A comparison of open source geospatial technologies for web mapping

Andrea Ballatore and Ali Tahir

School of Computer Science and Informatics,
University College Dublin,
Belfield, Dublin 4, Ireland
E-mail: andrea.ballatore@ucd.ie
E-mail: ali.tahir@ucd.ie

Gavin McArdle*

National Centre for Geocomputation,
National University of Ireland Maynooth,
Maynooth, Co. Kildare, Ireland
E-mail: gavin.mcardle@nuim.ie
*Corresponding author

Michela Bertolotto

School of Computer Science and Informatics,
University College Dublin,
Belfield, Dublin 4, Ireland
E-mail: michela.bertolotto@ucd.ie

Abstract: The past decade has witnessed a steady growth of open source software usage in industry and academia, leading to a complex ecosystem of projects. Web and subsequently geographical information systems have become prominent technologies, widely adopted in diverse domains. Within this context, we developed an open source web platform for interoperable GIServices. In order to implement this architecture, 14 projects were selected and analysed, including the client-side libraries and the server-side components. Although other surveys have been conducted in this area, little feedback has been formally obtained from the users and developers concerning their opinion of these tools. A questionnaire was designed to obtain responses from the relevant online communities about a given set of characteristics. This article describes the technologies and reports the results of the survey, providing first-hand information about open source web and geospatial tools.

Keywords: open source; free and open source software; FOSS; geographical information systems; GIS; web technologies; web architecture; spatial web services; web mapping; online survey; spatial DBMS; GIServices; web graphical user interfaces.

Reference to this paper should be made as follows: Ballatore, A., Tahir, A., McArdle, G. and Bertolotto, M. (xxxx) 'A comparison of open source geospatial technologies for web mapping', *Int. J. Web Engineering and Technology*, Vol. X, No. Y, pp.000–000.

Biographical notes: Andrea Ballatore is a PhD student in the School of Computer Science and Informatics at University College Dublin. His research interests include location-based services, human-computer interaction, geographic information systems and implicit feedback analysis. He received his MSc in Virtual Reality and Multimedia from the University of Turin, Italy, in 2006. In 2007, he worked as a Software Developer of large-scale information systems for a major Italian insurance company. From 2007 until 2009, he worked for Design Company 3dcadsoft in Dublin, developing and maintaining CAD tools for interior design.

Ali Tahir is a postgraduate research student in the School of Computer Science in Informatics at University College Dublin. He received his BSc in Computing and Information Systems from Oxford Brookes University, UK, in 2004 and subsequently MSc in Geographical Information Science from University of Nottingham, UK, in 2006. Afterwards, he has worked as a Junior Lecturer in National University of Sciences and Technology (NUST) in Pakistan until 2009. His research activities include location-based services and geo-visualisation.

Gavin McArdle received his BSc in Computer Science in 2003 from University College Dublin (UCD). He subsequently completed his PhD degree in 2008, where he investigated the benefits of using virtual reality environments for e-learning. In 2008, he began his Postdoctoral Research position in UCD, where he investigated map personalisation by utilising user preferences. Since June 2010, he has been a Postdoctoral Research Fellow at the National Centre for Geocomputation at NUI, Maynooth, examining visual techniques for analysing large movement datasets. His research interests include geo-visualisation, personalisation techniques and location-based services.

Michela Bertolotto received her PhD in Computer Science from the University of Genoa, Italy, in 1998. Subsequently, she worked as a Postdoctoral Research Associate in the Nation Center for Geographic Information and Analysis (NCGIA) at the University of Maine. Since 2000, she has been a faculty member at the School of Computer Science and Informatics of University College Dublin. Her research interests include web-based and wireless GIS, spatio-temporal data modelling, and 3D data handling and interfaces.

1 Introduction

The production and usage of free and open source software (FOSS) has grown considerably over the past decade, mainly due to its improved quality and economic factors (Deshpande and Riehle, 2008). FOSS combines the notion of free software and open source software. Free software relates to the user’s freedom to run, copy, distribute, study, change and improve the software (Stallman, 2009). Open source software is generally distributed under licences approved by the open source initiative (OSI).

Within the open source ecosystem, web technologies have emerged as one of the dominant domains. This has occurred due to the advent of the Web 2.0 in the last decade, which required free, interoperable, standard-compliant software components. As a result, several open source projects challenged commercial competitors and reached a considerable user base.

In parallel, desktop-based geographical information systems (GIS) have successfully merged with Web 2.0 technologies resulting in web mapping, which has become a ubiquitous feature of online services. The availability of open geographical datasets from sources such as OpenStreetMap has fuelled the development of a large number and variety of open source web mapping projects, contributing an alternative to well-established commercial services like Google Maps. Within this context, the open-geospatial consortium (OGC) has proposed standards for geographical data and facilitated the design of interoperable open source GIS tools. With such an abundance of different technologies available, it is difficult for a developer to weigh the merits of each approach. Although there is a body of theoretical work in this area, the speed of development has created a gap in the knowledge regarding the practical aspects of developing open source web-based geospatial data delivery services.

While implementing a complete web platform for providing open GIServices (McArdle et al., 2010), we analysed several open source projects. In order to make an informed decision regarding which technologies to adopt, information was collected from several web communities and the academic literature. To reduce the knowledge gap regarding real-life experiences of software development with these projects, an online survey was devised to collect opinions from the communities of users and contributors, which constitute an essential part of the open source phenomenon.

The goal of this article is to describe the selected projects and discuss the results of the online survey, giving the readers an indication of the characteristics of these open source tools. The results highlight the strengths and weaknesses of the projects being studied and will serve as a useful guide to the developers in understanding how this geospatial open source ecology is evolving and furthermore assist them in determining which open source technologies to use. As a result, developers no longer have to rely solely on the information provided by the websites of such open source projects, which of course can be biased and self-promoting.

This paper is organised as follows: Section 2 discusses the related work. A description of the open source projects is presented in Section 3. Section 4 describes the methodology of the online questionnaires, the results of which are detailed in Section 5. Finally, the conclusion is reported in Section 6.

2 Related work

In recent years, there has been a growing interest in the different aspects of the open source phenomenon. This interest is not restricted to the traditional computer science disciplines, such as software engineering, but also embraces human-computer interaction. Nichols and Twidale (2003) have conducted a qualitative analysis on the characteristics of open source projects in terms of their usability. A survey about usability by Andreasen et al. (2006) investigates practices and opinions among contributors via online questionnaires, while an empirical study captured the usability factors in open source projects from the contributors' perspective (Raza and Capretz, 2010).

Given the unusual makeup of the open source production system, consisting of paid and volunteered contributions, economists are interested in understanding and modelling its features. Deshpande and Riehle (2008) identify an exponential growth in the number of open source projects since 2000. Bitzer et al. (2007) have studied the motives driving the contributors to devote time and resources towards open source development, while

Bonaccorsi and Rossi (2003) have investigated the discussion of open source technologies.

As the wide-spread adoption of open source GIS technologies continues, interest in understanding and classifying their features has increased. A general review of free and open source technologies for geomatics is offered by Sanz-Salinas and Montesinos-Lajara (2009), in which the links between projects are highlighted. Other studies focus solely on open source desktop tools in terms of functionalities (Sherman, 2008; Steiniger and Bocher, 2009), while Steiniger and Hay (2009) provide a comparative study between open source and commercial desktop GIS packages.

The use of open source web mapping is growing in other disciplines, such as medicine (Boulos et al., 2010), astronomy (Terazono et al., 2009) and sociology (Ball, 2009). Despite this trend, there is no comprehensive study analysing the projects forming an open source web mapping architecture. In our previous work, we depicted a web architecture for map personalisation (McArdle et al., 2010). A set of interoperable open source projects contribute to the development of a platform able to offer functionalities through open web services. In this context, it was necessary to analyse a wide number of open source projects, to determine their suitability for the development of our web mapping architecture, including its three tiers (client tier, middle tier and data sources).

The research in this paper presents this process, which included information gathering from literature and uniquely combined this with feedback obtained from contributors and developers through an online questionnaire. The next section describes the 14 open source projects selected for this survey.

3 Technology review

During the implementation of the architecture described by McArdle et al. (2010), a range of technologies were examined and categorised into seven areas as follows: web GUI libraries, AJAX libraries, web mapping servers, web application frameworks, object-relational mapping, spatial libraries, spatial DBMS. Two projects for each of the seven categories were identified as suitable for developing a web mapping architecture and these were studied further. In total 14 projects, OpenLayers, Ext JS, Prototype, MooTools, GeoServer, MapServer, GeoTools, Java Topology Suite (JTS), Ruby on Rails, Grails, Hibernate, Hibernate Spatial, PostGIS, MySQL, were examined and included in the survey described in this article.

The projects are not in direct competition, as they are often inter-dependent. While in some situations, a technology can be a clone of another (e.g., Ruby on Rails and Grails), in other cases it can be a more specialised spatial extension of another (e.g., Hibernate and Hibernate Spatial). Other projects which appear similar may in fact be responsible for different functional areas and so are not competing with each other (e.g., OpenLayers and Ext JS). This survey does not attempt to cover all the open source projects currently active in the geospatial area but focuses on the specific domain of web mapping. The online communities associated with these projects are one of the most interesting aspects of the open source software projects.

Table 1 Projects overview including project owner, year of foundation, estimate of number of users and contributors

Project	*Owner*	*Year*	*License*	*# users*	*Website and users*
OpenLayers	(MetaCarta until 2007) Open Source Geospatial Foundation	2006	BSD-like	1,450	http://openlayers.org MetaCarta
Ext JS	Ext	2006	GNU GPL/Commercial	127,000	http://www.extjs.com Adobe, Amazon.com, Microsoft, Sony
Prototype JS	The Prototype Core Team, 37 Signals	2006	MIT	3,200	http://www.prototypejs.org Apple, CNN.com, Nasa ,Twitter
MooTools	The MooTools Dev. team	2005	MIT	2,500	http://mootools.net Jeep, Ferrari
MapServer	University of Minnesota	1996	X/MIT	1,880	http://mapserver.org Camptocamp, WebMapIt, Syncera IT Solutions
GeoServer	Lime Group, The Open Planning Project, Refractions Research	2001	GNU GPL	-	http://geoserver.org Alkante, Cramptocamp, Landgate
GeoTools	Open Geospatial Consortium	1996	GNU LGPL	1,059	http://www.geotools.org Vision of Britain
JTS	Martin Davis	2000	GNU LGPL	-	http://www.vividsolutions.com/jts GeoConnections
Ruby on Rails	David Heinemeier Hansson, 37 Signals	2004	MIT	20,000	http://rubyonrails.org Basecamp, Twitter, Yellow Pages
Grails	SpringSource (WMWare)	2006	Apache v2.0	4,600	http://grails.org Sky.com Wired.com
Hibernate	Red Hat	2001	GNU LGPL	57,000	http://www.hibernate.org UnionBank, Warner Music Group
Hibernate Spatial	GeoVise	2003	GNU LGPL	100	http://www.hibernatespatial.org N.A.
PostGIS	Refractions Research	2001	GNU GPL	1,790	http://postgis.refractions.net GlobeXplorer, Institut Gographique
MySQL	Sun Microsystems	1994	GNU GPL	-	http://www.mysql.com Twitter, LinkedIn, United Nations FAO

Table 1 reports the year the project started, the software license, well-known corporate and academic users and the link to the official project websites (accessed on 24 February 2011). Each project provides an official forum (or a mailing list) for general users and one devoted to the project developers and contributors. The number of users and developers presented in the table is based on the number of registered members on such forums. Certain forum administrators, such as those of MySQL, did not disclose the number of users. In the following sections (3.1 to 3.7), an overall description of the 14 projects is presented. Section 4 presents the methodology of the online questionnaire.

3.1 Web GUI libraries

Thanks to the diffusion of AJAX-oriented web technologies, web GIS has experienced remarkable growth over the past decade. Well-known services such as Google Maps and Virtual Earth provide developers with powerful and free web tools. Although these products are very popular, they are fully controlled by companies that can decide to alter or discontinue them arbitrarily. Furthermore, there is no clear separation between the visualisation tools and data, the two are mixed in a seamless way and so the GUI cannot be customised. On the contrary, open source mapping libraries offer viable alternatives to build powerful interactive web maps, allowing the developer to choose the data sources and formats. In this survey, OpenLayers, a popular open source web mapping project, and Ext JS, a library to build complex Web user interfaces are assessed.

OpenLayers is a JavaScript library for displaying spatial data in web browsers, without server-side dependencies. OpenLayers implements a JavaScript API for building rich web-based geographic applications, constituting the main open source alternative to the equivalent commercial tools. As with many open source geospatial technologies, OpenLayers implements industry-standard methods for geographic data access, such as the OGC web map service (WMS) and web feature service (WFS) standards. The architecture of the framework is written in object-oriented JavaScript, using components from prototype described by Orchard et al. (2009). At the conceptual level, OpenLayers aims to separate spatial visualisation and manipulation tools from spatial data, an idea that is not implemented in the commercial equivalent. OpenLayers is also widely used to display OpenStreetMap data (http://www.openstreetmap.org, accessed on 24 February 2011).

Ext JS, described in Orchard et al. (2009), is a cross-browser JavaScript library for building web interfaces. It includes a set of user interface widgets, an extensible component model and an API. The Ext JS library and its related products have experienced success recently, becoming ubiquitous in complex web applications. The library is currently used by several major international corporations for intranet and internet websites. Ext JS can work as a standalone library and can integrate with the Prototype Javascript library. The work has been extended by a project called GeoExt which embeds OpenLayers with Ext JS to provide a framework to build desktop-like web GIS applications (http://www.geoext.org, accessed on 24 February 2011).

3.2 AJAX libraries

Before the advent of AJAX, interaction with web pages was a synchronous cycle of request/response between the client and the server. AJAX, meaning asynchronous JavaScript + XML, is not a specific technology but an approach that can be implemented with diverse technologies, whose name and general principles were defined in a seminal article by Garrett (2005). Such an approach introduced an intermediated layer between client and server, letting the client request data for a specific subsection of the content, adding asynchronous calls between the atomic synchronous calls. Although the idea was first presented in 1992, it was not until Google released Gmail and Google Maps in the middle of the last decade that AJAX went mainstream. This approach adopted in Google Maps opened several possibilities to the web GIS field, which thanks to AJAX has grown considerably.

In this survey, we have selected two of the most popular Javascript libraries that implement the principles of AJAX to address diverse needs of web development: prototype and MooTools which have both been described by Orchard et al. (2009).

Prototype is a framework that provides functionality to assist JavaScript development. It provides several features, ranging from programming shortcuts to functions for dealing with AJAX requests. The framework extends object-orientation in JavaScript and introduces a class-based design. In contrast to other JavaScript frameworks like jQuery, prototype extends the W3C DOM, which is controversial and debated in the web developer community and will probably undergo a major revision. Prototype is currently being used in a number of open source and commercial projects to implement AJAX patterns and build JavaScript classes. The library is prominent on the web, where it is used by Apple, Microsoft and other major groups in their web applications.

My object-oriented tools (*MooTools*) are also a web-application framework to support JavaScript development (Newton, 2008). MooTools is based on the traditional object-oriented programming paradigm, with a similar class and inheritance structure as Java. This makes the framework intuitive to programmers with an object-oriented background. MooTools emphasises modularity and code reuse which can be customised depending on the application being developed. There are many advantages of using MooTools, of particular note are the built-in methods to handle AJAX requests, CSS and DOM elements. Additionally, at present, MooTools works independently of specific browsers and operates across an array of popular platforms. It is used by several corporations, including Jeep and Ferrari for their web applications. Like prototype, MooTools permits the easy integration of interactivity into a webpage.

3.3 Web mapping servers

Web mapping servers act as a framework to publish a GIS application online. Generally, these servers include functionality to query spatial DBMS, projection support, integration with other geographic libraries as well as vector and raster support. Moreover, interoperable web standards have been developed by the OGC in support of web mapping, including WMS, WFS, web coverage service (WCS), geography markup language (GML) and styled layer descriptor (SLD). WMS and WFS are widely used services to request maps and get information about the geographic features. These OGC standards have allowed numerous projects to be published online since the beginning of the FOSS movement (Sanz-Salinas and Montesinos-Lajara, 2009). MapServer and GeoServer have been included in this survey.

MapServer, described by Kropla (2005), is a widely used open source web mapping project for developing interactive and interoperable web GIS applications. Fully compliant with various OGC standards such as WMS, WFS, WCS and SLD, MapServer supports an extensive variety of vector and raster formats. MapServer runs as a common gateway interface (CGI) application with Apache and Microsoft web servers. MapServer has strong cartographic support and dynamic capabilities. A map file defines styles and symbology that are used to render a fully customised map. MapServer also has built-in support for many scripting languages such as Perl, Java, .NET, PHP and direct databases support for PostGIS, Oracle Spatial and MySQL.

GeoServer, developed more recently than MapServer, is a popular open source GIS project. It has a web administration tool to configure its spatial and non-spatial options. Apart from WMS, WCS and WFS, it additionally supports the editing of feature services on the client side using WFS-transactional. GeoServer is built on Java technology and runs on an integrated Jetty web server. It heavily relies on GeoTools (covered in Section 3.4). A significant feature of GeoServer is the use of SLD, an OGC standard to render the visual style of the map. Other features include an integrated OpenLayers, Google Earth support for overlays using keyhole markup language (KML), GeoWebCache for tile mapping and wide support for many DBMS like PostGIS, ArcSDE, Oracle and DB2. A detailed description of GeoServer and its applications are covered by Erle et al. (2005).

3.4 Spatial libraries

Spatial libraries provide spatial software components which can be exploited in desktop and web GIS applications. There are several open source spatial libraries available online. Proj. 4, an open source cartographic projection library is one such example that is widely used by many client and server side GIS applications, including MapServer (Section 3.3). Other examples such as geometry engine-open source (GEOS) define the topological predicates and operations required to render a map. According to Cậmara et al. (2009), the popularity of these libraries is due to their quality and stability. This survey covers GeoTools and the JTS.

GeoTools is an open source Java library that provides advanced GIS functionalities. It supports vector and raster geospatial data formats, DBMS access and rendering of complex maps. Being one of the oldest projects of Open Geospatial Foundation, it is fully compliant with OGC specifications including GML, WMS, WFS, grid coverage, coordinate transformation and SLD. The JTS, described in Section 3.4, can be used in conjunction with GeoTools as a geometry model for vector features. This library can be used in client and server-side geospatial applications. It is currently used by GeoServer (Section 3.3) for a range of geographic tasks. Turton (2008) provides a history and an overview of GeoTools.

JTS is a widely used open source Java library for handling 2D geometrical operations. It conforms to the geometry model and API defined in the simple features specification (SFS) proposed by OGC. JTS provides access to simple functions such as buffering, overlays (intersection, union and difference) as well as more advanced functionalities like spatial algorithms and structure support (spatial indexing, planar graph framework, geometry simplification, Delaunay triangulation and precision reduction). This library is used in a range of open source geospatial projects including GeoTools, PostGIS, GeoServer and Hibernate Spatial. Regnauld (2006) and Crooks (2006) refer to the JTS.

3.5 Web application frameworks

As part of the Web 2.0 transition, depicted by O'Reilly (2005), software applications have been moving to the World Wide Web. Developing powerful web applications has become a prominent activity in the software industry during the past decade, recently reaching the GIS field. From a software engineering perspective, the Agile methods and practices surveyed by Martin (2003) emphasise adaptation to these changing circumstances, flexibility, simplicity, usability and user-centred design. Agile development principles have inspired the design of a number of successful open source projects that have proven to be particularly effective in such a mutable and competitive context. For this survey, two leading frameworks for web development, Ruby on Rails and Grails, have been reviewed.

Ruby on Rails is a framework for a rapid web application development based on the model-view-controller paradigm. The project also enforces design principles such as convention over configuration and do not repeat yourself (DRY), lowering repetitive operations in the development cycle, driving the developer's attention to the business logic beyond the platform details. The framework models the entire application, from the web pages down to the underlying relational database, enforcing a rigid structure. The main language used in the framework is the dynamic language Ruby, created by Yukihiro Matsumoto blending parts of Perl, Smalltalk, Eiffel, Ada and Lisp. Ruby's syntax is supposed to map directly natural language and tends to be concise and without frills. Among the packages that compose the framework, ActiveRecord provides an object-relational mapping system for database access, while ActiveResource facilitates the creation of web services. Ruby on Rails has no native spatial support, although it is possible to map spatial data manually to a spatially-enabled database such as MySQL. Thomas and Hansson (2009) have compiled an authoritative and extensive guide for using Ruby on Rails.

Grails is a web application framework that mirrors closely the basic design ideas of Ruby on Rails. The main language of Grails is Groovy, a dynamic language for the Java Virtual Machine, designed to be extremely easy to learn for Java developers. Groovy shares a lot of design principles with Ruby but with a Java-like syntax. The architecture of Grails is built on Java-based technologies and claims full compatibility with Java, which is a relevant advantage over Ruby on Rails. Moreover, Grails handles the object-relational mapping with Hibernate (discussed in Section 3.6). Although Grails has no native spatial support, its compatibility with Java enables the use of several spatial tools, such as the Java topology library and Hibernate Spatial, discussed in Section 3.4 and Section 3.6, respectively. A comparison model for Rails-based web development frameworks has been proposed by Fernández-Villamor et al. (2008), while Rocher et al. (2009) have written an extensive and authoritative manual for Grails.

3.6 Object-relational mapping

Typically, developers utilise databases to store and maintain persistent objects. However, for this strategy to be successful, consideration of how objects are mapped to the relational database management system (RDBMS) must be considered. Databases are typically designed to store data, objects are inherently more complex, consisting of data combined with state and behaviour and so cannot be stored directly in the database. One of the powers of the object-oriented paradigm is the inheritance model which offers the opportunity for code reuse. This differs from RDBMS whose primary purpose is to remove redundancy among stored data rather than model real world entities and behaviour.

Within GIS applications, handling the mapping between the object-oriented paradigm and the relational model is extremely important due to the complex spatial objects which require efficient storage in databases. Several frameworks have been developed to handle this mapping by providing an interface to assist developers, hiding low-level implementation details. In this survey, Hibernate and its spatial extension, Hibernate Spatial, are investigated.

- *Hibernate* provides a framework for mapping an object-oriented domain model, in Java, to the traditional relational database paradigm (Bauer and King, 2004). There are always conceptual and technical difficulties when trying to map objects or classes in Java to relational tables. Hibernate solves these issues by using high-level object handling functions. In addition to handling this, Hibernate can map Java data types to SQL data types which permits Hibernate to include data query and retrieval facilities by generating appropriate SQL queries. This benefits the developer as it removes the task of handling the result set and any object conversions required by the query. Furthermore, one of the main benefits is that it enables applications to function with all the support of an SQL database. In this way, Hibernate permits the development of persistent classes following the object-oriented paradigm, including association, inheritance, polymorphism, composition and collections while providing an interface between Java and any underlying SQL relational database.
- *Hibernate Spatial.* While Hibernate natively supports common data types, spatial entities and geometries are not explicitly handled and require the use of a plug-in called Hibernate Spatial. Hibernate Spatial allows the mapping of these spatial objects to spatial databases which facilitates the use of indexes and optimisation. In a similar way to Hibernate, Hibernate Spatial abstracts from the way a specific database supports geographic data, and provides a standardised, cross-database interface to geographic data storage and query functions. Additionally, Hibernate Spatial supports most of the specifications of the OGC simple features interface standard making it a useful addition for any Java-based geospatial web application using spatial databases. Hibernate Spatial can be integrated with Grails to handle a transparent mapping to PostGIS and other spatial DBMSs. The successful use of this framework has been demonstrated by Silva et al. (2009).

3.7 Spatial DBMS

A spatial DBMS offers spatial data types in its data model and query language, provides spatial indexing and algorithms for spatial queries (Shekhar and Chawla, 2003). Oracle Spatial, PostGIS and MySQL are some of the popular Spatial DBMS packages in this domain. Advancement and awareness in geospatial technologies has given rise to standardisation and as a result most Spatial DBMS conform to the OGC SFS. This standard is widely followed by almost all SQL databases and provides a way to access and store geographic data types along with related functions and data operations. In this survey, PostGIS and MySQL are examined.

PostGIS is a spatial extension of the PostgreSQL database management system and is a well-known open source project for storing and querying geographical entities. PostGIS is widely supported by a large number of proprietary and open source web mapping servers. It is fully compliant with OGC standards such as SFS for SQL. Apart from the basic spatial support, PostGIS also has topology, data validation, coordinate transformation and programming API functionalities. As the project continues to grow, additional features for raster support, routing, three dimensional surfaces, curves and splines have been proposed. Obe and Hsu (2009) provide the most authoritative reference guide to PostGIS.

MySQL is a widely used open source RDBMS, which, since version 4.1, includes support for spatial extensions using an additional geometry column for the storage and analysis of geographic features. These features could be any entity having more than one dimension and associated locational information. MySQL fully implements the SFS and supports R-tree indexing to speed up database operations. It operates on multiple platforms and constitutes a key component of the open source LAMP framework (Linux, Apache, MySQL, PHP/Perl/Python), thus making this a popular open source project. Widenius et al. (2002) offer a reference guide to MySQL.

4 Online survey

First-hand knowledge regarding open source projects was obtained from active developers through online questionnaires (Wright, 2005). Given the nature of open source online communities, such surveys are a particularly suitable way of getting access to the community members. For this purpose, we have designed an anonymous questionnaire with 13 questions, three of which cover the respondent (contribution, affiliation and level of expertise), while ten are related to the characteristics of the project being analysed (learning curve, stability, performance, scalability, interoperability, extendibility, standards, documentation, community support, and frequency of updates). The full questionnaire is reproduced in Appendix.

Each question on the software is measured on a five-point discrete visual analogue scale (Uebersax, 2006), 1 being the minimum score and 5 the maximum. It is important not to force the responder to give an answer when it is not relevant or interesting to them, so each answer includes a ‘no answer’ option. In order to maximise the amount of responses, each questionnaire was designed to be completed in less than 90 seconds. Each questionnaire focuses on a specific project, asking the same questions for all the projects, which allows the responses to be compared in a quantitative way. A web home page, presenting a short description of the survey to the public and linking the 14 questionnaires, was created and published online on 15 April 2010. This page has been disseminated through several channels with an invitation to take the questionnaires for the technologies that the responder has used directly. The community members are encouraged to take the survey purely to contribute to academic research without any economic incentive. The announcement was posted on the official forums and mailing lists of each project. The questionnaires were subsequently published on boards for software developers, such as Stackoverflow (http://stackoverflow.com, accessed on 24 February 2011) and on high-activity academic mailing lists such as DBWorld (http://www.cs.wisc.edu/dbworld, accessed on 24 February 2011). Responses were received from 15 April 2010 to 15 June 2010.

The opinions collected in this survey represent the subjective and qualitative perception of a number of users in relation to open source online projects. Although online surveys based on self-selection are a cost-effective tool, they have some drawbacks (Wright, 2005). For example, the demographics are self-assessed, so a responder could easily provide false information about their affiliation and contribution to the project. However, given the context, this is unlikely to happen because there would no benefit for the responder.

The responders intrinsically form a non-probability sample whose representative accuracy is difficult to measure. The object of this survey is a set of open source projects and not their online communities, therefore, the collected feedback still offers valuable information about the software characteristics.

Our questionnaire is based on discrete visual analogue scales, which can be interpreted in different ways by different responders. In order to minimise this problem, the scales were labelled as clearly as possible, avoiding jargon and uncommon words, which could be problematic for non-native English speakers.

The responses obtained from the online survey are described in Section 5 mainly through average values. When using traditional Likert scales to obtain feedback from respondents, the average value does not provide a meaningful description of the results because the middle answer is usually labelled neither agree nor disagree. On the contrary, the middle answer of the visual analogue scales used in our online questionnaire represents the actual mid-point of the scale (3 on scale from 1 to 5), with a separate no answer option. Therefore, in this questionnaire average values will show overall trends in the results.

In the literature, several categories of bias have been identified in online surveys. For example, a central tendency bias occurs when respondents avoid extreme responses (Evans and Mathur, 2005). Similarly, a positive ideological bias has been identified within open source communities (Hars and Ou, 2002). However, these aspects were beyond the scope of our research, further statistical analysis could help understand them, for instance by taking into account the correlation between the responder's contribution and the scores they express (e.g., an attempt at publicising a project the responder has worked on).

Although these drawbacks exist, the questionnaire results provide valuable indicators about the project characteristics based on the users' perception. The statistical analysis and the result discussion are presented in Section 5.

5 Results

An overview of the analysis performed on the survey results is presented in Section 5.1, including the demographics and the overall scores. Section 5.2 focuses on a visual representation of the results for each project.

5.1 Overview

In total, the 14 questionnaires obtained 301 responses. Due to the nature of online surveys, it is not possible to generalise the results from the sample to the entire relevant community. Online surveys based on self-selection produce non-probability samples (Wright, 2005). Although these results might not express valid general knowledge on those online communities, which are not the subject of our survey, they can be used to reinforce a qualitative description of the projects being examined. In this analysis, the responses are treated as independent. Figure 1(a) shows the distribution of responses over the different projects.

There seems to be no correlation between the community size and the number of responses. In certain cases, few members of large communities have responded (e.g., Ext JS), while small communities showed a higher interest in the survey, perhaps in the attempt to increase the visibility of their project (e.g., Hibernate Spatial). Given the uneven distribution of number of responders, which varies from 57 to 4, it is beneficial to display the standard deviation [Figure 1(b)]. The standard deviation and the correspondent standard error in this context represent a measure of precision of a specific project. Intuitively, the precision of a score calculated over a high number of responses is likely to be higher than that over a very small sample.

Figure 1 (a) Number of responders [total: 301] (b) standard deviation per project (see online version for colours)

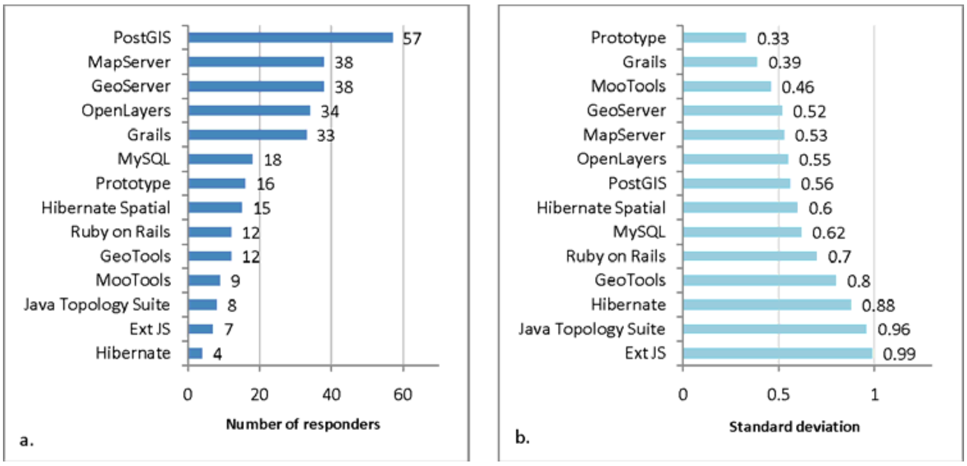
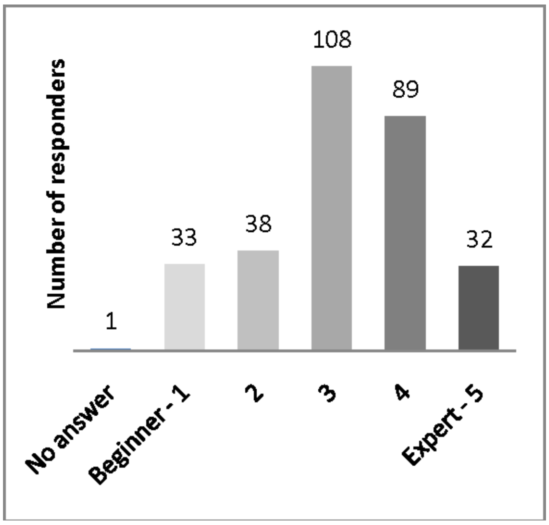


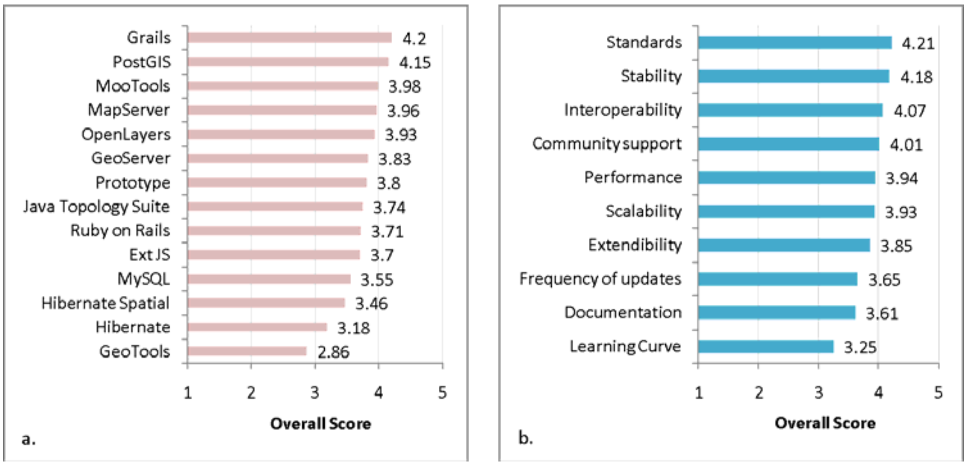
Figure 2 Level of expertise distribution [total: 301]



In terms of demographics, we consider three self-assessed characteristics of the responders: affiliation, contribution and level of expertise. Of the 301 responders over all 14 projects, 54 have said they are contributors (18%), while 244 are non-contributors (81%) and 3 did not reply (1%). Similarly, 62 consider themselves as affiliated to the project (20%), 231 non-affiliated (77%) and 8 did not answer (3%). The levels of expertise have the distribution displayed in Figure 2. It is possible to observe that most responders evaluated their expertise in relation to the project as 3/5 (36%) or as 4/5 (29%).

Figure 3(a) represents the scores of all the projects sorted from the highest to lowest value. The score of a project is the average of the average of each response. Grails, PostGIS and MooTools have obtained the highest score, while Hibernate Spatial, Hibernate and GeoTools the lowest.

Figure 3 (a) Overall scores for projects (b) overall scores for characteristics (see online version for colours)



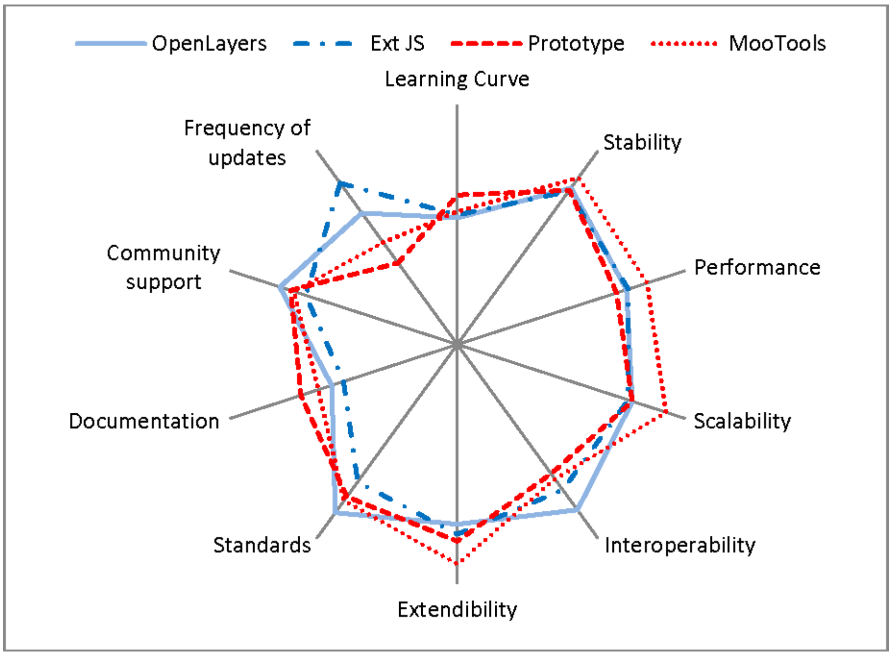
The score of a characteristic is the average of the responses across all the projects. Figure 3(b) summarises these scores sorted from the highest to lowest value. Support for standards, stability and interoperability have been ranked the highest. The responders have deemed the frequency of updates, the quality of documentation and the learning curve to be the least satisfactory characteristics. This result confirms the low priority that usability has for open source developers, who generally devote more resources to other goals, such as software quality (Andreasen et al., 2006; Lee et al., 2009).

5.2 Technology comparison

In the survey, feedback was obtained on 14 projects using the ten characteristics outlined above. The average score of each characteristic was calculated for each individual projects, resulting in 140 scores. In order to visualise these values, which represent non-comparable features on the same scale, radar charts are particularly suitable. Therefore, a project is represented by a polygon whose vertices indicate the average score for the ten characteristics. The origin of the chart, in which all the axis converge, corresponds to the minimum possible value of the score (1), while the outer end points of the axis show the maximum score (5). Therefore, the shape of the polygon reflects the scores: a project that obtained balanced scores tends to be a regular decagon, whilst uneven scores generate spikes and valleys in the polygon. Similarly, a small polygon represents a set of lower scores than a large polygon.

Figure 4 offers a visual representation of the opinions received from the responders in relation to the client-side technologies (OpenLayers, Ext JS, Prototype and MooTools, described in Sections 3.1 and 3.2). When the web GUI libraries are examined, it can be seen visually that they generally follow a similar pattern. The same can be said about the AJAX libraries (Prototype and MooTools). The only noticeable variation is a spike in the frequency of updates, in which Ext JS performed better.

Figure 4 Web GUI and AJAX libraries (see online version for colours)



The results for the web mapping servers and spatial libraries (GeoServer, MapServer, GeoTools and JTS, covered in Sections 3.3 and 3.4) are outlined in Figure 5. For the majority of characteristics, MapServer obtained more positive feedback than GeoServer. The difference between the spatial libraries is more pronounced, as GeoTools received lower scores than the JTS.

Figure 6 provides the radar representation of the results obtained in relation to the server-side technologies (Ruby on Rails, Grails, Hibernate and Hibernate Spatial, described in Sections 3.5 and 3.6). It is evident that Grails received noticeably higher scores than Ruby on Rails, except in relation to stability. Additionally, a valley is present in the community support for Hibernate.

The results related to the spatial DMBSs, covered in Section 3.7, are provided in Figure 7. While both projects are seen as having a similar learning curve, the opinions about PostGIS are noticeably more positive than those of MySQL.

The next section outlines possible directions for future work and draws a conclusion.

Figure 5 Web mapping servers and spatial libraries (see online version for colours)

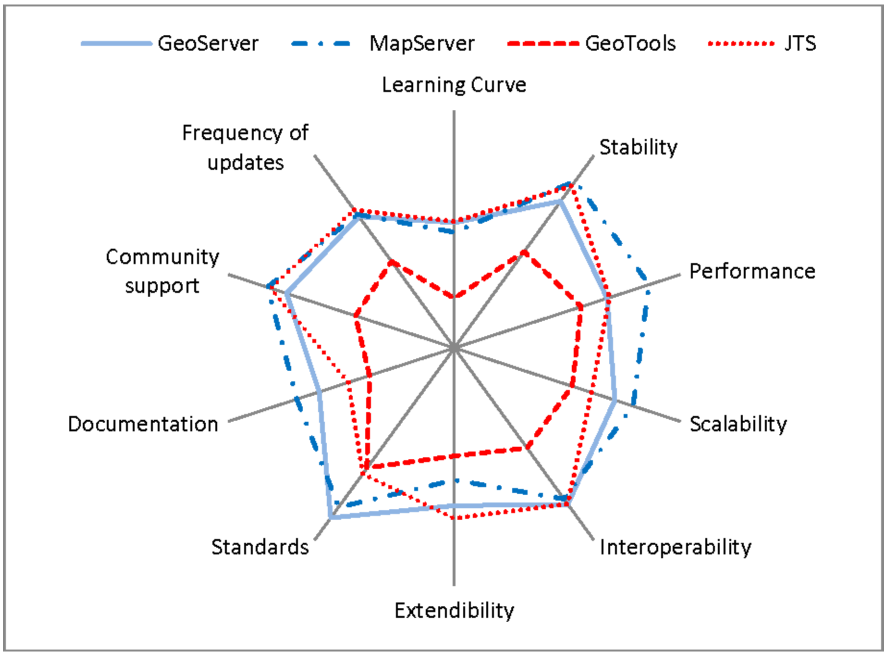


Figure 6 Web application frameworks and object-relational mapping (see online version for colours)

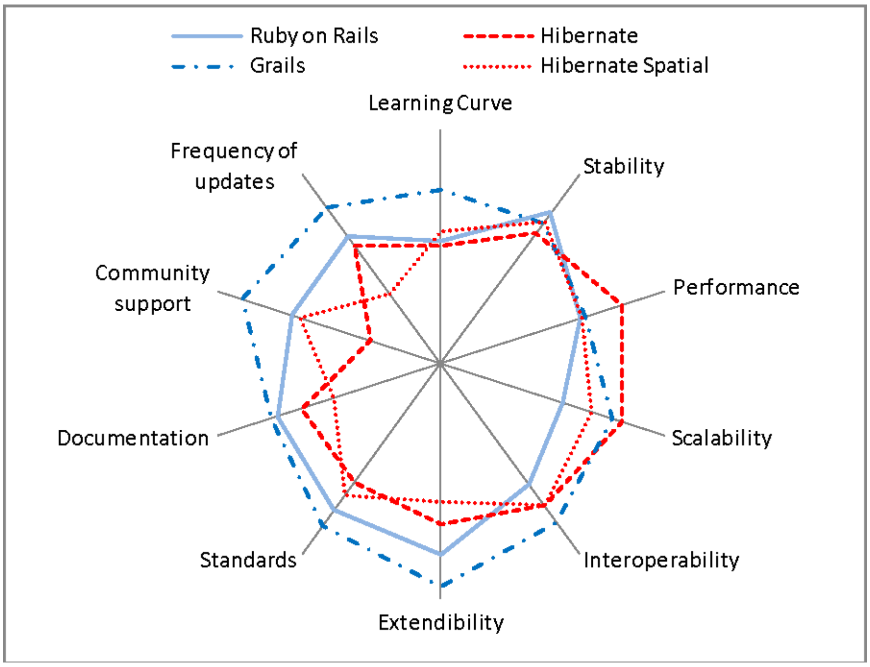
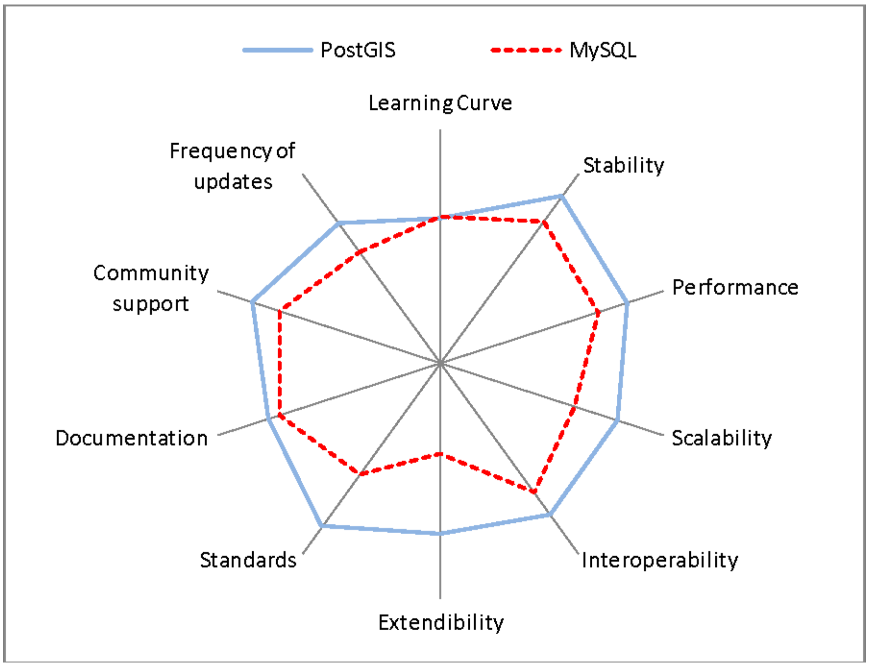


Figure 7 Spatial database management systems (see online version for colours)



6 Conclusions

This article provides an analysis of 14 open source projects in the area of web and GIS technologies. This research was carried out in the context of the development of a web platform for open GIServices (McArdle et al., 2010). In our previous work, we focused on the functionality and architecture of the platform. On the contrary, in this article we present a detailed survey of the open source GIS landscape, focusing on 14 representative web technologies. This study consists of two parts.

Firstly, general aspects of each project were examined through a literature review, which disclosed the lack of comprehensive surveys in the field of open source web GIS technologies. A general overview of the projects is presented, summarising their purpose, features and online community size.

Secondly, an online questionnaire was devised and published to collect the opinions from the relevant online communities of users and developers. Each project was assessed on ten criteria related to their practical use for software development and feedback was received from 301 responders. From the overall scores obtained across all the projects, it is possible to infer a higher satisfaction for the compliance towards open standards and software stability than for the project documentation and the learning curve.

A visual representation of the scores for each project allows the reader to compare their characteristics for use in different application domains. As the use of open source geo-spatial technologies increases, these results are applicable to a wide range of domains where interactive maps are required. Many traditional online services are now being augmented with spatial components. For web developers this poses a challenge, as they need to determine the appropriate technologies to adopt. Through the survey presented in this article, the core technologies are evaluated by 301 real users. The feedback collected offers first-hand information about the characteristics of the projects.

The results can be used by developers in any field who wish to include spatial support in their projects. For example, when evaluating alternative spatial DBMSs, a developer can take into account the results of our survey, which indicate that developers are more satisfied with PostGIS than MySQL. Similar conclusions can be drawn from all 14 technologies analysed. The results of this survey can also be considered by the project contributors to address issues in order to improve the overall developers' experience.

The scores obtained from the responders confirmed the choice of technologies for the implementation of the prototype of our web architecture for open GIServices (McArdle et al., 2010). The web mapping server MapServer, the web application framework Grails and the spatial DBMS PostGIS, which form the core components of our architecture, received a higher overall score from the responders than similar projects (GeoServer, Ruby on Rails and MySQL, respectively).

This work can be extended to include staff from the companies and organisations, thereby providing better control over the responders' demographics. The results obtained through the discrete visual analogue scales can also be analysed using alternative statistical techniques. Thus, the correlation between expertise level and the responses could be investigated, as well as the variation between contributors and non-contributors. Furthermore, the comparison between this study and surveys based on other approaches could reveal interesting trends. The developer's perception contrasted with findings from other sources might also assist in understanding further the relationship between human users and software tools.

References

Andreasen, M., Nielsen, H., SchrØder, S. and Stage, J. (2006) 'Usability in open source software development: opinions and practice', *Information Technology and Control*, Vol. 25, No. 3A, pp.303–312.

Ball, W. (2009) 'Learning the local political landscape with geographic information systems (GIS)', *Journal of Political Science Education*, Vol. 5, No. 3, pp.198–213, Routledge.

Bauer, C. and King, G. (2004) *Hibernate in Action*, Manning, Greenwich, CT, USA.

Bitzer, J., Schrettl, W. and Schroder, P. (2007) 'Intrinsic motivation in open source software development', *Journal of Comparative Economics*, Vol. 35, No. 1, pp.160–169, Elsevier.

Bonaccorsi, A. and Rossi, C. (2003) 'Why open source software can succeed', *Research Policy*, Vol. 32, No. 7, pp.1243–1258, Elsevier.

Boulos, M., Warren, J., Gong, J. and Yue, P. (2010) 'Web GIS in practice VIII: HTML5 and the canvas element for interactive online mapping', *International Journal of Health Geographics*, Vol. 9, No. 1, p.14, BioMed Central Ltd.

Cậmara, G., Vinhas, L. and de Souza, R. (2009) 'Free and open source GIS: will there ever be a geo-Linux?', in *Proceedings of International Opensource Geospatial Research Symposium (OGRS 2009)*, July, Nantes, France.

Crooks, A. (2006) 'Exploring cities using agent-based models and GIS', in *Proceedings of the Agent 2006 Conference on Social Agents: Results and Prospects*, September, Chicago, USA.

Deshpande, A. and Riehle, D. (2008) 'The total growth of open source', *Open Source Development, Communities and Quality*, pp.197–209, Springer.

Erle, S., Gibson, R. and Walsh, J. (2005) *Mapping Hacks: Tips & Tools for Electronic Cartography*, O'Reilly Media, Inc., CA, USA.

Evans, J. and Mathur, A. (2005) 'The value of online surveys', *Internet Research*, Vol. 15, No. 2, pp.195–219, Emerald Group Publishing Limited.

Fernández-Villamor, I. et al. (2008) 'A comparison model for agile web frameworks', in *Proceedings of the 2008 Euro American Conference on Telematics and Information Systems*, September, pp.1–8, ACM, Aracaju, Brazil.

Garrett, J. (2005) 'AJAX: a new approach to web applications', available at http://www.adaptivepath.com/ideas/essays/archives/000385.php (accessed on 28 June 2010).

Hars, A. and Ou, S. (2002) 'Working for free? Motivations for participating in open-source projects', *Int. J. Electron. Commerce*, Vol. 6, No. 3, pp.25–39, M.E. Sharpe, Inc., Armonk, NY, USA.

Kropla, B. (2005) Beginning MapServer: Open Source GIS Development (Expert's Voice in Open Source), Apress Berkeley, CA, USA.

Lee, S., Kim, H. and Gupta, S. (2009) 'Measuring open source software success', *Omega*, Vol. 37, No. 2, pp.426–438, Elsevier.

Martin, R. (2003) *Agile Software Development: Principles, Patterns, and Practices*, Prentice Hall, PTR Upper Saddle River, NJ, USA.

McArdle, G., Ballatore, A., Tahir, A. and Bertolotto, M. (2010) 'An open-source web architecture for adaptive location based services', in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, May, Vol. 38, No. 2, pp.296–301, Hong Kong.

Newton, A. (2008) MooTools Essentials: The Official MooTools Reference for JavaScript and AJAX Development, Apress Berkeley, CA, USA.

Nichols, D. and Twidale, M. (2003) 'The usability of open source software', First Monday, Vol. 8, No. 1, p.21, available at http:// rstmonday.org/article/view/1018/939 (accessed on 28 June 2010).

O'Reilly, T. (2005) 'What is Web 2.0: business models for the next generation of software', available at http://oreilly.com/web2/archive/what-is-web-20.html (accessed on 28 June 2010).

Obe, R. and Hsu, L. (2009) *PostGIS in Action*, Manning Publications Co., CT, USA.

Orchard, L., Pehlivanian, A., Koon, S. and Jones, H. (2009) *Professional JavaScript Frameworks: Prototype*, YUI, ExtJS, Dojo and MooTools, Wrox Press Ltd., Birmingham, UK.

Raza, A. and Capretz, L. (2010) 'Contributors' preference in open source software usability: an empirical study', *International Journal of Software Engineering & Applications*, Vol. 1, No. 1, pp.45–64.

Regnauld, N. (2006) 'Improving efficiency for developing automatic generalisation solutions', in *Proceedings of ISPRS Workshop: Multiple Representation and Interoperability of Spatial Data*, February, pp.1–5, Hanover, Germany.

Rocher, G., Davis, S. and Brown, J. (2009) The Definitive Guide to Grails: Agile Java Web Development using this Groovy-Based Framework, Apress, NY, USA.

Sanz-Salinas, J. and Montesinos-Lajara, M. (2009) 'Current Panorama of the FOSS4G ecosystem', *The European Journal for the Informatics Professional*, Vol. 10, No. 2, pp.43–51.

Shekhar, S. and Chawla, S. (2003) *Spatial Databases: A Tour*, Prentice Hall, NJ, USA.

Sherman, G. (2008) *Desktop GIS: Mapping the Planet with Open Source Tools*, The Pragmatic Programmers, LLC, Raleigh, NC, and Dallas, TX.

Silva, J., Ribeiro, C. and Lopes, J. (2009) 'Sharing botanical information using geospatial databases', available at http://inforum.org.pt/INForum2009/actas (accessed on 28 June 2010).

Stallman, R. (2009) 'Viewpoint why 'open source' misses the point of free software', *Communications of ACM*, Vol. 52, No. 6, pp.31–33, ACM, New York, NY, USA.

Steiniger, S. and Bocher, E. (2009) 'An overview on current free and open source desktop GIS developments', *International Journal of Geographical Information Science*, Vol. 23, No. 10, pp.1345–1370.

Steiniger, S. and Hay, G. (2009) 'Free and open source geographic information tools for landscape ecology', *Ecological Informatics*, Vol. 4, No. 4, pp.183–195, Elsevier.

Terazono, J., Asada, N., Demura, H., Hirata, N., Saiki, K., Iwasaki, A., Oka, R., Hayashi, T., Suzuki, T., Miyamoto, H. et al. (2009) 'Integrated lunar web-GIS environment using data obtained by lunar exploration, in *Lunar and Planetary Institute Science Conference Abstracts*, March, Vol. 40, p.1232, Texas, USA.

Thomas, D. and Hansson, D. (2009) *Agile Web Development with Rails*, The Pragmatic Programmers, LLC, Raleigh, NC, and Dallas, TX.

Turton, I. (2008) 'Geo tools', *Open Source Approaches in Spatial Data Handling*, pp.153–169, Springer.

Uebersax, J. (2006) 'Likert scales: dispelling the confusion', Statistical Methods for Rater Agreement website, available at http://john-uebersax.com/stat/likert.htm (accessed on 29 May 2010).

Widenius, M., Axmark, D. and DuBois, P. (2002) *MySQL Reference Manual*, O'Reilly & Associates, Inc., Sebastopol, CA, USA.

Wright, K. (2005) 'Researching internet-based populations: advantages and disadvantages of online survey research, online questionnaire authoring software packages, and web survey services', *Journal of Computer-Mediated Communication*, Vol. 10, No. 3, Blackwell Publishing Ltd., available at http://jcmc.indiana.edu/vol10/issue3/wright.html.

Appendix

A Online questionnaire

Title: [Survey on geo-spatial open-source technologies] Questionnaire on {Project name}

1 *Contribution to the project* – Did you actively contribute to the project as a developer?

No answer Yes No

2 *Affiliation with project owners* – Are you a project owner or are you affiliated with the project owners?

No answer Yes No

3 *Level of expertise* – What is your personal experience level with this software?

No answer 0 Beginner 1 2 3 4 5 Expert

4 *Learning curve* – Do you think it is easy or hard to learn how to use this software as a developer?

No answer 0 Hard 1 2 3 4 5 Easy

5 *Stability* – Do you think this software is stable and reliable for production use (e.g., few critical bugs, etc.)?

No answer 0 Unstable 1 2 3 4 5 Stable

6 *Performance* – In your experience, how would you define the performance of the software?

No answer 0 Slow 1 2 3 4 5 Fast

7 *Scalability* – In your experience, how would you define the scalability of the software?

No answer 0 Poor 1 2 3 4 5 Excellent

8 *Interoperability* – How would you define the integration of the software with other technologies?

No answer 0 Hard 1 2 3 4 5 Easy

9 *Extendibility* – Is it easy to extend the software functionalities with external plugins/addons?

No answer 0 Hard 1 2 3 4 5 Easy

10 *Standards* – How would you define the software support for widely-adopted standards?

No answer 0 Poor 1 2 3 4 5 Excellent

11 *Documentation* – What do you think of the documentation of the software (e.g., readability, completeness, quality, useful examples, etc.)?

No answer 0 Poor 1 2 3 4 5 Excellent

12 *Community support* – How would you define the technical support offered on the project forums/mailing lists?

No answer 0 Poor 1 2 3 4 5 Excellent

13 *Frequency of updates* – How would you define the new releases containing new features, improvements and bug fixes?

No answer 0 Rare 1 2 3 4 5 Frequent

14 *Comments* – If you have any comments about this survey and/or {Project name}, you can put them here.