PEER-TO-PEER BASED GIS WEB SERVICES

J.H.Guan [a], S.G. Zhou [b], L.C.Wang [a], F.L.Bian [a]

[a] School of Computer, Spatial Information & Digital Engineering Center, Wuhan University, Wuhan, 430079, China-jhguan@wtusm.edu.cn
[b] Department of Computer Science & Engineering, Fudan University, Shanghai, 200433, China-sgzhou@fudan.edu.cn

TS, WG II/3

KEY WORDS: GIS, Research, Interoperability, Infrastructure, Internet, Web based, Distributed

ABSTRACT:

Geographic Information System (GIS) based on Internet is a promising research field, which brings new approaches to access, share and disseminate geographic information. However, current Internet-based GIS relies heavily on centralized server, which inevitably has drawbacks such as single points of failure and network congestion etc. In recent years, Peer-to-Peer (simply P2P) and Web services are two of hot research topics in network computing and appear as two extremes of distributed computing paradigm. By adopting a decentralized network-based style, P2P technology can improve scalability/reliability, enhance the overall reliability and fault-tolerance, increase autonomy and enable ad-hoc communication and collaboration. On the contrary, current proposals for Web services infrastructures are mainly based on centralized approaches, which are prone to introducing single points of failure, hotspots in the network, and exposing vulnerability to malicious attacks. In this paper, we explore the techniques of building GIS Web services systems in P2P environment. By combining Web services and P2P technologies into GIS, we aim to add more flexibility and autonomy to GIS Web services systems, and alleviate to some degree the inherent limitations of centralized systems. We propose a P2P based GIS Web Service architecture, and as a case study, we present our ongoing project BP-GServices, *i.e.*, BestPeer based GIS Web Services, and the major techniques of BP-GServices implementation.

1. INTRODUCTION

In conventional GISs, geographical data is typically generated and stored locally and then utilized by a limited number of specialized computations or services on that site. For decades, GIS has relied heavily on centralized paradigm with the complete set of data stored on one single server. Remote computers can access the data via the interconnection networking with quite high networking access costs. To get the required geographic information or services, we have to gather information from various locations and process them by using centralized server paradigm, which has inherent drawbacks such as single points of failure, network congestion, and data inconsistency etc. With the popularity of Internet and the diversity of GISs, more and more geo-referenced information sources spread over the network. The inherent disadvantages of traditional GISs need to be solved for new applications on Internet or Web.

Nowadays Peer-to-Peer (P2P) and Web services are two of the hot research topics in network computing. Roughly, they appear as two extremes of distributed computing paradigm. Conceptually, P2P refers to a class of systems and applications that employ distributed resources to perform a critical function in a decentralized way. A P2P distributed system typically consists of a large number of nodes that can potentially be pooled together to share their resources, information and services. These nodes, taking the roles of both consumer and provider of data and/or services, may join and depart the P2P network at any time, resulting in a truly dynamic and ad-hoc environment. In addition, the distributed nature of such a design can eliminate the need for costly infrastructure by enabling direct communication among clients, and enable resource aggregation, so thus provide promising opportunities for novel applications to be developed (Ooi, 2002).

On the other hand, Web services technologies provide a language-neutral and platform-independent programming model that can accelerate application integration inside and outside the enterprise (Gottschalk, 2002). It is convenient to construct flexible and loosely coupled business systems by application integration under Web services framework. However, current proposals for Web services infrastructures are mainly based on centralized approaches such as UDDI (UDDI, 2004): a central repository is used to store services descriptions, which will be queried to discover or, in a later stage, compose services. Such centralized architecture is prone to introducing single points of failure, hotspots in the network and exposing vulnerability to malicious attacks. Furthermore, making full use of Web services capabilities using a centralized system does not scale gracefully to large number services and users. This difficulty is severed by the evolving trend to ubiquitous computing in which more and more devices and entities become services and service networks become extremely dynamic due to constantly arriving and leaving service providers.

To overcome the limitations of Web services systems causing by their centralized architecture, we explore the techniques of building GIS Web services applications under P2P environment. By fitting GIS Web services into P2P environment, we aim to add more flexibility and autonomy to GIS Web services systems, and alleviate to some degree the inherent limitations of the centralized systems. In this paper, we explore the techniques of building GIS Web services systems in P2P environment. We propose a P2P based GIS Web Service architecture, and as a case study, we present our ongoing project BP-GServices, i.e., BestPeer based GIS Web Services, and the major techniques of BP-GServices implementation.

The reminder of this paper is organized as follows. An overview of BestPeer is given in Section 2. Then, the BP-GServices project details of architecture and major implementing techniques are exposed in Section 3. Following that, a simplified prototype is introduced in Section 4. Finally, the conclusions are given in Section 5.

2. ABOUT BESTPEER

BestPeer is a generic P2P system, which is designed and implemented by National University of Singapore and Fudan University of China, to serve as a platform on which P2P applications can be developed easily and efficiently. Figure 1 illustrates a BestPeer network.

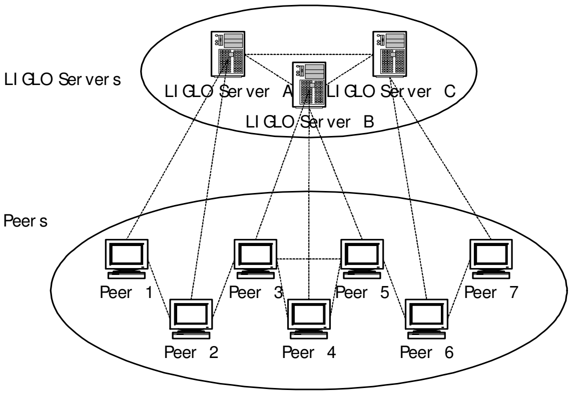


Figure 1. BestPeer network

BestPeer system consists of two types of nodes: a large number of normal computers (i.e. peers), and a relatively fewer number of Location-Independent Global names Lookup (LIGLO) servers. Every peer in the system runs the BestPeer software, and will be able to communicate and share resources with any other peers. There are two types of data in each peer: private data and public (or sharable) data. For a certain peer, only its public data can be accessed by and shared with other peers. A LIGLO server is a node that has a fixed IP and running Location-Independent Global Names Lookup Server software. It provides two main functions: generate a BestPeer Global Identity (BPID) for a peer and maintain peer's current status, such as the current IP address and whether the peer is currently online or offline (if this information is available). Through LIGLO servers, a node can exactly who its peer is.

BestPeer overcomes the limitations of existing P2Psystems and has following four good features (Ling, 2002; Zhou, 2002):

1) BestPeer combines the power of mobile agent and P2P technologies into a single system. On the one hand, P2P technology provides resource-sharing capabilities amongst peers and enables ad-hoc communication and collaboration of the system. On the other one, since agents can carry both data and code, they can effectively perform any kind of functions. With mobile agents, Bestpeer not only provides files and raw data, it also provides processed and meaningful information. More over, the use of agents allows Bestpeer nodes to collect information on the entire Bestpeer network, and this can be done offline. This allows a node to be better equipped to determine who should be directly connected peers or who can provide it better services.

2) BestPeer not only facilitates a finer granularity of data, files and services sharing, it also shares computational power of peer nodes in the system. Since mobile agents can carry data and code, the requester performs the filtering task at the provider's end and gets processed information. This feature has several advantages: (a) it allows filtering to be performed where the provider's end does not provide the capability; (b) it allows individual requester to filter the content according to what (s)he desires; (c) it facilitates extensibility – new algorithm or program can be used without affecting other parts of the system; (d) existing non-distributed objects can be easily extended for use by a P2P application by leveraging on the support provided by BestPeer; (e) it optimizes network bandwidth utilization as only the necessary data is transmitted to the requester.

3) BestPeer supports mechanisms to dynamically keep promising(best) peers in some proximity based on some criterion. Thus, BestPeer will always try to make a direct connection to these promising(best) nodes that have bigger possibility to provide information and data. In this way, promising peers are first traverse before the less promising ones. BestPeer currently supports two default reconfiguration strategies: MaxCount and MinHops. The former maximizes the number of objects a node can obtain from its directly connected peers; while the latter implicitly exploits collaboration with peers by minimizing the number of hops.

4) BestPeer uses location independent global names lookup (LIGLO) servers to provide each peer node of the system with a unique global identity (BPID). By the way, each peer node has a unique global identity that is different from any other peer in the system even if its IP address has been changed. A peer node with unique global identity can communicate with any other peer node and exchange sharable information. At the same time, LIGLO servers can maintains peer's current status in the system, such as the current IP address and whether the peer is currently online or offline (if this information is available).

3. BP-GSERVICES□BESTPEER BASED GIS WEB SERVICES FRAMEWORK

3.1 The Architecture of BP-GServices

As aforementioned, BP-GServices is an application system that is designed and implemented on BestPeer. Similarly, the BP-GServices also comprises two kinds of entities, i.e. several LIGLO servers and a large number of normal peer nodes. The former generates a BestPeer Global Identity (BPID) for a peer and maintain peer's current status, and the latter takes the roles of a services provider and a services consumer as well as a services registrar. Thus, there is no central UDDI registry in BP-GServices, all services and their services descriptions are distributed over the peer nodes. Figure 2 illustrates the internals of a BP-GServices peer node. The system is essentially composed of seven components that are loosely integrated.

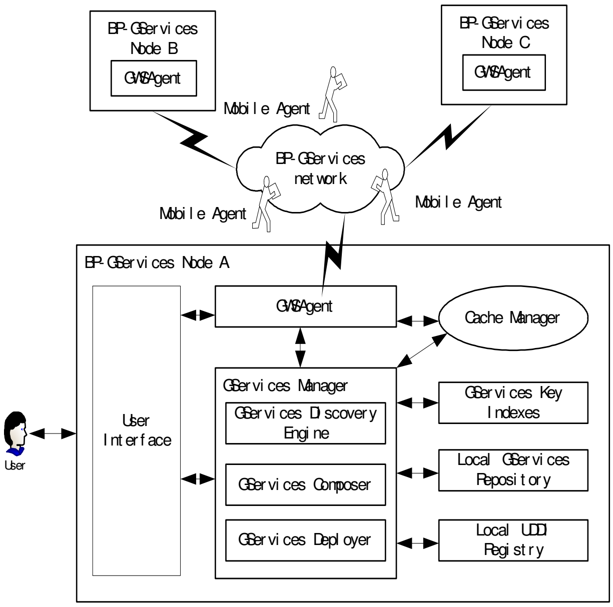


Figure 2: The architecture of BP-GServices Peer

The first component is a GServices Manager and also the most important component of BP-GServices. It can facilitate GIS services discovery, GIS services composition and GIS services Deploying. Corresponding to its functionalities, GServices Manager consists of three sub-components, namely the GServices Discovery Engine, the GServices Composer and the GServices Deployer. The GServices Discovery Engine is responsible for the publication and location of GIS services. The GServices Composer provides facilities for defining new composite GIS services from existing GIS services, and editing existing GIS services (local), which is finished in a visual interface (as a part of the user interface). The GServices Deployer facilitates the binding and invocation of requested GIS services as well as coordination of composite GIS services.

The second component is the GIS Web Services Agent System, or simply GWSAgent. GWSAgent mainly provides the environment on which mobile agents operate. Each BP-GServices node in the system has a master agent that manages the GIS services discovery and GIS services description retrieval. In particular, it will clone and dispatch worker agents to neighbouring nodes, receive GIS services or processed results and present them to the user. It also monitors the statistics and manages the network reconfiguration policies.

The third component is a Cache Manager, which is used for caching the results of GIS services discovery and retrieval in order to reduce the response time of subsequent answers. A Cache Manager has the following functions: 1) cache remote GIS services in secondary storage; 2) determine the caching/replacement policy; 3) by collaboration among the cache managers, form a P2P cache subsystem under the BP-services framework so that all peers can share the caching results among themselves as in BuddyWeb (Wang, 2002).

The fourth component is the User Interface. Here the user interface consists of several interface modules, corresponding to GIS services discovery and retrieval, GIS services composition and deploying. This provides a user-friendly environment for a user to submit their GIS services query, to maintain their sharable GIS services, and insert/delete their GIS services.

The system also includes the other three components: GServices Key Indexes, Local UDDI Registry and Local GServices Repository. Here, GServices repository can provide GIS services provided locally. The description (or publication) information of local GIS services is kept in the Local UDDI registry. GServices Key indexes holds lists of services keywords extracted from the description information of local services, mainly business names and service types, etc, in order to speed up GIS services discovery.

3.2 Nodes Initiating in BP-GServices

Each peer node in the system installs and runs the BP-GServices software, which is firstly used by the new peer to process the files it maintains. By now, the peer is only an autonomous information system and is not a participant of BP-GServices system. If the peer node wants to become a participant of BP-GServices system, the following process is taken.

1) The user uses an application table reflecting his/her favourites to register with a LIGLO server, which is similar to registering a mail server in Internet environment.

2) Then, the LIGLO server will provide the node with a global and unique identifier, i.e., BPID (BestPeerID), which includes two parts: LOGLOID and NodeID. The former is the IP address of LIGLO server, and the latter is the unique ID for the peer assigned by the LIGLO server. By the way, a user with the same BPID will be recognized as the same user even if its IP address has been changed.

3) At the same time, the LIGLO server will also send the node a list of peer nodes that have already registered in the network, i.e., the initial direct peers of the node.

4) After the above steps were over, the node's initiating process has been finished.

If a Peer P_i who has been participant wants to rejoin the BP-GServices system after disconnection of failure, taking the following process (Ling, 2002):

1) Firstly, Peer P_i sends its current IP address to its LIGLO server to allow its LIGLO server to update its IP address if it has changed.

2) Secondly, it sends an active message to each of peers in its ConfidantCircle (say P_j) to restore its connection with them. If P_i receives an Active_OK message from P_j, then it restores their connection successfully. Otherwise, with the help of LIGLO server, P_i should first determine the status of the confidant, i.e., InActive, Active but IP changed, and then it takes corresponding actions.

3) By now, it has finished its rejoining process.

3.3 Neighbor Nodes finding in BP-GServices

In BP-GServices, neighbor nodes of a peer node are these nodes that can provide more services similar to that in the given node than other nodes. Here we use information retrieval method to find neighbor nodes.

We can use documents clustering methods as (Baesa-Yates,

1999) to cluster the nodes in BP-GServices system. We treat each node as a document, whose content is the services description information (UDDI registry) contained in that node. Thus we can cluster the nodes in BP-GServices according to the services that the nodes can provide. Simply speaking, nodes in the same cluster may provide more similar services than these from different clusters. However, different from traditional documents clustering methods that are based on a global data view, BP-GServices is based on decentralized P2P network and the nodes in the system are dynamic. Thus, we adopt a simple local clustering strategy in BP-Services.

Here we use Boolean model to represent a peer services node. There are two reasons for using Boolean model: 1) Boolean model is easier to evaluate than vector space model (VSM); 2) It is difficult to set the document vector space because there is no deterministic global data view in P2P environment.

Given a peer services node *p*, there exists a set of keywords extracted from the services description document of *p*. We denote the set of keywords K_p, and treat it equal to the node *p* itself. For two services nodes *p* and *q*, suppose their keywords sets are K_p and K_q, the similarity of the two nodes are defined as follows:

$$sim(K_p, K_q) = \frac{|K_p \cap K_q|}{|K_p \cup K_q|}.$$

Above, |•| indicates the cardinality of a set.

After joining the network, the node (say *p*) can begin to find its neighbor by the following steps:

1) Through the ping-pong messages (Gnutella, 2004), it contacts the set of peers within a certain number (say *k*) of hops away from it. Let denote the set of peers as Peer(*p*, *k*)=$\{q_1,q_2,\ldots q_n\}$, and get these peers' keywords sets $\{K_i|i=1\sim n\}$.

2) Calculate the similarity of *p* and each peer in *Peer*(*p*, *k*), *i.e.*, $\{sim(p, q_i)|\ i=1\sim n\}$.

3) Suppose *q* is the peer in *Peer*(*p*, *k*) that has largest similarity with *p*, then take *q* as *p*'s neighbor node, and connect *p* and *q* by a direct link, which is termed *neighbor link* of *p* and *q*.

Through the process of neighbor finding, the peers that share services tend to be connected together by neighbor links, and consequently form clusters of services peers. Considering the dynamism of P2P system, the peers should update their neighbors regularly.

3.4 GIS Services Generating in BP-GServices

In generic GIS, geographic information and data may be stored in all kinds of databases or files. However, different from a generic GIS, geographic information and data in BP-GServices are provided in the form of Web services. Before these geographic information and data are provided to users, they must be wrapped into GIS Web services to be accessed by users more conveniently. GIS Web services in the GServices repository can be divided into two kinds: static GIS services and dynamic GIS Web services. Static GIS services have been generated before a user's GIS service request and have been put in the GServices repository to improve the GIS services accessing speed, and Dynamic GIS Web services are not formed until the GIS service request comes. There are two reasons to use dynamic services: 1) the storage space of each peer node is limited. 2) GIS services request of different users are different and it isn't realistic to produce all possible GIS services in the GServices repository. Figure 3 illustrates the process of GIS services generating.

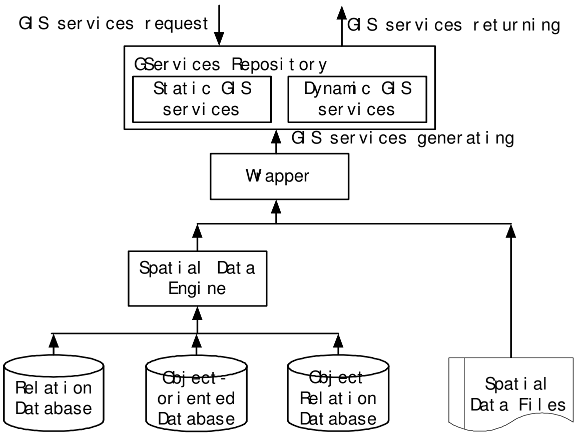


Figure 3. GIS services generating in BP-GServices

When a GIS service is required to be generating, the following steps are taken.

1) Firstly, read out the geographic information and data from spatial database or files according to the GIS service demand. When these geographic information and data are from spatial database, they need to be transformed into a common format, such as GML (GML, 2004) format.

2) Secondly, these geographic information and data are wrapped into a static GIS Web service or a dynamic GIS Web service according to different situation.

3) Finally, the generated GIS Web services are kept in GServices repository.

3.5 GIS Services Discovery in BP-GServices

UDDI registries of a generic Web Services system are mainly based on centralized approaches. When a user needs a GIS service, he only need search the centralized servers that have UDDI registries. Different from that of a generic Web Services system, these of GIS Web Services system are distributed on the peer nodes. The request of GIS services submitted by the user can be in the local GServices Repository or in the remote GServices Repository. Since different peer nodes can have same GIS Services in the GServices Repository, it is not necessary to search the targeted services by traversing all peers one by one and when a GIS Services that satisfies the requester's requirements is found, the discovery process should be stopped.

In BP-GServices, once a requester submits his/her GIS service requirement, say a service query Q, the following process will be launched:

1) Extract keywords from Q. By keywords matching in information retrieval, we can carry out the GIS service search process.

2) First, search the local services indexes at the local peer. If there are services matching the query, then go to 3); otherwise, go to 4).

3) The user see whether there are services (s)he wants by checking services' descriptions that is returned. If there is at least one service (s)he wants, then the process of GIS service discovery is over; otherwise, go to 5)

4) Select randomly an initial link of the local peer. Then clone and dispatch a worker agent with the GIS service query to the peer at the other end of the selected initial link. At that remote peer, doing the searching as at the local peer.

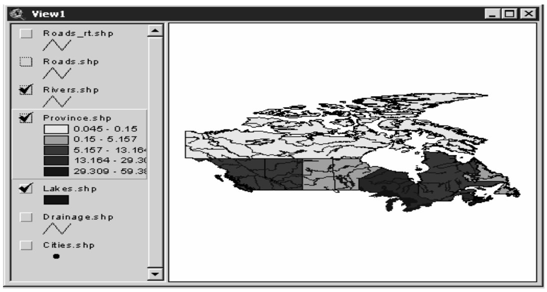
5) Cloning a working agent and dispatching it with the service query to the local peer's neighbour. At the neighbour peer, doing the searching as at the local peer.

6) At the remote peer, once there are services matching the query, then returning the matching services' descriptions to the user who decides whether the returned results contain the target service. If the target service is found, then the search task is over, and the working agent would return the source peer or be destroyed at the remote peer. If no target service is found, the working agent has to continue the search target till the target service is found or the working agent's TTL is 0.

Note in the service discovery process above, when the working agent gets to a peer along a neighbor link, its TTL will not decrease; Only walking along initial link, its TTL will decrease.

4. A PROTOTYPE

In order to assess the feasibility of the architecture, a simplified prototype is developed and some GIS Web services have been implemented. We use the Aglet Software Development Kit 2.0, J2SDK 1.4.2, BestPeer and Geotools 0.8.0 for implementing the prototype. Hardware includes five PCs, in which two PCs are used for LIGLO servers and the other three PCs are used as BP-GServices node. Geographic information of the states, cities, rivers, roads, and lakes of the Canada from ARCView GIS 3.2 are split into three parts to be stored in the three the BP-GServices nodes respectively and some basic GIS services are provided. Figure 4 shows the GIS Web services of the prototype and the query result for rivers, provinces and lakes.



☐ Roads_rt.shp
☐ Roads.shp
☑ Rivers.shp
☑ Province.shp
0.045 - 0.15
0.15 - 5.157
5.157 - 13.16
13.164 - 29.3
29.309 - 59.3
☑ Lakes.shp
☐ Drainage.shp
☐ Cities.shp

Figure 4. A query result of the prototype

5. CONCLUSIONS

In this paper, we explore the techniques of establishing GIS Web services systems in P2P environment. And a P2P based GIS Web services framework is proposed. By combining Web Services technology and P2P technology into GIS, we add more flexibility and autonomy to GIS Web services systems, and alleviate to some degree the inherent limitations of the centralized systems. As an ongoing project, implementation of BP-GServices is still underway. After the BP-GServices prototype is finished, we're ready to integrate existing geographic data and information into the system. We also plan to extend BP-GServices to embrace semantic GIS Web services in the near future.

REFERENCES

Ooi, B.C., Tan, K-L., Lu, H., et al,. 2002. P2P: harnessing and riding on peers. In: *Proceedings of NDBC 2002.*China.

Gottschalk, K,. Graham, S., Kreger, H., et al., 2002. Introduction to Web services architecture. *IBM Systems Journal,* 41(2), pp.170-177.

UDDI: http://www.uddi.org/ (accessed 15 Oct.2003)

Ng, W.S., Ooi, B.C., Tan, K-L., 2002. Bestpeer: A self-configurable peer-to-peer system. In: *Proceedings of ICDE2002, San Jose, CA*. pp.272.

BestPeer: http://xena1.ddns.comp.nus.edu.sg/p2p/. (accessed 10 Dec.2003)

Ng, W., Ooi, B.C. Tan, K-L. et al., 2003. PeerDB: a P2P-based system for distributed data sharing. In: *Proceedings of ICDE2003*. pp.633-644.

Ling, B., Lu, Z., Ng, W.S., et al., 2002. A content-based resource location mechanism in PeerIS. In: *Proceedings of WISE2002*. pp.279-290.

Wang, X., Ng, W.S., Ooi, B.C., et al., 2002. Buddyweb: a p2p-based collaborative web caching system. In: *Proceedings of the 2002 International Workshop on peer-to-peer computing*. pp.247-251

R. Baesa-Yates and B. Ribeiro-Neto.,1999. Modern Information Retrieval. *Addison Wesley, Reading, Mass.*

C. Ng and K. Sia.,2002. Peer clustering and firework query model. In: *Poster Proceedings of 11th World Wide Web Conference (WWW2002)*, Honolulu, Hawaii.

Cooper, B., Garcia-Molina, H., 2003. SIL: Modeling and Measuring Scalable Peer-to-Peer Search Networks. In: *Proceedings of DBISP2P.* pp.2-16.

Ganesan, P., Sun, Q., Garcia-Molina, H., 2003. YAPPERS: A Peer-to-Peer Lookup Service Over Arbitrary Topology. In *Proceedings of INFOCOM 2003,* San Franciso, CA, USA.

Graupmann, J., Biwer, M., Zimmer, P., 2003. Towards Federated Search Based on Web Services. In: *Proceedings of 10th Conference on Database Systems for Business, Technology and Web (BTW),* Leipzig, Germany. pp.384-393.

Zhou, S.G., Zhou, A.Y., Qian, W.N., et al., 2002. Building Web Services in P2P Environment. In: *Proceedings of GCC 2002.* Sanya, China.

Gnutella: http://gnutella.wego.com/ (accessed 8 Oct.2003))

GML.http://www.w3.org/Mobile/posdep/Presentations/GMLIntro (accessed 5 Mar.2003)

Aglets: http://www.trl.ibm.co.jp/aglets. (accessed 20 Sep.2001)

J2sdk1.4.2: http://wwws.sun.com/ (accessed 10 Sep.2003)

Geotools: http://sourcefroge.net/project/ (accessed 10 Apr.2002)

ACKNOWLEDGEMENTS

The work is supported by Nature Science Foundation under grant No. 60373019, and Hi-Tech Research and Development Program of China under grant No. 2002AA135340, and IBM Research Award, and Open Researches Fund Program of LIESMARS under grant No. SKL(01)0303.