Integration of GRASS Functionality in Web based SDI Service Chains

Johannes Brauner[1], Bastian Schaeffer[2]

[1]Professorship of Geoinformation Systems, Technische Universität Dresden, Germany, johannes.brauner@tu-dresden.de
[2]Institute for Geoinformatics, University of Münster, Germany, schaeffer@uni-muenster.de

Abstract

Bringing the full set of GIS (Geographical Information System) functionality into a SOA (Service-oriented Architecture) based environment is one of the substantial changes in the GIS domain. Standardized data access and coordinate transformation and visualization functionality are already available and commonly used in Spatial Data Infrastructures (SDI). Nevertheless, one problem still remains: SDIs do not offer real geoprocessing capabilities (as offered in classic desktop GIS) such as generalization or complex raster based operations. Therefore, the OGC (Open Geospatial Consortium) released the Web Processing Service (WPS) specification, which describes a standardized abstract interface to bring complex processing power to SOA-based GIS. Despite this existing specification, hardly any concrete implementations of these abstract concepts can be found, which offers a sufficient set of GIS functionality. Thus, it is promising to enhance a classic (desktop) GIS by a WPS interface to offer such existing GIS functionality in a SOA-based environment. This was the starting point to develop an approach for wrapping GIS functionality as a standardized service. This paper shows, that a semi-automatic approach to generate appropriate WPS interface descriptions for each GIS functionality is possible. However, the complex nature of spatial data often requires multiple processing steps. The creation of service chains addresses this aspect. This paper describes a BPEL-based approach to orchestrate geoprocessing workflows based on distributed processing services incorporating the wrapping approach. Finally, the developed concept is proven by applying it to a real world scenario, which monitors droughts in a Sub-Sahara region with a Normalized Differenced Vegetation Index (NDVI) calculated from earth observation data. The GRASS-enabled WPS has been developed by the Geoprocessing Community (www.52north.org/wps) at the 52°North initiative.

1. Introduction

Initiated by increasing network capacity and processing power, some efforts were made to integrate stand-alone geoprocessing applications and their expert functionality into a Service-Orientend Architecture (SOA) environment and therefore enable Web Services to execute geoprocessing tasks (Friis-Christensen et al., 2007; Kiehle et al., 2007). Since standardization and interoperability are key aspects in Spatial Data Infrastructures (SDI) (Groot & McLaughlin, 2000), open standards, such as developed by the Open Geospatial Consortium (OGC), create a solid foundation for integrating processing services in existing distributed environments and especially service chains.

Proceedings of the academic track of the 2008 Free and Open Source Software for Geospatial (FOSS4G) Conference, incorporating the GISSA 2008 Conference
29 September - 3 October 2008, Cape Town, South Africa
ISBN 978-0-620-42117-1

The OGC Web Processing Service (WPS) (OGC, 2007) implementation specification, which became an official standard in mid 2007, is a major attempt to address this issue in a standardized way. The WPS specification defines a standardized service interface to publish and perform geospatial processes over the web. Such a process can range from a simple geometric calculation (for example a simple intersect operation) to a complex simulation process (for example a global climate change model). However, in highly specialized environments such as the GIS-domain, not every processing step can be potentially handled by a single entity due to the intrinsic complexity of spatial data and the complexity of the problem to be processed. Thus, to automate whole business processes and solve complex geoprocessing problems, orchestrated geoprocessing workflows have to be built based on distributed services (Kaye, 2003).

Today, only a few and in most cases very specialized implementations of WPS-enabled geoprocessing functionality do exist (e.g. Foerster & Stoter, 2006; Ragia et al., 2006). To follow ideas of sustainability and in order to avoid redundant implementations, existing monolithic applications could be wrapped with a SOA interface (Papazoglou & van den Heuvel, 2007; Erl, 2005). In GIS terms, a canonical approach would be the exposition of monolithic GIS functionality and algorithms as interoperable WPS processes. GRASS, as a classic monolithic GIS and standard desktop application in the open-source GIS community, was chosen because of its open and very flexible interfaces and the huge amount of available and very mature functionality (Neteler & Mitasova, 2007). This paper proposes a concept and an implementation for such a GIS wrapper. GRASS is wrapped with the 52°North WPS. All used and developed software is released under an Open Source license.

The remainder of the paper is structured as follows. Section 2 introduces the related concepts and technologies. Based on this, Section 3 elaborates on our approach to integrate desktop GIS functionality in SDI service chains. The presented approach is validated by means of a real world scenario for drought monitoring in Africa. This involves the modeling and calculation of a geoprocessing workflow which applies the Normalized Differenced Vegetation Index (NDVI).

2. Background

This section describes the key concepts utilized in this study. First, the idea and technical background of a Web Processing Service (WPS) will be introduced. The next subsection provides a brief overview of concepts and technologies for composing and orchestrating SDI service chains

2.1 Web Processing Service Specification

The WPS specification (OGC, 2007) offers standardized interfaces to access calculations or calculation models which operate on spatial data. The specification does not define which specific kinds of processes are offered by a concrete WPS implementation, only that geoprocessing is offered and how to access it. The following three operations are provided (depicted in the order of a typical course of invocation):

- **GetCapabilities:** As defined in OWS Common Specification (OGC, 2007b) the WPS shall offer the GetCapabilities operation, which delivers general service metadata like offered operations and service maintenance information. The GetCapabilities operation is implicit to all other OGC service specifications. Additionally, in the case of WPS, GetCapabilities delivers short descriptions and identifiers for the offered geoprocessing algorithms.
- **DescribeProcess:** The *DescribeProcess* operation is invoked to gather complete process descriptions for one or more processes, including input and output parameters, data formats, data types, encoding information and other process meta information. Input and output parameters can either be optional or mandatory, limited or unlimited in their occurrence and support one of the three data structures: ComplexData for the spatial data itself (eg. GML data or binary coverage), literal data (e.g. a simple String) or BoundingBoxData (including its coordinate reference system). *DescribeProcess* response documents are frames for the Execute request.
- **Execute:** To finally access the processing functionality, the *Execute* operation is requested for a specific process. An *Execute* request contains, according to the DescribeProcess document, values for each input and output parameter, where involved spatial data can either be delivered in the request itself ("by request") or by including a URN which points to a web accessible location ("as reference").

Although process results can be stored on the server side for future usage ("storeSupported") and asynchronous communication is possible ("statusSupported"), the WPS basically remains a stateless service as required by the OWS Common specification (OGC, 2007b). As any OGC Web Service, the WPS communicates through HTTP-GET and HTTP-POST using a messaging concept based on an OGC-specific XML-encoding. Additionally, SOAP interfaces and interface descriptions are rudimentary specified but in fact not yet ready-to-use.

2.2 SDI Service Chains

With the advent of service-oriented architectures and advancements in the standardization of SDIs and especially OGC Web Services, it becomes feasible to compose those Web Services to service chains (Granell et al., 2005). Alameh (2003) described three basic patterns for this purpose. In the mainstream IT-world, the “Workflow-Managed Service Chaining with Mediating Services” has prevailed and was manifested by Web Service Orchestration (WSO). This term describes the interaction of loosely coupled Web Services in a workflow by specifying the communication on the message level (Peltz, 2003). The description includes the business-logic and execution order and can be thereby distinguished from Web Service Choreography, which is more focused on the public message exchange between multiple parties.

The Business Process Execution Language (BPEL) is an OASIS standard (OASIS, 2007) and is widely used in the mainstream IT-world (van der Aalst et al., 2005). It allows the description of business processes with elementary activities implemented as Web Services using a XML-encoding. BPEL scripts specify the roles involved in the message exchange, supported port types and orchestration information of a process composed of distributed Web Services. On top of required WSDL (Web Service Description Language) descriptions for each participating Web Service, specific binding and deployment issues remain at the Web Service and not at the BPEL process itself. Therefore, BPEL seems to be the ideal candidate to represent SDI service chains as workflows in SOA.

3. Concept

This section presents our selected approach in detail. At first, the concept for wrapping GRASS desktop GIS functionality as WPS processes is presented. Based on this concept, the integration of wrapped GRASS GIS functionality in SDI service chains is introduced.

3.1 Wrapping GRASS GIS Functionality for SDI Usage

To wrap GIS functionality successfully, four requirements can be identified: A programmable interface, multitasking capabilities, machine-readable interface descriptions and support of Coordinate Reference Systems (CRS).

Only one of them is mandatory: The ability to invoke GIS functionality by a programmable interface (e. g. shell scripts or interfaces for programming languages or another web accessible interface). By regarding service-oriented architecture as a technical model for SDIs, machine-to-machine communication and automatic service invocation have to be possible. GRASS offers shell scripting to accomplish this task.

Optionally, the WPS implementation and the wrapped GIS offer multitasking capabilities for parallel processing of incoming requests. This characteristic is not mandatory, because the specification allows requests to be rejected ("server busy") or the WPS implementation can queue requests for a sequential job processing. Due to the modular architecture of GRASS, multiple running instances and parallel request processing are possible.

For the automatic creation of process descriptions (DescribeProcess documents) it is mandatory, that the GIS package provides machine-readable and therefore structured descriptions for each of its functionality (for details see Section 3.2). Certainly, the descriptions can be created manually, but to save time, such GIS process descriptions are highly eligible. Fortunately, GRASS offers such process descriptions by invoking the desired functionality with the "--interface-description" parameter.

On a technical level, it is not relevant if the GIS supports coordinate reference systems (CRS). Therefore, this is an optional characteristic. For accurate and semantic correct results, it is highly recommended and mandatory that it supports CRS usage. GRASS includes both, unreferenced XY workspaces and CRS-enabled ones.

Additionally, the architectural gap of stateless Web Services such as a WPS and stateful desktop GIS has to be solved. On the one hand, classic GIS packages are modeled statefully. Data is imported once, can be processed, reused for other purposes and the results can be stored for later retrieval or ongoing work. On the other hand, the stateless architecture of Web Services does not permit to store intermediate results for ongoing work. Every request has to be responded by a concrete result without any further interactions. Hence, every bit of required information about data types, formats and algorithm parameters has to be included in the request in advance to processing kick-off. Otherwise, interoperability, as proposed by Erl (2005) for SOA, is not ensured.

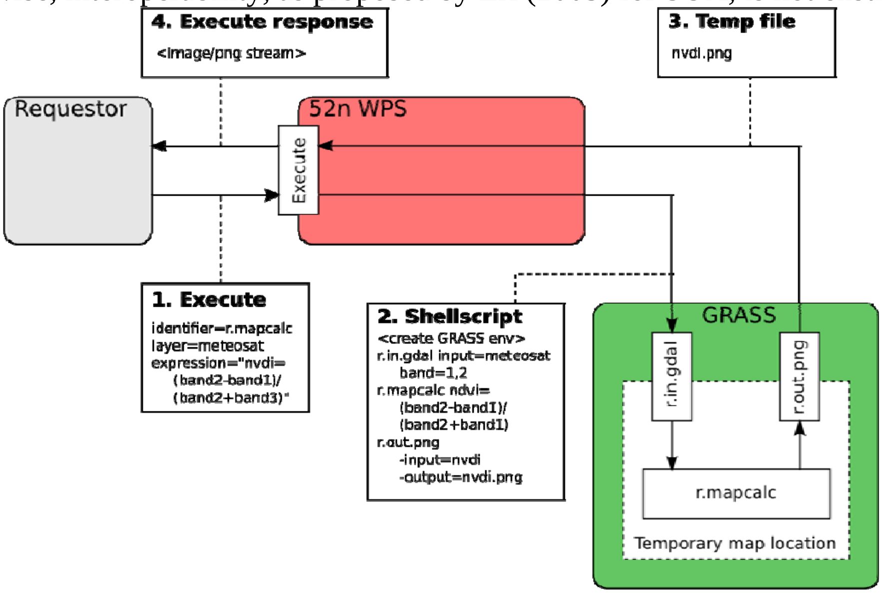

Figure 1. Proposed architecture for wrapping GRASS GIS r.mapcalc functionality. Execute request is received (1.) and an appropriate Shellscript is generated (2.) to invoke GRASS workspace creation, import, process and export functionality. Afterwards the result is handed over to the WPS (3.), which is responsible for sending it back to the requestor (4.).

These observations lead to the following architecture for wrapping GIS functionality by a WPS interface (see Figure 1). To overcome the limitations of stateless modeling the whole typical workspace concept (create workspace, import data, process data, export data) of classic GIS has to be repeated for each and every WPS request. Normally, GIS operate on their own data format, so it is necessary to convert input data to the GIS data format and export it after processing. Technically, at least four pieces of GIS functionality (create workspace, import, process, export) are necessary to create results for each requested process. Therefore, the WPS wrapper has to automatically invoke the required import and export functionality. In our approach, this issue is solved by Linux shell scripts, which invoke GRASS. It is important to mention, that the WPS interface itself remains unchanged and therefore interoperability is ensured.

3.2 Creating Process Descriptions Automatically

As stated above, it is convenient to automatically create process descriptions, if multiple or all GIS functionality shall be offered through GIS wrapping. Only if the GIS offers machine-readable process descriptions, such as GRASS does it, an automatic approach is applicable. In the case of GRASS, specifc XML descriptions are available, which can be converted to WPS process descriptions. For this task a XSLT (Exentsible Stylesheet Transformation) Filter was developed. It allows to automatically map GRASS input parameters to appropriate WPS parameters.

However, in case of GRASS, two semantic problems arise. First, not all classic desktop GIS functionality is suitable in an SDI environment as for instance visualization is part of the client side. Therefore, such functionality is not needed on the server side. Additionally, all functionality concerning administrative tasks for workspace creation and maintenance do not have to be accessible through a web interface, because they are implicit to the work flow (import, process, export) and are only invoked internally by the WPS. Finally, the filter is unable to decide which processes are feasible, instead the user has to remove unwanted processes or exclude them from the transformation beforehand.

Secondly, it is assumed, that on the GIS side all spatial data is already imported into the workspace. The XML descriptions reflect this fact by lacking information about data formats and types. All input parameters can describe themselves with a simple String. In a SOA this is intrinsic information, especially for complex spatial data input. The XSLT filter has therefore to decide, which of the given parameters is a complex one (representing e.g. a spatial data layer) or a simple parameter like a buffer distance. In the case of GRASS this is relatively simple, since all input spatial data layers are labeled as input. Only in cases where there is more than one complex input layer (e.g. for an intersection), the filter is unable to detect complex layers automatically. Here, human knowledge is required to modify the generated (but incorrect) DescribeProcess document. But still, this approach is much more convenient, than generating each process description by hand.

3.3 Integrating GRASS-Enabled WPS in SDI Service Chains

As briefly introduced in Section 2.2, SDI service chains, based on loosely coupled Web Services, can be described with BPEL. Since BPEL is the de-facto standard in the mainstream IT world (van der Aalst, 2003) we decided in favor of this executable language especially because of its broad vendor support.

Each Web Service participating in the workflow has to describe its service interface using WSDL. Since the WPS is a Web Service and its interface can be described with WSDL (OGC, 2008), a WPS can be integrated in a BPEL-based SDI service chain. Moreover, the WPS interface has not been altered to delegate the processes to the stateful GRASS desktop GIS as described in Section 3.1. Hence, it can be treated like any other WPS despite of the different state approaches.

However, WSDL descriptions are required and only marginally specified in the OGC WPS specification. To automate the WSDL creation task, WSDL can be dynamically created by using a XSLT script. This script enriches a WPS WSDL skeleton with specific information from each WPS instance via *GetCapabilities* and *DescribeProcess* metadata. In particular, the WSDL employs the defined generic WPS request and response schemata for the message exchange. Additionally, the native HTTP-POST binding (even though a SOAP binding is specified in the specification) is preserved in the skeleton which is allowed by the WSDL 1.1 specification. Dynamic elements are mainly the endpoint URL and identifiers.

In order to create a fully fledged integration, the standardized invocation and reuse of this defined service chains have to be taken into account. Therefore, we applied the WPS-T approach (Schaeffer, 2008), which enables the exposition of e.g. BPEL based service chains as simple WPS processes. This allows the standardized invocation of these service chains like any other WPS process. Additionally, reuse of created complex models can be easily achieved for everyone and therefore the creation of hierarchical service chains is possible with wrapped workflows as composite building blocks.

4. Monitoring Drought in Africa using GRASS-enabled WPS in SDI Service Chains

This section incorporates the introduced approach (Section 3) in the following drought monitoring scenario.

Alarming reports about food shortages and rising prices have increased in the past. Therefore it is important to find ways to meet these demands especially in developing countries. In the Sub Sahara region, the frequent drought periods have led to poor crop yield. By monitoring droughts, governments and aid agencies can better prepare for such situations in advance. To monitor such droughts, remote sensing data collected from satellites is required. In this scenario, a simple model is introduced which allows one to automatically process such satellite data in real-time. Figure 2 shows the abstract workflow (workflow with no concrete input data), which mainly incorporates GRASS-enabled WPS.

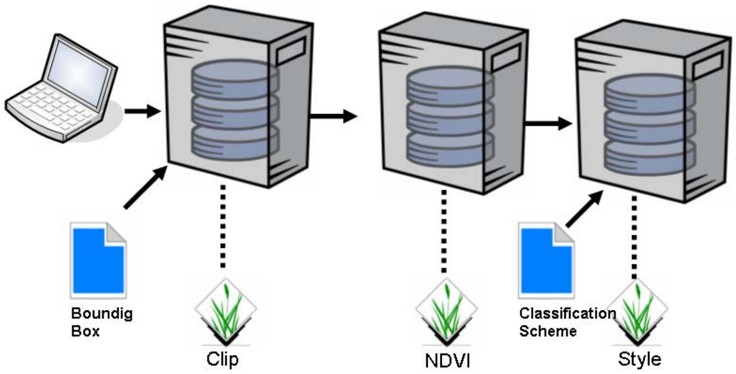

Figure 2. Workflow for monitoring drought using GRASS-enabled WPS.

At first, the incoming remote sensing data is clipped in order to analyze only the cloud free region of interest. The GRASS command r.region produces the clipped region. At second, the main processing step is involved by calculating the Normalized Differenced Vegetation Index data (NDVI; GRASS command r.mapcalc). The NDVI is chosen, because it provides solid proof of live green vegetation based on a simple calculation model. In detail, live green vegetation (chlorophyll) absorbs solar radiation in the visible light range (VIS) from 0.4 to 0.7 nm. The cell structure of the leaves reflects near-infrared light (NIR) from 0.7 to 1.1 nm. This implies, with more leaves the more radiation is reflected and therefore results in the NDVI formula (see [1]).

$$NDVI = \frac{NIR - VIS}{NIR + VIS} \qquad [1]$$

As the last step in this scenario, the NDVI calculation results have to be rendered. Therefore, a classification scheme is required which allows the rendering of the raw NDVI results. As the output of this process, the styled image is returned back to the caller.

The defined abstract workflow is deployed via WPS-T and exposed as a new WPS process (see Section 2.2). Therefore the newly available process can be integrated e.g. as a new building block in another workflow or be executed like any other WPS process. Figure 3 shows the results of the workflow after execution in uDig using the WPS client plugin (Foerster & Schaeffer 2007).

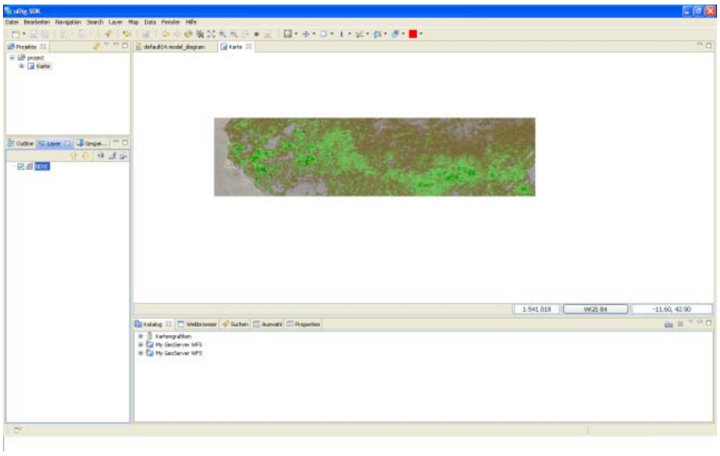

Figure 3. Calculated and colorized NDVI for the Sub-Saharan region based on Meteosat-8 data provided by the ITC, The Netherlands.

5. Conclusion

This paper presents an approach for integrating desktop based GRASS functionality in SDI service chains. Even though the presented scenario (Section 4) shows a running application, performance has to be carefully considered. As stated in Section 3.1, a new GIS workspace has to be created for each request to remain interoperable. The resulting overhead is a major performance bottleneck, because mainly in the case of chained workflows it is not necessary to export and reimport intermediate results. Keens (2006) proposes the usage of the Web Service Resource Framework (WSRF) to avoid redundant data transport for WPS. To save network bandwidth, Bernard et al. (2005) propose to bring the service to the data sources instead of the other way round.

Bucher & Joliviet (2008) state: “It is remarkable in the field of geospatial information, there is not one consensual classification of GIS functions”. Such taxonomies could indeed be helpful to overcome the semantic problems for the automatic generation of DescribeProcess documents (see Section 3.2). Semantic service descriptions, as for instance proposed by Lemmens et al. (2007), the service taxonomies stated in ISO 19119 (OGC, 2002) and the proposed WPS Profiles (OGC 2007) have also to be considered to solve these problems in future research.

However, by not altering the WPS interface, the GRASS-enabled WPS could be easily integrated in SDI service chains. Moreover, the presented approach could be extended to other valuable desktop GIS functionality and bring their capabilities in web based SDI service chains. The automatic creation of WSDL documents enabled to dynamically build orchestrated workflows. The presented approach solved the identified obstacles related to the application of BPEL in combination with OGC Web Services (Stollberg and Zipf, 2008) and therefore eliminated the need for proprietary orchestration languages as introduced by Stollberg. By orchestrating these service chains, we could automate spatial related business processes and therefore increase reliability and cost-effectiveness. Furthermore, it becomes possible to outsource special processing functionality to specialized services and thus allow novel business models by transforming fixed costs into variable costs. Additionally, by allowing the reuse of workflows by wrapping them as simple WPS processes, hierarchical workflows could be created which fully exploit the potential of value-added chains.

The scenario proves our approach. It can be easily extended with additional distributed geoprocessing functionality, such as comparing a 20-years mean NDVI with the lasted available NDVI. Besides, the paper focuses only on WPS as the only service type for service chains. In general, Geoprocessing Workflows are not limited to WPS. Any Web Service regardless of HTTP-GET/POST or SOAP binding can be potentially incorporated in the Geoprocessing Workflow, because the SOA principle is applied here. Especially the integration of Sensor Web Enablement (SWE) with continuous data streams is promising for future research. The data streams would allow the automated real-time processing of data and therefore seem to have a high potential for various domains especially in emergency response and real-time monitoring applications.

References

Alameh, N 2003, 'Chaining Geographic Information Web Services', *IEEE Internet Computing*, vol. 7, no.5, pp. 22-29.

Bernard, L, Craglia, M, Gould, M & Kuhn W 2005, 'Towards an SDI Research Agenda', *Proceedings of the 11th EC GIS & GIS Workshop*, pp. 147-151, Sardinia.

Bucher, B & Joliviet, L 2008, 'Acquiring service oriented descriptions of GI processing software from experts', *Proceedings of the AGILE 2008 Conference*, Girona.

Erl, T 2005, *Service-Oriented Architecture : Concepts, Technology, and Design*, Prentice Hall PTR, Upper Saddle River.

Foerster, T & Stoter, J 2006, 'Establishing an OGC web processing service for generalization processes', *Proceedings of the ICA workshop on Generalization and Multiple Representation*, ICA Comission of Generalization and Multiple Representation, Portland.

Foerster, T & Schaeffer, B 2007, 'A client for distributed geo-processing on the web', in JM Ware & GE Taylor (eds.), *Web and Wireless Geographical Information Systems*, Springer, Berlin, Heidelberg.

Friis-Christensen, A, Ostländer, N, Lutz, M, & Bernard, L 2007. 'Designing Service Architectures for Distributed Geoprocessing: Challenges and Future Directions'. *Transactions in GIS*, vol. 11, no. 6, pp. 799-818.

Granell, C, Gould, M & Ramos, F 2005, 'Service Composition for SDIs: integrated components creation', *Proceedings of the 2nd International Workshop on Geographic Information Management (GIM'05),* Copenhagen.

Groot, R & McLaughlin, J (eds.) 2000, *Geospatial data infrastructure: concepts, cases, and good practice*, Oxford University Press, New York.

Kaye, D 2003, *Loosely coupled : the missing pieces of Web services*, RDS Press, Kentfield.

Keens, S 2006, *OWS-4 WPS IPR: Discussions, findings, and use of WPS in OWS-4*, Techincal Report, Open Geospatial Consortium, Wayland, Document Number OGC 06-182.

Kiehle, C, Greve, K & Heier, C 2007, 'Requirements for Next Generation Spatial Data Infrastructures-Standardized Web Based Geoprocessing and Web Service Orchestration', *Transactions in GIS*, vol. 11, no. 6, pp. 819-834.

Lemmens, R, de By, R, Gould, M, Wytzisk, A, Granell, C & van Oosterom, P (2007), 'Enhancing Geo-Service Chaining through Deep Service Descriptions', *Transactions in GIS*, vol. 11, no. 6, pp. 849-871.

Neteler, M & Mitasova, H 2007, *Open Source GIS: A GRASS GIS Approach*, Springer, New York.

OASIS 2007, *Business Process Execution Language Version 2.0*, OASIS, Billerica, wsbpel-v2.0-OS.

OGC 2002, *OpenGIS Abstract Specification Topic 12: OpenGIS Service Architecture*, Open Geospatial Consortium, Wayland, Version 4.3, Document Number OGC 02-112.

OGC 2007, *OpenGIS Web Processing Service*, Open Geospatial Consortium, Wayland, Version 1.0.0, Document number OGC 05-007r7.

OGC 2007b. *OGC Web Services Common Specification*. Open Geospatial Consortium, Wayland, Version 1.1.0 with corrigendum 1, Document number OGC 06-121r3.

OGC 2008, *OWS-5 GeoProcessing Workflow Architecture Engineering Report*, Open Geospatial Consortium, Wayland, Document number OGC 07-138r1.

Papazoglou, MP & van den Heuvel, WJ 2007. 'Service oriented architectures: approaches, technologies and research issues', *The VLDB Journal*, vol. 16, no. 3, pp. 389-415.

Peltz, C 2003, 'Web Services Orchestration and Choreography', *IEEE Computer*, vol. 36, no. 8, pp. 46-52.

Ragia, L, El Isbihani, A & Kiehle, C 2006, 'A Web Service for Groundwater Vulnerability', *Proceedings of the 8th Conference on Protection and Restoration of the Environment*, Chania.

Ramsey, P 2006, 'uDig Desktop Application Framework', *Proceedings of Free and Open Source Software for Geoinformatics*, Lausanne.

Schaeffer, B 2008. 'Towards a Transactional WPS (WPS-T)', *Proceedings of the 6th Geoinformation Days*, Münster.

Stollberg, B, Zipf, A 2007, 'OGC Web Processing Service Interface for Web Service Orchestration - Aggregating geo-processing services in a bomb threat scenario', *Proceedings of W2GIS*, Cardiff.

van der Aalst, WMP 2003, 'Don’t go with the flow: Web services composition standards exposed', *IEEE Intelligent Systems*, vol. 18, no. 1, pp. 72-76.

van der Aalst, WMP, Dumas, M, ter Hofstede, AHM, Russell, N, Verbeek, HMW & Wohed, P 2005, 'Life After BPEL', *Proceedings International Workshop on Web Services and Formal Methods (WS-FM)*, Versailles.