Case Study: Geospatial Processing Services for Web-based Hydrological Applications

Laura Díaz, Carlos Granell, Michael Gould

Centre for Interactive Visualization, Department of Information Systems, Universitat Jaume I, Castellón, Spain

Abstract River discharge is an important issue to be monitored because of its significant influence on environmental systems, on human lives for water resource exploitation, and hazards related to floods and landslides. In this context, we have designed and developed a web-based Geoportal for hydrological applications that integrates geospatial processing services and web mapping viewers to provide an interactive and user-friendly interface to hydrological modelling experts and scientists. The server side provides hydrological model logic through a library of distributed geospatial processing services that prepares and presents all geospatial data –satellite imagery, cartography, digital elevation models, and sensor measurements– necessary for running the hydrological (river runoff) model .The Geoportal's client side facilitates catalogue service searching for appropriate geospatial data, interacts with the geoprocessing services according to the hydrological model parameters, and displays the results into a web mapping viewer by using the Google Map API to provide quick feedback to scientists about the status and behaviour of the hydrological model. This chapter provides an overview of the proposed Geoportal by integrating standards both for geospatial processing services and for geospatial data visualization. We emphasize the challenges and problems encountered during implementation regarding the interoperability of different geospatial standards and components.

1 Introduction

Geospatial web services –web services that serve and process geospatial information for a wide range of territory-based applications– have evolved to become interoperable pieces used to build modular and distributed GIS applications over Internet [18]. They have become key components of Spatial Data Infrastructures (SDI) [15], helping support some the most common requirements of scientific users of information systems such as discovery, access, process and visualization of geospatial datasets.

Scientists traditionally have been major consumers and collectors of huge amounts of data. As in the case of the field of hydrology, they have different needs than geospatial information users from the more cartographic side of the Geographic Information Systems (GIS) field. However they have in common the need to gain remote access (without huge downloads) to large quantities of data, and also to process them remotely using on-line services. Following the e-Science philosophy of connecting scientific research (science) [6], our aim here is to connect scientists to their data, resources, models, and services what is also an important challenge for SDI. Indeed such infrastructures are beginning to facilitate access to distributed, heterogeneous geospatial data through a set of policies, common rules, and standards that together help improve interoperability [17]. Traditional discovery and visualization-based SDI is evolving to a more service-based vision in which geospatial web services are used not only to access geospatial data, but also to transform them and process them, often in geospatial service chains [1][12].

The recently-published INSPIRE Directive[1] aims to harmonize spatial information across Europe and to improve geospatial data services according to common principles. Our goal here is to help hydrologists –and scientists in general interested in geospatial information– in approaching to INSPIRE philosophy to more efficiently meet their requirements. Hydrologists provide their knowledge and expertise, and SDI researchers play an active role to provide proper geospatial processing services, components and applications to facilitate connecting such hydrologists with their data and models within an interoperable architecture. In this way, the cooperation between data providers and users is fostered as proposed by the Global Monitoring for Environment and Security (GMES) initiative[2].

This chapter presents the design and implementation of a concrete web-based Geoportal for hydrological applications that integrates geospatial processing services and web mapping viewers like Google Maps to provide an interactive and user-friendly interface to run hydrological models. Our experience in distributing data and processing in the field of hydrology may be extrapolated to other specialized application fields like flooding, forest fires, urban and geological modelling, making then the SDI and the INSPIRE initiatives available to a broader audience.

The remainder of this chapter is structured as follows. Section 2 presents an introduction to hydrological models. Section 3 gives a brief state of the art of the geospatial services and applications for hydrological models. The description of the Geoportal application is the subject of the Section 4 that reviews the system architecture and the underlying open source components and technologies used for implementation. Section 5 describes the set of geospatial processing services and how are integrated into the Geoportal. Some lessons learnt with respect to user interface issues and the integration and interoperability of geospatial processing services together with the some conclusions are discussed in Section 6.

[1] http://inspire.jrc.it/

[2] http://www.gmes.info/

2 Hydrological Models

River discharge is an important issue to be monitored because of its significant influence on environmental systems, on human lives for water resource exploitation, and hazards related to floods and landslides. In this sense, hydrological models have gained more attention because they provide a physical based representation of the hydrological processes occurring in a given basin. The application of these models together with current technologies make possible to monitor and forecast river discharge in a better way. The first task for supporting such monitoring applications is to identify the people who would use the software. Often data providers, who own the data necessary, and scientists, experts in hydrological models, differ in goals and objectives, leading to a lack of collaboration among them.

The AWARE project[3] is a multidisciplinary project carried out by a team of hydrologists, remote sensing specialists, and information system researchers [19]. Its aim is to put together data providers, experts and scientists by developing a user-friendly web-based prototype that permits not only expert users (hydrologists and other scientists) but also other types of end users and data providers (such as water policy makers, water supply and hydropower companies, irrigation consortia, public authorities) to run concrete hydrological models. This implies usability requirements in our system design leading to an easy-to-use prototype that features intuitive interfaces and wizards assisting non-experts with both the complex tasks of such models and interpreting the results. Since expert users often are more comfortable directly handling and analyzing data, and feel that data provided should be accurately investigated by them, the Geoportal should be flexible enough to serve both experts and novice users. Furthermore, the design of the Geoportal should take into account usability, utility and flexibility requirements.

3 Overview of Available Geospatial Services and Applications for Hydrological Models

Several web-based hydrological applications are publicly available on the Internet. Most of them are built around a web mapping service in which several data sets are visualized by applying transparently hydrological model routines. However, it is important to highlight some general differences between these web solutions and our Geoportal. Firstly, our solution allows expert users to interact directly with the underlying hydrological model. A hydrological model normally involves heterogeneous datasets but also several model parameters and variables that must be calibrated. In our application expert users may try several model configurations until the results are acceptable for them. Secondly, in contrast to other applications

[3] http://www.aware-eu.info/

that use static datasets, expert users should load specific datasets of interest for the area of study. Actually many expert users own the local data necessary to run the model and so the first choice is to allow users to feed the model with the local data they possess. Since a goal of the AWARE project is to be compliant with the INSPIRE initiative, the Geoportal allows other non-expert users to discover and access geospatial data via SDI catalogue services [17]. For instance any user might be interested in searching catalogues for appropriate satellite imagery for the study basin (geographic constraint) and during the snowmelt station (temporal constraint). Finally, another key aspect is that our application is built on distributed geospatial processing services as we detail in the following sections. This aspect meets nicely with the term service-oriented science [6], which refers to scientific research structured as distributed networks of interoperating services. Next we sketch some web solutions for hydrological applications to provide an overview of current web-based hydrological applications before describing our Geoportal application in the following sections.

The National Water Information System (NWIS[4]) for the U.S. provides web-based access to hydrological data to the public and organizations. Basically, NWIS is a data distribution site where users can search and visualize static water data (already embedded in the system), making it impossible to load user data different from those stored in the system. The U.S. Geological Survey (USGS) also offers StreamStat[5], a web-based tool that allows users to obtain stream flow statistics, drainage-basin characteristics, and other information for user-selected sites. StreamStats users can choose locations of interest from an interactive map and obtain stream flow information for these locations.

A relevant tool is BASINS[6] –Better Assessment Science Integrating Point & Nonpoint Sources from the U.S. Environmental Protection Agency (EPA)–, which is a complete hydrological application for performing watershed-based studied using hydrological models similar to one used in our Geoportal. This application runs on an open source GIS called MapWindow[7], making it more attractive to open source community, yet it is a desktop application that implies that all functionalities and modelling tools are integrated in the application and then performed locally.

The IJEDI WebCenter for Hydroinformatics [23] is an online application to identify drought-vulnerable regions. The authors propose a combination of data mining techniques to characterize the behaviour of water basins and classify them according to the drought index. Although the goals of IJEDI and our application are slightly different, both deal with multiple kinds of data that have to be integrated and also provide a friendly user interface to be used by non-experts and experts users indistinctly. However, it is important to note that geoprocessing capa-

[4] http://waterdata.usgs.gov/nwis/

[5] http://water.usgs.gov/osw/streamstats/

[6] http://www.epa.gov/waterscience/basins/

[7] http://www.mapwindow.com/

bilities are not present in the IJEDI application in terms of distributed geospatial services just as our application does.

None of the previous applications execute hydrological models by using distributed geospatial processing services. Certain types of applications demand a distributed approach for multiple reasons such as efficiency and reliability. For example, web applications for spatial visualization often fully rely on the server to receive the data and visualize them, however some store data on the client side (cache) to handle them locally and improve the response time of data visualization [3]. In our context expert users also try to process large quantities of data remotely using on-line services rather than downloading the required data and processing them locally [8]. Indeed transferring large amounts of data from servers to clients can slow down the whole process, due to network problems or if the server is heavily loaded. Some efforts to define interfaces to access and process multiple kinds of geospatial data remotely are carried out by the Open Geospatial Consortium (OGC). Some basic interfaces (WMS, WFS, etc.) have been already applied to create web applications [2][4], yet these are shown to be insufficient to suit the specific processing and modelling requirements of hydrological applications. However, the recent OGC Web Processing Service (WPS) specification version 1.0 [21] provides interfaces for interacting with geospatial services by either creating them from scratch or wrapping existing off-line services as web services. In short, WPS offers three methods to provide the functionality of a certain geospatial processing service by first using the *getCapabilities* method, common in other OGC services, in order to know the available service methods. The WPS defines input and output parameters in a very detailed way by providing a *describeProcess* method. Finally, the *execute* method actually invokes the geospatial processing service with concrete input parameters and returns the results. Here then we propose a web application that takes advantage of the distributed processing capabilities for performing hydrological models. Similar approaches using WPS services have been taken in [7][11][24], though with some differences regarding our approach as explained in the next section.

This brief overview shows some relevant web applications for hydrological models pointing out that no web applications using distributed processing are present in the field of hydrology. The next section details the architecture of our Geoportal application that supports distributed geospatial processing services.

4 System Architecture and Software Components of the Geoportal Application

The software components and the system architecture of the Geoportal application are illustrated in Figure 1. The architecture follows a middleware approach composed of three layers. The presentation layer contains the software components used for the user interface (top Figure 1). Different servers and distributed geospatial processing services form the middleware layer (centre Figure 1) whereas geospatial data and database systems take place in the data layer (bottom Figure 1). Here we focus mainly on the components involved in the presentation and middleware layers, describing briefly the data layer. It is important to note that the Geoportal has been built entirely with open source components and technology.

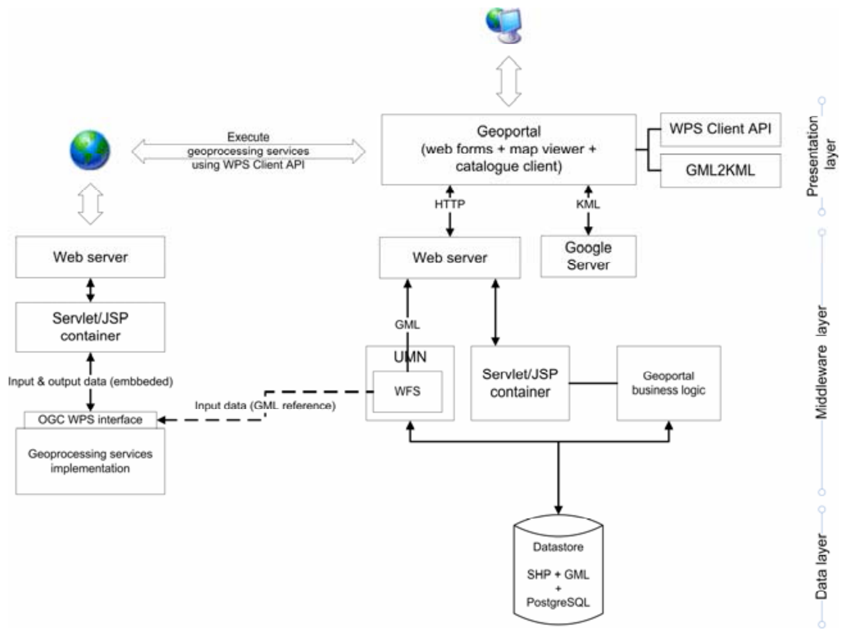

Fig. 1. Geoportal architecture and its components

The presentation layer provides the Geoportal user interface, which permits expert users to select and perform a hydrological model for a concrete basin of study. All users are validated and authenticated when logging in the Geoportal. This feature refers to the possibility to permanently store the current user session in a database. We use for the Geoportal the open source database system PostgreSQL[8] to validate users and to store the result of calibrations successfully completed. A user may perform multiple calibrations for the same basin and decide later which one to choose to forecast the basin discharge (the actual goal of the hydrological model).

The user interface of the Geoportal is composed of multiple web forms (like wizards) implemented using Apache Struts[9], an open source framework for build-

[8] http://www.postgresql.org/

[9] http://struts.apache.org/

ing web applications. Such web forms rely indeed on the Apache HTTP Server[10] and Apache Tomcat[11] to offer the Geoportal business logic, which are basically a set of Java servlets and Java Server Pages (JSP). Both servers are integrated using HTTP connectors, which forward internally Apache HTTP Server requests to Apache Tomcat, enabling then the execution of Java servlets through the Apache Server port.

A combination of Java servlets and JSP form the Geoportal business logic in the middleware. They perform a great range of functions such as guiding users through the web forms to search available data in catalogues, data preparation and collection, data modeling, calibration of model parameters, and interpretation of results. Other common web-based functionalities such as user authentication are also implemented as Java servlets. If some geospatial processing routine is necessary when a web form is filled out, the Geoportal is able to invoke the corresponding geospatial processing service through the WPS Client API component. The WPS Client API is a self-developed, key component, because it enables the communication between the presentation and middleware layers, facilitating connection, access and combination of distributed WPS services. It transforms user requests from the presentation layer into OGC WPS requests addressed to a concrete geospatial processing service. Once results are returned, it transforms responses to be properly delivered to the presentation layer (users). In particular, the WPS Client API is developed in a modular way being easily extended to support next versions of the OGC WPS specification. At this moment it supports connectivity to OGC WPS version 0.4 because version 1.0 is, at time of writing, still pending for approval. Like in the OGC WPS specification, it provides support for the data types specified by OGC like simple data types and GML (version 2.x). Also, it provides a simple caching method to just request the *getCapabilities* and *describeProcess* methods of all available processes of a given WPS service. This increases the processing speed when multiple processes of the same WPS are invoked simultaneously.

Another key component embedded in the user interface is a web mapping client or map viewer [16]. In our case we use Google Maps API [10] because it provides a friendly, interactive user interface for novice users. It offers good performance for rendering spatial data, and already incorporates high resolution satellite imagery and other interesting spatial data (for example road network layer in hybrid and map views), very useful for hydrological applications.

---

[10] http://httpd.apache.org/

[11] http://tomcat.apache.org/

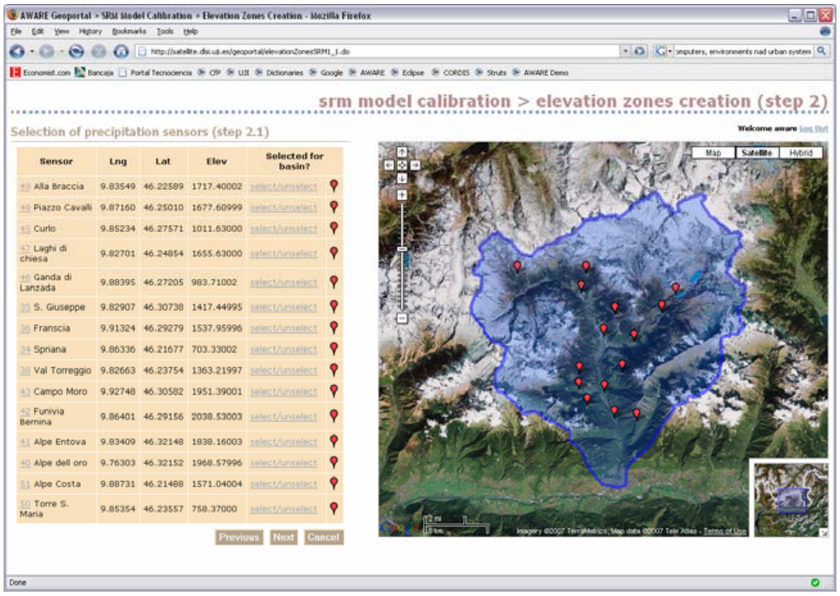

Fig. 2. Geoportal interface for basin boundary and network of precipitation stations

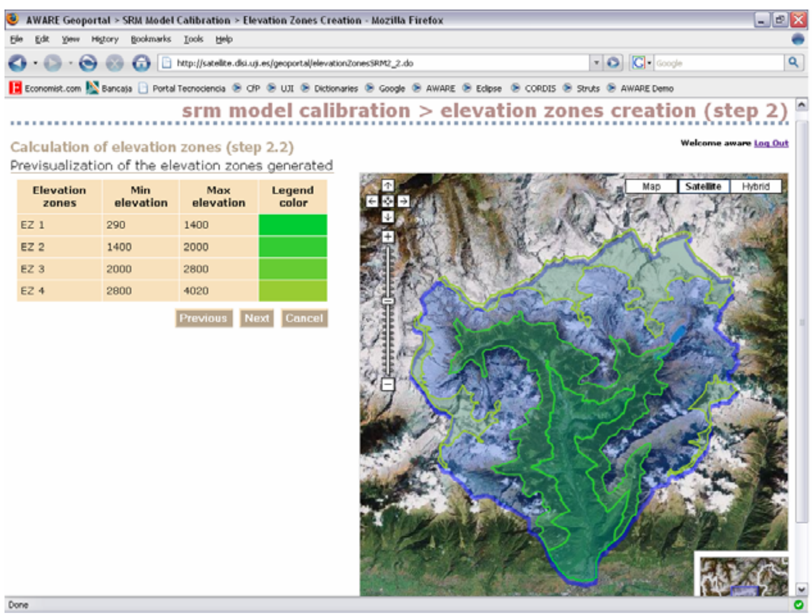

Fig. 3. Geoportal interface for basin boundary and elevation zones

Figures 2 and 3 show the Geoportal user interface in different steps for the calibration phase of one of the hydrological models. Both Figures refer to the Mallero river basin (319 km2), in the Italian Alps, which is one of the AWARE project test basins. In Figure 3, users can check the distribution of precipitation sensors according to the basin boundary. The term *mashup* is currently gaining momentum to integrate heterogeneous web data [9][13] and it is also crucial for our application. Google Maps is a key to our service mashup because it combines transparently both remote data like the base image and local data like the basin boundary and the network of precipitation sensors. In addition, users may also click on each red pushpin to get more detailed information such as the location, elevation and name of the corresponding precipitation sensor.

The table of precipitation sensors listed in Figure 3 displays a calculated column (fourth) as a result of invoking a WPS service. As mentioned previously, the middleware layer consists of a library of geospatial processing services. At this point, it is necessary to describe how information flows when executing such WPS services. Suppose the example of calculating the elevation given a sensor location (a pair of coordinates) and a DEM (Digital Elevation Model) file for the basin. When the geospatial processing service is required, the application interacts with this service via the WPS Client API that builds appropriate XML-based queries according to the method invoked. Once reported details of the input and output parameters of that process, the WPS Client API has two possibilities (see Figure 1) to built the *execute* query: either input data can be embedded in the query itself or can be passed by reference specifying a valid URL to remotely fetch such data. So, sensor location values are embedded in the very XML request while the DEM file is referenced by indicating its URL. The same is applicable to (huge) GML data when are used in a service. In this case, GML data can also come from querying a WFS as occurs in our Geoportal (see Figure 1). Both tasks of embedding GML data in WPS *execute* queries and extracting them from WPS *execute* responses are performed by the WPS Client API using XSLT transformations. Finally, when results are forwarded to the presentation layer, the Geoportal transforms the GML data into KML[12] data (Keyhole Markup Language) –a simple XML language tightly connected with Google Earth– to be loaded in Google Maps by using again XSLT transformations (GML2KML component in Figure 1). Furthermore, we use GML format for processing tasks but KML for data visualization.

Figure 3 visualizes the elevation zones generated for the basin. Details of the tasks involved in the hydrological model are out of the scope of this chapter, yet notice that huge amounts of KML data are rendered both with good performance and transparency. This example will be thoroughly examined in the next section.

In summary, all of the geospatial processing services implemented perform both basic, general geoprocessing routines and particular to our case study requirements. Next section details how these WPS services have been designed and implemented in our Geoportal application.

[12] http://code.google.com/apis/kml/documentation/

5 Geospatial processing services

In order to provide useful geospatial processing services that suit the concrete requirements of hydrological models, we have identified basic functions shared among the analysis tasks. The ultimate goal is to create a library of geospatial processing services in which customized and elaborated functions rely on other much more simple, atomic and well-tested functions. In this way, the reuse of geospatial processing services is fostered because the process of creating new complex geospatial processing services is made possible by mainly reusing already available geospatial processing services from shared libraries [5].

Our design strategy begins by identifying the atomic functions required for the use case. Then we consider a suitable basic geospatial processing service as one which performs a basic function, can be easily tested and is domain-independent enough to be applicable to other contexts. Some examples are geospatial processing services concerned with topological relations such as intersect with, within, crosses, contains, etc., as well as methods for calculating geospatial proximity or distances among geospatial objects and spatial operations like buffer, area and volume. On the other hand, customized geospatial processing services can be defined as those built on basic geospatial processing services to create more elaborated, customized, and dependent-domain. This kind of services normally performs a specific task in a certain domain which cannot be applicable to other contexts. In this sense, they are similar to the concept of opaque or aggregate service chaining defined by OGC as one approach for web service chaining [1].

Once identified the services, they are grouped into modules with similar functionality. Transforming them into executable WPS is straightforward: each module is a geospatial processing service (WPS) whereas each function either basic or customized is implemented as a process served by the WPS *describeProcess* interface. Table 1 shows the WPS library with the available processes. Only ElevationZoneCalculation and SnowCoverAreaCalculation are customized processes.

WPS module	**Concrete geospatial processing services**
Topology WPS	**Area**, **Intersect**, Union, Contains, Buffer, MaxExtent
ImageProcessing WPS	**Slicing**, SlicingRanges, **Vectorize**, **CoordinateElevation**, **StationsElevation, HypsometricElevation**, **ElevationZoneCalculation, SnowCoverAreaCalculation**
Chart WPS	**DepletionCurvesPlot, DischargePlot**
CoordinatesTransformation WPS	TransCoordGMLPoint, **TransCoordPoint**, TransCoordPoint7P
DataFormat WPS	TransSHPEPSG, TransSHPtoGML

Table 1. List of WPS services in the Geoportal. Bold denotes processes already working in the Geoportal.

For the implementation of the WPS library we have chosen the OGC WPS specification implementation [21] from the 52º North Open Source Initiative[13], which is an open source platform developed in Java. By using this framework, which provides us with the WPS interface to connect to, we have implemented the algorithms required for the processes listed in Table 1.

To illustrate one basic process we can see how the Area process within the Topology WPS works. First, the Geoportal connects through the WPS Client API to the Topology WPS. Then, when the *describeProcess* interface is requested, a Java object WPSProcess (internal class in the WPS Client API) is instantiated containing information of the process demanded. In this case, this object will specify that the Area process requires a geometry figure like a *Polygon Collection* in GML format. The Geoportal thus sends a *execute* request through the WPS Client API with the basin polygons in GML format, and gets an *execute* response from the Area process containing a real number specifying the basin area. Again, the WPS Client API extracts this value and forwards it to the Geoportal.

Figure 4 illustrates how a customized geospatial processing service is implemented as a chain of basic WPS controlled by the customized process itself. Each basic process is performed as explained previously yet in this case only the final result (step 8) is forward to the presentation layer. In particular, the SnowCoverAreaCalculation process in Figure 4 is composed of a sequence of four basic processes: Vectorize, SHP2GML, Intersect, and Area.

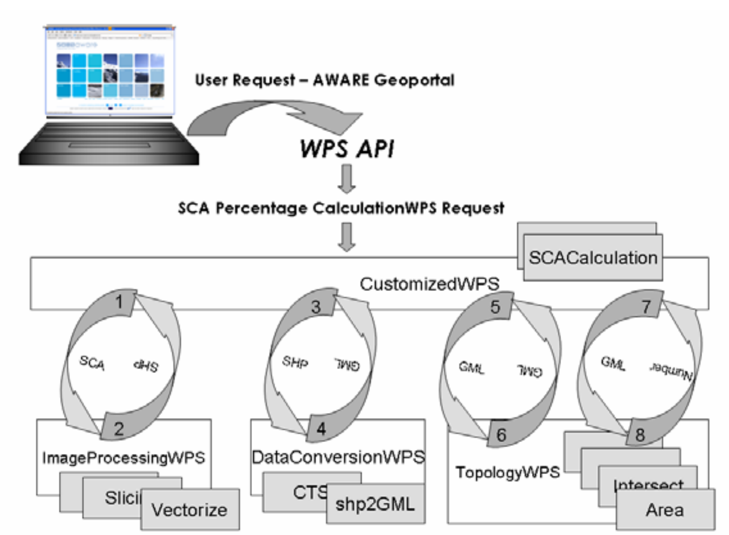

Fig. 4. Snow Coverage Area calculation WPS

[13] http://52north.org/

Figure 5 shows the ImageProcessing WPS and how it is integrated in the system architecture (see Figure 1). This WPS offers processes related to raster image operations like slicing or classifying, where each image cell is classified into categories according to a threshold or a range and a concrete image band. Also it provides a Vectorize process that creates vector data as a set of GML polygons representing the previous classified image. This WPS allows users to extract and process needed information along the model execution without being continuously managing the DEM file, whose size is sometimes too large to be efficient to work with.

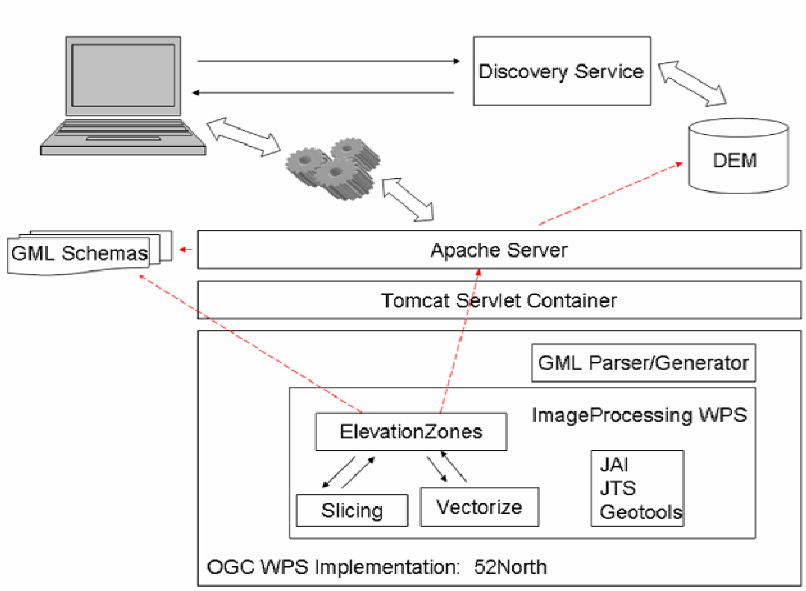

Fig. 5. Image Processing WPS

Another example is the geospatial processing service for creating the elevation zones (see results in Figure 3). To implement this complex task we have designed the customized ElevationZonesCalculation process belonging to ImageProcessing WPS module. Given a DEM file and an elevation range as inputs, this customized process classifies, vectorizes and extracts the polygons in GML format corresponding to the elevation zones. Figure 6 shows graphically the steps performs within the ElevationZonesCalculation process. Firstly, scientists search for DEM data for the study basin in available catalogue services. Once DEM references are found, the Geoportal invokes the ImageProcessing WPS service via the WPS Client API. This service uses internally a few open source libraries such as JAI[14] (Java Advanced Im-

[14] http://java.sun.com/javase/technologies/desktop/media/

aging API) for the image processing, JTS[15] (Java Topology Suite), GeoTools[16] for the geometric model, edge detector open source software to vectorize the images, and the GML parser integrated in the 52º North WPS implementation to generate and return GML format. We have tested with the ElevationZoneCalculation service implementation that interoperability and integration of all of these open source components are possible, though encountering some problems as discussed in the next section.

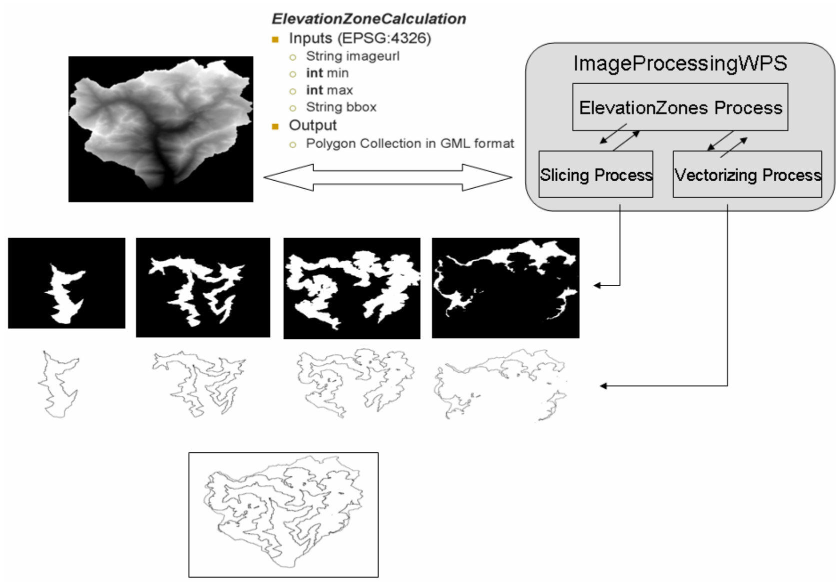

Fig. 6. Elevation zones calculation

6 Conclusions and lessons learnt

We have presented a web-based application that guides expert users in running hydrological models by processing data within a set of distributed geospatial processing services. We have also tested that complex functionality can be processed by connecting to simple, distributed and basic services which can be reusable in other different scenarios. The use of standards is crucial to implement such an application, making it possible in principle the interoperability among all the com-

[15] http://www.vividsolutions.com/jts/

[16] http://geotools.codehaus.org/

ponents involved. In addition the Geoportal has been built on top of an open SDI infrastructure, taking advantage of its benefits and leading to interoperable open software architecture for hydrological applications. We have also shown how it is technically possible that SDI can be used to solve real issues in a more flexible and scalable manner than ad-hoc and stand-alone applications [20]. A pending issue in our work is to provide suitable search mechanisms to find efficiently distributed data and services. In SDI context, these search mechanisms are traditionally catalogue services in which metadata records are essential to describe data and services [22]. However, we find that service metadata need still further research, especially for discovery of WPS services.

Although theoretically the use of standards should be sufficient to achieve interoperability, we found that each vendor implementation differ from each other. Some decisions taken with regards to specific vendor implementations have had a great influence in the target application, making it difficult to reach interoperability at programmatically level (in practice). Some lessons to keep in mind about data integration would be that simple, structured data (e.g. KML) is easier to manage and process than powerful but complex data (e.g. GML) which is more sensitive to failure when processing information. It is assumed that GML has great advantages as a language to integrate disparate formats and to serve data through services [14]. For instance GML documents permit representing complex spatial models by nesting geographic features in a XML way. Yet we have experienced more difficulties handling GML data rather than KML data. In some cases where data complexity is not a system requirement, KML may be a valid, efficient mechanism for exchanging and visualization geospatial data.

Extracting information from huge GML documents has been revealed as an important issue. Processing a GML document may become sometimes challenging due to low performance when working with huge data sets. Beyond text overhead and verbosity inherent in GML (and XML in general), this limitation is caused by the latency produced in data transfers between servers and clients. In our particular case, one way to overcome partly this limitation has been to work with referenced data rather than the GML data themselves. When different basic WPS services and customized WPS (chains) are called, we make reference to the GML data which are only transferred when they are really needed for processing or for visualization purposes.

Other important issue when working with GML is connecting GML data created by different vendor's services. Schemas created by different OGC services implementations are not completely valid, so they cannot be validated by most of the XML readers and then they cannot be parsed by most of the open source GML parsers [14]. Each application creates their own schemas that work normally only within its particular context. We performed extensive research on GML and GML schema, ending with a valid GML to produce basins and elevation zones. The flexibility and extensibility of GML can be then seen as a weak point at practice level when talking about interoperability. It has been a difficult task to generate GML data according to the needed schemas that were successfully parsed by all the GML parsers and generators used in the Geoportal. For example, it was relatively easy to generate a simple subset of GML that was accepted by components involved in the Geoportal like 52º North WPS Implementation and the GeoTools library. The compatibility problems arose when we tried to use standard and complete GML files together with other GML produced by other software components, in order to be transparently used by our WPS services. It was not possible, to our best knowledge, to find a general, valid schema able to be used for every GML files generated by the different software tools and technologies used in the Geoportal.

Integrating different open source libraries implies sometimes a high development complexity, because quite often on-going projects are not very stable and one becomes part of the testing team, facing development bugs which have to be solved. This is even more accentuated when the project belongs to a recent research technology, which is the case of our Geoportal implementation. The OGC WPS specification seems at the moment to be sufficiently mature to be implemented as we found using the 52º North WPS Implementation. However it is still in experimental phase.

Finally, we encountered problems in the implementation of the Geoportal user interface due to multiple projection systems. Google Map viewer uses a common geographic projection that refers to WGS84 (EPSG code 4326) as a pair of coordinates (longitude, latitude), yet geospatial data are given and processed, normally in hydrological applications, in distinct projection systems. For this reason, coordinate transformation services are necessary in the Geoportal incrementing then the response time to the user. Finally, some minor problems were found both integrating different technologies in the presentation layer (for example Javascript, JSP, XSLT, etc.) and also portraying files too large for the web mapping viewer.

A first observation derived from our experience in this project, which coincides with conclusions in previously referenced work, is that the approach based on distributed geoprocessing services leads to a collection of reusable geospatial processing services, available for other users in the case that they are well-documented and registered in open catalogues. This is possible in principle because WPS geospatial processing services do not work with pre-established datasets but rather they preserve a loosely-coupled relationship between data and processing capabilities (algorithms), making it possible to chain them to other geospatial web services such WMS and WCS. One of the problems being partially addressed in the Geoportal application is when a geospatial processing service exchanges and processes large amount of data, which still needs further research for the geospatial community.

Acknowledgments: This work has been partially supported by the AWARE project SST4-2004-012257 co-funded by the EU and the GMES initiative. Institut Cartogràfic de Catalunya (ICC) assisted in the design of the Geoportal.

References

1. Alameh N (2003) Chaining Geographic Information Web Services. IEEE Internet Computing 7(5):22-29
2. Anderson G, Moreno-Sanchez R (2003) Building Web-Based Spatial Information Solutions around Open Specifications and Open Source Software. Transactions in GIS 7 (4): 447–466
3. Brabec F, Samet H (2007) Client-Based Spatial Browsing on the World Wide Web. IEEE Internet Computing 11(1): 52-59
4. Caldeweyher D, Zhang J, Pham B (2006) OpenCIS-Open Source GIS-based web community information system. International Journal of Geographical Information Science 20: 885-898
5. Díaz L, Costa S, Granell C, Gould M (2007) Migrating geoprocessing routines to web services for water resource management applications. In Proceedings of 10th AGILE Conference on Geographic Information Science (AGILE 2007), Aalborg (Denmark)
6. Foster I (2005) Service-Oriented Science. Science 308: 814-017
7. Friis-Christensen A, Bernard L, Kanellopoulos I, Nogueras-Iso J, Peedell S, Schade S, Thorne C (2006) Building service oriented applications on top of a spatial data infrastructure – a forest fire assessment example. In Proceedings of 9th AGILE Conference on Geographic Information Science (AGILE 2006), Visegrad (Hungary)
8. Granell C, Díaz L, Gould M (2007) Managing Earth Observation data with distributed geoprocessing services. In Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS 2007), Barcelona (Spain)
9. Jhingran A (2006) Enterprise information mashups: integrating information, *simply*. In Proceedings of the 32nd international Conference on Very Large Data Bases. VLDB Endowment, 3-4.
10. Jones MT (2007) Google's Geospatial Organizing Principle. IEEE Computer Graphics and Applications 27(4): 8-13
11. Kiehle C (2006) Business logic for geoprocessing of distributed geodata. Computers & Geosciences 32: 1746-1757
12. Lemmens R, Wytzisk A, de By R, Granell C, Gould M, van Oosterom P (2006) Integrating Semantic and Syntactic Description to Chain Geographic Services. IEEE Internet Computing 10(5): 42-52
13. Liu X, Hui Y; Sun W; Liang H (2007) Towards Service Composition Based on Mashup. In Proceedings of 2007 IEEE Congress on Services, 332-339
14. Lu C-T, Dos Santos R, Sripada L, Kou Y (2007) Advances in GML for Geospatial Applications. GeoInformatica 11(1): 131-157
15. Masser I (2005) GIS Worlds; Creating Spatial Data Infrastructures. ESRI Press, Redlands, CA
16. Mitchell T (2005) Web Mapping Illustrated. O'Reilly Media, Sebastopol, CA
17 Nogueras-Iso J, Zarazaga-Soria J, Muro-Medrano P (2005) Geographic Information Metadata for Spatial Data Infrastructures – Resources, Interoperability and Information Retrieval. Springer, Berlin
18. Peng Z-R, Tsou MH (2003) Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks. Wiley, Hoboken, NJ
19. Rampini A, de Michele A, Lehning M, Blöschl G, Brilly M, Llados A, Sapio F, Gould M (2006) AWARE: A tool for monitoring and forecasting Available Water Resource in mountain environment. Geophysical Research Abstracts 8(10780)
20. Scholten M, Klamma R, Kiehle C (2006) Evaluating Performance in Spatial Data Infrastructures for Geoprocessing. IEEE Internet Computing 10(5): 34-40
21. Schudt P (ed) (2007) OpenGIS Web Processing Service Version 1.0.0, Open Geospatial Consortium
22. Smits PC, Friis-Christensen A (2007) Resource Discovery in a European Spatial Data Infrastructure. IEEE Transactions on Knowledge and Data Engineering 19 (1): 85-95

23. Soh L-K, Zhang J, Samal A (2006) A Task-Based Approach to User Interface Design for a Web-Based Hydrologic Information Systems. Transactions in GIS 10 (3): 417–449
24. Yuan Y., Cheng Q. (2007) Integrating Web-GIS and Hydrological Model: a Case Study with Google Maps and IHACRES in the Oak Ridges Moraine area, Southern Ontario, Canada. In Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS 2007). Barcelona (Spain), July 2007 (in press).