

## A MOBILE-AGENT AND GML BASED FRAMEWORK FOR INTEGRATING DISTRIBUTED GIS\*

Jihong GUAN<sup>a</sup>, Shuigeng ZHOU<sup>b</sup>, Fuling BIAN<sup>c</sup>

<sup>a</sup>School of Computer Science, Wuhan University, Wuhan, 430072, China, [jhguan@wtusm.edu.cn](mailto:jhguan@wtusm.edu.cn)

<sup>b</sup>State Key Lab of Software Engineering, Wuhan University, Wuhan, 430072, China, [zhousg@whu.edu.cn](mailto:zhousg@whu.edu.cn)

<sup>c</sup>Spatial Information and Digital Engineering Center, Wuhan University, [flbian@hotmail.com](mailto:flbian@hotmail.com)

Commission II, WG II/3

**KEY WORDS:** Distributed GIS, Mobile-agent, GML, Framework, Integration, Heterogeneity Handling

### ABSTRACT:

Geographic Information System (GIS) based on Internet as a promising research field brings new opportunities and challenges to geographic information accessing, sharing and disseminating. However, the inherent limitation of current Internet GIS architecture based on connection makes it incompetent to fulfil various requirements of GIS applications under dynamic, complicated and distributed computing environment such as the Mobile Internet. As a recently developed paradigm in distributed computing, mobile agent can especially benefit (Mobile) Internet applications on providing asynchronous task execution and more dynamics, supporting flexible and extensible cooperation, reducing communication bandwidth, enhancing real time abilities and higher degree of robustness, enabling off-line processing and disconnected operation. On the other hand, Geography Markup Language (GML) as an emerging standard of geographic information exchanging offers a new way to integrate geographic information on network. In this paper, we propose a new paradigm in constructing distributed GIS by combining mobile agent and GML technologies, in which mobile agent technology is used to overcome the inadequacy of other traditional distributed computing paradigms and GML to solve the heterogeneity problem of various GISs.

### 1. INTRODUCTION

Nowadays, the diversity of Geographic Information Systems (GISs) and the widespread availability of World Wide Web have led to an increasing amount of research on integrating a variety of heterogeneous and autonomous GISs into a cooperative environment to construct a new generation of GIS featuring in open architecture, distributed computing capability, interoperability and extensibility. On the other hand, the rapid emerging of mobile Internet and the constant increasing of wireless subscribers number provide new opportunities for geographic information services. Putting the Internet in the palm of a hand will enable us to access to geographic information with personal devices anytime, anywhere, and anyhow. Promising research and application in this context includes navigation systems for vehicles, advanced GPS receivers that are able to display geographical maps.

In the past decade, a significant amount of research has been done to design interoperable systems in which collections of autonomous and heterogeneous GISs can cooperate to carry out tasks. Generally, we can classify these systems based on three levels of integration: platform level, syntactical level and application level (Abel, 1998). However, as far as system architecture is concerned, current solutions for integration of distributed GIS applications are based on either C/S or B/S architectural model. But the inherent limitations of these architectures, i.e., requiring a proper bandwidth, high quality and stable performance of the network connection, and less supporting of group awareness and high-level cooperation, make them incompetent to fulfil various requirements of dynamic, complicated and distributed computing, especially for the case of mobile network application environment, because of its limitations of mobile devices (limited power, memory and

computational power) and wireless communication networks (low bandwidth, frequent disconnections, and long latency).

Such a situation calls for both new architecture and distributed computing paradigm for establishing distributed GIS under Internet environment.

Mobile agent is a recently developed computing paradigm that offers a full-featured infrastructure for the development and management of network-efficient applications for accessing information anytime, anywhere, and on both wire-based and wireless devices. Mobile agents are programs that can migrate autonomously from one host to another during their execution on behalf of their owners or creators. The agent-based computing can benefit Internet (especially mobile Internet) applications on providing asynchronous task execution and more dynamics, supporting flexible and extensible cooperation, reducing communication bandwidth, enhancing real time abilities and higher degree of robustness, enabling off-line processing and disconnected operation. Up to date, several application-oriented mobile-agent systems or projects have been reported in the areas of information retrieval, network management and electronic commerce, *etc.* However, no such systems related to spatial data retrieval and manipulation has ever been reported. Considering the advantages of mobile agents over traditional distributed computing technologies, it is worthwhile to introduce mobile agents into distributed GIS building.

On the other hand, XML (extensible Markup Language) has been recognized as a good method for web-based information integration. Related studies also suggested that XML has the potential capability to serve as a common protocol for information integration in a web environment. GML (Geography Markup Language) is the instantiation of XML in

\*This work has been supported by the Natural Science Fund of Hubei Province of China under research contract 2001ABB041

geo-spatial domain (OGC, 2000); it provides a grammar for decoding geo- spatial content. So it is natural to utilize GML for integrating GIS data.

In this paper, we propose a framework for integrating distributed GIS based on mobile agent and GML. It is the improvement of our MADGIS system architecture proposed earlier (Guan, 2001). This framework can be seen as a federated database approach based system for integrating distributed GIS by using mobile agent and GML technologies. Here, we use mobile agent to overcome the inadequacy of other traditional distributed computing paradigms, and use GML to solve the heterogeneity problem of various GISs.

## 2. MOBILE AGENT AND GML BASED FRAMEWORK

### 2.1 Overview

The proposed framework is based on the techniques of Java Applet, Mobile Agent and GML. The system implemented on the framework involves client sites, sever sites, Internet/Intranet connecting these sites, and mobile agents roaming on the Internet/Intranet for retrieving information on behalf of the users. Fig. 1 shows the infrastructure of the system.

In Fig. 1, a client site is a client machine, which is used for query submission and results visualization. A server site is also a GIS server, which provides spatial information services for local or remote requests. A user submits a query from a client machine to a server *via* web browser. The query is analyzed and optimized by the server, from which one or multiple mobile agents are created and dispatched to finish the query task cooperatively. Each mobile agent along with its sub-tasks travels from one remote server to another to gather the related information. Retrieved information is then taken back to its original site after the mobile agent finish its mission. All returned information is further merged there and presented to the user. The servers also provide docking facility for mobile

agents in case they cannot travel to the destinations promptly due to network problems.

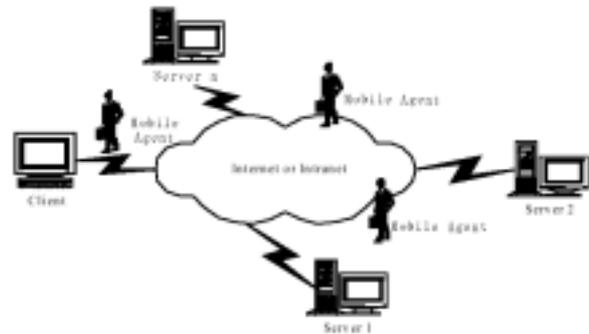


Figure 1. Framework Overview

### 2.2 The Client Site

A user can access to any GIS site within the system *via* a client machine or a local server. First, a user should log in one server in the system. Then the server returns a Web page to the client, in which there is a Java Applet termed as Client-Applet composed of one Mobile Agent Environment (MAE), one stationary agent and one mobile agent. The Client-Applet is executed at the client site to establish the MAE for the client and to start the stationary agent encoded in the Client-Applet. We call this stationary agent Client-Agent, which is responsible for the following two things:

- 1) To obtain the Data Sources Description Information (DSDI) from the server first visited. DSDI is something like a data dictionary or a navigation graph for data.
- 2) To create a Query Interface (QI) in the Web browser from which the user submits queries and gets retrieved information.

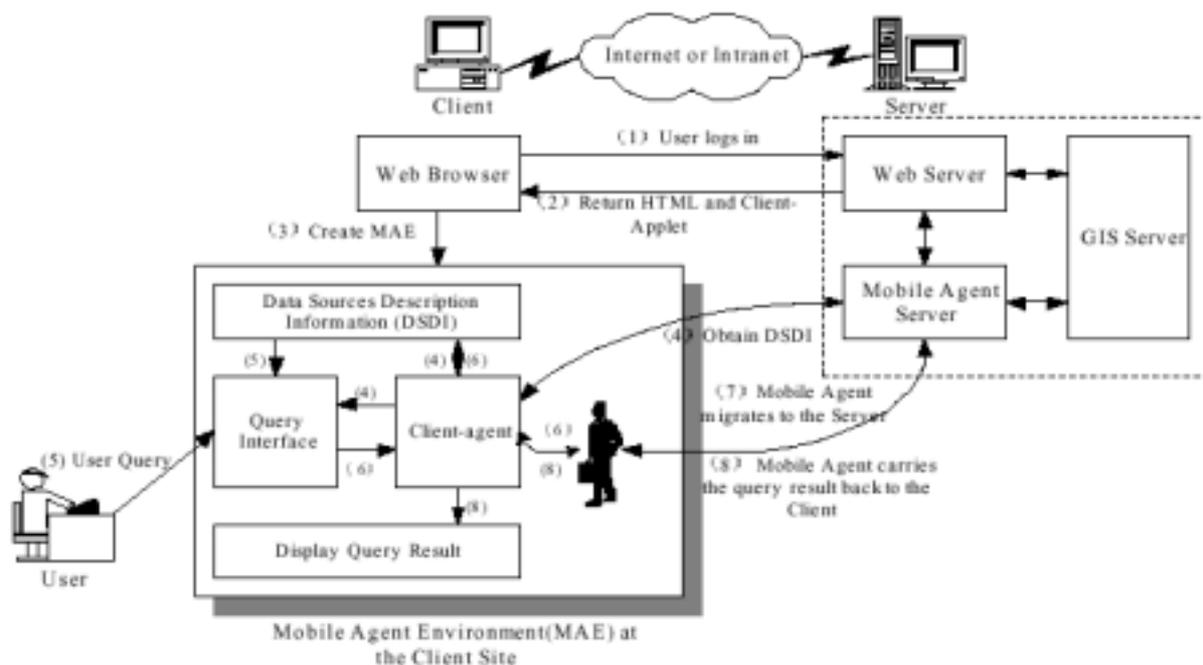


Figure 2. The Client Site

According to this, the query environment is set up at the client site. What a user can see is only the QI, while Client-Agent, mobile agent and its execution environment are at the backend. However, the user has to first login a server by Web browser. After a successful legal login, the user can start query operations. Then the server accessed takes charge of query processing and mobile agents manipulation. Typically, a whole query session consists of the following main steps:

- 1) At a client site, the user visits one server through Web browser by specifying the server site's URL.
- 2) The accessed server returns a Web page and the HTML page includes the Client-Applet.
- 3) The Client-Applet is executed on the client machine to establish the MAE at the client site and to start the stationary Client-Agent.
- 4) The Client-Agent obtains the DSDI and the QI.
- 5) A user constructs and submits queries through the QI.
- 6) When the Client-Agent gets user's query, it first does initial parsing and grammar detecting, and then creates mobile agent to take charge of the query tasks.
- 7) Mobile agent with user's query tasks leaves the client site and migrates to a server for query processing.
- 8) After the query task is fulfilled at the server site, the mobile agent migrates back to the client site and shows the result to the user via web browser.

The process described above and the interaction among the client site, server sites and mobile agents are demonstrated in Fig. 2.

### 2.3 The Server Site

A server in the system at least consists of three major components: Web server (or simply WServer), GIS server (GServer), and Mobile Agent server (MAServer). WServer is the interface for a user to connect to the system, and is responsible for providing Client-Applet to the user. GServer is composed of GIS database, spatial query processing and analyzing components, which provides the support for local query

processing and analyzing. MAServer is a common place for mobile agents' activities such as docking or routing or partial results transferring, and it provides interfaces to mobile agents for processing or analyzing local data. Certainly, there are some other components that are not our main concern here. Fig. 3 illustrates the architecture of a typical server in our system. And next the MAServer is discussed in detail.

MAServer consists of four stationary agents: Local Services Agent (LSA), Query Optimization Agent (QOA), Querying and Wrapping Agent (QWA) and Mediation and Transformation Agent (MTA). Their functions are outlined respectively as follows.

LSA has following duties:

- providing system's DSDI for clients who are logged in the server;
- maintaining the DSDI stored locally;
- maintaining the statistical information of the data resources stored locally;
- sensing and detecting the status of network traffic between the local server and other remote servers or other clients.

QOA is responsible for analyzing and optimizing users' queries. A user's query is first processed simply by Client-Agent through parsing and grammar detecting, and then the query is taken to the server site by a mobile agent (as termed main query agent). Then QOA takes charge of the query's optimization analysis and set the query strategy accordingly.

QWA is responsible for fulfilling the query tasks that the mobile agent brings and wrapping the query result to standard GML document. When a query has something to do with multiple data sources or sites, information retrieved from different sites has to be merged, which is taken by MTA. The query results integrated have to be transformed into standard SVG (W3C, 2000) documents.

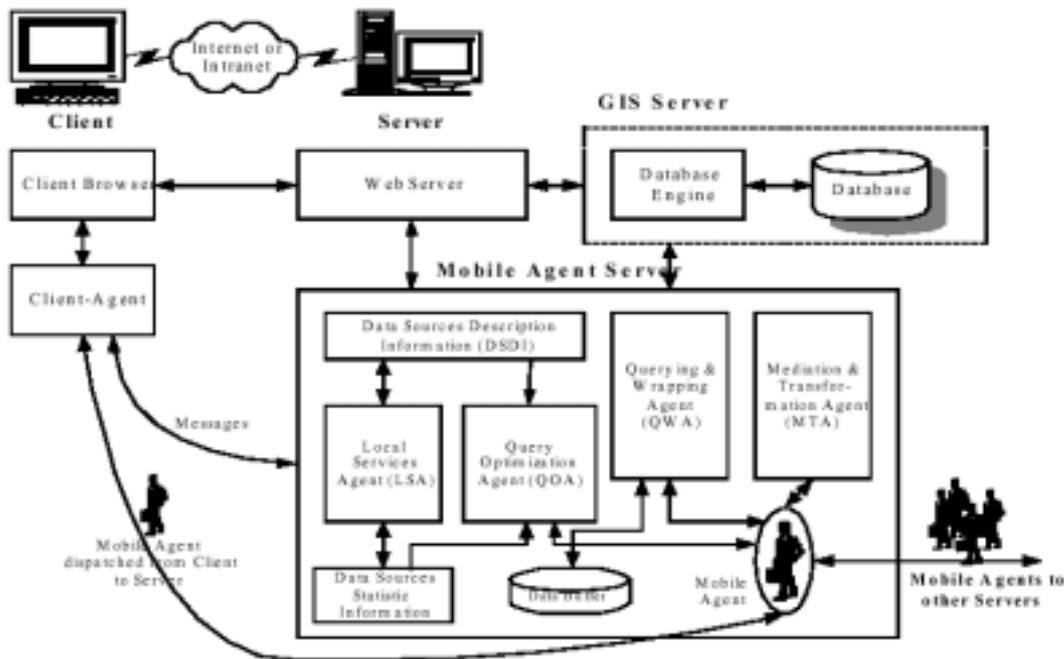


Figure 3. The Server Site

Fig. 3 also shows the procedure of how a query task is fulfilled by using mobile agent technology. It includes following steps:

- 1) The query task is given to QOA for optimisation. The goal of optimisation is to reduce computation cost and network transmission volume caused by query processing. After optimisation a query plan is created which includes a set of sub-queries, sites on which sub-queries executed and the execution sequence of these sub-queries. QOA returns the result of query optimisation to the main query agent. Then the main query agent decides whether to create sub-agent to go on with the query task.
- 2) If the query only concerns data in local database, then the main query agent will go on to finish the query task itself. The main query agent assigns the query task to the QWA, who is in charge of finishing the query and wrapping the results.
- 3) If the query concerns data from multiple sites, and the sub-queries are executed serialized strictly, then the main query agent or a mobile agent created by the main query agent will take the query job. The agent is dispatched according to its itinerary set previously, and each sub-query is related to one site accordingly and the sub-queries will be fulfilled one site by one site. When the last sub-query is finished, the dispatched agent can go back to the client site directly or to its original server site.
- 4) If the query concerns data from multiple sites, and the sub-queries can be executed parallel, and the network connections between the local site and other sites related to sub-queries are smooth, and then multiple mobile agents are created by the main query agent to execute sub-queries parallel so as to gain better efficiency. The main query agent can join the group to finish the task or just keep the role of coordinator of the multiple agents.
- 5) Query results gained from all related sites are brought back to the original server, then the MTA at the server site does data integration and transformation and creates one standard SVG (Scalable Vector Graphics) document as the final result of the query. The wrapping and mediation of geographical data are discussed in the next section.
- 6) The main query agent carries the final query result in SVG format to the client site, and shows the result to the user via browser. That's the end of the user's query.

Another function of the server is to provide the docking mechanism for mobile agents when the connection between the current site and the destination site of the migration has some

fault. LSA will take the role of agency for mobile agent's activation and deactivation when such thing happens.

## 2.4 Mobile Agent Environment

Mobile Agent Environment (MAE) exists both at the client sites and the server sites. It provides an environment for mobile agent creating, executing, dispatching and migrating. Except for the mobile agent, it includes the following functional modules: Mobile Agent Manager (MAM), Mobile Agent Transportation (MAT), Mobile Agent Naming (MAN), Mobile Agent Communication (MAC), and Mobile Agent Security (MAS). Figure 4 gives the structure of MAE. Its components' functions are outlined respectively as follows.

- MAM, the heart of MAE, is responsible for all kinds of management of mobile agents. It mainly provides a comprehensive environment for agent creating and executing, basic functions to make mobile agent code and data migrate precisely to its destination, functions for agent scheduling locally, support for agent's remote management.
- MAT controls the transferring of mobile agents, *i.e.*, sending and receiving mobile agents to and from remote sites.
- MAN manages mobile agents' naming service, which provides the mechanism of tracing mobile agents.
- MAC serves communication, which provides support for cooperation, events transmission among agents.
- MAS provides a two-facet security mechanism. On one hand, it is responsible for distinguishing users and authenticating their mobile agents in order to protect server' resources from being illegally accessed or even maliciously attacked. On the other hand, it ensures mobile agents not be tampered by malicious hosts or other agents.

## 2.5 Remarks of the Framework

The framework we proposed simplifies the structure of clients and reduces the load of network, and what's more it is easy for the system to deal with heterogeneity. The framework is adaptable to Internet especially for Mobile Internet application environment with limited bandwidth and vulnerable connection, in which mobile agent is the kernel technology and makes it the cornerstone of constructing distributed Web GIS. Mobile agent in deed is a kind of program that can migrate autonomously to

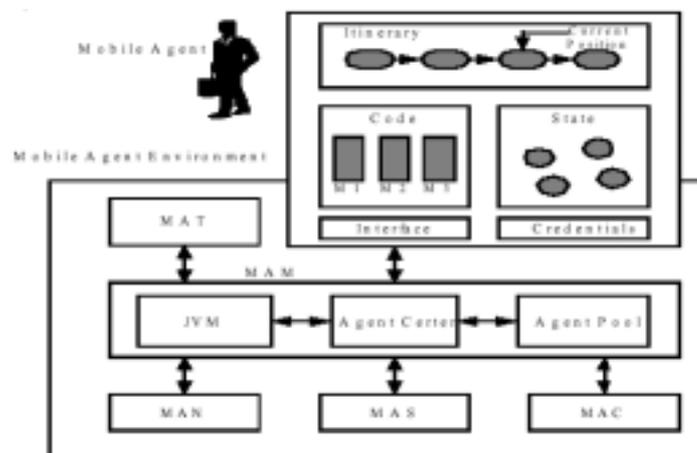


Figure 4. The structure of MAE

remote servers on behalf of its owner to fulfil the information retrieval tasks. The autonomous mobility of mobile agent is used to overcome the drawbacks of distributed systems on Internet under traditional Client/Server or Browser/Server architecture. The key technique features of our proposed framework are: Java Applet is downloaded from server to the client site to establish Mobile Agent Environment (MAE), which provides the environment for stable agents' execution and for mobile agents' creation, execution, dispatch and migration, and which is also in charge of managing agents and providing related communication and security mechanism.

Mobile agents are created in need by MAE and roam on the network for searching computing, data, information and services resources so as to finish the users' query tasks. GML is adopted as the standard format to integrate geographic data. That is, data retrieved by mobile agents from different GISs is represented and merged in GML format, and then are transformed to SVG format for Web browsers' presentation at the client sites. This is a new way for GISs integration under distributed environment with heterogeneous data resource. Integration and heterogeneity handling of the system are discussed in the following section.

### 3. INTEGRATION AND HETEROGENEITY HANDLING

In reality, any two GISs may be in different models or/and developed using different software packages, and data in the two GISs may be in different structures and semantic meaning may be expressed in different ways (Castano, 1999). Such a situation makes handling heterogeneity in data structures and semantics a major barrier in integrating distributed GISs.

In our system, an approach for geographical information integration based on GML is introduced. GML is adopted as the common format of geographical information exchange; and the mediator-based architecture is utilized. Stationary agents at server sites take the roles of mediator and wrapper. That is to say, QWA takes the role of wrapper, and QOA and MTA together act as mediator, as shown in Figure 5.

In order to handle the heterogeneity in GIS schemas and geo-spatial data structures as well as semantic conflicts caused by map and attributes names, common data schema, common data format and common query expression are used in our system.

The common schema provides definitions of all the features and feature collections accessible to the user. It is based on the OpenGIS specification. So it can be understandable to all data objects in the system. The common data format is based on the Spatial Archive and Interchange Format (SAIF), which uses an object-oriented approach and is possible to model most kinds of spatial data. In the common data format, geo-spatial data are structures as objects and are organized according to a class hierarchy described in OpenGIS (OGC, 1998). In our system, we use Document Type Definition (DTD) to describe the data schema and data format. That is to say, we use one DTD to represent the common data schema and the common data format, i.e. DSDI as we described in section 2. This DTD is called Global DTD (or G-DTD), while the local information of each server can be described by a Local DTD (L-DTD) respectively. Data wrapping is the procedure for data transformed from L-DTD to G-DTD. And the transformation from GML document to SVG document for the query results is in deed the transformation between different DTDs.

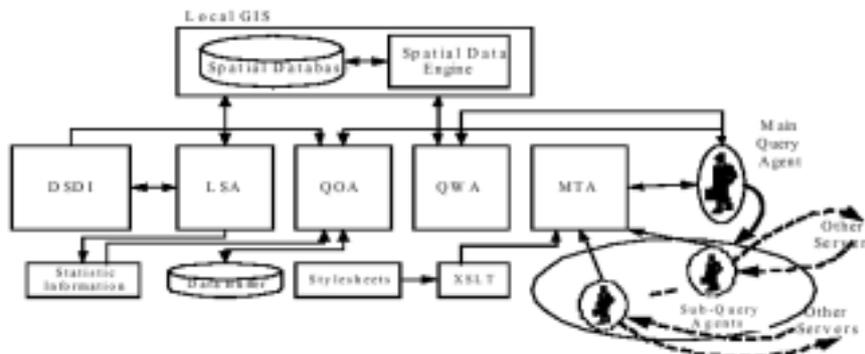


Figure 5. Integration framework based on Mobile Agent & GML

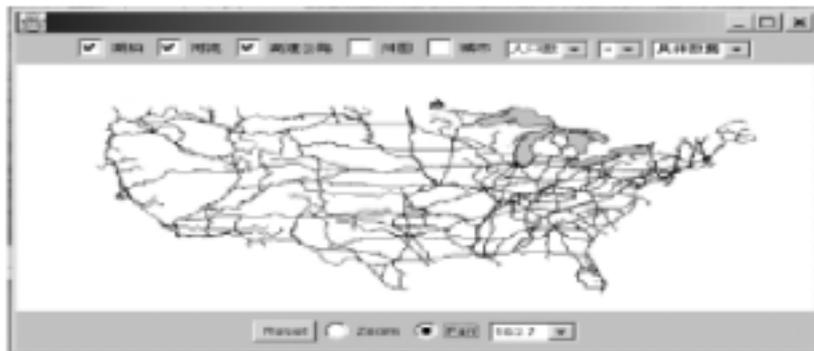


Figure 6. The Prototype

Based on the common query expression, a query has three major parts: an input list, an output list and a search condition. Sub-query transferring between different server sites is based on the common query expression. When adopting common query expression, common data schema and common data format, the user's query implies two transformation procedures: transformation from common query expression to local query expression, and transformation from local data schema (format) to common data schema (format).

#### 4. PROTOTYPE

In order to assess the feasibility of the architecture, a simplified prototype was developed. We use the Aglet Software Development Kit (ASDK) 2.0 (Aglets, 2001), JDK 1.3 and Geotools 0.8.0 (Geotools, 2001) for implementing the prototype. Two servers are implemented in the prototype, which are termed Server-A and Server-B. Geographic information of the states, cities, rivers, roads, and lakes of the USA from ArcView GIS 3.2 are split into two parts to store on the two servers Server-A and Server-B respectively. Figure 6 shows the interface of the prototype and the query result for roads, rivers and lakes.

#### 5. CONCLUSION

Internet has greatly changed the ways of data accessing, sharing and disseminating. The emergence of mobile agent technologies brings new opportunities and challenges to Internet based distributed computing applications. In order to fulfil the requirements of distributed GIS applications under Internet environment with limited communication bandwidth and unstable communication connectivity, we propose a framework for integrating distributed Web-based GISs by using mobile agent and GML technologies. The proposed architecture is based on the techniques of Java Applet, Mobile Agent and GML. The system implemented on this framework consists of clients, servers and mobile agents. The framework simplifies the structure of clients and reduces the load of network, and what's more it is easy for the system to deal with heterogeneity. The framework is adaptable to Internet especially for Mobile Internet application environment with limited bandwidth and vulnerable connection, in which mobile agent is the kernel technology and makes it the cornerstone of constructing distributed Web GIS. This paper presents the framework of the system and integration and heterogeneity handling based on mobile agent and GML. A simplified prototype was also established to assess the feasibility of the proposed approach.

#### REFERENCES

- Abel, D., 1998. Towards integrated geographical information geo-processing. *Int. Journal of Geographical Information Science*, 12, pp. 353-371.
- Aglets: <http://www.trl.ibm.co.jp/aglets>. (accessed 28 Sep. 2001)
- Geotools: <http://sourceforge.net/project/> (accessed 8 Oct. 2001).
- Guan, J., 2001. Mobile-agent based distributed Web GIS. In *Proceedings of 9th International Conference on Cooperative Information Systems (CoopIS'2001)*. LNCS Vol.2172, Springer-Verlag. Trento, Italy. pp. 53-66.

Lange, D.B., 1999. Seven Good Reasons for Mobile Agents. *Communications of the ACM*, 42(3), pp. 88-89.

Open GIS Consortium (OGC), 1998. OpenGIS Simple Features Specification for SQL (Revision 1). <http://www.opengis.org/techno/specs.htm> (accessed 6 Sep. 2001).

Open GIS Consortium (OGC), 2000. OpenGIS Geography Markup Language. <http://www.opengis.org/techno/specs>. (accessed 5 Nov. 2001)

W3C Scalable Vector Graphics (SVG) 1.0 Specification, 2000. <http://www.w3.org/TR/2002/CR-SVG-20000802/>. (accessed 10 Mar. 2002).