THE DEVELOPMENT OF A PROTOTYPE GEOSPATIAL WEB SERVICE SYSTEM FOR REMOTE SENSING DATA

Meixia Deng [a, *], Peisheng Zhao [a], Yang Liu[a], Aijun Chen[a] Liping Di [a]

[a] George Mason University, Laboratory for Advanced Information Technology and Standards - (mdeng, pzhao, yliu5, achen6, ldi)@gmu.edu

KEY WORDS: development, remote sensing, data, geospatial, Web service, information

ABSTRACT:

Recent progress on Web service technology may provide a solution to current problems in using remote sensing data. This paper presents a prototype geospatial Web service system to demonstrate the advantages of such a system over the traditional non service- based systems for reducing the difficulty of using remote sensing data for research and applications. The foundations of the prototype system are the OGC and W3C standards that define the interfaces for accessing geospatial data and the methods for constructing chainable geospatial Web services. In the system, we have implemented a limited number of interoperable services, including WCSportal, reprojection, visualization, classification, reformatting, subsetting, and georectification. The system allows users to dynamically chain the services with other services and services with data to produce the user-specified products over the Web. The successful implementation of this prototype system will provide us experience in for building large, operational Web systems for geospatial information and knowledge services.

1. INTRODUCTION

1.1 W3C and OGC Web Services Standards

Web technologies have developed significantly from their early days of being used only to provide an interface for distributed services with HTML forms calling from CGI scripts to the point where now XML based environments have enabled Web services.

The World Wide Web Consortium (W3C), created in October 1994, is an open, international organization with the goal of leading the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability. W3C activities are generally organized into groups. The Working Groups in W3C are in charge of technical developments. Web services activity in W3C currently consist of four major working groups: XML Protocol Working Group, Web Services Description Working Group, Web Services Architecture Working Group and Web Services Choreography Working Group [http://www.w3.org/2002/ws/]. These working groups develop the technical standards for Web services application, produce the documents used for Web services such as XML protocol , SOAP version 1.2, WSDL version 2.0, Web services architecture and choreography.

The Open GIS Consortium, Inc. (OGC) is a member-driven, non-profit international trade association that is leading the development of geoprocessing interoperability computing standards. Within the broader context of Web Services, OGC Web Services (OWS) represent an evolutionary, standards-based framework that enables seamless integration of a variety of online geoprocessing and location services [OGC Interoperability Program White Paper, 2001]. OGC develops standards in the context of OWS. OGC is the only international organization dedicating to develop geospatial implementation standards based on ISO, FGDC, INCITS, W3C, and other organizations' content standards.

W3C and OGC Web services standards are the foundation to implement our geospatial Web services and build our prototype geospatial Web service system for remote sensing data.

1.2 Web Services and Geospatial Web Services

What is a Web service? There are many definitions from different communities. The W3C Web services architecture working group gives the definition as the following:

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards [W3C note 2004].

As stated above, a machine-processable format WSDL plays a key role in describing a Web service. WSDL is the abbreviation for Web Service Description Language, an XML language for describing Web services. The W3C Web Service Description Working Group published their working draft WSDL version 2.0 on March 26, 2004 [W3C WD 2004]. WSDL is the standard language used to describe all Web services by both W3C and OGC.

In short, we can define a Web service as a program that performs a defined action and that can be found, invoked, and executed over the Web. Then it is quite straightforward for us to define a geospatial Web service. A geospatial Web service is a Web service which performs an action on geospatial data or

Comment [BMS1]: Meaning is not clear here. As written, implies that products are the produced on the Web. What is really going on? Do the products somehow get generated over the Web. Does the user specify the products, send a Web inquiry and then the system produces them?

Comment [BMS5]: "abstract" and "content" are not contrasting terms. Abstract can be constrasted with implementation and content with, for example, encoding.

Deleted: abstract and

Comment [BMS2]: Acronym should be expanded first time it is used

Comment [BMS3]: Acronym should be expanded first time it is used.

Deleted: about

Comment [BMS4]: What's the list here? 1)Web services about XML 2)SOAP 3)WSDL 4)Web services architecture 5) choreograpy? Are the Web services only about XML? Are Web services architecture and choreography two items or one?

* Corresponding author.

information. We need to distinguish geospatial Web services from Web services in general because of the special characteristics of geospatial data.

Geospatial data are the data which have spatial components (e.g. geo-location or geo-reference). Geospatial data differ from the general data such as numbers or characters largely in many ways. Geospatial data mostly involve in complex data type instead of the simple data type. And also, processing these data are much challenging. In general, geospatial data are much more complex and difficult for data users than the general data. Using geospatial data usually requires that the users have expert knowledge or enough resources and spend a lot of time to preprocess the data for further use. For example, NASA has archived huge amounts of satellite data in HDF/HDF-EOS format because the satellites data are all geospatial. How to use these data has been a headache for many scientists and researchers since the first day.

Geospatial Web services, which works with geospatial data, are quite different from a general Web service as well. For examples, an operation for a geospatial Web service may be very complicated and hierarchical while the operation for a general business Web service is usually simple and atomic. The service discovery, access, binding and chaining for a geospatial Web service may also highly differ from a general Web service.

1.3 Motivation to Build Geospatial Web Service System

The importance and necessity of building a geospatial Web service system that can provide ready-to-use information instead of raw data are obvious. Huge amounts of Earth remote sensing data have been collected by space agencies, and of terabytes data are expected to be collected daily. Those data are valuable not only for scientific research but also for other socio-economic activities. However, currently most of the data are only available to users in raw form. Users need to have both the expert knowledge and significant amount of resources to process the data into ready-to-use information. This situation limits many potential uses of the data by broader user communities. For instance, some users do not have enough knowledge or resources to use data for their analysis and information extraction and lots of valuable data are wasted, and sometimes the decision makers need to promptly respond to an incident or event, but the time needed to process and analyse data is too long to fulfil the task.

A geospatial Web service system based on the newly developed Web technologies, which allows users to dynamically chain the services with services and services with data to produce the user-specified products over the Web, may provide a solution for those puzzles. We propose to build a prototype geospatial Web service system to demonstrate its robust power and advantages over traditional non-service based systems for reducing the difficulty of using remote sensing data for research and applications.

2. SYSTEM DESIGN

To design our prototype system, we take the following into our major considerations:

1. The system interoperability
2. Web services interoperability
3. The personalization capability
4. Software reuse
5. System evolution capability/capacity

Our final goal is to provide products tailored to the individual user's needs, including selection of spatial or temporal coverage, map projection, format, bands etc. and information products produced on-demand by executing Web-executable geospatial process models. To realize our goal, the system must be an open, personalized, standard-based, and interoperable system [Di, L & Yang, W, 2002].

System interoperability means that the system can access and process data and services from multiple resources. Web service interoperability means that all geospatial Web services in the system must be standards-based and interoperable for coupling with data and other services.

The personalization capability of the system means that users will be able to define the geospatial models and service composition, use existing services or contribute new services to obtain the data and information that exactly match their own needs.

Software reuse means that all geospatial Web services and geospatial models are designed to be dynamically reusable by other systems and services through interoperable just-in-time integration. All reuse may involve some code modifications.

The capability for evolution of a system depends largely on the system architecture. A system that is component-based with each component using standard interfaces, and is modular and self-contained, is easy to evolve. We will design the prototype Web service system as a component-based system so that the system allows self evolution capability/capacity by allowing standards-based geospatial Web service modules and geospatial models to be plugged in.

Currently, the prototype geospatial Web service system to be developed here is a modelling and execution system which has the capability to access online geospatial data and services, to build models (service composites) which chain individual standards-based geospatial Web services together and register the models in the catalogue and to execute the chain for solving complex tasks dynamically and efficiently. Figure 1 is the abstract architecture for our prototype system.

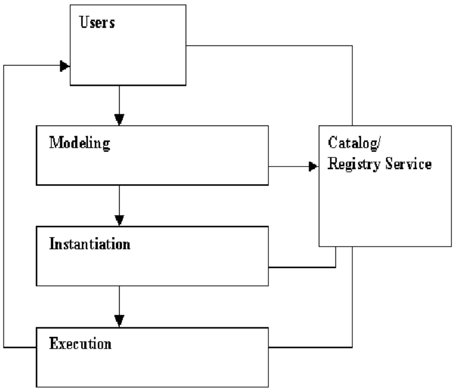


Figure 1. The Abstract Architecture

Deleted: (

Deleted: ,

Deleted: , etc.)

Comment [BMS6]: such as?

Deleted: a

Deleted: To use them

Deleted:

Deleted:

Deleted:

Deleted: (

Comment [BMS9]: Does this phrase stand for the ability to define geospatial models, or is it another user capability? If the second, it should not be in parentheses

Deleted:)

Deleted: .

Comment [BMS7]: its?

Deleted: and the

Formatted: Bullets and Numbering

Comment [BMS8]: What does this mean? The capability for individuals to do what?

3. IMPLEMENTATION

3.1 The Development of Geospatial Web Services

To build the prototype system, we need to have the geospatial data and services be available over the Web. With the OGC and W3C standards that define the interfaces for accessing the geospatial data and the method for constructing chainable geospatial Web services, we have implemented a limited number of interoperable services, including WCSPortral service, reprojection, visualization, classifications, reformatting, subsetting, and georectification. Those services are implemented with standard interfaces defined by OGC for taking coverage data from coverage servers or from the output of other services as the inputs.

WCSPortal service has been built first because the OGC Web coverage service (WCS) [OGC, 2003] provides a way to access and fetch geospatial data online [ ]. This WCSPortal service is a wrapper of WCS. It uses the GET method of WCS to get a coverage, save it locally, and return a URL for next step.

Comment [BMS10]: WCSPortal service. That's what the wording implies.

Some very useful geospatial Web services such as reformatting, re-projection, sub-setting, geo-rectification are also being implemented. We can envision that more and more services and data will be available on-line in the near future. Those services will be stand-alone applications that can be used independently as well as the components for chaining. And the availability of services on line will also save many data users significan time in preprocessing the data for their own use.

Comment [BMS11]: "such as" and "etc." are not both needed. Both imply that the list in not comprehensive.

Deleted: etc.

Some image classification services are also implemented for demonstration, such as "WICS_Grass_Unsupervised" , a SOAP based Web service. It uses Geographic Resources Analysis Support System (GRASS), an open source GIS system, to do an unsupervised image classification. We use one of our image classification services as an example to represent our service model and structure in Figure 2. The bottom part is the Image Classification Core Class. The basic classification algorithm is written in C (or another computer language). A Java class – ImageClassifier – provides Java access to those methods through the Java Native Interface .The Image Classification Core Class provide the functions to be used in classification services, such as supervised classification, unsupervised classification, and training. This Core Class will be called by the Web Image Classification Service.

Comment [BMS12]: Undefined acronym

The Web Image Classification Service consists of two different implementations, the HTTP GET & POST version and the SOAP version. Both versions provide four operations, GetCapability, DescribeClassifier, TrainClassifier, and GetClassifier. Both versions accept the same parameters. The only difference is the protocol of request and response.

The WICS Web interface is a Web application, including a set of html pages and a Java servlet. To the Web Image Classification Service, this Web interface is a client. A user can use these Web pages to submit a WICS request to a Web Image Classification Service (HTTP version) and specify whether he wants the raw data back or a picture result. The Web interface will generate pictures for the user if requested. Also the user can use other client tools to connect to both versions of the service.

The details for those services and the WSDL documents describing those services can be accessed by visiting our Website http://ws.laits.gmu.edu.

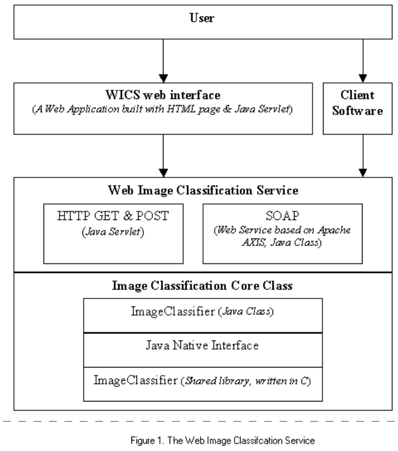

User
WICS web interface
(A Web Application built with HTML page & Java Servlet)
Client Software
Web Image Classification Service
HTTP GET & POST
(Java Servlet)
SOAP
(Web Service based on Apache AXIS, Java Class)
Image Classification Core Class
ImageClassifier (Java Class)
Java Native Interface
ImageClassifier (Shared library, written in C)
Figure 1. The Web Image Classifcation Service

Figure 2. The Web Image Classification Service Structure

3.2 The Development of the Prototype System

The real power of a geospatial Web service system relies on its ability to chain individual standards-based services. A service chain is a service composite. Built on XML, SOAP, and WSDL, and originated from XLANG and WSFL, the Business Process Execution Language for Web Service (BPEL4WS) specification is positioned to become the Web service standard composition language over all others. It supports two distinct usage scenarios: one is used to describe an abstract process which is non-executable and the other is to define an executable process. We adopt BPEL4WS in our geospatial Web service system. We use BPEL4WS to define a new Web service by composing a set of existing services (a service chain). The interface of the composite service is described as a collection of WSDL portTypes, just like other Web services. The chain is executed with the BPEL engine. Figure 3 is the current prototype system structure implementation flowchart.

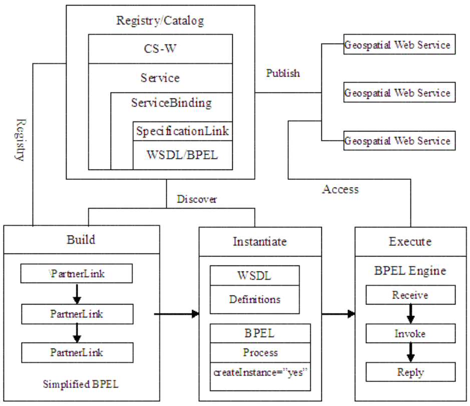
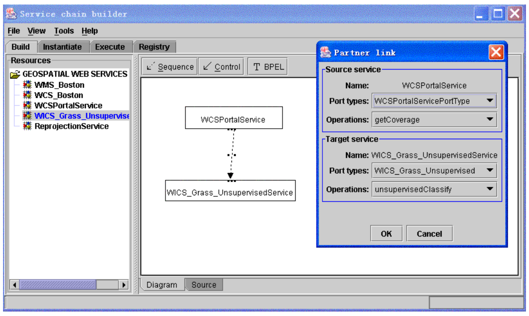


Figure 3. The Implementation of the Prototype System

The core part of the prototype system consists of four major components, Build, Instantiate, and Execute and Registry Catalogue. The component Build is used for the modelling process that provides the user interaction with the system to construct a service chain (composition).This modelling process will produce a BPEL document with an abstract description of the service chain. The component Instantiate will logically walk through the chain for validation, and an executable BPEL document will be produced in this step. These two components are coupled with the Registry Catalogue with the service discovery and registry functions. The Registry Catalogue also publishes the services. The component Execute can access services (or service composites), execute the service chain with theBPEL engine and return the result. For better performance, a service ontology is needed to be built into the catalogue service for users to search for data and services for their own geospatial interests.

3.3 Illustration

For demonstration simplicity, we test the developed prototype system with a user request to get land cover information for San Francisco, California. A user can use the system in different ways. For example, an expert user who wants to build his own service chain could use the system catalogue service and find the available services and data (or register his own services or data for use). Figures 4 is the instance of this expert use. The user searches through the system and finds the WCSPortal service to access SPOT image data for San Francisco and one unsupervised Web image classification service chains these two services together and executes the chain to get the classified image (Figure 5). There are two WSDL files and Two BPEL files are involved in this chain. The two WSDL files WCSPortalWrapper.wsdl and WICS_Grass_UnsupervisedWrapper.wsdl, describe the two services respectively. One BPEL file is used to define the chain as a new service interface, and another BPEL file is used to define the executable process for this services composition. The engine takes all the four files for execution. The intermediate result of the getCoverage in the service chain (Figure 6) is invisible for the user, here is presented for comparison. Users without their own models may go to the system and search for the existing service for land cover information retrieval (as we know, one expert user may have registered his model as a new service in the system) or the system will automatically provide a chain model based on ontology and type matching for user testing and evaluation.



Source service
Name: WCSPortalService
Port types: WCSPortalServicePortType
Operations: getCoverage

Target service
Name: WICS_Grass_UnsupervisedService
Port types: WICS_Grass_Unsupervised
Operations: unsupervisedClassify

Figure 4. An User Instance

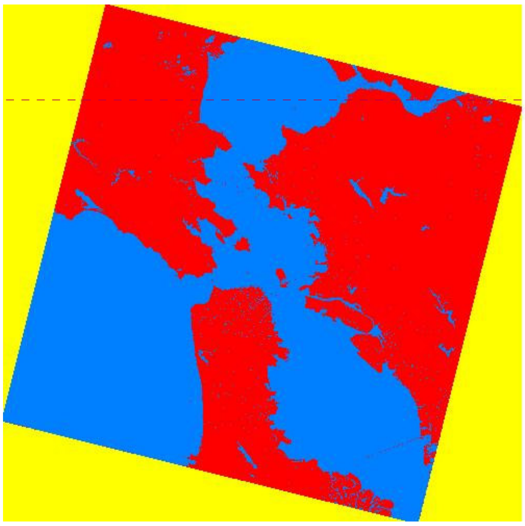


Figure 5. The Classified Image by User Request

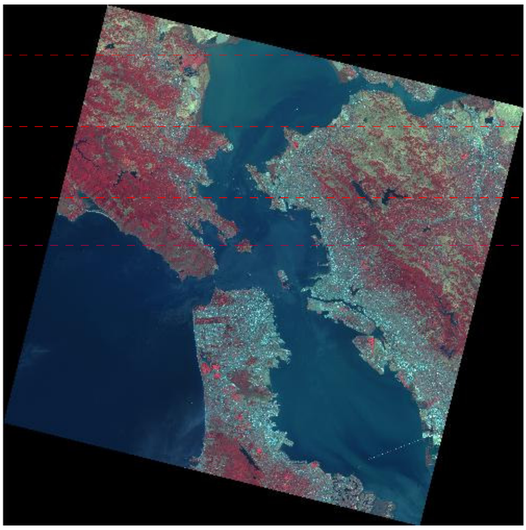


Figure 6. The Intermediate Result as GetCoverage

4. CONCLUSIONS

In this paper, we described a successful implemention of a limited number of interoperable geospatial Web services, based on W3C and OGC standards, for taking coverage data from coverage servers or from the output of other services as the inputs. Those services are available for stand-alone applications and service chaining. A prototype geospatial Web service system for remote sensing data is developed to access those services and construct service chains to solve complex tasks dynamically. Initial demonstration shows that the prototype system enables users to obtain the requested information and

Comment [BMS13]: This what? the modelling process?

Deleted: is

Deleted: ,

Deleted: T

Deleted: which describe the two services

Deleted: and two

Deleted: s

Deleted: which

Comment [BMS14]: Not clear Are there 1) Two BPEL files that define the chain as a new service interface 2)the executable process or two BPEL files that define the chain as 1) a new service interface 2)the executable process

Deleted: are involved in this chain

Comment [BMS15]: available for or capable of

Deleted: ble

knowledge tailored their individual needs, rather than only raw data that needs further processing. The system demonstrates the basic functionalities of catalogue search, service chain construction and registration, and chain execution. In the future, more functionalities will be added to the system, such as an ontology service, automatic service chaining based on user request and service type and data type matching. The development of this prototype system will provide us valuable experience for developing operational, interoperable, distributed, standards-compliant and intelligent Web-based geospatial information systems.

References

Di, L., W. Yang, D. Deng, and Ken. McDonald, 2002. "Interoperable, Personalized, On-demand Geospatial Data Access and Services Based on OGC Web Coverage Service (OWS) Specification", Proceeding of NASA Earth Science Technology Conference, CDROM, Pasadena, California.

OGC, 2003. The OGC Web Coverage Services Version 1.0.0. Editor, J. Evans. OGC 03-065r6. http://www.opengis.org/docs/03-065r6.pdf

OGC Interoperability Program White Paper, 2001. http://ip.opengis.org/ows/010526_OWSWhitepaper.doc

W3C note 2004. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

W3C WD 2004. http://www.w3.org/TR/2004/WD-wsdl20-20040326/

Formatted

Acknowledgement

This research project (Principal Investigator: Dr. Liping Di) is supported by grants from NASA Earth Science Technology Office (ESTO), NASA Earth Science Data and Information System (ESDIS) project, and Open GIS Consortium (OGC).