MapMint: The 100% Service Oriented GIS Platform

Fenoy, G.,[1] Bozon, N.[1] and Raghavan, V.[2]
[1]GeoLabs SARL, 1280 avenue des platanes 34970 Lattes, France, E-mail: gerald.fenoy@geolabs.fr
[2]Osaka City University, 3-3-138 Sugimoto, Sumiyoshi-ku, Osaka, Japan

Abstract
An innovative approach to spatial data infrastructure management and geospatial data processing are presented in this paper. Composed of several complementary open source technologies, the MapMint platform introduces a web architecture solely based on open geospatial standards, and orchestrated by a new generation of extensible web processing services. This collection of services automates the data identification, provision and processing workflows from one single node, and gives the ability to generate a variety of standard compliant new web services. It also provide generic methods and tools to configure and publish web mapping applications making use of the available data sources and created web services. The design and implementation of such architecture is explained in the context of centralized and standard compliant spatial data infrastructures, and the structural principles of the MapMint open source platform are presented.

1. Introduction

The establishment of numerous Spatial Data Infrastructures (SDI) (Granell et al., 2007) over the last decades has led to the design and development of new tools for managing, maintaining and using them online through collaborative processes. While publishing and consulting geospatial data onto the Internet may has become a rather common task among Geographic Information Systems (GIS) users (Longley and Maguire, 2005), deploying and managing the full technical side of geospatial servers still implies complicated and time consuming tasks. A number of open source or proprietary technologies are available for publishing and processing geospatial data online, but using them together for building various SDI blocks often requires either to prepare and pre-process GIS data or to adopt a loose coupling approach which may lead to develop specific source code. This paper first reviews some of the adopted open geospatial standards and provides details about their use and implementation for building SDI. Data discovery, delivery and processing services are especially highlighted in the first part, with details on the potential of Web Processing Services. An innovative service oriented architecture for SDI management is then presented in a second part, in which the MapMint open source platform design and development are explained. The resulting web technology is finally presented in a third part as an intelligent and extensible open source geospatial software suite, through a description of its main modules and administration interfaces. The expected research steps as well as ongoing and future developments within the MapMint platform are also reviewed.

2. Orchestrating Open Geospatial Standards

Although the SDI concept is still evolving and adapting to capitalize on new technologies and meet the ever changing needs of society (Williamson et al., 2014), great achievements on geographic data semantics and GIS software interoperability have been reached, and most former GIS data clearinghouses (Crompvoets et al., 2004) have become connected and interoperable SDI. This section intends to review existing geospatial web services and to highlight common practices for their implementation and orchestration.

2.1 Service Oriented Architectures

The geospatial approach to service oriented architecture (SOA) is best represented by the spatial data infrastructure (SDI) paradigm (Granell et al., 2009) in which standardized interfaces allow geospatial web services to communicate with each other according to an interoperable workflow. However, available literature reveals that data discovery and delivery services tend to become systematically available and sometimes tightly coupled, whereas the data processing services are less implemented and may be used for particular needs, sometimes involving the development of a specific part of the SDI architecture based on a different interface standard.

ISSN 1686-6576 / © Geoinformatics International

2.1.1 Data discovery and delivery
Most common open geospatial standards are usually adopted within modern SDI, and their use is encouraged in many aspects of the geographic information life cycle. It is commonly admitted, and furthermore recommended by the INSPIRE European directive (INSPIRE, 2007) that SDI should provide access to geographic information according to data identification and provision standards, including:

- Web Mapping Service (OGC WMS, 2001) and Web Map Tile Service (WMTS), which allow to serve georeferenced map images or tiles, generally using cartographic servers and cache mechanisms.
- Web Feature Service (OGC WFS, 2002) which provides an interface for accessing and querying vector geometries or attributes on the Web, also supported by many map engines and technologies.
- Web Coverage Service (OGC WCS, 2005) which allow to retrieve and query raster coverages and earth observation imagery across the Web.
- Catalog Service for the Web (OGC CSW, 2001) which allow to expose catalogue of records describing geospatial data, services or related resources onto the Web.

Such geospatial web services all rely on Geographic Markup Language (GML) (OGC GML, 200 and Huang et al., 2009) which remains the most appropriate way to express geographic data using XML grammar, and affords interoperability. Modern SDI or online GIS platforms implementing them should also comply to the Web Service Commons (OGC WSC, 2005) defined by the Open Geospatial Consortium, which provides useful information in the context of multiple organizations SDI producing a growing number of services.

2.1.2 Data processing
Processing operations are also increasingly needed in the SDI technical chain in order to transform GIS data, create new information and analysis, and share it to the masses through value added Web Services. The latter helps SDI of public interest to evolve towards intelligent systems and advanced functionalities, but may also sustain geospatial scientific research and innovation. Web processing has become a major technical concern in our data-driven society and is thus subject to standardization since 2007 according to the OpenGIS® Web Processing Service (OGC WPS, 2015) interface standard.

WPS provides rules for standardizing input requests and output responses to/from geospatial processing services. It defines mechanisms for listing processes available at a server (using the *GetCapabilities* request), defining input data required by a particular process (using the *DescribeProcess* request), initiating the calculation and managing the output response (using the *Execute* request), so it can be interpreted by client applications.

2.2 WPS Implementation
Several WPS implementations (Fenoy et al., 2013, Schaeffer et al., 2012 and de Jesus et al., 2012) are available and commonly used to provide processing capabilities to existing GIS platforms or SDI (Evangelidis et al., 2014) set up advanced spatial analysis capabilities to Web GIS applications (Ninsawat et al., 2007 and Castronova et l., 2013) model specific spatial phenomena (Dubois et al., 2013) or to implement multi input/output spatial based environmental models. Past research includes the design and development of the ZOO-Project open source WPS platform (Fenoy et al., 2013) which provides an extensible framework for creating and chaining WPS 1.0.0 and WPS 2.0 compliant Web Services. The WPS platform main properties are summarized in this section as an introduction to the service oriented MapMint architecture.

2.2.1 Web processing platform
The ZOO-Project WPS implementation intends to cover the full WPS spectrum from building WPS requests to chaining processes. It is composed of the four main components described below:

- ZOO-Kernel, a WPS implementation written in C, also known as the ZOO-Project WPS server, is a polyglot able to load shared libraries and execute source code written in several programming languages, which is useful for implementing scientific models and reusing legacy code.
- ZOO-Services, which provides a set of ready-to-use WPS Services covering a wide spectrum of models and algorithms, based on reliable open source libraries such as GDAL, CGAL (Boissonnat et al., 2000), GRASS GIS (Neteler and Mitasova, 2002) or SAGA GIS (Böhner et al., 2006).
- ZOO-API, which is a server-side Javascript API designed to instantiate and chain available WPS Services using straightforward expressions and logical operators with simple scripting.
- ZOO-Client, which is a client-side Javascript library providing simple methods to interact with WPS servers from web applications.

It is helpful for sending requests to any WPS compliant server and to parse the output responses using simple Javascript.

2.2.2 Nested geospatial interoperability
WPS not only executes a computation on a remote server machine, it does provide a very flexible interface to access data and use it as input of such a computation. This includes spatial databases and GIS data in several forms, but also the one already published through data discovery and delivery web services. Following an interoperable logic, a WFS or WCS request can indeed be taken as input of a WPS service which would execute a computation and return a new data source that could itself be published as new WFS or WCS, or any other service. '*Web services transforming web services into new web services*' defines WPS as a logical and sustainable catalyst for open geospatial web services orchestration.

3. 100% Service Oriented Architecture
In the light of existing SDI and established good practices, our current research extends the previously described WPS implementation in order to design innovative methods for SDI management, and to provide extensible tools to process, publish and share GIS data. As notified in the related specification, WPS can be used to readily insert non-spatial processing tasks into a web services environment, and this point is precisely the root of the proposed MapMint architecture.

3.1 Extensible and Modular SOA
WPS is the corner stone of the MapMint architecture in which everything is powered by a dedicated WPS Service. As shown in Figure 1, each functional module corresponds to a thematic WPS *Service Provider,* which hosts several complementary WPS Services empowering the SDI.

3.1.1 Structural elements
The SDI structural elements such as configuration, analytics, users management and user interfaces are first of all managed by WPS services which allow fast and efficient SDI deployment and maintenance. The key component installation parameters such as WPS and WMS/WFS/WCS servers location path, URL or version can be, for example, modified using a service which outputs the ZOO-Project main configuration file (*main.cfg*). The latter contains many other settings which can be edited, such as default encoding, which would imply modifications in the public part of the SDI. The SDI metadata, namely WSC such as *title, description and abstract* or *provider* information is shared by the WPS, WMS, WFS and WCS servers and included in the XML responses, but also sometimes used as user-defined content into client-side applications. SDI users and groups of users are also managed using WPS services able to query and edit embedded local user database or existing user's registry. The user permissions on data stores, data sources, maps, web applications or web services can also be defined using several security WPS services taking user-defined values and session information as input.

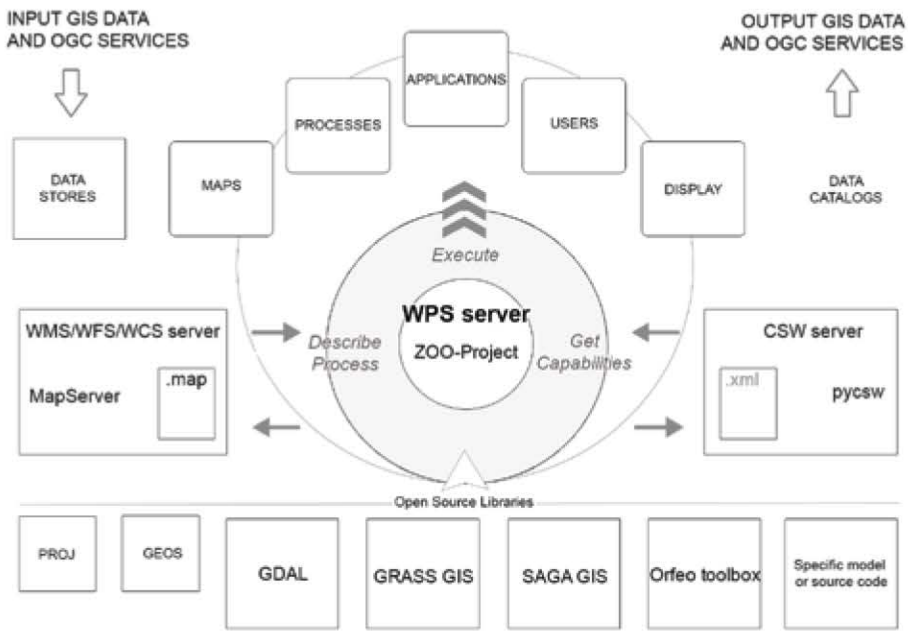

Figure 1: MapMint architecture overview

Such services are acting as security proxies that can be applied to most of the SDI resources, providing a granular permissions system according which users can read (*r*), write (*w*) and/or execute (*x*) at many different levels. Such WPS services allow to secure parts of the server-side infrastructure and to monitor the access to data or service, but also address authentication and session management tasks when needed on the client side.

3.1.2 Data abstraction approach

Inspired from the Geospatial Data Abstraction Library (GDAL; Warmerdam, 2008) capabilities and its many GIS format drivers, MapMint introduces a flexible data management system allowing to store and organize local or remote data stores. These can be directories of the local file system, in which case symbolic links would be established between the physical directory and the MapMint data store, local or remote databases, in which case ODBC and SQL would be used to interact with schemas databases and tables, or remote WMS/WFS/WCS servers in which case the connexion to the data stores would be established using standard *GetCapabilities* requests. Once a data store created, WPS Services based on the GDAL *ogrinfo* and *gdalinfo* utility programs would scan it and provide metadata information on every supported data sources it contains. This allow to list the data sources which another Service would write to a mapfile (Vatsavai et al., 2006), the MapServer map configuration file. Data sources become layer elements and can be transparently published as WMS (and thus be directly previewed using simple *GetMap* requests), but also as WFS or WCS Services according to their type, so that data attributes can be retrieved and displayed into attribute tables (using *GetFeature* requests) or pixels histograms (using *GetCoverage* requests). Once published, every data source is available through data provision services and potentially candidate for processing and publishing.

3.1.3 Processing web services

Basic GIS data transformations can be performed on vector and raster data sources alike, such as format and cartographic projection conversion which can be operated by WPS Services equivalent to the GDAL *ogr2ogr* and *gdal_translate* utility programs. These would output new data sources with new properties in the targeted data store, which is thus candidate for WMS/WFS/WCS publication. Vector data sources can be used as input of a number of other WPS Services dedicated to single geometry (such as buffer, centroid, convex-hull, boundary or simplify) or multiple geometries (such as union, intersection, difference or symmetric difference) spatial analysis operations, triangulations (Boissonnat et al., 2000) (such as Delaunay, Voronoi or Thiessen) or interpolations (such as inverse distance weighted, nearest neighbours or kriging). Raster data can also be used as input of a WPS Services dedicated to single-band or multiple bands image processing, allowing to perform several basic manipulations (such as resampling, scaling, filtering, cropping...), terrain analysis (such as relief, shaded relief slopes, roughness or contour lines) as well as advanced raster mathematics that can be computed between bands of one or several image data sources. These numerous algorithms provide advanced processing tools to the SDI and the created output are stored as new data sources which can directly be used and published. The MapMint data processing system is moreover highly extensible, as new algorithms and functionalities can be developed and added according to the SDI needs with seamless integration, through new WPS service. The latter would basically extend the WPS server *GetCapabilities* output response, and the new services could easily be used or added to existing chaining. Needed interfaces or configuration forms to be added on the client side may also be automatically generated according to the information extracted from the *DescribeProcess* output response.

3.2 Dynamic web map configuration

Data sources available as WMS/WFS/WCS can be used to create map compositions automatically saved as mapfiles from an intuitive cartographic user interface. Each newly created map project becomes a composite mapfile in which the *layer* blocks are dynamically written as the user adds selected data sources to the map and defines their properties. This part of the MapMint platform indeed provides a complete administration interface to the MapServer (Vatsavai et al., 2006) software, which automates its usage and significantly speeds up the map creation and publication processes. Such a mapfile authoring tool not only provides advanced web-based cartographic software for everyone, it also allows the user to quickly retrieve and easily update any map at any anytime.

3.2.1 Map contents

Every mapfile parameter is supported by the mapfile WPS Service and data layers can be represented using cartographic styles such as unique symbol, graduated symbols, continuous color, unique value, grey-scaled or time line, depending on their type (raster or vector) and their geometries (points, lines or polygons).

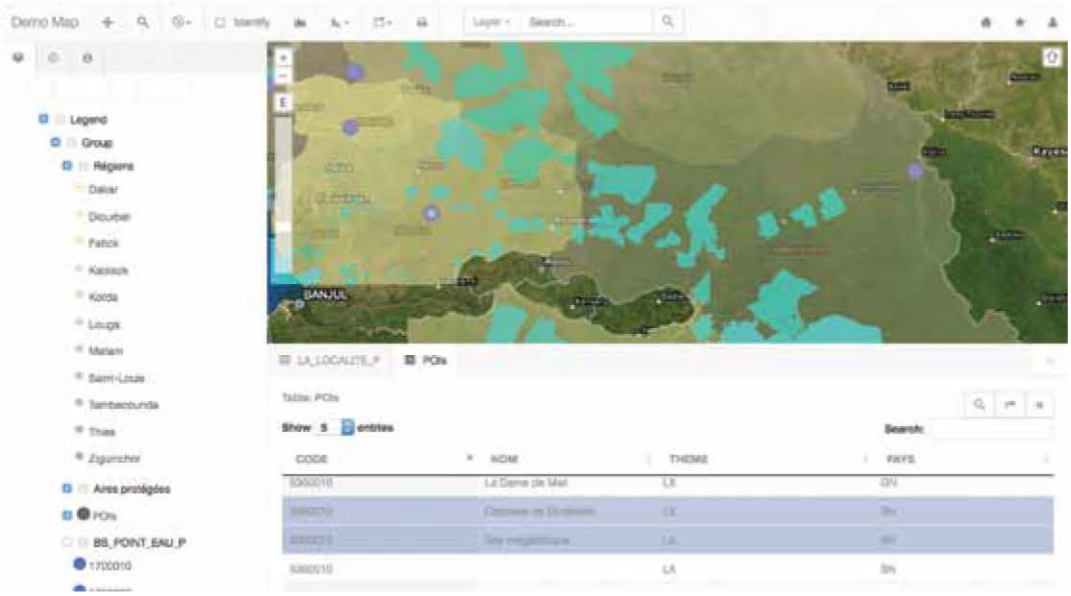

Figure 2: Example generated web-mapping application

Classifications can be performed according to attribute fields or a value ranges and the resulting classes are written as *expressions* in the corresponding *layer* block of the concerned mapfile. The cartographic message can moreover be customized using one of the available discretization methods (such as quantiles, Jenks, Fisher or kmeans) which will relatively alter the targeted classification transcribed in the mapfile. Advanced symbols can also be configured from the MapMint administration and layers can be publish as raster color ramps or vector choroplets. A number of other layer properties can be set using the MapMint visual mapping interface such as their position in the layer tree, their minimum or maximum display scale, the position and orientation or their text labels or their respective opacity to list only a few of the web cartographic capabilities. The interaction properties of selected data sources (i.e. map layers) can also be configured, for example to define if a layer can be used for attribute query, spatial analysis, search engine or any other WFS or WPS driven MapMint functionality.

3.2.2 Map containers

The created maps (i.e. mapfiles) can finally be taken as input of WPS Services dedicated to web applications publishing, along with user-defined parameters such as map elements presentation layout, map and layers default properties at initialization, layer events and rendering properties, as well as map controls or widgets to be included in the application. Simple navigation tools or map event based functionality available from the OpenLayers (Santiago, 2014) web-mapping library can thus be selected and made available in the published web applications. The publication process is finalized by a WPS service which role is to generate the application source code and resources from composite input including the selected mapfile, an HTML template, as well as generated javascript and cascading stylesheets according to the user choices and selections. Such *display* WPS service outputs ready to use web maps, but also any needed HTML markup (such as lists, trees, forms, toolbars, notifications...) or specific resource (such as document or image) that may be needed by applications defining the SDI (Figure 2).

4. Results

4.1 MapMint Open Source Framework

The presented architecture has been implemented into a functional web platform which source code was released under an open source MIT/X-11 style license (Fenoy et al., 2014). The MapMint SDI software suite is composed of a reliable C core engine including ZOO-Project, MapServer and GDAL by default, a collection of extensible Python WPS Services performing numerous SDI management tasks, and a set of modern user interfaces and experiences to interact with. This whole provide a generic and extensible system for the deployment and management of SDI, able to access, process and publish geospatial data through the use of standard compliant web services, but also to generate advanced custom Web GIS applications using those services. Both WMS, WFS, WCS, CSW and WPS services are powered and orchestrated within the SDI. They can be exploited internally by MapMint applications, or used by third-party SDI, web servers or external clients. Once the needed software components, libraries and dependencies are installed, and the *main.cfg* file correctly edited according to the server environment, the MapMint administration modules and public applications only require a web browser to be used. According to the authentication and session WPS services response, the user would access different user interfaces and functionalities. The user would create, edit and share some of the data stores, data sources, maps and/or applications, depending on his role and permissions. While SDI structural elements, configuration and analytics will often be edited by systems administrators only using the MapMint *Dashboard* module, most data scientists and managers would take advantage of the *Distiller* module for processing and publishing existing or new data sources in a standardized and way. Thematic cartographic composition would be performed within the *Manager* module by every user, whereas applications configuration and publishing may be accomplished only by authorized profiles using the *Publisher* module. The various SDI management tasks are most of the time performed through simplified user experiences, but more complexity is systematically proposed to more advanced users, for example to extend existing services using python and *mapscript,* or to adapt and enrich web mapping applications with new map templates, tools and interfaces.

4.2 Future Works

Significant research and development efforts are currently being conducted in order to optimize the ZOO-Project WPS server and strengthen its capabilities in the context of geospatial open data (Evans et al., 2014 and Lee and Kang, 2015) and high-traffic geospatial web applications. This research introduces scalable web processing techniques implementation inside the WPS server, in order to schedule, queue and thus prioritize on WPS processes according to their type, expected duration and status of their completion, taking advantage of the ZOO-Project *GetStatus* utility service. This will benefit the MapMint platform as its WPS engine will reach better performances and be able to manage concurrent requests faster. It will also better integrate massive SDI with the capacity of using high volumes of remote input data. The MapMint administration interface currently allows to request many WPS services individually or to query multiple services according to predefined WPS chaining workflows. Current research intends to define new techniques for WPS chaining through the design and development of a WPS processes model builder. The latter can be compared to a common GIS model builder able to configure input/output of multiple WPS services into a functional processing chain to be executed on a server. This may also give the ability to export both the service source code and the needed resources to use it to any distant WPS compliant server, which would introduce the concept of transactional web processing. Another major functional enhancement which has also been initiated is the integration of CSW (OGC WCS, 2015) support into the MapMint data management system. Inspired by the Publicamundi project (Tzotsos et al., 2015) architecture and cataloging abilities, WPS Services based on the *pycsw* library (a reference CSW python implementation) are actually being designed and will allow to directly expose the MapMint data sources also as CSW. This will significantly extend the MapMint platform capabilities, as the published data sources would feed the SDI maps and catalog at the same time. The CSW nodes could also be used as input of catalog oriented WPS services using such integrated metadata management system.

5. Conclusion

A new innovative approach to SDI management and geospatial data processing was presented in this paper. It has been implemented through the development and release of the MapMint open source platform. Used, adapted and maintained in production in several SDI, MapMint allows to generate geospatial web services and applications which are adding value to geospatial data generated by the research, government or industry sectors. As a generic and open source technology, MapMint can easily be deployed and scale to the need of any SDI thanks to a unique services oriented architecture. It can be adapted and extended to provide simple to advanced data transformation chains and solve various web challenges. Ongoing research and next development steps will provide additional methods which will ease, optimize and accelerate the use of the WPS driven platform for even more extensible and sustainable SDI.

References

Böhner, J., McCloy, K. R. and Strobl, J., 2006, SAGA-Analysis and Modelling Applications. University of Goettingen.

Boissonnat, J. D., Devillers, O., Teillaud, M. and Yvinec, M., 2000, Triangulations in CGAL. *In Proceedings of the Sixteenth Annual Symposium on Computational Geometry*, 11-18.

Castronova, A. M., Goodall, J. L. and Elag, M., 2013, Models as Web Services using the Open Geospatial Consortium (OGC) Web Processing Service (WPS) Standard. *Environmental Modelling and Software*, 41, 72-83.

Crompvoets, J, Bregt, A, Rajabifard, A. and Williamson, I. P., 2004, Assessing the Worldwide Developments of National Spatial Data Clearinghouses. *International Journal of Geographical Information Science*, 18, 665-689.

de Jesus, J., Walker, P., Grant, M. and Groom, S., 2012, WPS Orchestration using the Taverna Workbench: The eScience approach. *Computers and Geosciences*, 47, 85-86.

Dubois, G., Schulz, M., Skøien, J., Bastin, L. and Peedell, S., 2013, eHabitat, a Multi-Purpose Web Processing Service for Ecological Modelling. *Environmental Modelling and Software*, 41, 123-133.

Evangelidis, K., Ntouros, K., Makridis, S. and Papatheodorou, C., 2014, Geospatial Services in the Cloud. *Computers and Geosciences*, 63, 116-122.

Evans, M. R., Oliver, D., Zhou, X. and Shekhar, S., 2014, Spatial Big Data. Big Data: Techniques and Technologies in Geoinformatics, 149-173.

Fenoy, G., Bozon, N. and Raghavan, V., 2013, ZOO-Project: The Open WPS platform. Applied Geomatics, 5(1), 19–24

Fenoy, G., Bozon N., Raghavan V., 2014, MapMint source code, https://github.com/mapmint/

Granell, C., Gould, M., Bernabé, M. A. and Manso, M. A., 2007, Spatial Data Infrastructures. *Encyclopedia of Geoinformatics*, Idea Group Press.

Granell, C., Díaz, L. and Gould, M, 2009, Distributed Geospatial Processing Services. *Encyclopedia of Information Science and Technology*, 1186-1193.

Huang, C. H., Chuang, T. R., Deng, D. P. and Lee, H. M., 2009, Building GML-Native Web-Based Geographic Information Systems. *Computers and Geosciences*, 35(9), 1802-1816.

INSPIRE, 2007, Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community. *Official Journal of the European Union*. L108, 1-14.

Lee, J. G. and Kang, M., 2015, Geospatial Big Data: Challenges and Opportunities. Big Data Research, 2, 74-81.

Longley P. A. and Maguire D. J, 2005, The Emergence of Geoportals and their Role in Spatial Data Infrastructures. Computers, *Environment and Urban Systems*, 29(1), 3-14.

Neteler, M. and Mitasova, H., 2002, Open Source GIS: a GRASS GIS Approach. Springer Science and Business Media, 3rd edition.

Ninsawat, S., RaghavanV., Masumoto, S. and Chemin, Y., 2007, Web Processing Service for Spatial Analysis using PyWPS and GRASS GIS. *International Journal of Geoinformatics*, 3, 12-16.

OGC GML, OpenGIS®Geographic Markup Language Standard Specification, 2000, Open Geospatial Consortium.

OGC CSW, OpenGIS®Catalog Service for the Web Standard Specification, 2001, Open Geospatial Consortium.

OGC WMS, OpenGIS® Web Map Service Standard Specification, 2001, Open Geospatial Consortium.

OGC WFS, OpenGIS® Web Feature Service Standard Specification, 2002, Open Geospatial Consortium.

OGC WCS, OpenGIS® Web Coverage Service Standard Specification, 2005, Open Geospatial Consortium.

OGC WSC, OpenGIS® Web Service Common Standard Specification, 2005, Open Geospatial Consortium.

OGC WPS, OpenGIS®Web Processing Service 1.0.0 and 2.0, 2015 Standard Specifications, Open Geospatial Consortium.

Santiago, A., 2014, The book of OpenLayers 3. Lean Publishing.

Schaeffer, B., Baranski, B., Foerster, T. and Brauner, J., 2012, A Service-Oriented Framework for Real-Time and Distributed Geoprocessing. Geospatial Free and Open Source Software in the 21st Century, 3-20.

Tzotsos, A., Alexakis, M., Athanasiou, S. and Kouvaras, Y., 2015, Towards Open Big Geospatial Data for geodata.gov.gr, *Geomatics Workbooks*, 12, 247-258.

Vatsavai, R. R., Shekhar, S., Burk, T. E. and Lime, S., 2006, UMN-MapServer: A High-Performance, Interoperable and Open Source Web Mapping and Geo-Spatial Analysis System. *Geographic Information Science*, 400-417.

Williamson, I. P., Rajabifard, A. and Feeney, 2004, Developing *Spatial Data Infrastructures: from Concept to Reality*. CRC Press.

Warmerdam, F., 2008, The Geospatial Data Abstraction Library, 2008, *Open Source Approaches in Spatial Data Handling*, 87-104.