



Faculty Publications

2015-02-10

A review of open source software solutions for developing water resources web applications

Nathan R. Swain

Brigham Young University - Provo

Kilisimasi Latu

University of Melbourne

Scott D. Christensen

Brigham Young University - Provo

Norman L. Jones

Brigham Young University - Provo, njones@byu.edu

E. James Nelson

Brigham Young University - Provo

See next page for additional authors

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>



Part of the [Other Civil and Environmental Engineering Commons](#)

BYU ScholarsArchive Citation

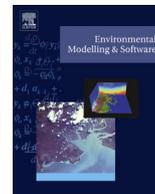
Swain, Nathan R.; Latu, Kilisimasi; Christensen, Scott D.; Jones, Norman L.; Nelson, E. James; Ames, Daniel P.; and Williams, Gustavious P., "A review of open source software solutions for developing water resources web applications" (2015). *Faculty Publications*. 4279.

<https://scholarsarchive.byu.edu/facpub/4279>

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Authors

Nathan R. Swain, Kilisimasi Latu, Scott D. Christensen, Norman L. Jones, E. James Nelson, Daniel P. Ames, and Gustavious P. Williams



Review

A review of open source software solutions for developing water resources web applications



Nathan R. Swain ^{a,*}, Kilisimasi Latu ^b, Scott D. Christensen ^a, Norman L. Jones ^a,
E. James Nelson ^a, Daniel P. Ames ^a, Gustavious P. Williams ^a

^a Civil and Environmental Engineering, Brigham Young University, Provo, UT, USA

^b Infrastructure Engineering, University of Melbourne, Melbourne, VIC, Australia

ARTICLE INFO

Article history:

Received 28 October 2014

Received in revised form

17 January 2015

Accepted 20 January 2015

Available online 10 February 2015

Keywords:

Water resources

Web app

FOSS

FOSS4G

ABSTRACT

Water resources web applications or “web apps” are growing in popularity as a means to overcome many of the challenges associated with hydrologic simulations in decision-making. Water resources web apps fall outside of the capabilities of standard web development software, because of their spatial data components. These spatial data needs can be addressed using a combination of existing free and open source software (FOSS) for geographic information systems (FOSS4G) and FOSS for web development. However, the abundance of FOSS projects that are available can be overwhelming to new developers. In an effort to understand the web of FOSS features and capabilities, we reviewed many of the state-of-the-art FOSS software projects in the context of those that have been used to develop water resources web apps published in the peer-reviewed literature in the last decade (2004–2014).

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Hydrologic model simulations are often used to estimate and analyze watershed responses to specific scenarios. While most stakeholders are capable of interpreting the results of hydrologic models, they may not have the technical expertise required to properly configure a simulation scenario. This is particularly true for hydrologic models that use spatially distributed data as input. Water resources web applications or web apps are being used to overcome many of the challenges of using hydrologic simulations in decision-making, as evidenced by the increasing frequency of articles describing this type of application (e.g.: Demir and Krajewski, 2013; Goodrich et al., 2008; Kulkarni et al., 2014; Sun, 2013). We define a water resources web app as a narrowly focused web-accessed application for performing common tasks related to hydrology and water resources modeling.

Spatial data is an important component of water resources web apps making software that facilitates spatial data use on the web a major theme of this paper. These spatial needs can be addressed with geographic information systems (GIS) software. Whereas,

proprietary software vendors conveniently offer all of the needed GIS functionality in a single software package, free and open source software (FOSS) projects tend to focus on a single category of functionality (Steiniger and Weibel, 2010). Thus, creating a water resources web app using FOSS requires the developer to synthesize several FOSS projects. Additionally, the abundance of FOSS GIS (FOSS4G) that is available can be overwhelming to new developers (for example, see the list of registered products at <http://www.opengeospatial.org/resource/products>). In an effort to understand the complex web of FOSS features and capabilities, we reviewed many of the state-of-the-art FOSS software projects in the context of those that have been used to develop water resources web apps that have been published in the literature in the last decade (2004–2014). The scope of the review, while focusing on water resources web apps, also includes web apps from other disciplines in the earth sciences that share the same specialized requirements for web development.

We divided the FOSS projects reviewed into two categories: GIS components and tools and web development software. We provide a brief description of each software project along with a table of relevant web apps from the literature that used the software. This paper aims to be an introductory guide that will save developers of water resource web apps time and effort in the genesis of their projects.

* Corresponding author: Tel.: +1 801 422 2811; fax: +1 801 422 0159.

E-mail address: nathan.swain@byu.net (N.R. Swain).

2. Web GIS software review

The vector and raster spatial data associated with hydrologic analyses require special consideration for water resources web apps. Fortunately, there is an abundance of FOSS4G available that can be used by water resources web app developers to acquire, modify, store, visualize, and analyze spatial data. However, the abundance of FOSS4G can also be overwhelming and confusing to novice developers (Steiniger and Hunter, 2012a,b). While many of the GIS software projects have been reviewed in the literature (Chen et al., 2010; DeVantier and Feldman, 1993; Li et al., 2007; Schut, 2007; Steiniger and Hunter, 2012a,b; Zhao et al., 2012), our goal is to focus on the FOSS4G projects that have been selected and implemented by web app developers in the water resources and related earth sciences fields.

Many of the web apps reviewed use software projects that implement Open Geospatial Consortium (OGC) standards (OGC, 2012a). The OGC publishes specifications for data delivery over the internet such as web mapping service (OGC-WMS; OGC, 2006), web feature service (OGC-WFS; OGC, 2014) and web coverage service (OGC-WCS; OGC, 2012c) and data format standards like simple features interface standard (OGC-SFS; OGC, 2010b), geography markup language (OGC-GML; OGC, 2012b), and keyhole markup language (OGC-KML; OGC, 2008). In addition, OGC specifies standards for data search such as catalog service for the web (OGC-CSW; OGC, 2007a) and geoprocessing such as web processing service (OGC-WPS; OGC, 2007b).

The flexibility offered to developers by the interoperability of OGC compatible projects is illustrated by the fact that the system architecture of several of the reviewed projects is described generically in terms of OGC standards rather than naming specific implementations of each standard (Han et al., 2012; Li et al., 2013; Sun et al., 2012). For example, Han et al. (2012) describe OGC-WMS, OGC-WFS, and OGC-WCS as required components of the architecture of their system rather than specifying a specific implementation of the standards such as MapServer or GeoServer. Other projects implement custom versions of the OGC standards (Blower et al., 2013, 2009; Feng et al., 2011; Frehner and Brändli, 2006; Oulidi et al., 2012).

There are many types of GIS software that are tailored to specific GIS tasks (Steiniger and Hunter, 2012a,b; Zhao et al., 2012). For simplicity, we organized the GIS review into four broad categories: spatially enabled databases for storage, spatial data publishing for sharing spatial data, mapping libraries for visualizing spatial datasets, and spatial analysis for geoprocessing and spatial algorithms.

2.1. Spatial databases

Spatial databases store geographical data in a file system that is suitable for large datasets with thousands of features and provide an efficient mechanism to store, query, analyze, and update these data (Steiniger and Hunter, 2012a,b). Many spatial databases are extensions of existing structured query language (SQL) databases and implement the OGC-SFS standard (SQL option), which defines how spatial objects should be represented. Of the web apps reviewed, three spatially enabled SQL databases were used including: MySQL Spatial, PostGIS, and SpatiaLite. Table 1 provides a summary of the spatial databases used by the web apps from the literature along with a brief description of each spatial database.

2.1.1. PostGIS

PostGIS is a spatial database extension for the PostgreSQL database (Holl and Plum, 2009; Nguyen, 2009). Steiniger and Hunter, 2012a,b claim that PostGIS provides the most extensive

implementation of the OGC-SFS standard. In addition, PostGIS boasts impressive support for raster data and analysis, for which it incorporates the GDAL library (Warmerdam, 2008) to support a wide array of raster formats. The extension provides three new column types including *geometry*, *geography*, and *raster* and it supports spatial indexing schemes for fast retrieval (Nguyen, 2009). PostGIS also includes a large library of spatial database functions (~400 in version 2.1 not including variants) for basic analysis of vector and raster objects (e.g. clip, buffer, intersection, and union), conversion between the vectors and rasters, and spatial reference system transformations.

2.1.2. SpatiaLite

SpatiaLite is the spatial extension for the SQLite database (Steiniger and Hunter, 2012b). The project aims to be roughly equivalent to PostGIS, but far lighter weight in the SQLite fashion. It uses the geometry library of GEOS (Foundation, 2014) to implement OGC-SFS (Zhao et al., 2012). Like PostGIS, SpatiaLite boasts a large library of database functions for performing spatial analysis (~400 in version 4.2 not counting variants). However the functions assume planar geometry and effectively ignore the spatial reference system of the data. SQLite performs well in single user environments, but it is not well equipped to handle multiple concurrent connections as occurs often in a web environment (Furieri, 2008).

2.1.3. MySQL spatial

Stefan Steiniger and Andrew J. S. Hunter (2012) state that MySQL Spatial provides a basic implementation of OGC-SFS. MySQL spatial supports vector data formats, but does not support rasters at this time. MySQL spatial also provides a database function library, though not as extensive as PostGIS or SpatiaLite (~90 functions in version 5.7). All calculations assume Euclidean (planar) geometry. The spatial types and functions can be used with several MySQL storage mechanisms including MyISAM, InnoDB, and ARCHIVE and spatial indexing is supported in MyISAM and InnoDB tables.

Table 1

Summary of spatial databases used by web apps in the literature.

FOSS4G	Web app
SpatiaLite	BASHYT (Cau et al., 2013)
MySQL Spatial	Automated Geospatial Watershed Assessment (AGWA; Goodrich et al., 2008, 2011)
	Flood Assessment Modeling Tool (Kulkarni et al., 2014)
PostGIS	Object-Oriented and OpenGIS Hydro Information System (30-HIS; Leone et al., 2006)
	Open Source Web Fire Mapper (Davies et al., 2009)
	Spatial Forest Information System (Li et al., 2007)
	WebGIS for Geospatial Vector Data Sharing (Fang and Feng, 2009)
	integrated Geospatial Urban Energy Information & Support System (iGUESS; de Sousa et al., 2012)
	Cloud Framework for Hydro Information System (Blagoj Delipetrev et al., 2012)
	Hydrogeological Information System (HydriS; Oulidi et al., 2012)
	Natural Resources Information System (Singh et al., 2012)
	Water Management Decision Support System (EDSS; Sun, 2013)
	National Operational Assessment of Hazards (NOAH; Alconis et al., 2013)
	Web-based Hydrologic Transport Model (Brooking and Hunter, 2013)
	Web Application for Water Resources (Blagoj Delipetrev et al., 2014)
	Iowa Flood Information System (IFIS; Demir and Krajewski, 2013)
	eHabitat 2.0 (Dubois et al., 2013)
	Emissions Inventory (Gkatzoflias et al., 2013)
	Web Application for Water Resources (Blagoj Delipetrev et al., 2014)

2.2. Spatial data publishing

Spatial data can be published using a class of software called a geospatial data server. The role of a geospatial data server is to make spatial data available in web-friendly formats. This is done by offering the data or visualizations of the data as OGC standardized web services, which can then be rendered on a web page in a browser using a mapping library or plugin (discussed in the next section). The primary OGC standards that are applicable to geospatial data servers are the web mapping service (OGC-WMS), web feature service (OGC-WFS), and web coverage service (OGC-WCS). OGC-WMS is concerned with serving raster and vector data as maps (images), whereas OGC-WFS allows direct access to the data including reading, writing, and updating. OGC-WCS is used to serve raster or image layers. Three FOSS4G spatial data publishing software projects were used in the web apps reviewed: MapServer, GeoServer, and deegree. Table 2 provides a summary of the geospatial data servers that are used by web apps reported in the literature.

2.2.1. MapServer

MapServer is a Common Gateway Interface (CGI) application written in the C programming language that can be installed on any operating system (Gkatzoflias et al., 2013; Vatsavai et al., 2006). The C implementation also gives MapServer exceptional performance compared to the Java implementations of the other projects (OSGeo, 2014). It is capable of serving spatial datasets as OGC web services including OGC-WMS, OGC-WFS, and OGC-WCS. MapServer supports numerous raster and vector data formats via the GDAL libraries including TIFF, GeoTIFF, ESRI shapefiles, and PostGIS.

MapServer is configured via special files called Mapfiles. It also includes an Application Programming Interface (API) called MapScript that can be used to configure the server and interact with the server's data programmatically. MapScript is available for several programming languages including Python, Java, and PHP. The datasets that MapServer serves can be stored on the file system of the server or in spatially enabled databases (such as PostGIS).

2.2.2. GeoServer

GeoServer is a Java-based web server that implements the OGC-WFS, OGC-WCS, OGC-WMS, and OGC-WPS web service standards

Table 2
Summary of geospatial data servers used by web apps in the literature.

FOSS4G	Web app
MapServer	Web-based Hydrologic Geographic Information System (WHYGIS; Choi et al., 2005a,b) Open Source Web Fire Mapper (Davies et al., 2009) integrated Geospatial Urban Energy Information & Support System (iGUESS; de Sousa et al., 2012) Natural Resources Information System (Singh et al., 2012) BASHYT (Cau et al., 2013) Emissions Inventory (Gkatzoflias et al., 2013) Spatial Forest Information System (Li et al., 2007)
deegree	Spatial Forest Information System (Li et al., 2007) Hydrogeological Information System (HydriS; Oulidi et al., 2012)
GeoServer	WebGIS for Geospatial Vector Data Sharing (Fang and Feng, 2009) Geospatial Model Sharing Platform (GeoMSP; Feng et al., 2011) USGS Geo Data Portal (Blodgett et al., 2012) Cloud Framework for Hydro Information (Blagoj Delipetrev et al., 2012) Web-based Hydrologic Transport Model (Brooking and Hunter, 2013) Web-based groundwater database management system (Iwanaga et al., 2013) Web Application for Water Resources (Blagoj Delipetrev et al., 2014) Environmental Data System (EDS; Melis et al., 2014)

(Iacovella and Youngblood, 2013). As a Java application, GeoServer can be used with any of the major operating systems. It is packaged as a web archive (WAR) for use with existing servlet container applications such as Apache Tomcat and Jetty (GeoServer, 2013).

GeoServer provides a graphical web administration tool for configuration. Alternatively, GeoServer can be configured programmatically through a Representational State Transfer (REST) interface. Other features of GeoServer include integrated OpenLayers and Google Earth™ support, GeoWebCache automated spatial caching, tile mapping, and wide support for spatial databases such as PostGIS, ArcSDE, Oracle, and DB2. GeoServer relies heavily on GeoTools (GeoTools, 2014), an open source Java library that provides GIS support for spatial data types such as vector and raster layers (Ballatore et al., 2011).

2.2.3. deegree

The deegree project is a Java implementation of OGC web services and can be run on all operating systems. It provides implementations of the OGC-WFS, OGC-WMS, OGC-CSW, and OGC Web Map Tile Service (OGC-WMTS; OGC, 2010a) web services. It also provides support for web processing (OGC-WPS). Like GeoServer, deegree provides a web administration tool for configuration and it offers a REST-like interface for programmatically configuring the server. It supports various data sources such as PostGIS, shapefiles, and OGC-GML (Müller, 2007).

2.3. Mapping libraries

Mapping libraries are needed to visualize spatial data in a web environment. Mapping libraries or plugins consume data from OGC-WMS, OGC-WFS, and OGC-WCS web services and render the maps for presentation in a client (i.e.: web browser). The mapping libraries used in the web apps reviewed for this paper are all JavaScript libraries that run in web browsers. We found three mapping libraries in our review: OpenLayers, Google Maps™, and Google Earth™. Table 3 shows a summary of the mapping libraries used by web apps from the literature review.

2.3.1. OpenLayers

OpenLayers is a web-mapping client library for rendering interactive maps on a web page (Hazzard, 2011). It is a pure JavaScript library for building rich web-based geospatial applications similar to Google Maps™. OpenLayers is capable of rendering vector and raster data from a variety of formats including GeoJSON, OGC-KML, OGC-GML, and OGC web services. It leverages WebGL and Canvas 2D for better performance. OpenLayers also provides methods for drawing on the map and editing data interactively. It allows developers to use a variety of services for base maps including Open Street Map, Bing, MapQuest, and Google. OpenLayers does not currently support a 3D globe-type environment. It does not require a plugin and does not have the use restrictions that are imposed by the Google license (Steiniger and Hunter, 2012a,b), although using some of the proprietary base maps (e.g.: Google and Bing) used in OpenLayers may invoke licensing restrictions.

2.3.2. Google Earth™

While Google Earth™ is not FOSS it is considered in this review because of its popularity and with some restrictions it is cost-free for most users. It is not free for commercial use and private users are limited to 25,000 map requests per day (Steiniger and Hunter, 2012a,b). One notable feature of Google Earth™ is the ability to easily animate and display data in a 3D globe environment using the OGC-KML format. The disadvantage to using Google Earth™ as a map renderer is that it requires the user to install a browser plugin, which is not supported on all operating systems or in 64-bit

Table 3
Summary of the mapping libraries used by web apps in the literature.

FOSS4G	Web app
OpenLayers	WebGIS for Geospatial Vector Data Sharing (Fang and Feng, 2009) Geospatial Model Sharing Platform (GeoMSP; Feng et al., 2011) USGS Geo Data Portal (Blodgett et al., 2012) integrated Geospatial Urban Energy Information & Support System (iGUESS; de Sousa et al., 2012) Cloud Framework for Hydro Information (Blagoj Delipetrev et al., 2012) DEM Explorer (Han et al., 2012) Water Management Decision Support System (EDSS; Sun, 2013) Web-based Hydrologic Transport Model (Brooking and Hunter, 2013) Cloud Framework for Hydro Information (Blagoj Delipetrev et al., 2012) Web-based groundwater database management system (Iwanaga et al., 2013) Flood Assessment Modeling Tool (Kulkarni et al., 2014)
Google Earth and/or Maps	Web GIS Based Hydrograph Analysis Tool (WHAT; Lim et al., 2005) Geoportal for Hydrological Applications (Díaz et al., 2008) Automated Geospatial Watershed Assessment (AGWA; Goodrich et al., 2008, 2011) GODIVA2 (Blower et al., 2009) Forest Fires Online/Offline Mapping and Monitoring Application (FOMA; Carvalheiro et al., 2010) Available WAtER Resource (AWARE; Granell et al., 2010) Virtual Sensor System (Hill et al., 2011) Novel Google Earth Visualizing (Sun et al., 2012) National Operational Assessment of Hazards (NOAH; Alconis et al., 2013) Fire Logic Animation (FLOGA; Bogdos and Manolagos, 2013) Iowa Flood Information System (IFIS; Demir and Krajewski, 2013) Emissions Inventory (Gkatzoflias et al., 2013) Environmental Data System (EDS; Melis et al., 2014) CyberFlood (Wan et al., 2014)
Combination	ncWMS (Blower et al., 2013) Web Application for Water Resources (Blagoj Delipetrev et al., 2014)

web browsers. Additionally, the Google Earth API for the web plugin has been deprecated as of December 12, 2014 and will lose support completely on December 14, 2015 (Google, 2014).

2.3.3. Google Maps™

Google Maps™ provides a 2D mapping environment with high-resolution base map imagery. The new version of Google Maps™ (version 3) provides a library that allows users to draw shapes on the map and edit spatial data interactively. Like Google Earth™, Google Maps™ is capable of displaying spatial data in OGC-KML format. Alternatively, data can be added dynamically using the JavaScript API. Unlike Google Earth™, Google Maps™ does not require a browser plugin.

2.4. Spatial analysis

In this section we present the FOSS software projects that were used to support spatial analysis in the web apps reported in the literature. Spatial analysis in water resources web apps can be achieved by using software projects that implement the OGC-WPS standard. An OGC-WPS can be installed on a stand-alone server that is optimized for geoprocessing, which tend to be designed to handle multiple simultaneous requests and a heavy processing load. Note that the deegree and GeoServer projects that were discussed in the spatial data publishing also include OGC-WPS functionality. Although it is unclear whether the web apps that used GeoServer and deegree used the OGC-WPS features, both projects

will be described in this section. Table 4 presents a summary of the web apps that used geoprocessing web service software.

2.4.1. 52° North WPS

The 52° North WPS project represents a full implementation of the OGC-WPS standard (52°North, 2014; Schut, 2007). 52°North WPS provides an extensible, pluggable framework for publishing geoprocessing algorithms as web services. It can be linked with existing geoprocessing libraries such as GRASS (GRASS Development Team, 2014), Sextante (Olaya and Gimenez, 2011), and ArcGIS® Server for out-of-the-box geoprocessing capabilities (Steiniger and Hunter, 2012a,b). 52°North WPS also allows developers to publish custom Python scripts (Sanner, 1999), R scripts (Chambers, 2013), and Java processes as web services. A number of geospatial data types are supported as input such as GeoTiff, Arc-Grid, Shapefiles, OGC-GML, and OGC-KML, and OGC data services (OGC-WMS, OGC-WFS, and OGC-WCS). All results can be stored as simple web accessible resources or as OGC web services.

2.4.2. PyWPS

PyWPS is an implementation of OGC-WPS written in Python. Like any OGC-WPS, PyWPS does not process data itself, rather, it provides the link between the web and the local tools on the server such as GRASS, GDAL, and R scripts. Castronova et al. (2013) implemented an instance of PyWPS to demonstrate how the OGC-WPS standard can be extended to offer scientific modeling as a web service.

2.4.3. GeoServer WPS

GeoServer provides a full implementation of OGC-WPS in addition to the spatial data publishing services. The processes can be called with GeoServer resources as inputs and they can output to new GeoServer resources. GeoServer provides the JTS Topology Suite (Vivid Solutions, 2014) for default geoprocessing capabilities and it allows for custom processes written in Java.

2.4.4. deegree WPS

The deegree project is an implementation of OGC-WPS. No default processes are provided with deegree WPS as of version 3.2.0. However, it does provide a mechanism for publishing custom Java processes. In older documentation, deegree promised connections to GRASS, Sextante, and a proprietary processing library called FME that are currently not present (Safe Software, 2014).

3. Web development software

Water resources web apps require a strategy for developing the web interface and synthesizing all of the software components. As a minimum, they require a web server and HTML for building the web pages of the web app. However, a scripting language on the server is often required to handle interaction with database, other software, and other logic of the web app. We review and summarize the web development software used by the water resources

Table 4
Summary of geoprocessing software used by web apps in the literature.

FOSS4G	Web App
52°North WPS	Geoportal for Hydrological Applications (Díaz et al., 2008) Available WAtER Resource (AWARE; Granell et al., 2010) USGS Geo Data Portal (Blodgett et al., 2012)
PyWPS	integrated Geospatial Urban Energy Information & Support System (iGUESS; de Sousa et al., 2012) eHabitat 2.0 (Dubois et al., 2013) Modeling Web Services via OGC-WPS (Castronova et al., 2013)

Table 5
Summary of web development software used by web apps in the literature.

Category	FOSS	Web app
Programming Languages	PHP	SICI hydrological and geomorphological Catastrophe information system (Guzzetti and Tonelli, 2004)
		Cloud Framework for Hydro Information System (Blagoj Delipetrev et al., 2012)
		Natural Resources Information System (Singh et al., 2012)
		Novel Google Earth Visualizing (Sun et al., 2012)
		Fire Logic Animation (FLogA; Bogdos and Manolakos, 2013)
		Iowa Flood Information System (IFIS; Demir and Krajewski, 2013)
		Emissions Inventory (Gkatzofias et al., 2013)
		Web Application for Water Resources (Blagoj Delipetrev et al., 2014)
		Virtual Database for Distributed Ecological Data (Frehner and Brändli, 2006)
		Object-Oriented and OpenGIS Hydro Information System (3O-HIS; Leone et al., 2006)
	Java	USDA Conservation Reserve Program DSS (Rao et al., 2007)
		Geoportal for Hydrological Applications (Díaz et al., 2008)
		Open Source Web Fire Mapper (Davies et al., 2009)
		WebGIS for Geospatial Vector Data Sharing (Y. Fang and Feng, 2009)
		Web-based Participatory Wind Energy Planning (WePWEP; Simao et al., 2009)
		Available WATER Resource (AWARE; Granell et al., 2010)
		SPARROW DSS (Booth et al., 2011)
		Geospatial Model Sharing Platform (GeoMSP; Feng et al., 2011)
		DEM Explorer (Han et al., 2012)
		Hydrogeological Information System (HydriS; Oulidi et al., 2012)
Web Frameworks	PERL	Web-based Hydrologic Geographic Information System (WHYGIS; Choi et al., 2005a,b)
	Ruby on Rails	Integrated Geospatial Urban Energy Information & Support System (iGUESS; de Sousa et al., 2012)
	CodeIgniter (PHP)	Automated Geospatial Watershed Assessment (AGWA; Goodrich et al., 2008, 2011)
	Django (Python)	Water Management Decision Support System (EDSS; A. Sun, 2013)
		ubertool (Flaishans et al., 2014)
		Web-based Interactive River Model (WIRM; Walker and Chapra, 2014)
		Web-based Interactive River Model (WIRM; Walker and Chapra, 2014)
		WATER quality InformaTion System (WAITS; Peres et al., 2013)
		Web-based groundwater database management system (Iwanaga et al., 2013)
		Web-based water infrastructure database (WATERiD; Jung et al., 2013)
Content Management Systems	Backbone.js (Client Side)	Earth Science Environmental Simulator (ESES; Van Knowe et al., 2014)
	CKAN	
	Drupal	

and earth science web apps reported in the literature. We organized the web development software covered in this section into hierarchical layers of programming language, web frameworks, and content management systems. A summary of web software used to implement the web apps reported in the literature is shown in Table 5.

3.1. Programming languages

Programming languages can be used in web development to make websites more dynamic and to handle advanced logic beyond simply returning static HTML. For example, programming languages could be used to execute simulation runs or prepare a complex visualization when a user submits a request. Practically any programming language could be used to fulfill this purpose, provided the language provides mechanisms for working with HTTP requests. The following section provides a summary of the programming languages that were used in web app development.

3.1.1. PHP

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a very popular scripting language that is especially suited for web development. PHP is embedded in the HTML pages of the website and the code is executed on the server when the web page is requested. It includes support for a wide range of databases via ODBC and provides a database abstraction layer called PDO. It also has many extensions that add common web functionality such as managing sessions and cookies, user authentication, and file uploads (Royappa, 2000).

3.1.2. Java

A significant number of projects used Java-based solutions for web development. Several developers used Java Platform, Enterprise Edition (Java EE, formerly J2EE) and JavaServer Pages (JSP) to develop their web apps, while others did not specify the Java framework used. These Java technologies are free and open source and provide a powerful, cross-platform development environment for creating and running large-scale, multi-tiered, scalable, reliable, and secure network applications (Oracle, 2012).

3.2. Web frameworks

Web frameworks provide a scriptable approach for building websites with the intent of alleviating much of the low-level coding typically associated with static website design. A web framework typically offers features for accessing databases, building pages from dynamic templates, managing users and sessions, and creating a secure website. Most web frameworks follow some form of the Model View Controller (MVC) development paradigm, where the model consists of the data of the website (often a database model), the view is the presentation of the data, and the controller provides the logic that interprets the data for the view and handles user input.

3.2.1. Python web frameworks

The scientific modules such as SciPy and NumPy have made Python a popular scripting language for scientific computing (Millman and Aivazis, 2011; Oliphant, 2007). As such, Python-powered web frameworks are a popular choice for building scientific web apps. There are over fifty Python web frameworks, but the

most popular are Django, Grok, Pylons, TurboGears, web2py, and Zope2 ([The Python Wiki, 2014](#)). Python web frameworks vary greatly in default functionality. For example, Django and TurboGears provide a significant amount of functionality that is enabled by default to make development quicker and easier, while Pylons provides only minimal default functionality to allow greater flexibility for the developer.

3.2.2. CodeIgniter

CodeIgniter is an MVC PHP web framework with exceptional performance and virtually no configuration. Some of the features of CodeIgniter include full-featured database classes, form and data validation, security and cross-site filtering, session management, and file uploading. It is an attractive option for web developers who already use PHP, but want to use a structured framework approach ([Upton, 2007](#)).

3.2.3. Ruby on Rails

Ruby on Rails, or simply Rails, is a web framework written in the Ruby language. Rails web applications are organized using the MVC pattern like the other web frameworks. Rails features convention over configuration, meaning that it will do a lot of the heavy lifting of web development automatically. Many other web frameworks are influenced by Rails development. There are tens of thousands of sites developed using Rails including Twitter and GitHub ([Tate and Hibbs, 2006](#)).

3.2.4. Client side frameworks

To avoid the delay caused by frequent interaction between the browser/client and the server, some web apps are developed to run completely in the client (web browser) as pure JavaScript applications. The initial request downloads the source code for the application from the server and initiates the web app. [Walker and Chapra \(2014\)](#) developed a client-side web app that runs the Web-based Interactive River Model (WIRM). The web app was developed using Backbone.js ([Sugrue, 2013](#)), a JavaScript MVC client-side framework. They used the Python web framework Django on the server to handle user authentication and database interaction.

3.3. Content management systems

Content Management Systems (CMS) are often built on a web framework and provide a higher level of abstraction to web development. CMS web sites use a GUI in the browser with a limited amount of coding. The focus of a CMS is to allow the developers to manage content independently according to the web template that is chosen. The user interface of a CMS has a front-end and back-end structure where the front-end is accessible to users and only the administrators can access the back-end for maintaining and development purposes ([Rojas-Sola et al., 2011](#)). This type of system is ideal for non-technical administrators of the website.

3.3.1. Drupal

Drupal is a widely used FOSS CMS. It provides a browser-based graphical user interface to develop a website minimizing the need to write code. It comes with only basic functionality enabled. Developers add functionality to a website by installing modules from an extensive library. Custom modules can be created using PHP and a series of hooks into other routines provided by the Drupal developers. Once the site development is complete, end users can easily maintain and update the site using the same interface ([Drupal, 2013](#)).

3.3.2. CKAN

CKAN is a specialized CMS for hosting datasets using a built-in data management system ([CKAN, 2013](#)). CKAN is built on the Pylons Python web framework. The data management system comes ready to host data out of the box and the data can be stored with a rich set of metadata. CKAN also provides a set of Python programming interfaces for building custom extensions and a REST API for uploading and downloading data programmatically ([CKAN, 2013](#)).

4. Discussion

We selected the free and open source software projects included in this review based on a literature review of 45 water resources and earth science web apps that were developed in the last decade (2004–2014) and reported in the literature. The FOSS projects presented in this review do not represent a comprehensive or even representative sampling of all FOSS web GIS and web development software available. Rather, we narrowed the long list of available FOSS software projects to only those projects that have been used by existing water resources and earth science web apps that have been published in recent peer-reviewed literature. This review extends previous FOSS4G and FOSS web software reviews by focusing on only those FOSS projects that have been tried and proven in existing water resources and earth science web apps.

The quality and capabilities of the web apps included in the review vary significantly. Some of the web apps were developed as prototype or demonstration systems (e.g.: [Bogdos and Manolagos, 2013](#); [Feng et al., 2011](#); [Oulidi et al., 2012](#)), while others were developed as full-featured data and modeling services that were currently in operation at the time of writing (e.g.: [Alconis et al., 2013](#); [Blodgett et al., 2012](#); [Demir and Krajewski, 2013](#)). The web apps address data and modeling needs in a wide range of applications including water resources, wild fires, water quality, urban planning, flood warning, ecology, and geology. Of the web apps included in the review, approximately 80% were published in the last 5 years (2009–2014) and almost 45% were published in the last 2 years indicating a growing interest in web apps as a medium for earth science modeling and data.

Each water resources or earth science web app included in the review included at least one FOSS software component, with a majority of the web app projects using several FOSS projects to address various spatial data needs. We included web apps that had proprietary components in addition to at least one FOSS component. In several cases, we included web apps in which the GIS capabilities were provided entirely by proprietary software or were not specified, but the web software was a FOSS solution ([S. Fang et al., 2013](#); [Flaishans et al., 2014](#); [Frehner and Brändli, 2006](#); [Rao et al., 2007](#); [Simao et al., 2009](#); [Van Knowe et al., 2014](#); [Walker and Chapra, 2014](#)). We will focus the remaining discussion on each category of FOSS reviewed.

4.1. Spatial database comparison

No other category of FOSS software exhibited as strong a preference for one project as the spatial database category. Of the web apps reviewed, 19 reported using an SQL database with a spatial extension. The PostgreSQL database with the PostGIS spatial extension was overwhelmingly the preferred solution with 15 web apps using PostGIS. MySQL with Spatial extension was used by 3 web apps and 1 web app used SQLite with the Spatialite extension.

It was not unexpected that SQLite with Spatialite was not as popular in the web apps reviewed, because SQLite is suboptimal for web environments as discussed in Section 2.1.2. It is surprising that MySQL Spatial was not selected as often, because MySQL is the

most popular FOSS SQL database for general web development. However, the spatial implementation of PostGIS is superior on several fronts. One difference is that PostGIS has extensive support for raster data, while MySQL Spatial has no raster support. Another key difference is that PostGIS boasts a library of about 400 database functions (not counting variants) to perform spatial analysis on both raster and geometry columns, whereas the spatial function library of MySQL Spatial is minimal with only about 90 functions. The other primary difference is that MySQL spatial functions ignore the spatial reference system and use only Euclidean (planar) distances while PostGIS has support for spatial reference systems. Table 6 provides a summary of the notable features of the spatial databases reviewed.

4.2. Spatial data publishing comparison

The line was not so clearly drawn in the other categories. Of the web apps reviewed, 16 reported using software for spatial data publishing. MapServer was used in 7 web apps, GeoServer was used in 8 web apps, and deegree was used in 2 web apps (one web app used both MapServer and deegree).

These three software projects are comparable in terms of their implementations of applicable OGC standards. GeoServer and deegree provide web interfaces for configuring the data on the server making them more user-friendly than the file-based configuration of MapServer. However, MapServer can be configured programmatically via MapScript in a number of different development environments including, PHP, Python, Perl, Ruby, Java, and .NET. GeoServer and deegree can be configured programmatically via REST APIs. In terms of performance, MapServer tends to outperform GeoServer and deegree by virtue of its C implementation. GeoServer and deegree also provide a WPS implantation that can be used for geoprocessing capabilities. Table 7 provides a summary of the notable features of the spatial data publishing software included in the review.

4.3. Mapping library comparison

There were 24 web apps that reported using mapping libraries with 12 web apps that used Google Maps™ or Google Earth™, 10 web apps that used OpenLayers, and 2 web apps that used a combination of OpenLayers and the Google libraries. The Google mapping libraries are frequently used because many people are familiar with the popular Google mapping service (<https://www.google.com/maps>). The primary advantage of Google Earth™ over Google Maps™ or OpenLayers, is that it provides a 3D-globe mapping environment. In terms of supported data formats, Google Maps™ and Google Earth™ are limited—only accepting data in OGC-KML format or via the JavaScript APIs. OpenLayers boasts support for a wide range of formats including OGC-KML, OGC-GML, GeoJSON, OGC mapping services such as OGC-WMS and OGC-WFS, and many others. Finally, OpenLayers and Google Maps™ allow for interactive user input via drawing on the map. Table 8 shows a comparison of the notable features of the mapping libraries presented in the review.

4.4. Spatial analysis comparison

The spatial analysis software was the least used category of software with only 6 web apps that specified using spatial analysis software. The 52° North WPS project was used by 3 of the web apps and PyWPS was used by 3 web apps. It should be noted that many web apps cited using GeoServer and deegree for spatial publishing, but it was unclear whether any of those projects made use of the GeoServer and deegree OGC-WPS functionality. All four software projects implement the OGC-WPS specification.

The primary difference between the implementations is in the default processes that are supported. The 52° North WPS offers the most processes “out-of-the-box” with the ability to link to the GRASS, Sextante and ArcGIS Server geoprocessing libraries as well as custom processes written in Python, R, and Java. PyWPS only supports processes written in Python and R, but it can be linked to GRASS via the Python GRASS bridge. GeoServer WPS offers the JTS Topology Suite processes in the default configuration and allows developers to write custom processes written in Java. The current version of deegree WPS only supports custom processes written in Java, though connections to GRASS, Sextante, and FME are in the works. As spatial data publishing projects, GeoServer and deegree have the advantage of being able to operate on data that is stored locally and store the results as OGC web services, resulting in fewer file transfers. The 52° North WPS is capable of storing results as OGC web services, though it is unclear if it is able to do so without the aid of GeoServer or deegree. A summary of the notable features of the spatial analysis software reviewed is shown in Table 9.

4.5. Web development software comparison

The web software category was the most widely varying category with at least 9 different FOSS software projects or languages used by water resources and earth science web app developers. Although all of the applications were web apps and necessitated some web development strategy, only 34 specified what web software was employed. The most popular web development strategy used to create the web apps reviewed were Java solutions—numbering 17 in all. The next most popular approach was PHP web development with 8 web apps that used this method. Of the remaining web apps, 1 web app used Perl as a scripting language on the server, 2 web apps used the Django Python web framework, 1 web app used the Ruby on Rails web framework, 1 web app used the CodeIgniter PHP web framework, 2 web apps used the Drupal content management system, 1 web app used the CKAN data management system, and 1 web app used a client-side framework called Backbone.js in conjunction with Django. It is important to note that the web development software presented in this review is not a comprehensive sampling of all the web development software available.

Most web frameworks provide strategies for solving common web development challenges such as user management, database interaction, creating dynamic HTML, and handling file uploads. The primary difference between web software lies in the approach that the web framework takes to solving web development tasks. For example, Drupal and CKAN provide user management systems that

Table 6
Comparison of the notable features of spatial databases.

Spatial database	Number web apps	Spatial functions	OGC-SFS	Vector format	Raster format	Spatial reference calculations	Concurrent access
PostGIS	15	~400	✓	✓	✓	✓	✓
MySQL Spatial	3	~090	✓	✓			✓
SpatialLite	1	~400	✓	✓			

Table 7

Comparison of the notable features of spatial data publishing software.

Spatial data publishing	Number Web apps	Implementation Language	OGC-WFS	OGC-WMS	OGC-WCS	OGC-WPS	Web configuration	File configuration	REST API	Scripting API
MapServer	7	C	✓	✓	✓			✓		✓
GeoServer	8	Java	✓	✓	✓	✓	✓		✓	
deegree	2	Java	✓	✓		✓	✓		✓	

Table 8

Comparison of the notable features of mapping libraries.

Mapping library	Number web apps	Input formats					3D Globe	2D Map	Plugin required	Draw on map
		OGC-KML	OGC-GML	OGC-WFS	OGC-WMS	GeoJSON				
Google Earth™	14	✓					✓		✓	
Google Maps™	–	✓						✓		✓
OpenLayers	12	✓	✓	✓	✓	✓		✓		✓

Table 9

Comparison of the notable features of spatial analysis software.

Spatial analysis	Number web apps	Implementation language	OGC-WPS	GRASS GIS	Sextante	ArcGIS Server	GDAL	PROJ	JTS topology	Python scripts	R scripts	Java processes	Local data
52° North WPS	3	Java	✓	✓	✓	✓				✓	✓	✓	
PyWPS	3	Python	✓	✓			✓	✓		✓	✓		
GeoServer	–	Java	✓						✓			✓	✓
deegree	–	Java	✓									✓	✓

require virtually no configuration—complete with login, logout, forgotten password, and user profile pages. Django also provides a user management system, but the burden of creating the login, logout, and user profile web pages and logic rests on the developer. Neither approach is better than the other.

Consequently, selecting a web framework depends largely on the needs and complexity of the project and the preferences of the developer. Some of the factors to consider when selecting a web framework include the programming experience of the developer, the supporting libraries available (e.g.: geoprocessing libraries), the size or scale of the project, and the functionality required by the project.

In terms of the current review, Java frameworks were likely the most popular for web apps for a few reasons. Many of the popular software implementations of OGC standards are Java implementations (e.g.: GeoServer, deegree, and 52° North WPS). There are mature GIS libraries available for Java including GeoTools and JTS Topology Suite. It is not surprising that many of the web apps used PHP for web development, as PHP is part of the LAMP (Linux, Apache, MySQL, PHP) stack of software that is used to power many websites. After Java and PHP, Python was the next most popular approach. Python has recently gained traction as a scripting language in the earth sciences fields, making Python web frameworks

a natural choice for water resources and earth science web apps. Developers with limited programming experience may consider using a system that uses a graphical approach to web design such as Drupal, while more experienced programmers may wish to use a scripting language like PHP or a Python web framework. Table 10 show a summary of some of the notable features of the FOSS web software reviewed.

5. Conclusion

We performed a review of water resources and earth science web apps that were published in the peer-reviewed literature in the last decade to determine which FOSS4G and FOSS web software was used to develop the web apps. The FOSS projects presented in this review do not represent a comprehensive or even representative sampling of all FOSS web GIS and web development software available. The review highlighted 11 FOSS4G software projects and 9 FOSS projects for web development that were used to develop 45 water resources and earth sciences web apps. This constitutes a significantly reduced list of possible FOSS software projects that could be used to meet the needs of water resources web app development—greatly lowering the barrier for entry in this area of development.

Table 10

Comparison of the notable features of web software.

Web software	Number web apps	Language	MVC or similar	Server language	CMS	GUI configuration
Java Framework	17	Java	✓			
PHP	8	PHP		✓		
Perl	1	Perl		✓		
Django	3	Python	✓			
Ruby on Rails	1	Ruby	✓			
CodeIgniter	1	PHP	✓			
Drupal	2	PHP			✓	✓
CKAN	1	Python	✓		✓	✓
Backbone.js	1	JavaScript	✓			

The software review includes FOSS4G projects in the categories of spatial databases, spatial data publishing, mapping libraries, and spatial analysis. In the spatial database category, SpatiaLite, MySQL Spatial, and PostGIS were used in water resources and earth sciences web apps. In the spatial publishing category the web apps reviewed used MapServer, GeoServer, and deegree. In the mapping library category, Google Earth™, Google Maps™, and OpenLayers were used. Software used from the spatial analysis category include 52° North WPS, PyWPS, deegree, and GeoServer. The web apps reviewed were developed using a variety FOSS in the web development category including, Java frameworks, PHP, Perl, the Django Python web framework, the CodeIgniter PHP web framework, Ruby on Rails, Backbone.js, Drupal, and CKAN.

While this review addresses the challenge of identifying FOSS software to provide a web framework and spatial data capabilities for water resources web apps, there are still other hurdles that need to be overcome to make development of such web apps more viable. For example, synthesizing each of the software components into a coherent system is no small task. One solution for this problem would be to collect several FOSS software projects into a single platform that would provide the functionality needed by water resources web apps. Such a platform would provide a development environment optimized for water resources web apps and earth science web apps in general. A majority of the web apps that were included in this review were built from “scratch”. A water resources web app platform would enable future web apps to be developed much more rapidly.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1135482.

References

- 52° North, 2014. Home – 52° North Initiative for Geospatial Open Source Software GmbH. Retrieved 12 March 2013, from: <http://52north.org/>.
- Alconis, J.A., Eco, R.C., Lagmay, A.M.F.A., Aracan, K.A.B., Seveses, L.J.Z., December 2013. The NOAA Initiative: Disaster Management Using WebGIS (Paper Presented at the AGU Fall Meeting, San Francisco, CA).
- Ballatore, A., Tahir, A., McArdle, G., Bertolotto, M., 2011. A comparison of open source geospatial technologies for web mapping. *Int. J. Web Eng. Technol.* 6 (4), 354–374.
- Blodgett, D., Booth, N., Kunicki, T., Walker, J., Lucido, J., 2012. Description of the US Geological Survey Geo Data Portal Data Integration Framework. *Selected Topics in Applied Earth Observations and Remote Sensing IEEE J.* 5 (6), 1687–1691.
- Blower, J.D., Haines, K., Santokhee, A., Liu, C.L., 2009. GODIVA2: interactive visualization of environmental data on the web. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* 367 (1890), 1035–1039.
- Blower, J.D., Gemmill, A.L., Griffiths, G.H., Haines, K., Santokhee, A., Yang, X., 2013. A Web Map Service implementation for the visualization of multidimensional gridded environmental data. *Environ. Model. Softw.* 47 (0), 218–224. <http://dx.doi.org/10.1016/j.envsoft.2013.04.002>.
- Bogdos, N., Manolakos, E.S., 2013. A tool for simulation and geo-animation of wildfires with fuel editing and hotspot monitoring capabilities. *Environ. Model. Softw.* 46, 182–195.
- Booth, N.L., Everman, E.J., Kuo, I.L., Sprague, L., Murphy, L., 2011. A web-based decision support system for assessing regional water-quality conditions and management actions. *JAWRA J. Am. Water Resour. Assoc.* 47 (5), 1136–1150.
- Brooking, C., Hunter, J., 2013. Providing online access to hydrological model simulations through interactive geospatial animations. *Environ. Model. Softw.* (0). <http://dx.doi.org/10.1016/j.envsoft.2013.01.011>.
- Carvalho, L.C., Bernardo, S., Orgaz, M.D.M., Yamazaki, Y., 2010. Forest fires mapping and monitoring of current and past forest fire activity from meteosat second generation data. *Environ. Model. Softw.* 25 (12), 1909–1914.
- Castronova, A.M., Goodall, J.L., Elag, M.M., 2013. Models as web services using the Open Geospatial Consortium (OGC) Web Processing Service (WPS) standard. *Environ. Model. Softw.* 41, 72–83. <http://dx.doi.org/10.1016/j.envsoft.2012.11.010>.
- Cau, P., Manca, S., Soru, C., Mironi, D., Gorgan, D., Bacu, V., ..., Giuliani, G., 2013. An interoperable, GIS-oriented, information and support system for water resources management. *Int. J. Adv. Comput. Sci. Appl.* 3 (3), 75–82. <http://dx.doi.org/10.14569/SpecialIssue.2013.030309>.
- Chambers, J., 2013. The R Project for Statistical Computing. From: <http://www.r-project.org/>.
- Chen, D., Shams, S., Carmona-Moreno, C., Leone, A., 2010. Assessment of open source GIS software for water resources management in developing countries. *J. Hydro-environ. Res.* 4 (3), 253–264. <http://dx.doi.org/10.1016/j.jher.2010.04.017>.
- Choi, J., Engel, B., Farnsworth, R., 2005a. Web-based GIS and spatial decision support system for watershed management. *J. Hydroinf.* 7, 165–174.
- Choi, J., Engel, B., Theller, L., Harbor, J., 2005b. Utilizing web-based GIS and SDSS for hydrological land use change impact assessment. *Trans. ASAE* 48 (2), 815–822.
- CKAN, 2013. CKAN, the World's Leading Open-source Data Portal Platform. From: <http://ckan.org/>.
- Davies, D.K., Ilavajhala, S., Wong, M.M., Justice, C.O., 2009. Fire information for resource management system: archiving and distributing MODIS active fire data. *Geosci. Remote Sens. IEEE Trans.* 47 (1), 72–79.
- de Sousa, L., Eykamp, C., Leopold, U., Baume, O., Braun, C., 2012. iGUESS – a web based system integrating urban energy planning and assessment modelling for multi-scale spatial decision making. In: Seppelt, R., Voinov, A.A., Lange, S., Bankamp, D. (Eds.), Paper Presented at the International Congress on Environmental Modelling and Software Managing Resources of a Limited Planet, Sixth Biennial Meeting, Leipzig, Germany.
- Delipetrev, B., Jonoski, A., Solomatine, D.P., 2012. Cloud Computing Framework for a Hydro Information System. Faculty of Computer Science, University of Goce Delcev, Stip, Republic of Macedonia.
- Delipetrev, B., Jonoski, A., Solomatine, D.P., 2014. Development of a web application for water resources based on open source software. *Comput. Geosci.* 62, 35–42.
- Demir, I., Krajewski, W.F., 2013. Towards an integrated Flood Information System: centralized data access, analysis, and visualization. *Environ. Model. Softw.* 50, 77–84.
- DeVantier, B.A., Feldman, A.D., 1993. Review of GIS application in hydrologic modeling. *J. Water Resour. Plan. Manag.* 119 (2), 1–13.
- Díaz, L., Granell, C., Gould, M., 2008. Case Study: Geospatial Processing Services for Web Based Hydrological Applications. Springer.
- Drupal, 2013. Drupal. From: <https://drupal.org/>.
- Dubois, G., Schulz, M., Skøien, J., Bastin, L., Peedell, S., 2013. eHabitat, a multi-purpose Web Processing Service for ecological modeling. *Environ. Model. Softw.* 41, 123–133.
- Fang, Y., Feng, M., 2009. A WebGIS Framework for Vector Geospatial Data Sharing Based on Open Source Projects.
- Fang, S., Xu, L., Zhu, Y., Liu, Y., Liu, Z., Pei, H., ..., Zhang, H., 2013. An integrated information system for snowmelt flood early-warning based on internet of things. *Inf. Syst. Front.* 1–15.
- Feng, M., Liu, S., Euliss Jr., N.H., Young, C., Mushet, D.M., 2011. Prototyping an online wetland ecosystem services model using open model sharing standards. *Environ. Model. Softw.* 26 (4), 458–468.
- Flaishans, J., Hong, T., Snyder, M., Pascale, C., Purucker, S., 2014. Fronting integrated scientific web applications: design features and Benefits for Regulatory environments. In: Paper Presented at the International Environmental Modelling and Software Society (iEMSS) 7th International Congress on Environmental Modelling and Software, San Diego.
- Foundation, O.S.G., 2014. GEOS – Geometry Engine Open Source. Retrieved 23 October 2014, from: <http://trac.osgeo.org/geos/>.
- Frehner, M., Brändli, M., 2006. Virtual database: spatial analysis in a web-based data management system for distributed ecological data. *Environ. Model. Softw.* 21 (11), 1544–1554. <http://dx.doi.org/10.1016/j.envsoft.2006.05.012>.
- Furieri, A., 2008. SpatiaLite – a Complete Spatial DBMS in a Nutshell. Geographic Information Science. In M. Raubal, H. Miller, A. Frank & M. Goodchild (Eds.), (vol. 4197, pp. 400–417): Springer Berlin/Heidelberg.
- GeoServer, 2013. GeoServer Documentation. From: <http://docs.geoserver.org/>.
- GeoTools, 2014. GeoTools – the Open Source Java GIS Toolkit. Retrieved 23 October 2014, from: <http://geotools.org/>.
- Gkatzoflias, D., Mellios, G., Samaras, Z., 2013. Development of a web GIS application for emissions inventory spatial allocation based on open source software tools. *Comput. Geosci.* 52 (0), 21–33. <http://dx.doi.org/10.1016/j.cageo.2012.10.011>.
- Goodrich, D.C., Guertin, D.P., Burns, I.S., Nearing, M.A., Stone, J.J., Wei, H., ..., Pierson, F., 2011. AGWA: the automated geospatial watershed assessment tool to inform rangeland management. *Rangelands* 33 (4), 41–47.
- Goodrich, D.C., Guertin, D.P., Burns, I.S., Nearing, M.A., Stone, J.J., Wei, H., ..., Pierson, F., 2008. RHEM Web Tool: Rangeland Hydrology Erosion Model Web Tool. *Rangelands*. Retrieved February 28, 2014, 2014, from: <http://apps.tucson.ars.ag.gov/rhem/>.
- Google, 2014. Announcing Deprecation of the Google Earth API. Retrieved January 9, 2015, from: <http://googledevelopers.blogspot.com.au/2014/12/announcing-deprecation-of-google-earth.html>.
- Granell, C., Díaz, L., Gould, M., 2010. Service-oriented applications for environmental models: reusable geospatial services. *Environ. Model. Softw.* 25 (2), 182–198.
- GRASS Development Team, 2014. GRASS GIS: the World's Leading Free GIS Software. Retrieved 23 October 2014, from: <http://grass.osgeo.org/download/>.
- Guzzetti, F., Tonelli, G., 2004. Information system on hydrological and geomorphological catastrophes in Italy (SICI): a tool for managing landslide and flood hazards. *Nat. Hazards Earth Syst. Sci.* 4 (2), 213–232.
- Han, W., Di, L., Zhao, P., Shao, Y., 2012. DEM Explorer: an online interoperable DEM data sharing and analysis system. *Environ. Model. Softw.* 38, 101–107.
- Hazzard, E., 2011. OpenLayers 2.10: Beginner's Guide. Packt Publishing, Birmingham.

- Hill, D.J., Liu, Y., Marini, L., Kooper, R., Rodriguez, A., Futrelle, J., ... McLaren, T., 2011. A virtual sensor system for user-generated, real-time environmental data products. *Environ. Model. Softw.* 26 (12), 1710–1724.
- Holl, S., Plum, H., 2009. PostGIS. *Geoinformatics*, 03/2009, pp. 34–36 doi:citeulike-article-id:4463470.
- Iacovella, S., Youngblood, B., 2013. *GeoServer Beginner's Guide*. Packt Publishing.
- Iwanaga, T., El Sawah, S., Jakeman, A., 2013. Design and implementation of a Web-based groundwater data management system. *Math. Comput. Simul.* 93, 164–174.
- Jung, J.K., Sinha, S.K., Whittle, L.G., 2013. Development of a water infrastructure knowledge database. *J. Infrastruct. Syst.*
- Kulkarni, A., Mohanty, J., Eldho, T., Rao, E., Mohan, B., 2014. A web GIS based integrated flood assessment modeling tool for coastal urban watersheds. *Comput. Geosci.* 64, 7–14.
- Leone, A., Shams, S., Chen, D., 2006. An object-oriented and OpenGIS supported hydro information system (30-HIS) for upper Mersey river basin management. *Int. J. River Basin Manag.* 4 (2), 99–107.
- Li, S., Saborowski, J., Nieschulze, J., Li, Z.-y., Lu, Y.-c., Chen, E.-x., 2007. Web service based spatial forest information system using an open source software approach. *J. For. Res.* 18 (2), 85–90. <http://dx.doi.org/10.1007/s11676-007-0017-9>.
- Li, Y., Fang, X.-L., Jiao, S.-X., 2013. A spatial decision support system for water resource management of Yellow River Basin in China. *Hydraul. Eng.* 83.
- Lim, K.J., Engel, B.A., Tang, Z., Choi, J., Kim, K.S., Muthukrishnan, S., Tripathy, D., 2005. Automated Web GIS Based Hydrograph Analysis Tool, WHAT1. Wiley Online Library.
- Melis, M.T., Locci, F., Dessì, F., Frigerio, I., Strigaro, D., Vuilleumoz, E., 2014. SHARE Geonetwork, a System for Climate and Paleoclimate Data Sharing. Paper presented at the Proceedings of the 7th International Congress on Environmental Modelling and Software (iEMSS).
- Millman, K.J., Aivazis, M., 2011. Python for scientists and engineers. *Comput. Sci. Eng.* 13 (2), 9–12.
- Müller, M., 2007. deegree-Building Blocks for spatial data. *OSGeo J.* 1 (1).
- Nguyen, T.T., 2009. Indexing PostGIS databases and spatial query performance evaluations. *Int. J. Geoinf.* 5 (3), 1–9.
- OGC, 2006. OpenGIS Web Map Server Implementation Specification.
- OGC, 2007a. Catalogue Services Specification.
- OGC, 2007b. OpenGIS Web Processing Service. OGC. (2008). OGC KML.
- OGC, 2010a. Open GIS Web Map Tile Service Implementation Standard. Retrieved 23 October 2014, from. <http://www.opengeospatial.org/standards/wmts>.
- OGC, 2010b. OpenGIS Implementation Standard for Geographic Information – Simple Feature Access – Part 2: SQL Option.
- OGC, 2012a. Geospatial and Location Standards for, 2012, from. <http://www.opengeospatial.org/>.
- OGC, 2012b. OGC Geography Markup Language (GML) – Extended Schemas and Encoding Rules.
- OGC, 2012c. OGC WCS 2.0 Interface Standard – Core: Corrigendum.
- OGC, 2014. OGC Web Feature Service 2.0 Interface Standard.
- Olaya, V., Gimenez, J.C., 2011. SEXTANTE, a Versatile Open-source Library for Spatial Data Analysis.
- Oliphant, T.E., 2007. Python for scientific computing. *Comput. Sci. Eng.* 9 (3), 10–20.
- Oracle, 2012. Your First Cup. Retrieved 28 October 2014, from. <http://docs.oracle.com/javasee/6/firstcup/doc/gkhoy.html>.
- OSGeo, 2014. Benchmarking 2011. Retrieved January 9, 2015, from. http://wiki.osgeo.org/wiki/Benchmarking_2011.
- Oulidi, H.J., Löwner, R., Benaabidate, L., Wächter, J., 2012. HydrIS: an open source GIS decision support system for groundwater management (Morocco). *Geo-spatial Inf. Sci.* 12 (3), 212–216.
- Peres, A., Miletto, E.M., Kapusta, S., Ojeda, T., Lacasse, A., Gagnon, J., 2013. WAITS – an IT Structure for Environmental Information via Open Knowledge, Dynamic Dashboards and Social Web of Things.
- Rao, M., Fan, G., Thomas, J., Cherian, G., Chudiwale, V., Awawdeh, M., 2007. A web-based GIS Decision Support System for managing and planning USDA's Conservation Reserve Program (CRP). *Environ. Model. Softw.* 22 (9), 1270–1280.
- Rojas-Sola, J.L., Castro-García, M., Carranza-Canadas, M. d. P., 2011. Content management system incorporated in a virtual museum hosting. *J. Cult. Herit.* 12 (1), 74–81. <http://dx.doi.org/10.1016/j.culher.2010.10.004>.
- Royappa, A.V., 2000. The PHP web application server. *J. Comput. Sci. Coll.* 15 (3), 201–211.
- Safe Software, 2014. FME Products. Retrieved 23 October 2014, from. <http://www.safe.com/fme/>.
- Sanner, M.F., 1999. Python: a programming language for software integration and development. *J. Mol. Graph Model* 17 (1), 57–61.
- Schut, P., 2007. OpenGIS Web Processing Service. Open Geospatial Consortium Inc., Wayland, MA, USA, p. 87.
- Simao, A., Densham, P.J., Haklay, M., 2009. Web-based GIS for collaborative planning and public participation: an application to the strategic planning of wind farm sites. *J. Environ. Manag.* 90 (6), 2027–2040.
- Singh, P.S., Chutia, D., Sudhakar, S., 2012. Development of a web based GIS application for spatial natural resources information system using effective open source software and standards. *J. Geogr. Inf. Syst.* 4 (3), 261–266. <http://dx.doi.org/10.4236/jgis.2012.43031>.
- Steiniger, S., Hunter, A.J.S., 2012a. Free and open source GIS software for building a spatial data infrastructure. In: Bocher, E., Neteler, M. (Eds.), *Geospatial Free and Open Source Software in the 21st Century*. Springer Berlin, Heidelberg, pp. 247–261.
- Steiniger, S., Hunter, A.J.S., 2012b. Review: the 2012 free and open source GIS software map – a guide to facilitate research, development, and adoption. *Comput. Environ. Urban Syst.* <http://dx.doi.org/10.1016/j.compenvurbysys.2012.10.003>.
- Steiniger, S., Weibel, R., 2010. GIS Software: a Description in 1000 Word Encyclopedia of Geography. University of Zurich – Zurich Open Repository and Archive, London CB.
- Sugrue, J., 2013. *Beginning Backbone.js*. Apress.
- Sun, A., 2013. Enabling collaborative decision-making in watershed management using cloud-computing services. *Environ. Model. Softw.* 41, 93–97.
- Sun, X., Shen, S., Leptoukh, G.G., Wang, P., Di, L., Lu, M., 2012. Development of a Web-based visualization platform for climate research using Google Earth. *Comput. Geosci.* 47 (0), 160–168. <http://dx.doi.org/10.1016/j.cageo.2011.09.010>.
- Tate, B.A., Hibbs, C., 2006. *Ruby on Rails: Up and Running*. O'Reilly Media, Inc.
- The Python Wiki, 2014. *WebFrameworks – Python Wiki*. Retrieved April 9, 2014, from. <https://wiki.python.org/moin/WebFrameworks>.
- Upton, D., 2007. *CodeIgniter for Rapid PHP Application Development*. Packt Publishing Ltd.
- Van Knowe, G., Waight, K., Barlow, P., Yalda, S., Zoppetti, G., Johnson, R., ... Tang, B., 2014. Creating an On-line Interactive Earth Science Environmental Simulator.
- Vatsavai, R., Shekhar, S., Burk, T., Lime, S., 2006. UMN-MapServer: a High-performance, Interoperable, and Open Source Web Mapping and Geo-spatial Analysis System.
- Vivid Solutions, 2014. *JTS Topology Suite*. Retrieved 23 October 2014, from. <http://www.vividsolutions.com/jts/JTSHome.htm>.
- Walker, J.D., Chapra, S.C., 2014. A client-side web application for interactive environmental simulation modeling. *Environ. Model. Softw.* 55, 49–60.
- Wan, Z., Hong, Y., Khan, S., Gourley, J., Flamig, Z., Kirschbaum, D., Tang, G., 2014. A cloud-based global flood disaster community cyber-infrastructure: development and demonstration. *Environ. Model. Softw.* 58, 86–94.
- Warmerdam, F., 2008. *The Geospatial Data Abstraction Library Open Source Approaches in Spatial Data Handling*. Springer, pp. 87–104.
- Zhao, P., Foerster, T., Yue, P., 2012. The geoprocessing web. *Comput. Geosci.* 47 (0), 3–12. <http://dx.doi.org/10.1016/j.cageo.2012.04.021>.