RESULTS OF AN EVALUATION OF THE ORCHESTRATION CAPABILITIES OF THE ZOO PROJECT AND THE 52° NORTH FRAMEWORK FOR AN INTELLIGENT GEOPORTAL

V. Rautenbach[a], S. Coetzee[a,*], M. Strzelecki[b], A. Iwaniak[b]

[a]*Centre for Geoinformation Science, University of Pretoria, Pretoria, 0002, South Africa*
[b]*Wrocław University of Environmental and Life Sciences, ul. Grunwaldzka 53, 50-357 Wrocław, Poland*

Commission IV, WG IV/5

KEY WORDS: Services, Web based, Distributed, Spatial Infrastructures, Standards, Thematic, Cartography

ABSTRACT:

The aim of a spatial data infrastructure (SDI) is to make data available for the economical and societal benefit to a wide audience. A geoportal typically provides access to spatial data and associated web services in an SDI, facilitating the discovery, display, editing and analysis of data. In contrast, a spatial information infrastructure (SII) should provide access to information, i.e. data that has been processed, organized and presented so as to be useful. Thematic maps are an example of the representation of spatial information. An SII geoportal requires intelligence to orchestrate (automatically coordinate) web services that prepare, discover and present information, instead of data, to the user. We call this an intelligent geoportal. The Open Geospatial Consortium's Web Processing Service (WPS) standard provides the rules for describing the input and output of any type of spatial process. In this paper we present the results of an evaluation of two orchestration platforms: the 52° North framework and ZOO project. We evaluated the frameworks' orchestration capabilities for producing thematic maps. Results of the evaluation show that both frameworks have potential to facilitate orchestration in an intelligent geoportal, but that some functionality is still lacking; lack of semantic information and usability of the framework; these limitations creates barriers for the wide spread use of the frameworks. Before, the frameworks can be used for advance orchestration these limitations need to be addressed. The results of our evaluation of these frameworks, both with their respective strengths and weaknesses, can guide developers to choose the framework best suitable for their specific needs.

1. INTRODUCTION

A spatial data infrastructure (SDI) is the basis for spatial *data* discovery, evaluation and application for users and providers within all levels of government, the commercial sector, the non-profit sector, academia and by citizens in general (GSDI, 2004). The entry point to an SDI is a geoportal, which provides access to spatial data and associated web services in an SDI, facilitating the discovery, display, editing and analysis of data.

A spatial information infrastructure (SII) should provide access to information, i.e. data that has been processed, organized and presented so as to be useful. The geoportal of an SII requires intelligence to orchestrate (automatically coordinate) web services that prepare, discover and present information, instead of data, to the user. The SII's geoportal is referred to as an intelligent geoportal: a geoportal that provides complex functionality through user interface for a user in a specific application domain (Iwaniak *et al*., 2011).

Standards by the Open Geospatial Consortium (OGC) and the International Organisation for Standardisation's (ISO) technical committee ISO/TC 211, *Geographic information/Geomatics*, facilitate interoperability in a geoportal. The most widely used web service standards in geoportals are the Web Map Service (WMS) and the Web Feature Service (WFS). The Styled Layer Descriptor (SLD) is a visualisation standard from the OGC for specifying user defined styles and the manner in which the WMS shall render the map (OGC, 2007a). The WMS and WFS services, together with SLD enable the display of spatial data in a geoportal (OGC, 2006, OGC 2010). Spatial processing can be added to the geoportal with Web Processing Service (WPS) implementations (OGC 2007b). The WPS provides a standardised interface for executing processing services via the Internet.

Interoperability of web services, whether in an SDI or an SII, requires the development and use of uniform standards. Multiple OGC web services are required to prepare and present information in an intelligent geoportal of an SII. Web service orchestration makes it possible for web services to collaborate in a predefined pattern or to be automatically arranged based on local decisions about their interactions with one another at the message and/or execution level (Sun *et al.*, 2010, Suazo & Aguirre, 2005). In orchestration a single process controls and executes the underlying processes in the particular order in which they were specified and assigns the necessary input parameters for the WPS processes. The orchestration process can be altered depending on the output of an individual process; semantic descriptions of individual processes or services available to the orchestration process are used for this.

Cartography methods can be used to create and prepare information from raw data for presentation in the SII's intelligent geoportal. An example of the representation of information is a thematic map, representing spatial or physical phenomena and statistical information associated with a particular geographic area (Slocum *et al*., 2009). Thematic maps have become increasingly popular in the last few years, especially in using web services to produce thematic maps. This is contributed to the increased use of web services in general, but also in SDIs specifically (Maguire & Longley, 2005).

* Corresponding author. Email: serena.coetzee@up.ac.za • Tel: +27 12 420 3823 • Fax: +27 12 420 6385

At a recent FOSS4G conference, Garnett and Fenoy (2011) presented on a WPS shootout, for which they tested five WPS frameworks: 52° North, constellation, deegree, GeoServer, PyWPS and ZOO project. The shootout evaluated conformance of these frameworks to the OGC WPS 1.0.0 standard and the interoperability of the respective WPS frameworks. All the evaluated frameworks, except the constellation project, passed the tests set out in the shootout. The frameworks we evaluate in this study were included in the shootout and thus are compliant with the OGC standard and provide interoperability. In another study, Lopez-Pellicer *et al*. (2011) conducted a survey of the availability of WPS services, which showed that out of the 9,329 OGC services discovered on the web, only 58 were WPS services. The survey confirms the OGC implementation statistics (OGC, 2011), which state that the WMS specification is the most implemented of the OGC web services with 46.1% of services discovered. Lack of documentation, profiles and few WSDL descriptions available are some of the limitations that were identified; these limitations create semantic and technical barriers for orchestration of web services.

Business process execution language (BPEL) is an XML based workflow language that composes services by choreographing service interactions (Weerawarana *et al*., 2005). BPEL provides control logic operations, event handling, extensibility, and web service policies. Stollberg and Zipf (2007) investigated the efficiency of BPEL and WPS for the orchestration of OGC web services. They found that OGC web services do not support the Simple Object Access Protocol (SOAP) and could therefore not be used in BPEL orchestration engines. Another problem is that orchestration engines could not transfer raw binary data served by the WMS GetMap, for example, thus BPEL could not be used for web service orchestration. A WPS service was successfully used to orchestrate web services in these experiments. OGC has since included SOAP support for their web service standards, eliminating one of the limitations. As a solution to the transfer of raw data, Fleuren and Muller (2008) proposed using a PostGIS database to store intermediate data for orchestration with an ActiveBPEL engine.

In 2010, Foerster *et al*. stated that composition of web services is still difficult due to interoperability problems. Experiments were done using the 52° North framework to successfully perform map generalization and transformation. They suggested a Content Transformation Services to address interoperability problems. Workflow modellers emerged to solve the above-mentioned shortcomings of WPS services. The modellers have since become part of the WPS frameworks that we evaluated.

In this paper we present the results of an evaluation of two frameworks for the orchestration of web services, namely the 52° North geoprocessing framework (www.52north.org/wps) and the ZOO project (http://www.zoo-project.org). Our evaluation is aimed at comparing the capabilities of each framework for producing thematic maps in an intelligent geoportal. The frameworks were selected for their prominence at conferences and the number of papers written about them. Both frameworks are open source implementations of the OGC WPS standard, developed by research companies in Germany and Japan respectively, and have grown to an international user base. The remainder of the paper is structured as follows: first, we provide a brief overview of the WPS standard and present the two orchestration frameworks; next, we describe the evaluation methodology and results; finally, we discuss the results and their implications for SDIs, SIIs, web standards and thematic maps over the Web.

2. OVERVIEW OF WPS STANDARD

The web processing service specification was first released in 2005 as a discussion paper (Foerster & Stoter, 2009), since then WPS has become an OGC web service standard with version 1.0.0 currently available (OGC, 2007b). The purpose of the WPS standard is to describe a service that provides spatial data processing functionality, which can be executed in a web environment. WPS provides a standardised interface that facilitates the publishing of spatial processes on the web. The OGC WPS standard (2007) describes a process as any algorithm, calculation or model that operates on spatially referenced data. The standard also describes publishing as the making available of machine-readable binding information and human-readable metadata for service discovery and use. WPS can thus standardize the interface for executing any spatial processes or calculations and provide the process as a web service that can be accessed via the Internet (Fenoy *et al*., 2010).

At the end of 2011 there were 21 WPS implementations registered with OGC, but no OGC compliant implementations of the WPS standard are available presently (OGC, 2011). A survey done in 2011 on the availability of WPS standards showed that most of the WPS implementation were implemented by universities, private companies and government agencies (Lopez-Pellicer *et al*., 2011). 84% of WPS servers discovered were implemented in Europe, with universities as the implementer. A workflow modeller is in some cases implemented together with the WPS implementation. According to Fenoy *et al*. some of the most popular open source WPS implementations are 52° North, ZOO project, deegree, WPSint and PyWPS (2010).

3. OVERVIEW OF WPS FRAMEWORKS

In this section a brief overview of the 52° North framework and ZOO Project are provided.

52° North framework, founded in 2004, is an open source framework that enables the deployment of spatial processes on the web in a standardised way (www.52north.org/wps). 52° North framework is a Java based pluggable framework that offers spatial processes via the OGC WPS interface (Baranski, 2008). WPS processes can be created using standard Java libraries and other 3rd party libraries. The 52° North framework has a web admin tool that can be used to upload WPS processes, connect to a repository of WPS processes and to adjust server settings.

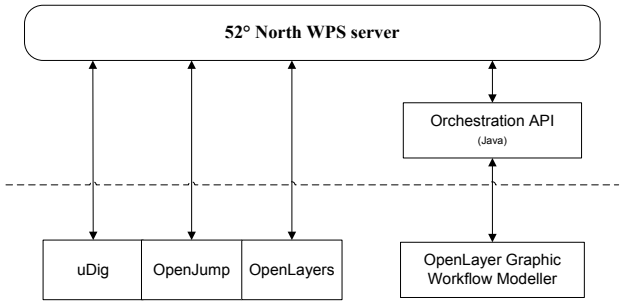

Figure 1. Depicts the architecture of the 52° North framework

52° North framework provides a WPS orchestration API and a graphical modelling tool for geoprocessing workflows based on the orchestration API (refer to figure 1). The modelling tool,

known as the WPS Workflow Modeller, allows visual modelling of geoprocessing workflows in an Internet browser environment. The workflow modeller allows the programmatic chaining of WPS services and complex data inputs from e.g. other OGC services such as WFS.

The ZOO project, founded in 2008, is an open source project that provides a developer friendly WPS framework for the development of WPS services (http://www.zoo-project.org/). The ZOO project consists of three main components, namely (refer to figure 2): the ZOO kernel, ZOO services and the ZOO Application Programming Interface (API). The ZOO kernel is a server-side C kernel, which forms the core of the ZOO project and allows the creation, management and chaining of WPS 1.0.0 compliant web services. The kernel supports a wide variety of programming languages: C, Python, Java, JavaScript, Fortran and PHP, and can easily be connected to external spatial libraries, scientific models, cartographic engines and spatial databases. The large variety of languages is convenient for developers, allowing the re-use of existing code to create new web services with the ZOO project.

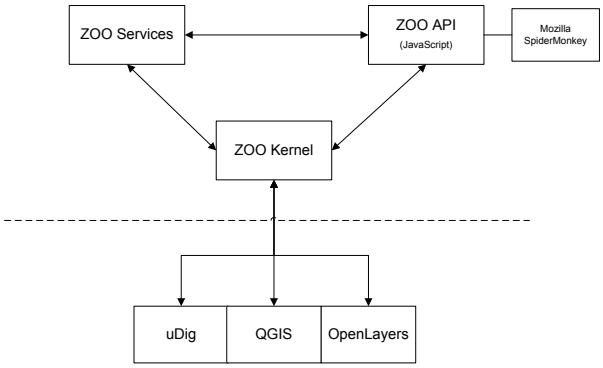

Figure 2. Depicts the architecture of the ZOO framework, adapted from http://www.zoo-project.org/

The ZOO service provider consists of the Service Shared Object (SSO) and a metadata ZOO configuration file (.zcfg) per provided service (Fenoy *et al*., 2010). This facilitates the GetCapabilities and DescribeProcess request specified in the WPS 1.0.0 standard specification. The kernel parses the zcfg file using Flex and Bison. The ZOO API is a server-side JavaScript library designed to simplify the WPS process creation and chaining (Fenoy *et al*., 2010). The API provides ready-to-use JavaScript functions for handling WPS requests via HTTP. Orchestration of WPS services is possible with the API through the use of a specific method; the method also provides the ability to add a level of control and logic to the WPS chaining. The *proj4js* library is incorporated into the API, allowing server-side reprojection of spatial data.

The frameworks are platform independent, but it can be deduced from the available installers, ease of installation on the different frameworks, and the operating system used in the documentation, that the ZOO framework is designed originally for Linux and the 52° North framework for Windows.

4. EVALUATION AND RESULTS

For our evaluation of the suitability of the 52° North and ZOO Project frameworks for thematic maps, we included the following:

1. Documentation and support available.
2. Installation of the framework.
3. Support for protocols, such as, RESTful, SOAP and HTTP GET/POST.
4. Process management capabilities, e.g. whether user interference is possible once the orchestrated process has been started.
5. Workflow construction capabilities.
6. Support for asynchronous processing.
7. Capacity to provide and use semantic descriptions.
8. Data transport mechanism, e.g. whether the data is included in the request or sent pointed to by a URL.
9. The flexibility of parameters, e.g. whether are parameters required and whether additional parameters can be added.
10. Ease of extending the framework, e.g. the ability to add additional functionalities required.
11. Ease of integration with other GIS applications.
12. Support for the orchestration of external OGC services, e.g. calling a GeoServer WFS to obtain attribute data.

For our evaluation of the frameworks, we orchestrated OGC standard web services and components to produce thematic maps. The orchestration process is based on experiments we are currently busy with. Our current findings show that on both frameworks it is possible to orchestrate a thematic web service from standard components, such as WMS, WFS and WPS, that make use of other standards, such as SLD and Common Query Language (CQL). Some customized programming is required, such as the classification of the dataset and generation of SLD. These processes need to be wrapped into WPS services that make the orchestration of standard web service components possible. The evaluation was performed on a computer running Mac OS X Lion 10.7.2 with Xcode 4.2. The ZOO project version 1.2, 52° North WPS 2.0 RC7 and 52° North workflow modeller version 1.0.0 released December 2011 was used for this evaluation. Table 1 shows the results of our evaluation. A discussion of these results follows in Section 6.

Criteria	52° North WPS framework	ZOO Project
1. Documentation and support available	Documents and tutorials are available online, however, they are outdated and refer to old versions of the software. An online forum and mailing list are available; a post to forum was answered quickly and very proficiently. Documentation mainly covers Windows OS.	Documents and tutorials are available online in the following formats: HTML, epub and pdf. Documentation is up-to-date and detailed. The available tutorial is aimed at OSGeo live disk (Linux). An online forum and mailing list are available; a post to forum was answered quickly and very proficiently.
2. Installation of the framework	An installer is available for Windows, with an embedded version of Tomcat. For Linux and Mac a WAR file must be deployed in a Tomcat server container.	The installation of ZOO is not simple. An installer is available for Mac, but still required additional libraries. In our experience it was a daunting process to obtain the library dependencies and correct versions in order to compile the framework.

3. Support for protocols	Framework support for SOAP and HTTP GET/POST. Exposes the service as a WSDL document.	Framework support for HTTP GET/POST. SOAP support is currently not implemented in the framework (except for WPS requests and responses that can be packed into a SOAP:Envelope).
4. Process management capabilities	No mechanism for process intervention is currently implemented, but can easily be supported through the use of signals provided by the inter process communication (IPC), such as PAUSE and QUIT. Status reporting is supported through shared memory messaging.	No mechanism for process intervention is currently implemented, but can easily be supported through the use of signals provided by the inter process communication (IPC), such as PAUSE and QUIT. Status reporting is supported through shared memory messaging.
5. Workflow construction capabilities	A graphic user interface (GUI) is available for implementing the orchestration API, but offers no control logic operations, e.g. if and while.	The ZOO API is a JavaScript API that runs server-side. It provides control logic operators and the functionality for web service chaining.
6. Support for asynchronous processing	Synchronous and asynchronous invocation.	Synchronous and asynchronous invocation.
7. Capacity to provide and use semantic descriptions	The user can create a process description file, containing information, such as, title and abstract, can create an optional process description with data inputs and process outputs. The response of the DescribeProcess request is constructed from the information in the description file.	Each service must have a zcfg configuration file, which contains some metadata: title, abstract, process version, service type and information on the input and output variables, to name a few. The response of the DescribeProcess request is constructed from the information in the configuration file.
8. Data transport mechanism	A literal can be sent via the URL as a value or pointed to using an URL within the request URL.	mimeTypes (multipurpose internet mail extension) are used, especially the JSON type for complex data.
9. The flexibility of the parameters	Parameters can be set to be mandatory or optional, and multiple occurrences of the same variable are possible.	Parameters can be set to be mandatory or optional, and multiple occurrences are possible.
10. Ease of extending the framework	The source code of the entire project is available in a revision control system. Additional tutorials are available on how to extend the framework, making this easy for a novice to understand.	The source code of the entire project is available in a revision control system. There is no tutorial on how to extend the framework.
11. Ease of integration with other GIS applications	Plugins are available for uDig, OpenJUMP and OpenLayers to provide them with the capability of calling a 52° North WPS process. Extensions for WPS4R (an R backend), GRASS, SEXTANTE and ArcGIS connections are available, allowing the implementation of a new process using their functionalities.	Plugins are available for uDig, QGIS and OpenLayers to provide them with the capability of calling a ZOO WPS process. Extensions for GDAL/OGR, GRASS and MapServer connections are available, allowing the implementation of a new process using their functionalities.
12. Support for the orchestration of external OGC services	External WMS, WFS and WPS services can be added to the map in the workflow GUI, however, the WFS and WPS can only be used as input to a WPS and the WMS only provides base maps and additional layers to the output map.	External OGC web services can be called in the ZOO API (JavaScript) with GET and POST requests.

Table 1. Comparison of 52° North and ZOO Project framework according to the criteria specified

6. DISCUSSION OF RESULTS

Our goal was to evaluate the 52° North and ZOO frameworks' capabilities for the orchestration of web services to produce thematic maps. The 52° North framework is an easy to use WPS framework with a graphic modelling tool, outstanding for beginners. The documentation available is excellent for novices, but more expert documentation is required. The ZOO framework provides support for multiple programming languages, which allows the re-use of existing code to create new web services with the ZOO project making it convenient for developers. The documentation is extensive and describes most aspects of the framework in elaborate detail.

The difference in ease of installation between the two frameworks was clear. This could be due to the fact that 52° North is a more mature project: it was started four years prior to the ZOO project. Documentation for both frameworks is available, however the 52° North documentation is outdated and refers to old versions of the software and library dependencies. Both frameworks currently support HTTP GET/POST and the SOAP protocol. 52° North exposes its WPS services as WSDL document. The ZOO project does not support WSDL for the GetCapabilities request, however this is presently an open ticket

classified as a major defect in the Trac project management system used by ZOO.

The 52° North framework provides a GUI workflow modeller, which is easy to use and requires no programming experience. The workflow modeller allows the user to create an orchestrated service from the available web services. WPS services from the framework, as well, external WPS and WFS services can be called, e.g. GeoServer services, and orchestrated in the workflow modeller. The graphic modeller (http://giv-wps.uni-muenster.de:8080/geoserver/ows) has an easy to interpret GUI with minimal options making it user-friendly. Literal data nodes can be created and linked to the specific web service. The modeller provides the choice to execute a single WPS service or service chain. The orchestration API written in Java is also available for uses, but this requires some programming expertise. The ZOO project has no GUI modeller, but a server-side JavaScript API that runs on SpiderMonkey, called ZOO API, for web service orchestration. The ZOO API gives the power of JavaScript to the user allowing them to call any web service via HTTP and use control logic operations. The JavaScript ZOO API provides more functionality to the user than the 52° North workflow modeller, but requires a certain level of JavaScript skill.

The ease of extendibility of the framework is an important consideration for developers; this allows them to customize the framework to suite their requirements. Both frameworks are open source, and thus the source code can be examined and change by the user in accordance with the licence. The evaluated frameworks both have well-structured source code that can be obtained from a revision control system. The community forums provide support for users and developers.

At present the 52° North orchestration API and ZOO API do not comply with any standard, such as BPEL. The 52° North framework developers are busy developing a BPEL workflow modeller to rectify this limitation. The ZOO API uses the well-known JavaScript and thus provides additional functionalities for calling the ZOO WPS services. There are mentions of creating a graphic workflow modeller similar to YAWL (http://www.yawlfoundation.org/), a business process modeller with native data handling for XML, XQuery and XPath.

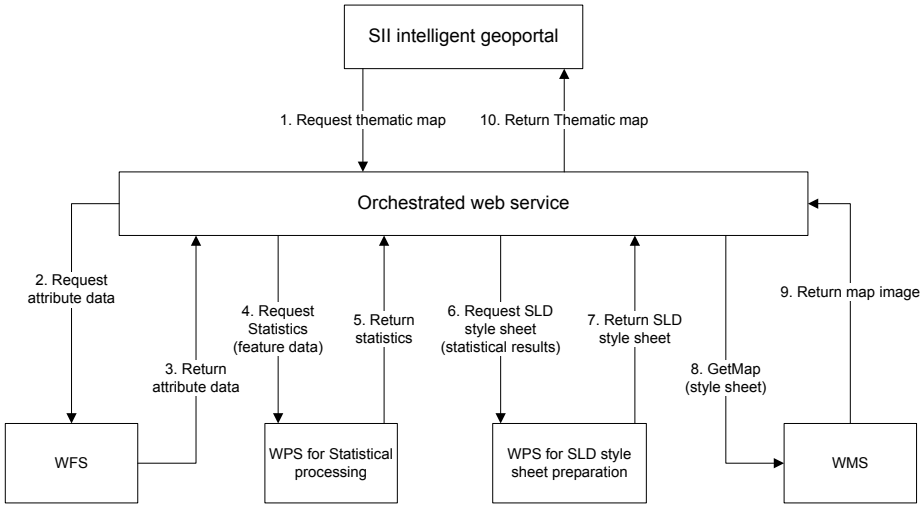

Figure 3. A sequence diagram depicting the orchestration of web services to create thematic maps

A current limitation to automatic orchestration of web services in an intelligent geoportal is the lack of semantic information. Both frameworks are excellent for orchestration of web services in a predefined pattern, but limitations occur when run-time decisions have to be made based on the output of a previous process. At present, the frameworks provide basic semantic information: title, abstract, list of input and output literals, from which the response of the GetCapabilities request is constructed. Additional semantic information is required for the automatic orchestration of the available web services so that the next process can be selected based on the semantic information and ontologies, providing semantic interoperability. Semantic interoperability ensures that the contents of the data and the service are correctly understood when the components are connected (Yue *et al*., 2007).

Figure 3 depicts the orchestration process of web services to produce thematic maps. The orchestration with 52° North could not perform the last step of calling a WMS from GeoServer WMS that produces the resulting thematic map. The 52° North development team are planning to extend the functionalities of the workflow modeller to allow the orchestrated a WMS. The rest of the orchestration process in our experiment could be performed with the current functionalities of the 52° North WPS framework. With the ZOO framework the entire orchestration process to produce thematic maps could be performed. The ZOO API, which relies on JavaScript, provides developers with the ability to call external OGC web services and to use control flow logic operators.

Results of the evaluation show that both frameworks have potential to facilitate orchestration in an intelligent geoportal, but that some functionality is still lacking. The orchestration of web services allows the development of more advanced web services for the preparation, discovery and presentation of information. The results of our evaluation of these frameworks, both with their respective strengths and weaknesses, can guide developers to choose the framework best suitable for their specific needs and requirement.

7. CONCLUSION

In this paper we presented our evaluation of the orchestration capabilities of the 52° North and ZOO project framework for producing thematic maps. In our experiments we showed that

there are limitations and that they differ for each framework. A limitation of both frameworks have is the lack of semantic information needed for automatic orchestration.

The functionality of the framework is very important, but even if these functionalities are excellent, it is important to have up-to-date documentation and an easy to install framework. A cumbersome installation creates a barrier, resulting in only dedicated and technically proficient developers using the framework. This barrier also prevents use of the framework by a larger community. This concern was reinforced by the results of Lopez-Pellicer *et al*. (2011) study, which identified that the top implementers of WPS are Universities. From these results it can be deducted that WPS implementations are still an academic exercise and that the service is not being used by the industry.

In further work we aim to evaluate the performance of the different frameworks, and also compare these to the performance of other workflow software. The development teams of the evaluated frameworks are addressing some of these limitations identified in the paper. We hope our research will contribute to the use and improvement of workflow modellers for the orchestration of web services.

8. REFERENCES

52° North. (2010). 52° North WPS. Retrieved August 16, 2011, from 52° North: http://52north.org/wps

Baranski, B. (2008). Grid Computing Enabled Web Processing Service. *GI Days 2008*. Münster, Germany, June 16-17, 2008.

Fenoy, G., Bozon, N., & Raghavan, V. (2010). ZOO Project: The open WPS platform. *WebMGS 2010, 1st International workshop on pervasive web mapping, geoprocessing and services.* Politecnico di Milano, Como, Italy, August 26-27, 2010.

Fleuren, T., & Muller, P. (2008). BPEL Workflows Combining Standard OGCWeb Services and Grid-enabled OGC Web Services. *Software engineering and advanced applications, SEAA Conference*, Parma, Italy, September 3-5, 2008.

Foerster, T., & Stoter, J. (2009). Establishing an OGC Web Processing Service for generalization processes. *Workshop of the ICA Commission on Map Generalisation and Multiple Representation*. June 25, 2009.

Foerster, T., Lehto, L., Sarjakoski, T., Sarjakoski, L., & Stoter, J. (2010). Map generalization and schema transformation of geospatial data combined in a Web Service context. *Computers, Environment and Urban Systems*, 34, 79–88.

Garnett, J., & Fenoy, G. (2011). WPS Shootout. *FOSS4G*, Denver, United States of America, September 12-16, 2011.

GSDI. (2004). The SDI Cookbook, Edited by Nebert D. January 25, 2004. Retrieved July 20, 2011, from GSDI: http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf

Iwaniak, A., Kaczmarek, I., Kubik, T., Lukowicz, J., Paluszyński, W., Kourie, D., Cooper, A., & Coetzee, S. (2011). An Intelligent Geoportal for Spatial Planning. *25th International Cartographic Conference*, Paris, July 4-8, 2011.

Lopez-Pellicer, F.J., Rentería-Agualimpia, W., Béjar, R., Muro-Medrano, P.R., & Zarazaga-Soria, F.J. (2011). Availability of the OGC geoprocessing standard: March 2011 reality check. *Computers & Geosciences*.

Maguire, D., & Longley, P. (2005). The emergence of geoportals and their role in spatial data infrastructures. *Computers, Environment and Urban Systems*, 29, 3–14.

OGC. (2006). OGC Web Map Server Implementation Specification 1.3.0. Retrieved March 22, 2012, from OGC: http://www.opengeospatial.org/standards/wms

OGC. (2007a). Styled Layer Descriptor profile of the Web Map Service Implementation Specification, Retrieved March 22, 2012, from OGC: http://www.opengeospatial.org/standards/sld

OGC. (2007b). OGC Web Processing Service 1.0.0. Retrieved November 2, 2011, from OGC: http://www.opengeospatial.org/standards/wps

OGC. (2010). OpenGIS Web Feature Service 2.0 Interface Standard, Retrieved March 22, 2012, from OGC: http://www.opengeospatial.org/standards/wfs

OGC. (2011). Implementation Statistics. Retrieved December 21, 2011, from OGC: http://www.opengeospatial.org/resource/products/stats

Slocum, T., McMaster, R., Kessler, F., & Howard, H. (2009). *Thematic Cartography and Geovisualization*, 3rd Edition, Prentice Hall, Upper Saddle River.

Stollberg, B., & Zipf, A. (2007). *OGC Web Processing Service Interface for Web Service Orchestration*, In: Lecture Notes in Computer Science, Vol. 4857, pp. 239- 251, Springer Berlin and Heidelberg.

Suazo, N.C., & Aguirre, J.O.O. (2005). Aspect-oriented web services orchestration. *2nd International Conference on Electrical and Electronics Engineering and XI Conference on Electrical Engineering*. Mexico City, Mexico. September 7-9, 2005.

Sun, J., Liu, Y., Dong, J.S., Pu, G., & Tan, T.H. (2010). Model-based methods for linking web service choreography and orchestration. *Asia Pacific Software Engineering Conference*. Sydney, Australia, November 30 - December 3, 2010.

Weerawarana, S., Curbera, F., Leymann, F., Storey, T., & Ferguson, D. F. (2005). *Web Services Platform Architecture*. Prentice Hall.

Yue, P., Di, L., Yang, W., Yu, G., & Zhao, P. (2007). Semantics-based automatic composition of geospatial Web service chains. *Computers & Geosciences*, 33, 649–665.

ZOO Project. (2011). ZOO Project, Open WPS platform. Retrieved November 2, 2011, from ZOO Project: http://www.zoo-project.org/

9. ACKNOWLEDGEMENTS

This research has been partially supported by the South Africa/Poland Agreement on Cooperation in Science and Technology fort the Joint Research Grant, "Volunteered Geographical Information (VGI) for Spatial Data Infrastructures (SDIs) and Geoportals".