GeoBrain-A Web Services based Geospatial Knowledge Building System

Liping Di
Laboratory for Advanced Information Technology and Standards (LAITS)
George Mason University
9801 Greenbelt Road, Suite 316-317
Lanham, MD 20706
ldi@gmu.edu

***Abstract* – Huge volumes of geospatial data have been collected by both public and private sectors. In order for the geospatial data to be useful, information has to be extracted from the data and converted to knowledge. However, currently it is very difficult for most of users to obtain the geospatial data and to turn them into useful information and knowledge. In order for geospatial information and knowledge to become the mainstream ones so that everyone can obtain and use them at will, geospatial knowledge building systems are needed urgently to automate the path from geospatial data to information and knowledge. This paper presents GeoBrain, a prototype geospatial knowledge building system currently under development. GeoBrain is a three-tier standard-based open geospatial web service system which fully automates data discovery, access, and integration steps of the geospatial knowledge discovery process under the interoperable service framework. It also automates a range of geo-computational services at a limited number of geospatial domains and greatly facilitates the construction of complex geocomputation services and modeling. The paper discusses the geo-object and geo-tree concepts that the system is based on, the system architecture, the interoperable geospatial web service framework, and the individual components of the system. The paper also discusses the use of the system in NASA EOSDIS data environment to make peta-bytes of NASA EOS data and information, especially those in ECS data pools, easily accessible by broader user communities.**

I. Introduction

One of the biggest assets of NASA ESE is the multi-petabytes of geospatial data, products, and information collected by its EOS program. In order to process, archive, manage, and distribute the data, NASA has invested large amounts of money and resources to develop the EOS data and information system (EOSDIS). In order for the geospatial data to be useful, information has to be extracted from the data and converted to knowledge. However, currently it is very difficult for scientists and general users to obtain the geospatial data and to turn them into useful information and knowledge. Therefore, geospatial knowledge building systems are needed urgently to automate the path from geospatial data to information and knowledge. This paper discusses a prototypical system called GeoBrain that is currently in development.

This project is funded by NASA REASoN Program

II. Problems With Current Data Systems

Remote sensing data are widely used in many aspects of socio-economic activities ranging from environmental monitoring to military operations. The EOS data are especially attractive and useful because of diverse data sources, multi-disciplinary coverage, science-oriented data collection, and no or low cost of the data. However, there are significant problems associated with uses of EOS data in the research and applications, including 1) difficulty to access to the huge volume of EOS data; 2) difficulty to use the data; and 3) lack of enough resources to analyze the data.

Currently, typical steps for a user to obtain EOSDIS data for applications are as following:

1. A data user goes to the EOS Data Gateway (EDG) to search and find the data they want.
2. The user places an order to the data. The minimum size of orderable data in EOSDIS is granule, which may cover larger/or smaller geographic area than the user wants.
3. A NASA data center (DAAC) takes the order, and pulls out the data granules from the archive or copies the granules from on-line devices to distribution media, such as CD-ROM and tapes, or stages the data in a FTP site for retrieval. Normally, DAAC distributes the data in whole granule in the archival form. Services aren't normally provided to transform data from archival to user-specified form.
4. The user obtains the data and then does the preprocessing to transform the data to the user requested form, so that multi-source data can be in the same format, map projection, spatial and temporal coverage, spatial and temporal resolution for the integrated analysis.
5. The user performs the analysis and modeling of the multi-source datasets.

The process for a user from ordering to actually obtaining the data usually takes weeks. Therefore, many applications requiring real or near-real time data cannot be conducted. Second, because users cannot get the data in user-specified form, they need to spend a lot of time and resources to pre-process the data. The pre-process

normally reduces the size of data significantly. For example, if a user wants to obtain MODIS level 1b datasets that cover the state of Virginia, what the user actually gets from DAACs are MODIS granules covering almost a half of USA. Therefore, the user has to subset the data by himself. Another obstacle is the data format. NASA processes and archives EOS data in the HDF-EOS format because the format is powerful and flexible, and normally DAACs distribute data only in this format. However, only a few existing analysis software packages can ingest the data and are general enough to handle small structure difference in HDF-EOS datasets produced by different producers. This added another layer of difficulty for people, especially for those who are not very familiar with data formats and format conversions. Other problems include the difference in the map projection, the difficulty in georectification of swath data, etc.

In addition, global change studies often deal with global-scale issues and require analyzing large volumes of multi-disciplinary data. Most of scientists don't have enough local computer hardware and software resources to handle the huge data volume. Adding all those problems together, a typical process from step 1 to 5 will take more than 1 month to complete. If a researcher wants to modify models or to apply models to new data, they need to go through the steps again. The scientific productivity is significantly hampered by the current system.

III. Objectives and Expected Significance

The goal of this project is to solve the problems by developing a next generation geospatial information system, called **GeoBrain.** It is an open, interoperable, distributed, standard-compliant, multi-tier web-based geospatial information services and knowledge building system. This system will shorten the time required for going through steps 1 to 5 of EOS data analysis from weeks to just *minutes or seconds*. At the back-end, the system will access the multi-petabytes of EOS data at NASA data pools running at DAACs. At the front-end, end-users will be able to find and obtain data in the form that exactly matches their requirements so that the data are ready for analysis. In addition to obtaining the data with great easiness, the system will also provide an interoperability framework allowing end-users to develop individual geospatial web-service modules and to use these modules and those provided by GeoBrain for constructing complex web-executable geospatial models. By executing the models on the system on-line against any subsets of the petabytes of EOSDIS data in the data pools, end-users obtain not the raw data but solutions to their scientific questions. Through the proper peer review, the user-developed modules and models can be plugged into the system to become its operational capabilities. To other end-users, the newly available web-executable models represent the types of geospatial products available at the system although the products will only be produced when users request them. These models and modules can be reused to construct even more complicated geospatial models. This accumulation of knowledge through sharing and reuse of geospatial process models will make the system evolvable and increasingly capable with time. The framework also makes the idea of interoperable systems developed by a community for the community a reality.

With the successful implementation of the GeoBrain system, it will be able to provide the following products and services to EOS data users:

1) All ECS data pool products tailored to individual users' needs in terms of spatial and temporal coverage, map projection, format, bands, etc so that the data are ready to analyze.
2) The virtual geospatial data and information products produced on-demand by executing the peer-reviewed web-executable geospatial process models on the on-line EOS data.
3) The customized information products produced by web-executable geospatial process models running on GeoBrain with on-line EOS data. The model is created interactively by end-users with web services modules and models available in GeoBrain.

With the GeoBrain system, users can explore easily the large volume of ECS data in data pools through their Internet-connected desktop computers just as if they have both a huge, homogenized database and a powerful analysis system at their desktop. The fast Internet connection to end-users' desktop is not necessary because all data pre-processing and reduction are done at server side and the users will obtain solutions instead of raw data. This unique ready-to-explore geospatial data-rich environment is valuable to EOS data users.

IV. The Geo-object and Geo-tree Concepts

The GeoBrain system is being built on the concepts of geo-object and geo-tree. The concepts were first introduced in 1997 in an internal NASA discussion on the next generation of EOSDIS and were published in 1999 [1]. First, we consider a granule of geoinformation (either a dataset, a query result, or geo-computation output which describes some aspects of Earth) to be a *geo-object*, which consists of data itself, a set of attributes (metadata), and a set of methods (transformation and creation methods) that can operate on it. A geo-object stored at a data center is an *archived geo-object*. All geoinformation and knowledge products are derived from archived geo-objects. Thus, from object-oriented point of view, all processes for geo-information/knowledge discovery are the processes of creating new geo-objects from existing ones.

If we consider a user request is a user-defined geo-object, or called *user geo-object*, the object is either an archived geo-object in a data archive or can be derived by executing geo-processing algorithm (e.g., unsupervised

classification) with a set of input geo-objects. An input geo-object, if not exists in an archive, can be further derived by executing a geo-processing algorithm with a set of input geo-objects and so on. The decomposition of user geo-object will construct a tree structure representing the process workflow, which is called a *geo-tree*. The construction of a *geo-tree* is a geospatial modeling process; and the geo-tree itself is a geospatial model that contains the knowledge of a specific application domain.

With a geo-tree, we know how to produce the user-object specified by the tree although the object does not really exist in any archives. We call such geo-object the *virtual geo-object*. In fact, any sub-tree in the geo-tree is a virtual geo-object. Since a geo-tree only captures the process steps, not a specific product, it represents a type of geo-objects that it can produce, not an instance (an individual dataset). The virtual geo-object can be materialized on-demand for users when the system has all required methods and inputs available. When user requests such an object, the user has to specify the geographic location, time, format, etc. Those specifications will instantiate the virtual geo-object. By propagating the specifications down to each node of the geo-tree, the whole geo-tree is instantiated. This process is called *instantiation of geo-tree*. Only by doing the instantiation we can know if the virtual geo-object can be materialized because in many cases the required archival geo-objects may be not available for the users-specified geographic region and conditions. After the instantiation, the geo-tree is executable in the system and the virtual geo-object can be produced. The production processing is called *the materialization of virtual geo-object*, which will produce an instance of the virtual geo-object.

The geo-processing algorithm (method) in each node of a geo-tree is called a *geospatial service module*. The algorithm may only take care of a tiny part of overall geo-process or may be a large aggregated process. However, the service should be well defined, and has a clear input and output requirements, and can be executed independently. If a geospatial service is implemented in the web environment, it is called a *geospatial web service module*. All those service modules can be reused in constructing different geospatial models. If we have enough elementary service modules available, we can construct any complex geospatial models. In a federation of distributed data and information systems, there are many independent data and service providers and subsystems. Services needed for producing a virtual geo-object may be scattering in multiple service providers. Therefore, we need standards to publish, find, binding and execution of services. The *geo-tree* concept requires one service's output to be the input of another service. Standards on service chaining are required. Also a service in GeoBrain may require inputs from data providers other than NASA ECS Data Pools. A *common data environment* providing standard interfaces to the data provider's archives is required.

From service point of view, a geo-tree is a complex *service chain.* A service chain is defined as a sequence of services where, for each adjacent pair of services, occurrence of the first action is necessary for the occurrence of the second action. When services are chained, they are combined in a dependent series to achieve larger tasks. The construction of geo-tree is a service-chaining process. There are three types of chaining defined in ISO 19119 and OGC [2]:

- User-defined (transparent) – the Human user defines and manages the chain.
- Workflow-managed (translucent) – the Human user invokes a service that manages and controls the chain, where the user is aware of the individual services in the chain.
- Aggregate (opaque) – the Human user invokes a service that carries out the chain, where the user has no awareness of the individual services in the chain.

GeoBrain implements both transparent and translucent chaining methods. The opaque chaining, which requires artificial intelligence, will be studied in this project.

V. The GeoBrain Implementation

As discussed in the previous section, standards play an important role in the construction of GeoBrain. The system uses the Open GIS Consortium (OGC) standards for the data finding and access, OGC and W3C standards for the web services. It also leverages the new and ongoing development in the knowledge representation and management, especially the workflow management.

The reasons to choose OGC standards for system interoperability is that OGC is the only international organization dedicating to develop implementation standards for geospatial interoperability based on ISO, FGDC, INCITS, and other organizations' abstract or content standards. OGC standards are fully tested both in NASA's and other data providers' environments. OGC specifications are widely used by geospatial communities for sharing data and resources and are becoming ISO standards. For standards that are not available at OGC, GeoBrain will W3C and OASIS web services standards because it is basically the geospatial version of web service system.

Figure 1 shows the GeoBrain architecture. As shown in the figure, GeoBrain is built upon the open, consensus-based standards that will allow other foreign systems to interoperate with it through standard interfaces. GeoBrain is a three-tier system that includes the interoperable data server tier, the middleware geospatial services and knowledge management tier, and the integrated geoinformation client tier. In the figure, all standard system-wide interfaces are shown as white arrows. Any other systems using the same standard interfaces as

GeoBrain can be interoperable and federal-able with GeoBrain. All other types of arrows represent either private/internal interfaces or user community defined interfaces. The size of arrows represents the size of data traffics. Since the system provides customized data product and information to the end-users by performing the data reduction and processing at the data sever and middleware tiers, the size of data traffics at the end-users side is much smaller than those at data server side.

V. 1 The Interoperable Data Provider Tier

This tier consists of data servers providing data to the geospatial service middleware, the application and data analysis systems, application clients, and human users (called *requestors* hereafter) through a common data environment. The NASA Web GIS Software Suite (NWGISS) data servers [3] will be reused in the GeoBrain data provider's tier. *The common data environment* is a set of standard interfaces for finding and access data in diverse data archives, ranging from small data providers to multiple-petabyte EOSDIS. The environment allows geospatial services and value-added applications to access diverse data provided by different data providers in a standard way without worrying about their internal handling of data.

The interface standards for the common data environment are OGC Web Data Services Specifications, including Web Coverage Services (WCS) [4], Web Feature Services (WFS) [5], Web Map Services (WMS) [6], and Web Registries Services (WRS) [7]. The specifications allow seamless access to geospatial data in a distributed environment, regardless the format, projection, resolution, and the archive location.

V.2 The Middleware Geospatial Service and Knowledge Management Tier

The middleware tier consists of multiple components that perform geospatial data processing, information extraction, and knowledge management. Figure 2 shows possible OGC adoption of W3C standards for geospatial web services. The standards are used in the development of the middleware tier. The following paragraphs provide detail descriptions.

Geospatial service module warehouse: This warehouse contains individual geospatial web services executables. Geospatial web services are self-contained, self-describing, modular applications that can be published, located, and dynamically invoked across the Web. Once a service is deployed, other applications (and other Web services) can discover and invoke the deployed one. The GeoBrain service modules are developed based on the OGC web service standards currently under development.

The service modules, in GeoBrain, act as individual building blocks for dynamically constructing complex geoprocessing models represented by geo-trees. The richness of the service modules will, in certain extent, decides the capability of the in-house process power of GeoBrain system. However, the GeoBrain's power is not limited by those service modules. In an interoperable federated environment, the service modules existing in different members of the federation can and will be shared. Therefore, a complex geospatial model (geo-tree) can be built by service modules in all member systems and can be executed across the member systems. This is the power of the standards and federation.

Geospatial model/workflow warehouse: This warehouse will contain geo-trees that describe geospatial models encoded in a workflow language. The geo-trees capture the knowledge of the geospatial process and modeling. In other aspect, each geo-tree represents the type of virtual geospatial products the GeoBrain can serve. The workflow language to be used is BPEL4WS. The geospatial models in the warehouse come from two sources; the first one is from the GeoBrain development team, and the second one is from the user community. GeoBrain will become more and more knowledgeable and capable when more and more models are included in it.

Virtual data type/workflow manager: This component serves two functions: to external users it is an OGC WRS server that help requestors to find both data instances and data types. Internally, it manages the model warehouse, works as the client of WRS servers in data pools and other real data providers, and passes the instantiated geo-tree to the workflow execution manager to execute. As we said before, a geo-tree in the warehouse represents a type of virtual data product that GeoBrain can generate on demand. And each sub-tree of a geo-tree represents a type of virtual data, except for those leaf nodes.

Suppose that we have a ground-water pollution potential geo-tree in the warehouse, the difference between type query and the instance query will be distinguished by the following query examples. If a requestor asks if GeoBrain can serve ground-water pollution potential map, it is a type query. The GeoBrain will return with yes and provide detailed metadata information about the geo-tree. However, if a requestor asks if GeoBrain serves the ground-water pollution potential map for the State of Maryland at 30-meter spatial resolution, this is a query for an instance. For this query, the manager will first propagate the spatial coverage and resolution requirements to each node of the tree, and check if all inputs to the service modules in the tree can be instantiated by issuing the WRS requests to data servers' data catalog to find if the data are available. If all required inputs are available, then manager will answer yes to query and return an ID and ancillary information for requestor to retrieve the data. If the requestor also requests the metadata, the details about the geo-tree will also be available to the requestor.

There is no difference between real and virtual data from the point of view of a requestor, because both virtual and real data will be requested and retrieved in the same way. In a federation environment, the virtual data generated

from GeoBrain can be an input to another tree resided in different system.

The model/workflow execution manager acts as a WCS, WFS, or WMS server to requestors, depending on the type of requested geoinformation. If the requested geoinformation is not virtual one, the manager retrieves the data from the data provider server and deliver to the requestor. If it is virtual, the manger manages the execution of the workflow and deliver the materialized virtual geo-object to the requestor. Basically, the execution manager will act internally to individual web geospatial services as a service requestor which will push the input data to the service and pull out the output from the service. The selection of an execution manager instead of passing the workflow from one service to another for the execution is due to this method allows managing the execution of complex workflows although it will increase the requirement for the network traffic and temporal storage. The execution manager will also manage the status of the workflow execution and the temporal storage. The execution of workflows is provided in two models, the transparent, and the translucent. The translucent is the default execution model, which the manager will control the execution of the workflow. It can be invoked by any OGC compliant WCS, WMS, or WFS client. The transparent model will allow the user control of the execution of the workflow through the GeoBrain Client.

Interactive model/workflow editor server. GeoBrain provides the interactive modeling environment allows users to construct and test their models through GeoBrain client. Models will be constructed graphically in the client. The editor server will provide all available web service modules classified by their service categories, and both virtual data types represented as workflows, and the real data types, to the client. The editor server will allow the user to instantiate, run, modify, debug, save the model and submit it to peer-review server for review.

Service module development environment. The GeoBrain encourage end-users to develop additional service modules by providing the service module development environment. The development environment includes a set of libraries for handling the interface protocols, data encoding and decoding, and general utilities functions. By using those libraries, web-service modules developed by users will be standard-complaint and interoperable. The environment also provides the test space and test run to users.

Peer-review and collaborative development server: In GeoBrain, any user developed web-service modules and the geospatial models are subject to peer review before being inserted into GeoBrain as the operational capabilities. The peer-review server facilitates the process by providing the common environment for reviewers to run and evaluate the submission through the GeoBrain client. The peer-review panel consists of PI, Co-Is, and collaborators of this project and their designated representatives.

Product and service publishing interface: This component will publish the product and services available to external registries so that they also can be found though the registries.

V.3 The Integrated Multiple-protocol Geoinformation Client (MPGC) Tier

GeoBrain will provide an integrated, multiple-protocol geoinformation client to users for free. The client provides not only accesses to all virtual and real data/information provided by GeoBrain and all other OGC-compliant providers but also the geospatial modeling/workflow interfaces, peer-review interfaces, and the collaborative development interfaces.

The Project Component: In MPGC, geospatial information is organized in individual project folders. Each folder contains all necessary data and information for completing a geospatial project. The project can be opened, built, saved, modified, and deleted. A project folder can be created by specifying the geophysical parameters as well as their spatial, temporal, and semantic attributes. The client then sends a search request to GeoBrain's virtual data manager to find data that meet the request, regardless the data is virtual or real. If the GeoBrain is a member of federation, all data available in the federation, regardless virtual or non-virtual, will be searched. Matched data products' ID will be returned to client. The search is an OGC WRS request. It can be sent to any OGC WRS server, not just GeoBrain. The users can add individual data products into the project. If a requested product does not exist anywhere, the user can build a model, through the model development client, to ask GeoBrain to produce it on demand. The model can be added to the project as a virtual data product. When the user requests the GeoBrain client to build the project, the client will send the data requests to the data servers for obtaining the data. The data server may be the virtual data execution manager of GeoBrain that materializes the virtual data, NWGISS data servers, or other OGC compliant servers if a federation is built based on OGC standards. The servers will return the data to client that exactly match users' request. Once all required data are in the project, users can explore, analysis, visualize through MPGC or export to other system. The user community will set the priority for export formats.

The Interactive Geospatial Model Development component works with GeoBrain's virtual data manager, the model editor server, the workflow execution manager, and the peer-review server to dynamically develop, test/execute, modify, save, and submit the model. The client will allow users to develop and test the model graphically. The client will use icons to display all available service modules categorically and allow clicking and dragging a module to the model. All GeoBrain virtual

data types and the real data types will also be displayed graphically, but only virtual data type has an attached geo-tree, allowing users to copy whole or part of a geo-tree and paste it to the model. There are two cases that a geospatial model cannot be built in the GeoBrain, 1) required service modules are not available, and 2) required input data types are not available. Once the model is built, the user can test the model by submitting it to GeoBrain's virtual data manager to instantiate and then to the workflow execution manager to execute. The result will be immediately returned to user for inspection. If the result is satisfied, the users can save the model locally for later use (e.g., insert into projects) or submit it to peer-review server for becoming an operational model.

The multi-source data manipulation component works on the data in the project folder or local data for the manipulation, analysis, and visualization of both vector and raster/grid data seamless.

VII. The Development Approach and Current Status

We develop GeoBrain by using the following principles: 1) component-based development; 2) reuse of the existing components; and 3) adherence to the standards. By adhering to the three principles, we will ensure the GeoBrain system is flexible, widely applicable, reusable, evolvable, and interoperable.

Component-based development. We design the architecture of GeoBrain as modular as possible. Geospatial web services are self-contained, self-describing, modular applications. The service modules are dynamically chained together to form complex geospatial models for produce sophisticated geospatial information products on-demand. The functional sizes of individual service modules are important factor affecting the flexibility, applicability, and reusability of service modules in different geospatial models. If the module's functionality is too small, it needs a lot of modules to construct a complex geospatial model, hence affecting the system performance. If too many functions are aggregated into a service module, the module cannot easily be plugged into many different geospatial models, hence, affecting the flexibility, applicability, and reusability of the module. Research on the modularity of geospatial web services is required

Reuse of existing components. Since the introduction of the geo-tree concepts for geospatial data services, LAITS, at GMU has developed various components reusable in GeoBrain system. The most notable one is NWGISS. It is the only OGC compliant servers and client system in the world that works with all generic HDF-EOS files. Funded by ESTO, ESDISP, and OGC, NWGISS provides interoperable, personalized, on-demand data access and services (IPODAS) to EOSDIS data with built-in georectification, reprojection, sub-setting, resampling, reformatting, and visualization functions [8]. Currently, NWGISS consists of five components: a Map Server, a Coverage Server, a Catalog Server, a multi-protocol geoinformation client (MPGC), and a Toolbox. Because more than 95% of EOS data are coverage data, basically NWGISS servers can be reused as the data server tier of the GeoBrain. In the NWGISS MPGC, the project component has been fully implemented except for the virtual data insertion. The basic functions in the data manipulation component are also implemented and can be reused.

Adherence with standards. Standards implemented in GeoBrain make the system interoperable, reusable, extensible, and evolvable. Because OGC standards are widely used in the geospatial system development, we will adhere to OGC standards in the GeoBrain development. For standards not available in OGC, especially those for workflow management and interoperability, we will test and if necessary extend standards from W3C and other relevant organizations.

Currently, the major development effort for the GeoBrain system is concentrated on the middleware service layer and the MPGC client's model development component. For the service modules, we are working on converting modules in an open source geographic information system called GRASS into web service modules [9]. Two of GRASS modules, the unsupervised classification and the format input/output, have been converted into unsupervised classification service and reformatting service as the prototype. The virtual data type/workflow manager is being implemented by using the OGC CIS-W/WRS specifications. An initial version of model construction client has been developed and simplified BPEL is used as the language for encoding the geo-tree. BPEL is also selected as the workflow language for GeoBrain. The Collaxa BPEL server has tested successfully to execute a manually created BPEL workflow that chains the WCS service with the GRASS-based unsupervised classification services [10]. The execution of the BPEL workflow by Collaxa BPEL server can create any unsupervised classification maps on demand.

VIII. Conclusions and Future Work

Web service technology is a very promising technology for solving problems related to geospatial interoperability and knowledge discovery. GeoBrain will demonstrate this point. The successful use of OGC standards in NASA EOS data environment, as shown in NWGISS, has provided us the experiences needed for implementing the standard-based GeoBrain. The service-oriented architecture (SOA) on that GeoBrain is based, makes the evolvable system possible.

In the next several years, we will concentrate on developing and refining individual components of GeoBrain. It is expected that by end of 2004, a first version

of GeoBrain with essential components built will be available for testing. In order to test GeoBrain in the realistic NASA data environment, we will build a testbed with sizable local data holding. The testbed will also connect to the EOSDIS data pools for accessing the data in the pools.

REFERENCES

[1] Di, L., and K. McDonald, 1999 "Next Generation Data and Information Systems for Earth Sciences Research", in *Proceedings of the First International Symposium on Digital Earth*, Volumn I. Science Press, Beijing, China. 8 pp.

[2] ISO, 2001. ISO 19119:Geographic Information – Services. http://www.isotc211.org.

[3] Di, L., W. Yang, M. Deng, D. Deng, and K. McDonald, 2001. "The Prototypical NASA HDF-EOS Web GIS Software Suite (NWGISS)", Proceedings of NASA Earth Science Technology Conference, College Park, MD. CD-ROM.

[4] Evans JD (ed) (2003) Web Coverage Service (WCS), Version 1.0.0. OpenGIS© Implementation Specification. Open GIS Consortium Inc. http://www.opengis.org/docs/03-065r6.pdf.

[5] Vretanos PA (ed) (2002) Web Feature Service Implementation Specification, Version 1.0.0. OGC 02-058. Open GIS Consortium Inc. http://www.opengis.org/docs/02-058.pdf.

[6] de La Beaujardière J (ed) (2001) Web Map Service Implementation Specification, Version 1.1.1. OGC 01-068r2, Open GIS Consortium Inc. http://www.opengis.org/docs/01-068r2.pdf.

[7] Reich L (ed) (2001) Web Registry Server Discussion Paper, OpenGIS Project Document 01-024r1. Open GIS Consortium Inc., http://www.opengis.org/docs/01-024r1.pdf.

[8] 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. 3pp.

[9] GRASS homepage, http://grass.baylor.edu.

[10] Collaxa BPEL website. http://www.collaxa.com/home.index.jsp.

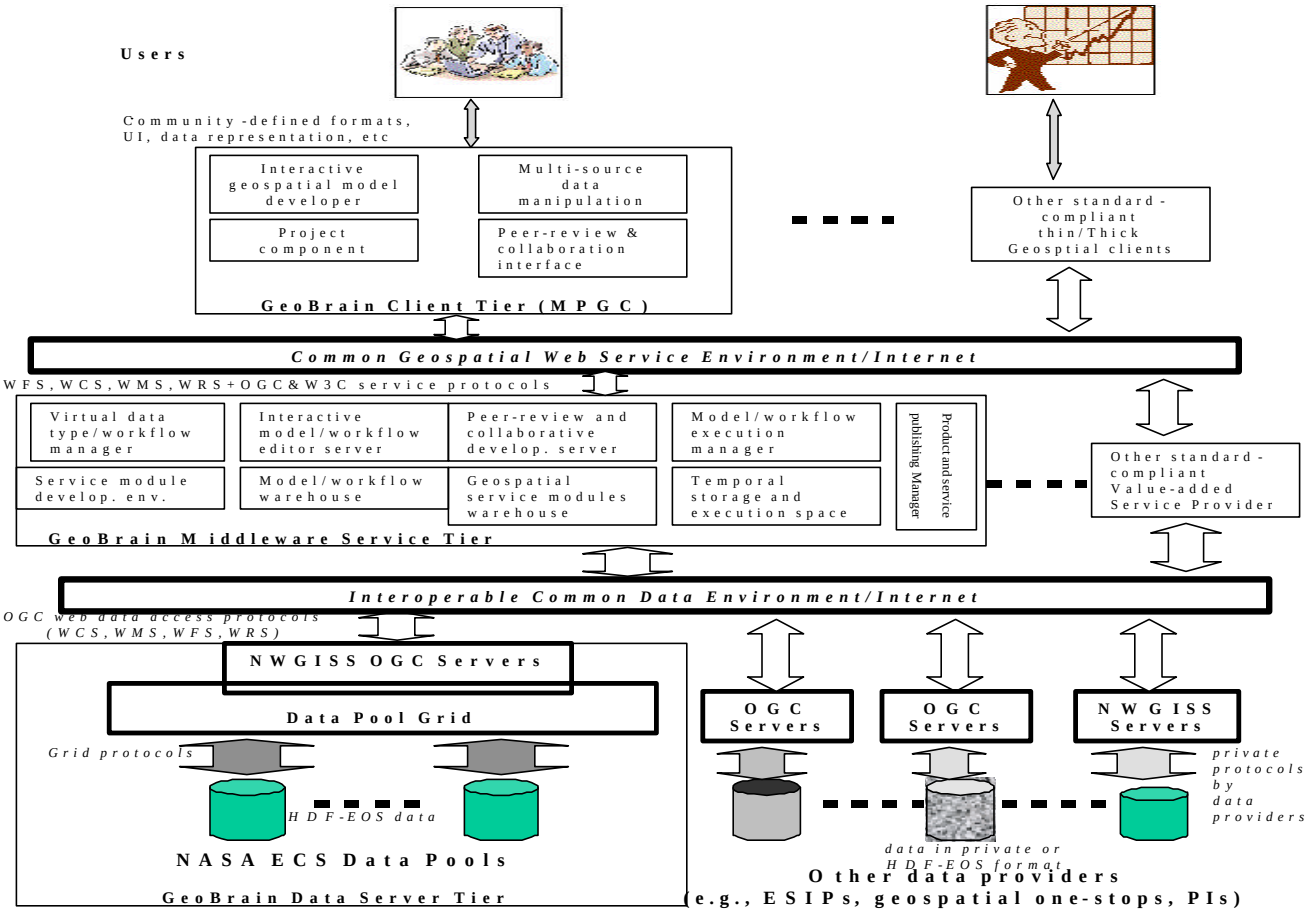

Figure 1. The GeoBrain Architecture

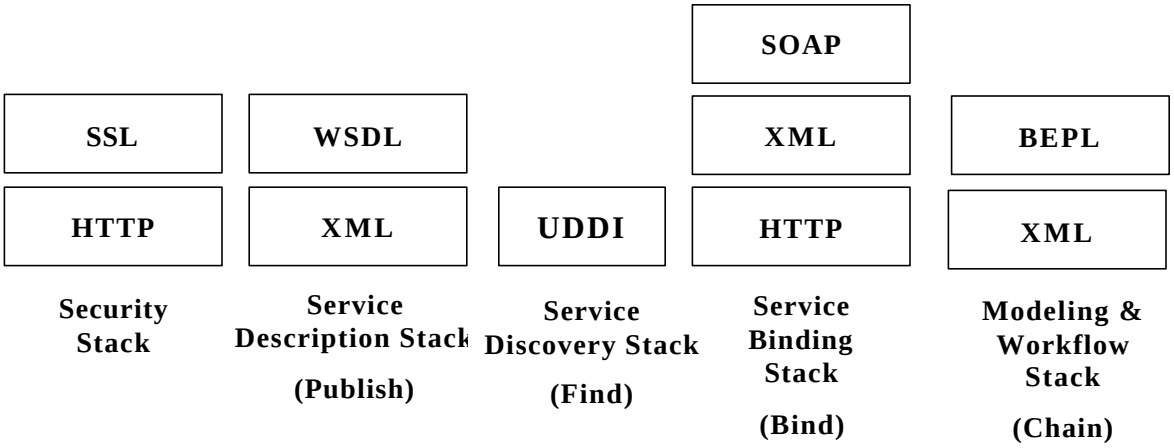

Figure 2. Possible OGC adoption of W3C standards for geospatial web services