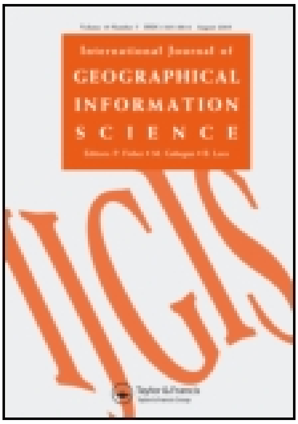

International Journal of Geographical Information Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tgis20

An optimized framework for seamlessly integrating OGC Web Services to support geospatial sciences

Zhenlong Li [a], Chaowei Phil Yang [b], Huayi Wu [a], Wenwen Li [c] & Lizhi Miao [a]

[a] Center for Intelligent Spatial Computing, George Mason University, Fairfax, VA, USA

[b] Joint Center for Intelligent Spatial Computing, George Mason University, Fairfax, VA, USA

[c] Department of Earth System and Geoinformation Science, George Mason University, Fairfax, VA, USA

Available online: 29 Oct 2010

To cite this article: Zhenlong Li, Chaowei Phil Yang, Huayi Wu, Wenwen Li & Lizhi Miao (2011): An optimized framework for seamlessly integrating OGC Web Services to support geospatial sciences, International Journal of Geographical Information Science, 25:4, 595-613

To link to this article: http://dx.doi.org/10.1080/13658816.2010.484811

International Journal of Geographical Information Science
Vol. 25, No. 4, April 2011, 595–613

An optimized framework for seamlessly integrating OGC Web Services to support geospatial sciences

Zhenlong Li[a], Chaowei Phil Yang[b]*, Huayi Wu[a], Wenwen Li[c] and Lizhi Miao[a]

[a]*Center for Intelligent Spatial Computing, George Mason University, Fairfax, VA, USA;* [b]*Joint Center for Intelligent Spatial Computing, George Mason University, Fairfax, VA, USA;* [c]*Department of Earth System and Geoinformation Science, George Mason University, Fairfax, VA, USA*

(*Received 6 November 2009; final version received 28 March 2010*)

OGC Web Services (OWS) are essential building blocks for the national and global spatial data infrastructure (NSDI and GSDI) and the geospatial cyberinfrastructure (GCI). Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), and Catalogue Service for Web (CSW) have been increasingly adopted to serve scientific data. Interoperable services can facilitate the integration of different scientific applications by searching, finding, and utilizing the large number of scientific data and Web services. However, these services are widely dispersed and hard to be found and utilized with acceptable performance. This is especially true when developing a science application to seamlessly integrate multiple geographically dispersed services. Focusing on the integration of distributed OWS resources, we propose a layer-based service-oriented integration framework and relevant optimization technologies to search and utilize relevant resources. Specifically, (1) an AJAX (Asynchronous JAvaScript and eXtensible Markup Language)-based synchronous multi-catalogue search is proposed and utilized to enhance the multi-catalogue searching performance; (2) a layer-based search engine with spatial, temporal, and performance criteria is proposed and used for identifying better services; (3) a service capabilities clearinghouse (SCCH) is proposed and developed to address the service issues identified by a statistical experiment. A science application of data correlation analysis is used as an example to demonstrate the performance enhancement of the proposed framework.

Keywords: spatial data infrastructure; geospatial cyberinfrastructure; performance; distributed geographic information processing; spatial computing; OGC Web Services; spatial web portal; interoperability; geoportal

1. Introduction

Over the past decade, we have witnessed the increasing importance of Web services. Web services are widely used to deploy business-to-business scenarios that crosscut company boundaries. Web services are geographically dispersed and are used by a wide array of companies and government organizations because of the reusability, flexibility, and platform independence (Shen *et al*. 2007). The geospatial sciences are data-intensive domains, where research and development typically produces and analyzes large volumes of distributed heterogeneous geospatial data sets. Subsequently, these domains are also experiencing an increased need for both computational power and quantity of information and to make large

*Corresponding author. Email: cyang3@gmu.edu

ISSN 1365-8816 print/ISSN 1362-3087 online

DOI: 10.1080/13658816.2010.484811
http://www.informaworld.com

spatial data available over the Internet (Vaccari *et al.* 2009). For example, the Terra instruments (http://terra.nasa.gov/) generate about 195 gigabytes of level 0 data (raw data at full instrument resolution) daily and can reach 850 terabytes when processed to higher level science products (Parkinson *et al.* 2006). The data can be accessed online through a variety of Web-based applications such as Earth Observing System (EOS) Data Gateway (EDG). The sharing of large volume of data sets encourages researchers and organizations to pay attention to consensus development of standard protocols and tools to publish and interoperate these large volumes of data sets (Vaccari *et al.* 2009).

The Open Geospatial Consortium (OGC) has gained momentum with thousands of adopters of OGC Web Services (OWS) since the OGC Web Map Service (WMS) specification published in 1999 (Kralidis 2007). WMS is the standard in providing simple HyperText Transfer Protocol (HTTP) interfaces to serve geo-referenced mapped images over the Internet (Beaujardiere 2004). WMS has been increasingly adopted to provide the interoperable integration of different Web applications from a large number of data within various scientific domains. A remarkable number of government organizations, research institutions, and private companies have deployed WMS servers. For example, National Aeronautics and Space Administration (NASA) Earth Observations (NEO) established a time-enabled WMS server to serve the massive amounts of monthly and daily satellite data related to climate and environmental changes (Ward 2009). Private companies, such as Environmental Systems Research Institute (ESRI) and Intergraph, developed interoperable interfaces into their software products and set up their own WMS servers. A WMS server list maintained by Skylab Mobilesystems Ltd. shows that there were 994 WMSs and 339,254 layers available in the Internet in March 2009 (Skylab 2009). According to the service status checker maintained by the Federal Geographic Data Committee (FGDC, http://registry.fgdc.gov/statuschecker/wmsResultsReport.php), 13,681 WMSs were registered with the Geospatial One Stop (GOS, http://gos2.geodata.gov/wps/portal/gos) by September 2009.

To leverage the success of WMS and to support direct data sharing through the Web, the OGC published the Web Feature Service (WFS) and Web Coverage Service (WCS). WFS provides an interface for describing data manipulation operations for geospatial features using HTTP (Vretanos 2005). WCS defines a standard interface that enables users to access geospatial coverage, such as satellite images, digital aerial photos, and other phenomena represented by values at each measurement point (Whiteside and Evans 2008). WFS and WCS have also been adopted by many geographic information system (GIS) vendors, such as ESRI, to support interoperability in their products, such as ArcGIS Desktop 9.3.

Furthermore, numerous Open Source projects have implemented OWS to provide both server and client functions, such as GeoServer (http://geoserver.org), MapServer (http://mapserver.org/), and OpenLayers (http://openlayers.org/). By December 2009, 462 compliant/implementing products have been registered in the OGC implementing products Web page (http://www.opengeospatial.org/resource/products/implementing). WMS, WFS, and WCS are widely supported by these products. Driven by the Open Source projects, major GIS vendors, and the demand for sharing and interoperating heterogeneous geospatial data, more and more OWSs are developed and become available over the Internet.

The increasing number of OWSs provides a great opportunity and also many challenges for geospatial sciences. Because generic searching engines, such as Google or Yahoo, have not been developed to support the geospatial characteristics of geospatial services, geospatial catalogues are utilized to catalogue and enable users to search information across multiple servers (Kottman 1999, Nogueras-Iso *et al.* 2005). OGC Catalogue Service for Web (CSW) (Nebert *et al.* 2007) catalogues have become a major mechanism for cataloguing Web-based geospatial data. Government agencies and research institutes, such as

NASA, National Oceanic and Atmospheric Administration (NOAA), and United States Geological Survey (USGS), have built their own CSW services to support the cataloguing and discovery of geospatial data and services. However, each CSW has its own focused field and many CSWs have yet to be integrated to support comprehensive geospatial applications. For example, the CSW of Virtual Arctic Spatial Data Infrastructure (VASDI) is focused on the arctic geospatial region. The Air Quality Community CSW is focusing on air quality in the United States. If scientists need to make an air quality application using high-resolution satellite data, they may need to dig into several different CSW services to find the best resources. Therefore, the first challenge is to integrate multiple catalogues to search and discover data and services for an application in an acceptable manner.

OWS was proposed as a solution to the integration problem (Alonso *et al*. 2004). By selecting reliable services based on the service performance and then integrating them seamlessly, it can help solve complicated problems (Nikola and Miroslaw 2004). Many initiatives, such as geospatial cyberinfrastructure (CI) (GCI, Brodaric *et al*. 2009, Wang and Liu 2009, Yang *et al*. 2010) and Digital Earth (Yang *et al*. 2008) aim to utilize the interoperable OWSs and other services to foster the integration of heterogeneous geospatial information, Web-based mapping, and geo-analytical services across the Internet (Zhang *et al*. 2006) to support geospatial sciences. Besides OWS, another popular Web service is the Tiled Map Service (TMS) which provides a simple HTTP interface to serve the tiled maps of geo-referenced data. The tiled maps could be pre-rendered and pre-cut at fixed scales on the TMS server or be compiled on-the-fly from a shapefile or a spatial database; therefore, TMS provides a high-performance method for obtaining two-dimensional raster (e.g., coverage or grid) data and pre-rendered vector (e.g., Shapefile) data. Though no official standard is available for TMS yet, Open Source Geospatial Foundation (OSGeo) has released a non-official TMS specification that standardizes the way in which map tiles are requested by clients and the ways that servers describe their tiled maps (http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification). During the past decade, TMS has been widely used in various online maps, such as Google Maps (http://maps.google.com/maps), Microsoft Bing Maps (http://www.bing.com/maps/), and Yahoo Maps (http://maps.yahoo.com/). It is another challenge to determine how to integrate, evaluate, and utilize the TMS effectively and efficiently.

Over the past two decades we witnessed the growth of spatial data infrastructures (SDIs) for support of different domains and different geographic regions from global SDI (GSDI), national SDI (NSDI), to local SDI (LSDI) (Maguire and Longley 2005, Yang *et al*. 2005, Vaccari *et al*. 2009). The booming of the SDIs is primarily driven by the sharing of heterogeneous and distributed geospatial information across the Internet (Yang and Raskin 2009). OWSs are acting as the essential building blocks for these SDIs, as OWSs made transparent the differences of heterogeneous information through a set of international standards. Therefore, a third challenge is to enable the integrated OWS to be interoperable by designing a Service-Oriented Architecture (SOA) framework.

To handle the challenges and to optimize the integration performance on service discovery, integration, and utilization, we propose a framework that includes the following components and technologies: (1) an AJAX (Asynchronous JAvaScript and eXtensible Markup Language)-based synchronous multi-catalogue search mechanism to enhance the searching performance; (2) a service capabilities clearinghouse (SCCH) to enable seamless and high-performance integration of thousands of OWSs; (3) an SOA to make the framework reusable and interoperable by deploying layer-based CSW services to serve the integrated services through a standard interface. As TMS is also based on HTTP, the technologies described in this article are also suitable for TMS when it becomes more complicated (i.e., return metadata), such as the GetCapabilities request provided in the non-official TMS specification.

2. Related work

A number of specifications and languages have been developed to support generic service integration. For example, Web Service Conversation Language (WSCL) allows the abstract interfaces of Web services (http://www.w3.org/TR/wscl10/). WSCL focuses on the modeling of the sequence of interactions between Web services (Wang *et al.* 2004) and is an early initiative toward the Web services integration. The Web Services Flow Language (WSFL) proposed by IBM describes how Web services may be composed into new Web services to support business processes (Leymann 2001). The Business Process Execution Language for Web Services (BPEL4WS) specification is proposed as the Web services standard for composition (Sanjiva and Francisco 2002). The Web Services Choreography Interface (WSCI), another W3C specification from BEA Systems, Intalio, SAP, and Sun Microsystems, is an eXtensible Markup Language (XML)-based interface description language that describes the flow of messages exchanged by a Web service participating in choreographed interactions with other services (http://www.w3.org/TR/wsci/). All these approaches focus on the high-level service integration, composition, or communication among services and are useful for the integration of the services from different domains. However, they are not flexible or detailed enough when referring to specific OWS, such as WMS.

For geospatial Web services, Thakkar *et al.* (2002) proposed a mediator-based integration framework to dynamically integrate geospatial Web services as information sources and to compose new Web services. Yue *et al.* (2007) proposed a semantics-based automatic composition strategy to integrate geospatial Web services based on the SOA. This strategy allows the automatic composition of geospatial Web services based on geospatial semantics. Lutz *et al.* (2007) proposed a rule-based description framework for the composition of geospatial information services, which was based on enhanced ontologies. The semantic-based integration concept can be leveraged to improve the intelligence of the integrated system (Oren *et al.* 2008). These approaches are valuable when dealing with the integration and chaining of various types of geospatial services. To seamlessly integrate a large number of specific OWS, such as tens of thousands of WMS services, these approaches are not suitable.

For geospatial information discovery, Ramroop and Pascoe (2001) proposed a virtual Geo-resource Centre to collect and store distributed geospatial resource metadata by using Lightweight Directory Access Protocol (LDAP). In this Geo-resource Centre, a selector broker is used to process metadata queries in the search for appropriate data sets. When users conduct searches against the virtual Geo-resource Centre instead of the distributed geospatial resources, the selector broker will use the pre-defined semantic and syntactic mappings to translate the initial searches. The various geospatial metadata standards are made transparent by such a Geo-resource Centre. Similarly, Chen and Mohapatra (2005) proposed a service broker that gathers all the requests from distributed servers and intelligently processes them. Lee *et al.* (2006) proposed a tool to generate an adapter for the integration of Web services, where users can access several Web services like a single Web service through the adapter because the differences among the services are integrated by the adapter. Actually, it has been proved that the brokering and centralizing techniques may provide an efficient search when discovering multiple catalogues or databases (Laurini 1998, Mena *et al.* 2000, Ramroop and Pascoe 2001, Koponen and Virtanen 2004). By adapting the centralizing concept, we propose the SCCH to enable seamless and high-performance integration of thousands of OWSs. By adapting the brokering concept and focusing on the CSW services, we use a CSW adapter to facilitate the AJAX-based synchronous multi-catalogue search.

Based on the above analysis, we propose a performance-improved framework to seamlessly integrate OWSs. The framework is implemented in one of our spatial Web portal (Yang *et al*. 2007a) prototypes. This article details the framework. Section 3 introduces the proposed framework. Sections 4 and 5 discuss the methods used in the framework and the relevant experimentations. Section 6 describes a spatial Web portal prototype based on the integration framework to demonstrate how the optimized framework can be adapted for application integration.

3. Framework

To address the challenges and maintain the flexibility of the spatial Web portal prototype, we designed the framework in layers including integration layer, interoperation layer, and presentation layer (Figure 1).

Integration layer is the core layer of this framework. It consists of two components: discovery client and centralizing server. For the discovery client, we propose an AJAX-based synchronous multi-catalogue search mechanism to enable users to search multiple catalogues synchronously (Section 4.1). The client is connected to a *CSW adapter (*Figure 2). The centralizing server is responsible for integrating thousands of OWSs seamlessly. To implement the centralizing server, we propose a *SCCH* to harvest, pre-process, and cache the service capability information (Section 4.2).

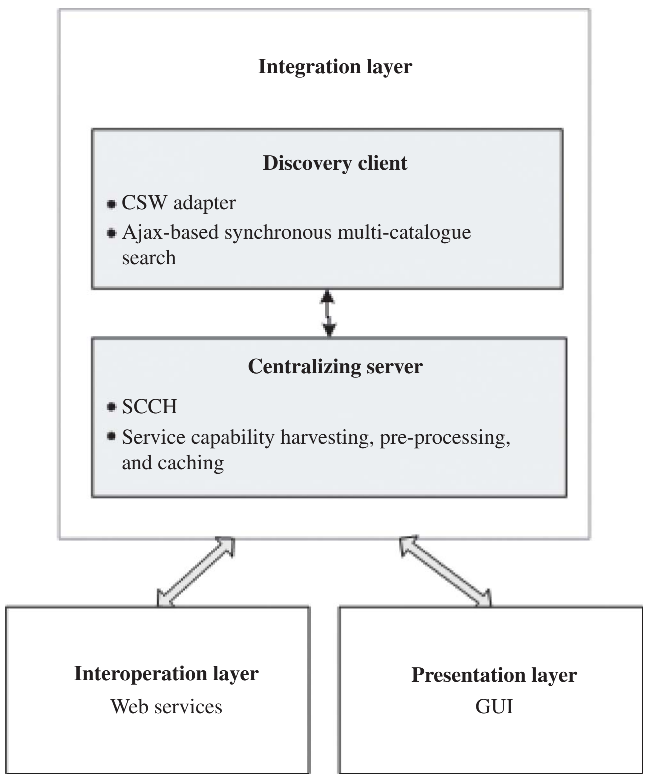

Figure 1. A framework to seamlessly integrate OWS.

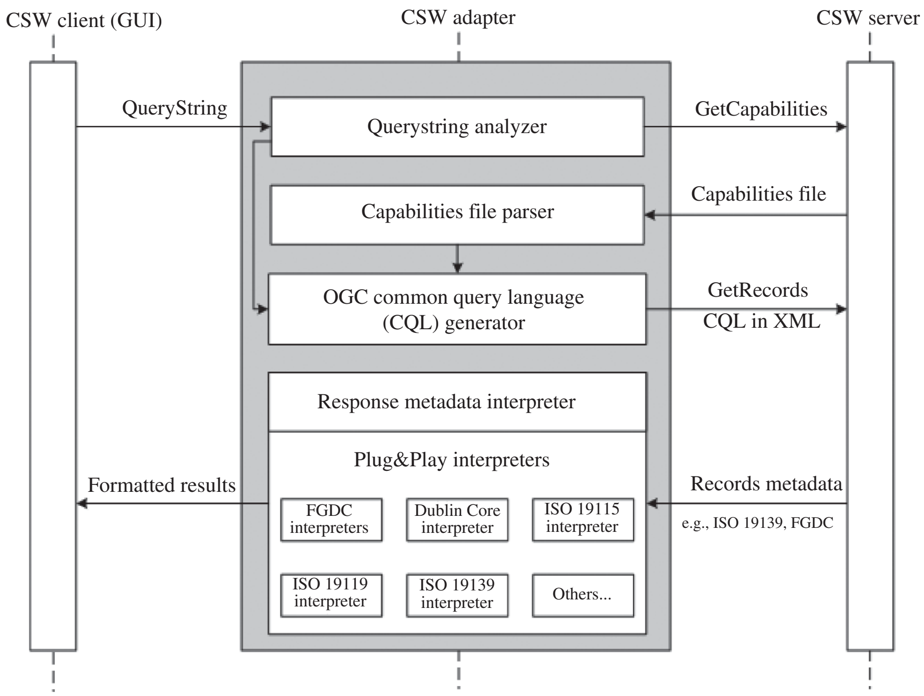

Figure 2. *CSW adapter* architecture and its communication sequence with the CSW client and server.

Interoperation layer provides a layer-based CSW and application programming interface (API) (Section 4.3) to enable other applications' access to this framework through a standard interface.

Presentation layer provides various functional components, such as visualization, time dimensional analysis, and map composition. Meanwhile, an interactive graphic user interface (GUI) is provided for user interactions.

This article will focus on the integration layer and interoperation layer. The AJAX-based synchronous multi-catalogue search mechanism, SCCH, and layer-based search are detailed in Sections 4 and 5.

4. Methods

4.1. AJAX-based synchronous multi-catalogue search

When searching from multiple catalogue services, the traditional approach is to send the request, receive and process the response sequentially. Therefore, the time spent increases linearly with the increased number of servers. For example, if we search from *N* number of CSW services at one time, the total processing time is

$$T_{\text{waiting}} = T_1 + T_2 + T_3 + \cdots + T_n \qquad (1)$$

where T_{waiting} is the total time spent when all *N* requests return a response. T_n is the response time of the *N*th service.

Multithreading can be used to handle problems related to multiple and concurrent requests (Solomon and Rankings 1997, Yang *et al*. 2005, Yang *et al*. 2007b). By adapting the multithread principle, N requests to N servers can be sent simultaneously and can be processed synchronously. Because these N servers are independent logically and even physically, the total time is determined only by the longest response time of these N servers:

$$T_{\text{waiting}} = \text{Max}\{T_1, T_2, T_3, \ldots, T_n\} \tag{2}$$

where T_{waiting} is total time consumed when all N requests return a response. T_n is the response time of the Nth service.

Equation (2) is only true when T is longer than the resulting overhead which increases as the number of connections increases. Actually, when T is shorter than the resulting overhead, techniques to reduce the waiting time is not urgent. Therefore, when T is much longer than the resulting overhead, the total waiting time can be dramatically reduced by applying the multithread principle. The response times of these N servers may be different because of the number of discovered records as well as the server performance; therefore, T_{waiting} in Equation (2) may be very long. To further reduce users' waiting time, AJAX is adapted to refresh the result page immediately once any of the N servers returns records. The waiting time can be denoted as

$$T_{\text{waiting}} = \text{Min}\{T_1, T_2, T_3, \ldots, T_n\} \tag{3}$$

where T_{waiting} is the time before users get the first record. T_n is the response time of the Nth service.

To implement the AJAX-based multi-catalogue search, we need to handle multiple catalogue services in a single client. CSW interface contains two important operations: GetCapabilities and GetRecords (Nebert *et al*. 2007). The GetCapabilities operation retrieves service metadata describing a server's capabilities, such as the supported output schemas, result types, and spatial comparison operators. The GetRecords operation conducts a search based on the search parameters and returns the results as the primary method of resource discovery. A GetRecords request is complicated because of the following: (1) the query string, often encoded as XML, usually contains various filters, spatial operators, and time constraints; (2) the response is also XML encoded and needs to be interpreted; and (3) different CSW services from different vendors may support different query filters, output schemas, and application profiles because of the coexistence of multiple profiles in CSW specification. For example, both ebRIM (Electronic business Registry Information Model) and ISO (International Organization for Standardization) profiles are supported by CSW. These issues make it harder to dynamically support multi-CSW without changing the program.

To address this problem, a *CSW adapter* (Figure 2) is proposed to deal with the differences among various CSW services. Its communication sequence includes the following: (1) the CSW client collects user's input (such as CSW server URL, searching keywords, time span, and geographic region) and submits a search request to a *CSW adapter*; (2) the *QueryString analyzer* extracts the parameters and sends a GetCapabilities request to CSW servers; (3) the *Capabilities file parser* interprets the query filters and response schemas supported by this CSW server; (4) the *Common query language (CQL) generator* then uses this information and query parameters to build the CQL that satisfies this CSW server and sends the request; and (5) the *Response metadata interpreter* receives the response and selects a proper metadata interpreter to interpret and format the results. For example, FGDC interpreter will be selected if a CSW server returns FGDC standard metadata.

The *CQL generator* is used to integrate the different query filters supported by different CSWs based on service capabilities information. The *Response metadata interpreter* is used to handle different response formats because of the different application profiles. All the metadata interpreters can plug-and-play to make the *CSW adapter* flexible and scalable. By using the *CSW adapter*, the CSW client can provide an intuitive and interactive GUI to collect user's input and display the formatted results. The differences of different CSW services are encapsulated and made transparent to others by the *CSW adapter*.

Based on the *CSW adapter*, we adopted an architecture pictured in Figure 3, where *AJAX-based request/response controller* is responsible for (1) sending requests to the distributed CSW servers simultaneously and (2) displaying results dynamically after getting a response from a server.

4.2. Service capabilities clearinghouse

The AJAX-based multi-CSW search mechanism provides an efficient and effective way to locate desired geospatial resources. Once the resources are located, how to integrate them efficiently is another challenge. OWSs provide the GetCapabilities operation to allow users to retrieve the metadata of the targeted service. After parsing the server's metadata, other standard operations (such as GetMap for WMS, GetFeature for WFS, and GetCoverage for WCS) can be invoked to retrieve the data. As a result, the typical workflow for a OWS client is (1) invoke GetCapabilities request to download the service capabilities in XML; (2) interpret the XML file to extract the capabilities information; (3) construct the query string and invoke other operations to fetch the real data, and finally (4) utilize the returned maps, features, and data in applications. When accessing thousands of services, steps 1 and 2 have serious performance issues including (1) time consumed in downloading the capabilities file and (2) time consumed in interpreting large capabilities files on-the-fly.

As WMS is the most widely used OGC service, we use it as an example. We experimented to get 1074 WMS response times of GetCapabilities request for each service (Figure 4) and found that 698 of 1074 (65%) WMS services failed to return the capabilities

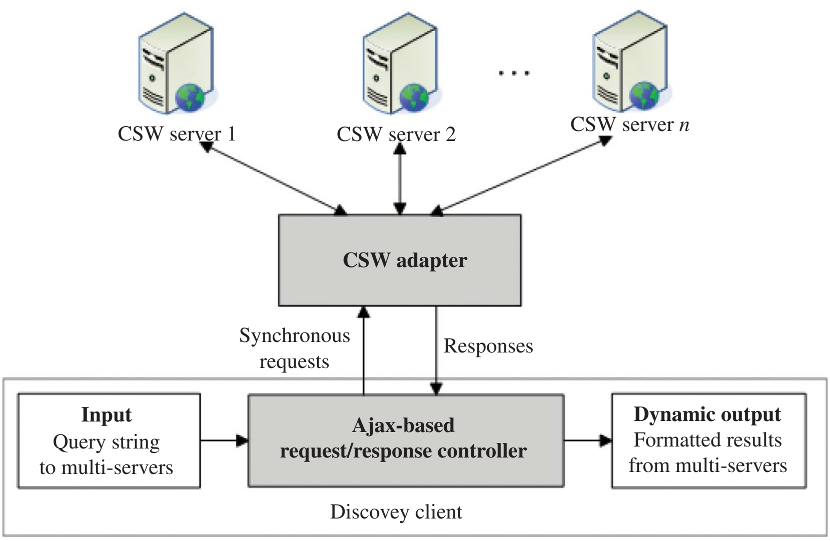

Figure 3. Architecture of AJAX-based synchronous multi-catalogue search.

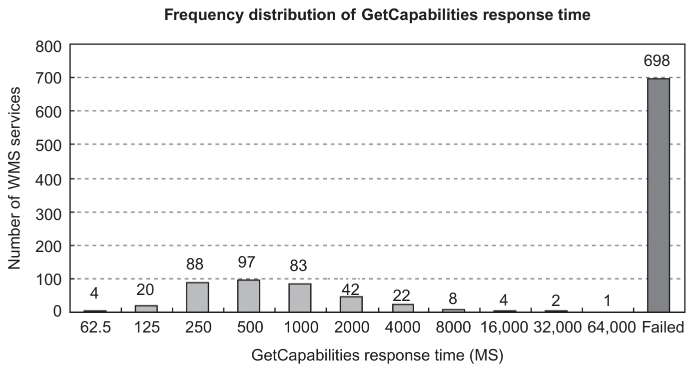

Figure 4. Frequency distribution of GetCapabilities response times of 1074 WMS services.

files within 2 min and some never responded. Therefore, it will be very helpful to display the service's status and to give users a quicker response when the same service is accessed. For those 376 services that returned responses within a minute, there were 84 services (22%) that had a response time longer than 2 s and 167 services (44%) that had a response time longer than 1 s. For one or several WMSs, response time in seconds is not an issue to the performance; however, when dealing with the integration of thousands of services, this becomes a serious problem that can reduce the system performance dramatically.

Caching data on the client side can reduce the load on network transmission and server processing (Yang *et al*. 2005) and improve the overall system performance. Instead of sending a GetCapabilities request and interpreting a response every time a user accesses a service, we can cache the interpreted information (such as version, abstract, keywords, geographic region, and reference system) on a centralizing server once the first GetCapabilities request is invoked. However, two issues occur when trying to cache the capabilities file: (1) the service capabilities file may change in an irregular manner and (2) the cached data may be so overwhelming that it will overflow the centralizing server. The first issue can be addressed by re-downloading and re-interpreting the capabilities file on a designed schedule. For the second issue, another experiment was conducted to test the size of capabilities files. Figure 5 shows the frequency distribution of the size of capabilities files of the 376 WMS services; most of the files (76%) are smaller than 20 KB and only four are bigger than 1 MB. Based on this experiment, the centralizing server can easily handle tens of thousands of capabilities files.

We propose a *SCCH* to accelerate the service integration and access based on the experiments (Figure 6). The major goal of SCCH is to harvest and pre-process the service capabilities files and to cache the processed information in a database. SCCH works as a centralizing service broker to cache service metadata on a centralizing server. Therefore, the cached information can be accessed not only by the user invoking the operation, but also by everyone connecting to SCCH. OWSs are harvested by SCCH from CSW search results, user input, and service crawler (Li *et al.* 2010): (a) the *capabilities file parser* is responsible for extracting the layer-based metadata and store them in the service layer repository; (b) the *capabilities updater* is running in the background to update the service metadata by reprocessing the capabilities information on a pre-defined frequency (e.g., daily) to keep them consistent with providers; (c) the *layer-based performance evaluator* is responsible for testing and recording the response time for each layer. APIs are provided for other components to retrieve information from the SCCH.

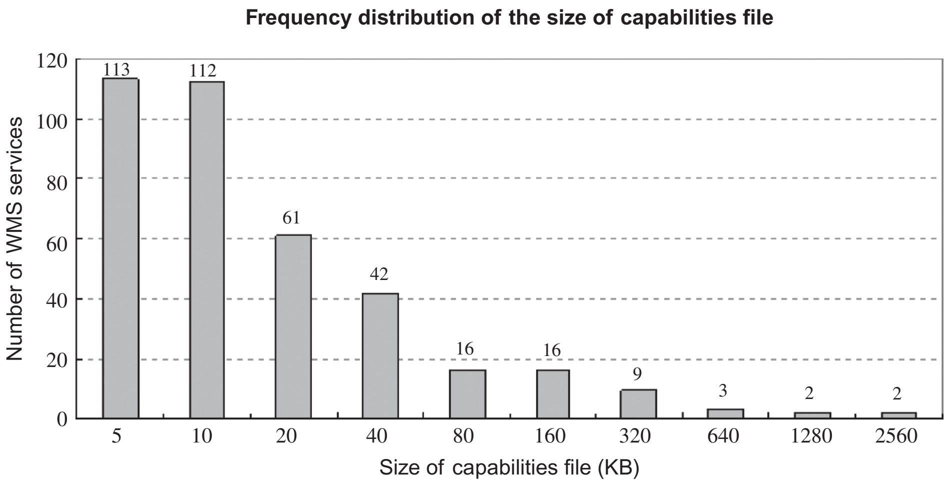

Figure 5. Frequency distribution of the Capabilities files' sizes of the 376 WMSs.

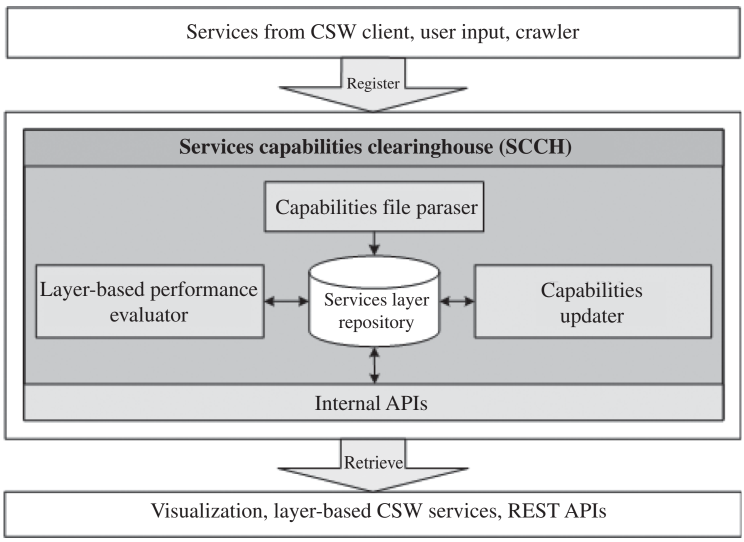

Figure 6. Architecture and workflow of SCCH.

When a service metadata is requested, SCCH will search the requested records in its database and respond to the users immediately if the service is already registered to reduce the waiting time. If the requested service is not in SCCH, SCCH will automatically harvest this service and return the requested records, in which case it needs a little more time for downloading and interpreting capabilities files on-the-fly. SCCH automatically collects services from users' requests, which makes it a self-growing system.

4.3. Layer-based search

A common characteristic of OWSs (including TMS) is that their service unit is layered, which means every service contains one or more layers. When users access a service, they

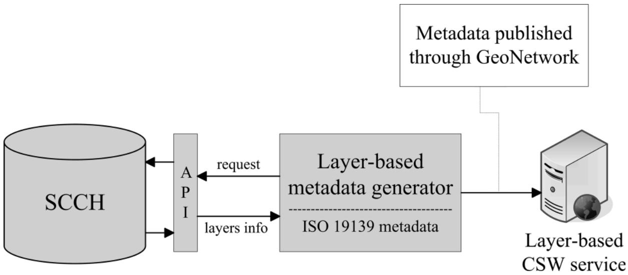

Figure 7. Workflow for transforming layers' meta-information to layer-based CSW service.

are accessing layers from the service; therefore, it is helpful to allow users to search against layers directly based on the layers' keywords, geographic extent, and so on. Furthermore, layers that provide the same geo-information from different services have different performance; selecting the best performance layers is extremely useful for layer aggregation.

SCCH acts as a layer repository and has detailed meta-information for each layer; this meta-information is adequate to build the query string to fetch real data from the server. Based on such a scenario, we suggest a layer-based search API to provide functions that can search the layer information (such as name, title, keywords, reference system, and geographic extent) from SCCH. This layer-based search API not only enhances the search efficiency but also makes SCCH interoperable.

OGC CSW provides a standard interface to allow users to discover geospatial information. To further improve the interoperability of SCCH, we established a layer-based catalogue service using CSW standards by (1) converting each layer's meta-information to a standard metadata record encoded in XML and (2) publishing the converted metadata records as a CSW service. Figure 7 illustrates the workflow and architecture of the transformation from layers' meta-information to CSW. The *layer-based metadata generator* is responsible for collecting layers' information from SCCH through API and then generating standard metadata in XML. ISO 19139 is selected because it is widely used to describe, validate, and exchange geospatial metadata. Each resulting ISO 19139 metadata contains a layer's meta-information that is extracted from the Capabilities document, such as the layer name, bounding box, keywords, reference system, and abstract. Then, the resulting ISO 19139 metadata files can be ingested and published as a CSW service by a geospatial catalogue server, such as GeoNetwork (http://geonetwork-opensource.org/), an open-source software which provides metadata editing, publishing, and search functions. After the service is published, the layers that are registered in SCCH can be searched directly in a CSW client.

5. Results

Three experiments were conducted to test the performance of the AJAX-based synchronous multi-catalogue search, SCCH, and layer-based search.

To test the performance of AJAX-based multi-CSW search, an experiment with the same search criteria to compare the user waiting time of a multi-CSW search was performed (Figure 8). It was found that multithreading reduced the waiting time from 12 s (Equation 1) to 7.5 s (Equation 2) and further reduced to 1.5 s (Equation 3) with AJAX technologies, including multithreading.

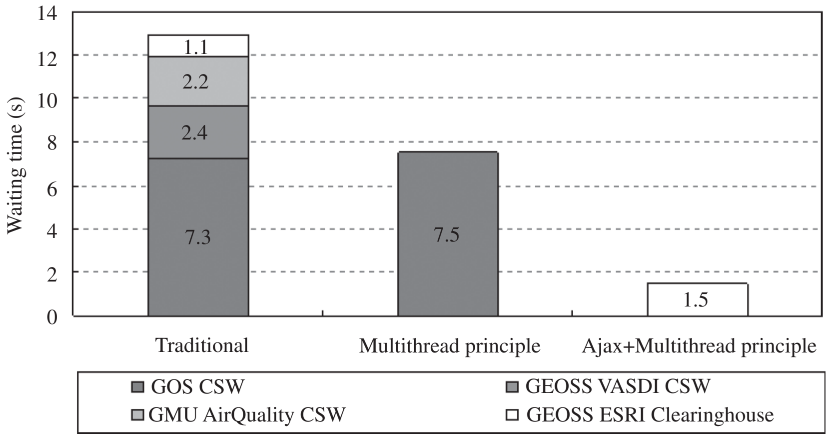

Figure 8. Comparison of multi-CSW search time with different techniques.

To test the performance of SCCH, we compared the waiting time of fetching service capability information with and without the SCCH. Fifteen WMS services with different layer numbers were selected and the SCCH was deployed at Fairfax, VA. The test site was located in San Diego, CA. The 15 services selected were distributed across the United States. We tested and recorded the average response time (from sending a request to receiving capability information) with and without SCCH and found a shorter response time with SCCH for all of the 15 selected services (Figure 9).

To test the performance of the layer-based search, we conducted three searches against the SCCH with 19,513 WMS layers registered using the layer-based search API. For all the searches, we set layer response time to be less than 1 s and they must support EPSG: 4326 reference system to enable layer overlay. Table 1 shows the search results and SCCH response time of the three searches: we found more than 10 layers for each search and we can then select the most appropriate layer from each search and overlay them to composite a new map. The SCCH response time was no more than 2 s.

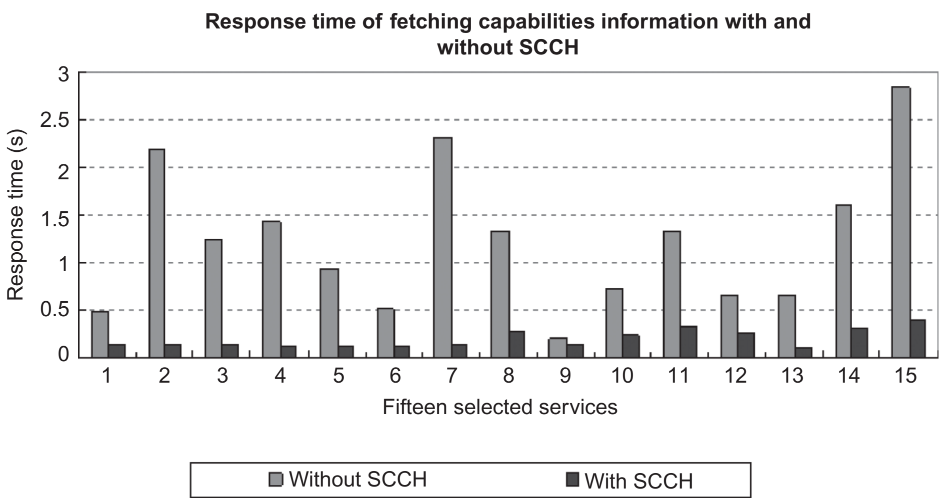

Figure 9. Response time for fetching capability with and without SCCH for 15 services.

Table 1. The search results and SCCH response time of the four searches.

Keyword	Search result	SCCH response time (s)
DEM	23 layers from 6 servers	1.16
Stream	28 layers from 19 servers	1.24
State boundary	108 layers from 50 servers	1.83

6. A spatial Web portal prototype based on the framework

This section proposes a spatial Web portal architecture by adapting the framework, and a spatial Web portal prototype is used to demonstrate how this framework can be utilized for supporting geospatial sciences.

6.1. Spatial Web portal architecture

A spatial Web portal (Yang *et al.* 2007a) is based on the portlet technologies (Hepper 2006) and provides a flexible and extensible architecture to integrate distributed and heterogeneous geospatial contents, such as data, services, and applications, onto a single website by providing a single entry point to the integrated geospatial resources. The framework proposed above can be easily adapted to construct an effective spatial Web portal to support geospatial sciences.

The spatial Web portal, through its connection mechanism to other systems (Figure 10), allows users to discover, integrate, utilize, and, finally, share WMS services. SCCH, described in Section 4.2, is the core component of this portal. The *multi-catalogue search*

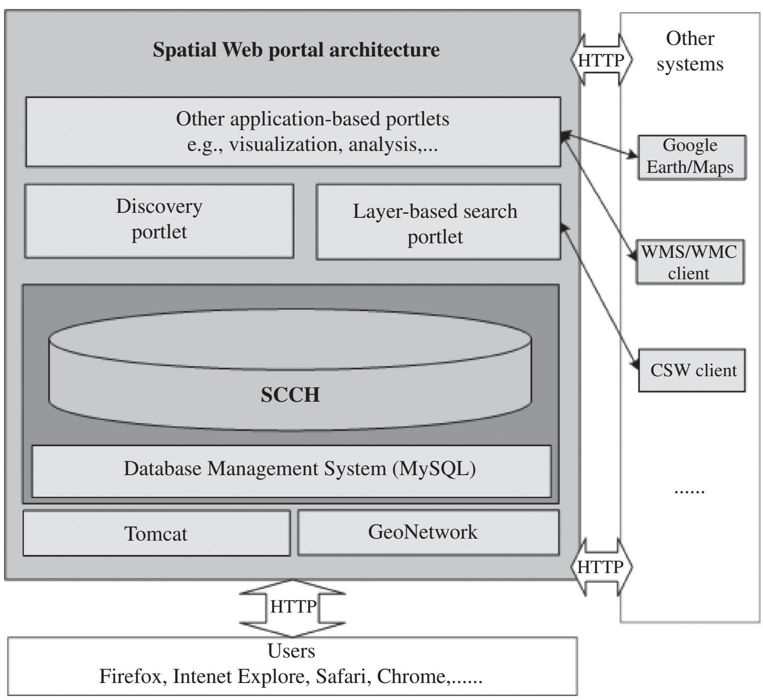

Figure 10. The spatial Web portal architecture based on the framework.

portlet is a CSW client adapting the search mechanism described in Section 4.1 to allow users to dig into multiple geospatial catalogues efficiently and effectively. The *layer-based search portlet* implements *layer-based search APIs* and allows users to discover the needed WMS layers more directly and accurately by searching layers crossing services with detailed constraints such as keywords, bounding box, reference system, and average response time.

6.2. The spatial Web portal prototype

A spatial Web portal prototype (Figure 11) was developed to integrate thousands of WMS seamlessly. Currently, 1145 WMS services including 19,513 WMS layers have been registered in SCCH.

OpenLayers (Figure 11) and Google Earth plugin (Figure 12) are integrated in this portal to provide smooth visualization of the Web services.

6.3. An animation usage scenario for the spatial Web portal prototype

The Earth's surface is always changing and Earth observation data with time dimension is extremely valuable in earth science studies. Currently, a lot of Earth observation data is published through time-enabled WMS. The animation portlet provides a flexible and effective approach to analyze time dimension. By animating Earth observation data, such as MODIS (Moderate Resolution Imaging Spectroradiometer) data products, we can intuitively visualize and examine the changing spatial and temporal patterns of phenomena, such as sea ice coverage, in Arctic area and global snow cover.

Figure 11. Graphic user interface of the spatial Web portal prototype.

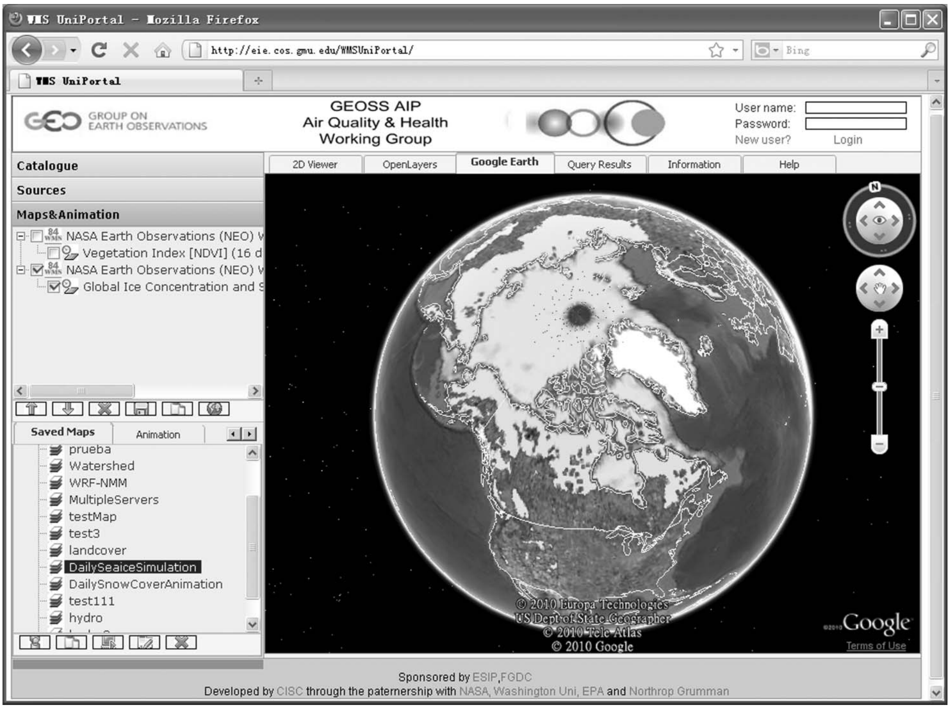

Figure 12. Google Earth plugin is integrated in the spatial Web portal.

The following is a scientific usage scenario analyzing correlation between snow cover and wildfire: (1) animate the monthly global active wild fires (NEO WMS http://neowms.sci.gsfc.nasa.gov/wms/wms, layer: MOD14A1_M_FIRE) to detect the seasonal patterns of wild fires; (2) animate global snow cover (NEO WMS, layer: MOD10C1_D_SNOW) from the year 2000 to 2008 by month to examine the change of snow cover for the same place at the same time of a year; (3) overlay the two layers and animate them simultaneously to visualize and analyze the potential correlation between wild fires and snow cover (Figure 13).

7. Discussion and conclusion

This article proposed a three-layer framework to facilitate the seamless integration of OWSs to support sharing of heterogeneous geospatial data and service through discovery, visualization, and animation: (1) for service discovery, a search mechanism has been proposed by adapting AJAX techniques including multithreading to allow users to discover multiple CSW services effectively and conveniently; (2) for service integration, a centralizing server (SCCH), acting as a service repository, is proposed to pre-process and cache the capabilities information for the distributed services; (3) a layer-based performance evaluator is proposed and used to evaluate each service layer's performance stored in SCCH; (4) to facilitate service interoperability, a layer-based CSW service and APIs have been introduced to serve the layer-based metadata and service quality information through open standard-based interfaces.

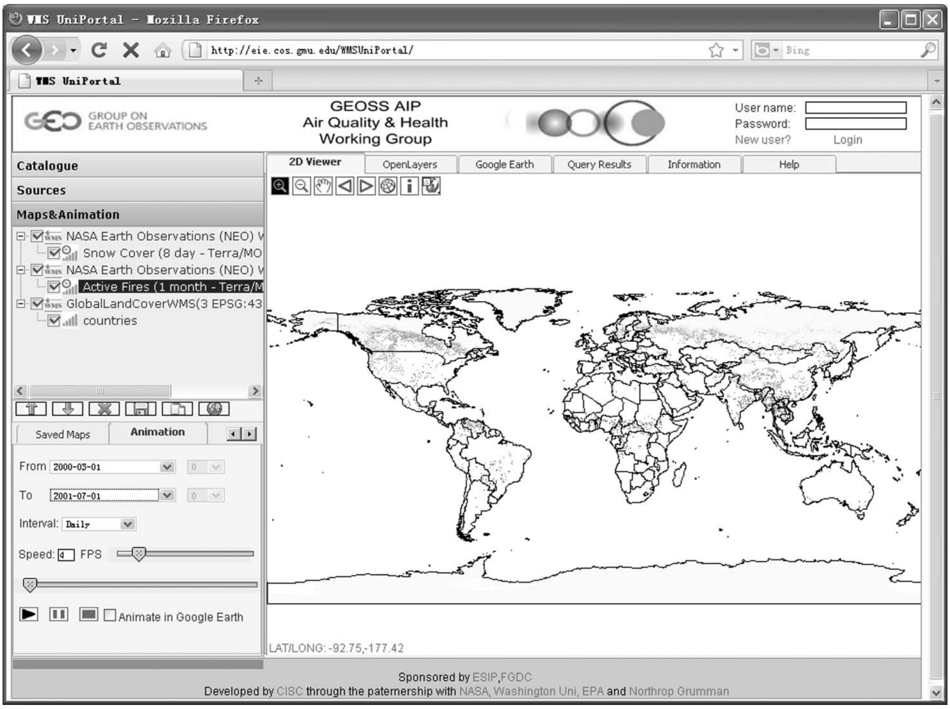

Figure 13. Correlation analysis between snow cover and wildfires in March, 2000 to 2009.

A spatial Web portal architecture is introduced and a spatial Web portal prototype is developed to integrate tens of thousands of WMS layers seamlessly, while providing efficient search and comprehensive visualization/animation to the integrated services. This framework is proved to be critical for enhancing the performance of multi-catalogue search and for integrating thousands of WMSs seamlessly.

The performance evaluator used in this framework records the response time of the service layer, which is only one dimension of Quality of Service (QoS) (Parasuraman *et al*. 1988); the service reliability, accuracy, and consistency can also be recorded because of the characteristics of geospatial data. For example, some layers may have a very short response time but the returned data is not consistent with its metadata or the data is not accurate enough (Wu and Zhang 2007). Therefore, further research is needed to upgrade the layer-based one-dimension performance evaluator to a multi-dimension evaluator to provide more comprehensive and reliable performance indications.

WMS is used to illustrate the seamless integration capability of this framework. Similar to WMS, both WFS and WCS are organized as layer structure (Vretanos 2005, Whiteside and Evans 2008). Therefore, SCCH can be easily adapted to support WFS and WCS.

In the future, we plan to enhance the framework to integrate intelligence for the performance and interoperability of the discovery component of the framework using semantic search (Alesso and Smith 2005, Li *et al.* 2008) with Semantic Web for Earth Environmental Terminology (SWEET) (Raskin 2005), Web Ontology Service (WOS) (Lacasta *et al*. 2007), BUSTER system (Bremen University Semantic Translator for Enhanced Retrieval) (Visser 2004), and spatio-temporal conceptual schemas (Sotnykova *et al*. 2005). Initiatives and studies on the cloud computing can also be potentially used to further improve the overall performance of this framework by (1) adapting a workflow engine, (2) inserting a grid layer (Zhang and Tsou 2009) between presentation layer and integration layer of this framework, and (3) connecting to a grid-based gateway (Wang and Liu 2009).

Global concerns, such as water, disaster, climate, and energy, call for coordinated, collaborative global Earth observations and OWSs to discover, share, and interoperate the accumulated geospatial resources. With the increasing number of OWSs available on the Internet and the increasing demand of interoperability, the performance issues of effective discovery and seamless integration of these heterogeneous and distributed OWSs will become even more complicated and prominent. We expect that the performance-improved framework proposed in this article can raise the awareness of integrating these dispersed OWSs to better support geospatial sciences by leveraging SOA, QoS, Distributed Geographic Information Processing (DGIP) principles, and other technologies.

References

Alesso, H.P. and Smith, C.F., 2005. *Developing semantic web services*. Natick, MA: A K Peters, 445.

Alonso, G., *et al*., 2004. *Web services: concepts, architectures, and applications*. New York: Springer Verlag.

Beaujardiere, J., 2004. *OGC web map service interface* [online], version 1.3.0, OGC 03-109r1. Available from: http://portal.opengeospatial.org/files/index.php?artifact_id = 4756 &passcode=b9mnkb6rr7uc1hs1t1ue [Accessed 6 September 2009].

Brodaric, B., Fox, P., and McGuinness, D.L., 2009. Geoscience knowledge representation in cyberinfrastructure. *Computers & Geosciences*, 35 (4), 697–699.

Chen, H. and Mohapatra, P., 2005. Using service brokers for accessing backend servers for web applications. *Journal of Network and Computer Applications*, 28 (1), 57–74.

Hepper, S., 2006. *Java*™ *portlet specification* [online], Version 2.0, Sun Microsystems. Available from: https://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS_JCP-Site/en_US/-/USD/ViewFilteredProducts-SingleVariationTypeFilter [Accessed 30 October 2009.

Koponen, T. and Virtanen, T., 2004. A service discovery: a service broker approach. *In*: *Proceedings of the 37th Hawaii international conference on system sciences* [online]. Available from: http://csdl.computer.org/comp/proceedings/hicss/2004/2056/09/205690284b.pdf [Accessed 2 November 2009].

Kottman, C., 1999. The OpenGIS abstract specification. Topic13: Catalog Services (version4). OpenGIS Project Document 99–113, OpenGIS Consortium Inc.

Kralidis, A.T., 2007. Geospatial web services: the evolution of geospatial data infrastructure. In: A. Scharl and K. Tochtermann (Eds.), *The Geospatial Web*, (London: Springer), 223–228.

Lacasta, J., *et al*., 2007. A Web Ontology Service to facilitate interoperability within a spatial data infrastructure: applicability to discovery. *Data & Knowledge Engineering*, 63, 947–971.

Laurini, R., 1998. Spatial multi-database topological continuity and indexing: a step towards seamless GIS data interoperability. *International Journal for Geographic Information Science*, 12 (4), 373–402.

Leymann, F., 2001. Web Services Flow Language (WSFL 1.0), IBM software Group. Available online at http://www.itee.uq.edu.au/~infs7201/Assessments/Assignment%201%20Material/WSFL.pdf (accessed 1 May 2010).

Li, W., Yang, C., and Yang, C., 2010. An active crawler for discovering geospatial web services and their distribution pattern. *International Journal of Geographic Information Science*, 24(8), 1127–1147.

Lutz, M., *et al*., 2007. A rule-based description framework for the composition of geospatial information services. *In*: F.T. Fonseca, M.A. Rodríguez, and S. Levashkin, eds. *GeoS – Volume 4853 of lecture notes in computer science*. Berlin: Springer, 114–127

Maguire, D.J. and Longley, P.A., 2005. The emergence of geoportals and their role in spatial data infrastructures. C*omputers, Environment and Urban Systems*, 29, 3–14

Mena, E., *et al*., 2000. OBSERVER: an approach for query processing in global information systems based on interoperation across pre-existing ontologies. *International Journal of Distributed and Parallel Databases*, 8, 223–271.

Nebert, D., Whiteside, A., and Vretanos P., 2007. *Catalog services implementation specification* [online], Version 2.0.2, OGC 07-006r1. Available from: http://portal.opengeospatial.org/files/index.php?artifact_id=20555&passcode=773t9zth2s77ma80h1n5 [Accessed 17 October 2009].

Nikola, M. and Miroslaw, M., 2004. Current solutions for web service composition. *IEEE Internet Computing*, 8 (6), 51–59.

Nogueras-Iso, J., *et al*. 2005. OGC catalog services: a key element for the development of spatial data infrastructures. *Computers & Geosciences*, 31 (2), 199–209.

Oren, E., Gueret, C., and Schlobach, S., 2008. Anytime query answering in RDF through evolutionary algorithms. *In*: *Proceeding of international semantic web conference*, 26–30 October 2008, Karlsruhe, Germany, 98–113 (Berlin: Springer).

Parasuraman, A., Zeithaml, V.A., and Berry, L.L., 1988. SERVQUAL: a multiple-item scale for measuring consumer perceptions of service quality. *Journal of Retailing*, 64 (1), 12–40.

Parkinson, C.L., Ward, A., and King, M.D., 2006. *Earth science reference handbook* [online], National Aeronautics and Space Administration, Goddard Space Flight Center. Available from: http://eospso.gsfc.nasa.gov/ftp_docs/2006ReferenceHandbook.pdf [Accessed 17 October 2009].

Ramroop, S. and Pascoe, R., 2001. Use of LDAP to partially implement the OGIS discovery service. *International Journal of Geographic Information Science*, 15 (5), 391–413.

Raskin, R., 2005. *Semantic web for earth and environmental terminology (SWEET)* [online], Available from: http://archive.esipfed.org/tech_center/presentations/SWEET.ppt [Accessed 17 October 2009].

Sanjiva, W. and Francisco, C., 2002. Business processes: understanding BPEL4WS, Part 1. IBM developer Works

Shen, J., *et al*., 2007. Analysis of business process integration in Web service context. *Future Generation Computer Systems*, 23 (3), 283–294.

Skylab, 2009. OGC WMS server list. Available from: http://www.skylab-mobilesystems.com/ger/wms_serverlist.html [Accessed 10 September 2009].

Solomon, D. and Rankings, R., 1997. Microsoft SQL Server 6.5 unleashed. Indianapolis, IN: Sams, 1–15.

Sotnykova, A., *et al*., 2005. Semantic mappings in description logics for spatio-temporal data-base schema integration. *Journal on Data Semantics*, III, 143–167.

Thakkar, S., *et al*., 2002. Dynamically composing web services from on-line sources. In: *Workshop on Intelligent Services Integration, The Eighteenth National Conference on Artificial Intelligence* (*AAAI*), 28 July–1 August 2002, Edmonton, Alberta Canada.

Vaccari, L., Shvaiko, P., and Marchese M., 2009. A geo-service semantic integration in Spatial Data Infrastructures. *International Journal of Spatial Data Infrastructures Research*, 4, 24–51.

Visser, U., 2004. Intelligent information integration for the semantic web. *Lecture Notes in Computer Science*, Vol. 3159. Berlin, Heidelberg: Springer, 150.

Vretanos, P.A., 2005. OGC web feature service implementation specification [online], Version 1.1.0. OGC 04-094. Available from: http://portal.opengeospatial.org/files/index.php?artifact_id=8339&passcode=b9mnkb6rr7uc1hs1t1ue [Accessed 12 October 2009].

Wang, H., *et al*., 2004. Web services: problems and future directions. *WebSemantics: Science, Services and Agents on the World Wide Web*, 1 (3), 309–320.

Wang, S. and Liu, Y., 2009. TeraGrid GIScience gateway: bridging cyberinfrastructure and GIScience. *International Journal of Geographic Information Science*, 23 (5), 631–656.

Ward, K., 2009. *NEO* [online]. Available from: http://neo.sci.gsfc.nasa.gov/View.html?page=AboutWMS [Accessed 12 October 2009].

Whiteside, A. and Evans, J.D., 2008. *OGC web coverage service (WCS) implementation standard* [online], Version 1.1.2 07-067r5, Available from: http://portal.opengeospatial.org/files/index.php?artifact_id=27297&passcode=b9mnkb6rr7uc1hs1t1ue, last [Accessed 12 October 2009].

Wu, H. and Zhang, H., 2007. Quality of geospatial information service (QoGIS): concept and research framework. *Geomatics and Information Science of Wuhan University*, 32 (5), 15–18.

Yang, C. and Raskin, R., 2009. Introduction to distributed geospatial information processing research. *International Journal of Geographic Information Science*, 23 (5), 553–560

Yang, C., *et al*., 2005. Performance improving techniques in WebGIS. *International Journal of Geographic Information Sciences*, 19 (3), 319–341.

Yang, C., *et al*., 2008. Distributed geospatial information processing: sharing earth science information to support Digital Earth. *International Journal of Digital Earth*, 1 (3), 259–278.
Yang, C., *et al*., 2010. Geospatial cyberinfrastructure: past, present and future. *Computer, Environment and Urban Systems*, 34 (4), doi: 10.1016/j.compenvurbsys.2010.04.01.
Yang, P., Cao, Y., and Evans, J., 2007a. WMS performance and client design principles. *Journal of GIScience & Remote Sensing*, 44 (4), 320–333.
Yang, P., *et al*., 2007b. The emerging concepts and applications of the spatial web portal. *PE&RS*, 73 (6), 691–698.
Yue, P., *et al*., 2007. Semantics-based automatic composition of geospatial web service chains. *Computers & Geosciences*, 33, 649–665.
Zhang, T. and Tsou, M., 2009, Developing a grid-enabled spatial web portal for Internet GIServices and geospatial cyberinfrastructure. *International Journal of Geographic Information Science*, 23 (5), 605–630.
Zhang, T., *et al*., 2006. Building an intelligent geospatial cyberinfrastructure: an analytical problem solving approach. *Proceedings of SPIE*, 64200A-64200A-14, doi:10.1117/12.712656.