



Evaluating open source GIS for libraries

Evaluating open
source GIS for
libraries

Francis P. Donnelly

Newman Library, Baruch College, City University of New York, New York, USA

131

Abstract

Purpose – The purpose of this paper is to provide an overview of free and open source (FOSS) geographic information system (GIS) software within the broader contexts of the open source software movement and developments in GIS, and pose the question of whether libraries and academic departments should consider adopting FOSS GIS.

Design/methodology/approach – The process of creating a basic thematic map was used to test six FOSS GIS software packages, as it is a multi-step process and a good example of a common task GIS users would want to perform. The test demonstrates what the process of creating a thematic map using FOSS GIS involves and how it compares to proprietary software like ArcGIS.

Findings – Each of the individual FOSS GIS packages had their own particular strengths and weaknesses and some performed well for thematic mapping. The FOSS packages generally were weaker compared to ArcGIS in terms of support for various projection and coordinate systems, joining attribute data to GIS files, and automatic labeling, but their advantage is that they were free in terms of cost and licensing restrictions. When coupled with plug-ins and helper applications the viability of the FOSS GIS packages increased.

Originality/value – There is little published research to date on the capabilities and applications of FOSS GIS despite the fact that it is a growing movement. This research provides a practical, application-based overview and critique of FOSS GIS.

Keywords Geographic Information Systems, Open systems, Knowledge mapping, Cartography

Paper type Technical paper

Received 28 July 2009
Revised 9 August 2009
Accepted 15 September
2009

Introduction

Geographic information systems (GIS) can be narrowly defined as:

[...] an integrated collection of software and data that is used to visualize and organize place-based data for the purposes of performing geographic analyses and creating maps (Wade and Sommer, 2006, p. 90).

Broader definitions also incorporate the GIS community, the group of people who create and share the data, and the activity of using GIS for problem solving and knowledge creation (Longley, 2008). The power and versatility of GIS has led to its adoption in a number of disciplines outside of geography and earth sciences.

GIS has evolved from custom-built programs created by government agencies on mainframe computers in the 1960s, to personal desktop based software in the 1980s, to integrated-web based solutions in the 2000s. Desktop GIS software continues to be popular as it allows users to manipulate free downloadable data and to create their own data in an unlimited way, as opposed to a web mapping application where the user is limited to what the designer provides. The desktop GIS market consists of a handful of companies, such as Intergraph, MapInfo, and ESRI, that create proprietary software. ESRI, the creators of the popular ArcGIS software, is the largest company in the GIS industry and provides software, support, consulting, training, and publications.



According to the company, ESRI software is used by 300,000 organizations worldwide, including 7,000 colleges and universities (ESRI, 2008).

There is currently a growing movement in the development and adoption of free and open source software (FOSS) for GIS. Steiniger and Bochner (2009) cite an increasing number of FOSS GIS projects, growing support of FOSS projects by government entities, increasing downloads of FOSS GIS software, and a growth in the number of use cases for open source GIS software for geospatial databases as evidence of this growing movement. The Open Source Geospatial Foundation, OSGeo at www.osgeo.org/ was established to support and promote collaboration and development of FOSS, and a number of books have recently been published to introduce web-based and desktop-based FOSS GIS to a wider community of end users (Sherman, 2008; Neteler and Mitasova, 2008; Davis, 2007; Mitchell, 2005). *The State of Open Source GIS*, the most comprehensive overview of FOSS GIS software to date, provides summaries of 16 GIS applications and ten shared software libraries (Ramsey, 2007).

Should libraries begin to offer access and services for open source GIS software? What are the advantages of adopting FOSS over proprietary software? Has FOSS reached a stage of maturity where it can support the needs of introductory GIS users? In this paper, I will provide an overview of open source software and GIS, followed by a discussion of its implementation in libraries. Several FOSS GIS applications will be tested and compared to ESRI's ArcGIS software through the process of creating a basic thematic map. Thematic mapping is a good example of a common GIS application that many new users wish to learn, and it captures some of the breadth and complexity of available GIS features. This research will illuminate the advantages and disadvantages of adopting individual FOSS GIS applications as well as provide an overall picture of the current state of FOSS GIS, so that libraries, academic departments, and others can make informed decisions regarding the adoption of FOSS GIS.

Overview of open source

In the introduction to their research on open source technology and policy, Deek and McHugh (2008, p. 1) introduce the open source movement as:

[...] a worldwide attempt to promote an open style of software development more aligned with the accepted intellectual style of science rather than the proprietary modes of invention that have been characteristic of modern business. The idea – or vision – is to keep the scientific advances created by software development openly available for everyone to expand and improve upon.

Under the open source model, software and its source code is freely shared. This allows a community of programmer-developers to form around the original creator of a project to create a collaborative social structure where contributors modify code to add improvements and fix bugs. Programmers make these contributions of their own free will for a variety of reasons, such as: gaining recognition within the FOSS community, improving the software for their own purposes, improving their technical skills, improving the software for the benefit of others, getting paid for creating FOSS, and promoting FOSS as a movement (Deek and McHugh, 2008). As a result of the transparency of the process and the pool of skilled, willing contributors, open source software can develop and improve rather quickly (Weber, 2004; Raymond, 2001). This approach stands in contrast to the traditional methods for creating proprietary software, where programmers are paid by a company to create software that is sold for

profit, and the source code for the software is hidden and copyrighted. Raymond's (2001) *The Cathedral and The Bazaar* is one of the earliest works that helped to define and explain the history, philosophy, social relations, and economics behind the open source movement, which traces its roots back to the late 1960s.

While all open source software is free, not all free software is open source. Software that is freely distributed but is copyrighted (thus denying the right to copy the software and modify the code) is not considered FOSS. Several open source licenses have been approved by the Free Software Foundation (FSF) and the Open Source Initiative (OSI), two of the primary advocacy and community groups of the FOSS movement. These licenses stipulate that software can be run for any purpose, can be modified and adapted for any need, and can be copied and redistributed for any purpose provided that the source code is freely distributed as well. The open source licenses differ in terms of how protective they are in guaranteeing the freedom of the source code.

The Linux operating system, an open source descendant of the UNIX operating system, is regarded as the defining open source project that aptly illustrates the open source process and ethos (Deek and McHugh, 2008, p. 89). The backbone of the modern internet runs primarily on open source software and protocols, such as the Apache web server and TCP/IP (Raymond, 2001). The first truly successful mass-market open source project is the Mozilla Foundation's Firefox browser, which was released in 2004 with critical acclaim (Deek and McHugh, 2008, pp. 43-4). Much of the Web 2.0 evolution has also been driven by open source blog, wiki, and content management software.

The open source movement in GIS stretches back at least to the early 1980s, with the creation of Geographic Resources Analysis Support System (GRASS). GRASS was created by the US Army Corps of Engineers Construction Engineering Resource Laboratory (CERL) as a low-cost, public domain alternative for resource management applications (Mitasova and Neteler, 2004). The project developed and grew with the use and contributions of many other government agencies. GRASS use began to decrease in the 1990s as commercial software grew in sophistication and ease of use and CERL transferred the management of GRASS to a non-profit foundation. In 1999 GRASS was re-released as a pure open source product under a GNU GPL license, and its popularity and use among government agencies and academia experienced a resurgence (Mitasova and Neteler, 2004).

Other developments in FOSS GIS have been more recent and have stemmed from many different sources. MapWindow was created by researchers at Utah State University in 1998 for spatial modeling, as ESRI's Map Objects module for ArcGIS was unable to meet all of their needs (Ames, 2004). The QGIS project was launched in 2002 by a group of volunteer developers who wanted to create an open source GIS viewer and editor for beginner and intermediate users (Pohlmann, 2005). It has also developed to become a more user-friendly front-end to the more powerful GRASS. The Regional Council of Infrastructures and Transportation of Valencia, Spain began development of gvSIG (Generalitat Valenciana, Sistema d'Informacio Geografica) in 2003 with the goal of building a GIS system of similar functionality to ESRI's ArcView 3. × software. The regional government is endeavoring to migrate its entire IT infrastructure to a Linux based system, which necessitated the development of a new GIS (Alfaro and Rico, 2005). JUMP GIS was also launched by government agencies, in this case provincial government ministries in Canada. When the agencies lost financial support for the project it split into several forks, such as OpenJUMP, which continue to be developed by

government agencies, companies, and interested volunteers. user-friendly Desktop GIS (uDig) was created by the Canadian company Refractions Research in 2004. The company had established itself in the GIS world by creating the PostGIS extension to the open source PostgreSQL database, which enhances the database's abilities to store and manipulate geospatial features. Many of the FOSS GIS tools incorporate the GDAL/OGR shared libraries. Created in 1998, these libraries provide functions for: converting files into different formats, reprojecting files, creating subsets of files, viewing attribute tables, and viewing metadata. GDAL/OGR can also be used as stand-alone, command line tools. As noted previously, the number of applications, publications, and downloads has indicated that FOSS GIS has rapidly grown over the course of this decade.

Libraries, FOSS, and GIS

There has been a growing awareness of FOSS in the library community as it is seen as an opportunity to improve service while lowering the costs of software and support (West, 2007). Libraries have adopted FOSS such as Firefox and Open Office for providing computer services (West, 2007), Firefox extensions like LIBX and Zotero for research services and support (Ritterbush, 2007), FOSS for running wireless networks (Feher and Sondag, 2008) and server applications (Bisson and Eby, 2007). There has been serious consideration of adopting open source ILS systems such as Evergreen and Koha (Grant, 2008). The Open Source Systems for Library Services site at: www.oss4lib.org advocates and educates librarians about FOSS and lists over one hundred applications in a number of categories that include bibliographic management, information retrieval, metadata tools, OPACs, protocols, public services, and repositories (Chudnov, 2007). The adoption of FOSS has also grown in the academic and education community in general, who see the benefits of Web 2.0 applications like blogs and wikis as tools for contributory, educational learning (Guhlin, 2007).

The use and support of GIS in libraries has also grown significantly over the past fifteen years. GIS began to appear in US libraries in the 1980s with the introduction of desktop GIS software and grew with the distribution of the Census Bureau's TIGER line files with the 1990 Census (Gabaldón and Replinger, 2006). The role of libraries in providing GIS services has grown to include: data collection, data discovery tools and support, technical support, workshops and training, marketing and outreach, technical approaches to delivering geographic data, and web mapping (Morris, 2006).

A number of studies have documented the growing trend of libraries offering GIS services. A 2005 American Research Libraries survey of its members found that 31 out of 57 respondents said that they offered support for GIS and remote sensing data and services (Salem and ARL, 2005). A 2006 survey of member institutions of the Alliance Science List of public and private institutions in the US Northwest and the Oberlin group of private liberal arts schools throughout the USA found that 31 had implemented GIS services, 15 were considering adopting services, and 57 had no plans of doing so (Gabaldón and Replinger, 2006). The study found that public universities, schools with larger budgets, and schools that offered PhD or Masters degrees were more likely to offer GIS services. A study of the number of GIS related articles published in journals included in the two major library and information science databases found that 146 GIS articles were published in 69 different LIS publications between 1990 and 2005, and that the trend in number of articles published was increasing (Michalec and Welsh, 2007).

There is not as much survey data to measure whether libraries with GIS services are offering FOSS GIS software, but the data that exists suggest that it is not widely supported. A survey of 17 Canadian libraries that offer GIS services asked “Which GIS software products do your clients use at your facility?” Of those libraries, 100 percent responded that patrons used ESRI products, 53 percent used INDRISI, and 24 percent used MapInfo. Only one institution responded that they use a FOSS GIS application, TatukGIS (University of Waterloo Map Library, 2007). In a survey of smaller academic libraries (Carnegie Classification Master’s Colleges and Universities I and II), 22 libraries out of 138 responded that they did offer GIS services, and 21 said that they supported ESRI’s ArcView. GRASS was the only FOSS GIS listed as a survey item, and the number of libraries that offered it was zero (Kinikin and Hench, 2005).

While libraries have been slow to adopt FOSS GIS, there is some evidence that suggests that there is a core, dedicated group of academic departments and researchers who are. Examples include the California Soil Resource Lab at UC Davis, which supports GRASS and the GIS toolkits GDAL and OGR, the independent research organization Oxford Archeology in the UK, which supports GRASS and gvSIG, and the Utah Water Research Lab at USU, which created MapWindow. A master’s thesis that rigorously compared ArcGIS 9.0 to GRASS 6.0 for conducting an advanced land use analysis found that, while ArcGIS was more intuitive to use, GRASS performed comparably to ArcGIS in most areas and outperformed it some others (Buchanan, 2006), concluding that it was suitable for doing advanced GIS analysis. In Tufto and Cavallini’s (2005) discussion of whether wildlife biologists should use free software, they cite GRASS as a definite alternative to proprietary GIS for scientific research and specifically point to its strength in easily performing complex tasks and its ability to integrate with the open source R statistical package. FOSS4G, an annual conference dedicated to FOSS GIS, draws hundreds of participants from around the globe and consistently includes research presentations from students and faculty from many disciplines who integrate FOSS GIS into their projects.

While FOSS GIS has been accepted by a dedicated community of faculty and students doing advanced research, is it suitable for GIS novices and intermediate users who need basic GIS functions? While Tufto and Cavallini (2005) pointed to GRASS’s ability to perform complex operations, they also noted that rather simple tasks like making a map with a legend were needlessly complex. Can FOSS GIS be used as an introductory teaching tool? Which FOSS GIS applications, if any, should a library implement?

Methodology: evaluating FOSS GIS

I compared a number of FOSS GIS software applications to ArcGIS by performing a basic GIS task: creating a thematic map. The goal of this research was to determine the suitability of adopting FOSS GIS in libraries to serve the needs of beginning and intermediate GIS users. Creating a thematic map is a good example for testing basic GIS functionality for two reasons. First, it is a multi-step process that requires several tools. The features that will be tested in this process include:

- (1) Adding a shapefile (a common GIS file format created by ESRI).
- (2) Selecting a subset of geography to create a new GIS layer with just that subset.
- (3) Converting the projection of a GIS layer to a different projection.

- (4) Adding an attribute table and joining it to a shapefile based on a common field.
- (5) Classifying data into categories based on a specific method.
- (6) Symbolizing data using a color scheme.
- (7) Labeling features.
- (8) Creating a map layout with typical map elements.
- (9) Exporting the map out as a static file.

Second, it is an example of an application that most new GIS users would want to perform. In the survey of 17 Canadian university libraries that offer GIS services, when asked how often clients use their resource center to create maps that include cartographic information, 53 percent of the institutions said regularly and 24 percent said often (University of Waterloo Map Library, 2007). The survey also includes three more commonly used tasks that are part of this thematic mapmaking example: reprojecting data, creating subsets of data, and working with attribute tables.

Six FOSS GIS applications were chosen for review: GRASS, QGIS, uDig, gvSIG, OpenJUMP, and MapWindow (see Table I). These six were chosen as they all have a medium to large, worldwide user community based on downloads (Steiniger, 2008) and offer strong support for vector-based GIS. All of the applications are cross platform and run on Windows, Linux, and Macintosh, with the exception of MapWindow which runs only on Windows. All of the applications were tested using the Windows XP operating system with Service Pack 3, on a machine with a 2.8 GHz Pentium 4 Processor and 1 GB of RAM.

In preparation for the tests, a shapefile of the world's administrative boundaries was downloaded from the UN Environment Program's Geportal. Country boundaries were extracted with small modifications to boundaries and attributes. The native projection of the file was defined as simple WGS 84. An attribute table with internet users per 100 population for every country in North America was generated and downloaded from the UN Data online database, and was prepared so that it could be joined to the world shapefile based on ISO codes. The goal of this test was to create a thematic map of internet users by country in North America. Based on the steps enumerated above, the countries of North America needed to be extracted from the world file and exported to a new shapefile. The new shapefile was converted to a more appropriate projection, in this case the Albers North America Equal Area Conic projection. The attribute table of internet users was added to the GIS program and joined to the GIS features based on ISO country code fields. Data were classified based on the natural breaks method into three categories and symbolized with different colors. Country names were labeled and a finished map created with a title, legend, scale bar, and north arrow. The finished map was exported out into a static vector-based (pdf) and raster-based (an image file like .png or .jpg) format.

The interface and ease of use of each application was noted over the course of the tests. If the software could not perform a particular task, then plug-ins for that software were downloaded to accomplish it, or other software applications (the GDAL/OGR toolkit and the Calc spreadsheet program) or web tools (Spatial Reference and ColorBrewer) were used to complete it (see Table II). This was done to simulate how this exercise would be accomplished if one had to use that particular software application.

Software	URL	Year launched	Operating Systems	Interface	Shapefiles?
gvSIG 1.12	www.gvsig.gva.es/	2003	Windows, Mac, Linux	Standard	Yes
QGIS 1.0	www.qgis.org/	2002	Windows, Mac, Linux	Standard	Yes
uDig 1.1.1	http://udig.refractorions.net/	2004	Windows, Mac, Linux	Semi-standard	Yes
OpenJUMP 1.2F	http://openjump.org/	2002	Windows, Mac, Linux	Standard	Yes
MapWindow 4.6.602	www.mapwindow.org/	1998	Windows	Standard	Yes
GRASS 6.4.0 (via QGIS plugin)	http://grass.itc.it/	1982	Windows, Mac, Linux	Standard (using QGIS)	No

Table I.
FOSS GIS software

Resource	Type	URL	Description	Use
GDAL/OGR	Software	www.gdal.org/	Command line tools	Projection transformation, file format conversion, metadata viewing
fTools	Software	www.ftools.ca/	Third-party plugin for QGIS	Expands QGIS capabilities including table joins and some projection transformations
Calc	Software	www.openoffice.org/	Spreadsheet (part of Open Office Suite)	Create scatter plots for natural breaks, modify and create dbf files (Excel 2007 does not support dbf)
Spatial reference	Website	http://spatialreference.org/	Reference for coordinate systems	Look up EPSG codes, generate projection files
ColorBrewer	Website	www.colorbrewer.org/	Reference for color palettes	Generate good color schemes with associated RGB and HSV values
SQLite Manager	Software	http://addons.mozilla.org/en-US/firefox/addon/5817	Plugin for Firefox browser	Create desktop geodatabases for GRASS, necessary in GRASS for table joins

Table II.
FOSS helper applications

Findings

Each of the FOSS GIS packages had its own particular strengths and weaknesses, and overall there was not a single package that was able to match all of the functionality that ArcGIS has for creating thematic maps. However, when an individual FOSS package was paired with some plugins or other tools, several performed reasonably well relative to ArcGIS. Open source projects tend to have a more modular design than proprietary software, where several individual pieces of software or plugins are created to perform separate tasks. This was certainly the case with the FOSS GIS packages that were tested here.

Most of the FOSS software had a standard GIS interface. Like ArcGIS, there was a main window for viewing geographic data, a table of contents to the left of the window that lists the different features and files that are part of the project, and drop-down menus and toolbars above the window for accomplishing different tasks (for example, see Figure 1). Clicking on a feature in the table of contents would open a series of tabs or windows for viewing or modifying that particular feature. There were few deviations from this norm, with the exception of GRASS, which used an older interface. For this reason, GRASS was tested using the QGIS GRASS plugin, which allows users to use the familiar GIS interface offered by QGIS while allowing access to the powerful GRASS tools through window-based icons and the command-line. A modernized version of the native GRASS graphic interface will be released in the near future.

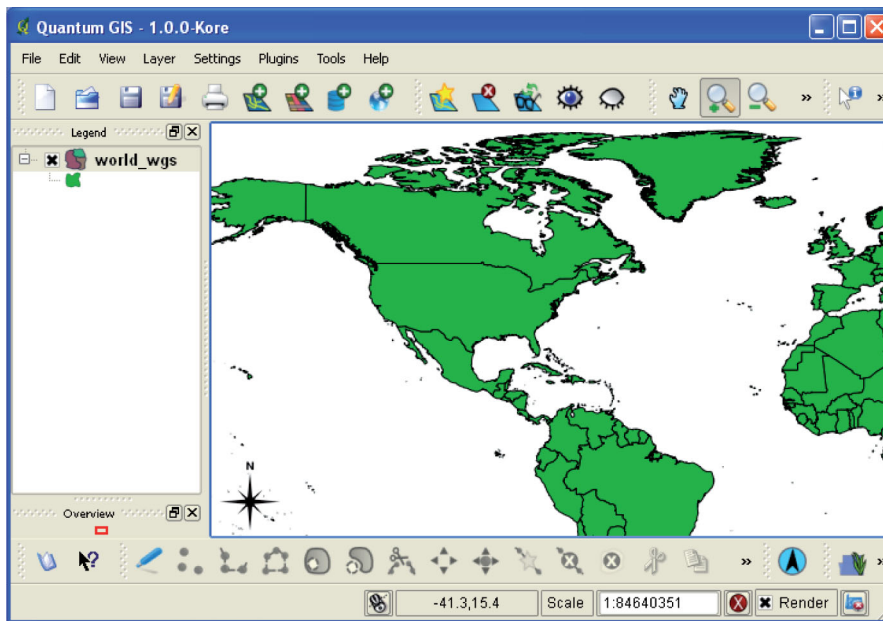


Figure 1.
An example of a standard
GIS interface (QGIS) with
the initial GIS data file of
the world

ESRI shapefiles could be directly added and viewed in the data window of all the software with the exception of GRASS. GRASS uses its own file format and has a set of commands that allows you to convert many GIS formats to the GRASS format. Before you can add a file to GRASS, you must set up a working environment that is based on three parts: a database, which is essentially a directory that will contain GIS data, a location, which is a subfolder that is defined for a geographic area and coordinate system, and a mapset, which is defined by a specific project for a specific user or group of users. This can be off-putting to new users, who simply want to add and view some data. The advantage is that it forces users to carefully plan how they want to organize their data and requires that all data is projected in the same format, which prevents potential problems with misaligning different data layers. Once the environment is established, there are tools for transforming data from shapefile to GRASS format, transforming the projection of the data, and importing the data into the mapset.

Each of the FOSS GIS packages had its own method for handling the projection of the data. In some cases (uDig, MapWindow) the data window took the projection of the first layer that was added to the window, similar to ArcGIS. In other cases (gvSIG, QGIS) you had to know what projection the layer was in and then define the window based on that projection. The software would render layers properly even if the window definition was different from the layer, but you would encounter problems later on when attempting to add additional layers. Since these programs did not have the ability to detect what the projection of the layer was, the user has to rely on metadata or another tool to determine what the projection is so that he/she can define the window properly. The vector OGR tools of the GDAL/OGR command line toolkit

can be used to determine the projection of the file. The following command returned summary information about the projection, which was then used to define the window:

```
ogrinfo -al -so world_wgs.shp
```

The next step was to select the North American countries of the world file and export them out as a new file. All of the software packages provided the ability to point and click on countries or draw a box around them to make a selection. You could also open the attribute table for the countries and select the North American ones from the table. Each software package had a select by attributes option, which opened a SQL window for building queries to select features based on a common attribute. Since all of the countries in the file had a continent attribute, it was easy to select all of the countries in North America with the SQL window. Although it was not necessary for this test, most of the software packages allow you to select features geographically based on their location relative to another layer. The variety of geographic selection possibilities ranged from extensive (gvSIG, GRASS, OpenJUMP) to limited (MapWindow) to none (uDig). QGIS did not have native support for geographic selection, but the installation of a plugin called fTools provided good support. All of the programs allowed you to export the selection to a new file.

Transforming the new North American countries file from simple WGS 84 to a map projection suitable for North America, Albers Equal Area Conic, proved to be troublesome. The open source programs rely on a specific library of map projections created by the European Petroleum Survey Group (EPSG). The library is extensive and covers all projections that would be suitable for small areas (UTM for different zones, State Plane systems, etc.) and many countries. However, the library does not contain many continental or global projections suitable for thematic maps, such as the various Albers and Lambert projections for continents and the Robinson and Winkel Tripel projections for the world.

The OGR tools can be used to transform projections, but since it also relies on the EPSG library it is necessary to give OGR a file it can use to reference the Albers projection. The online tool Spatial Reference: <http://spatialreference.org/> can be used to look up projection information and view it in a number of standardized formats that are recognized within the GIS community, such as OGC WKT and Proj4. Using Spatial Reference, you can download the record for the Albers North American Equal Area Conic Projection and save it as a text file that looks like this:

```
PROJCS["North_America_Albers_Equal_Area_Conic",  
  GEOGCS["GCS_North_American_1983",  
    DATUM["North_American_Datum_1983",  
      SPHEROID["GRS_1980",6378137,298.257222101]],  
    PRIMEM["Greenwich",0],  
    UNIT["Degree",0.017453292519943295]],  
  PROJECTION["Albers_Conic_Equal_Area"],  
  PARAMETER["False_Easting",0],  
  PARAMETER["False_Northing",0],  
  PARAMETER["longitude_of_center",-96],  
  PARAMETER["Standard_Parallel_1",20],  
  PARAMETER["Standard_Parallel_2",60],  
  PARAMETER["latitude_of_center",40],  
  UNIT["Meter",1],  
  AUTHORITY["EPSG","102008"]]
```

Typing the following command in OGR reprojected the data:

```
ogr2ogr -t_srs albers_ogcwkt.txt na_albers.shp na_wgs.shp
```

The command took the existing North American countries shapefile (na_wgs.shp), transformed it (-t_srs) to the Albers projection stored in the text file (albers_ogcwkt.txt) and created a new shapefile that has North America in the Albers projection (na_albers.shp). In some cases, it was necessary to define the shapefile as WGS 84 before reprojecting, as some of the software (gvSIG, OpenJUMP) did not give the North American file the same projection as the world file when it created the subset layer.

It was necessary to use spatial reference and OGR for software that lacked projection transformation support (uDig) and that supported only EPSG transformations if plugins were installed (QGIS and OpenJUMP). gvSIG supported both EPSG and ESRI projections, but its support for ESRI projections was incomplete and running the transformation to Albers caused it to crash. The Spatial Reference site can also be used to get projection information in the PROJ format, which can be used to define a custom projection in QGIS for non-EPSG projections like Albers. This enables users to define the data window properly and allows GRASS to make projection transformations by referencing the custom projection (as GRASS only supports EPSG). MapWindow was the only program that natively supported the ESRI projection library that included Albers.

Once the shapefile of North America was created and projected properly, the next step was to join the attribute table of internet users per country downloaded from the UN to the shapefile using the ISO country code field in each. gvSIG was the only program that allowed you to add an attribute table stored outside of a database (in dbf format) to a project and dynamically join it to a shapefile. However, the dynamic join had a bug that caused problems when it came time to classify the data, which necessitated exporting the shapefile with the join out as a new shapefile to make the joined fields permanent. QGIS (with the fTools plugin) and OpenJUMP could not do dynamic joins, but were able to reference external files (dbf for QGIS and delimited text for OpenJUMP) and join them to a shapefile to create a new shapefile. GRASS could recognize and read dbf files but did not allow dbf joins. GRASS can join tables stored in SQLite databases, which are simple file-based databases. By exporting the North American vector data from the GRASS layer (which is stored in dbf) to a SQLite database that has the UN data stored in a table (a Firefox add-on called SQLite Manager is a free tool for working with SQLite databases), we could join the data and map it. MapWindow had a plugin for table joins that was not functioning and uDig did not support joins, which required manual input of the attribute data.

The data could be classified and symbolized once they were joined to the shapefile (see Figure 2). All of the FOSS GIS programs supported classifying data based on equal intervals, where data values are divided into an equal number of classes that a user specifies. All of the programs supported manual divisions created by the user. QGIS, GRASS, gvSIG, and uDig supported quantile divisions, where data values are sorted from lowest to highest and are divided into classes that have an equal number of data points. For software that did not support quantiles, manual sorting and grouping was necessary. gvSIG was the only software that supported the natural breaks method, also known as Jenks Optimization. In this system, data are grouped into classes using an algorithm that determines where the largest gaps in the data are based on the number

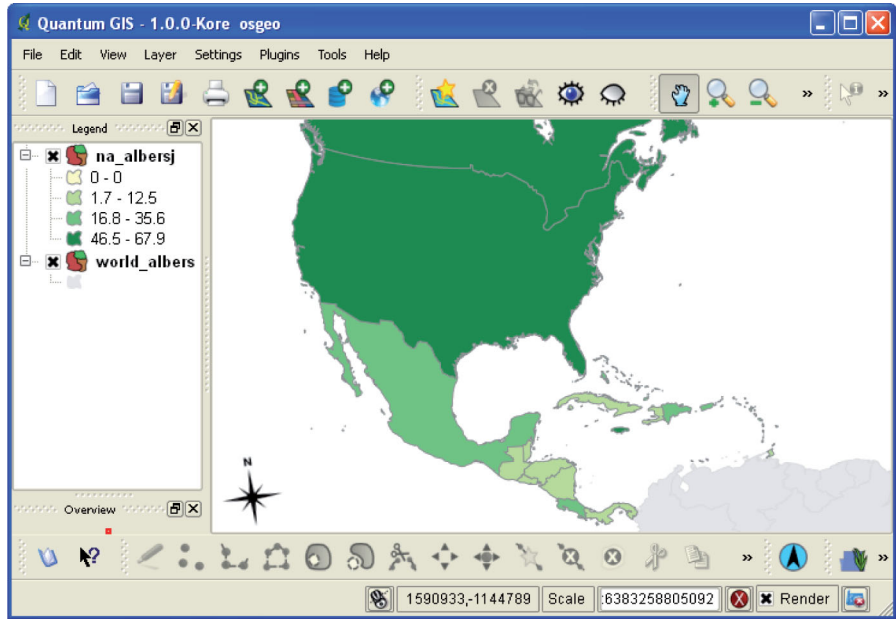


Figure 2.
Reprojected, classified,
and symbolized data for
North America

of classes. For GIS software that lacks this method, the simplest method for creating natural breaks would be creating a scatter plot of the data in a spreadsheet program and identifying break points manually.

The software packages fell into two categories for symbolization: those that offered predefined color palettes, and those that did not. QGIS, GRASS, gvSIG, and MapWindow did not offer any predefined palettes, and had a rather basic swatch of colors to choose from. In most cases, you have to modify the color choices for each data class in order to get a decent color scheme. These four packages also offered the option of inputting RGB or HSV hexadecimal values to specify colors, either by typing them in manually or using a sliding palette typically found in graphic design packages. This option is valuable as it allows you to use the online ColorBrewer tool at www.colorbrewer.org/ to choose appropriate color schemes, which is essential for creating an effective thematic map (Harrower and Brewer, 2003). You specify the number of classes and the tool lets you choose from a variety of color palettes and provides RGB and HSV values. These values can be entered in the GIS color dialog box and can be saved as custom colors for reuse. uDig and OpenJUMP offered predefined palettes that appear to be based on the ColorBrewer tool. This made the process of choosing a good color scheme easier and faster, but the ability to customize was a little more limited.

One weakness of the FOSS GIS packages relative to ArcGIS is the ability to automatically label features. ArcGIS has evolved to the point where it can automatically place labels that require little manual adjustment. Conflict detection to avoid overlapping labels is advanced, labels can change based on scale, duplicate labels (for multi-part features) can easily be turned off, users can set global options to get better placement and build a hierarchy of precedence for feature labeling. ArcGIS can create special annotated label features where you can store attributes for each

individual label, such as font size, font face, horizontal and vertical placement, and slope. ArcGIS also allows you to convert the label attributes into simple text or graphics which can be moved and changed at will.

In contrast, automatic labeling in FOSS GIS is not as mature. uDig, OpenJUMP, and MapWindow all have conflict detection and duplicate feature mechanisms, but it was still difficult to get most labels to draw at any scale; if there is a conflict, labels were often dropped rather than repositioned. gvSIG had no conflict detection, so labels overlapped and tended to be placed just off-center within the polygons. QGIS (and GRASS via QGIS) had the weakest automatic labeling, as labels were created for every single polygon (for example, every single island in Hawaii and Alaska gets labeled with United States) and all of the labels overlap. To overcome these weaknesses, each program had support for storing annotation properties (general placement, font, slope, etc) in special annotation fields in the layer, similar to ArcGIS. QGIS had the strongest support for annotation, as you could specify fields with X and Y coordinate for label placement. For example, if you added a latitude and longitude field that had the coordinates for the geographic center of each country, you could reference that field for placing labels in the center of each country and then modify the coordinates in the table to optimize placement. gvSIG provided the option of creating separate, dedicated annotation layers. MapWindow's annotation features were more limited, but it did have a novel solution for manual label placement that allows you to drag labels wherever you wish, without converting the labels to text. MapWindow creates a special label file, .lbl, that remembers the coordinates of where the label is placed for each project.

ArcGIS has a separate window for working with map layouts. Most of the FOSS GIS software have their map layout capabilities listed under a print layout menu, and with a few exceptions they all had support for adding standard map elements such as legends, scale bars, north arrows, frames, a title, and free text (for example, see Figure 3). gvSIG had the most sophisticated map layout of all the FOSS GIS programs, but was difficult to work with as it had problems redrawing a map when you added an element (such as text) while zoomed in to do detailed work. It also had trouble configuring page sizes that are commonly used in North America. MapWindow had the most stable print layout, but it used a static template that allowed for limited customization, and there was no way to add free text. QGIS (and GRASS via QGIS) had a basic layout screen, but it was customizable and the scale of the map could easily be detached from the current data view so that the map would not change if the view changed. This is actually a key feature; the basic layout tools in uDig and OpenJUMP were more difficult to use because the map always reflected what was in the view. OpenJUMP was the only program that did not have a native map layout; you needed to download a plugin to get basic functionality.

Each of the programs provided a way to export maps out to a static file format. In general, it is good to have at least one vector (.pdf, .svg, .ps) and one raster (.png, .jpg, .gif) option for saving maps. The vector formats are good for creating stand-alone map documents and for preserving the full quality of the map, while raster formats are useful for saving maps that will be embedded into documents or web pages. If a program lacks support for one format or another, you can achieve the export by external means. If you have Adobe Acrobat, or one of many free programs for creating pdfs, then a pdf driver will appear in your print menu and you can simply print to a pdf

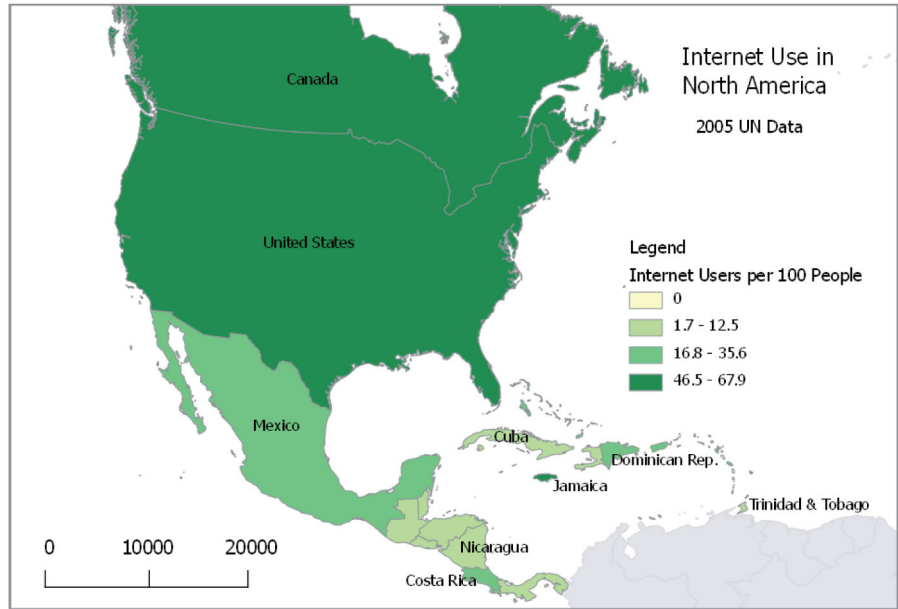


Figure 3.
Final map layout
produced in QGIS

file instead of to a printer. If a program lacks raster support, the options for creating a raster file are more limited (screen captures with edits, or sending to pdf and exporting from the pdf program to a raster format). QGIS, GRASS, and MapWindow did not support pdf export, requiring an external pdf driver. gvSIG was the only program that did not support raster exporting at all; you can export your data view, but not a finished map.

Each software package allows you to save the project in a file that preserves links to the particular data layers you were using and saves the symbolization, classification, scale, labeling, and map layout. GIS users frequently have problems when they move data, as the links from the map file to the data can be broken. This is common on Windows machines when data is transferred to a usb drive, as Windows randomly assigns drive letters each time so that E:\data\mymap on one machine becomes F:\data\mymap on another. ArcGIS provides users the options to save data using relative paths, so that the path only contains the name of the data folder and the subfolders below it while excluding folders above it. The FOSS GIS options do not provide this alternative and save all links as absolute, with the exception of MapWindow which appears to save relative links. Since all of the FOSS GIS applications use an open XML format for saving projects, it is possible to open the project file in a text editor and edit the path to restore links if they become broken.

Discussion

While FOSS GIS has not evolved to the point where it matches ArcGIS in every aspect of the thematic mapping process, it does represent a viable alternative. Many of the basic features, such as support for many GIS file formats, data and attribute table

selections and queries, data selection, and data export are present (see Tables III and IV). The lack of functionality relative to ArcGIS for certain components such as data classification and symbolization is inconvenient but not insurmountable, and in all likelihood as the software matures these features will improve. Many of the shortcomings of a particular software package could be easily overcome with the addition of plugins or helper applications. In most instances, each FOSS GIS package had one or two small, special features that either ArcGIS or its FOSS peers did not have.

The two issues to consider when adopting FOSS GIS for novice users and for teaching use are projections and attribute table joins (automatic labeling may or may not be an issue depending on the goal of a particular project). Coordinate systems and projections are perhaps one of the most difficult concepts for new GIS users to grasp regardless of which software package they use. The lack of internal support for projections outside of the EPSG library adds a new layer of complexity for learning this concept if you are using FOSS GIS and require projections that are not supported by this library. However, the extra steps of using the GDAL/OGR tools and learning about different projections via the Spatial Reference website does force new users to really learn this concept.

The inability for joining attribute tables to shapefiles is more problematic, as it is not easily accomplished using an outside program unless a user is familiar with databases. Many of the FOSS GIS applications, uDig in particular, have been designed to work with geodatabases such as PostGIS, where geographic data and attribute tables can be stored together and database operations like table joins can easily be accomplished. However, these geodatabases are server based and are not an ideal solution for desktop users. Unlike ArcGIS, FOSS GIS currently does not support a simple desktop geodatabase that can be directly accessed, although SQLite could be a future candidate and is supported by GRASS and the GDAL/OGR tools. Attribute table joins are a common task introduced when teaching GIS and are a good example of how GIS is a powerful tool, as you can easily visualize data that is stored in a regular spreadsheet file. If the GIS software does not support joins, then the instructor has the dubious task of introducing new users to relational databases while they are teaching GIS. While GIS users certainly should learn how to use databases, it is not always desirable or possible to teach both at the same time. Beyond this issue, even veteran GIS users want the ability to do a basic table join on the fly for small projects without having to go through the trouble of setting up a database.

There are some compelling reasons to consider FOSS beyond functionality. ArcGIS and other proprietary software is costly for institutions and is financially prohibitive for most students, whereas open source software is free. When installed in a classroom or lab environment, ArcGIS is typically installed so that it points to a license server which gives the software permission to open when launched. This limits installation to machines hardwired to the campus network and requires the setup and maintenance of a dedicated license server. FOSS GIS can be installed on any machine, fixed or wireless, and does not have a proprietary license. Currently ArcGIS only runs on Windows, whereas most FOSS GIS applications are cross-platform. Adopting ArcGIS forces institutions to adhere to an almost annual cycle of updates, and failing to keep up with the most updated version can cause

Table III.
Comparison of FOSS
features – data
processing, classification,
and symbolization

Software	Queries ^a	Projections	Table joins	Data classes ^b	Colors
gvSIG	pc, ts, at, geo	EPSG, some ESRI, cannot view projection info for layer; must define window, transformations supported	dbf	man, eq, qu, nb	basic swatch, RGB, HSV
QGIS	pc, ts, at, geo (w/plugin)	EPSG, can add custom projection, cannot view projection info for layer, must define window, transformations for EPSG (w/plugin)	dbf (w/plugin)	man, eq, qu	basic, RGB, HSV
uDig	pc, ts, at	EPSG, can add custom projection, can view projection info for layer, window takes projection of first layer added, no transformations	no	man, eq, qu	palettes
OpenJUMP	pc, ts, at, geo	Support unknown, cannot view projection information for layer or window, transformations for EPSG (w/plugin)	txt	man, eq	palettes
MapWindow	pc, ts, at, geo	ESRI, can view projection info for layer, window takes projection of first layer added, transformations supported	no	man, eq	basic swatch, RGB
GRASS (via QGIS plugin)	pc, ts, at, geo, cl	EPSG, can add custom projections, can view projection info for layer, projection must be defined in advance for mapset, transformations supported	SQLite	man, eq, qu	basic swatch, RGB, HSV

Notes: ^a pc (point and click), ts (table select), at (attribute select), geo (geographic select), cl (command line select); ^b man (manual), eq (equal intervals) qu (quantiles), nb (natural breaks)

Software	Labels	Map layout	Export	Special features
gvSIG	No conflict detection, fair auto placement, annotated labels	All standard elements, manual label placement difficult, can detach view from map	pdf, ps	Clear separation of data view, maps, and tables into separate elements
QGIS	Multiple labels for each polygon, no conflict detection, poor auto placement, extensive support for annotated labels	All standard elements except north arrow (which is in data window), manual label placement good, can detach view from map	Many rasters, svg	Easy to search for and add plugins, excellent documentation, extended capabilities via GRASS plugin
uDig	Conflict detection, good auto placement, labeling scale dependent and will not place if overlap, no annotated labels	All standard elements, no manual labels, cannot detach view from map	Many rasters, pdf	Can filter color schemes based on intended output
OpenJUMP	Conflict detection, good auto placement, labeling scale dependent and will not place if overlap, annotated labels	No map composer, must download plugin that provides limited support, no manual labels	Varies (w/ plugin)	Attribute table indicates whether polygons are single or part of multi-part feature
MapWindow	Conflict detection, fair auto placement, annotated layers, special label movement tool	Static layout, all standard elements, no manual text, cannot detach view from map	Some rasters	Good solution for manual label placement, has metadata editor
GRASS (via QGIS plugin)	Multiple labels for each polygon, no conflict detection, poor auto placement, extensive support for annotated layers	All standard elements except north arrow (which is in data window), manual label placement good, can detach view from map	Many rasters, svg	Easier than native GRASS interface but with complete power via command line, excellent documentation

Table IV.
Comparison of FOSS GIS features – mapping and special features

incompatibility issues across campus or between institutions. Recent examples include incompatibilities between map files (mxds) between ArcGIS 9.1 and 9.2, geodatabase incompatibilities between 9.2 and 9.3, and six service packs released between versions to fix bugs and add features.

Despite the advantages of FOSS GIS, it would likely be an addition to the GIS toolkit for large research libraries and not a replacement for proprietary GIS products. Libraries with GIS services will still need to provide ArcGIS to fulfill the needs of their patrons. Documentation, support, and training for ArcGIS, MapINFO, and other products still outstrips that of the FOSS alternatives. ESRI also supplies educational materials like its virtual campus courses and temporary free or low cost software to students to overcome some of the cost issues.

Large libraries can still leverage FOSS GIS software to cut costs, and for small to medium libraries FOSS GIS could be a solution for providing some GIS services as opposed to providing none at all. It could be especially appropriate for institutions that lack a strong geography or earth sciences program, but where there is interest from students and faculty in other disciplines who want to learn GIS. Students pursuing geography, planning, or earth sciences degrees may want exposure to ArcGIS as they are likely to use it in their future careers. However, students studying biology, business, or sociology who are interested in taking GIS as an elective so they can apply it to their field may benefit from learning GIS through FOSS GIS, as they will have easy access to the software after they graduate. There have been experiments with offering online distance learning courses using QGIS and GRASS since students can easily download the software at no cost, which is particularly important for students in developing countries (Schweik *et al.*, 2007).

While some of the individual FOSS GIS packages performed better than others in this vector-based thematic mapping test, it is important to note that each of the FOSS GIS packages was designed for a specific end use which does not necessarily include thematic mapping. For example, GRASS was originally built for power users for conducting land management raster analysis; enhanced vector support (Mitasova and Neteler, 2004) and map layout capabilities (Neteler and Mitasova, 2008, p. 271) were added later. While gvSIG was initially launched to become a fully functional, stand-alone GIS program, QGIS, MapWindow, OpenJUMP, and uDig were initially constructed as viewers or as platforms for developers to extend and build their own GIS. Some of them have grown to be more fully developed packages, while others focus on specific applications. GIS encompasses more than just map making and many of these packages may be suitable for other applications for intermediate to advanced users, such as accessing web-based GIS data, serving as a front-end for server-based geodatabases, and conducting advanced raster analyses.

FOSS GIS is a growing movement that will continue to evolve. The creation of more and better documentation online and in print, case studies of using FOSS GIS in library and classroom instruction, and surveys to collect data about GIS applications and use in academic communities are all possible avenues for additional research in exploring this expanding world of GIS. As adopters of open source technology, academic libraries should consider adding FOSS GIS and becoming involved with the open source movement. GIS and data librarians are in a unique position as they gain familiarity with several applications of GIS under many circumstances, since they have patrons from many different fields whose needs for GIS range from basic to advanced.

Providing constructive feedback for bugs and improvements will help the software evolve, and contributing tutorials and documentation will help improve FOSS GIS' accessibility to newcomers and increase its adoption by experts. Within a broader context, supporting movements like open source, which promotes the free exchange of technology, and the related open access movement that promotes the free flow of information (Krishnamurthy, 2008) to support education and research is probably in the best interest of libraries and their patrons.

References

- Alfaro, A. and Rico, G. (2005), "gvSIG: open source solutions in geospatial technologies", *GIS Planet Conference 2005, Lisbon*, available at: www.gvsig.gva.es/fileadmin/conselleria/imagenes/Documentacion/articulos/GIS_PLANET_2005_gvSIG.pdf (accessed March 29, 2009).
- Ames, D. (2004), "Why was MapWindow GIS developed?", *MapWindow GIS*, p. 29, available at: www.mapwindow.org/wiki/index.php/ (accessed March 29, 2009).
- Association of Research Libraries (ARL) Salem, J.A. (2005), *Spatial Data Collections and Services*, SPEC Kit, 291, Association of Research Libraries, Office of Leadership and Management Services, Washington, DC.
- Bisson, C. and Eby, R. (2007), "Open-source server applications", *Library Technology Reports*, Vol. 43 No. 3, pp. 48-53.
- Buchanan, T. (2006), "Comparison of geographic information system software (ArcGIS 9.0 and GRASS 6.0): implementation and case study", Master's thesis, Fort Hays State University, Hays, KS, available at: www.toddbuchanan.net/thesis_ver.pdf (accessed April 14, 2009).
- Chudnov, D. (2007), "The future of FLOSS in libraries", in Gordon, R.S. (Ed.), *Information Tomorrow: Reflections on Technology and the Future of Public and Academic Libraries*, Information Today, Medford, NJ, pp. 19-32.
- Davis, S. (2007), *GIS for Web Developers: Adding Where to Your Web Applications*, Pragmatic Bookshelf, Raleigh, NC.
- Deek, F.P. and McHugh, J.A. (2008), *Open Source: Technology and Policy*, Cambridge University Press, New York, NY.
- ESRI (2008), "ESRI: more to offer than just a great career", available at: www.esri.com/library/brochures/pdfs/esri-recruitment.pdf (accessed April 14, 2009).
- Feher, J. and Sondag, T. (2008), "Administering an open-source wireless network", *Information Technology & Libraries*, Vol. 27 No. 3, pp. 44-54.
- Gabaldón, C. and Repplinger, J. (2006), "GIS and the academic library: a survey of libraries offering GIS services in two consortia", *Issues in Science & Technology Librarianship*, Vol. 48 No. 1.
- Grant, C. (2008), "Gone open yet?", *Public Library Quarterly*, Vol. 27 No. 3, pp. 223-41.
- Guhlin, M. (2007), "The case for open source", *Technology & Learning*, Vol. 27 No. 7, pp. 16-21.
- Harrower, M. and Brewer, C.A. (2003), "ColorBrewer.org: an online tool for selecting colour schemes for maps", *Cartographic Journal*, Vol. 40 No. 1, pp. 27-37.
- Kinikin, J. and Hench, K. (2005), "Follow-up survey of GIS at smaller academic libraries", *Issues in Science & Technology Librarianship*, Vol. 43 No. 1, available at: www.istl.org/05-summer/index.html (accessed December 22, 2008).
- Krishnamurthy, M. (2008), "Open Access, open source and digital libraries", *Program: Electronic Library and Information Systems*, Vol. 42 No. 1, pp. 48-55.

- Longley, P.A. (2008), "Geographic information systems (GIS)", in Kemp, K.K. (Ed.), *Encyclopedia of Geographic Information Science*, Sage Publications, Thousand Oaks, CA, pp. 190-4.
- Michalec, M. and Welsh, T.S. (2007), "Quantity and authorship of GIS articles in library and information science literature, 1990-2005", *Science & Technology Libraries*, Vol. 27 No. 3, pp. 65-77.
- Mitasova, H. and Neteler, M. (2004), "GRASS as open source free software GIS: accomplishments and perspectives", *Transactions in GIS*, Vol. 8 No. 2, pp. 145-54.
- Mitchell, T. (2005), *Web Mapping Illustrated*, O'Reilly Media, Sebastopol, CA.
- Morris, S.P. (2006), "Geospatial web services and geochiving: new opportunities and challenges in geographic information services", *Library Trends*, Vol. 55 No. 2, pp. 285-303.
- Neteler, M. and Mitasova, H. (2008), *Open Source GIS: A GRASS GIS Approach*, 3rd ed., Springer, New York, NY.
- Pohlmann, F. (2005), "Find your way around open source GIS", *IBM Developer Works*, available at: www.128.ibm.com/developerworks/library/os-gis/ (accessed December 19 2008).
- Ramsey, P. (2007), "The state of open source GIS", *Refractions Research White Papers*, available at: www.refractions.net/expertise/whitepapers/opensourcesurvey/survey-open-source-2007-12.pdf (accessed December 19, 2008).
- Raymond, E.S. (2001), *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, 2nd ed., O'Reilly Media, Sebastopol, CA.
- Ritterbush, J. (2007), "Supporting library research with LibX and Zotero: two open source Firefox extensions", *Journal of Web Librarianship*, Vol. 1 No. 3, pp. 111-22.
- Schweik, C., Fernandez, M., Hamel, M., Kashwan, P., Lewis, Q. and Stepanov, A. (2007), "Reflections of an online geographic information systems course based on open source software", working paper, National Center for Digital Government, Amherst, MA, available at www.umass.edu/digitalcenter/research/workingpapers.html (accessed March 20, 2009).
- Sherman, G.E. (2008), *Desktop GIS: Mapping the Planet With Open Source Tools*, Pragmatic Bookshelf, Raleigh, NC.
- Steiniger, S. (2008), "An overview of free and open source desktop GIS (FOS-GIS)", available at: www.spatialserver.net/osgis/ (accessed January 22, 2009).
- Steiniger, S. and Bochner, E. (2009), "An overview on current free open source desktop GIS developments", *International Journal of Geographical Information Science*, forthcoming, available at: www.spatialserver.net/osgis/ (accessed January 22, 2009).
- Tufto, J. and Cavallini, P. (2005), "Should wildlife biologists use free software?", *Wildlife Biology*, Vol. 11 No. 1, pp. 67-76.
- University of Waterloo Map Library (2007), "GIS usage survey", available at: <http://blogs.lib.uwaterloo.ca/maplibrary/category/gis-usage-survey-results/> (accessed December 22, 2008).
- Wade, T. and Sommer, S. (2006), *A to Z GIS: An Illustrated Dictionary of Geographic Information Systems*, ESRI Press, Redlands, CA.
- Weber, S. (2004), *The Success of Open Source*, Harvard University Press, Cambridge, MA.
- West, J. (2007), "Open-source software on the desktop", *Library Technology Reports*, Vol. 43 No. 3, pp. 40-7.

About the author

Francis P. Donnelly (Frank) is an Assistant Professor and the Geospatial Data Librarian at the William and Anita Newman Library at Baruch College, CUNY, where he has served since September 2007. He is the subject specialist in geography, demography, and GIS and is responsible for building a collection of monographs and maintaining a campus-wide repository of GIS data. He offers instruction, consultations, and support services to students and faculty who are interested in GIS and demographic research, and he has extensive experience working with US Census data. He holds Master's degrees in both Geography and in Library and Information Science. Prior to coming to Baruch he worked for the University of Washington Libraries in Seattle for two years, and spent three years as a socio-economic researcher and GIS analyst in the non-profit and government sectors. Francis P. Donnelly can be contacted at: francis.donnelly@baruch.cuny.edu