Yang and Tao, 2006, Chapter 5 Distributed Geospatial Information Service, Frontiers of Geogrphic Information Technology, Rana S. and Sharma J. (eds.) Springer, pp. 103-120.

Distributed Geospatial Information Service (Distributed GIService)

CHAOWEI (PHIL) YANG
Assistant Professor of Geographic Information Sciences
Earth Systems and GeoInformation Sciences,
School of Computational Sciences, and
Center for Earth Observing and Space Research
George Mason University, 4400 University Drive
Fairfax, VA, 22030-4444
E-mail: cyang3@gmu.edu telephone: 703-993-4742, fax: 703-993-1980

C. VINCENT TAO
Canada Research Chair in Geomatics
Director, Geospatial Information and Communication Technology (GeoICT) Lab
Department of Earth and Space Science and Engineering
York University
4700 Keele Street
Toronto, ON, Canada M3J 1P3
E-mail: tao@yorku.ca telephone: 416-736-5221, fax: 416-736-5817

Abstract

Distributed Geospatial Information Service (Distributed GIService) in this context refers to an emerging paradigm for offering geospatial data and processing services by using distributed computing technologies. Through on-line access and integration of distributed data and geoprocessing services, Distributed GIService provides reduced technology risk, better ability to leverage the value of legacy data and systems, and more efficient information sharing. Development of Distributed GIService is largely driven by 1) the increasing use of service-oriented architecture (SOA), 2) the adoption of interoperable standards for sharing geospatial information resources, 3) the fast development of enabling distributed computing technologies, (e.g., agent, grid, and P2P) for transparent and reliable access to computing infrastructure. This paper provides practitioners with an introduction to Distributed GIService and discusses some fundamental issues in Distributed GIService from a geospatial computing perspective.

Key Words: Geospatial Information Services, Interoperability, GIS, Agent, Grid, P2P, Web Services, GeoComputation, Resource Sharing

1. Introduction

Since its inception in the 1960s, Geospatial/Geographic Information System (GIS) has evolved rapidly and has become part of main stream Information Technology (IT). The projected market size is predicted to grow to $30 billion by 2005 (Gewin 2004). One of the main reasons for this fast-paced development is that GIS continually benefits from the advancement of computing technologies

(Tomlinson 1998). The birth of computer networks in the 1970s, in conjunction with the emergence of the Internet in the 1990s, has broadened the way people share geospatial data and computing resources.

In the computing and geospatial communities, several important observations / issues are evident:

1) Use of Computing resources. It is generally the case that computing capacity is very much under-utilized. According to an IBM survey (Conry-Murray, 2004), only 6%, 60% and 20% of CPU power of desktops, mainframes and workstations, respectively, are utilized. Collectively this represents an immense, untapped pool of computing power.

2) Compatibility of geospatial data. Geospatial data are highly heterogeneous as they have typically been collected using different techniques, in different formats, and with different resolutions, projections, coordinate systems, etc. (Goodchild et al. 1999; Vckovski 1998). This heterogeneity has limited the degree to which data can be shared and merged seamlessly and transparently.

3) Legacy systems. To date there are many legacy systems that were developed in the absence of system and application standards. The resulting lack of interoperability among these legacy systems is a growing concern in the geospatial user community (Tao, 2001).

4) Reliability and scalability. With potentially more and more users concurrently accessing centralized geospatial repositories, reliability and scalability of the centralized repository approach is becoming increasingly difficult.

With the growing use of geospatial data and applications, there is a demand to rapidly develop and deploy new capabilities and new functions based on existing legacy systems (Arsanjani 2004). Given the fact that the technology is evolving very quickly, it is not economical or practical to replace existing legacy systems with completely new components in order to build more comprehensive systems.

Interoperability is recognized as a key solution to the integration of multiple legacy systems. Many organizations, (e.g., OGC[1] and FGDC[2]) have been actively working on developing standards and specifications to facilitate sharing of geospatial data and processing capabilities. The International Standards Organization (ISO) TC 211 working group was also created in 1994 to address this issue at an international level and has received strong support from many nations. There are many governments, private and academic initiatives that have become the driving force for interoperability development; for example, The National Map[3], Geography Network[4], GeoConnections[5], Geospatial–One-Stop[6], and Digital Library[7], etc. These efforts have resulted in a series of standards, specifications and successful applications.

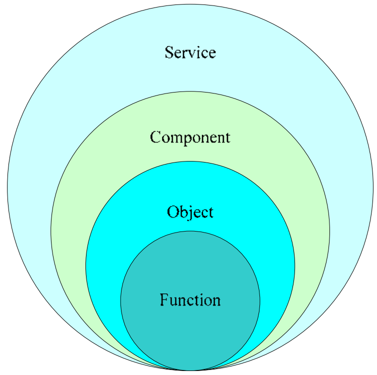


Figure 1. The reusability of developed systems has evovled from function, object, component, to service, with an increasing size of reusable unit.

From a software engineering perspective, the cost and time required for software development largely depends on how much previous development can be leveraged / reused for the subsequent development (e.g., Sommerville 2000). Software reusability is a key to assess system development Reusabiltiy can be classified into four levels, as shown in Figure 1 (Zimmermann, Krogdahl and Gee 2004): (1) function level, (2) object level, (3) component level, and (4) service level. Early software development was mainly based on software module integration at the function level; for example, early GIS software packages such as Intergraph's MGE[9], etc. Software development in the early 90's evolved from the function level to object-level integration; for example, ESRI ArcView [9], etc. The component level of software development gained popularity in the late 90's. Typical examples are ESRI's ArcGIS[10]and Intergraph's Geomedia[11], With the growing use of Web Services-based software development approaches, GIServices is becoming the main stream geospatial application in the internet era. Many GIServices specifications, for example: (1) Web Mapping Service (WMS[12]), (2) Web Feature Service (WFS[13]), (3) Web Coverage Service (WCS[14]), and (4)Web Registry Service (WRS[15]), have been endorsed and deployed.

There were many early research efforts in the field of GIServices. Yang (2000) systematically examined Web-based GIS and GIServices for developing CyberGIS, a WebGIS system. Tao and Yuan (2000) implemented a prototype system, GeoServNet, designed for renting software components and providing a service registry for component providers. Tao (2001) then improved the GeoServNet's capability to offer on-line 2D/3D analysis and visualization services. Luo (2001) and Tsou (2002) have built a series of agent-based GIServicess. Tsou and Buttenfield (1998, 2002) proposed a dynamic metadata GIServices framework to address the problems in dynamic assembly, sorting, and searching of datasets. Yang (2002) proposed a GIServicess environment to organize, manage, share, and utilize distributed geospatial computing resources. Peng and Tsou (2003) systematically examined Internet GIS and GIServices. These and many other relevant research efforts were pivotal investigations for Distributed GIService.

Distributed computing provides an enabling infrastructure for Web Services and Distributed GIService. The combination of GIService architecture with emerging distributed computing technologies, such as Agent[16], Peer-to-Peer (P2P[17]), and Grid[18], has demonstrated advantages by offering more reliable, scalable and efficient capabilities. Currently some research efforts are being devoted to utilization of emerging distributed computing platforms or technologies to build transparent GIServices (Yang et al. 2004a).

In the following sections, we will first introduce the general architecture of Distributed GIService. We then describe emerging distributed computing technologies as well as their use in building Distributed GIService based applications. The fourth section provides an example of Distributed GIService implementation. The last section is devoted to discussion of future research priorities.

2. Distributed GIService Architecture

A service is a software application that can be discovered, described, and accessed based on a specific description language and protocol, and does not depend on the context or state of other services or supporting platforms (Daconta, Obrst and Smith 2003[19]). Service-Oriented Architecture (SOA[20]) is a collection of services with a mechanism to describe, categorize and discover relevant services, provide described services, and integrate an application based on implemented services (Brown, Johnson and Kelly 2002). A typical SOA involves three components: service consumer, service broker and service provider (Brown, Johnson and Kelly 2002). OGC (2003) adopted this architecture to describe the service trading architecture with 'trader', 'importer', and 'exporter' as illustrated in Figure 2. Trader registers offers from the exporter and returns offers upon request to the importer according to the preset criteria. Exporter registers offers with the trader object. Importer obtains offers, if satisfied, from the trader object. In fact, trader plays the role of "matchmaker" in a service-based architecture. This trading interaction process can be modeled in the sequence diagram shown in Figure 3.

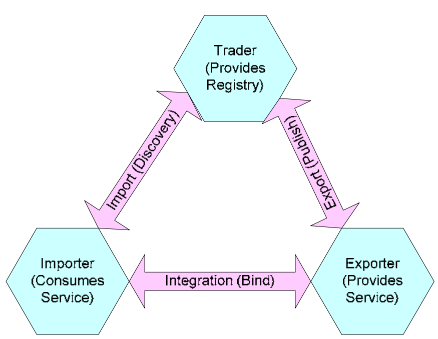


Figure 2 Service Trading (revised from OGC, 2003)

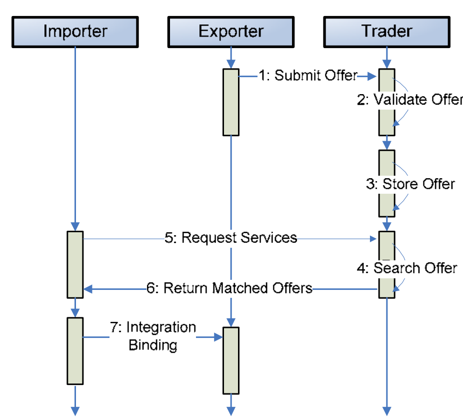


Figure 3 Trading Interactions (revised from OGC 2003)

Based on the SOA architecture, OGC (2003) developed a GIService framework as illustrated in Figure 4. This framework clearly illustrates the common set of interfaces required for enterprise-wide interoperability. In Figure 4, the importer is a multi-source integrated application client. It can be a map viewer, image exploitation client, discovery client, etc; the exporter is a group of service providers. They include Data Services, Portrayal Services and Processing Services providers; the trader is essentially the registry services that provide service type registry, service instance registry, as well as other services. Encoding techniques are unique in the geospatial application. The encoding specifications define the way in which geospatial data is transferred in the GIServices workflow. After the Services providers (Exporter) ***publish*** their services in the Registry services (Trader), the application client (Importer) can ***find*** the requested services from the Registry services. Then the application client will ***bind*** the Service provider and connect to them directly to obtain the services. These OGC services are defined as:

Client Services: the client-side components of client applications that interact with users
Registry Services: provides a common mechanism to classify, register, describe, search, maintain and access information about network resources (data and services)
Processing Services: the foundational application-building-block services that operate on geospatial data and metadata to provide value-added service
Portrayal Services: Portrayal Services provide specialized capabilities supporting visualization of geospatial information. Portrayal Services are components that, given one or more inputs, produce rendered outputs such as cartographically portrayed maps, perspective views of terrain, annotated images, views of dynamically changing features in space and time, etc
Data Services: the foundational service building blocks that serve data, specifically geospatial data. Currently, Data Services include Web Object Service (WOS), WFS, Sensor Collection Service (SCS), Image Archive Service (IAS) and WCS.

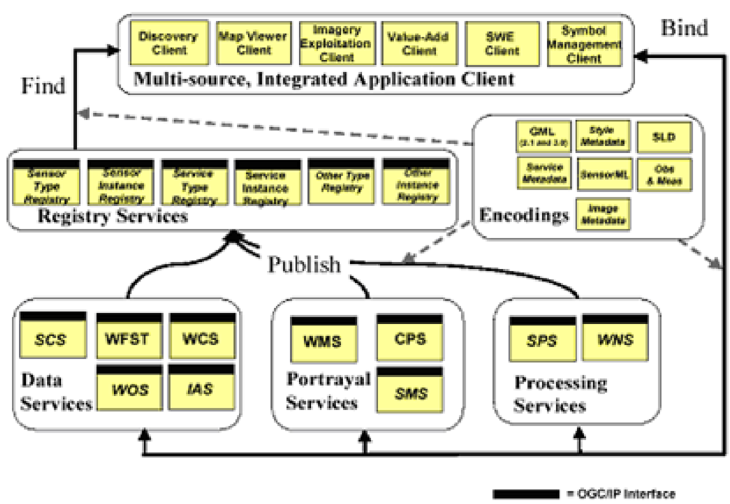


Figure 4. OpenGIS Web Service Framework Components (from OGC 2003)

The above logic is clear. However, in the software world, the realization of the above logic requires careful design. From an operational system point of view, Jeng (2001) suggested a middleware-based approach to achieving interoperability. He defined the following requirements for service architecture:

1) The deployment, discovery, and invocation of service-computing applications can withstand everyday use with minimal maintenance effort

2) A set of guidelines and meta-architecture that can be instantiated into service-computing frameworks or applications
3) Services should be described in a universal manner; well-formedness of services needs to be defined
4) Service-computing roles should be definable; examples of roles are: service requesters, service providers, and service administrators.
5) Service computing applications are executed by dynamic composition of services.
6) Service-computing applications shape their collaborations through capability-based lookup at runtime, not static binding. In other words, collaboration among services needs to be defined as framework instead of hard-wired scenarios.
7) A Service computing platform is required to implement the above concepts, relieving service-computing developers from underlying details such as communication mechanisms, programming languages, operating systems, and middleware.

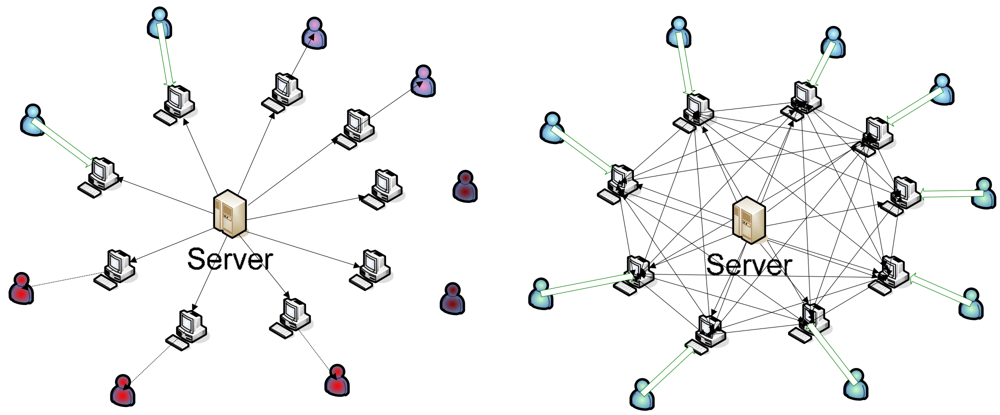
There has been a realization that high performance distributed computing technologies would make Distributed GIService more reliable, scalable, and flexible (Yang et al, 2004b).

3. Distributed Computing Technologies

Distributed GIService relies upon the connection of networked computers to ensure high performance services. Issues, such as parallel execution, reliability, flexibility, and fault tolerance, must be considered in building an effective Distributed GIService. Among the emerging distributed computing platforms, P2P, Agent, and Grid are the most popular ones that have drawn much research. These distributed computing technologies are focused on the utilization and sharing of existing networked computing resources for pervasive access. Implementation generally requires the use of middleware. The following subsections will describe the use of these technologies from a geospatial perspective.

3.1 P2P Computing

According to a story related by Gelernter (2000), “If a million people use a Web site simultaneously, doesn't that mean that we must have a heavy-duty remote server to keep them all happy? No; we could move the site onto a million desktops and use the internet for coordination.” P2P computing takes advantage of all networked computers, especially widely distributed PCs (Barkai 2001). The basic strategy is to use all P2P-networked computers to serve as not only clients but also servers.



a. Traditional Client-Server model: PC is served by Server and some PC users are not satisfied

b. P2P Computing Model: All PCs are connected and provide server functions to satisfy more users

Figure 5. Comparison between Tradition and P2P Computing Models: In a traditional client-server computing model (a) the Server handles all the client requests. Only a few users are satisfied when many users concurrently access the server. Whenever a server failure occurs, requests are blocked. In a P2P computing model (b), clients also respond to requests as servers. The more clients join, the more requests can be accommodated. When the server is inactive, clients can still respond to other clients' requests.

As illustrated in Figure 5, in the traditional client-server architecture, the server provides services to clients, and all services are processed on the server. If the server fails, or the server load is too heavy, response is very slow or requests are blocked. In the P2P computing environment, clients are also configured to provide services instead of only requesting services from the server. When server load is heavy, clients respond to the request, thereby reducing response time. The more clients that are integrated into the P2P computing environment, the better the system performance. This architecture has been used by portals such as Napster for MP3 file sharing and Bit Torrent[21] for file downloading. Although there are some legal issues with the specific applications, the technology itself has demonstrated great potential for sharing large volume information. For example, if NOAA were in distributing a large animation file for a hurricane case, P2P technology would be a good choice.

In order to transfer a client into a server and redirect requests to the transferred client, a middleware layer is used in the P2P computing platform, shown in Figure 6 (Barkai 2001). The P2P platform can be constructed on a number of computers (so called a P2P community) in order to support the same P2P applications. A P2P middleware includes software installed on each PC and some control components installed on a server. The middleware layer schedules the P2P computing, creates processes, manages resources, checks security, and conducts access control. The P2P middleware also provides application interfaces for developing P2P applications.

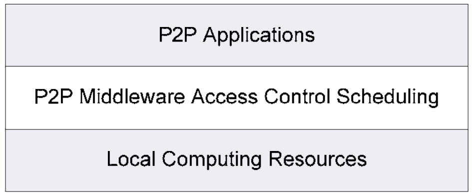


Figure 6. P2P Platform Architecture

Besides file sharing, the middleware could also comprise a series of programs installed on each 'participating' PC to perform processing. When the program operates, it utilizes connected PCs to process client requests when the local user is not using the PC. Once processing is complete, the program returns results to the requesting client and responds to another processing request. When local users begin to use their PCs, the programs will stop and wait until local users' operations are finished. Therefore, P2P computing has the capacity to take advantage of a massive number of networked computers, especially PCs, which are currently only used to 6% capacity.

As an example, Intel has a P2P Program, named Philanthropic, to demonstrate the power of distributed computing by linking millions of PCs around the globe to create one of the most powerful computing resources in the world. Nearly two million PCs joined to provide resources for scientific research. Other companies, such as IBM, are actively involved in software development and commercialization of P2P distributed computing platforms for customers. The information science field can benefit from such a computing platform to process some normally time consuming information analyses and share information (Pitoura et al. 2003).

P2P computing provides an infrastructure for sharing the widely untapped computing power within the Internet. Lowe (2002) and Li (2003) discussed adopting P2P to share maps and conduct GIS operations. The P2P advantage of decentralization, pervasive computing, cost efficiency, scalability and reduced bottlenecks (Flenner et al 2003) provides potential QoS support to DGIService. In the P2P environment, communication and coordination among all peers is a key issue. Agent technology, especially mobile multi-agent system[22, 23] is considered to be a useful alternative to address this issue.

3.2 Agent Computing

The Agent concept resulted from the research of artificial intelligence, and refers to a component of software that is capable of acting in order to accomplish tasks on behalf of its user. The objective of an agent is to accomplish a particular task within a specific context. Once created, an agent is able to find needed information, to interpret the information, and to communicate with other agents to accomplish its task in order to provide a specific service.

The introduction of agent technology to GIS produced some impressive research. For example, Michael Batty and his group have conducted research on the simulation of person behavior patterns in GIS and have tested the possibility and usability of agents[24]. Doran[25] has built an agent-based model of ecosystems for sustainable resource management. Luo (1999) researched the possibility of developing an agent-based distributed GIS and developed a prototype at Peking University. Rahimi et al (2002) designed and developed a complex, administrative infrastructure with a multi-agent system for geospatial information gathering and integration.

Agent technology can be used to build registry search, service discovery and integration, parallel administration, parallel service evaluation, and parallel service register functions. The mobile and multi-agent system approach is also an ideal complement to the P2P platform on the P2P communication and coordination.

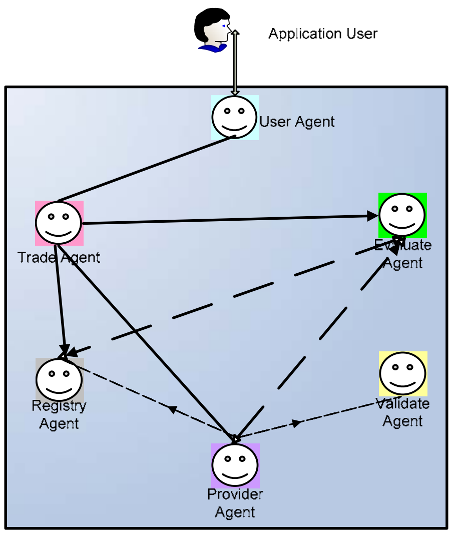


Figure 7 An agent-based service architecture

A potential agent-based system for Distributed GIService is illustrated in Figure 7: 1) An application user sends an application request to the user agent, the user agent interprets the application request and communicates with trade agent to discover proper services with registry agent, selects services with evaluation agent, and comunicates with service provider agent to assemble the application (as illustrated by solid lines); 2) Evaluation agent will communicate with registry agent and provider agent to evaluate services (as illustrated by dashed lines). 3) Provider agent will contact registry agent to add service entry after validation by validate agent (as illustrated by thin lines).

3.3 Grid Computing

Grid defines a blueprint for a new computing infrastructure (Foster and Kesselman 2004). Grid can help us to develop computing systems to provide and/or ingest computing services; just as we are able to plug any electric device into the electric power grid to share electrical power. Grid envisions the unification of all computing resources to provide universal access. Data storage from widely dispersed servers, for instance, can be managed in the grid and can be accessed in the same way in which our local data are accessed. Grid also makes storage more reliable and more efficient, for example, a failure of one server will redirect data access from that failed server to another active server transparently. unified computing context embodies many research topics. The construction of such a data grid or computing grid (Johnson 2000) includes the development and deployment of a series of middleware, advocated by NSF[26], on each of the computers participating in the data or computing Grid. These middleware supply functions such as service controls, uniform data access, security services and co-scheduling.

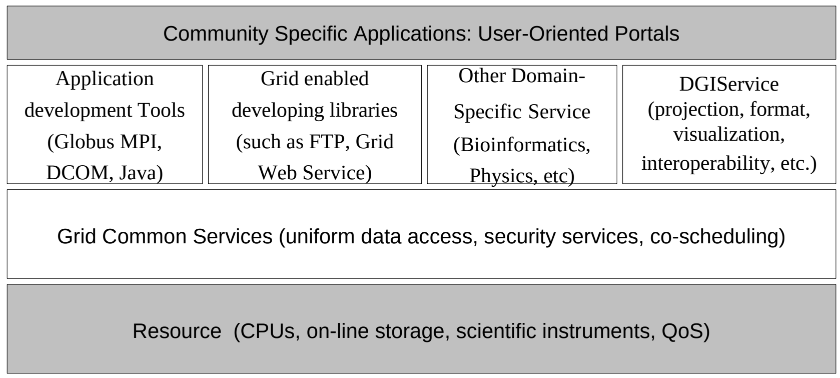


Figure 8 Grid Architecture, the white boxes are implemented as middleware

A typical grid platform includes four layers, as illustrated in Figure 8 (Johnson 2000): 1) the resource layer includes all the computing resources, such as CPU and online storage. 2) The grid common services layer includes the unification services, such as uniform data access, security services, co-scheduling. 3) The domain-specific layer includes specific systems and developing interfaces, such as those for Bioinformatics and Physics. 4) User-oriented portal layer includes grid-supported information service applications. Grid computing users access Grid through Grid Portals. The implementation or deployment of a Grid is based on the middleware components developed. For example, Globus toolkit[27] is a set of middleware for implementing grids. NSF Middleware Initiative[28] lists a set of middleware that can be used for deploying Grid. Armstrong and Wang (2004) are developing GeoMiddleware to support geospatial interoperability for geospatial grid. Yang et al (2004a) are utilizing high performance techniques in GIS (Yang et al 2004b) to develop Geospatial Resource Broker (GRB), a geospatial middleware, for building geospatial grid.

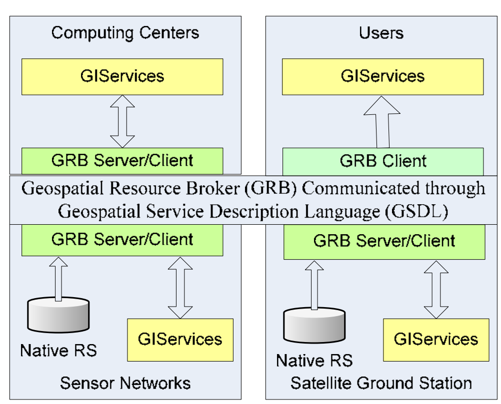


Figure 9 A Distributed GIService can be implemented by deploying GRB on any computing devices, such as sensor networks, computing centers, and satellite ground stations to share GIServices. GRBs will communicate through GSDL. Users can access Distributed GIService through a GRB Client and human-computer interfacing GIServices, such as GIS interfaces. GIServices and local resources can be shared through a GRB Server/Client.

A GRB-based Distributed GIService is illustrated in Figure 9. Native RS refers to local computer resources; Geospatial Service Description Language (GSDL, Yang 2004a) is a geospatially extended Web Service Description Language (WSDL[29]) by including GIService specifications. GSDL is used by GRB to communicate. GRB is installed on a computer to enable that computer as a GRB Node, which can be either a server for providing GIServices or a client for ingesting GIServices. As a client, a GRB node will access geospatial computing resources, including data and computing. As a server, a GRB node will provide its GIServices, such as geospatial data, analytical, modeling, and visualization services, uniformly to the GRB clients.

An end-user who does not want to provide any service to the Grid could install the GRB client only. For computers where both the GRB client and server are installed, users can access the Distributed GIService in the same way as a computer with only the client installed. GIServices that require intensive computing for a single request are installed on high-performance computers and made available to GRB clients. Users will access the grid through human-computer interfacing GIServices. The grid will provide ubiquitous access to geospatial computing resources.

3.4 Supporting Distributed GIService

P2P, Agent, and Grid-based computing provide new ways for utilizing large numbers of networked computers for the purpose of sharing computing resources. They offer powerful mechanisms to support more reliable, scalable, and flexible geospatial data and processing sharing over the network. As discussed, transparency is important in system design so that end users cannot notice any change of their computing environment. This can be implemented by concealing network communication details and by simplifying the middleware deployment process. It is our view that the integration of emerging distributed computing technology will dramatically expand application of GIServices, thereby benefiting the user community. The next section provides an example of a Distributed GIService implementation.

4. Implementation of Distributed GIService

Apart from the Services illustrated in Figures 3 and 4, other computing services that rely upon geospatial characteristics are also required in order to implement an operational Distributed GIService; these must provide: 1) verification to ensure any added services are what they claim to be; 2) evaluation, which tracks service quality, usability, etc,; and 3) balance, to allow for service selection based on performance optimization criteria. Therefore, except for the consumer, provider, and trader (suggested by most SOA applications such as OGC 2003), Verification Service, Evaluation Service, and Load Balancing Service (as part of Trading Service) are also required for enabling Distributed GIService.

Verification Service: Verification service evaluates other services prior to these being added to the service registry. This is necessary to ensure that the submitted service has the claimed functions and that it complies with specific standards.

Evaluation Service: Evaluation service evaluates services based on a service's performance to keep services' historical performance through a credit system.

Balance Service: Balance Service selects services based on specific performance criteria and services' credits in order to ensure that distributed resources are used more efficiently, optimally and uniformly than might otherwise be the case.

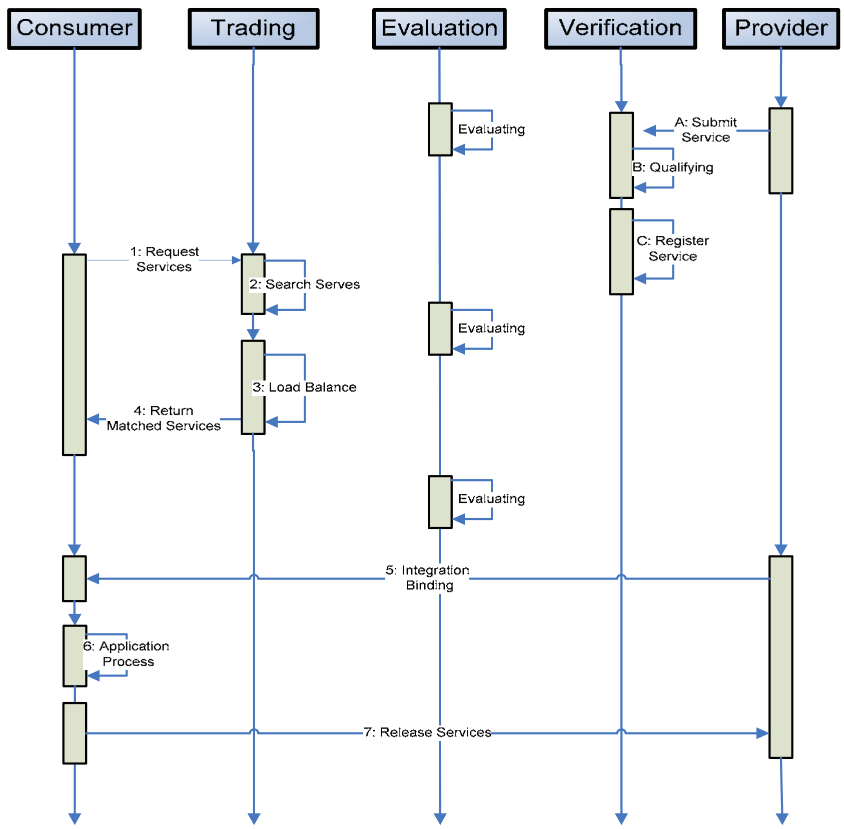


Figure 10 services-evaluating, service-registering, and application-assembling operations within a Distributed GIService

Added these three services, a Distributed GIService has parallel operations as illustrated in Figure 10: 1) Service providers submit services to verification service for verification and register as illustrated through A-B-C; 2) The evaluation service will periodically evaluate registered services based on their operational performance; 3) Application assembling within Distributed GIService includes service request, service search, application binding, application operation, and service release as illustrated in steps 1-2-3-4-5-6-7. Although a Distributed GIService, including the services in Figure 10, can be implemented using different computing platforms, such as DCOM[30], CORBA[31], and DCE[32], the pervasive access of HTTP services[33] across different computing platforms has resulted in the success of Web Services[34] for implementing Distributed GIService.

Web Services are based on the Web environment and are supported by essentially four relevant technologies:
1) HyperText Transport Protocol (HTTP). HTTP provides a de facto pervasive protocol for conducting web request;

2) Simple Object Access Protocol (SOAP[35]). SOAP provides the protocol for Web Services to communicate;
3) Universal Description, Discovery and Integration (UDDI[36]). UDDI facilitates the universal archive and discovery of services; and
4) Extensible Markup Language (XML[37])/ WSDL. XML provides a naturally understandable language for describing information and WSDL provides a way for describing the services provided through Web.

These technologies provide SOA-required transmission, messaging, describing, archiving, and binding support to Web Services. Based on GSDL, GRB, and Web Services supported WMS (a GIService), a Grid-based Distributed GIService can be constructed as illustrated in Figure 11.

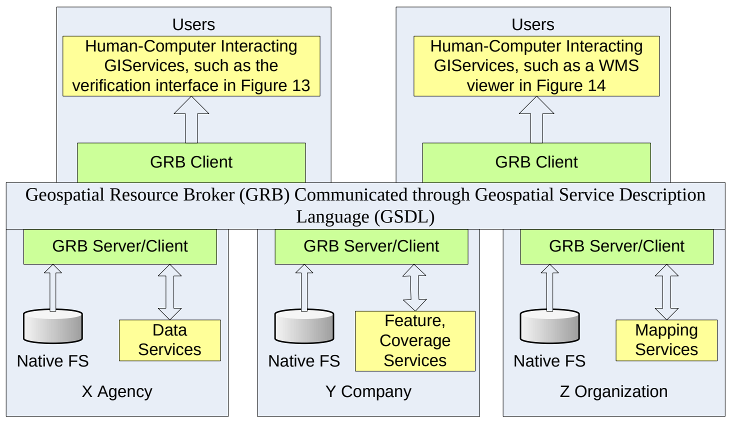


Figure 11. A Distributed GIService: Different GIServices, such as data service, feature service (WFS), and coverage service (WCS), and mapping service (WMS), will be integrated through GRB.

While the coordination of Service selection, integration, validation, and evaluation is taken care of by GRB, Human-Computer Interacting GIService provided to Users will be transparent within the Distributed GIService. During the interaction process (e.g., a Zoom-In including overlay), Distributed GIService may integrate services from different GIServices providers. For example, a Zoom-In will require data services, feature service (WFS) from X, coverage service (WCS) from Y, and require mapping (WMS) from Z. The integration processes are operated by the functions provided by the GRB. The service information exchanged are described in GSDL (Sillitti, Vernazza, and Succi 2002; Yang et al 2004a), as illustrated in Figure 12. Interfaces are illustrated in Figures 13 and 14. Based on legacy systems, only minor changes are needed to WMS/WFS/WCS-enabled legacy GISs in order to develop applications. Therefore, Distributed GIService architecture will allow a fast deployment of applications across a network.

```
...
<service name="Web Map Service X">
      <documentation> map service server 1</documentation>
        <port name="mapPort" binding="tns:mapbinding">
          <soap:address location="http://X.COM/wms/"/>
        </port>
</service>
<service name="Web Map Service Y">
      <documentation> map service server 2</documentation>
        <port name="mapPort" binding="tns:mapbinding">
          <soap:address location="http://y.com/wms/">
        </port>
</service>
...
```

Figure 12 example of GSDL for services binding

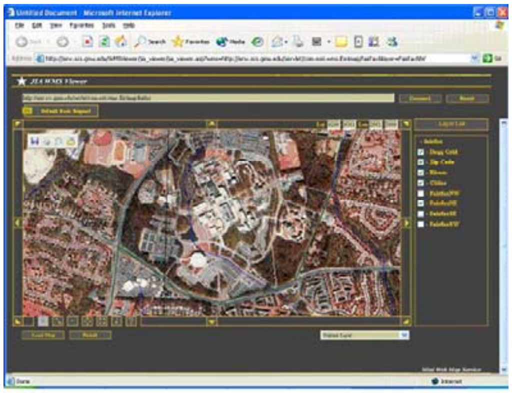


Figure 13 A GIService verification interface for testing, qualifying WMS service

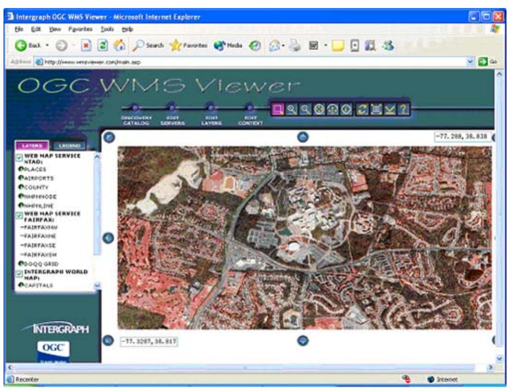


Figure 14 The Intergraph WMS viewer is accessing different WMS services and geospatial data services transparently.

In this example, only data service and WMS/WFS/WCS, several GIServices, are used to illustrate the functionality of a Distributed GIService. A fully functional Distributed GIService can be developed using one of two approaches: 1) add functions to expand GIServices; or 2) expand the implementation area from application, department, community, to the Internet, as a route of infra-Grid, intra-Grid, extra-Grid, and inter-Grid suggested by Joseph, Ernest and Fellenstein (2004) for grid evolution.

5. Conclusion and Discussion

Technically, Distributed GIService is driven by: 1) GIServices supported by Interoperability and Web Services, and 2) distributed computing technologies, such as Grid, P2P, and Agent. P2P is useful for conducting stateless computing or large volume data sharing. Agent computing can be used for performing parallel system administrating, evaluating, and coordinating. Combination of enabling distributed computing technologies with GIServices will provide more reliable, affordable, stable, and scalable services and solutions. Though distributed GIService is being used largely as part of research experiments or government applications, its commercial potential is enormous and transformational. The following are issues that remain to be addressed and / or researched:

1) Many OGC GIService specifications, including WMS, WFS, and GML, etc. are getting popular in the geospatial industry, with more and more software vendors are adopting these specifications. For example, the OCC WMS standard is now endorsed by the ISO. However, the sustainability of OGC interoperability standards depends on the underlying business model and on the clear demonstration of economic and technological benefits to the end user community.
2) New distributed computing technology (Grid or P2P) provides a new platform to implement Distributed GIService. In order to take advantages of the new computing platform, so called geospatial middleware (e.g., GRB) needs to be developed. It is important to conduct integration research using the latest advancements in the distributed computing field; for example, Grid Interoperable standardization framework, such as Open Grid Service Architecture (OGSA, Foster et al 2002[38]), needs to be considered.
3) A distributed GIService cluster would require installation of software on participating user computers. As such, it is critical to have reliable security protocols in place for protecting participating users. Also, copyright information from contributing parties is also important in the vastly connected network.
4) Transparent access and integration of GIServices provided by or served for different user communities, as well as a high level of ontology and knowledge among different communities, is required. Systematic research and development on geospatial syntax, semantics, ontology, and schematics is considered as a cornerstone that will eventually achieve strong interoperability among the communities[39].

Clearly, distributed GIService will introduce many new opportunities for both software vendors and end users. Software vendors can focus on their niche and develop more a competitive edge with less concern for system integration. Users can benefit by getting the best performance/price solutions as well as a wide variety of offerings. On the government side, this framework is also attractive as it lowers the cost of development, shortens the deployment time and is open enough for further upstream enhancement. In fact, for information and resource sharing, the framework is in many ways most valuable to academic institutions and public organizations (Lambert and Widya 2001; Sui and Goodchild 2001). The extremely broad range of applications and strong technical advantages of Distributed GIService suggest it to be a promising emerging paradigm for geospatial computing.

References

Armstrong M and Wang S (2004) GeoMiddleware to Support Interoperability for Grid Computing, position paper for specialist meeting on spatial webs, Santa Barbara, CA: Dec. 2-5, 2004.

Arsanjani A (2004) Service-Oriented modeling and architecture[40].

Barkai D (2001) Peer-to-Peer Computing: Technologies for sharing and collaborating on the Net, Intel Press

Brown A, Johnson S and Kelly K (2002) Using Service-Oriented Architecture and Component-Based Development to Build Web Service Application, A Rational Software White Paper

Buttenfield BP (1998) Looking Forward: Geographic Information Services and Libraries in the Future. Cartography and Geographic Information Systems 25(3): 161-171

Conry-Murray A (2004) Grid Computing's Promises and Perils, Network Magazine, 19 (2):30-40

Daconta M, Obrst L and Smith K (2003) The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management, Wiley: Indianapolis, Indiana, pp. 58

Denise S (1997) Managing the Web-Enhanced Geographic Information Service, The Journal of Academic Librarianship, 6(23) 498-504

Endrei M, Ang J, Arsanjani A, Chua S, Comte P, Krogdahl P, Luo M and Newling T (2004) Patterns: Service-Oriented Architecture and Web Services, IBM Redbook, pp.18-30

Foster I and Kesselman C (2004) The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers, Inc. San Francisco: CA.

Gelernter D (2000) The Second Coming — A Manifesto[41].

Gewin V (2004) Mapping Opportunities, Nature, 427: 376-377.

Goodchild MF, Egenhofer MJ, Fegeas R, Koffman CA (1999) Interoperating Geographic Information Systems. Kluwer Academic Publishers, Boston

Graziano A, Russo S, Vecchio V, and Foster P (2002) Metadata Models for QoS-Aware Information Management Systems, In Proceedings of SEKE'02, July 15-19, 2002, Ischia, Italy: pp.103-106

Jeng J (2001) Towards a Universal Service-Computing Platform Via Virtual Service Machine, In Proceeding of ACM-SAC'2001: 663-667 (Las Vegas, NV)

Johnson WE, Gannon D, Nitzberg B, Tanner LA, Thigpen B and Woo A (2000) Computing and Data Grids for Science and Engineering, IEEE SC'2000, Dallas, Texas[42]

Joseph J, Ernest M and Fellenstein C (2004) Evolution of grid computing architecture and grid adoption models, IBM Systems Journal 43(4):624-645.

Lambert JMN and Widya I (2001) Quality of Service and Service Provisioning on a Competitive Market, Lecture Notes in Computer Science 1890, Springer: 17-25

Li B (2003) Peer-to-Peer Sharing of Cartographic Data and Software, in Maps and the Internet, Michael Peterson (eds), Elsevier Science, pp.157-168.

Lowe J (2002) GIS Meets the Mapster, Geospatial Solution[43].

Luo Y (1999) Study on Agent-based Distributed GIS, Ph.D Dissertation: Peking University, Beijing (in Chinese)

Luo Y, Wang X and Xu Z (2003) Extension of Geospatial Metadata and Agent-based Geospatial Data Navigation Mechanism, In Proceeding of ACM GIS'2003, pp. 102-109

Luo Y, Wang X and Xu Z (2001) Design of a framework for multi-user/application oriented WebGIS Services, In Proceedings of 2001 International Conference on Computer Networks and Mobile Computing, pp.151-156

OGC (2001) The OpenGIS Abstract Specification, Topic 12: OpenGIS Service Architecture, Ver. 4.3[44]

OGC (2003) OpenGIS® Web Services Architecture, OpenGIS© Interoperability Program Report, OGC 03-025

Peng Z and Tsou M (2003) Internet GIS, Wiley: Hoboken, NJ

Pitoura E, Abiteboul S, Pfoser D, Smaras G and Vazirgiannis M (2003) DBGlobe: A Service-Oriented P2P System for Global Computing, SIGMOD 32(3): 77-82

Rahimi S, Cobb M, Ali D, Paprzycki M and Petry F (2002) A knowledge-based multi-agent system for Geospatial Data Conflation, Journal of Geographic Information and Decision Analysis, 6(2):67-81

Rudkin S and Smith A (2001) A scheme for component based service deployment, Lecture Notes in Computer Science 1890, Springer: 68-80

Shi X (2004) Semantic request and response for standardized Web services[45]

Sillitti A, Vernazza T and Succi G (2002) Service Oriented Programming: A New Paradigm of Software Reuse, in ICSR-7 (C. Gacek Ed.) LNCS 2319, pp.269-280

Sommerville I (2000) Software Engineering (6th Edition), Addison Wesley

Stallings W (2002) High-speed Networks and Internets, Upper Saddle River: New Jersey: Prentice Hall

Tao V (2001) Online GIServicess, Journal of Gespatial Engineering, 3(2): 135-143

Tao V and Yuan S (2000) GeoServNet: Renting Geotools over Internet, Geo-Informatics, 14(12):12-15

Tomlinson R (1998) The Canada Geographic Information System. In Foresman T W (ed.), The History of Geographic Information Systems, Upper Saddle River, Prentice Hall: 21-32.

Tsou M (2002) An Operational Metadata Framework for Searching, Indexing, and Retrieving Distributed Geographic Information Services on the Internet. In Geographic Information Science (GIScience 2002), M. Egenhofer and D. Mark (eds.), Lecture Notes in Computer Science Vol.2478, Springer-Verlag, Berlin, pp.313-332.

Tsou M and Buttenfield B (1998) Client/Server Components and Metadata Objects for Distributed Geographic Information Services, In Proceedings of the GIS/LIS'98, Fort Worth, Texas, pp.590-599

Tsou M and Buttenfield B (2002) A Dynamic Architecture for Distributing Geographic Information Services, Transactions in GIS 6(4):355-381

Vckovski A (1998) Interoperable and Distributed Processing in GIS, Taylor & Francis, London

Yang C (2000) Theory, Techniques and Implementation Methods of WebGIS, Unpublished Ph.D dissertation (in Chinese) (Beijing: Peking University)

Yang C, Kafatos M, Yang R, Wong D and Qu J (2002) Geographic Information Service Environment, In CD Proceeding of GeoInformatics'2002, Jun.1-3, Nanjing, China.

Yang C, Kafatos M, Wong D, Yang R and Cao Y (2004a) GridGIS: A next generation GIS, CITSA 2004, Jul. 21-25, Orlando, FL, pp.22-27.

Yang C, Wong D, Yang R X, Kafatos M and Li Q (2004b) Performance improving techniques in WebGIS, IJGIS, in press.

Yang R, Yang C and Kafatos M (2002) A distributed framework and its application in supporting automatic cooperative work in NASA environmental research, Nov.10-14, Denver, CO, USA, Proceeding of Integrated Remote Sensing at the Global, Regional and Local Scale: 293-297.

Zimmermann O, Krogdahl P and Gee C (2004) Elements of Service-Oriented Analysis and Design: An interdisciplinary modeling approach for SOA projects[46].

Website References

1. http://www.opengis.org/, accessed December 25, 2004.
2. http://www.fgdc.gov/, accessed December 25, 2004.
3. http://nationalmap.usgs.gov/, accessed December 25, 2004.
4. http://www.geographynetwork.com/, accessed December 25, 2004.
5. www.GeoConnections.org, accessed December 25, 2004.
6. http://www.geo-one-stop.gov/, accessed December 25, 2004.

7. http://www.alexandria.ucsb.edu/, accessed December 25, 2004.

8. http://imgs.intergraph.com/mge/, accessed December 25, 2004.

9. http://www.esri.com/software/arcview/index.html, accessed December 25, 2004.

10. http://www.esri.com/software/arcgis/, accessed December 25, 2004.

11. http://imgs.intergraph.com/geomedia/, accessed December 25, 2004.

12. http://portal.opengis.org/files/?artifact_id=5316, accessed December 25, 2004.

13. https://portal.opengeospatial.org/files/?artifact_id=7176, accessed November 21, 2004.

14. https://portal.opengeospatial.org/files/?artifact_id=3837, accessed November 21, 2004.

15. http://www.opengis.org/docs/01-024r1.pdf, accessed December 25, 2004.

16. http://agents.umbc.edu/, accessed December 25, 2004.

17. http://www.openp2p.com/, accessed December 25, 2004.

18. http://www.ggf.org/, accessed December 25, 2004.

19. http://www.service-architecture.com/web-services/articles/service-oriented_architecture_soa_definition.html, accessed December 25, 2004.

20. http://www.service-architecture.com/, accessed December 25, 2004.

21. http://bittorrent.com/, accessed December 25, 2004.

22. http://dis.cs.umass.edu/, accessed December 25, 2004.

23. http://citeseer.ist.psu.edu/533959.html, accessed December 25, 2004.

24. http://www.casa.ucl.ac.uk/research/agent.htm, accessed December 25, 2004.

25. http://cswww.essex.ac.uk/staff/doran.html, accessed November 20, 2004.

26. http://www.nsf-middleware.org/, accessed December 25, 2004.

27. http://www.globus.org/, accessed December 25, 2004.

28. http://collab.nsf-middleware.org/Lists/NMIR6/AllItems.aspx , accessed December 25, 2004.

29. http://www.w3.org/TR/wsdl, accessed December 25, 2004.

30. http://www.microsoft.com/com/default.mspx, accessed December 25, 2004.

31. http://www.corba.org/, accessed December 25, 2004.

32. http://www.opengroup.org/dce/, accessed December 25, 2004.

33. http://www.w3.org/Protocols/, accessed December 25, 2004.

34. http://www.w3.org/2002/ws/, accessed December 25, 2004.

35. http://www.w3.org/TR/soap/, accessed December 25, 2004.

36. http://www.uddi.org/specification.html, accessed December 25, 2004.

37. http://www.w3.org/XML/, accessed December 25, 2004.

38. http://www.globus.org/ogsa/, accessed December 25, 2004.

39. http://www.ncgia.ucsb.edu/projects/nga/, accessed on 12/20/2004

40. http://www-106.ibm.com/developerworks/webservices/library/ws-soa-design1/, accessed 11-28-04

41. http://www.edge.org/3rd_culture/gelernter/gelernter_p1.html, accessed 11/20/04.

42. http://www.sc2000.org/techpapr/papers/pap.pap253.pdf, accessed December 25, 2004.

43. http://www.geospatial-online.com/geospatialsolutions/article/articleDetail.jsp?id=8141, accessed December 25, 2004.

44. http://www.opengeospatial.org/docs/02-112.pdf, accessed 12/9/04

45. http://www-106.ibm.com/developerworks/webservices/library/ws-semantic/, accessed 2004-11-28.

46. http://www-106.ibm.com/developerworks/webservices/library/ws-soad1/, accessed Nov.29, 2004.