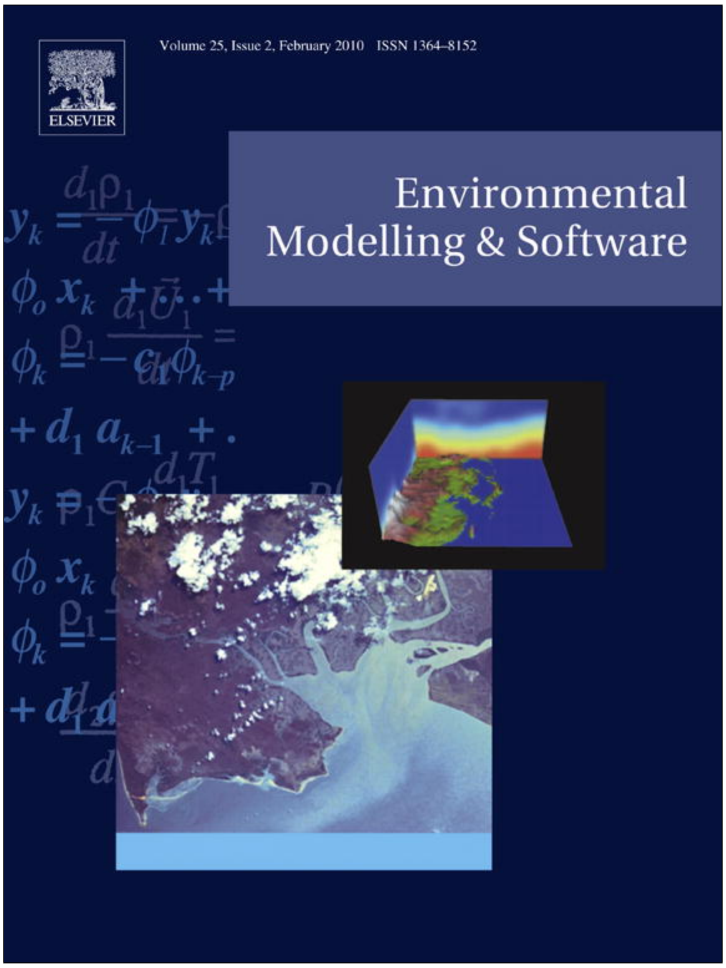
Volume 25, Issue 2, February 2010 ISSN 1364-8152
ELSEVIER
Environmental
Modelling & Software

Environmental Modelling & Software 25 (2010) 182–198

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

Service-oriented applications for environmental models: Reusable geospatial services

Carlos Granell*, Laura Díaz, Michael Gould

Center for Interactive Visualization (CeVI), Department of Information Systems, Universitat Jaume I, 12071 Castellón, Spain

ARTICLE INFO

Article history:
Received 3 November 2008
Received in revised form
11 August 2009
Accepted 21 August 2009
Available online 19 September 2009

Keywords:
Geospatial processing services
Application and service integration
Service reuse
Environmental models
Service-oriented architecture, SOA
Spatial data infrastructure, SDI

ABSTRACT

Environmental modelling often requires a long iterative process of sourcing, reformatting, analyzing, and introducing various types of data into the model. Much of the data to be analyzed are geospatial data—digital terrain models (DTM), river basin boundaries, snow cover from satellite imagery, etc.—and so the modelling workflow typically involves the use of multiple desktop GIS and remote sensing software packages, with limited compatibility among them. Recent advances in service-oriented architectures (SOA) are allowing users to migrate from dedicated desktop solutions to on-line, loosely coupled, and standards-based services which accept source data, process them, and pass results as basic parameters to other intermediate services and/or then to the main model, which also may be made available on-line. This contribution presents a service-oriented application that addresses the issues of data accessibility and service interoperability for environmental models. Key model capabilities are implemented as geospatial services, which are combined to form complex services, and may be reused in other similar contexts. This work was carried out under the auspices of the AWARE project funded by the European programme Global Monitoring for Environment and Security (GMES). We show results of the service-oriented application applied to alpine runoff models, including the use of geospatial services facilitating discovery, access, processing and visualization of geospatial data in a distributed manner.

Software Availability

Name of software: AWARE Application—Geoportal
Developers: Centre for Interactive Visualization (CeVI), Universitat Jaume I
Contact information: Av Vicent Sos Baynat, s/n, Universitat Jaume I, 12071 Castellón, Spain
Hardware required: None
Software required: Internet browser (Firefox, Internet Explorer, etc.)
Program language: Java (server) and client script technologies (JavaScript, XML, XSLT, etc.)
Availability and cost: Users can access directly at website http://geoportal.dlsi.uji.es/aware for testing purposes. AWARE Application still remains closed to AWARE project partners although user accounts can be made available to researchers upon request to the authors.

1. Introduction and problem statement

Environmental modelling such as that used for estimating river runoff often requires a long iterative process of sourcing, reformatting and introducing various types of data into the model. This is true for wide range of geosciences disciplines (climatology, geomorphology, remote sensing, etc.), each of which has multiple data models, formats, and protocols to choose from. The choice is based partially on modelling requirements but also on the data processing software available. In hydrology often it is necessary to bring together disparate datasets to try to interpret the resulting runoff predictions and ultimately to improve environmental decision-making (Liu et al., 2008; Denzer, 2005). This implies challenging tasks for scientists such as locating and gathering appropriate geospatial datasets (digital terrain models, satellite imagery, measuring station point data, etc.) for their models. Once geospatial datasets are collected scientists in practice then waste considerable time on repetitive, time-consuming operations to integrate such disparate datasets (reformatting, resampling, transformation, interpolation, etc.) rather than focusing on real scientific analysis and decision-making (McColl and Aggett, 2007; Goodall et al., 2008; Liu et al., 2008; Denzer, 2005).

* Corresponding author.
E-mail addresses: carlos.granell@lsi.uji.es (C. Granell), laura.diaz@lsi.uji.es (L. Díaz), gould@lsi.uji.es (M. Gould).

1364-8152/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.envsoft.2009.08.005

To overcome these limitations Mineter et al. (2003) raised the need for a new generation of environmental applications shifting from centralized, desktop applications toward the provision of distributed geospatial services and components. They foresee the use of emerging technologies like web services (Alonso et al., 2004) and Grid (Foster et al., 2001), and emphasize the need of software modularity and reuse by means of structuring applications as a set of connected services. This paper responds to their call, to address some of these unresolved research topics in terms of distributed computing, data availability, and interoperable services. Our aim here is to move beyond the initial obvious steps of incorporating services for gathering (accessing) geospatial data in environmental systems (Goodall et al., 2008), to focus on the next key link of the modelling chain: provision of geospatial processing capabilities as remote services in order to fulfil some of the important needs of distributed environmental applications. Furthermore, the issues addressed in this paper are the following:

- The lack of accessibility and interoperability of geospatial data.
- Exposing common scientific operations as modular geospatial services to be reused in workflows within environmental models.
- Enhancing integration of processing capabilities with data discovery, collection and visualization tasks using a stepwise "wizard" web application.

In this paper we assume that environmental models are scientific models that rely on correlating and analyzing data on watersheds (since our case study is in hydrology) and that requires some level of cooperation among various scientists facilitated by the use of Information Technologies (IT). We address key IT issues involved in geosciences disciplines by exploring how environmental applications can be built on distributed geospatial services and components, relying on open architectures. The goal is to seamlessly integrate some of the tedious and often disconnected tasks associated with geospatial data, such as discovery, collection, consistency-checking, transformation, interpolation, etc., to hopefully allow scientists the freedom to focus on more scientific tasks such as analysis, decision-making, and interpretation of modelling results. In this sense, we take here a modeller perspective rather than a decision-making focus in order to build operating environmental models based on distributed components. Future steps will address customized user interfaces in function of end user's needs and level of expertise (non-expert users, decision makers, technicians, etc.).

We propose the application of a service-based architecture and some service design principles addressing trade-offs among modularity, reuse, and efficiency, and aimed at achieving an optimal service granularity. These services can be shared, integrated, and most importantly reused to assemble ad-hoc, distributed web applications. In particular we have demonstrated our application in the running of two well-known hydrological models, in the framework of the European Union-funded AWARE[1] project, though our approach is generic enough so that it may be applied to other geosciences disciplines and areas.

In the following section we introduce the research project in which this work has been conducted. In Section 3 we outline some relevant concepts and review related work. Section 4 presents the architecture of our approach and elaborates on the key components and modules. Section 5 details the service design principles and the strategy for enhancing service integration and reuse. Examples of the resulting AWARE Application for hydrological models are described in Section 6. Finally, Section 7 concludes by summarizing the key features of our approach and discussing ongoing work.

2. Context

Our work has being carried out in the framework of the AWARE ("A tool for monitoring and forecasting Available WAter REsource in mountain environments") project, funded under the GMES (Global Monitoring for Environment and Security) programme of the European Union. The aim of AWARE is to offer on-line geospatial processing services and other appropriate tools to help monitor and forecast water resources derived from a specific quantity and distribution of snowmelt in Alpine regions. One of the project results has been a web service-based application, hereafter referred to as AWARE Application, which allows hydrologists and other scientists to calibrate and execute river runoff models, and to interpret the results. In particular, the AWARE Application supports two hydrological models: the Snowmelt Runoff Model, or SRM (Martinec et al., 1994), which is shown in Section 6, and the TUW-HBV model (Parajka et al., 2005).

The SRM model simulates and forecasts daily stream flow in mountain basins where snowmelt is a major runoff factor, as in the case of the Alps. It is out of the scope of this paper to describe this hydrological model in detail; however it is important to indicate the data types necessary for this model in order to better understand its complexity and heterogeneity. In particular, the SRM model requires a wide variety of datasets such as measurements of ground temperature and precipitation collected from the meteorological community, stream gauge measures, satellite imagery from remote sensing specialists representing the snow coverage area of the watershed, digital elevation models, locations of the weather stations, the geographic boundary of the watershed, and many other variables and parameters related to the physical characteristics of the watershed. It seems probable that analyzing inefficiently these disparate datasets (often with several tools and information systems or even manually) may lead to incomplete and inaccurate results, as well as consuming an inordinate amount of time. The problem may get worse because of the vast amount and heterogeneity of data sources that potentially meet the model requirements. Among the important real impediments and challenges of the researchers involved in the AWARE project were lack of data accessibility (need to discover appropriate geospatial datasets), service and data interoperability in terms of integrating collected datasets with other disparate datasets, and then the lack of services for data consistency-checking prior to input into the models. These challenges motivated us to investigate possible partial solutions to facilitate the tedious tasks of collection, validation, processing, analyzing, and integration of geospatial data for use in the selected runoff models.

3. Related work

Our work is centred on the concept of service to create distributed applications needed for the collaborative research environment. This section discusses approaches for distributed computing and also reviews related work on geospatial services integration and reuse. As mentioned earlier, instead of discussing the environmental models themselves, we analyze some of the relevant environmental applications and tools which help scientists to run their models.

3.1. Services and architectural styles for collaborative research

Geosciences research is a multidisciplinary field that demands not only heterogeneous data and models but also includes

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

a multitude of expert profiles such as technologist, remote sensing specialist, and geoscientist: experts who collect, store, manage, organize, and process data using environmental models to produce meaningful information for decision makers. This scenario requires the use of new architectural styles which oppose centralized, isolated solutions, and instead support distributed processing capabilities and remote communications, necessary ingredients to successful collaborative and multidisciplinary research.

A recent trend in collaborative science on the Web is the concept of Web Science (Berners-Lee et al., 2006; Shneiderman, 2007). This term, actually still a vision, covers many aspects in the Web context such as tools, data representation, infrastructures, mechanisms and so on to eventually facilitate discovery, integration, processing, and analysis of datasets from disparate and distributed data sources. Hey and Trefethen (2005) propose the use of cyberinfrastructure to support the needs of multidisciplinary collaborative research. Cyberinfrastructure allows research teams to share distributed data resources (e.g. datasets, processing power, etc.) through high-speed networks. Other authors (Goodall et al., 2008; Denzer, 2005) propose cyberinfrastructure and distributed infrastructures as solutions to the challenge of generic interoperability and integration. Several attempts have been made to provide these services in diverse disciplines as for example the Geosciences Network (GEON[2]) project focused on developing a cyberinfrastructure for integrative geosciences research.

Recent approaches in enterprise business integration, in search of simplified processes, are mostly driven by the emergence of the Service-Oriented Architectures (SOA), which are focused on an architectural style to design applications based on a collection of best practices, principles, interfaces, and patterns related to the central concept of service (Papazoglou and Heuvel, 2007; Aalst et al., 2007). In SOA, services play a key role and become the basic computing unit to support development and composition of larger, more complex services, which in turn can be used to create flexible, ad-hoc and dynamic applications. The main design principle behind SOA is that a service is a standards-based, loosely coupled unit composed of a service interface and a service implementation. Service interface describes the functional capabilities of a service. Service implementation implements what a service should execute. This principle provides a clean separation of concerns especially between service interfaces (what services offer to the public community) and internal implementations (how services work). Essentially SOA introduces a new philosophy for building distributed applications, where services can be discovered, aggregated, published, reused, and invoked at the interface level, independently of the specific technology used internally to implement each service.

At the time of implementation SOA-based services must make use of concrete languages and protocols. Here is where web service technology gains importance because it increasingly is becoming the choice to implement SOA-based applications. Web services (Alonso et al., 2004) are, by definition, loosely coupled independent units and are well described (interface description contains functional properties), thereby promoting one of the goals of SOA: enabling interoperability or the ability of services to interact with minimal knowledge of the underlying structure of other services (Sheth, 1999). Interoperability is achieved (or optimized) by using standard interfaces. Web service technology includes various standards such as Web Service Description Language (WSDL) for the description of service interfaces, Universal Description, Discovery and Integration registry (UDDI) for their advertisement and discovery, and Simple Object Application Protocol (SOAP) that enables communication among services (Curbera et al., 2002).

In the geospatial context, the current research trends for access and discovery of large-scale geospatial datasets and for improving data and service interoperability are being addressed by a European Union framework directive called INSPIRE (INSPIRE, 2007), which is concerned with coordinating Spatial Data Infrastructure (SDI) procedures and methodologies at the European member state level. SDIs are collaborative, Internet-based information systems designed to facilitate geospatial data sharing by harmonizing data specifications and mandating their widest possible accessibility and at the lowest possible cost (Masser, 2005). Indeed, an SDI should be formed by several interconnected systems that in turn could be seen as SDIs themselves. The INSPIRE technical architecture includes metadata, spatial datasets, and network services within a layered architecture that differentiates the Presentation layer (applications and Geoportals), the Service layer, and the Data Sources layer, as illustrated in Fig. 1. Essentially, client applications access geospatial data stored in repositories through services in the middleware layer. Although SDI nodes may rely technologically on cyberinfrastructure to provide increased distributed hardware capacity for handling huge datasets, conceptually, the distributed GIS approach to SOA-based applications is perhaps best represented by the SDI paradigm, in which standardized interfaces are the key to allowing geospatial services to communicate with each other in an interoperable manner responding to the true needs of users (Foster, 2005; Friis-Christensen et al., 2007; Kiehle et al., 2006; Alameh, 2003). As will be seen in Section 5, the vision of creating and sharing services within the SDI-SOA paradigms has guided us in the development of service-oriented applications to minimize the problems of data accessibility, and services and operations interoperability.

3.2. Geospatial services

Many of the benefits of general services can be extrapolated to geospatial services as well. Services are basic pieces that allow users to access and share information faster and more efficiently by essentially decoupling service description from implementation. What make geospatial services slightly different from "common" services is the inherent characteristics of geospatial data on which they operate, which are diverse, huge, and complex (Granell et al., 2007). This complicates enormously the integration of geospatial data because of the variety of existing data models, data formats, data semantics, and spatial relationships (contains, cross, touch, etc.), which in practice are limiting factors to ensuring true geospatial interoperability. Nevertheless, service-oriented applications involving geospatial data are still possible in part because the geospatial community, under the auspices of the Open Geospatial Consortium (OGC[3]), has proposed specific interface descriptions, some complementary to those used for web services (e.g. WSDL, SOAP) mentioned earlier, others more appropriate for dealing with the "special" features of geospatial data (for example to offer better support in defining geospatial data schemas). That is to say, SOA and web services principles remain intact, the main difference residing in the description languages used.

The INSPIRE directive's implementation rules propose a network of services classified in groups according to functionality, i.e. what the service does in terms of capabilities, to embrace all needed geospatial or GIS-like functionalities. Each group is called a service type. As services are key in the INSPIRE Directive, the

[2] http://www.geongrid.org.

[3] http://www.opengeospatial.org.

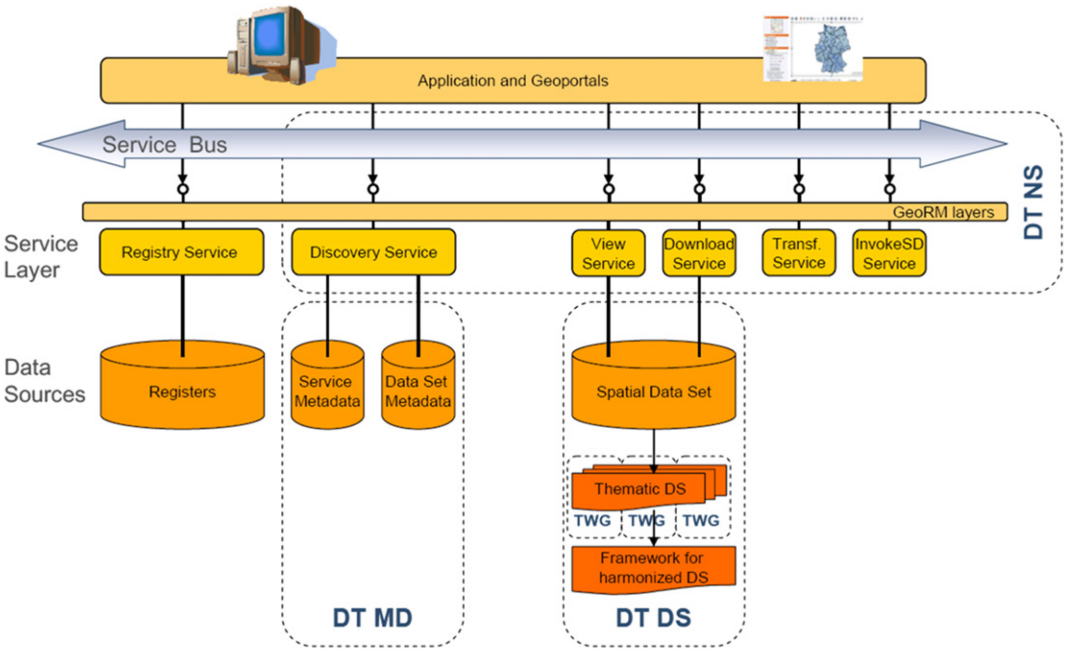

Fig. 1. INSPIRE technical architecture (INSPIRE, 2007).

Service layer becomes the core of the INSPIRE architecture. Fig. 1 shows the Service layer which contains the INSPIRE Service Types (yellow boxes) as contemplated in the directive. These service types are: Registry, Discovery, View, Download, Transformation and Invoke. Transformation services and invoke services limit their functionality to schema and coordinate transformation, while certain advanced aspects such as service chaining need further discussion and consensus. This paper focuses on service aspects such as processing capabilities in general, service chaining and reuse.

Geospatial services may manage different data models and therefore they require different service interfaces. This becomes clear when one wishes to chain one service returning bitmap files (PNG, JPEG) and other computing spatial operations over vector cartographic data. Both are geospatial services however they assume different data models. Aside from the more relevant examples of OGC standards such as Web Feature Service (WFS), Web Map Service (WMS) and Catalogue Service for Web (CSW), OGC has also released, especially for research fields requiring raster image and other geosciences data processing capabilities, the Web Processing Service (WPS) specification version 1.0 (Schut, 2007). In short, WPS provides a service interface for exposing and executing processes of any granularity, ranging from a simple polygon area calculation to entire environmental models. A WPS-based service (geospatial processing service) offers three methods to expose the functionality of a process (geospatial or not): the *getCapabilities* method, common among OGC services, asks about the available processes within the service. Each process' input and output parameters are described in a very detailed way by the *describe-Process* method, and finally the *execute* method actually invokes the geospatial process with concrete input parameters and returns the results. WPS has been used widely for interfacing the geospatial services described in Section 5.

3.3. Geospatial service integration and reuse

Service composition is a key mechanism in SOA, as it allows creation of value-added services by integrating other services. Indeed, service composition includes necessarily the reuse of existing services because services can be dynamically used not only once but also shared among multiple applications, now considered best practice, to assemble dynamic applications. The web service community has long pursued the dream of multiple reusable services becoming available to a wider community, and which are ready to be used and combined for multiple unforeseen purposes (Petrie and Bussler, 2008). Academic and industrial consortia have already proposed multiple solutions for web services composition and execution both in domain-independent contexts (Dustdar and Schreiner, 2005; Agarwal et al., 2004; Milanovic and Malek, 2004; Ko and Neches, 2003) and in the geospatial domain in particular (Alameh, 2003; Chang and Park, 2006; Lemmens et al., 2006; Yue et al., 2007; Lutz, 2007).

In the geospatial domain, Alameh (2003) was one of the first attempts at addressing the problem of geospatial service chaining or composition. Chang and Park (2006) present a web service-based model for dynamic and interoperable Internet GIS applications. They focus also on addressing the interoperability and integration issues in the context of distributed systems. Some elemental GIS components implemented as XML web services are shown, that can be distributed on multiple servers and then integrated by client applications when necessary. Then they adapt specific standards for modelling geospatial data (e.g. GML), though standards for service interfaces are avoided, in contrast to our approach. Although in theory implementing services as XML-based web services should increase chances of distributed system interoperability, still many interoperability problems often arise in practice when different tools from different providers are pieced together (Díaz et al., 2008a; Lu et al., 2007).

Recent works have highlighted the need to incorporate geoprocessing capabilities in distributed applications, leading to so-called geoprocessing services. The ability to not only access and visualize geospatial data but also process them seems to be a great benefit for SDI, since this opens the door to creating richer services that might be applied to wider scenarios. Michaelis and Ames (2009) have performed a feasibility study of the WPS specification in client-side applications. They conclude that "the WPS proposal was found to be workable as currently designed, and is indeed suitable for many GIS tasks." Kiehle (2006) and Yang et al. (2008) also discuss the use of WPS-based geoprocessing services applied to real world examples. Friis-Christensen et al. (2007) have proposed a similar approach for distributed geoprocessing based on SDIs, however, they propose a different approach for creating geospatial services, concentrating all required functionalities in a single, publicly accessible geospatial service. Although their system has advantages in terms of performance, flexibility and reuse decrease greatly. We take an intermediate approach to defining services, maximizing to the extent possible service reuse while offering services at different granularity levels for performance reasons. The optimal balance between service reuse and performance will depend ultimately on the specific requirements of the target application (see Section 5).

As many authors point out, the idea of heterogeneous service discovery and composition remains still an open issue, not only for technical reasons but also because users often prefer to use reliable, trusted services from know, trusted service providers. Technologically speaking, current service composition approaches tend still to be quite predefined and static, where service interactions in terms of input and output matching among services is anticipated at design time. On the other hand, from a practical viewpoint, sharing and composing services would make more sense in concrete domains where potential users could take advantage of sharing and, more importantly, quick and simple reuse (and adaption) of "trusted" services provided by colleagues. For example researchers running critical climate change models would likely not trust an anonymous, badly-documented service with no inherent guarantee as to the results obtained. Therefore, our assumption here is that service-oriented applications can be successfully applied either to narrow (concrete communities) or wider (the entire Web) scenarios, however in practice the services which are focused on a given community will naturally increase the level of services reuse.

Semantic issues and challenges also have widely researched in web service domain (McIlraith et al., 2001). Several research works have proposed ontology-based approaches to enhance resource discovery and service interoperability in the geospatial domain (Lacasta et al., 2007; Smits and Friis-Christensen, 2007; Reitsma and Albrecht, 2005; Lutz, 2007; Yue et al., 2007), though discovering semantically suitable geospatial services still remains a very challenging task (Lutz, 2007). Semantic aspects are out of the scope of this paper, and to simplify the AWARE Application was designed to allow users to discover and retrieve datasets only relevant to user-defined context properties gathered as the AWARE Application is being executed (see Sections 5 and 6). Even in small communities with few services available, the possible combinations of these services may be potentially in the dozens or hundreds, for which discovery for any service-oriented application becomes an important aspect of the workflow.

3.4. Applications and tools for environmental models

Many applications and tools currently exist to enhance the interaction with environmental models, and these possess a varying degree of sophistication and functionality. Most are built on top of well-known geospatial software packages, meaning that for the most part they remain standalone desktop applications (Best et al., 2007; Teng et al., 2008; Pecar-Ilic and Ruzic, 2006; Mineter et al., 2003; Jeong et al., 2006). In contrast to these applications, we find distributed, web-based solutions, normally built around web mapping viewer clients which allow the user to visualize multiple datasets (Soh et al., 2006; Goodall et al., 2008), either taken from static repositories or (rarely) as a result of applying data transformations on-the-fly.

Regarding desktop solutions, Best et al. (2007) describe a system using the ESRI Model Builder,[4] with which basic OGC services like WMS and WFS are integrated. Basically, processing tasks are embedded in the system and as such they are neither widely available for other users nor general enough to be reused in other scenarios. Interestingly, they introduce the concept of scientific workflows using geospatial web services in an ecology use case. In our case study, environmental models are split into big steps that in turn contain scientific workflows, which perform various tasks by orchestrating geospatial services. This hierarchy is behind the proposed service design strategy (see Section 5). Teng et al. (2008) present a tool to support spatially distributed hydrological modelling built using ArcGIS.[5] Though not service-based, this scientific tool, like ours, hides the complexity of the computation algorithms behind a user-friendly interface using a stepwise web application.

Pecar-Ilic and Ruzic (2006) present a tool based on Autodesk MapGuide Viewer[6] that aims to provide data conversion and transformation operations among different reference systems for the Danube River data. Similarly, Jeong et al. (2006) describe a hydrology application based on the Interactive Data Language (IDL[7]) software to analyze and visualize hydrologic data. Nevertheless, all of the application examples seen so far follow an "extension" approach, in which existing GIS software packages are "extended" to locally process and display specific datasets.

In the category of distributed applications, Soh et al. (2006) describe a web application to identify drought-vulnerable regions. They propose a combination of data mining techniques to characterize the behaviour of water basins and classify them according to the drought index. Their goals are slightly different from ours however both approaches deal with multiple data types that must be integrated using friendly user interfaces for use by non-experts and experts users indistinctly. It is important to note that geoprocessing capabilities are not present in (Soh et al., 2006) in terms of distributed geospatial services accessible via Web protocols.

Concrete examples of web service technology applied to environmental models, and specifically to hydrology, are actually very limited. Goodall et al. (2008) explore to some extent web service interfaces to provide data access for the National Water Information System in the United States. Nevertheless, none of the previously mentioned applications provide distributed processing capabilities when executing on-line environmental models. In the following sections we describe the conceptual architecture of our system and the strategy followed to design, compose, and reuse services that permit the re-organization of environmental applications as a distributed network of interoperating services.

4. System design

This section describes the AWARE Application's system architecture, which was based on principles from the contexts of INSPIRE, SDI, and SOA frameworks, to overcome the lack of data, and services and operations interoperability.

4.1. Architecture

The AWARE Application integrates a set of modules—grouping client components and services—developed using existing tools

[4] http://www.esri.com/software/arcview/extensions/spatialanalyst.
[5] http://www.esri.com/software/arcgis/index.html.
[6] http://www.autodesk.com/products.
[7] http://www.ittvis.com/idl.

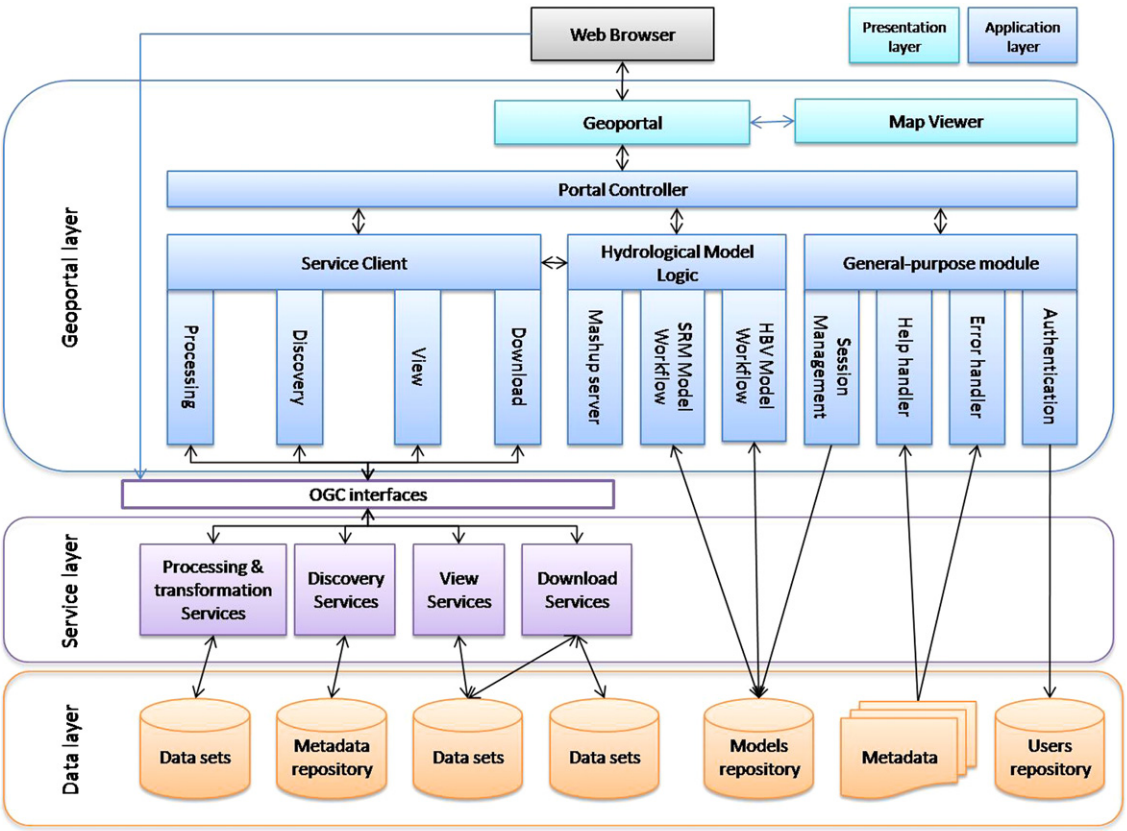

Fig. 2. AWARE architecture and its components.

(e.g. libraries), and servers (Apache Server,[8] Apache Tomcat,[9] Google servers, etc.) and databases for persistent data storage. This may look a priori quite complex because of the multitude of heterogeneous components involved, however we have accommodated them in a service-oriented architecture composed of three layers according to the INSPIRE principles, which serves to greatly reduce the complexity of creating and reusing distributed services.

Fig. 2 illustrates the service-oriented, layered architecture proposed for the AWARE Application. Components and services in each layer perform similar tasks with accordance to the goal of the layer in which they are placed. For example components belonging to the Geoportal layer, top layer of the architecture, are concerned mainly with two tasks, represented in turn as two contained layers: Presentation layer (light blue boxes) and Application layer (dark blue). The former deals with the user interface, user interaction, and data visualization. The latter is concerned with application integration—also called business logic or business integration—as well as instantiation and invocation of service instances at the Service layer. The Application layer also contains the description of other components that perform supporting functions, both in a general sense (user validation, data consistency, etc.), and also for the particular environmental models, such as managing scientific workflows and processing visualization data that will be sent to the Presentation layer. The Service layer comprises a variety of distributed service instances logically clustered according to service types (discovery, download, etc.). Here a service instance is considered a concrete implementation of a service type. The Data layer is focused on databases, data repositories as well as data and services metadata registries. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Although multi-layered approaches have been proposed in other Web-based GIS applications (Chang and Park, 2006; Moreno-Sanchez et al., 2007), INSPIRE-based architectures offer more benefits to end users (and developers), in addition to common technical aspects, such as standard interfaces, service types, policies, agreements, and so on that globally enhance data and service interoperability.

4.2. Multi-layered approach

4.2.1. User interface and data visualization

As Fig. 2 illustrates, the Presentation layer within the Geoportal layer contains two different modules, each one devoted to the capabilities offered in this layer: the Geoportal module for user interface and interaction, and the Map Viewer module for data visualization. Users access the AWARE Application via a web browser, connecting to the Geoportal module, a web-based interface that assembles the user interface's components. This module is a set of web pages that mainly gathers requirements for running environmental models, guides users step-by-step through the creation of input parameters and then model execution, and visualizes model results. The application's user interface has been designed to be simple and consistent throughout the set of web pages. For example scientists can decide to go forward and backward through a given model execution process depending on whether or not a certain step has successfully executed. If not, they

[8] http://httpd.apache.org.
[9] http://tomcat.apache.org.

are able to go back and modify input parameters to tune the results of that step in the model execution.

The Geoportal module plays the role of a one-stop Geoportal (Bernard et al., 2005; Maguire and Longley, 2005), a web application that offers expert users an integrated view to access all of the capabilities necessary for particular contexts. In our case, the Geoportal layer as a whole also acts as a gateway to facilitate connection to remote services—functions, transformations, interpolation routines, data access, etc.—and to configure and run environmental models for a particular watershed of study. One of the strengths of the AWARE Application compared with other environmental model tools and systems discussed in the previous section is that it does not require any special software packages and desktop GIS systems on the client side. The only required software in client machines is a web browser (e.g. Firefox, Internet Explorer) with Internet connection. This is an example of emerging client solutions which are tailored to certain workflows and thus are flexible, and inexpensive in terms of software licensing (Moreno-Sanchez et al., 2007).

Data visualization is carried out by the Map Viewer module. This module includes the mapping mashup client component itself, using the Google Map API (Chow, 2008) that retrieves rendered maps from Google servers, and web client technologies (JavaScript, XML, XSLT, etc.) that allows visualization of additional local data layers and also interaction with the graphical elements represented on a map (e.g. icons). This module then represents information rather than processing it, and is concerned exclusively with the Application layer's modules. Apart from providing data visualization capabilities via traditional 2-D maps, the AWARE Application also provides other useful modules for environmental applications such as creation of diagrams and line plots. Diagrams provide interactive visual synthesis and exploration of scientific data (Wood et al., 2007) and are generated dynamically using geospatial services, as will be described in Section 5.

4.2.2. Service integration and interoperability

In addition to the Presentation layer, the Application layer within the Geoportal layer addresses the issues of service and data integration and enables communication between end users and the distributed services at the Service layer. Service and data integration is at the core of the Application layer. For this reason we have developed a set of modules (blue boxes in Fig. 2) to deal with all aspects traditionally involved in service integration such as discovery, composition, instantiation, and invocation of services. Note also that users can directly invoke the available services without using the Geoportal layer (see direct arrow between top Web Browser box and services in Fig. 2). This behaviour is common in OGC-based services since users can invoke such services both directly via HTTP queries encoded according to OGC-standard specifications (WMS, WPS, etc.) and via Geoportals that hide to some extent the complexity of the underlying HTTP calls. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

As shown in Fig. 2, the Portal Controller plays a central role in the architecture managing interactions between the Presentation layer's and Application layer's modules. This component manages common but necessary capabilities in web applications such as handling input requests and forwarding results to client browsers (normally to the Map Viewer module at the Presentation layer). When the Portal Controller intercepts a new user request for a certain action (e.g. user authentication), it delegates such an action to a dedicated component to deal with it. These components are logically grouped in three modules: a module containing general-purpose functions, another for accessing remote services, and finally a specific module devoted to concrete integration tasks for the environmental models supported in our application. In the following we describe how these modules and their components interact with each other.

The general-purpose module contains functions present in most current Web applications however some are especially useful in environmental domains. The right side of Fig. 2 shows a set of four components for general-purpose capabilities (dark blue boxes):

- *Session Management*. This component allows users to save the status of the current model execution (especially useful during calibration of the hydrological models) at any step in the process, and to allow restart of the model execution at any point and using previous model results. This also facilitates sharing of model calibrations among colleagues because sessions are stored in easily managed XML files.
- *Help Handler*. This provides online contextual help for key items that need to be well-understood and interpreted correctly by scientists, e.g. introduction of input data, appropriate data format, and graphic legends.
- *Error Handler*. Similar to the Help Handler component, this provides concrete error messages ranging from wrong input data to network problems.
- *Authentication*. Users need to be logged into the system to both start a new model calibration and restart previous ones. This helps the session manager to function correctly. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

The Service Client module (dark blue box on the left in Fig. 2) allows users to communicate with service instances at the Service layer. This component collects user queries, encodes them in OGC-standard format, and connects to the corresponding distributed services. As each type of service (discovery, download, etc.) uses different encodings and service interfaces, we have implemented concrete adaptors (vertical boxes connected to Service Client box) for each service type in such a way that other specifications and service interfaces can be easily added, thus providing extensibility and scalability to our system. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

Business integration itself, in terms of data and services, is the responsibility of the Hydrological Model Logic module, which implements auxiliary functions, concrete data constraints, and the business logic of the hydrological models. The structure of this module has been centred on the concept of *workflow*, as defined by Aalst and Hee (2002): a combination of several tasks that follow some rules (e.g. iterations, sequence) to explicitly identify the order in which they are carried out. A task in our context can be thought of as a single unit that performs either auxiliary functions in a general programming language like Java or represents geospatial services such as vector data retrieval querying WFS or soliciting a map from WMS. *Scientific workflow* is another type of workflow that refers to IT-supported scientific activities (e-science) handling heterogeneous data as in the case of the environmental models managed by the components in the Hydrological Model Logic module.

In order to better understand the need of scientific workflows, we outline briefly the hierarchical structure of the SRM model (Martinec et al., 1994) in terms of steps, scientific workflows and tasks. The SRM model is composed of a sequence of large steps created to match the perceived needs of expert users. Each of these steps is in turn a set (one or more) of scientific workflows that are executed without user supervision. Each scientific workflow consists of a predefined chain of tasks—executions of geospatial services, iterations, conditional sentences depending on the value

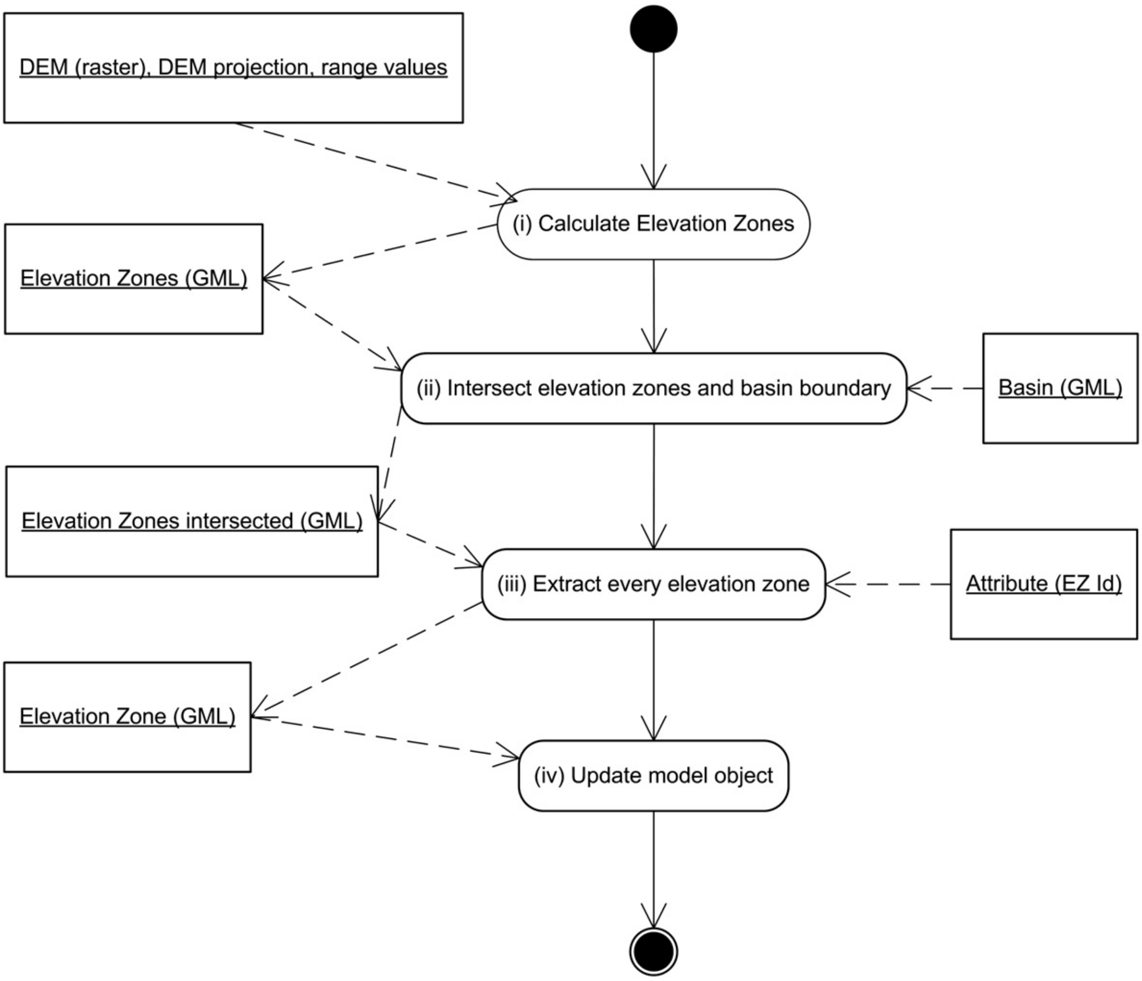

Fig. 3. UML activity diagram of the scientific workflow that calculates the Elevation Zones of a watershed.

of input variables and parameters, etc.—that differ in complexity depending on the difficulty of the tasks.

The Hydrological Model Logic module's ultimate goal is thus to manage the execution of the scientific workflows that shape the hydrological model in question. Rather than providing a unique module, we have intentionally decoupled business integration into several modules and components (Service Client, Mashup Server, etc.) in order to more easily accommodate other hydrological (environmental) models within our system. Because two hydrological models are supported at this moment, two specialized components are needed—SRM Model Workflow and TUW-HBV Model Workflow—to manage the business logic of each model. The addition of additional hydrological models requires implementing a couple of components: a specialized model logic component and the model object (see Models repository in Fig. 2) that stores and retrieves the current status of the model as it is being executed within the AWARE Application. The remaining modules are fully reused without change.

Fig. 3 shows the activity diagram of a scientific workflow (one of the workflows included in step 2 of the SRM model) that calculates the Elevation Zones of a watershed. The specialized SRM Model Workflow component performs the scientific workflow in Fig. 3 by orchestrating (e.g. iterations, sequence) some of the geospatial services available in the Service layer and by managing the data flow. This workflow contains tasks (oval boxes) and input and output data flow (squared boxes). The execution of each task involves the data flow among the precedent and subsequent tasks and service calls to geospatial services via the Processing component in the Service Client module, so each task is actually performed in distributed geospatial services that reside in the Service layer. Once the scientific workflow's results are obtained, the Mashup Server component prepares them for visualization by executing transformation services that convert mainly processing data (workflow results) from Geographic Markup Language (GML) cartographic format (Portele, 2007) to visualization data encoded in KML.[10] Geospatial data ready for visualization are then streamed to the Map Viewer module (Presentation layer).

While scientific workflows lead to a composition mechanism in the Application layer, the service design principles proposed in Section 5 will describe a service integration mechanism in the Service layer, taking into account a trade-off between service reuse and efficiency (performance) to create geospatial services at a variety of granularity levels.

4.2.3. Services and data repositories

Geospatial services occupy the main part of the proposed architecture. These services can provide geospatial and non-geospatial data and functionality, e.g. data extraction from a remote repository, coordinate transformations, format transformations, interpolation

[10] http://www.opengeospatial.org/standards/kml.

routines, map rendering, diagram generation, etc. Service design principles and the set of geospatial services and geospatial processing services will be described in detail in Section 5.

Data repositories are out of the scope of this paper; however it is worth mentioning the presence also of internal data repositories, different from those for geospatial data and metadata, devoted to application requirements such as user profile information and model status. In contrast to the public data and metadata repositories, the internal repositories are managed only by the modules in the Application layer.

4.3. Increasing data accessibility and interoperability

A key task in environmental research is to access the right data at the right time from remote repositories. Data accessibility implies in turn some smaller steps. First the data should be *described* properly, next *searching* methods to locate the data in the corresponding data repositories should be clearly known, and finally it is necessary to interact with these repositories to *retrieve* the data. In the following we see how describe-search-retrieve actions are closely related to the use of standard interfaces and services types proposed in our architecture: a success factor for data retrieval and interoperability.

4.3.1. Service types

Fig. 1 illustrates the service types defined by the INSPIRE directive. Each type defines common capabilities offered by a group of services. Specific service types like discovery services offer end users a common mechanism to search discoverable geospatial data. As users progress through the AWARE Application, automatic queries are issued via a discovery service against catalogues to collect relevant metadata records according to user's requirements. Once relevant data are located, download service types that are also integrated in the AWARE Application enable users to retrieve actual geospatial datasets.

4.3.2. Standard interfaces

Service interoperability is achieved by utilising open geospatial standard interfaces. Interfaces are critical because they indicate how to interact with available services in a uniform and unambiguous manner. It is crucial that descriptions for service interfaces are widely published and become standards for widespread use. Today most of the web services deployed in SDIs use OGC interfaces (Friis-Christensen et al., 2007; Moreno-Sanchez et al., 2007; Kiehle et al., 2006).

Our viewpoint has been to consider environmental applications as distributed applications deployed on top on SDIs, because of the need for connection of systems at different scales, taking advantage of the inherent characteristics of the SDIs: standard interfaces, standard metadata, and well-known specialized services.

The AWARE Application offers two important features compared with other environmental modelling systems discussed in Section 3. One is that it increases data accessibility because the AWARE Application is connected with other SDI nodes (geodata servers). The immediate consequence is that users can discover and access datasets and services as required, assuming that they support SDI's standards and interfaces. For example, wherever meteorological datasets may be placed, if such datasets are reachable through services interfaced according to OGC, users should be able to easily access, retrieve, and add their content to the AWARE Application for specific purposes. Another feature is that the proposed service-oriented architecture makes it possible to "wrap" and expose scientific tools and operations as distributed processing services described with standards-based service interfaces (see Section 5).

The AWARE Application provides an entry point to combine and integrate data discovery and collection tasks together with data processing services in the same application (see Section 6), what facilitates a streamlined running of environmental models in contrast to what traditionally have been disparate tasks carried out in an uncontrolled manner (Mineter et al., 2003).

5. Reusable geospatial services

The core of the AWARE architecture is the Service layer where the geospatial services reside. These services are based on the INSPIRE service types and provide users with the capacity of discovering, accessing and processing geospatial data as part of the application workflow, thus permitting execution of (hydrological) models in a distributed manner.

Several things should be taken into account when exposing scientific applications as distributed services. First, potential service types should be identified according to a well-established framework such as that promoted by INSPIRE. This helps developers to organize the spectrum of potential services since services of the same type normally share the same design process and interfaces. Second, in the design phase potential services' functionality within each type should be refined, taking into consideration design principles like reusability and efficiency in order to raise the level of service reusability without incurring excessive service management overhead and thus lowering performance. Finally, attention should be paid during the implementation phase, to choosing the most appropriate interface specification for the service in question and implementing the desired service functionality. Fig. 4 illustrates these three service creation steps that are described in detail in the following section.

5.1. AWARE service types

The top two rows in Fig. 4 show the service types stack highlighting the correspondence between INSPIRE and AWARE service types. AWARE services belong to AWARE service types which are derived directly from INSPIRE service types. The first row of Fig. 4 enumerates the service types as they are listed in the INSPIRE directive. The second row below shows the service types defined in the AWARE Application. With a few remarks we have followed the same classification of the INSPIRE service types.

5.2. Design principles

The AWARE project's overall goal is to provide hydrologists with distributed and reusable tools to monitor and predict the water availability, and secondarily to alleviate the need to maintain multiple generic desktop software packages for the purpose of a few occasional operations. The unstructured methodology of the hydrologists, using different desktop scientific tools, data and algorithms is migrated to a collection of standardized services accessible via a web-based entry point. There is a need to pay close attention to the design strategy for creating services according to specific hydrological model requirements, because services become the basic computing unit upon which other modules and components will rely.

5.2.1. Services granularity

Service design principles in SOA seek to minimize strong coupling to therefore help guarantee that services are self-contained, modular, extendable and reusable (Papazoglou and Yang, 2002). Creating services for specific application requirements implies the necessity to find the right level of granularity. Service granularity refers to the size of a service in terms of the amount of

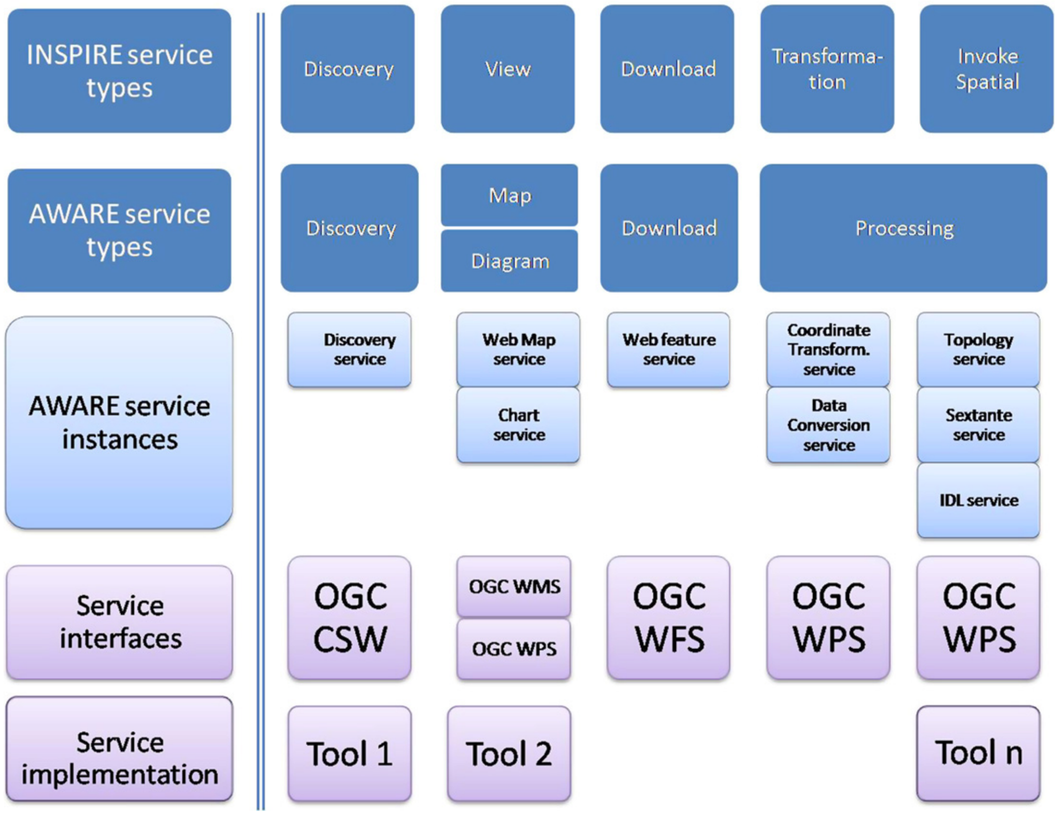

Fig. 4. Service perspectives: types, design and implementation. Service types is about INSPIRE vs. AWARE service types. Design is about granularity (instances) and interfaces. Implementation is about specific tools and programming.

functionality carried out (Haesen et al., 2008). Coarse-grained services encapsulate larger groupings of capability within a single interface, reducing the number of service requests from the client necessary to accomplish a task; however on the downside they might return excessive quantities of data, or it might be difficult to modify them to meet new requirements. Fine-grain services are those which perform specific tasks that are no longer decomposable in smaller pieces (subject to the application requirements). Small services normally require less complicated input and output data, meaning that they are more easily composed.

Having a small, stable set of coarse-grained services is often considered as best practice in designing services in SOA. However, we prefer to consider the spectrum of granularity levels, from coarse-grained to fine-grained services, in order to show how geospatial services at different granularity levels might have a positive impact on service reusability and performance. Finding the adequate granularity is a matter of balancing between multiple criteria (flexibility, modularity, reusability, and performance) to meet the ongoing needs in a specific application (Feuerlicht, 2006; Haesen et al., 2008). Sometimes these indicators are opposed as for example the binomial flexibility-performance. Coarse-grained services normally offer better performance however their flexibility decreases when adapting to new requirements. Creating fine-grained services that can be easily reused in other workflows is our goal but we must craft the right balance of fine-grained and coarse-grained services to meet the ongoing needs in our context.

Our strategy has followed a top-down methodology. Together with the hydrologist team, we have split the hydrological models into increasingly smaller pieces in order to identify the relevant processes. This recursive methodology continues until we encounter the desired level of granularity for the given processes. The criterion to stop the top-down decomposition approach is to consider a given process specific yet functional enough to not to be split again, that is, subsequent divisions would make no sense for the specific application requirements. The stop decision stems from a consensus between service designers and hydrologist experts. The resulting processes then become candidate processes for implementation as service processes within geospatial processing services. Suitable basic processes are those which perform a basic function (subject to application requirements) and can be potentially reused in other workflows. The ultimate goal is then to create a library of well-documented, stable geospatial services in which customized and elaborated functions (workflows) rely on other much more functionality-focused and well-tested services. In this case it makes sense to talk about fine-grained services in order to increase their reusability. This service design methodology is intimately related with the workflow tasks that have been developed in the Application layer to perform the service integration.

As described in Section 4, scientific workflows provide a composition mechanism in the Application layer to expose environmental models as a set of scientific workflows consisting internally as integrated chains of distributed geospatial services. To pursue the maximum reusability we have exposed as distributed processes all the basic tasks involved in the scientific workflows designed in the Application layer, so that these processes (or tasks) might be reused. However it is common to find chains of tasks that are called repeatedly along the workflows, which involve two or more of the basic functions mentioned before. A recurring practice in GIS application development and in SOA in general is to combine elementary operations into more complex tools in order to address specific user requirements. In these cases, repetitive chains of processes have been grouped forming a new process with larger

Table 1
Services and processes used in the AWARE application.

AWARE service	*Type*/specification	Service processes	Description
Catalogue	*Discovery*/OGC Catalogue Service for Web (CSW)	N/A	It offers the functionality to search and provide all earth observation data catalogued of the study areas in the AWARE project.
Web map	*View*/OGC Web Map Service (WMS)	N/A	It provides the user with some graphical maps of datasets over the study area.
Chart	*View*/OGC Web Processing Service (WPS)	Depletion Curves Plot Discharge Plot, HBV Runoff Plot, HBV SWE Plot, Sensor Data Chart	It provides diagrams (e.g. line plots) to represent some of the useful information, not as maps, but as descriptive plots showing some information in a graphical way.
Web Feature	*Download*/OGC Web Feature Service (WFS)	N/A	It provides users with some vector data (GML) over the study areas.
Coordinate transformation	*Processing*/OGC Web Processing Service (WPS)	TransCoordGMLPoint, TransCoordPoint, TransCoordPoint7P	It converts coordinates from a source reference system to a target one.
Data conversion	*Processing*/OGC Web Processing Service (WPS)	Shp2GML	It converts from shapefile format to GML format.
Topology	*Processing*/OGC Web Processing Service (WPS)	Area, Intersection, Buffer, MaxExtent, Snow Percentage, Get Feature By Attribute, Thiessen	Topological operations and interpolation algorithms.
Sextante	*Processing*/OGC Web Processing Service (WPS)	Coordinate Elevation, Stations Elevation, Elevation Curves, Elevation Zones, Hypsometric Elevation, Reclassify, Vectorize	Image processing algorithms, raster computations.
IDL	*Processing*/OGC Web Processing Service (WPS)	Snow Interpolation, Calibration, Simulation, K Coefficient Computation	It wraps polynomial interpolations and routines in IDL.

granularity and showing better performance. This strategy decreases development time and gains efficiency by avoiding unnecessary service calls and by minimizing data exchange over the network.

Returning to Fig. 4, the third row shows service instances generated after applying the service creation methodology described early. The processing service type includes most of the service instances since environmental models primarily deal with operations such as coordinate transformations, format/schemas transformations, spatial operations, algorithms execution, etc. that process data to produce meaningful information. Table 1 lists the geospatial services and their contained processes. Some of these services are discovery or download services yet most services offer some sort of processing capability. These are called geoprocessing services and each contains several processes with similar functionality. Most processes within the Topology geoprocessing service are fine-grained and thus are highly reusable as for example Area, Intersection, Buffer, and MaxExtent, which are concerned mainly with topological relations and geospatial proximity or distances among geospatial objects. Other fine-grained processes however are rarely reused in other scenarios because they are subject to specific application needs. Examples are the processes within the Chart geoprocessing service, almost entirely devoted to producing line plots and diagrams specific to the AWARE context. Fine-grained services with higher reusability levels are Classify, Vectorize, and Thiessen.

Fig. 5 depicts an example of processes within a scientific workflow. The Elevation Zones process contains an integrated chain invoking first the Reclassify process and then the Vectorize process, both processes taken from the Sextante geoprocessing service (see Table 1). SEXTANTE (Olaya, 2007) is a collection of geoprocessing routines developed by University of Extremadura (Spain), and available as free software. Although Reclassify and Vectorize processes expose well-known pieces of functionality and are independently reused in other scientific workflows along the hydrological models, in the given example in particular these processes are integrated forming a more coarse-grained service because the Elevation Zones process as a whole is actually called several times as part of different scientific workflows. The fact that a given service is reused several times justifies its level of granularity. In term of reusability, the more fine-grained a service is, the better. However, it is always recommended to use coarse-grained services for improving performance, so long as they are somewhat reusable. Both rules hold for the Elevation Zones use case. Furthermore, finding the right balance between service efficiency and reuse is often a subjective matter and depends on the specific

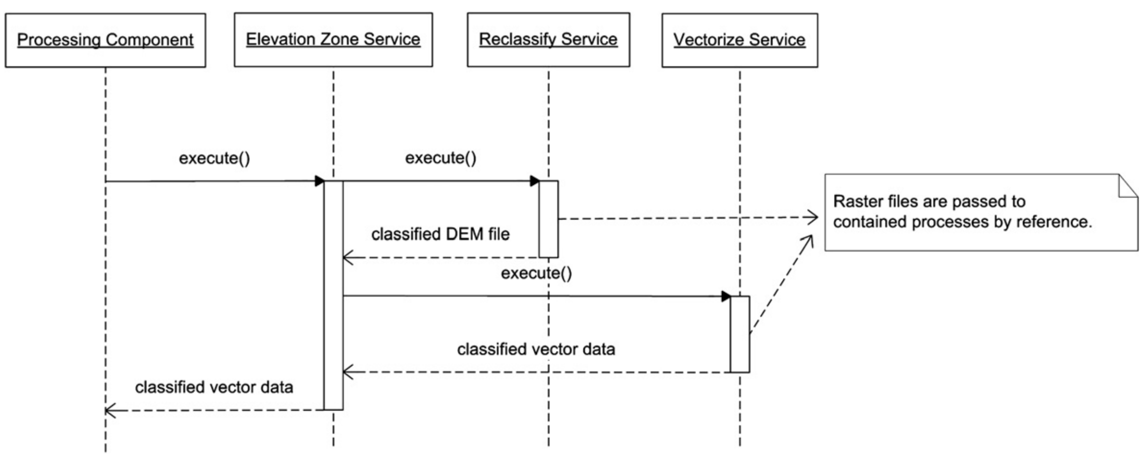

Fig. 5. A simplified sequence diagram for the Elevation Zones process.

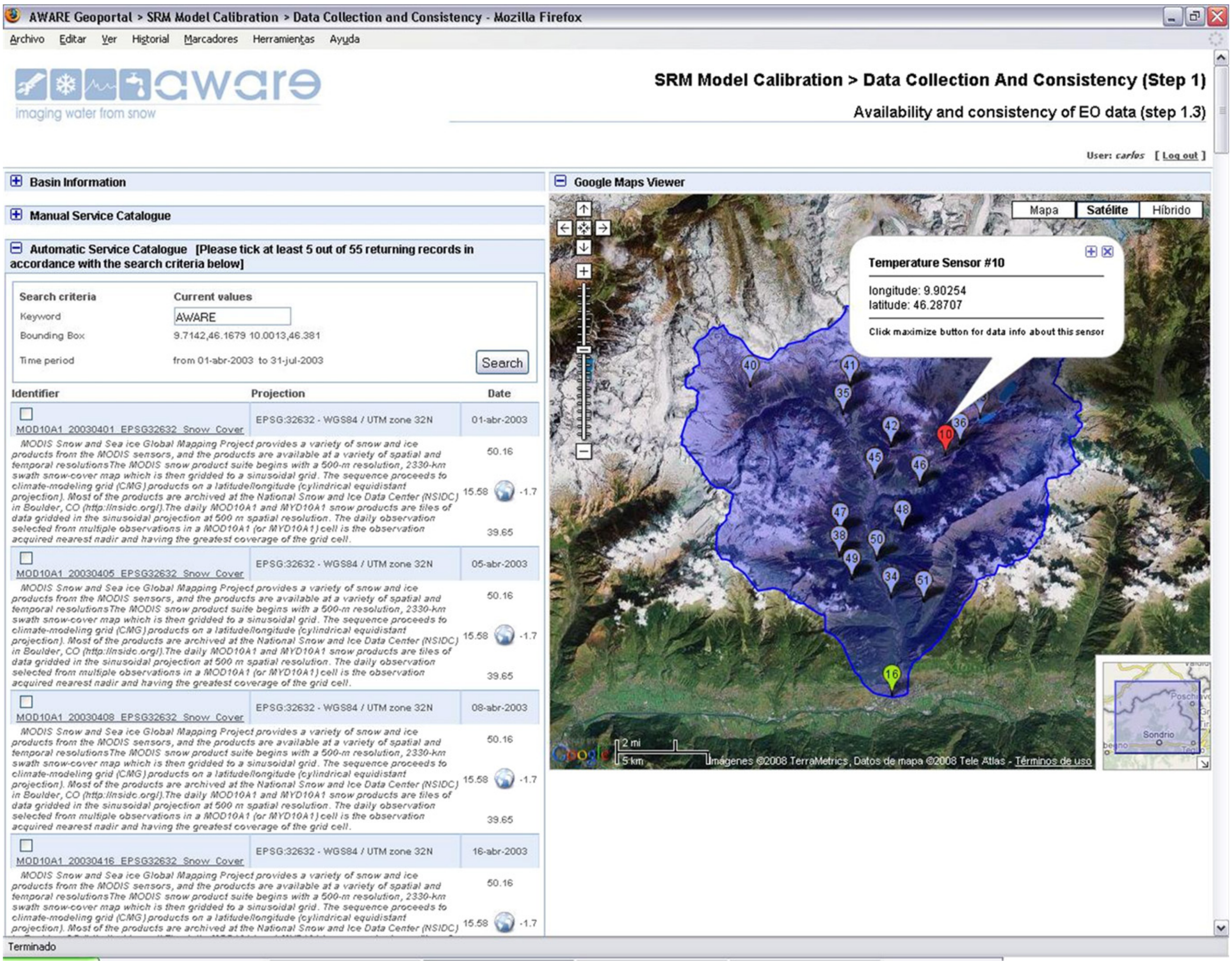

Fig. 6. User interface for retrieving metadata records from a catalogue service.

application requirements. A good practice, therefore, is to create services at multiple levels of granularity to be able to test the balance between the advantages of fine- and coarse-grained services.

In the given example, the client (the Processing component in the Service Client module) interacts with geospatial services independently of the level of granularity. It prepares the requests and processes results. The Elevation Zones process itself calls and manages the execution of the contained processes. In particular, the first process called is Reclassify, which traverses each DEM cell reading elevation values. According to the desired elevation ranges for the Elevation Zones, the process then assigns each cell to an elevation zone (500–1000 m, 1001–1500 m, etc.). Reclassify produces a classified raster file which is fed to the Vectorize process, which performs a common format transformation operation, converting the input raster file into the equivalent in vector polygons. The resulting vector file is encoded in GML format and delivered to the Elevation Zones process which sends it to the Processing component in the Application layer.

Friis-Christensen et al. (2007) discuss extensively about the challenges in designing service architectures for distributed geoprocessing. A first issue to be tackled, especially in the geospatial domain, is the optimal transport of data among the services of a chain, since huge amounts of geodata may take inordinate time to be transported over the network. A simple solution is to implement all the processes involved in the chain using the same geoprocessing service (and at the same server location) when possible. For example the Elevation Zones, Reclassify, and Vectorize processes belong to the Sextante geoprocessing service. In the given example, the DEM file is passed once in the initial request (Elevation Zones). Subsequent calls to the DEM file are local to the contained processes (e.g. Reclassify) so that data transport is greatly minimized. This approach maintains the desired service flexibility because the contained processes can be called independently, making service implementation easier and improving performance because service calls and data transport decrease notably.

5.3. Implementation principles

5.3.1. Standard interfaces

Currently most of the web services deployed in SDIs use interfaces defined by the OGC as part of the recent OWS Web Services specifications initiatives,[11] such as those described by (Moreno-Sanchez et al., 2007). The INSPIRE directive follows these

[11] http://www.opengeospatial.org/projects/initiatives/ows-4.

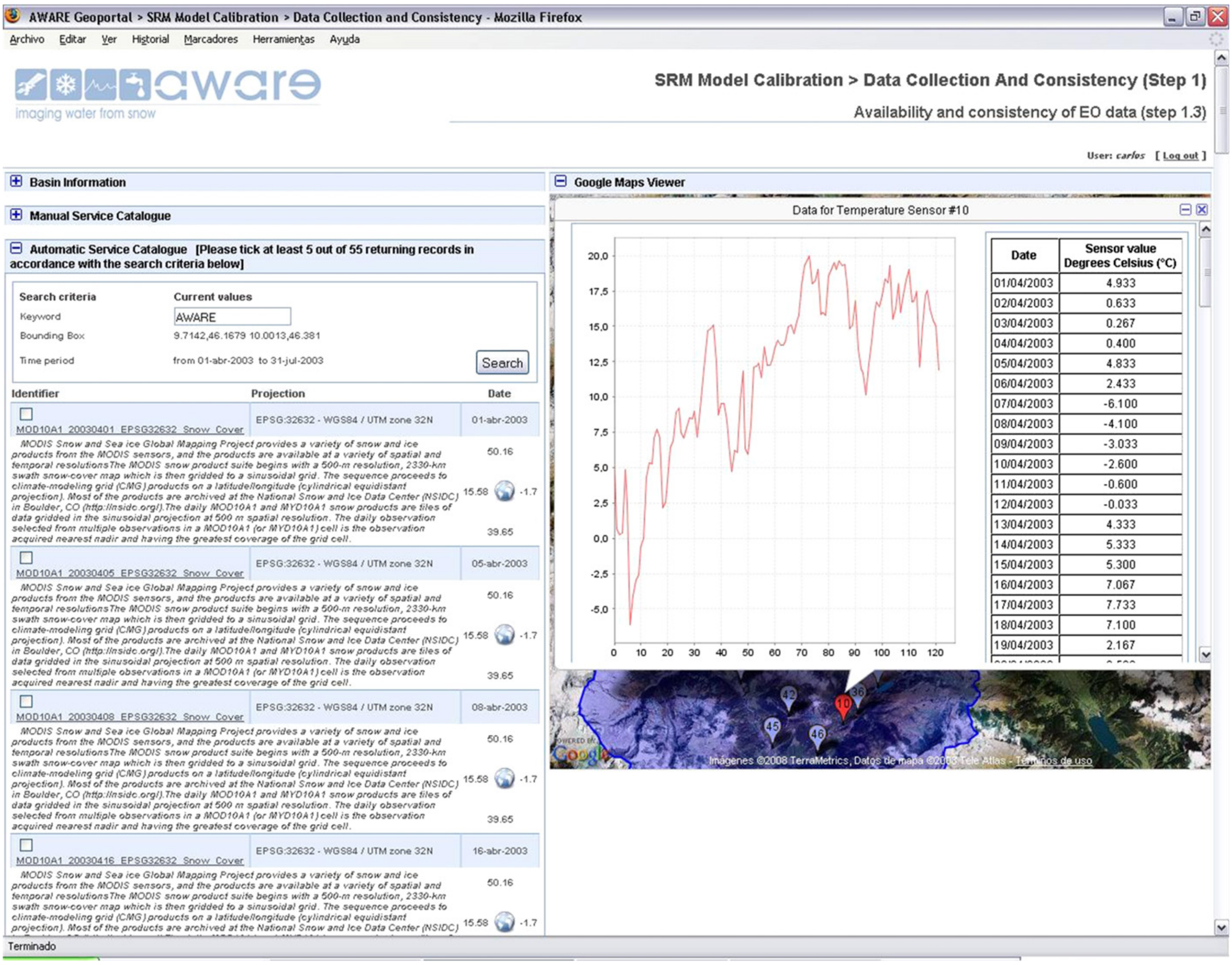

Fig. 7. User interface to inspect visually temperature sensor data.

same guidelines according to the first available implementation rules. The row labelled service interfaces in Fig. 4 represents the OGC specifications used for each service instance. It is important to note here that a given AWARE service type may be described by means of various OGC service interfaces, and vice versa, the same service interface may be used in various service types. This illustrates that many-to-many relations between services at the abstract level (type) and specifications at interface level are possible. For instance, we offer Web Map and Diagram service types, which correspond to the AWARE View services (at abstract level), but are interfaced with two distinct specifications: one provides maps via WMS interface and the other generates diagrams such as line plots via a WPS interface. Results are identical (images) however clearly with different semantics. Also, the same specification (WPS) may describe many service types. This demonstrates the flexibility of the WPS specification to allow wrapping of nearly any kind of process.

5.3.2. Wrapping strategy

Wrapping is a key part of service implementation. Once potential services and processes have been identified and designed following the design principles exposed in Section 5.2, we proceed to implement all of the processes by wrapping (encapsulating code in more easily readable forms) existing tools, when possible, as explained below.

Most desktop geospatial packages provide processing capabilities which can be migrated to geoprocessing services, thereby exposing well-tested GIS operations to web access as distributed services. However, many of these processing capabilities will have been designed and implemented by a software house for their own purposes, and so often these will not fit the needs of the concrete geospatial processing tasks of other user communities. This impediment is partially being addressed by an increasingly availability of diverse FOSS (Free and Open Source Software) projects, which permit users to more freely choose and mix those software tools that best fulfil their own requirements. FOSS projects, by the very nature of their licenses, may be modified and accommodated to suit concrete user needs. Given the wide spectrum of FOSS tools and libraries offering spatial functionality, the wrapping strategy reflects then the need of reusing (mostly) existing FOSS but also closed commercial tools, in order to wrap them as standard-conformant distributed service processes.

In our project, some FOSS tools have been reused merely to the extent of creating the service interface, leaving unchanged the original service implementation. In other cases modifications of the tool code has been necessary to adapt them to our needs. In the

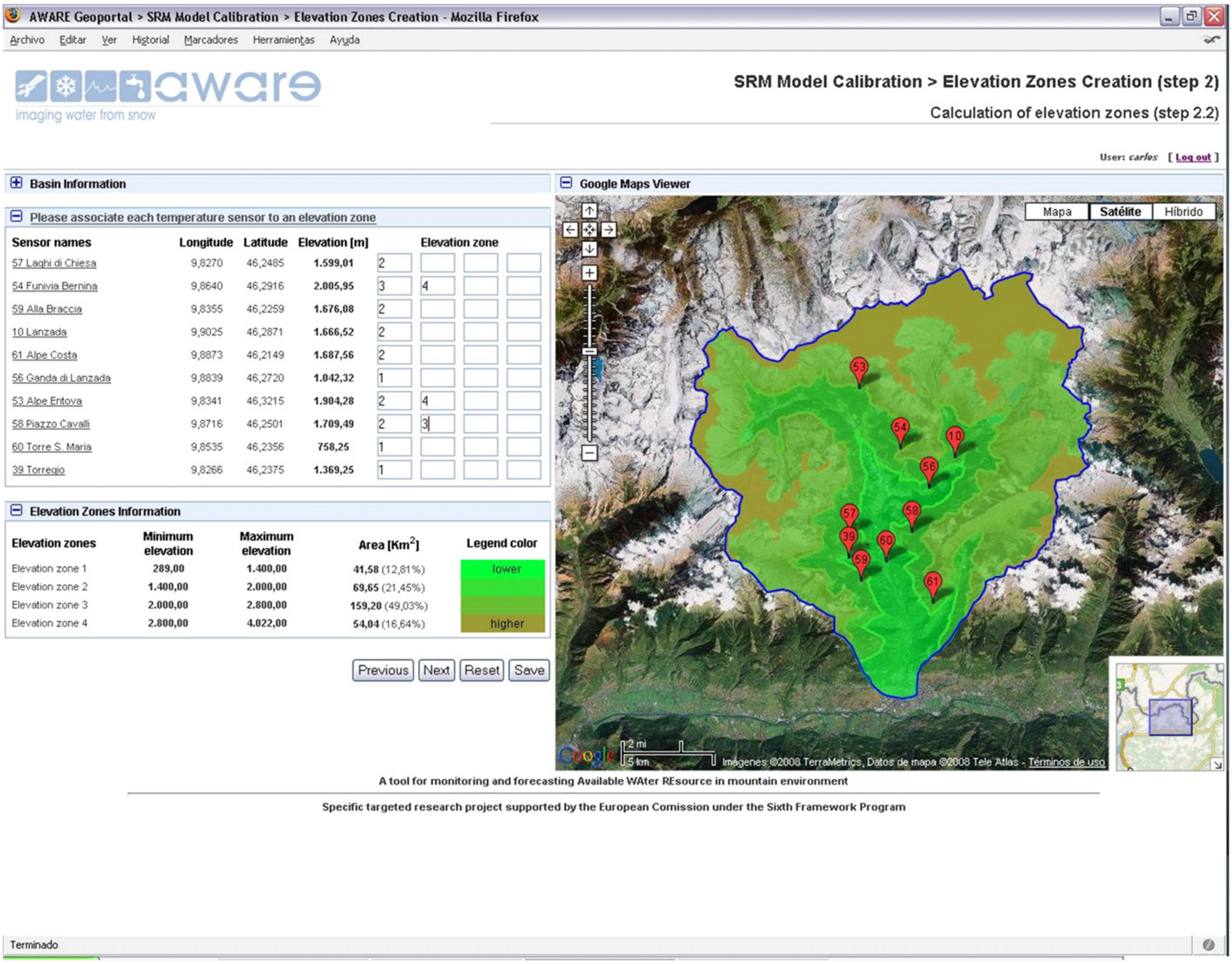

Fig. 8. User interface for displaying the results in workflow in Fig. 7.

cases where we needed functionality that was not already available; we have implemented it from scratch making use of available FOSS tools such as GeoTools,[12] gvSIG,[13] SEXTANTE,[14] JFreeChart,[15] etc.

We have used an open source implementation of the OGC WPS specification developed by the German initiative 52North[16] that provides a framework for exposing our processes as WPS-encapsulated processes. To illustrate with an example, the GeoTools library supports the implementation of the processes concerned with topological tasks involving geometric area and intersection. JFreeChart library, which provides an API to create charts and line diagrams, has been integrated in the Chart geoprocessing service to deliver the diagram functionality required in our application. In the same way the SEXTANTE library provides a set of more than two hundred raster and vector analysis operations, some of them required in our application, as in the cases of Reclassify and Vectorize. Some of SEXTANTE's analysis operations are used in the implementation of the Sextante geoprocessing service, applying some adapter patterns to expose SEXTANTE functionality as distributed web processes (Díaz et al., 2008b).

In other cases our scientists possessed scientific routines already implemented in software modules (e.g. interpolation routines) using specific IDL and Fortran libraries which were not suitable for easy migration to distributed web environments. These legacy routines were then wrapped using dynamic libraries and Java bridges to expose the embedded functionality as processing services. As the granularity of these processes was already given, the service design phase was unnecessary, requiring only implementation efforts to adapt these processes as geospatial services.

6. AWARE Application

This section provides an overview of the main characteristics of the AWARE Application described in the earlier sections through a selected set of figures that capture relevant steps during the calibration phase of use of the SRM model. The watershed of study is the Mallero river basin (319 km^2) located in the Italian Alps, one of the test watersheds of the AWARE project. This basin has been studied for 3 melting seasons 2002, 2003 and 2004 using ground measurements collected from ARPA Lombardia and also snow cover maps resulting from the processing of MODIS satellite data. Rather

[12] http://geotools.codehaus.org.
[13] http://www.gvsig.org.
[14] http://www.sextantegis.com.
[15] http://www.jfree.org/jfreechart.
[16] http://52north.org.

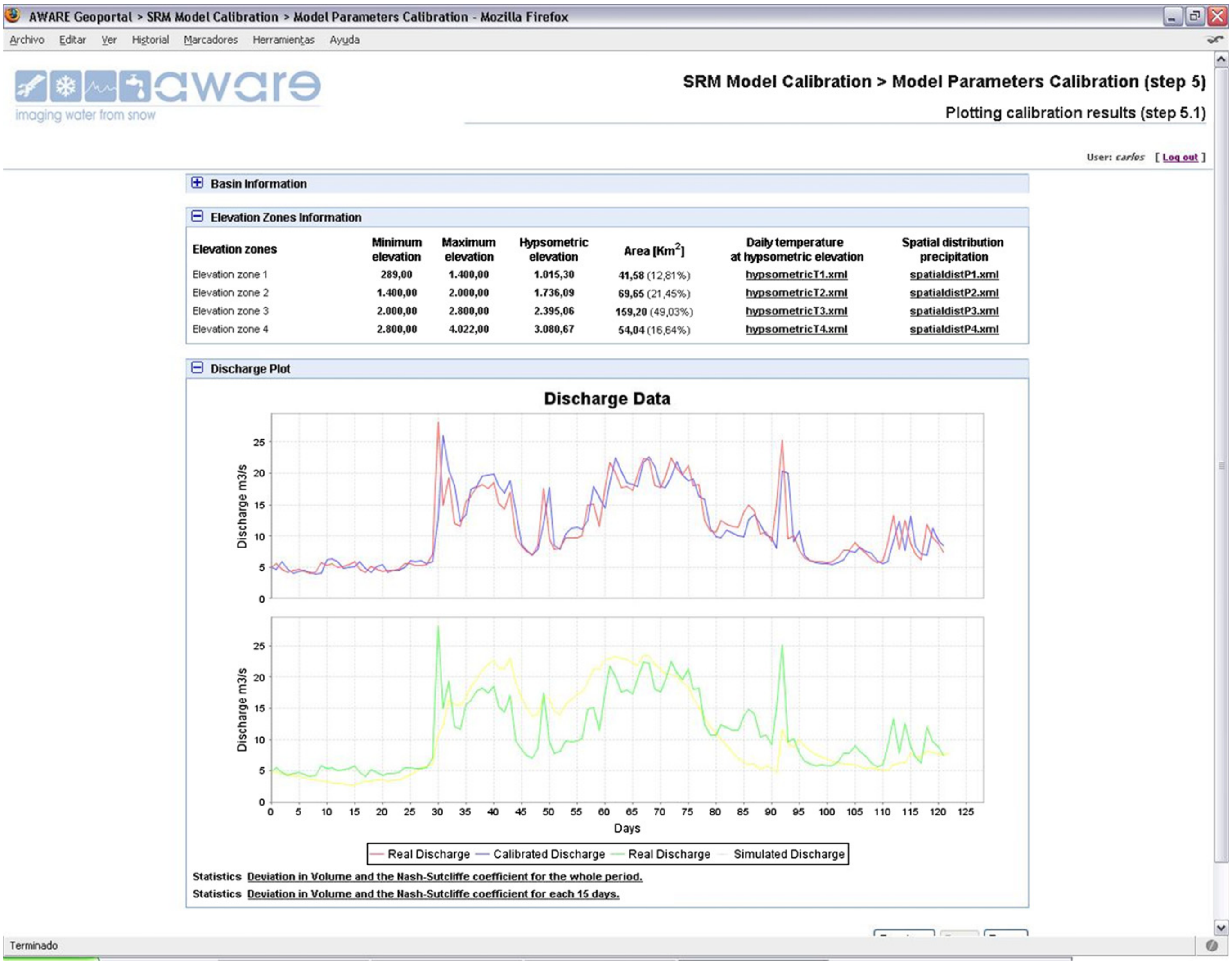

Fig. 9. User interface with plots comparing calibrated, real and simulated discharges.

than focusing on the underlying technology used to assemble the user interface, this section demonstrates the overall viability of the AWARE Application, built upon a distributed set of interoperable geospatial services in the Mallero basin scenario. Readers interested in how client technologies (JavaScript, AJAX, XSLT, etc.) have been interrelated (mashed-up) to compose the user interface may refer to Granell et al. (2008).

Fig. 6 shows the AWARE Application interface for the first step of the SRM model, in which initial model input data are collected and validated. The AWARE Application integrates a discovery service that connects to a metadata catalogue (GeoNetwork[17]) containing descriptions of satellite imagery products. The Discovery component in the Service Client module (see Section 4) offers searching capabilities by transparently building automatic queries against the catalogue service, in order to retrieve datasets relevant to user-defined context parameters gathered in the previous steps of the current user session. Certain user-defined context properties that are transformed into constraints at search time are for example the relevant time period for data, the geographic extent (bounding box) and default data description keywords (e.g. basin name). If results are not acceptable, users can retry the search by refining the search parameters. Otherwise, as the Fig. 6 depicts, for each metadata record retrieved multiple fields are displayed, such as the title, projection, date, bounding box, etc., allowing the user to select the best data available for input to the model. Fig. 6 also shows how the tasks of discovering, visualizing, and collecting data are unified in the same web page by combining simple service requests: users can recover the full metadata record in XML format by clicking on the record identifier link (this provokes a *getRecordById* query, discovery service), view the map by clicking on the globe icon (via *getMap* query, view service), and also download the vector data (GML) representing the snow-covered areas for the current satellite image (via *getFeature* request, download service). Note that complex queries are necessarily managed by the Discovery component in the Service Client module when for example various searching constraints are concatenated with logical operators (AND, OR, etc.) in the same query and, when the number of hits is high it becomes necessary to manage catalogue service interactions so that the metadata records are delivered in fixed-size sets in order to minimize network traffic.

The right side of Fig. 6 shows the map viewer based on the Google Maps API, displaying, in this case, the basin extent (in blue) together with the location of meteorological sensors (icons). Each icon is numbered with the sensor identifier and classified according to the fill colour (red-temperature; blue-precipitation, and green-

[17] http://geonetwork-opensource.org.

stream gauge). As explained previously, the right-hand map shown in Fig. 6 is displayed in the Presentation layer by accessing View services, while the data displayed on the map has been processed and integrated by the modules and services devoted to data and service integration in the Application and Service layers. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

Fig. 7 shows an example of the possible formats of maps offered by the AWARE Application for data visualization and exploration (Wood et al., 2007), a useful feature in environmental applications. Users may click on a sensor icon to obtain more information about the sensor and its data. This action is executed on the server side via requests to the Chart geoprocessing service, one of the AWARE View services in the Service layer, handled by the Processing component in the Service Client module. This service provides the capability of displaying data in a graphical way according to requirements of the hydrologists. In this sense, the browser remains simple (used mostly for user interaction and for data entry) rather than becoming the integration platform for complex, interactive web client applications. Again, both the plot and raw table displayed in Fig. 7 are processed on the server side, in this case forwarding the results for rendering as HTML code by the browser.

Sections 4 and 5, through the calculation of Elevation Zones process example (see Fig. 3), illustrate how the conjunction of modules and components in the Application layer together with geospatial services in the Service layer allows complex workflows by integrating remote geospatial services at different granularity levels. Tedious tasks such as raster analysis computations, spatial intersections and coordinates transformations are executed transparently to users, even integrating the workflow results with the Map Viewer module. The Mashup Server component takes the resulting Elevation Zones in GML (workflow's results in Fig. 3) and performs convenient transformations (coordinates, format, etc.) in order to generate the results illustrated in Fig. 8.

Fig. 9 shows the calibration results according to the SRM model's requirements. In this case, data are presented to users in the form of line plots to facilitate the interpretation of model results. The top line plot compares the calibrated and measured stream discharge, whereas the lines plot just below displays the measured against the simulated discharge. It is worth remembering that such lines plots are generated by executing distributed processes within the Chart geoprocessing service (see Section 5).

7. Conclusions

We have presented an overview of the architecture of the AWARE Application, a service-oriented application allowing hydrologists and geoscience researchers in general, to access geospatial data and services available in SDIs. The application, accessed via a Geoportal, guides expert users stepwise through the calibration and running of hydrological models by orchestrating and remotely executing a set of distributed geospatial services. The AWARE Application offers the capability to users, on one hand, of discovering data and services and, on the other hand, accessing and invoking, among others, view and processing services to successfully prepare and run the hydrological model, thus saving time, money and effort on arriving at conclusions to support environmental decision-making.

Geospatial data repositories that use interfaces and protocols different than OGC standards cannot easily be integrated into our application. This is a limitation and needs further research to improve interoperability. However, ongoing promising EU projects such as GIGAS[18] are attempting to address these challenges by promoting the coherent, uniform adoption of standards, protocols, and service interfaces among various international initiatives at different scale such as GMES, INSPIRE and GEOSS (Global Earth Observation System of Systems).

This paper also discusses the principles for creating an open architecture capable of adapting to service-orientation. The layered AWARE architecture adapts the principles proposed in the SDI/INSPIRE technical architecture to establish an open, interoperable architecture based on standard interfaces and reusable services. The set of services described here has been designed taking into account a trade-off among granularity, modularity and reuse principles. These services have been publicly exposed using standard OGC interfaces, permitting their access not only via the AWARE Application but also directly from other remote users or SDI-based applications. This strategy leads to an open library of geospatial processing services potentially reusable in other thematic scenarios, providing added value to the scientific community, especially for collaborative research teams.

As Mineter et al. (2003) envisioned a few years ago, the future of environmental applications is the conjunction of new technologies in a distributed environment together with data visualization techniques based on 2-D maps. Continuing that thought, current tendencies for geosciences disciplines go beyond static, 2D maps and are looking toward multi-dimensional virtual globes, which are gaining wide acceptance for scientific research and collaboration among other areas (Tuttle et al., 2008). Craglia et al. (2008) have proposed research challenges for producing the next generation of virtual globes to improve applications in many global domains but especially highlighting the environmental domain. Multiple interconnected virtual globes will allow geoscience researchers to connect and combine their data to jointly study the same phenomena from different perspectives, to search through time and space, and to continuous monitor how the state of the Earth in environmental scenarios changes to increase understanding of dynamic Earth processes. Virtual globes will possibly become in the next service integration and data visualization platform for geosciences disciplines though, as Craglia et al. (2008) point out, many research issues need to be tackled in the following years. We believe that our paper makes a modest contribution to that future goal.

Acknowledgements

This work has been partially supported by the AWARE project SST4-2004-012257 under the EU GMES initiative. Institut Cartogràfic de Catalunya (ICC, Spain) assisted in the design of the AWARE Application in its initial stage, and Istituto per il Rilevamento Elettromagnetico dell'Ambiente (IREA, Italy) provided some data services and the catalogue service detailed here; both were partners of the AWARE project. We would also like to thank the ARPA Lombardia, Dipartimeto di Sondrio, Italy which kindly granted for demonstration purposes the hydro-meteorological data of the Mallero river basin, used in this project. Finally, thanks to the anonymous reviewers for their valuable comments.

References

van der Aalst, W.M.P., van der Hee, K.M., 2002. Workflow Management: Models, Methods, and Systems. MIT Press, Cambridge.

van der Aalst, W.M.P., Beisiegel, M., van der Hee, K.M., Konig, D., Christian Stahl, C., 2007. An SOA-based architecture framework. International Journal of Business Process Integration and Management 2 (2), 91–101.

Agarwal, S., Handschuh, S., Staab, S., 2004. Annotation, composition and invocation of semantic web services. Journal of Web Semantics 2 (1), 31–48.

Alameh, N., 2003. Chaining geographic information web services. IEEE Internet Computing 7 (5), 22–29.

Alonso, G., Casati, F., Harumi, K., Machiraju, V., 2004. Web Services. Concepts, Architectures and Applications. Springer, Heidelberg.

[18] http://www.thegigasforum.eu.

Bernard, L., Kanellopoulos, I., Annoni, A., Smits, P., 2005. The European geoportal – one step towards the establishment of a European Spatial Data Infrastructure. Computers, Environment and Urban Systems 29 (1), 15–31.

Berners-Lee, T., Hall, W., Hendler, J., Shadbolt, N., Weitzner, D.J., 2006. Creating a science of the web. Science 313, 769–771.

Best, B.D., Halpin, P.N., Fujioka, E., Read, A.J., Qian, S.S., Hazen, L.J., Schick, R.S., 2007. Geospatial web services within a scientific workflow: predicting marine mammal habitats in a dynamic environment. Ecological Informatics 2 (3), 210–223.

Chang, Y.-S., Park, H.-D., 2006. XML web service-based development model for Internet GIS applications. International Journal of Geographical Information Science 20 (4), 371–399.

Chow, T.E., 2008. The potential of maps APIs for Internet GIS applications. Transactions in GIS 12 (2), 179–191.

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., Weerawarana, S., 2002. Unravelling the web services web: an introduction to SOAP, WSDL, and UDDI. IEEE Internet Computing 6 (2), 86–93.

Craglia, M., Goodchild, M.F., Annoni, A., Camara, G., Gould, M., Kuhn, W., Mark, D., Masser, I., Maguire, D., Liang, S., Parsons, E., 2008. Next-Generation Digital Earth. A position paper from the Vespucci initiative for the advancement of geographic information science (editorial). International Journal of Spatial Data Infrastructures Research 3, 146–167.

Denzer, R., 2005. Generic integration of environment decision support systems – state-of-the-art. Environmental Modelling & Software 20 (10), 1217–1223.

Díaz, L., Granell, C., Gould, M., 2008a. Case study: geospatial processing services for web-based hydrological applications. In: Sample, J.T., Shaw, K., Tu, S., Abdelguerfi, M. (Eds.), Geospatial Services and Applications for the Internet. Springer, New York, pp. 31–47.

Díaz, L., Granell, C., Gould, M., Olaya, V., 2008b. An open service network for geospatial data processing. In: Academic Proceedings of the 2008 Free and Open Source Software for Geospatial Conference (FOSS4G 2008), pp. 410–419. Cape Town, South Africa.

Dustdar, S., Schreiner, W., 2005. A survey on web services composition. International Journal of Grid and Web Services 1 (1), 1–30.

Feuerlicht, G., 2006. Service granularity considerations based on data properties of interface parameters. Computer Systems Science and Engineering 21 (4), 315–320.

Foster, I., 2005. Service-oriented science. Science 308 (5723), 814–817.

Foster, I., Kesselman, C., Tuecke, S., 2001. The anatomy of the grid: enabling scalable virtual organisations. International Journal of Supercomputer Applications 15, 200–222.

Friis-Christensen, A., Ostländer, N., Lutz, M., Bernard, L., 2007. Designing service architectures for distributed geoprocessing: challenges and future directions. Transactions in GIS 11 (6), 799–818.

Goodall, J.L., Horsburgh, J.S., Whiteaker, T.L., Maidment, D.R., Zaslavsky, I., 2008. A first approach to web services for the national water information system. Environmental Modelling & Software 23 (3), 304–401.

Granell, C., Diaz, L., Gould, M., 2007. Managing earth observation data with distributed geoprocessing services. In: Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS 2007). IEEE CS, pp. 4777–4780.

Granell, C., Díaz, L., Gould, M., 2008. Geospatial web service integration and mashups for water resource applicationsProceedings of the XXI Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS 2008). International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVII Part B4, 661–666.

Haesen, R., Snoeck, M., Lemahie, W., Poelmans, S., 2008. On the definition of service granularity and its architectural impacts. In: Proc. of International Conference on Advanced Information Systems Engineering (CAiSE'08). LNCS, vol. 5078. Springer, pp. 375–389.

Hey, T., Trefethen, A.E., 2005. Cyberinfrastructure for e-Science. Science 308, 817–821.

INSPIRE EU Directive, 25 April 2007. Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). Official Journal of the European Union, L 108/1 50. http://eur-lex.europa.eu/JOHtml.do?uri=OJ:L:2007:108:SOM: EN:HTML.

Jeong, S., Liang, Y., Liang, X., 2006. Design of an integrated data retrieval, analysis, and visualization system: application in the hydrology domain. Environmental Modelling & Software 21 (12), 1722–1740.

Kiehle, C., 2006. Business logic for geoprocessing of distributed geodata. Computers & Geosciences 32 (10), 1746–1757.

Kiehle, C., Greve, K., Heier, C., 2006. Standardized geoprocessing – taking spatial data infrastructure one step further. In: Proc. of the AGILE Conference on Geographic Information Science. University of West Hungary.

Ko, I.-Y., Neches, R., 2003. Composing web services for large-scale tasks. IEEE Internet Computing 7 (5), 52–59.

Lacasta, J., Nogueras-Iso, J., Béjar, R., Muro-Medrano, P.R., Zarazaga-Soria, F.J., 2007. A web ontology service to facilitate interoperability within a spatial data infrastructure: applicability to discovery. Data & Knowledge Engineering 63 (3), 947–971.

Lemmens, R., Wytzisk, A., de By, R., Granell, C., Gould, M., van Oosterom, P., 2006. Integrating semantic and syntactic descriptions to chain geographic services. IEEE Internet Computing 10 (5), 42–52.

Liu, Y., Gupta, H., Springer, E., Wagener, T., 2008. Linking science with environmental decision making: experiences from an integrated modeling approach to supporting sustainable water resources management. Environmental Modelling & Software 23 (7), 846–858.

Lu, C.-T., Dos Santos Jr., R.F., Sripada, L.N., Kou, Y., 2007. Advances in GML for geospatial applications. Geoinformatica 11 (1), 131–157.

Lutz, M., 2007. Ontology-based descriptions for semantic discovery and composition of geoprocessing services. GeoInformatica 11 (1), 1–36.

Maguire, D.J., Longley, P.A., 2005. The emergence of geoportals and their role in spatial data infrastructures. Computers, Environment and Urban Systems 29 (1), 3–14.

Martinec, J., Rango, A., Roberts, R., 1994. The Snowmelt Runoff Model (SRM) user's manual. In: Baumgartner, M.F. (Ed.), Geographica Bernensia, P 29. Department of Geography, Univ. of Berne, Berne, Switzerland. http://hydrolab.arsusda.gov/~rroberts/srmmanual/srmman.html.

Masser, I., 2005. GIS Worlds: Creating Spatial Data Infrastructures. ESRI Press, Redlands.

McColl, C., Aggett, G., 2007. Land-use forecasting and hydrologic model integration for improved land-use decision support. Journal of Environmental Management 84 (4), 494–512.

McIlraith, S.A., Son, T.C., Zeng, H., 2001. Semantic web services. IEEE Intelligent Systems 16 (2), 46–53.

Michaelis, C.D., Ames, D.P., 2009. Evaluation and implementation of the OGC web processing service for use in client-side GIS. GeoInformatica 13 (1), 109–120.

Milanovic, N., Malek, M., 2004. Current solutions for web service composition. IEEE Internet Computing 8 (6), 51–59.

Mineter, M.J., Jarvis, C.H., Dowers, S., 2003. From stand-alone programs towards grid-aware services and components: a case study in agricultural modelling with interpolated climate data. Environmental Modelling & Software 18 (4), 379–391.

Moreno-Sanchez, R., Anderson, G., 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. International Journal of Geographical Information Science 21 (10), 1135–1163.

Olaya, V., 2007. SEXTANTE: a gvSIG-based platform for geographical analysis. In: Proc. of the Free and Open Source Software for Geospatial conference (FOSS4G) Victoria, Canada.

Papazoglou, M., van den Heuvel, W.-J., 2007. Service oriented architectures: approaches, technologies and research issues. The VLDB Journal 16 (3), 389–415.

Parajka, J., Merz, R., Blöschl, G., 2005. A comparison of regionalisation methods for catchment model parameters. Hydrology and Earth System Sciences 9, 157–171.

Pecar-Ilic, J., Ruzic, I., 2006. Application of GIS and web technologies for Danube waterway data management in Croatia. Environmental Modelling & Software 21 (11), 1562–1571.

Petrie, C., Bussler, C., 2008. The myth of open web services: the rise of the service parks. IEEE Internet Computing 12 (3), 94–96.

Portele, C., 2007. OpenGIS geographic markup language (GML) encoding standard. http://www.opengeospatial.org/standards/gml (accessed 30.04.08).

Papazoglou, M.P., Yang, J., 2002. Design methodology for web services and business processes. In: Proc. of Workshop on Technologies for E-Services (TES 2002). LNCS, vol. 2444. Springer, pp. 54–64.

Reitsma, F., Albrecht, J., 2005. Modeling with the semantic web in the geosciences. IEEE Intelligent Systems 20 (2), 86–88.

Sheth, A.P., 1999. Changing focus on interoperability in information systems: from system, syntax, structure to semantics. In: Goodchild, M., Egenhofer, M., Fegeas, R., Kottman, C. (Eds.), Interoperating Geographic Information Systems. Kluwer Publisher, Norwell, pp. 5–30.

Shneiderman, B., 2007. Web science: a provocative invitation to computer science. Communication of ACM 50 (6), 25–27.

Schudt, P. (Ed.), 2007. OpenGIS Web Processing Service Version 1.0.0. Open Geospatial Consortium. http://www.opengeospatial.org/standards/wps (accessed 04.09.08).

Smits, P.C., Friis-Christensen, A., 2007. Resource discovery in a European spatial data infrastructure. IEEE Transactions on Knowledge and Data Engineering 19 (1), 85–95.

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.

Teng, J., Vaze, J., Tuteja, N.K., Gallant, J.C., 2008. A GIS-based tool for spatial and distributed hydrological modelling: CLASS spatial analyst. Transactions in GIS 12 (2), 209–225.

Tuttle, B.T., Anderson, S., Huff, R., 2008. Virtual globes: an overview of their history, uses, and future challenges. Geography Compass 2 (5), 1478–1505.

Wood, J., Dykes, J., Slingsby, A., Clarke, K., 2007. Interactive visual exploration of a large spatio-temporal dataset: reflections on a geovisualition mashup. IEEE Transactions on Visualization and Computer Graphics 13 (6), 1176–1183.

Yang, C., Li, W., Xie, J., Zhou, B., 2008. Distributed geospatial information processing: sharing distributed geospatial resources to support Digital Earth. International Journal of Digital Earth 1 (3), 259–278.

Yue, P., Di, L., Yang, W., Yu, G., Zhao, P., 2007. Semantics-based automatic composition of geospatial web service chains. Computers & Geosciences 33 (5), 649–665.