Computing and Informatics, Vol. 24, 2005, 201–222

AN AGENT APPROACH TO SPATIAL INFORMATION GRID ARCHITECTURE DESIGN

Yingwei Luo, Xiaolin Wang, Zhuoqun Xu

Department of Computer Science and Technology
Peking University, Beijing, China, 100871
e-mail: lyw@pku.edu.cn

Manuscript received 2 September 2004; revised 11 April 2005
Communicated by Ladislav Hluchý

Abstract. Spatial information grid (SIG) is a spatial information infrastructure that has the capability of providing services on-demand. SIG is a distributed network environment, which links spatial data resources, computing resources, storage resources, software, tools and users. SIG can integrate massive distributed heterogeneous spatial information resources, provides uniform management and process, and, furthermore, coordinate different resources to complete large-scale and complex spatial tasks and applications. In this paper, agent technology is adopted to construct a SIG framework, which contains three layers: users/applications layer, agent services layer and information layer. Different applications can get their spatial information via agent services, and agent services make the procedure of navigating and accessing spatial information transparent to users. Also, the implementation issues of the framework are discussed in detail, including Geo-Agents, an agent-based distributed GIS system, spatial information management, collaboration and parallel mechanism, load control strategy, and a sample.

Keywords: Spatial information grid, geo-agents, framework, spatial metadata, collaboration, parallel, load

1 INTRODUCTION

Spatial information has an obvious feature of distribution. On the one hand, spatial information covers a scope of the whole planet where people live; it is a distribution of spatiality. On the other hand, spatial information is related to every aspect of human society; it is a distribution of social characteristics. Just because of the nature of distribution of spatial information, it is mandatory to organize and maintain all spatial information in decentralized way. It is the best choice to organize and maintain spatial information by different departments in different places. In fact, people do organize and maintain spatial data like that.

With the development of computer networks, the use of spatial information is facing a new challenge. More and more heterogeneous spatial information is connected with the network and opened to others. However, users do not know where spatial information is located, what kinds of spatial information can be used, and how to use them. Also, some large-scale and complex spatial applications (for example, Digital City) may involve different departments and need the cooperation of distributed spatial information to achieve perfect result [1].

Because of the unceasing increase of spatial information, it is urgent for people to make spatial information available and shareable. New mechanisms that allow spatial information to inter-operate are needed. In this paper, an agent-based spatial information grid (SIG) framework is presented for synergetic management of heterogeneous spatial information on a network.

In the aspect of building a spatial information-sharing framework, many works are accomplished by research institutions and standardization organizations, especially OGC and ISO TC211.

The Open Geospatial Consortium, Inc. (OGC) is a non-profit, international, voluntary consensus standards organization that is leading the development of standards for geospatial and location based services [2]. Through member-driven consensus programs, OGC works with government, private industry, and academia to create open and extensible software application programming interfaces for geographic information systems (GIS) and other mainstream technologies. OGC's Simple Feature Model (SFM) summarizes the common sense to spatial information of people. Based on SFM, Geographical Markup Language (GML) is given, which is an XML encoding for the transport and storage of geographic information, including both the geometry and properties of geographic features. In order to manage spatial data, services and applications, OGC defines data metadata, services metadata and application metadata specifications. Catalogue Services Specifications (CSS) are organized and implemented for the discovery, retrieval and management of metadata. OGC's Interoperability Program (IP) is used to help the participants mutually form engineering methods of concord spatial interfaces. OGC also gives a common architectural framework for OGC Web Services (Web Services Architecture, WSA), where service model is described, roles and operations in the model are identified, and concept level, technique level and system level are divided. These three levels can define and gradually implement Web services. For interfaces of Web services, OGC also has specific discussion and presented interface specification of Web services such as Web Feature Service

(WFS), Web Map Service (WMS), Geolinking Service (GLS), etc. The work of OGC greatly enlightens and guilds the study in spatial information sharing and interoperation.

The work of ISO TC211 aims at establishing a structured set of standards for information concerning objects or phenomena that are directly or indirectly associated with a location relative to the Earth [3]. The work of TC211 is similar to that of OGC, but compared to OGC it is focused on cooperation between governments to come to international specifications, while OGC focuses on folk cooperation of people.

The specification and standards of OGC and ISO TC211 guide developers to adopt different implementing technologies (such as COM/DCOM, CORBA, Web Service, Grid Computing and Agent) to exploit middleware, component and integrating infrastructure to help users share massive distributed spatial resources.

Grid is a unified and integrated approach to build distributed scientific computing environments that incorporate computation, data management, scientific instruments and human collaboration to support large-scale scientific and engineering applications [4]. It provides standardized middleware services within geographically dispersed and inter-organizational environments, and greatly simplifies the construction and operation of the application system that interconnect various resources. At present, there are some productive grid systems and toolkits. The Argonne lab and ISI have developed a range of basic service – Globus toolkits to support efficient and effective computation in grid environments [5]. These services have been deployed in numerous projects ranging from collaborative design to distributed supercomputing. Also, a new open grid service architecture OGSA is proposed in 2002 [6].

Grid technology has been applied in many scientific and engineering applications. There are some projects that focus on applying grid computing into spatial information sharing and interoperation, such as Earth System Grid (ESG) [7], Earth Observation System (EOS) [8], and Spatial Data Processing Grid (SDPG) [9]. These projects adopt different implementing technologies to design layer architecture, and most of them adopt OGC's specification to unify spatial resources. ESG uses Globus toolkits and other data grid software to construct a prototype system, and implement the interactive analysis of remote climate data by moving and replicating data collections, and sharing computing resources. EOS uses CORBA to construct an easy-to-use and grid-based EO application system, and will be widely employed in climatology, oceanography, geography and other earth observation subjects. ESG and EOS are just domain-oriented application systems, and cannot be directly used in other applications. SDPG focuses on virtual data object and OPENGIS to unify access to various storage systems and GIS systems. SDPG follows open grid service architecture (OGSA) and uses Web Service to build a service framework, which supports the movement of service and data. Cluster-based GIS system is also proposed to improve performance of spatial data processing and service in SDPG.

2 AGENT-BASED SIG FRAMEWORK

2.1 Spatial Information Grid

SIG (Spatial Information Grid) is a spatial information infrastructure that has the capability of providing services ondemand. SIG integrates and shares massive distributed heterogeneous spatial information resources, and provides uniform management and process. SIG is a distributed network environment, which links spatial data resources, computing resources, storage resources, software, tools and users. SIG can coordinate different spatial information resources to complete different spatial tasks and applications. In such an environment, users can present all kinds of requests for spatial data and its process, and SIG can joint distributed data, computing, network and software resources to cooperate and accomplish different users' requests. SIG adopts a new architecture, method and technology to manage, access, analyze and integrate distributed spatial data, take full advantage of services from existing spatial information systems, so as to realize effective spatial information share and interoperation, and provides on-line spatial information analyzing processes and services. SIG should provide the following functions [10]:

- The capability of processing massive spatial data. Storing, accessing and managing massive spatial data from TB to PB; efficiently analyzing and processing spatial data to produce model, information and knowledge; and providing 3D and multimedia visualization services.
- The capability of high performance computing and processing on spatial information. Solving spatial problems with high precision, high quality, and on a large scale; and process spatial information in real time or on time, with high speed and high efficiency.
- The capability of sharing spatial resources. Sharing distributed heterogeneous spatial information resources and realizing interlink and interoperation at application level, so as to make best use of spatial information resources, such as computing resources, storage devices, spatial data (integrating from GIS, RS and GPS), spatial applications and services, GIS platforms (such as ESRI ArcInfo, MapInfo, ...), ...
- The capability of integrating legacy GIS system. A SIG can not only be used to construct new advanced spatial application systems, but also to integrate legacy GIS system, so as to keep extensibility and inheritance and guarantee investment of users.
- The capability of collaboration. Large-scale spatial information applications and services always involve different departments in different geographic places, so remote and uniform services are needed.
- The capability of supporting integration of heterogeneous systems. Large-scale spatial information systems are always synthetical applications, so SIG should

provide interoperation and consistency through adopting open and applied technology standards.

- The capability of adapting dynamic changes. Business requirements, application patterns, management strategies, and IT products always change endlessly for any departments, so SIG should be self-adaptive.

2.2 SIG: An Agent Framework

Agent is an autonomous, interactive, initiative and reactive computing entity in a distributed environment. Agent encapsulates some computing resources and can reach its designed goals initiatively. An agent is not only able to work on itself, but also to impact environment, to receive feedback information from environment, and to readjust its own behavior. At the same time, an agent can cooperate with other agents. Agent system reduces the restrictions of concentricity, non-openness and sequential control, provides distributed controlling, dynamic emergency processing and parallel processing services [10, 11]:

- Using agent technology can simplify a big and complex problem, and disassemble the problem into small and simple ones. That is because agent can interact with outside through a flexible and context-sensitive method, not through some predefined interfaces.
- In real world, there are many problems that involve the cooperation of numerous distributed problem-solving entities and data resources. Agent technology provides a natural modeling method to solve those problems. The entities and their relations in real word can be easily mapped to agents, which have their own resources and have the capabilities to collaborate to solve problems.
- Several agents can be grouped into a team to complete a special complex task. Agents in the same team can coordinate their activities and collaborate to solve conflicts among them in order to fulfil team goals.

Agent technology will provide a new thought and method for processing massive distributed heterogeneous spatial information efficiently [12, 13]. An agent-based SIG framework is illustrated in Figure 1. This framework contains three layers: Users/applications layer, agent services layer and information layer. Different applications can get their spatial information via agent services, and agent services make the procedure of navigating, accessing and processing spatial information transparent to users. The exchange format of spatial information between applications and agent services is XMLized (such as GML 1.0/2.0/3.0) [2]. The agent services layer plays a most important role in the framework.

Interface Agent presents personalized UI (user interface) for different users in different applications. Get users' requests, communicate with the facilitator, and then deal with these requests and return results to users. By the facilitator's

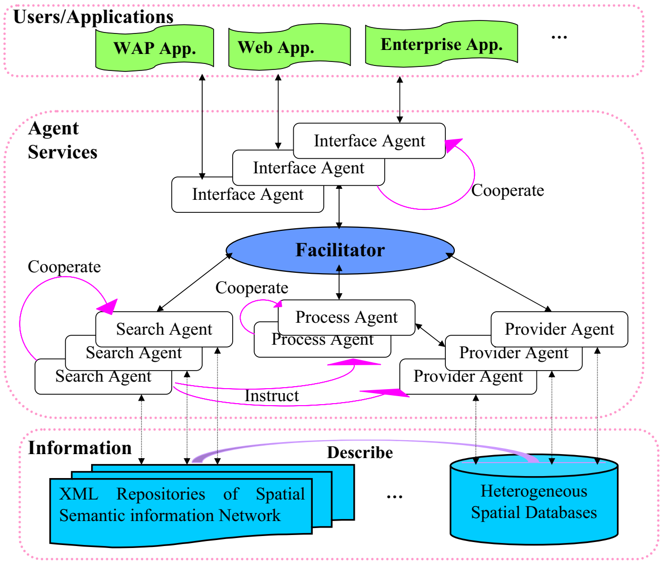

Fig. 1. SIG – an agent framework

help, different interface agents can negotiate with each other to combine or re-plan these requests so that these requests can be replied more rationally and effectively.

Facilitator administrates agents in the system, including register and messages from agents. Facilitator also prompts cooperation among agents, including initializing negotiation among interface agents; acting as a bridge lying between interface agents and search agents, provider agents and process agents; scheduling search agents to navigate users' requests; helping search agents find proper provider agent to access to spatial database, and assigning process agents to filter spatial information.

Search Agent is interested in finding and locating spatial information by the users' requests. XML repositories of spatial semantic information network that describe different heterogeneous spatial databases support the whole searching procedure of search agents. By the facilitator's scheduling, search agents can become parallel to carry one request. As the result, search agents can acquire the constitution and the unique locators of spatial information.

Provider Agent accesses spatial information from a right spatial database under instructed by the result of search agent. Because the spatial databases are heterogeneous, provider agents are constructed based on different GIS products. Provider agent also translates spatial information to XMLized spatial information (such as GML 1.0/2.0/3.0).

Process Agent filters or queries XMLized spatial information according to the users' requests.

2.3 Geo-Agents: An Agent Implementation

Aiming at the above-mentioned SIG framework, an agent-based distributed GIS Geo-Agents is designed and implemented (Figure 2) [14, 15]. Geo-Agents consists of four types of GIS agents: Facilitator, Interface agent, GIS function agent and GuServer, where Facilitator and Interface agent are identical with those in the above SIG framework, and GIS function agent is mapping to Search Agent and Process Agent; GuServer is mapping to Provider Agent in SIG framework.

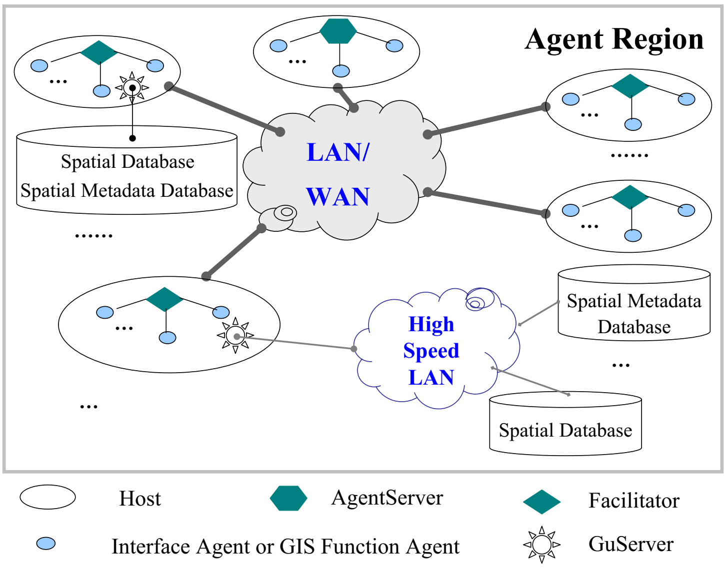

Fig. 2. The Geo-Agents system

Facilitator is the manager of Geo-Agents. The functions of the Facilitator include registering available GIS agents, searching for practicable GIS agents, managing all active agent instances, coordinating communication and coordinating cooperation.

GIS function agent encapsulates spatial querying, spatial processing or spatial analyzing services. The encapsulated services in GIS function agent may come from different existing GIS platforms. Each GIS function agent can complete the same type of problem. According to the features of GIS, GIS function agent can be classified into other two types, namely basic function agent and domain-oriented function agent. The basic function agent completes basic GIS services, such as spatial data search, spatial data access, network analysis, overlay analysis, buffer analysis and so on. Domain-oriented function agent is responsible for application tasks in various domains, and can be constructed by domain-oriented model and used generally in one domain.

The interface agent provides interfaces for users or applications to hand tasks. Just like Structure Query Language (SQL) of database, Geo-Agents provides GeoScript, an agent manipulating language to describe GIS tasks [15]. When solving a practicable problem, users or applications can simply use GeoScript statements to describe the task, and then hand the statements to the interface agent. Interface agent has a GeoScript interpreter and can disassemble the task to subtasks autonomously, recruit GIS function agents to complete the task concurrently.

GuServer is in charge of spatial information accessing services, which manages spatial information and spatial metadata in spatial databases.

In Geo-Agents, every GIS agent consists of five units: control subsystem, functional subsystem, communication subsystem, human-computer interface and data resource. GIS agent is a reactive agent, which is not only able to carry out its own task independently, but also to communicate with other agents, to exchange information and to cooperate with others.

The Agent Region mode is used to control the distributed scenario of Geo-Agents in a network. An Agent Region consists of one or more hosts, which must be installed with the Facilitator (and/or other GIS agents). There are many Facilitators in an Agent Region. One Facilitator can cooperate with other Facilitators to control and coordinate every GIS agent to run correctly, and hold the distributed controls of the whole system.

In an Agent Region, there is only one Facilitator that will be configured as AgentServer. The Facilitator is used to manage and coordinate agents inside one Agent Region. Beside the functions of Facilitator, AgentServer serves as a bridge among different Agent Regions. That is to say, an agent in one Agent Region can only communicate with agents in other Agent Regions through AgentServer. Of course, an authorization is required. The AgentServer manages a table to register other Agent Regions that are authorized with itself each other. Between two directly authorized Agent Regions, agents between them can communicate via the coordination of their AgentServers. The authorization relation in Geo-Agents can be passed one by one: If a directly authorized AgentServer chain can be found for two Agent Regions that have no direct authorization, agents between the two Agent Regions can communicate via the coordination of the AgentServer chain.

3 SPATIAL INFORMATION MANAGEMENT

3.1 Spatial Metadata

In order to manage distributed spatial information more effectively in a network, based on the extended contents of spatial metadata, a hierarchical spatial metadata database (metabase) framework is designed (illustrated in Figure 3), where the spatial metadata not only contains the semantic information about simple spatial information, but also the semantic information about field-oriented spatial information set [16]. The hierarchical spatial metabase framework constitutes the semantic network for spatial information.

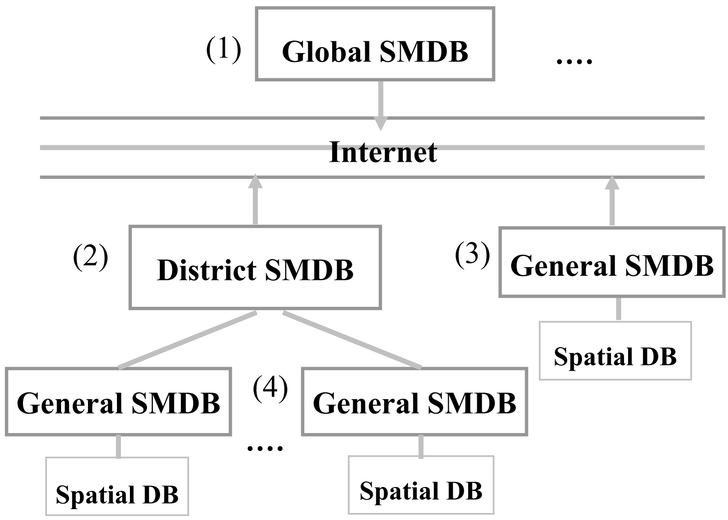

Fig. 3. The framework of spatial metadata database

The spatial metabases shown in Figure 3 are divided into three levels: (1) is Global Spatial Metabase (Global SMDB), (2) is District Spatial Metabase (District SMDB), (3) and (4) are General Spatial Metabase about Spatial Database (General SMDB). (3) and (4) are the same in content, but they are in different position in the framework. (1), (2) and (3) are regarded as independent spatial metabase. District SMDB and independent General SMDB can, but do not have to, be registered in a Global SMDB; General SMDB indicated by (4) must be registered in a District SMDB. The General SMDBs registered in the same District SMDB are correlative.

General SMDB has an all-sided description about spatial database, which is designed after OGC's Simple Feature Model and FGDC's Geospatial Metadata Standard. District SMDB registers all metadata information of General SMDBs registered in the district and the relations among those General SMDBs. Global SMDB manages a number of District SMDBs and independent General SMDBs, and all metadata information of those SMDBs (of course, including the General SMDBs registered in District SMDB) are registered in it. Global SMDB is the top-level index for spatial data, so users can locate the target data quickly through Global SMDB.

3.2 Agent-Based Spatial Data Navigation Mechanism

According to the spatial data requirement specification, and with the help of spatial metadata framework, we design five types of GIS function agents to cooperate to complete spatial data navigation using Geo-Agents: data accessing agent (DAA), location searching agent (LSA), DA controlling agent, LS controlling agent and data mapping agent. Figure 4 illustrates the relation of those five types of agents.

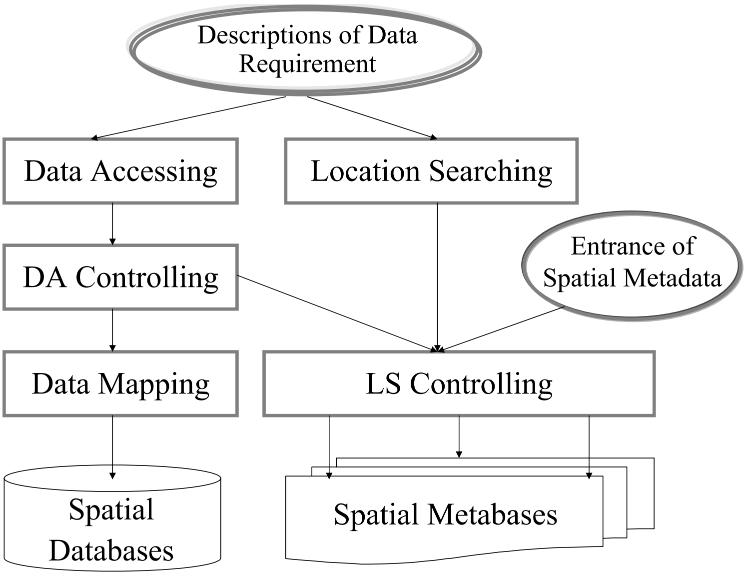

Fig. 4. The relations among data searching agents

The data accessing and the location searching agents receive a requirement, and interpret it to a comparable requirement task that can be recognized by computer. The data accessing agent hands the requirement task to DA controlling agent and then waits for the content of the target spatial data. The location searching agent hands the requirement task to the LS controlling agent and the result is the map-layer members and their locations of the target spatial data.

First, the DA controlling agent hands the requirement task to the LS controlling agent to obtain the map-layer members and their locations of the target spatial data; then it hands the result from the LS controlling agent and the requirement task to the data mapping agent. The data mapping agent interacts with spatial database and obtains the content of the target spatial data according to the map-layer members and their locations of the target spatial data and other related information. Because the target spatial data may be distributed in different spatial databases, the DA controlling agent can classify the map-layer members by spatial database according to their locations, divide the requirement task into some data accessing subtasks by the classification, and then hand those data accessing subtasks to different data mapping agents to obtain the content of the target spatial data from different spatial databases. Different data mapping agents can become parallel to increase the efficiency of data accessing.

When the LS controlling agent receives the requirement task, it can connect to several spatial metadata databases according to the entrance of SMDBs, look for and match the map-layer members and their locations of the target spatial data in different spatial metadata databases concurrently.

Every GIS task requires spatial data, even different types of spatial data. Thus, spatial data searching is taking place continually in a distributed GIS. Different GIS tasks might have the same requirement of spatial data. It is unnecessary to repeat spatial data searching for the same requirement. The result of spatial data searching should be reused.

The descriptive specification for spatial data requirement makes it possible to reuse the result of spatial data searching. In the five types of agents, DA controlling agent and LS controlling agent are worth reusing and can be reused. Firstly, they are transparent to users or applications. Secondly, they control the whole process of spatial data searching, and their results just meet the requirement of users or applications. Lastly, the requirement task that drives them is comparable and recognizable.

4 COLLABORATION AND PARALLEL

4.1 Collaboration

In human society, there are some relations among individuals, which make individual collection become human society, and make individuals become socialization people. Just like human society, multi-agent system must let its agents cooperate with each other to become an integrated system, so as to complete more complicated tasks.

The goal of Geo-Agents is to construct a multi-agent system like human society, so collaboration mechanism is necessary. There are two collaboration modes in Geo-Agents: collaboration among interface agents and collaboration under control of an interface agent.

4.1.1 Collaboration among Interface Agents

Collaboration among interface agents is coordinated by AgentServer. AgentServer manages a table to register different cooperation groups in one Agent Region. There are three types of collaboration among interface agents: direct cooperation, peer group cooperation and charge-tributary group cooperation.

Direct cooperation: during the execution of an interface agent, it needs some results of other interface agents (see Figure 5).

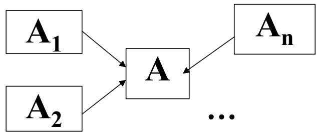

Fig. 5. Direct cooperation

Peer group cooperation: several interface agents form a team. All team members cooperate to complete a same complex task. Each member assumes a subtask, and accomplishes it independently. Team members are peers in the team: after completing a subtask, each member sends its result to all other members, so each member can obtain the same final result of the task (see Figure 6).

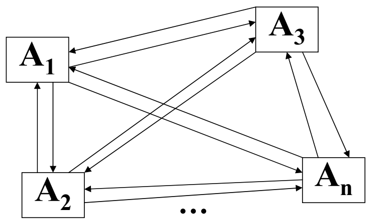

Fig. 6. Peer group cooperation

Charge-tributary group cooperation: like in peer group cooperation, all team members cooperate to complete a same complex task. However, there is the team-charger. Team member completes a subtask and sends its result to the team-charger. The team-charger assembles all sub-results to the final result of the task, and sends the final result to all team members (see Figure 7, where A is the team-charger).

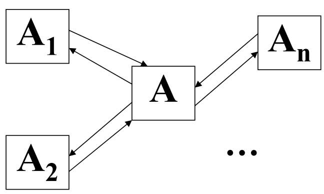

Fig. 7. Charge-tributary group cooperation

In Geo-Agents, the collaboration among interface agents still has some restrictions:

- Collaboration exists inside one Agent Region only. Collaboration among different Agent Regions is not supported now.
- Each team member can communicate with others only once in a group.
- The result type of each team member is the same in one group, and all members cooperate to complete the same task.
- In one group team, the number of team members is pre-fixed, and no team member can join the team dynamically.
- Communication among interface agents in one group can take place only after all team members are joined in.

4.1.2 Collaboration under Control of Interface Agent

After an interface agent accepts a complex task, it will recruit some GIS function agents, and then organize them to cooperate to complete the task. This is hierarchy collaboration under control of interface agent (see Figure 8).

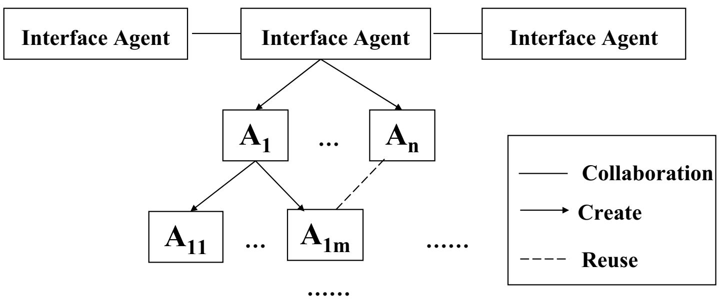

Fig. 8. Hierarchy collaboration under control of interface agent

The hierarchy of agents is similar to human society. Different agents play different roles in the system when cooperating to complete the same task. Interface agent locates at top-level. The up-level agents may create some under-level agents or reuse other existing agents to complete subtasks. Those agents who belong to the same parent can execute sequentially or concurrently.

In hierarchy collaboration:

- Facilitator/AgentServer is the collaboration coordinator;
- Up-level agents assign subtasks to down-level agents and down-level agents return to up-level agents;
- The results of some agents may be reused by others.

4.2 Parallel

In a multi-agent system, the smallest executing unit is an agent, and many agents can execute independently and concurrently. Geo-Agents likes a kind of cluster parallel system, and parallel is performed by GIS function agents. Parallel of GIS function agents takes place in collaboration under control of interface agent, and exists as three modes: isomorphic cooperative parallel, heterogeneous cooperative parallel and exclusive parallel.

Isomorphic cooperative parallel: parallel agents are of the same type, and each of them completes a subtask of the same complex task. Assembling all subtask results will get the final result. Isomorphic cooperative parallel meets two situations: one is that all agents process a different data resource; the other is that all agents process different parts of the same data resource.

Heterogeneous cooperative parallel: it is similar to isomorphic cooperative parallel, but parallel agents are of different types.

Exclusive parallel: parallel agents are of the same type, and complete the same task. However, the data resource each agent processes is different. Some agents may succeed, and some will fail. Of course, none knows aforehand which one will success or not. Only one successful result is needed. Once an agent returns successful result, all other agents must stop forcibly. If all agents fail, the task will fail too.

Although parallel is an intrinsic feature of a multi-agent system, the problem how to design parallel algorithms and construct suitable GIS function agents for real GIS problems in Geo-Agents still requires more attention.

Spatial data is often involved in a large spatial scope, and the content is diverse. So people store spatial data in different places according to spatial scope and content. Distribution is an intrinsic feature of spatial data; but, on the other hand, massive related spatial data for a domain is always stored in the same spatial database, so spatial data has another feature of centralization.

The distribution of spatial data brings along many inconveniences because a spatial task always uses various spatial data from different spatial databases; but, on the other hand, just because of the distribution, many GIS function agents can cooperate to complete the same spatial task concurrently in different hosts. This strategy can make the best of distributed computing resources, and computing can be performed in the host where spatial data is stored, so as to reduce the transferring quantity of spatial data in the network.

Because of the centralization, the data resource required for a problem may be in one host. A Data Priority strategy (an agent will try to execute in the host where the data resource is located) is adopted in Geo-Agents, so although GIS problems can be parallel, the parallel agents will execute place in the same host. Because the computing resources are limited in one host, and agents also exhaust some computing resources, this parallel will exhaust more computing resources than sequential execution by one agent. However, from another viewpoint, although an agent parallel in one host could not improve the performance, it provides a simplified and clear structure for design applications. For example, if there are two spatial metadata databases in one host, it is more convenient and clear to build two spatial metadata access agents to access different metadata databases.

According to traditional viewpoint, parallel should improve performance. However, parallel is a kind of collaboration, and improving performance is not an intrinsic feature of collaboration. Firstly, collaboration brings new thought for software construction. Collaboration enables software construction organized as human society; thus, complex software design can be built more easily and has a clearer architecture. Secondly, collaboration improves the capacity of software systems. For a software system, the capacity is primary. Collaboration makes some tasks achievable, which cannot be completed in traditional system. Only after a task can be completed correctly, improving performance is valid. Of course, aiming at the disadvantage brought by centralization of spatial data, a "peer hosts" mechanism is designed to partially improve the performance of Geo-Agents (see Section 5.2).

5 LOAD CONTROL

In Geo-Agents, load is mainly controlled when GIS agent is created. There are two control strategies: agent scheduling strategy and peer host strategy. Agent scheduling is a random load controlling strategy used to balance the load of the whole system. Agent scheduling is an intrinsic feature of Geo-Agents. During the running time of Geo-Agents, agent scheduling strategy automatically controls the network load and host load among Geo-Agents according to its all-around running information. Peer host is a load controlling strategy through man-made configuration. The goal of peer host strategy is to reduce host load of a certain host. When constructing a distributed GIS application using Geo-Agents, peer host strategy will be adopted for this host if the heavy host load of a certain host can be predicted.

5.1 Agent Scheduling Strategy

5.1.1 The Principles to Create an Agent

In Geo-Agents, GIS agent is designed as reusable and ordinary GIS agent. Reusable GIS agent means its result can be used again.

When creating a reusable GIS agent, if its type and task are same with an existing active reusable GIS agent (in Geo-Agents, there are some determinate reusable GIS agent types, and the task of each type is comparable [15]), it is unnecessary to create a new reusable GIS agent. Otherwise, the requirement to create a reusable GIS agent is the same when creating an ordinary GIS agent.

When creating an ordinary GIS agent, there are some different requirements because of considering the network load:

1. A GIS agent must be created in an appointed host.
2. A GIS agent must be created in local host by the Facilitator.
3. A GIS agent will be created in appropriate host (for example, the host where spatial data is stored).
4. A GIS agent can be created in any host.

In the first two cases, the host load they bring is inevitable, and there is no proper method to reduce or balance relative host load. But in the last two cases, it is necessary to control the creation of agent because of the randomicity, otherwise, there will have too many agents in one host.

During the creation of GIS agent, the network speed must be considered. In Geo-Agents, some assumptions are given: accessing an agent in the same Agent Region is faster than in an other Agent Region; accessing a data resource in the same Agent Region is faster than in another Agent Region; the priority of accessing a data resource is higher than accessing an agent. For example, supposing there are two Agent Regions, the data resource is in Agent Region1, and the user in Agent Region2 wants to create an agent to process the data resource in Agent Region1. At this situation, the agent should be created in Agent Region1.

According to the above requirements, the following principles must be obeyed when creating an agent:

1. If possible, an agent should be created in the host where data resource it processes is located.
2. If an agent needs to process several data resources, the agent should be created in one of the hosts where data resources are located, if possible.
3. If an agent couldn't be created in the host where data resource locates, it should be created in one of the Agent Regions where data resources locate if possible.
4. If possible, an agent should be created in the same Agent Region with its requestor.
5. In an Agent Region, an agent should be created in the host where its load is light.

5.1.2 Agent Scheduling Algorithm

Scheduling agents when creating agents is a powerful means to control host load and network load. The goals of scheduling agents are:

1. making the most of CPU of all hosts; in other words, eliminating the situation that some hosts are too busy and other are idle at the same time;
2. reducing the average response time of all tasks, and the network transmission as much as possible.

When designing an agent scheduling algorithm, three problems must be considered firstly [17]:

Determinate or heuristic algorithm: determinate algorithm is adopted only when the actions (such as computing requirements, data requirements, communication requirements and so on) of all agents are foreseeable; if the load of a system is unpredictable, a heuristic algorithm is needed. Heuristic algorithm means that the host allocation is instructed by rules from experience and by heuristic information during the working progress.

Distributed or centralized algorithm: centralized algorithm must gather all global information to give a decision, but distributed algorithm can give a decision by partial information only. Gathering all global information in centralized algorithm is rather demanding for the decision-making host, and would become the bottleneck of a system.

Optimized or hypo-optimized algorithm: centralized algorithm can obtain optimized result, but the cost is higher than with hypo-optimized result, because more information and complete processing are needed.

According to the agent creating principles and to the agent scheduling goals, a heuristic, distributed and hypo-optimized agent scheduling algorithm is developed:

> In Geo-Agents, agent scheduling is completed by the AgentServer. When creating an agent, a request is sent to AgentServer to ask it find an appropriate agent type. So, the AgentServer maintains a host table where the number of created agents in each host is recorded. If an appropriate agent type can be found in one host only, the agent should be created in that host, and the number of created agents in that host increments by 1; if an appropriate agent type can be found in several hosts, after considering the agent creating principles the AgentServer will select the host where the number of created agents is minimum to create the agent, and the number of created agents in that host increments by 1.

This scheduling algorithm is referred to as heuristic, distributed and hypo-optimized algorithm because:

1. it is not able to predict how many agents will be created in the system and how many agents are executing at the same time;
2. the scheduling criterion is the number of created agents, not the number of executing agents in one host; and
3. the scheduling criterion is obtained at the volley and it does not bring additional burden for the AgentServer.

5.2 Peer Host Strategy

When it is foreseeable that the host load in one host is too heavy, the peer host strategy is adopted to reduce the host load: connecting the host with other hosts via high speed LAN, which have the same software and hardware configuration and can complete the same tasks as the host. The host is referred to as peer main host, and the other hosts as peer secondary hosts. All main and secondary hosts form a peer host group (see Figure 9).

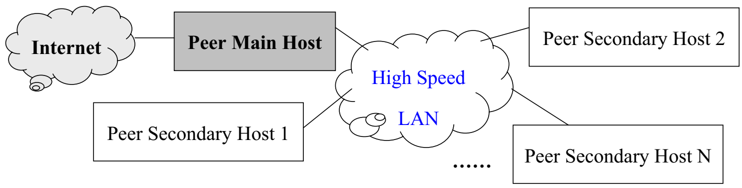

Fig. 9. Peer host group

When creating an agent in the main host, if the host load of the main host is too heavy, the agent can be created in any of the secondary hosts. Of course, in order to balance the host load of all peer hosts, the host where the host load is the lightest is selected.

In Geo-Agents, the host requesting of a peer host is that one where the GuServer is located. The GuServer maintains one or more spatial databases and spatial metadata databases, and is responsible for managing and accessing spatial data and spatial metadata. Spatial information is centralized in the host where the GuServer is located. According to the agent scheduling strategy, all agents which need the GuServer to manage and access spatial information will be created in the host where the GuServer is located. Thus, the host load of the host where the GuServer is located will be too heavy, and the performance of the system will be very low. Of course, enhancing the software and hardware configuration of the host where the GuServer is located can improve performance of the system, but cannot reduce the host load of that host. Peer host will play an important role here (see Figure 10).

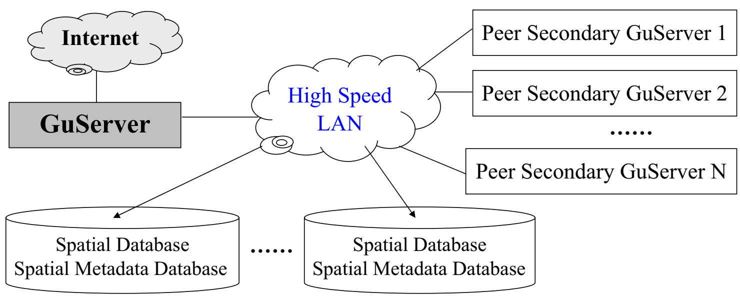

Fig. 10. The peer host group of the GuServer

Here, the host with installed GuServer is the main host, and the hosts with installed peer secondary GuServer are the secondary hosts. All GuServers manage and access the same spatial databases and spatial metadata databases. When creating agents which will use spatial information managed by the GuServer, they can be created in average in the peer host group of the GuServer.

6 A SAMPLE – OVERLAY ANALYSIS

Overlay analysis overlaps two or more layers into one layer following a particular rule. In traditional way, it is a time-consuming and heavy load task, because the operation needs to transmit all layers to local machine, and then overlap the layers one by one.

In Geo-Agents, two GIS functional agents are designed to complete overlay analysis: Overlay Manager Agent (OMA) and Overlay Agent (OA); Data Accessing Agent (DAA, see Section 3.2) is used here as well. OA just overlaps two layers, and DAA is used to search and get the needed layers from distributed spatial databases. Overlay analysis is described in a series of GeoScript statements, and OMA is responsible to interpret GeoScript statements, and then manipulate OA and DAA to complete the task. The whole procedure of overlay analysis is illustrated in Figure 11.

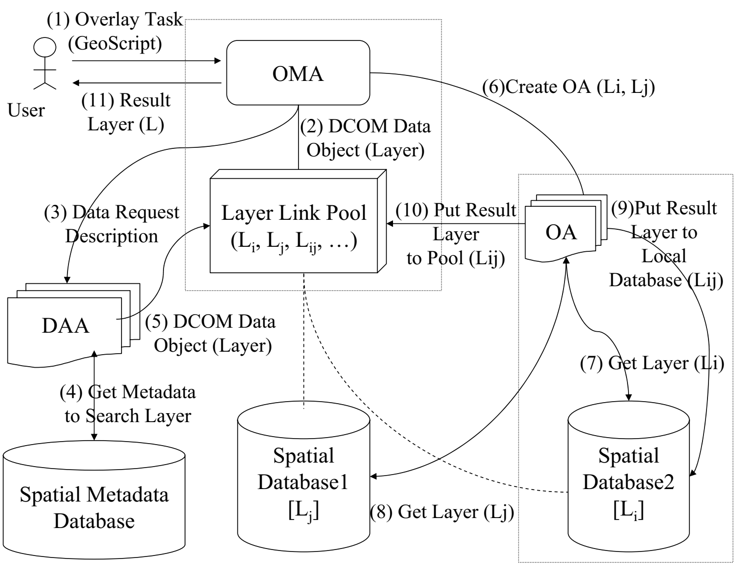

Fig. 11. Collaboration diagram of OMA, OA and DAA: overlay analysis

OMA maintains a layer link pool that contains all layers and temporary layers for overlay analysis. Every overlay layer may be a data object or a data request description. For a data object, it can be put into the layer link pool directly. For a data request description, OMA will create a DAA to search the request data, get the data object of the requested data and then put it into the layer link pool. OMA selects two overlay layers and creates an OA to overlap the two layers. The result layer (a temporary layer) of OA will be put into the layer link pool too. When only one layer is left in the layer link pool, and no active OA exists, the only layer left in the layer link pool is the final result of the overlay analysis.

Using an agent to complete overlay analysis can improve its performance considerably:

1. several DAAs can be created to search different layers concurrently (this is isomorphic cooperative parallel);
2. several OAs can be created to overlap different layers concurrently (this is isomorphic cooperative parallel too);
3. OA can run on the local site where the overlay layer is located; this will reduce the translation of layer in the network; and
4. DAA and OA can be parallel, it is unnecessary to wait for all overlay layers to be available (This is heterogeneous cooperative parallel).

7 CONCLUSIONS

Agent technology is used to construct a SIG framework, which contains three layers: Users/applications layer, agent services layer and information layer. Different applications can get their spatial information via agent services, and agent services make the procedure of navigating and accessing spatial information transparent to users.

By describing, organizing, managing, processing, exchanging, indexing and deploying data, information and knowledge in spatial information resources, and with the help of communication and network infrastructures, SIG joins different distributed spatial data capture and output systems, spatial data sets, large spatial data storage systems, GIS platforms and spatial application servers, organizes them as a virtual organization, and, furthermore, forms a infrastructure for spatial information in the Internet.

The construction of SIG will give a powerful technology support for users to share, access, analyze and process spatial information, and will provide a powerful management infrastructure for spatial applications, so as to guarantee that spatial information coming from any resources can be sent in a suitable way to any authorized users anywhere and anytime.

Acknowledgement

This work is supported by the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 20020001015; the National

Science Foundation of China under Grant No. 60203002; the National Grand Fundamental Research 973 Program of China under Grant No. 2002CB312000; the National High Technology Development 863 Program under Grant No. 2002AA135330 and No. 2002AA134030; the Beijing Science Foundation under Grant No. 4012007; the Doctoral Foundation of Xinjiang Bingtuan.

REFERENCES

[1] Aijun, Ch. et al.: Geospatial Information Layer Sharing Model For Digital Earth (in Chinese). Journal of Software, Vol. 13, 2002, No. 8, pp. 1436–1440.

[2] Open GIS Consortium. Availaible on: `http://www.opengeospatial.org/`.

[3] ISO TC211. Availaible on: `http://www.isotc211.org/`.

[4] Foster, I.—Kesselman, C.—Tuecke, S.: The Anatomy Of The Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications, Vol. 15, 2001, No. 3, pp. 200–222.

[5] Foster, I.—Kesselman, C.: Globus: A Metacomputing Infrastructure Toolkit. International Journal of Supercomputing Applications, Vol. 11, 1997, No. 2, pp. 115–128.

[6] Foster, I.—Kesselman, C.—Nick, J. M.—Tuecke, S.: The Physiology Of Gird: An Open Grid Service Architecture For Distributed System Integration. Available on: `http://www.globus.org/ogsa/`, 2002.

[7] The Earth System Grid. Availaible on: `http://www.earthsystemgrid.org/`.

[8] The Earth Observation System. Availaible on: `http://eospso.gsfc.nasa.gov/`.

[9] Xiao, N.—Fu, W.: SDPG: Spatial Data Processing Grid. Journal of Computer Science and Technology, Vol. 18, 2003, No. 4, pp. 523–531.

[10] Nwana, H. S.: Software Agent: An Overview. Knowledge Engineering Review, Vol. 11, 1996, No. 3, pp. 205–244.

[11] Wooldridge, M.—Jennings, N. R. —Kinny, D.: Methodology For Agent-Oriented Analysis And Design. Proceedings Of The 3rd International Conference on Autonomous Agents, Seattle, WA, USA, 1999, pp. 69–76.

[12] Tang, C.—Xu, L.—Feng, S.: An Agent-Based Geographical Information System. Knowledge Based Systems, Vol. 14, 2001, No. 5–6, pp. 233–242.

[13] Tsou, M.-H.—Barbara, P.—Buttenfield: An Agent-Based Communication Mechanism For Distributing Geographic Information Services On The Internet. GIScience 2000: First International Conference on Geographic Information Science, October 28–31, Savannah, Georgia.

[14] Yingwei, L. et al.: The Model Of Distributed GIS-Oriented Multi-Agent System (in Chinese). Acta Scientiarum Naturalium Universitatis Pekinensis, Vol. 38, 2002, No. 3, pp. 375–383.

[15] Yingwei, L.: The Study On Agent-Based Distributed GIS (in Chinese). [Ph. D. Dissertation], Peking University, Beijing, 1999.

[16] West, L.—Hess, T.: Metadata as a Knowledge Management Tool: Supporting Intelligent Agent and End User Access to Spatial Data. Decision Support Systems, Vol. 32, 2002, No. 3, pp. 247–264.

[17] Lin, Hwa—Chu—Raghavendra, C. S.: A Dynamic Load Balancing Policy With A Central Job Dispatch. IEEE Trans. on Software Engineering, Vol. 18, 1992, No. 2, pp. 148–157.

Yingwei Luo is now an Associate Professor in the Department of Computer Science and Technology, Peking University. His research interests include geographic information system, software agent and mobile computing.

Xiaolin Wang is now an Associate Professor at the Department of Computer Science and Technology, Peking University. His research interests include geographic information system and virtual server.

Zhuoqun Xu is now a Full Professor at the Department of Computer Science and Technology, Peking University, His research interests include geographic information system, artificial intelligence and parallel computing.