

An Open Service Network for Geospatial Data Processing

Laura Díaz¹, Carlos Granell¹, Michael Gould¹, Víctor Olaya²

¹Department of Information Systems, Universitat Jaume I, Castellón, Spain
{laura.diaz, carlos.granell, gould}@uji.es

²Universidad de Extremadura, Plasencia, Spain
volaya@unex.es

Abstract

Geospatial data processing involves operations which historically have been offered by commercial Geographic Information Systems (GIS) as built-in functions for processing vector and raster data. Traditionally these have been closed, black-box solutions, however two emerging trends are changing this situation. On one hand Service Oriented Architectures allow distributed geoprocessing in the form of web services, offering data and processing capabilities in a decentralized way. On the other hand, the rapid growth of Free and Open Source Software (FOSS) projects for a variety of GIS domains opens new doors to hybrid customization. We present a unified solution offering geoprocessing services based on FOSS projects and open specifications. This solution offers several benefits in addressing limitations present in traditional centralized GIS applications and facilitates the rapid development of FOSS geoprocessing services. Furthermore, we increase service interoperability and reusability by exposing geospatial operations as distributed web processing services via standards. We demonstrate the viability of our prototype using the scenario of executing hydrological models by chaining multiple geoprocessing services, most of them implemented reusing existing FOSS projects such as gvSIG and SEXTANTE.

Keywords: Distributed geospatial processing, web processing services, WPS, SEXTANTE, open specifications, FOSS.

1. Introduction

The field of hydroinformatics integrates scientists' knowledge and understanding of water resources with applications and technology to improve decision-making in critical contexts such as runoff prediction in major drainage basins. These modelling, monitoring and prediction applications have special requirements different from those of common geospatial data users, and require advanced tools and information infrastructures for accessing huge volumes of data, and for its later storage, processing, analysis and visualization.

Scientists and experts need to have access to the data, models and tools to be able to perform their tasks efficiently. Traditionally, these professionals have had limited access to these data and have had to find them from different sources, which can be a highly time-consuming task. Furthermore, the processing of geospatial data, in order to extract useful information, has been done locally by experts using multiple desktop GIS applications. However, this paradigm –closed solutions that are locally operated and maintained– creates impediments to analysis and processing

of geospatial data. The first impediment is lack of accessibility: users often do not have access to the necessary geospatial data or they are unaware of where to find available geospatial data. In Europe the improvement in discovery of large-scale geospatial data is being addressed by an EC directive called INSPIRE¹, which is concerned with defining Spatial Data Infrastructures (SDI) at the European member state level. SDI are collaborative, Internet-accessible information systems designed to facilitate geospatial data sharing by harmonizing data specifications and mandating their widest possible accessibility and at the lowest possible cost. The current trend in geospatial processing applications calls for SDI access following the Service Oriented Architecture (SOA) paradigm based on standards-based web services that are effective, simple to use and reusable.

The second impediment concerns the geoprocessing capability itself. Most desktop GIS applications provide internal processing capabilities designed to operate locally on geospatial data. These capabilities are designed and implemented by the software house, and often do not fit the needs of the concrete geospatial processing tasks of certain user communities (Friis-Christensen et al, 2007). This impediment is partially addressed by an increasingly diverse variety of FOSS projects, which permit users to freely choose and mix those software components that best fulfil their requirements. In addition FOSS projects, by the very nature of their licenses, may be modified and accommodated to concrete user needs. Moreover, to provide users with just the minimum functionality to fulfil their concrete needs, the trend is to expose this functionality available under SOA, based on web services that are effective, simple to use and available in an ad hoc manner. In this way, users do not need permanent maintenance of multiple desktop GIS applications, because their geospatial processing tasks now become distributed web services. One of the goals of SOA is to enable interoperability among existing technologies and provide an open and interoperable environment based on reusability and standardized components.

The present paper addresses some of the issues of integrating distributed geospatial processing services, by proposing services design principles and a methodology for simplified deployment, reuse and execution of distributed services based entirely on FOSS projects. We also describe an architecture that supports a distributed processing services infrastructure. The proposed Application has been implemented in the framework of the European Union-funded AWARE project², following the principles of INSPIRE.

The paper is structured as follows. The next section reviews some related FOSS projects for geoprocessing. Section 3 introduces the proposed architecture. In section 4 we address the design of interoperable distributed services using FOSS projects. Section 5 provides a deeper look into the use case implementation. The final section contains our conclusions and future work.

2. Geoprocessing with FOSS

Many open specifications and FOSS projects have been developed over the past 20 years in the field of geoprocessing. Today most of these comply with standards created by Open Geospatial Consortium (OGC), an international membership organization that defines interfaces for geospatial

¹ <http://inspire.jrc.it>

² <http://www.aware-eu.info>

services. An example of a web application based on basic geospatial services is reported by (Moreno-Sanchez et al, 2007). Apart from the basic OGC specifications to access and retrieve geospatial data, the OGC recently released the OGC Web Processing Service (WPS) specification (Schut, 2007), providing interfaces for accessing processing services. WPS allows developers to wrap existing services, algorithms, and computation models as distributed web services.

The wider open source community also has been very active in the context of geospatial processing FOSS projects. For instance, GRASS³ is a GIS free software package for raster and vector data analysis that has been around for more than 20 years. Another example is a more recent development, SEXTANTE⁴ that provides a set of tools for spatial analysis integrated as an organized set of extensions for the open source GIS gvSIG⁵.

Although such tools have been widely validated in desktop environments, they are still underused in distributed and web environments, where validating and exchanging large amounts of data still remains a critical constraint for successful distributed applications. In order to foster the proliferation of distributed geospatial processing services, new open source frameworks, built around the WPS specification are being implemented (Granell et al, 2008), for example PyWPS and 52N WPS frameworks. PyPWS (Cepický and Becchi, 2007) allows developers to make native connections to GRASS commands, encapsulating them as contained processes in a given WPS service. The second, 52N WPS (Foerster and Schäffer, 2007), enables the implementation and deployment of WPS services.

Mixing GRASS commands and PyWPS is at the core of the GeOnAS project (Di et al, 2007). Its basic functions are data discovery, data analysis, processing and visualization via the Web. Our approach takes another perspective: integrate FOSS libraries such as SEXTANTE, Geotools, JFreeChart, etc. with self-implemented functionality and the 52N WPS implementation to create a library of WPS to run hydrological models in a distributed manner. The two research works seem similar in terms of implementation, however in contrast to Di et al (2007), we focus on architectural aspects related with the integration and reuse of geoprocessing services within SDI contexts (Friis-Christensen et al, 2007) and following INSPIRE technical architecture. We encourage the wrapping as geoprocessing services not only of traditional geospatial functionality but also any related functionality (such as graphing/charting) considered to be useful for the scientific community and worth being reutilized in other scenarios.

3. Proposed Architecture

The AWARE Application is a distributed, web-based application for running hydrological models in the realm of the SDI (Masser, 2005) and the INSPIRE vision (INSPIRE, 2007). The AWARE Application architecture follows the INSPIRE technical architecture to establish an open and interoperable architecture based on standard interfaces and reusable components. Its primary goal is to identify the services needed to meet user requirements, and to facilitate the composition and reuse of concrete instances of each kind of services types.

³ <http://grass.itc.it/>

⁴ <http://www.sextantegis.com/>

⁵ <http://www.gvsig.gva.es/>

Figure 1 illustrates the AWARE architecture, which is composed of loosely-coupled layers. The Geoportal layer, the top layer in the architecture, is composed of two layers. The Presentation layer is concerned with user interface and interaction, while the Application layer allows the description and implementation of concrete components, for the business logic, and for service chaining control and service invocation. The Service layer combines a set of service instances grouped in types according the INSPIRE Service types such as discovery, view, download, transformation and processing services. Spatial data sets and metadata reside in the Data layer.

The core of the architecture is the Application layer within the Geoportal layer, responsible for the data processing logic and service integration, and the Service layer, where, among others, the distributed geospatial processing services reside, as will be described in the following section.

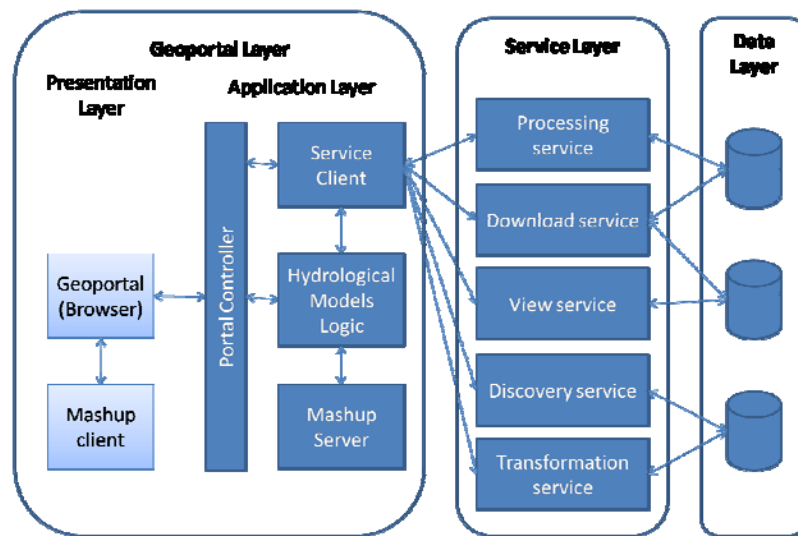


Figure 1. AWARE architecture overview

4. Geospatial Processing Services

This section focuses on the methodology used to expose scientific processes as distributed geospatial processing services and integrate them as compound services implemented with the OGC WPS interface that we have developed in the AWARE project.

4.1 Service Design

A benefit of adapting web service technologies in enterprise environments is that existing and well-tested business components can be migrated to Web environments simply by exposing these components as web service artifacts (Papazoglou, 2008). This approach provides even greater benefits when implemented with FOSS projects, because developers can modify components rather than simply connecting black boxes.

Our service design methodology is based on wrapping. In order to provide useful geospatial processing services which suit the concrete requirements of hydrological models, we have identified atomic functions shared among the model tasks. An atomic or basic process performs an operation domain-independent enough to be applicable to other contexts. One example of basic process is the 2-D area calculation that we can find as part of many complex GIS operations. In this way we implement the area computation as a single process in a given AWARE Processing Service.

Customized or complex processes can be defined as those built on multiple basic geospatial processing services to create more elaborated and customized processes. An example of complex or customized processes is the process for calculating multiple iso-elevation zones given a Digital Elevation Model. This process is basically a chain of other basic processes.

The INSPIRE directive (currently being transposed to national law in EU member states) establishes the legal framework for setting up and operating an Infrastructure for Spatial Information in Europe, and on this base the wrapping of specific functionality as distributed web processing services provides users with tools they need to achieve the required information (Kiehle et al, 2006; Friis-Christensen et al, 2007). Exposing these processes according to standard client-server interfaces and implementing rules of INSPIRE helps to approach the interoperability needed (ORCHESTRA, 2008). Furthermore, on one hand, wrapping atomic and standard functionality aims the processes to be reuse in multiple scenarios, and on the other hand, providing these services with standard interfaces constitutes one ingredient to achieve GIS interoperability in the wider sense (Díaz et al, in press).

Once the processes needed to perform the functionality to run the hydrological models are identified, they are grouped into modules with similar functionality. Each module is designed as a web service implemented as a WPS. Table 1 shows these services within the AWARE application: *Data Conversion* service based in INSPIRE transformation service type. *Topology*, *Sextante* and *SRMIDL* services which belong to AWARE Processing service type and *Chart* service which we have classified as View service type.

Table 1. AWARE WPS

Web Processing Services	SERVICE Type	Service processes
DataConversionWPS	Transformation Service	CoordTransformer, Shp2GML
TopologyWPS	Processing Service	Area, Intersection, Buffer, MaxExtent, SnowPercentage, GetFeatureByAttribute, Thiessen
SextanteWPS	Processing Service	CoordinateElevation, StationsElevation, ElevationCurves, ElevationZones, HypsometricElevation, Reclassify, Vectorize
SRMIDLWPS	Processing Service	SRMIDL
ChartWPS	View Service	DepletionCurvesPlot, DischargePlot, HbvRunoffPlot, HbvSWEPlot, SensorDataChart

4.2 An Implementation Example: Sextante Web Processing Service

In Section 2 we reviewed some FOSS projects devoted to implementing the OGC WPS standard. The 52N WPS framework is the implementation of choice in our case because it offers the chance to expose geoprocessing functionality as OGC- and W3C-compliant services, and because it was well-documented. Figure 2 shows the relationship among processing service type specification (WPS, WSDL/SOAP) and implementation within the Service layer (see Figure 1). It shows the steps carried out to implement multiple *Sextante* WPS from its abstract definition down to the concrete realizations.

52N WPS as an implementation of the OGC WPS standard allows developers to extend a WPS instance adding new processes, the so-called “Algorithms”. The basic operational unit of the OGC WPS is the notion of process –a spatial operation, with inputs and outputs of a defined type. This

means that a WPS instance running at the Service instance layer (e.g. *Sextante* WPS, etc.) may offer one or various operations or processes at the Service operations layer (e.g. Reclassify etc.) just as normal web services do.

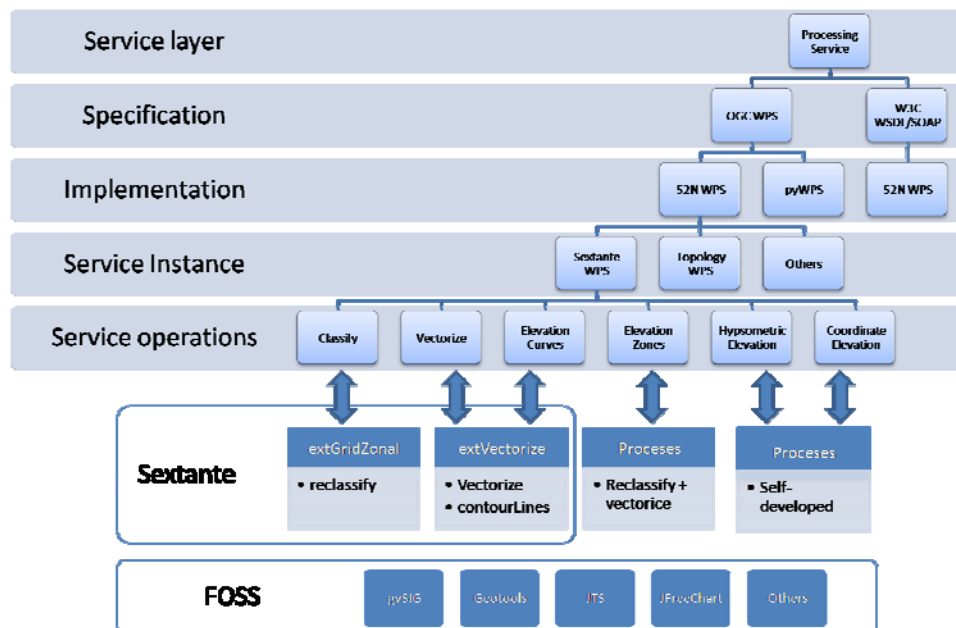


Figure 2. *Sextante* WPS Implementation

To illustrate the implementation of one of these services based on 52N WPS, we describe in this section the *Sextante* WPS in more detail, which actually wraps SEXTANTE functionality as reusable web processing services using OGC WPS interfaces.

SEXTANTE (Olaya 2007) comprises a set of extensions, each of which represents a single analysis process. Each extension is based on a so-called Algorithm/Extension architecture, which formally separates the processing itself from other tasks such as creating the corresponding User Interface (UI) or handling output data, among others. This architecture is particularly suitable for exposing SEXTANTE algorithms through a WPS interface. SEXTANTE algorithms must contain two main methods: *defineCharacteristics()* and *processAlgorithm()*. These two methods are equivalent to WPS operations: *DescribeProcess* and *Execute*.

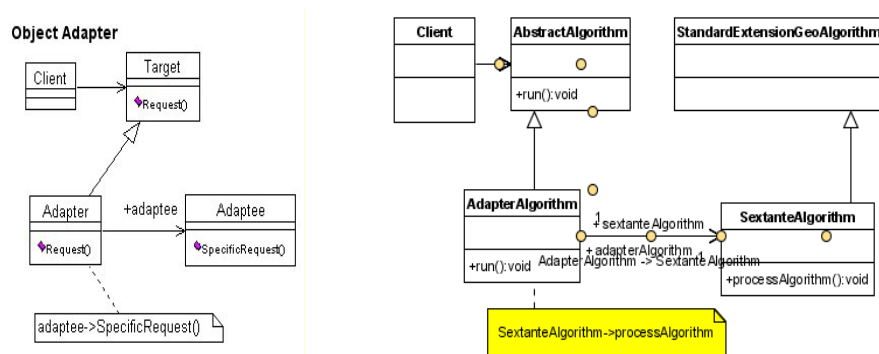


Figure 3. Adapter pattern. The left side depicts generic patterns, while the right side depicts the pattern implementation applied to SEXTANTE algorithms

To integrate SEXTANTE into the 52N WPS framework in a way that allows the easy addition of functionality to SEXTANTE when needed, we have followed the well-known Adapter pattern (Gamma et al, 1995) as illustrated in Figure 3 (left side). The right side of Figure 3 shows the class diagram for the adapter pattern implementation. The abstract class *AbstractAlgorithm* must be extended when adding a new process to a 52N Service. In this way all the algorithms created in 52N to expose SEXTANTE functionality will implement the Adapter pattern, having an instance of the *SextanteAlgorithm* to be reused. When the process (52N algorithm instance) is executed it will call the correct method in the *SextanteAlgorithm* interface called *ProcessAlgorithm*.

5. Use Case

Many datasets are available to provide scientists with a more complete view and context of hydrologic processes, from meteorological data collected from weather stations at different locations to satellite imagery capturing the snow coverage area of interest. As introduced previously, analyzing these disparate datasets together is a tedious task and hence is often performed inefficiently resulting in incomplete and inaccurate results. Our application then facilitates the collection, validation and integration of spatial data for analyzing a given watershed, and provides access only to those specific processes that are needed for any given model, reducing wasted software implementation, maintenance and cost.

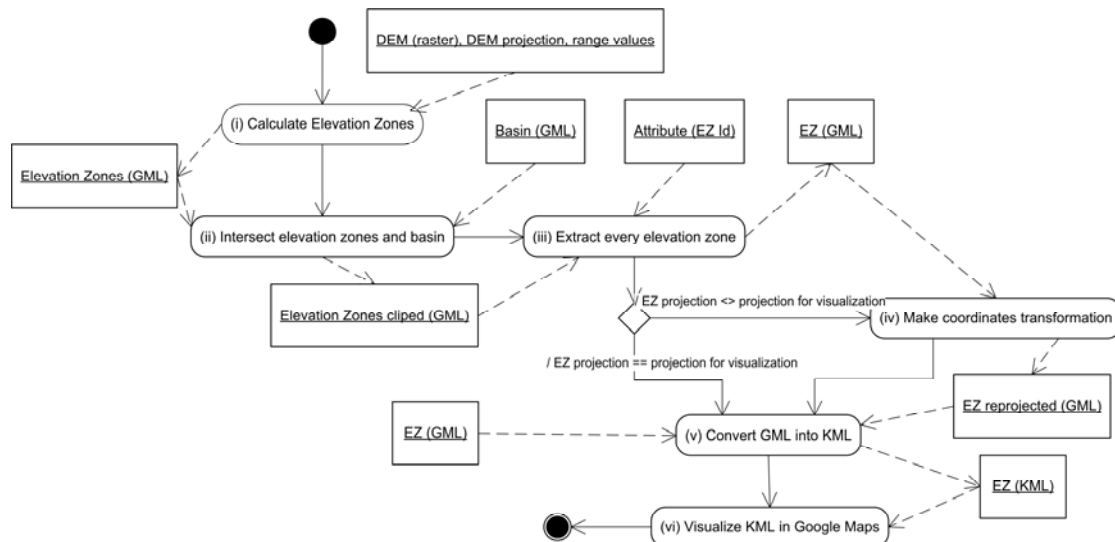


Figure 4. Activity diagram for calculation of elevation zones. *EZ* stands for Elevation Zone

Figure 4 summarizes the major steps contained in one of the tasks within the major hydrological model supported in our application. It is dedicated to the calculation and visualization of elevation zones for a watershed, and allows us to explore the issue of integrating distributed geospatial processing services because it involves multiple input data and processing services. The first step (i) consists of calculating the elevation zones given a Digital Elevation Model (DEM) file, its projection and a range of pairs (min-max) of elevations. It makes a request to the *ElevationZonesCalculation* process, which is a customized process that internally calls two raster analysis processes also available in *Sextante* WPS. The next step (ii) in the workflow permits spatial computations such as the spatial intersection operation, which computes the intersected region

between the elevation zones (polygon collection) and the watershed boundary (polygon) by sending a request to the *Intersect* process in *Topology* WPS and returning the intersected area. Once delimited the elevation zones within the watershed extent, each elevation zone (polygon collection) is extracted from the elevation zones file according to an attribute representing the category (iii) by sending a request to the *Topology* WPS and executing *getFeatureById* process to extract each elevation zone from the main file. For the former, the resulting elevation zone is forward to the Presentation layer where GML data is transformed into KML (Keyhole Markup Language) and then visualized. For the latter, we previously perform a coordinate transformation routine via a spatial processing service in order to transform the coordinates set of the elevation zone in a given source reference system to a target reference system, optimal for visualization. We use GML format for processing tasks but KML for data visualization.

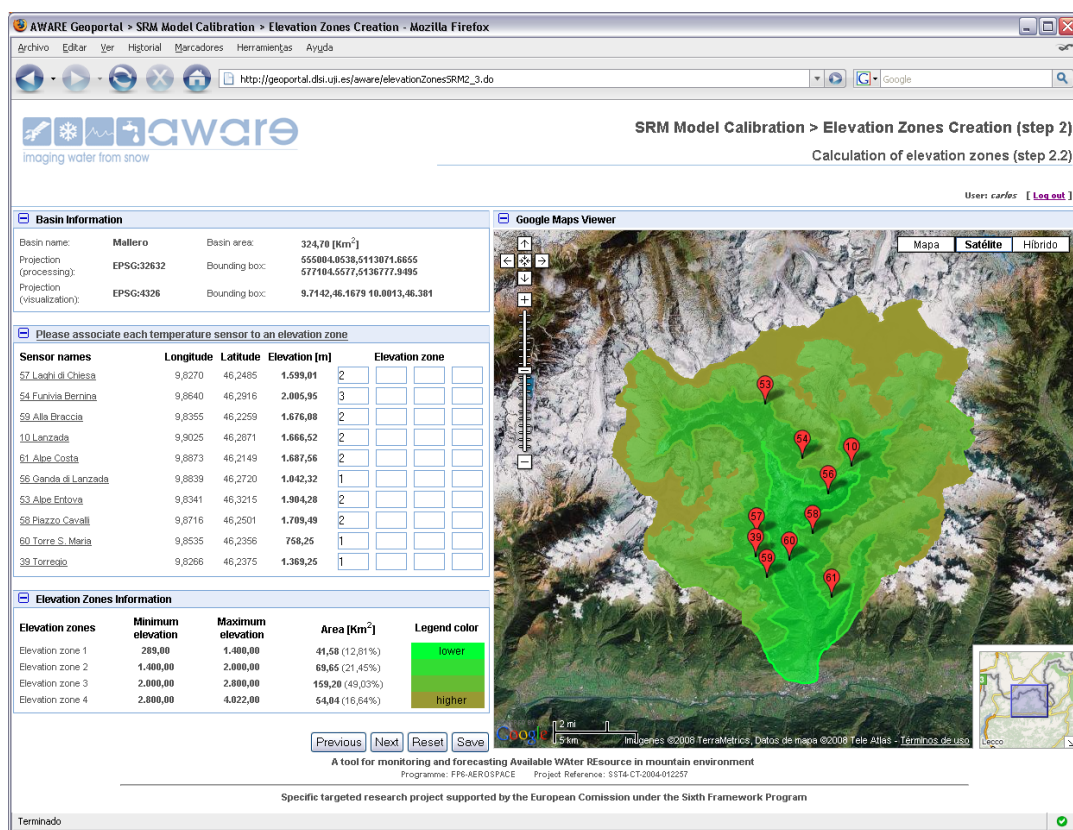


Figure 5. User interface displaying the elevation zones calculated

Figure 5 shows the results of executing the workflow in Figure 4. As regards user interface, the left side displays useful information (basin information, elevation zones information) and web forms for data entry by users, while the right side contains the web mapping viewer to provide graphical feedback to users. In summary, this example has shown how multiple distributed geospatial processing services are invoked and integrated in one workflow step from user's perspective, using mostly backend FOSS projects.

6. Conclusions

A first observation derived from the use case scenario, which coincides with conclusions in previously referenced work, is that the approach based on distributed geoprocessing services leads to a collection of reusable geoprocessing services, available for other users in other scientific domains in the case that they are well-documented and registered in open catalogues. This is possible in principle because WPS-based geoprocessing services do not work with pre-established datasets but rather they preserve a loosely-coupled relationship between ad hoc data and processing capabilities making it possible to chain them to other geospatial web services.

The OGC WPS services have been tested in different contexts (Friis-Christensen et al, 2007; Foerster & Schäffer, 2007; Díaz et al, 2008), illustrating that it is possible to combine several geospatial processing services for accessing, processing and visualizing data within an SDI.

OGC standards seem to be mature enough to provide specifications to create interoperable web services with all the functionality needed to create a distributed application on top of an SDI. We have created a web application by which hydrologists can discover, access, and analyze geospatial data as hydrological models are being run. These tools are not only available for this group of hydrologists, but they are also remotely accessible by anyone accessing this SDI. Using standard interfaces and following the technical architecture defined by INSPIRE we have created these tools as standard web services able to be integrated in other systems that follow INSPIRE guidelines. Further information on these WPS is available at <http://www.geoinfo.uji.es/demos.html>.

References

- Cepický, J. & Becchi, L., 2007, 'Geospatial Processing via Internet on Remote Servers – PyWPS', *OSGeo Journal*, vol 1. Available at <http://www.osgeo.org/journal/volume1>
- Di, L., Zhao, P., Han, W., Wei, Y., Li, X., 2007, 'GeoBrain Web Service-based Online Analysis System (GeOnAS)'. In *Proc. of NASA Earth Science Technology Conference*.
- Díaz, L., Granell, C., & Gould, M. (in press). Spatial Data Integration over the Web. In V.E. Ferragine, J.H. Doorn, L.C. Rivero (Eds.): *Encyclopedia of Database Technologies and Applications*, 2nd edition. Information Science Reference.
- Díaz, L., Granell, C., & Gould, M. (2008). Case Study: Geospatial processing services for web-based hydrological applications. In J.T. Sample, K. Shaw, S. Tu, M. Abdelguerfi (Eds.): *Geospatial Services and Applications for the Internet*. Springer. ISBN 978-0-387-74673-9.
- Foerster, T. & Schäffer, B., 2007, 'A Client for Distributed Geo-processing on the Web', In *Proc. of W2GIS, Cardiff (UK)*. Berlin: Springer LNCS 4857: 252-263.
- Friis-Christensen, A., Lutz, M., Ostländer, N., Bernard, L., 2007, 'Designing Service Architectures for Distributed Geoprocessing: Challenges and Future Directions', *Transactions in GIS* 11(6): 799-818
- Gamma, E., Helm R., Johnson, R., Vlissides, J., 1995, *Design Patterns*. Addison-Wesley
- Granell, C., Díaz, L. & Gould, M., 2008, *Distributed Geospatial Processing Services*. In M. Khosrow-Pour (Ed.): *Encyclopedia of Information Science and Technology*, 2nd edition. Information Science Reference.

- INSPIRE EU Directive, (2007) Document available at
http://eur-lex.europa.eu/LexUriServ/site/en/oj/2007/l_108/l_10820070425en00010014.pdf
- Kiehle, C., Greve, K., and Heier, C., 2006. Standardized Geoprocessing - Taking Spatial Data Infrastructures one Step Further. In Proc. of the 9th AGILE Conference on Geographic Information Science, Visegrád (Hungary).
- Masser, I. (2005) GIS Worlds: Creating Spatial Data Infrastructures. Redlands: ESRI Press
- Moreno-Sanchez, R., Geoffrey, A., Cruz, J. & Hayden, M., 2007, 'The potential for the use of Open Source Software and Open Specifications in creating Web-based cross-border health spatial information systems'. Intl. Journal of GIS 21(10): 1135-1163.
- Olaya, V., 2007, SEXTANTE: a gvSIG-based platform for geographical analysis, FOSS4G 2007, accessed at
http://www.foss4g2007.org/presentations/view.php?abstract_id=123
- ORCHESTRA Consortium (2008) Orchestra, an open service architecture for risk management.
<http://www.eu-orchestra.org/docs/ORCHESTRA-Book.pdf>
- Papazoglou, M.P., 2008, Web Services: Principles & Technology. Pearson, Essex
- Schut, P. (ed), 2007, 'OpenGIS Web Processing Service 1.0.0'. OGC.
<http://www.opengeospatial.org/standards/wps>