GEOSPATIAL SENSOR WEB DATA DISCOVERY AND RETRIEVAL SERVICE BASED ON MIDDLEWARE

Nengcheng Chen[a,b] *, Liping Di[b], Genong Yu[b] , Zeqiang Chen[a] , Jie He[a]

[a]State Key Lab of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS) Wuhan University, 129 Luoyu Road, Wuhan, hubei 430079 China - cnc_dhy@hotmail.com
[b]Center for Spatial Information Science and Systems(CSISS), George Mason University, 6301 Ivy Lane, Suite 620, Greenbelt, MD 20770, USA - (nchen3, lid,gyu)@gmu.edu

WG II/6

KEY WORDS: Distributed GIS, Data Infrastructure, Interoperability, Sensor Web, Middleware

ABSTRACT:

Recent advances in Sensor Web geospatial data capture, such as high-resolution satellite imagery and Web- ready data processing and model technologies, have led to the generation of large amounts of datasets from real-time or near real-time observation and measurement. How to find and retrieve the sensor or data complying with criteria such as specific time, location, and scale has become a bottleneck in Sensor Web-based applications. We proposed to answer the above question by building a catalogue service through a harvest service based on search engine and multi protocol data retrieval service middleware based on schema mapping and matching, called Data Discovery and Retrieve Service (DDRS). DDRS hides the complexity of the geospatial world behind its standard interfaces. DDRS consists of a harvest server, a discover server, a retrieve server and a portal server. The retrieve server explores three standard interfaces-Web Feature Service (WFS), Web Coverage Service (WCS) and Catalogue Service for Web (CSW) to portal server. The DDRS is designed to be capable to discover geospatial service and data object, and retrieve data from diverse homogeneous and heterogeneous service sources: WCS server from different sources, different versions and real time observation and measurement; CSW server from different sources and versions; WFS server from different sources, different versions and real time observation and measurement. The DDRS is designed and implemented by the service middleware and Java 2 enterprise edition (J2EE) technology.

1. INTRODUCTION

1.1 Geospatial Sensor Web

A sensor network is a computer-accessible network of many spatially distributed sensors that monitor conditions such as temperature, sound, vibration, pressure, motion or pollutants at different locations. The term Sensor Web was first proposed by the NASA Sensor Web Applied Research Planning Group. In 2001, NASA (Delin and Jackson, 2001) defined the sensor Web as "a system of intra-communicating spatially distributed sensor pods that can be deployed to monitor and explore new environments".

In 2005, the GeoICT Lab (Liang et al., 2005) in Canada broadened the above scope by including a wide range of sensors and applications in the definition of the Sensor Web, which they defined as an electronic skin of the Earth, offering full dimensional, full-scale, and full-phase sensing and monitoring at all levels, global, regional, and local. The Sensor Web includes both in situ sensors and remote sensors. Sensors can be mobile or stationary.

The Geospatial sensor Web has been defined (Di, 2007) as the sensor Web that performs Earth observations (EO).It is envisioned that in the future the major new EO sensors will be Web-ready. Some existing sensors, such as EO-1, will also be converted into Web-ready sensors. Legacy data systems and even simulation models that provide data and information to application, can be simulated as virtual Web-ready sensors.

1.2 OGC data service protocol and interface

The most popular OGC protocols for data access include Web Map Services (WMS) (de la Beaujardiere, 2006), Web Feature Services (WFS) (Vretanos, 2002) and Web Coverage Services (Evans, 2003; Lee et al., 2005). They are designed to access geospatial data encoded in a raster map (e.g., JPEG picture), feature (e.g., weather station data) and grid (e.g., space airborne images) respectively. Geospatial CSW (Nebert and Whiteside, 2005) is designed to provide capabilities for advertising and discovering shared geospatial data and services over the Web. The metadata of data and services are both registered in a catalogue, So that the users could discover, bind and invoke these services by searching the CSW.

.

1.3 OGC SWE standard

In the past five years, OGC and ISO have been formulating standards and protocols for the geospatial sensor Web. The OGC Sensor Web Enablement (SWE) initiatives have developed three information models: SensorML (Botts, 2006), Observation and Measurements (O&M) (Cox, 2006), and Transducer Markup Language (TML) (Havens, 2006).The SWE initiatives have developed three service implementation specifications based on the assumption that all sensors are

* An associate professor in wuhan university. His current research is focused on Geospatial Sensor Web, Polar information service, Web and Wireless GIS (W2GIS) and Geo Grid Computation. Email:cnc_dhy@hotmail.com

connected to the Web:Sensor Planning Service (SPS) (Simonis, 2005), Sensor Alert Service (SAS) (Simonis and Echterhoff, 2006), and Sensor Observation Service (SOS) (Na and Priest, 2006). Besides, Web Notification Service (WNS) (Simonis and Wytzisk, 2003), by which a client may conduct an asynchronous dialog with one or more other services. When the above four services and three information model work together, it is possible to use Web services to discover, access, and control sensors or sensor data.

1.4 Other data protocols

In atmosphere, ocean, solar, and cryosphere science communities, data access protocols such as netCDF (Network Common Data Form) and OPeNDAP (Open-source Project for a Network Data Access Protocol) have been widely used. In order for users to find data served by those protocols, Unidata has developed THREDDS (Thematic Realtime Environmental Distributed Data Services) data catalogue system. Although Earth system science, global change, and applied geospatial research need to use data from all those communities for integrated analysis and modeling, there is no interoperability among those protocols. The interoperability gaps among protocols make cross-protocol and cross-community data access extremely difficult.

1.5 Middleware

Middleware is connectivity software that consists of a set of enabling services that allow multiple processes running on one or more machines to interact across a network. Middleware is essential to migrating mainframe applications to client/server applications and to providing for communication across heterogeneous platforms. This technology has evolved during the 1990s to provide for interoperability in support of the move to client/server architectures. The most widely-publicized middleware initiatives are the SUN’s Java 2 Enterprise Edition (J2EE), Open Software Foundation's Distributed Computing Environment (DCE), Object Management Group's Common Object Request Broker Architecture (CORBA), and Microsoft's COM/DCOM (Eckerson, 1995). IBM referred to middleware as “the sweet, nougaty center of infrastructure”. Internet2's middleware site calls it the glue between the network and applications. In general, it is software that users rarely see as it usually does not have a user interface. It relies heavily on open standards (such as LDAP and XML), and facilitates the exchange of data between systems.

1.6 Problems and Challenge

At present, geospatial sensor data harvest, discovery and retrieve face the following core problems:
1) Recently a few search engines that are specialised with respect to geographic space have appeared. However, users do not always get the effective OGC link information they expect when searching the Web. How to improve the precision of service discovery is a big challenge.
2) The interfaces and protocol of sensor data service mode and sensor Web data service mode are too complexity to confuse the users.

1.7 Organization

To address the above issues, this paper presents a data discovery and retrieve service middleware based on service harvest, discovery and retrieve in section 2, including components and interaction protocol. Geospatial search engine is wrapped in the harvest server to improve the precision of service discover. The three kinds of implementation approaches in DDRS are detailed in section 3. Finally, section 4 summarizes this paper and presents the outlook for the future.

2. ARCHITECTURE

2.1 Architecture

The middleware services architecture for the DDRS consists of the following four key servers and a service DB: a harvest server, a discover server, a retrieve server and a portal server. Figure 1 shows its architecture and interaction.

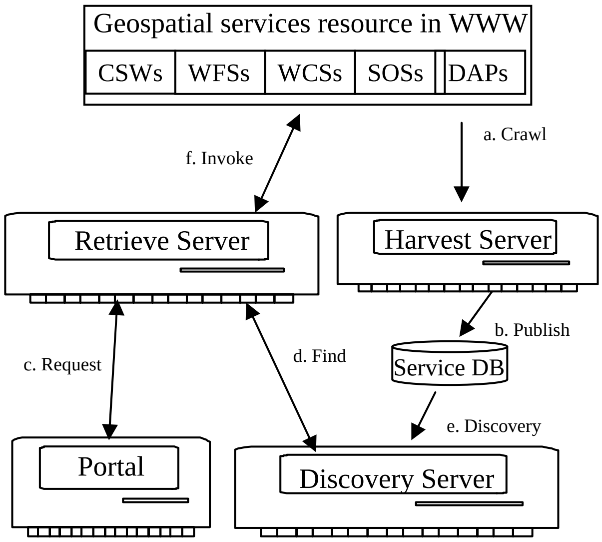

Figure 1. Architecture of DDRS

Geospatial service resources in WWW include two types of distributed data sources: geospatial sensor data and geospatial Sensor Web data. Geospatial sensor data sources require users to send the WCS, WFS or OPeNDAP request to the existing geospatial database. Geospatial Sensor Web data sources refer to the case where users request data that does not yet exist, but whose creation can be scheduled by the Sensor Planning Service in an SWE-driven system and which is accessed in real-time through a Sensor Observation Service.

The harvest server is a virtual data service provider. It is in charge of harvesting geospatial data service resources from World Wide Web (WWW) using search engine and CSW harvest interface. The search engine is used to find the geospatial data service URL, and the CSW harvest is used to register the geospatial data service as a “ServiceType” object in geospatial services metadata database. The geospatial services database server is used to store and manage the metadata of diverse geospatial data services.

The sensor data discover server supports discovery of geospatial data services. It is the broker of DDRS. The information model used in the server integrates those from ISO 19115-geospatial metadata, ISO 19115.2-geospatial metadata extensions for imagery and gridded data, ISO 19119-geospatial services, and ISO 19130-Sensor and Data Models.

The retrieval server is capable to retrieve data from different data service sources. Different plug-in Web services are developed to support the retrieval of data through CSW, WCS, WFS, SOS and OpenDAP. Within the retrieval server, it has a CSW, a WCS and WFS enabled proxy service. A proxy service is to enable the integrated retrieval of data from internal services and external services. The proxy service dynamically supports the following conversions: WCS server from different sources, different versions and real time observation and measurement; CSW server from different sources and versions; WFS server from different sources, different versions and real time observation and measurement.

The geospatial data service portal is the presentation logic. It interacts directly with the retrieve server, consisting of user authentication, service search, contents management and data visualization modules. Client applications invoke a core service component through the portal to achieve the corresponding authentication, search, management and visualization function.

2.2 Interaction

The interaction diagram is shown in figure1. It shows that, a harvest server communicates with www and crawls the service URL from the www and publishes the service to the CSW server; a portal server communicates with a retrieve server to access the service; the retrieve server communicates with the discovery server to find a service instance and invoke a specified service.

2.3 Interactions between a harvest server and WWW

The "a.Crawl" and "b.Publish" in figure 1 shows how a producer of geospatial data interacts with a harvest server that supports the transactional profile. The harvest server searches the www and registers the service instance as a "serviceType" object to the CSW database using CSW publish operation. For example, the harvest server invokes the "Crawl" operation on a specified URL to discover an SOS. When the producer determines that the SOS is communicating with a new sensor, the harvest server invokes the "RegisterSensor" operation and registers the new sensor with the SOS. The producer invokes the "InsertObservation" operation and publishes sensor observations from all sensors of the SOS. The SOS instance then invokes the appropriate adapter and updates the corresponding database.

2.4 Interactions between a retrieve server and a Portal server

The "c.Request" in figure 1 shows how a portal server interacts with a retrieve server that supports the service consume.The EO data retrieval server is responsible for observational data obtained through the uniform standard interface and operation. Three standard interfaces and operations are exposed to the portal: the "GetRecords" operation in the CSW service, the "GetCoverage" operation in the WCS service and the "GetFeature" operation in the WFS service.

2.5 Interactions between a retrieve server and a discover server

The "d.Find" and "e.Discovery" in figure 1 shows how a retrieve server interacts with a discovery server that supports the service discover. A Sensor data consumer (retrieve server) discovers geospatial service instances from a CSW catalogue database (services database) using the "GetRecords" operation of discover server. The consumer then obtains the service-level for each service instance by requesting the capabilities document and inspecting the contents.

2.6 Interactions between a retrieve server and a geospatial data service

The "f.Invoke" in figure 1 shows how a retrieve server interacts with a geospatial data service that supports the service invocation. For example, for the SOS, the retrieve server invokes the "DescribeSensor" operation to retrieve detailed sensor metadata for those sensors advertised in the observation offerings of the identified SOS instances. This metadata is in SensorML or TML. The consumer selects an appropriate SOS and binds the SOS server to perform "GetObservation" operation. Finally, the consumer calls the "GetObservation" operation to actually retrieve the observations and serves the O&M data as a coverage or a feature to the consumer using a WCS-T or WFS-T server.

3. IMPLEMENTATION

A service is a self-contained, stateless function which accepts one or more requests and returns one or more responses through a well-defined, standard interface. Services can also perform discrete units of work such as editing and processing a transaction. Services should not depend on the state of other functions or processes. The technology used to provide the service, such as a programming language, does not form part of this definition. A Web Service is a programmable application logic accessible using standard Internet protocols. Web Services combine the best aspects of component-based development and the Web. Like components, Web Services represent black-box functionality that can be reused without worrying about how the service is implemented. Unlike current component technologies, Web Services are not accessed via object-model-specific protocols, such as DCOM, RMI, or IIOP. Instead, Web Services are accessed via ubiquitous Web protocols (ex: HTTP) and data formats (ex: XML).

The DDRS middleware is implemented by the web service and J2EE technology. The harvest service is implemented using the combination of geospatial search engine and OGC CSW publish interface; the retrieve service is implemented using schema mapping and matching.

3.1 Implementation of harvest service

A high precision OGC Web map service retrieval based on capability aware spatial search engine is put forward by (chen, 2007). The precision and response time of WMS retrieval was evaluated, results showed that the mean execution time during per effective hit of proposed method is 0.44 times than that of traditional method, moreover, the precision is about 10 times than that of traditional method. WMS ontology record could be generated by ontology reasoning, registered and served by CSW. In this paper, we extend this approach to crawl, detect, and register the geospatial service including WMS, WCS, WFS and SOS. The procedures of the geospatial data service crawl and detect are listed in the following:

1) Crawl: We use the popular open source search engine (such as Nutch) to track every known page and relevant links, generate the URL database from the specified URL links.

2) Query: Once we have the web content, we can get ready to process queries. The indexer uses the content to generate an inverted index of all terms and all pages. We divide the document set into a set of index segments, each of which is fed to a single searcher process. For example, we use the “WFS” and “Web Feature Service” key words to query the indexed web content, the potential “WFS” URL database can be achieved.
3) Parse: The potential “WFS” URL database has the links whose content has the “WFS” and “Web Feature Service” key words. The content is parsed using html document parser, some “WFS” relevant links can be found and stored in “Potential URL Database”.
4) Detect: We send get or post “WFS GetCapabilities ” request to the above “WFS” links and get the response. We can get the URL and metadata of “WFS” if the response contains “WFS_Capabilities” element and the corresponding information.
5) Combine: The above “WFS” URLs are compared with each other, a uniform “WFS” URL database is generated.
6) Generate: Each WFS service is registered as a service record in OGC CSW, the record is generated through the capability of WFS.

3.2 Implementation of retrieve service

3.2.1 Methodology

Just as figure 2 shows, three kinds of services are concerned in the DDRS. They are same version homogeneous service, different version homogeneous service and heterogeneous service. The same version homogeneous service has the same information model and schema, the different version homogeneous service has the same information model but different schema, the heterogeneous service has the different information model and schema.

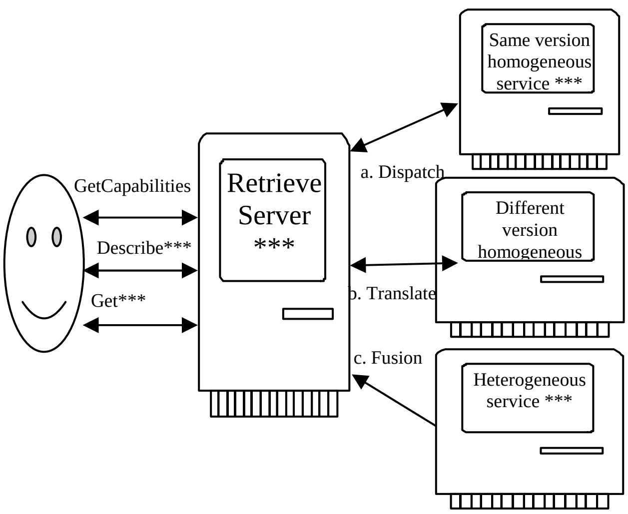

Figure 2. three kinds of approaches for service retrieve

Meanwhile, three kinds of approaches are used to handle service retrieve. They are directly request and response dispatch (a.Dispatch), dynamic schema mapping (b.Translate), and information model matching (c.Fusion).

The translate process consists of two phases. In the first phase, the high-level specifications, expressed as a set of inter-schema correspondences, are translated into a set of mappings that capture the design choices made in the two schemas. The design choices include the hierarchical organization of the data as well as schema constraints (i.e., foreign key constraints). During the second phase, these mappings are translated into queries (SQL, XQuery, or XSLT) over the source schema that generate data to populate the target schema. An important feature of the mapping algorithm is that it takes into consideration target schema constraints in order to guarantee that the generated data will not violate the integrity of the target schema.

For the heterogeneous service, besides the schema translation of request, the fusion of different information model of response must be conducted before the schema translation. For example, the fusion between O&M resultModel and GML WCS profile or GML Feature profile, the metadata of real-time observation generated from a SOS is registered as an “objectType” in the CSW server. If the observation result is coverage information, metadata “WCSCoverage” is generated; if the observation result is feature information, metadata “WFSLayer” is generated. “WCSCoverage” contains the dataset name, its geographical extent, data format, spatial reference and resolution information, “WFSLayer” contains the dataset name, its feature type, its geographical scope, and spatial reference information.

3.2.2 Case study

Just as figure 3 shows that the CSW 2.0.0 service is the same version homogeneous service to the CSW 2.0.0 service, the CSW 2.0.1 or CSW 2.0.2 service is the different version homogeneous service to the CSW 2.0.0 service, and the THREEDS service or ECHO service is the different version homogeneous service to the CSW 2.0.0 service. So, DDRS uses “dispatch” approach to retrieve CSW 2.0.0, “translate” approach to retrieve CSW 2.0.1 or CSW 2.0.2 and “fusion” approach to retrieve THREEDS or ECHO.

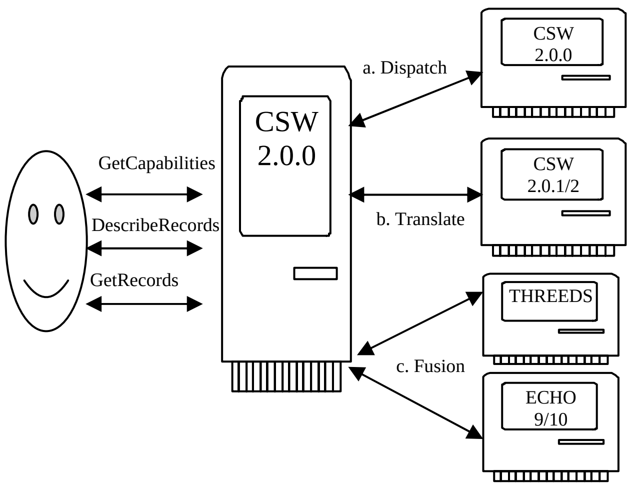

Figure 3. Uniform CSW server for multi data source

Just as figure 4 shows that the WFS 1.0.0 service is the same version homogeneous service to the WFS 1.0.0 service, the WFS 1.1.0 service is the different version homogeneous service to the WFS 1.0.0 service, and the SOS 1.0.0 service or OpenDAP service is the different version homogeneous service to the WFS 1.0.0 service. So, DDRS uses “dispatch” approach to retrieve WFS 1.0.0, “translate” approach to retrieve WFS 1.1.0 and “fusion” approach to retrieve OpenDAP or SOS 1.0.0.

Just as figure 5 shows that the WCS 1.0.0 service is the same version homogeneous service to the WCS 1.0.0 service, the WCS 1.1.0 service is the different version homogeneous service to the WCS 1.0.0 service, and the SOS 1.0.0 service or OpenDAP service is the different version homogeneous service to the WCS 1.0.0 service. So, DDRS uses "dispatch" approach to retrieve WCS1.0.0, "translate" approach to retrieve WCS1.1.0 and "fusion" approach to retrieve OpenDAP or SOS 1.0.0.

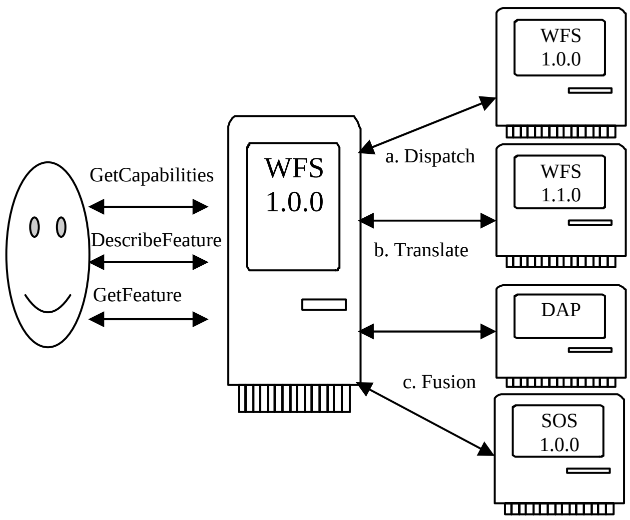

Figure 4. Uniform WFS server for multi data source

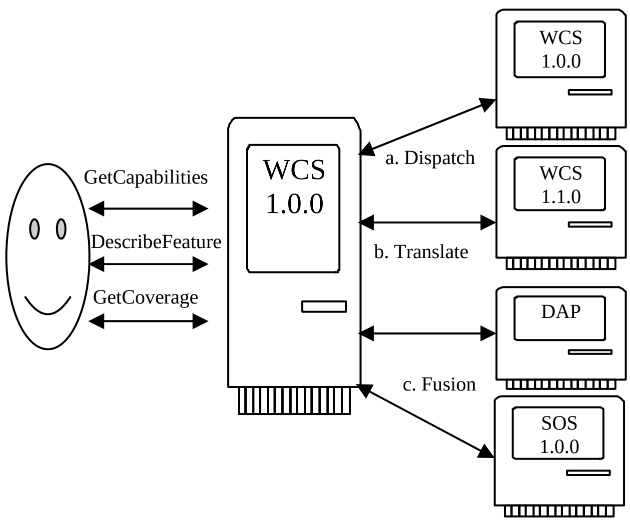

Figure 5. Uniform WCS server for multi data source

4. CONCLUSIONS AND OUTLOOK

How to improve precision of the geospatial service discovery and simply the service retrieve is a big challenge in the interoperability and system integration. We proposed to answer the above question by building a catalogue service through a harvest service based on search engine and multi protocol data retrieval service middleware based on schema mapping and matching, called DDRS. DDRS consists of a harvest server, a discover server, a retrieve server and a portal server. The dispatch, translate and fusion approaches are used to handle the uniform retrieve of homogeneous and heterogeneous service.

Both the observational data harvest, discovery, retrieve and presentation services use service-oriented architecture. The entire process is implemented by plug-and-play components, can be packaged into Web services, and achieved better performance by the service components optimum combination.

DDRS hides the complexity of the geospatial world behind its standard interfaces: For sensor data services, WFS, WCS and OPeNDAP are used to retrieve feature type data, Grid coverage data and Earth scientific data in retrieve server; For Sensor Web data sources, the sensor observation service is adopted to retrieve observations and measurements encoded using O&M specification in retrieve server. Multiple versions of the data services, such as WFS 1.0.0, WFS 1.1.0, WCS 1.0.0, and WCS 1.1.0 can be discovered and retrieved in retrieve server; Three standard interface WCS 1.0.0, WFS1.0.0 and CSW 2.0.0 are exposed to portal server.

The next step is to study how to use sensor web technology to provide self-adaptive geospatial data service.

REFERENCES

Botts, M. (Ed.), 2006. OpenGIS® Sensor Model Language Implementation Specification (Version 1.0). OGC Document Number: 05-086r2, 117pp.

Beaujardiere, J. (Ed.), 2006. OpenGIS® Web Map Server Implementation Specification (Version 1.3.0). OGC Document Number : 06-042, 85pp.

Chen, A., Di,L., Wei,Y., Liu,Y., Bai,Y., Hu,C., Mehrotra,P., 2005. Grid enabled Geospatial Catalogue Web Services. In: American Society for Photogrammetry and Remote Sensing (ASPRS) 2005, 7-11 Mar, Baltimore, USA.

Chen N., Gong J., Chen Z., 2007. A high precision OGC Web map service retrieval based on extended search engine and capability match. LNCS ,Vol.4683,558-567.

Cox, S. (Ed.), 2006. OGC™ Observation and Measurement (Version 0.13.0). OGC Document Number: 05-087r3, 136pp.

Di, L., 2007. Geospatial Sensor Web and Self-adaptive Earth Predictive Systems (SEPS). In: ESTO/AIST Sensor Web PI Meeting 2007, 13-14 February, San Diego, USA.

Delin, K., Jackson, S., 2001. The Sensor Web: a new instrument concept. In: Proceedings of the SPIE International of Optical Engineering, vol. 4284, pp. 1－9.

Eckerson, Wayne.1995. Searching for the Middle Ground. Business Communications Review *25,* 9 (September 1995): 46-50.

Evans, J.D. 2003. OGC™ Web Coverage Service Specification (Version 1.0.0). OGC Document Number: 03-065r6, 67pp.

Havens S. (Ed.), 2006. OpenGIS® Transducer Markup Language Implementation Specification (Version 1.0. 0). OGC Document Number: 06-010r2, 136pp.

Lee E., Kim M., Kim M., Joo I., 2005. A web services framework for integrated Geospatial coverage data. Journal of Lecture Notes in Computer Science 3480,1136-1145.

Liang, SHL., Croitoru, A., Tao, CV. A distributed geospatial infrastructure for Sensor Web. COMPUTERS & GEOSCIENCES, 2005, 31 (2): 221-231.

Na A., Priest M. (Eds.), 2006. OpenGIS® Sensor Observation Service Implementation Specification (Version 0.1.5). OGC Document Number: 06-009r1, 187pp.

Nebert, D., Whiteside, A. (Eds.), 2005. OGC™ Catalogue Services Specification (Version 2.0.0). OGC Document Number: 04-021r3, 187pp.

Simonis I., Wytzisk A. (Eds.), 2003. Web Notification Service (Version 0.1.0). OGC Document Number: 03-008r2, 46pp.

Simonis I. (Ed.), 2005. OpenGIS® Sensor Planning Service Implementation Specification (Version 0.0.30). OGC Document Number: 05-089r1, 152pp.

Simonis I., Echterhoff J. (Eds.), 2006.OGC® Sensor Alert Service Implementation Specification(version 0.9.0).OGC Document Number: 06-028r5,144pp.

Voges, U., Senkler, K. (Eds.), 2005. OpenGIS® Catalogue Services Specification 2.0—ISO19115/ISO19119 Application Profile for CSW 2.0 (Version 0.9.3). OGC Document Number: 04-038r2, 89pp.

Vretanos, P.A. (Ed.), 2002. OGC™ Web Feature Service Implementation Specification (Version 1.0.0). OGC Document Number: 02-058, 105pp.

ACKNOWLEDGEMENTS

This work is supported by grants from Chinese 863 project (No. 2007AA12Z230, PI: Dr. Nengcheng Chen), NSFC project (No. 40501059, PI: Dr. Nengcheng Chen) and grants from U.S. NASA ESTO/AIST Sensor Web program： A General Framework and System Prototypes for the Self-Adaptive Earth Predictive Systems (SEPS)--Dynamically Coupling Sensor Web with Earth System Models (No. AIST-05-0064, PI: Dr. Liping Di).