4) There are no costs for obtaining support directly from Apache, but costs may accrue when accessing a third party.

Time considerations for Apache are shown in Table 4.

Apache Tomcat 4.1.29

The Apache Tomcat product, also a member of the Apache Software Foundation, provides the application logic layer in the proprietary configuration. It utilizes the Java Software Development Kit (SDK) for its object model and execution environment. Tomcat is designed to be configured with Apache to provide java-based application development through the HTTP protocol. The installation and configuration of Tomcat did get a little complex, but the directions from ESRI were understandable to configure a working system. The installation did not take long and it was similar in the look and feel of the Apache install. The performance criteria ratings for Tomcat are shown in Table 5.

Table 4. Time Considerations for Apache Web Server 2.0.48

Number	***Criterion***	***Time (hrs)***
1	Average time (in hours) taken to install the product, including requirements gathering and the physical installation of the product	**2**
2	Average time (in hours) taken to make full use of the product once the entire WBGIS is configured and deployed based on its function in the system	**2**

Table 5. Performance Criteria Ratings for Tomcat 4.1.29

Criterion	*Comments*	*Rating*
1	Tomcat supports all of the W3C standards that Apache supports and it is designed to work alongside it, making it versatile in this configuration.	**5**
2	Tomcat operates on any operating system that supports the cross-platform language Java. There is only one file option for downloading Tomcat, since it runs based on the Java Runtime Environment installed on the machine (which is free to download from Sun Microsystems), so wherever the JRE is present, Tomcat will work.	**5**
3	Since Tomcat works in the Java Runtime Environment, it has the ability to perform server-side processing through Java Server Pages and other Servlet technologies. It is not, however as advanced in the server-side processing area as Microsoft's ASP.NET, which compiles and executes all code on the server and sends static HTML to the client, making full use of server-side processing.	**4**
6 - Application Server	Being an Application Server in this configuration, Tomcat provides support for the Java language, a high-level programming language that provides a multitude of functionality options for a spatial website. The Java Runtime Environment manages compilation and execution of code on the server and ArcIMS has components designed to be able to access Tomcat's services. Sun Microsystems, Inc. is the chief proprietor of Java technology and provides a wealth of documentation on their website about customization with Java Server Pages and Tomcat also provides lots of support in the form of documentation, mailing lists, bug reports and third party options.	**5**

Accessibility and Economy considerations for Tomcat are as follows.

1) Tomcat is bound to the same license agreement as the Apache Web Server; therefore, the source code is fully available for modification and redistribution of modifications. Since Tomcat is a Java-based application, and Java is a cross-platform language, virtually any operating system can support the installation of the product.
2) Total Cost of Ownership

a. Hardware costs for minimal system configuration: $729 (Dell, Inc. 2006)

b. Software Licensing: $0

c. On-going maintenance: $0

3) Much like Apache Web Server, Tomcat has a detailed help file and a strong user community of support through bug lists, mailing lists and online discussion boards. Certain other help and support may be obtained from Sun Microsystems, Inc., chief proprietor of the Java language.

4) There is no cost for obtaining online help from the user community of Tomcat either through mailing lists or the like, but cost may be accrued when seeking help from Sun Microsystems or another third-party organization.

Time considerations for Tomcat are shown in Table 6.

ArcIMS 9.0

ESRI's main product for distributing spatial data over the web, ArcIMS, is a fully-functional software package that provides more than just a map server. It provides other kinds of spatial data services, including feature services, metadata services and

Table 6. Time Considerations for Apache Tomcat 4.1.29

Number	***Criterion***	***Time (hrs)***
1	Average time (in hours) taken to install the product, including requirements gathering and the physical installation of the product	**2**
2	Average time (in hours) taken to make full use of the product once the entire WBGIS is configured and deployed based on its function in the system	**2**

routing and geocoding services. For the scope of this research, the functionality of the map output through data stored as shapefiles was the primary area of focus. ArcIMS produces relatively clear vector and raster data output and allows for a rich user experience with the data, providing the ability to interact with features on the map through query and identify, as well as advanced tools like buffering and shape drawing. Since the proprietary configuration does not make use of a backend database server, ArcIMS in this configuration serves as both a database server and a map server; therefore, both criteria will be judged in this section. The performance criteria ratings for ArcIMS are shown in Table 7.

Table 7. Performance Criteria Ratings for ArcIMS 9.0

Criterion	***Comments***	***Rating***
1	ArcIMS supports all of the crucial W3C components that only require an updated browser and no plug-ins if using the standard HTML viewer. As far as OGC standards, ArcIMS has implemented the OGC Web Map Service 1.1.1, and is implementing Web Feature Service 1.0, Styled Layer Descriptor Implementation Specification 1.0, Geography Markup Language (GML) Encoding Specification 3.0, GML 2.1.2 and 2.1.1 and Filter Encoding 1.0 (OGC 2006). While ArcIMS has only been compliant with one OGC specification, more are on the way.	**3**
2	ArcIMS runs on the following operating systems: HP-UX 11.i, IBM AIX 5.1, 5.2, 5.3, Red Hat Enterprise Linux AS/ES 3.0, SUSE Linux Enterprise Server 9, Windows 2000, Professional, Server and Advanced Server, Windows Server 2003 Standard, Enterprise and Small Business, Windows XP Professional Edition and Sun Solaris 8, 9 and 10 (ESRI Support 2006). The only supported web browsers are Internet Explorer 6.x and Netscape Communicator 7.1. Mozilla Firefox is not supported.	**4**

Table 7. Continued

Criterion	*Comments*	*Rating*
3	ArcIMS supports server technologies that can aid in the delivery of rich experiences to thin clients. Technologies like Microsoft's ASP and ASP.NET technologies as well as Sun's Java Server Pages can aid in this effort. ESRI's support website makes note that for clients accessing an HTML viewer, only 64 MB of RAM is recommneded, and for Java viewers 128 MB of RAM is recommended. However, for all components offered by the ArcIMS server, 256 MB of RAM is recommended. So for some thin clients that do not have that much RAM, having access to all of the functionality of ArcIMS will not be as rich as clients that have the minimum amount of RAM specified here (ESRI Support 2006).	**3**
4 – Database Server	From a database server standpoint, ArcIMS in this configuration supports only the storage of file-based spatial data through shapefiles and various image formats. The list of supported raster datasets is quite impressive (ESRI Support 2003).	**4**
7 – Map Server	From a map server standpoint, ArcIMS does provide for quality output of spatial data and has an impressive list of functionality built into the product out of the box, but because of the size and complexity of the application has a lengthy and intricate installation process. Based on the experience installing different configurations of ArcIMS, the user performing the install should be technically savvy to understand the installation instructions. This install would be difficult to the lay user. The different types of servers built into ArcIMS (image, feature, etc.) provide for a rich GIS experience and allow lots of customization options for the user. The support for server technologies also aids application developers in developing customized websites implementing their own security and unique graphical user interfaces (GUI). ESRI's support for ArcIMS is extensive and thorough. The online support website details the installation process very specifically and accurately. Support is also in the form of phone and fax and it is easy to know where to access other forms of support. Overall, as a map server, the quality and performance of the product is excellent, but the installation and configuration of the entire system can prove difficult and time consuming to users new to ArcIMS.	**5**

Accessibility and Economy considerations for ArcIMS are as follows.

1) For this research, ArcIMS was installed on a Windows XP operating system with a 2.8 GHz Intel Celeron processor with 512 MB of RAM and 20 GB of hard drive space. The application was able to install under these conditions. The minimum system requirements that ESRI recommends for ArcIMS are as follows.
    a. CPU Speed: 1.0 GHz or higher
    b. Processor: Intel Pentium or Intel Xeon
    c. Memory/RAM: 256 MB RAM for all components
    d. Disk Space: 914 MB for complete installation (ESRI Support 2006)

   The operating systems supported were listed previously and encompass a broad range. Overall, the hardware and software restrictions for ArcIMS are not so much that they would alienate any users that cannot afford the minimum hardware and software needed to run the system. It seems that the minimum requirements for ArcIMS to run are relatively affordable, but as with any server software product, the higher the performance of the hardware, the higher the performance of the software.

2) Total Cost of Ownership
    a. Hardware costs for minimal system configuration: $729 (Dell, Inc. 2006)
    b. The minimum costs for software licensing and materials are $10,000. This price is good for installation on a single machine with up to two processors working concurrently to serve the data (personal phone call from ESRI Sales March 20, 2006, unreferenced).

c. On-going maintenance and licensing fees: $5,000 per additional processor per machine and additional costs for yearly license upgrades based on the contract with ESRI (personal phone call from ESRI Sales March 20, 2006, unreferenced).

3) With an ESRI subscription and a service contract, phone support is available for any issue with the software as well as online support. It is easy to obtain support for ArcIMS either through their website or by phone.

4) Costs associated with obtaining help other than product documentation will be based on the support contract and/or the product licensing agreement.

Time considerations for ArcIMS are shown in Table 8.

Performance Criteria Ratings for the Complete System

Table 9 shows the complete proprietary system evaluation results for the nine criteria mentioned in the conceptual framework.

Table 8. Time considerations for ArcIMS 9.0

Number	***Criterion***	***Time (hrs)***
1	Average time (in hours) taken to install the product, including requirements gathering and the physical installation of the product (average between 7 hours for the first installation and 5 hours of the second installation)	**6**
2	Average time (in hours) taken to make full use of the product once the entire WBGIS is configured and deployed based on its function in the system	**2**

Table 9. Proprietary Configuration Results

Total Time to setup and deploy complete system	*9.5 for complete configuration (16.5 hours total for first & second)*
Total Cost of Ownership	*$15,729 (hardware & software)*
Number of Open Standards	*Extensive W3C, 8 OGC*
Number of Operating Systems	*12*
Server-side scripting	*Yes*
DBMS Rating	*4*
Web Server Rating	*4*
Application Server Rating	*5*
Map Server Rating	*5*

Open Source Configuration Results

For the open source configuration, the packaged product MapServer for Windows (MS4W), put together by DM Solutions Group (2006,) turned out to be a quick and efficient way to get a map service up and running. The packaged product contains Apache HTTP Server 2.0.55, PHP 4.4.3, MapServer 4.4/4.6 and a host of other utilities for spatial data accessibility and for customization through scripting. This configuration also utilized two types of data stores: data stored in shapefiles with the dBASE Database Management System and the PostgreSQL database server with the PostGIS extension. Both were evaluated as database servers whether or not they worked in the configuration. The steps taken in the installation process of MS4W are summarized in Table 10.

Table 10. Installation and Setup Log for MapServer for Windows

Dates	Hours	Task	Description
1/7/2006	2	Install MapServer	Installed MS4W from MapTools.org, Chameleon, MapLab, OGC Workshop apps
1/7/2006	1	Install PostgreSQL/PostGIS	Installed PostgreSQL/PostGIS version 8.1
2/20/2006 - 3/11/2006	6	Create First Map File	Use MapLab to create a map file from the CSPDC shapefiles with the same data from the AXL file and with custom symbology and labeling
3/12/2006	<1	Run and test Final Map File	Use MapBrowser to view the final map file and test all the available functionality
Total Hours	10		

The installation of MS4W was extremely simple and took up much less time than the complete installation of ArcIMS. The MS4W website also contained detailed installation instructions that aided the first time user efficiently. The DM Solutions Group and other contributing developers have created other packaged applications that can run on this preconfigured group of applications known as MS4W, which are listed on the downloads page of the MS4W site. The application MapLab, which is an open source project specifically designed to work with the MS4W installation, provides an intuitive and graphically pleasing environment for creating and editing MapServer .MAP files, which are the main configuration files for spatial data served through MapServer, much like ArcIMS' .AXL file. Some research had to be done with learning the syntax of map files to become familiar with basic concepts, but MapLab allows a user to create a map file through a graphical user interface with much ease and precision.

Three tools are available within MapLab: MapEdit for visually editing map configuration files, MapBrowser, for viewing the completed map, and GMapFactory for creating custom mapping applications with PHP and MapScript. For this research, MapEdit and MapBrowser are the two primary tools that are utilized to create a basic map configuration file and preview the data through the browser. Based on the installation and setup log, the map file creation process was quite quick and efficient after an initial learning process about the map configuration file syntax and getting used to the MapEdit interface. A map configuration file is a text file with certain keywords that pertain to settings for the map. The MapEdit interface allows for a graphical editing environment for the map file.

The map file was named CSPDC.MAP and the settings for the selected element appear in the main panel of the window (in this case settings for the parent "map" element) with a tree view of the main configuration settings on the left panel. Each layer in the map file is specified as a separate entry, each with its own CLASS element that contains configuration information for labeling and style symbology for the layer. This map file is the most basic setup for the two shapefiles that were used in the ArcIMS configuration. Other layers could easily be added by using the tools provided in MapEdit.

MapBrowser, the primary tool for viewing completed map files, can be accessed by clicking on the third icon from the far left of the top navigation bar. The structure of the navigation bar makes it easy to switch between view and edit modes. Once the map file changes are made, the map file can be viewed by simply clicking on the MapBrowser icon. MapBrowser provides a clean graphical view of the completed map file and provides a set of zoom tools within the map view for zooming in, out and to full extent, zoom to custom view (which is set under "Current Map View"), recenter and identify. The zooming tools work pretty well as long as the minimum and maximum extents are set properly in the WEB element of the map configuration file. It was difficult to effectively utilize the identify tool, however, for an unknown reason.

Performance Criteria Results for Individual Components

The individual software components that make up the open source configuration were evaluated on the factors listed in the conceptual framework.

Apache Web Server 2.0.55

In this revision of the Apache Web Server, as compared to the version used in the proprietary configuration, there was minimal difference between the two performance wise other than that there were a few times where reloading the page while in MapLab sometimes took some time. It was likely due to the amount of available memory and processing power on the server computer, which was the laptop mentioned under the accessibility and economy considerations for ArcIMS. The setup process for the product was completely behind the scenes, since the entire MS4W installation was a packaged installation that ran as a single batch file. This particular configuration utilized this batch file to create a simple one-click installation. There is an option to install it separately and then configure the product later to work with MapServer, but this was unnecessary since the batch file took care of the installation and configuration all at once. The Performance Criteria Ratings for Apache Web Server 2.0.55 is summarized in Table 11.

Table 11. Performance Criteria Ratings for Apache Web Server 2.0.55

Criterion	***Comments***	***Rating***
1	The Apache Web Server conforms to the most basic W3C standards for data transfer over the web (most notably HTTP) and many more standards that would be too many to list here.	**5**
2	Apache also runs on 31 operating systems (Apache Software Foundation 2005).	**5**
3	Apache supports any kind of application server, so depending on which application server is used, server-side processing may be available	**5**
5 - Web Server	Since stress testing measures of the website could not be adequately performed, the processing and memory management considerations for Apache can only be based on minimal client requests. Apache performed well in a non-stressed HTTP environment. Setup was completely behind the scenes and took much less time than the setup of the other version used with ArcIMS. The only issues noted were some page refresh issues while working in MapLab. Sometimes the page had to be repeatedly refreshed when making changes to the map file and sometimes while browsing the data in MapBrowser.	**4**

The Accessibility and Economy considerations for Apache 2.0.55 are as follows.

1) Since the license agreement and other restrictions have not changed for this version of Apache, the considerations for the product have not changed as well compared to the version used in the proprietary configuration.
2) Total Cost of Ownership.
    a. Hardware costs for minimal system configuration are still based on the minimum required features desirable in the hypothetical Dell computer configuration: $729.00 (Dell, Inc. 2006)
    b. Software Licensing: $0
    c. On-going maintenance: $0

3) This version of Apache has a detailed help file for reference that comes with the packaged product and contains all the other resources that the other version offers..
4) There are no costs for obtaining support directly from Apache, but costs may accrue when accessing a third party.

Time considerations for Apache are summarized in Table 12.

PHP Application Server 4.4.3

PHP, a server technology that allows for rapid application development of interactive websites, is the chief application logic layer in the open source configuration. PHP provides the runtime environment for all functionality employed on any running application of MS4W. The installation of PHP is also in the background of the batch file, so no actual setup wizard appears for the installation. After installing the complete system, the MapLab application ran pretty well except for a few errors that would occur now and then if the user gave a bad command, but this is merely the fault of the actual coding of the application and not the application server. Overall, PHP provides a rich library of functionality that can be embedded in a spatial website with MapServer. The performance criteria ratings for the PHP 4.4.3 are summarized in Table 13.

Table 12. Time Considerations for Apache Web Server 2.0.55

Number	Criterion	Time (hrs)
1	Average time (in hours) taken to install the product, including requirements gathering and the physical installation of the product	**2**
2	Average time (in hours) taken to make full use of the product once the entire WBGIS is configured and deployed based on its function in the system	**<1**

Accessibility and economy considerations for PHP 4.4.3 are as follows.

1) PHP does not impose many huge restrictions on use of the software. It is open source and free to use and redistribute modified copies of the software, but no warranties are offered for the product.

Table 13. Performance Criteria Ratings for PHP 4.4.3

Criterion	Comments	Rating
1	PHP supports all W3C standards that are relevant to dynamic websites. As far as OGC standards go, PHP is able to plug into any ODBC-compliant database management system that has an ODBC driver and it also supports the handling of common spatial data formats (shapefiles)	**5**
2	PHP can run on most UNIX, Linux, Mac OS and Windows systems (PHP Group 2006). PHP is supported in both Internet Explorer and Mozilla Firefox	**4**
3	PHP, by its very acronym implies server-side execution since it stands for Hypertext Pre-Processor, meaning that application code is pre-processed before results are sent to the browsing client. PHP definitely makes as most use of server-side scripting possible.	**5**
6 - Application Server	PHP, as an application server provides a fully developed object oriented scripting language and runtime environment for developing sophisticated web applications. It provides for its own memory management and thread execution and adequate documentation for the product is available from the PHP Group (2006) and through many other websites, vendors and mailing lists.	**5**

2) Total Cost of Ownership
    a. Software Licensing: $0
    b. On-going maintenance: $0
3) Help for PHP users is in the form of detailed help files that come with the installation of the product, many third party organizations that provide web-based examples and some that charge money for services, as well as mailing lists and weblogs for collaboration.
4) The only costs that may accrue for obtaining help would be through the PHP group or another third-party organization.

The time considerations for PHP 4.4.3 are summarized in Table 14.

MapServer 4.4/4.6

The MapServer product, originally developed by the University of Minnesota, is the map server layer in the open source configuration. It is responsible for receiving requests from Apache and sending responses for data viewing in the format of a particular image type to the client. PHP aids MapServer by providing the logic for interpreting some client requests, but primarily MapServer is responsible for the data repository. MapServer can connect to multiple types of data stores, including data stored as shapefiles and data stored in database servers, like PostgreSQL with the PostGIS extension. This configuration utilized two types of data stores: data stored in shapefiles using the dBASE Database Management System and data stored in the PostgreSQL

Table 14. Time Considerations for PHP 4.4.3

Number	***Criterion***	***Time (hrs)***
1	Average time (in hours) taken to install the product, including requirements gathering and the physical installation of the product	**2**
2	Average time (in hours) taken to make full use of the product once the entire WBGIS is configured and deployed based on its function in the system	**<1**

database with the PostGIS extension for data access with MapServer. The shapefile configuration is the most simple to accomplish because all that is required is a valid shapefile, which is comprised of only three separate files. The PostgreSQL configuration is more difficult to accomplish because you have to install the complete database server and any associated tools and utilities for configuring the system, and then you have to use SQL to import the data from the source files in order to use them with MapServer. The results which follow analyze whether both types of data stores worked with MapServer or not. The performance criteria ratings for MapServer 4.4/4.6 are summarized in Table 15.

Table 15. Performance Criteria Ratings for MapServer 4.4/4.6

Criterion	***Comments***	***Rating***
1	MapServer conforms to all of the W3C standards that are compatible with Apache. MapServer is also implementing the following OGC standards: Web Map Service 1.1, Web Map Service 1.0, Web Map Context Documents 1.0, Web Feature Service 1.0, OpenGIS® Styled Layer Descriptor (SLD) Implementation Specification 1.0, Geography Markup Language 2.0, Filter Encoding 1.0, Web Map Service 1.1.1 (OGC 2006).	**3**
2	MapServer can run on 4 flavors of Linux (Verbose, SuSe, Debian and RedHat), Windows 2000, XP and 2003, Mac OS X, FreeBSD and most versions of UNIX operating systems and works in Internet Explorer and Mozilla Firefox (Regents of the University of Minnesota 2006).	**5**
3	MapServer makes use of server-side processing by interpreting requests sent in by Apache and with the help of PHP sends image data back to the client for viewing. It operates exclusively on the server, sending only the resulting image of the map to the client.	**5**
7 – Map Server	MapServer provides high quality vector and raster output, supports any map projection and coordinate system, allows for basic viewing functionality, multiple data layer support, allows for extensibility through PHP and supports common spatial data formats. The one major drawback to MapServer is the lack of spatial analysis functionality that is present in the most current version. MapServer's primary role is just that: serving maps, so the spatial analysis functionality that is built into it is limited as compared to a product like ArcIMS that is more of a full-featured GIS. MapServer allows for querying and identifying, however, which shows that it leans towards full GIS functionality. There are other products that can provide a mapping application with more sophisticated GIS functionality, such as PostGIS with some of its spatial query functionality. While using MapBrowser in this configuration, the identify functionality does not seem to work, but it is likely to do with the setup of the map configuration file rather than the MapServer executable itself. The setup of the product was easy due to the batch file, but without technical savvy knowledge of compiling programs, installing MapServer from source could prove difficult. Some binary distributions of MapServer are easier to install than compiling from source.	**4**

Accessibility and economy considerations for MapServer are as follows.

1) There are no restrictions on the use of MapServer that make it unlike any other open source product. The minimum system requirements do not limit users who wish to deploy a web mapping application on a computer with enough performance power to be an effective server. MapServer's minimum requirements are definitely low enough to accommodate many obsolete systems.
2) Total Cost of Ownership
    a. Software Licensing: $0
    b. On-going maintenance: $0
3) From the user communities to the University of Minnesota, there are multiple ways of obtaining help and support outside of the detailed documentation and help file that come with any downloaded release. Many corporations provide consulting for a fee for the MapServer product. DM Solutions Group is one example.
4) No costs for obtaining basic help and documentation through help file, some websites and mailing lists, but costs may accrue depending on the policy of the third party delivering the consulting service.

The time considerations for MapServer 4.4/4.6 are summarized in Table 16.

Table 16. Time Considerations for MapServer 4.4/4.6

Number	***Criterion***	***Time (hrs)***
1	Average time (in hours) taken to install the product, including requirements gathering and the physical installation of the product	**2**
2	Average time (in hours) taken to make full use of the product once the entire WBGIS is configured and deployed based on its function in the system	**<1**

dBASE Database Management System

As the shapefile's data storage format for attribute data, the dBASE management system allows for a basic data structure with a high degree of portability amongst other database management systems. While not much is known about the internal functionality of the proprietary dBASE format used by ESRI's shapefile, the data stored in a shapefile's associated .dbf file can be ported to many other applications, including Microsoft Excel. The PostgreSQL database management system with PostGIS provides utilities for importing data from shapefiles as well, so there are ways of effectively utilizing the shapefile format in this configuration. MapServer utilizes the shapefile as the primary data store, so it is definitely compatible with the dBASE format. Since the dBASE format is not a full featured application, for this analysis the focus was only on the database server considerations. Those considerations for dBASE are summarized in Table 17.

PostgreSQL/PostGIS 8.1

As the secondary DBMS for this configuration, PostgreSQL provides a complete enterprise database management system completely open source with advanced security features, transaction handling and efficient data storage and access. In a production

Table 17. Database Server considerations for dBASE

Function	***Criterion***	***Time (hrs)***
Database Server	dBASE is a Relational Database Management System that enforces data integrity, security and reliability through primary keys and indexing, but is not a full featured database management system in the sense that it stores data in a native SQL format like a database server product like PostgreSQL. There are a many features that are not included in the dBASE format because it cannot run as a process on a computer. It is primarily a static data file format that can be accessed by other database systems but does not provide for transaction processing or many security features available in a full featured database management system. Also, since the same shapefiles are used in both configurations, the same ratings are given for both the basic database server considerations since they both rely on the same database file format.	**4**

environment and for applications that require many users concurrently accessing data, it is recommended to utilize a DBMS like PostgreSQL for handling data transactions. This software is installed as a service on the server machine and talks to MapServer via PostGIS for processing data requests. The performance criteria ratings for PostgreSQL/PostGIS are summarized in Table 18.

Accessibility and economy considerations for PostgreSQL/PostGIS are as follows.

1) PostgreSQL and PostGIS are both released under the GNU General Public license, so both allow for alteration of source code and redistribution of modified source code. The only main restrictions are the types of operating systems that it supports.

Table 18. Performance Criteria Ratings for PostgreSQL/PostGIS 8.1

Criterion	***Comments***	***Rating***
1	PostgreSQL supports the essential W3C standards that allow for efficient communication with the Apache Web Server. PostGIS also implements the following OGC standards: OpenGIS® Simple Features Implementation Specification for SQL 1.1 and Simple Features - SQL - Types and Functions 1.1 (OGC Resources 2006).	**4**
2	PostgreSQL/PostGIS can be installed on AIX, Windows 2000, XP and 2003, HP-UX, IRIX, QNX, SCO and Solaris operating systems (PostgreSQL Global Development Group 2006).	**5**
3	All processing done by PostgreSQL and PostGIS is executed on the server	**5**
4 - Database Server	PostgreSQL/PostGIS support a full featured relational database management system with multiple transaction handling, security features and utilize the SQL language for storage and manipulation of data. Other utilities for accessing spatial data formats are available for converting into PostgreSQL and lots of helpful documentation and help files accompany the installation of the product. The PostGIS product is developed by Refractions Research, so support and documentation come from this organization as well (Refractions Research 2005). They develop the product free of charge, but only charge clients for the use of their consulting services.	**5**

2) Total Cost of Ownership
    a. Software Licensing: $0
    b. On-going maintenance: $0
3) Detailed help files come with the installation and many user groups, websites and mailing lists offer other forms of support. Support is also available from Refractions Research for PostGIS and other third party organizations.
4) Costs may accrue when working with other third parties such as Refractions Research.

Time considerations for PostgreSQL/PostGIS are summarized in Table 19.

Performance Criteria Ratings for the Complete System

Table 20 shows the complete open source system evaluation results for the nine criteria mentioned in the conceptual framework. The DBMS rating is for the dBASE system only.

Discussion

Based on the results of both configurations, there are definite advantages and disadvantages to both approaches both financially and functionally. Some basic factors to consider in this part of the analysis are the total time to implement the completed system in hours and the total cost of ownership. The other factors in the analysis are arbitrarily decided based upon the experience of installing and using the software.

To summarize all of the considerations for both configurations, Table 21 shows the results from both configurations.

Table 19. Time Considerations for PostgreSQL/PostGIS 8.1

Number	***Criterion***	***Time (hrs)***
1	Average time (in hours) taken to install the product, including requirements gathering and the physical installation of the product	**1**
2	Average time (in hours) taken to make full use of the product once the entire WBGIS is configured and deployed based on its function in the system	**<1**

Table 20. Open Source Configuration Results

Total Time to setup and deploy complete system	*<10 hours*
Total Cost of Ownership	*$729 (hardware only)*
Number of Open Standards	*Extensive W3C, 10 OGC*
Number of Operating Systems	*15*
Server-side scripting	*Yes*
DBMS Rating (dBASE)	*4*
Web Server Rating	*5*
Application Server Rating	*5*
Map Server Rating	*4*

Table 21. Results from Both Configurations

Considerations	**Open Source Configuration**	**Proprietary Configuration**
Total Time to setup and deploy complete system	***<10 hours***	***9.5 hours***
Total Cost of Ownership	*$729 (hardware only)*	*$15,729 (hardware & software)*
Number of Open Standards	*Extensive W3C, 10 OGC*	*Extensive W3C, 8 OGC*
Number of Operating Systems	*15*	*12*
Server-side scripting	*Yes*	*Yes*
DBMS Rating	*4*	*4*
Web Server Rating	*4*	*4*
Application Server Rating	*5*	*5*
Map Server Rating	*4*	*5*

First, based on time considerations, ArcIMS had a steep learning curve and a detailed and complex installation process that was a lot to get used to in the first installation attempt. However, had Tomcat or a licensed copy of ServletExec been used from the beginning, the time to setup would have been 5 hours shorter. So, since this rework had nothing to do with the learning curve, the time to complete the original configuration with ServletExec is disregarded. The installation process for MS4W was so simple that no research had to be done whatsoever before the installation. This is a definite advantage in the open source configuration. However, once MS4W was installed, the process of creating a map file was time consuming. It is crucial to know the basics of map file syntax in order to be able to understand the MapEdit interface. All of the properties and settings in MapEdit are directly reflected in the map file. The maximum extents of the data have to be manually set in the map file, for example, and without knowledge of where those maximum extents are set, the map file will not display the data correctly upon loading. This is the main reason why the total time to setup the open source system is almost as much as the proprietary.

Second, the financial aspect of the open source configuration is obviously the biggest advantage that the system has going for it as opposed to the proprietary configuration. It costs no money whatsoever to set up a complete web based mapping application and low to no costs for support and documentation. The proprietary configuration requires a steep amount of money for software licensing as well as for on-going maintenance and support.

For both configurations, the number of OGC and W3C specifications that are implemented and supported are impressive. It appears that there are slightly more OGC specifications in the open source configuration that are mainly attributed to the two OGC specifications that PostGIS conforms to. Both systems make good use of these specifications.

The number of operating systems that both configurations support are also quite impressive. There are a few more distributions of Linux systems that the open source configuration supports, but overall, both configurations provide support for a broad number of operating systems.

Regarding the four ratings, technically the proprietary configuration wins with 18/20, with the open source at 17/20, but both configurations excel in some areas and do not in others. The proprietary configuration excels in functionality and map output. The map server component provides for a much wider array of options, particularly labeling, feature rendering, querying and spatial analysis functionality. MapServer inherently is not built to support much of the spatial analysis functionality that ArcIMS has built in. The application server for ArcIMS also excels at providing the functionality for most of the GIS-related tasks. ArcIMS also provides for many more types of services than MapServer does alone. It allows for feature streaming services, image services and metadata services, among others. MapServer only provides static map output in an image format. However, due to the data format used in the proprietary configuration, the rating was less than perfect only because all of the considerations for an ideal database system were not met in this format, and this is reflected as well in the open source configuration. The same goes for the web server product in both configurations; since both could not be tested in a networked environment and some security features could not be utilized, the lower rating was applied because full use could not be made of it.

From the open source standpoint, MS4W excelled in providing dynamic application support and provides for clean vector and raster output. The main reason why the map server component was given a rating of 4 was because of certain issues with labeling and symbology of the datasets. While MapServer allows for feature labeling, there were some difficulties in getting each unique label to show up in the current map extent rather than labeling each segment of the roads shapefile. Multiple map labels of the same feature would appear when viewing the roads shapefile instead of only showing a single label for the entire feature. ArcIMS has some way of putting fewer labels in the map view than MapServer did, which is why the rating is lower then perfect.

Ultimately, based on these experimentations, there are feasible options in both the open source and the proprietary realms. The determining factors for choosing particular products stem from the role that each component plays in the system. It is wise to choose components that excel in the areas that the system needs to excel in and that work together with the rest of the components to achieve the goal of the application. Table 22 summarizes the software components that make up both of the configurations for comparison with a previous study done by Raghavan et. al (2002).

Based on the information in Table 22, major differences can be seen between the open source and proprietary options and between both of the options implemented in this study with those from another open source study from Raghavan et. al. The open source configuration does not have a GIS server for serving spatial analysis functionality over the web, unlike the proprietary and the previous open source study. This is a major drawback to solely using MapServer to build a comprehensive web-based GIS application. GRASS has similar abilities to the vast array of GIS services ArcIMS offers, making it closer to a fully-functional web-based GIS.

Table 22. Software Components for Individual Studies Compared

	Open Source Configuration	Proprietary Configuration	Raghavan et. al (2002) Open Source Configuration
Web Server	*Apache HTTP Server 2.0.55*	*Apache HTTP Server 2.0.48*	*Apache HTTP Server*
Map Server	*MapServer 4.4/4.6*	*ArcIMS 9.0*	*GRASS with GRASSLinks*
Database Server	*dBASE (Shapefiles)*	*dBASE (Shapefiles)*	*PostgreSQL/PostGIS*
Application Server	*PHP 4.4.0*	*Apache Tomcat with Java 2 SDK*	*Common Gateway Interface (CGI) & GRASSLinks*
GIS Server	*None*	*ArcIMS Spatial Server, Application Server, Feature Server, Image Server, Metadata Server, ArcMap Server*	*GRASS*

Limitations

Limitations of this research encompass mainly the inability to adequately test the ArcIMS and MapServer web applications over a real intranet or Internet to test the speed and overall performance over a network. Because of this limitation, the alternative was to center on the overall usability issues of the individual software components used in both the proprietary and open source configurations, with testing of the websites through a local machine connection. This means that the web server and the client were located on the same machine for testing purposes. At this level, the applications can be run to evaluate basic performance, such as functionality of map tools, map refresh rates, map output quality and other factors. Another large drawback to the open source configuration was the lack of spatial analysis capabilities built into MapServer. Utilizing a product like GRASS would extend GIS capabilities to a higher degree.

Other limitations of this study include the inability to test and experiment with all other open source products that exist for producing WBGIS applications. MapServer is not the only product for serving GIS data and maps and it would be ideal if every product could be evaluated and tested for the purposes of comparison. Based on time considerations, evaluating multiple open source software components in order to show a more broad look at open source options was not possible. Another limitation was the inability to quickly setup the PostgreSQL/PostGIS environment for serving spatial data to MapServer. The initial installation of the product was simple and quick, but more time and research needed to be dedicated to setting up PostGIS to work with MapServer, creating spatial tables and importing data into those spatial tables from the source shapefiles. Because this system could not be utilized in time, the primary way for storing data was done through shapefiles.

Chapter 5: Conclusion

This study is significant in that it attempts to bridge the knowledge gap between the real pros and cons of proprietary and open source software through concrete comparison and analysis of certain critical factors related to overall performance. Current literature solely dedicated in this direction is difficult to find. The topic presented in this study strives to orient future research in the field towards the awareness of open source products within the overall realm of the WBGIS software market.

Key concepts related to this study include:

- Distinction between proprietary and open source software
- Factors for judging performance of a Web-Based GIS Application
- The relationship between market share and overall ease of use of a WBGIS software solution
- Time and financial resources in WBGIS application development
- Security, stability, scalability and maintainability of a WBGIS software solution and/or configuration
- The importance of Open Standards as they relate to system interoperability and how geographic data is exchanged.

The objectives of this research were to effectively compare two WBGIS systems built with open source and proprietary tools to identify criteria that can be used to evaluate individual software components for constructing a web based GIS application. Based on the results of the analysis, open source tools have proven to be able to provide robust and quality output for mapping and GIS applications over the web. When considering software tools to use in building a WBGIS, open source should definitely be a consideration.

This research attempts to contribute to the WBGIS field as a whole and to more formally address the issues of open source tools and their role in the WBGIS field that has mostly been dominated with successful proprietary tools. This study could potentially contribute to such research areas as Public Participation in GIS (PPGIS), GIS in K-12 education and GIS for any purpose where there are significant financial barriers.

For organizations faced with the decision to implement a web-based GIS system with either open source or proprietary tools and are either new to the field or do not have the technical expertise, there are several options for gaining more information.

- Consult with other organizations that have implemented their own WBGIS systems to see how they benefit from their particular configuration.
- Obtain current literature on the topic of open source GIS and mapping, such as Mitchell's Web Mapping Illustrated (2005) that provide a comprehensive discussion from a beginner's viewpoint on web mapping tools.
- Contact professional organizations, software vendors and contracting companies that provide GIS services to hear their viewpoint on the subject and to find out what kinds of products and services they offer that can benefit the organization's needs.
- Look for example applications on the Internet, previous research, trade magazines and other publications to obtain different perspectives and to learn about successes and failures of current options for building spatial websites.

Suggested areas for further research and study encompass the vast array of open source GIS applications for publishing maps, data and metadata on the web. There are countless other software products in the open source world that perform a wide variety of GIS and mapping tasks (OpenSourceGIS.org 2006). Research should be conducted more in the alternative means for producing GIS web applications with the focus more on spatial analysis functionality. Lots of alternative open source software and freeware exist that attempt to offer more advanced spatial functionality and visualization techniques that should be researched to determine other feasible options for constructing spatial websites.

Appendix A

Figure 4 References:

Internet Explorer 6 Logo Copyright © 1995-2004 Microsoft Corporation.
Blue Globe Image Available from: http://grant.henninger.name/images/.
Sun and Java Logo Copyright © Sun Microsystems, Inc.
Apache Tomcat Logo Copyright © 1999-2006, The Apache Software Foundation.
Apache Web Server Logo Copyright © 1999-2006, The Apache Software Foundation.
ArcIMS Logo Copyright © 1995-2006 ESRI. All rights reserved.

Figure 5 References:

Internet Explorer 6 Logo Copyright © 1995-2004 Microsoft Corporation.
Mozilla Firefox Logo Copyright © 2005–2006 Mozilla Corporation. All rights reserved.
Blue Globe Image Available from: http://grant.henninger.name/images/
PHP Logo Copyright © 2001-2006 The PHP Group. All rights reserved.
Apache Web Server Logo Copyright © 1999-2006, The Apache Software Foundation.
MapServer Logo Copyright © 1996-2006 Regents of the University of Minnesota.
PostgreSQL Logo Copyright © 1996 – 2006 PostgreSQL Global Development Group.
PostGIS Logo © 2005 Refractions Research.

References

ANDERSON, G., and MORENO-SANCHEZ, R., 2003, Building web-based spatial information solutions around open specifications and open source software. Transactions in GIS 7(4): 447-466.

APACHE SOFTWARE FOUNDATION, 2005, Index of /dist/httpd/binaries. [online]. Available from: http://archive.apache.org/dist/httpd/binaries. Accessed: March 2, 2006.

APACHE SOFTWARE FOUNDATION, 2006a, Apache Tomcat Project. [online]. Available from: http://tomcat.apache.org/. Accessed: February 17, 2006.

APACHE SOFTWARE FOUNDATION, 2006b, Using Apache with Microsoft Windows. [online]. Available from: http://httpd.apache.org/docs/2.0/platform/windows.html. Accessed: March 2, 2006.

BARR, M., 2005, Open Source versus Proprietary Software; A Discussion. [online]. Available from: http://www.matthewbarr.co.uk/opensource.htm. Accessed: July 6th, 2005.

BRADSHAW, C., 2006, Cultivating The Open Source Community; Market Map 2006. Geospatial Solutions Magazine. January 2006. p16.

CAMARA, G., 2004, Developing open-source GIS: What are the challenges?. INPE – Brazil. [online]. Available from: http://www.dpi.inpe.br/gilberto/present/terralib_edugila.ppt. Accessed: March 25, 2005.

CENTER FOR DISEASE CONTROL (CDC), 2004, Public Health GIS News and Information. CDC. [online]. Available from: http://www.cdc.gov/nchs/data/gis/cdcgis56.pdf. Accessed: February 15, 2006.

DELL, INC, 2006, The Dell Online Store: Build your system; Dimension B110. [online]. Available from: http://configure.us.dell.com/dellstore/config.aspx?c=us&cs=19&l=en&oc=DB110PRF1&s=dhs. Accessed: March 5, 2006.

DM SOLUTIONS GROUP, 2006, MapServer for Windows – MS4W. [online]. Available from: http://www.maptools.org/ms4w/. Accessed: March 10, 2006.

ESRI, 2006a, ArcIMS: Publish Maps, Data and Metadata on the Web. Environmental Systems Research Institute, Inc. [online]. Available from: http://www.esri.com/software/arcgis/arcims/index.html. Accessed: February 17, 2006.

ESRI, 2006b, Browse Live Mapping Sites Powered by IMS Solutions. [online]. Available from: http://www.esri.com/software/internetmaps/index.html. Accessed: February 17, 2006.

ESRI SUPPORT, 2003, FAQ: What image formats are supported by ArcIMS?. [online]. Available from: http://support.esri.com/index.cfm?fa=knowledgebase.techarticles.articleShow&d=26111. Accessed: March 5, 2006.

ESRI SUPPORT, 2005, Install: Install Apache 2.0.48 with Tomcat 4.1.29 with mod_jk2 using J2SDK 1.4.2 for ArcIMS 9.0 on Windows NT/2000/XP. [online]. Available from: http://support.esri.com/index.cfm?fa=knowledgebase.techarticles.articleShow&d=26149. Accessed: February 25, 2006.

ESRI SUPPORT, 2006, System Requirements for ArcIMS. [online]. Available from: http://support.esri.com/index.cfm?fa=knowledgebase.systemRequirements.matrix&pName=ArcIMS&productID=16&pvName=9.1&versionID=100&PID=16&PVID=282. Accessed: March 5, 2006.

GONZALES, M. L., 2003, The New GIS Landscape: With GIS Web Services, now even small companies can benefit from analytically rich geospatial information. Intelligent Enterprise. 6(3):p21(5).

INTERGRAPH, 2006a, GeoMedia WebMap System Requirements. [online]. Available from: http://support.intergraph.com/GeoMediaWebMap/gmwm/default.asp. Accessed: February 17, 2006.

INTERGRAPH, 2006b, SG&I – GeoMedia WebMap. [online]. Available from: http://www.intergraph.com/gmwm/. Accessed: February 17, 2006.

INTERGRAPH, 2006c, SG&I – Web Mapping Gallery. [online]. Available from: http://www.intergraph.com/sgi/customers/webmapping.asp?products=gmwm. Accessed: February 17, 2006.

LOWE, J. W., 2002, Spatial on a Shoestring: Leveraging Free Open-Source Software. Geospatial Solutions Magazine. June 1, 2002.

MAPINFO CORPORATION, 2005, MapXtreme 2004 Features. [online]. Available from: http://extranet.mapinfo.com/products/Features.cfm?ProductID=1849&productcategoryid=. Accessed: February 19, 2006.

MITCHELL, T., 2005, *Web Mapping Illustrated* (Reading: O'Reilly Media).

NETCRAFT, LTD., 2006, Web Server Survey Archives. Available from:

http://news.netcraft.com/archives/web_server_survey.html. Accessed: September 29, 2005.

NEUMANN, A., 2003, Delivering Interactive Topographic Web-Maps using Open-Source Database Technology. Paper for the 2003 SVG Open Conference, Vancouver. [online]. Available from: http://www.svgopen.org/2003/papers/abstract_neumann_topographic_maps_served_by_spatial_databases/. Accessed: March 25, 2005.

NEW ATLANTA, 2006, New Atlanta ServletExec 5.0. [online]. Available from: http://www.newatlanta.com/products/servletexec/index.jsp. Accessed: February 17, 2006.

OGC, 2006, OGC Resources: Registered Products. [online]. Available from:http://www.opengeospatial.org/resources/?page=products&conf=0&sort=organization&direction=desc. Accessed: February 15, 2006.

OPEN SOURCE GEOSPATIAL FOUNDATION, 2005, Open Source Geospatial Foundation. [online]. Available from: https://www.osgeo.org/. Accessed: February 19, 2006.

OPENSOURCEGIS.ORG, 2006, Open Source GIS. Available from: http://opensourcegis.org/. Accessed: March 13, 2006.

PHP GROUP, 2006, PHP: Installation and Configuration. [online]. Available from: http://us3.php.net/manual/en/install.php. Accessed: March 12, 2006.

POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2006, PostgreSQL: Frequently Asked Questions. [online]. Available from: http://www.postgresql.org/docs/faq/. Accessed: March 13, 2006.

RAGHAVAN, V., SANTITAMONT, P., MASUMOTO, S., and HONDA, K., 2002, Implementing Web GIS Applications using Open Source Software. GISDevelopment.net. [online]. Available from: http://www.gisdevelopment.net/technology/gis/techgi0062pf.htm. Accessed: March 25, 2005.

RAMSEY, P., 2002, Open Source GIS Fights the Three-Horned Monster. GeoWorld Magazine. Available from: http://www.geoplace.com/gw/2002/0208/0208gis.asp. Accessed: February 15, 2005.

REFRACTIONS RESEARCH, 2005, Refractions Research. [online]. Available from: http://www.refractions.net/. Accessed: March 12, 2006.

REGENTS OF THE UNIVERSITY OF MINNESOTA, 2006, How-Tos – UMN

MapServer. [online]. Available from: http://mapserver.gis.umn.edu/docs/howto. Accessed: March 13, 2006.

REICHARDT, M., 2004, Raising the standards for spatial web & wireless. GEO:connexion Sept. 2004 v3 i8 p44(3). Adams Business Media.

SMOTRITSKY, Y., 2004, Running GIS on Open Source. Connect: Information Technology at NYU. [online]. Available from: http://www.nyu.edu/its/pubs/connect/fall04/pdfs/smotritsky_GIS.pdf. Accessed: March 25, 2005.

TURNER, A., 2005, GIS Options: RELU Upland Moorland Scoping Study Project. University of Leeds, UK. [online]. Available from: http://www.geog.leeds.ac.uk/people/a.turner/publications/GIS%20Options%20Working%20Paper.pdf. Accessed: March 25, 2005.