Mobile Agents Collaboration for Information Gathering

Chih-Lin Hu and Wen-Shyen E. Chen

Institute of Computer Science
National Chung-Hsing University
Taichung, Taiwan, R.O.C.
{clhu, echen}@cs.nchu.edu.tw

Abstract

The proliferation of the diverse information in the Internet makes the information gathering via a single mobile agent difficult. In addition, the limitations of the traditionally sequential task restrain the single agent to be applied to large, complicated applications. As a result, the applicable scope of a single mobile agent is limited. The advent of the collaborating multi-agents infrastructure not only alleviates this difficulty of gathering information throughout networks but also makes the ubiquitous Internet computing possible. In this paper, we introduce an infrastructure of the mobile agents collaboration for Information Gathering which provides the agent interoperability, asynchronous agent activity, and agent control mechanisms. We employ the persistent repository, which is vital to support the transparent recovery of failure and agent mobility. This infrastructure is simple, but able to deal with those complex tasks for the application of multi-agents for information gathering.

Keywords: Mobile Agent, Mobile Computing, Information Gathering, Distributed Information Management, Multi-agents System

1 Introduction

With the growth of the Internet and WWW, and the availability of low cost mobile devices, such as mobile PCs and PDAs, the mobile users are able to access information available on the fixed network anywhere, anytime. However, in the wireless network environments, the mobile devices face several limitations, such as low bandwidth, low computing power, small memory capacity, low battery lift, etc., restricting the mobile computing. The advent of the mobile agent technology [3–8, 11, 12, 20, 21] is expected to overcome these limitations. Mobile agents are specialized independent programs executing on behalf of users. They are transported to multiple remote hosts in the network to carry out assigned tasks. As they iterate, they perform work such as collecting information or delivering requests.

Using mobile agents to retrieve information faces several crucial challenges. First, the variety and amount of the data sources are increasing dramatically day by day. Second, in the Internet, the general information is unorganized, of diverse format, and may be distributed on several servers through heterogeneous networks all over the world. Third, the availability and reliability of information in services are changing constantly. Fourth, the information is likely to be erroneous for the dynamic nature of the information sources and that might incur more problems. Consequently, in large-scale network environments, it is becoming more and more difficult for a single mobile agent to gather or retrieve information efficiently and. In addition, many constraints imposed by the legacy networks [1], such as variable bandwidth and latency, unpredictable disconnection, unreliability, as well as the physical concerns with regard to the mobile devices, degrade the efficiency of a single mobile agent. The collaborating multi-agents approach [3, 15, 16], which provides methods for agent collaboration throughout networks, has been proposed to address the challenges.

Collaboration of multiple agents brings several advantages. First, the operation is distributed and asynchronous to enhance the availability. Second, the distributed operation diminishes the influence of the single point of failure as opposed to a single agent method. Third, the throughput is increased through parallelism. Finally, collaboration between multiple mobile agents can improve an agent’s capacity to extend the application scope in Mobile Computing. As a result, they provide needed flexibility and efficiency for information gathering.

The rest of this paper is organized as follows. Section 2 provides an overview of the current agent technologies. Section 3 describes the motivations and consideration of designing this infrastructure. Section 4 introduces the collaborating multi-agents infrastructure for the information gathering. Section 5 describes the mobility and the communication among collaborating multi-agents as well as introducing a scenario illustrating the infrastructure. Finally, concluding remarks are given in Section 6.
2 Current Agent Technologies

Nowadays, agent technologies are popularly proliferated worldwide and lots of important and innovative works are accounted to overcome many barriers in effective mobile computing and collaborative computational systems. There are three common agent technologies in the literature: intelligent software, mobile agent, and multiple agents [3-8, 11, 12, 20, 21]. The overview of the approaches follows.

2.1 Intelligent agent in design systems

Intelligent Agents with built-in intelligence are programs that act on behalf of their human users in order to perform laborious information gathering tasks. Some examples of such tasks are locating and accessing information from various on-line information sources, resolve inconsistencies in the retrieved information, filter away irrelevant or unwanted information, integrate information from heterogeneous information sources, and adapt overcome to their human users' information needs [2, 15].

2.2 Mobile agent in computing

Mobile agents [3-8, 11, 12, 21] are specialized as independent programs executing on behalf of network users. They are transported to multiple remote hosts in the network to process the information. As they itinerate, they perform work such as collecting information or delivering requests. There are obvious advantages to this when the amount of data that would have to be transported eclipses the amount of code required to perform the reasoning or when the amount of processing time available at the local host is limited. In additions, the agent mobility enhances the throughput of every computing element in the network and creates a powerful computing environment. Agent mobility will considerably influence the future of computing, even though several problems are still unresolved and a set of issues, such as security at the remote host, sufficient computational efficiency etc., need to be addressed.

2.3 Collaborating agents systems

Most of the current research and developments in collaborating agents [13, 16] are more relevant to design integrating heterogeneous, semiautonomous knowledge-based software components into coordinated applications with regard to the design process. Although these application tools are useful, they are still not tools. To integrate these application tools, it needs developing interoperability standards and functions so that information can be exchanged and shared throughout applications across heterogeneous programming languages, machines, and computing platforms.

3 Motivation and Considerations

The traditional distributed computing via client/server model has confronted several challenges such as problematic legacy network, scaling and protocol problems. Although the mobile agent technique is suitable for mobile computing environments, most current proposed frameworks are implemented by using a single mobile agent. As mentioned in Section 1, applying only a mobile agent seems difficult to implement large and complicate applications, especially in large-scale network environments. In the following, we account the main motivations and considerations in constructing this infrastructure for mobile agents collaboration.

Sequential task routine of a single mobile agent:
The mobile agent routine is traditionally sequential: a mobile agent is transferred sequentially from one location to another to collect information or deliver requests. The sequential routine is time-consuming and inflexible, requires a mobile agent to carry a great deal of payload, if two or more agent servers have information flow among them. In addition, a mobile agent may encounter single point of failure if one agent server of routine fails or becomes unavailable.

Distributed information sources
Information sources are inherently distributed across the publicly accessible networks, heterogeneous platforms and of diverse data formats. Instead of overloading a single mobile agent, it's better to accept the distributed framework of multiple agents specializing for diverse information sources.

Complexity hiding
The network user with mobile devices may disconnect after launching the mobile agents. It is necessary for a framework to hide the complexity of retrieving the distributed information in order to shield the network users from overloading.

Quality of information
Network user can appropriately compartmentalize the main task as several sub-tasks and each sub-task is delegated to one or multiple agents to perform it. If one agent is terminated prematurely, the others can continue to accomplish the remaining sub-task. The agents responsible for the same sub-task can share the piece of information to interact or negotiate to find the better data.

Information flow
The agents share data or process information with each other when one agent collaborates on the work with another agent. To facilitate the interoperability, it requires several levels of mechanisms, such as knowledge representation and communication language to embody the agent.

Where to share the information?
After launching multiple agents to do a complicate
task, these agents may share each resulted data to improve or refine the task. It is infeasible for these agents to deliver their data back to network user and forward again, or to directly communicate with others. The former is the network user may have disconnected and the latter is to avoid overloading agents and complexity of agent structure. That is, setting an intermediate known by agents is a requisite to facilitate the interoperability.

Where does the control mechanism lie?
The network users with mobile devices require a mechanism to strategically monitor, capture or coordinate the process of the task after they disconnect from the network. In this case, should the control mechanism lie in human user’s supervisor agents of another type, or component of the framework?

Transparency & reliability
At the unpredictable network, mobile agents may be terminated prematurely or fall out of the users’ control for server failures, network disconnection etc. A repository is needed to restore the agents and to continue the collaborating task.

Broker
Because services and resources are distributed throughout the world, it is convenient for user agents to locate them through brokers. It serves as a “bulletin board” where the service and resource agents advertise what they provide. The agents can request the broker to provide the service information.

4 Overview of the Collaborating Multi-Agents Infrastructure
In this paper, we account our multi-agents guideline that is substantially different from the Intelligent Software Agents (e.g., [2, 13, 14]) which rely on preplanned, domain-specific coordination in order to behave like on behalf of human being to perform complex task-based or decision support. We expect to develop an infrastructure that is more suitable for the generic mobile agents collaboration as well as extending the applicable range. Essentially, collaborating multi-agents infrastructure includes six fundamental components: Mobile Agent Server, Info Manager, Persistency Repository, Broker, Service/Resource Server and Agents Administrator, as shown in Figure 1. A brief description of the components follows.

4.1 Mobile Supporting Server
Mobile supporting server compensates the limited capacity of the mobile devices. It provides an interface as the entry point of the mobile network users to the fixed network. Mobile supporting server accepts the request from the mobile user and delegates the request to mobile agents. In addition, it provides extra functions to the mobile users according to their demands. Mobile users subscribe their application via filling a form (in HTML format) to mobile supporting server. Accordingly, mobile supporting server parses the subscribed form, generates an agent and sends it to other destinations. Mobile supporting server provides mobile network users the abstraction apart form the complex infrastructure to contact the public and large-scale Internet. It receives the user agents and reifies the requests as mobile agents to execute their tasks. As the mobile agents itinerate back with the resulted information, mobile supporting server has to pre-processes, organize information into a form suitable for the users and then returns to mobile users. Furthermore, it may also serves as a filter leaching the irrelevant, unwanted information or diminishing amount of information if the mobile user asks that.

![Figure 1 A Collaborating Multi-Agents Infrastructure](image-url)
4.2 Info Manager

Info Manager is the avenue to make the collaboration among multi-agents successful. It provides an interoperable intermediate, meeting pointing, where mobile agents in the same task can gather and share information to refine their goals and allows mobile agents to contact each other in an asynchronous fashion. Once the original mobile agent is launched from mobile supporting server, it may spawn a batch of mobile agents to cooperate their assignment according to the mobile user’s will. By firstly sending register of the batch to Info Manager, Info Manager allows this batch of mobile agents to apply a privilege InfoGroup where mobile agents in the same task will participate and partake the data.

Info Manager also provides the facility for the mobile agents to accumulate information or discover interesting information in the large-scale, public network environments, such as Internet. Mobile agents can register their interested information type and location reference in Info Manager and then the Info Manager allows mobile agents to participate the interested InfoGroup to retrieve and share the public data that the group members contribute. Another issue to study is the security [4, 22]. Although the infrastructure in its current form does not enforce security, we believe that the Info Manager will assist in authenticating and validating mobile agents and mobile users.

4.3Persistency Repository

The mobile agents roaming throughout the unpredictable network, such as Internet, are vulnerable. To remedy the susceptibility, the independent Persistency Repository is a prerequisite facility for robustly maintaining communications among agents. Mobile supporting server releases mobile agents to network and sends the relative referred information to Info Manager. Info Manager then requests the Persistency Repositories to keep all of mobile agents’ registration references, the collaborating data, internal state as well as the private InfoGroup membership in the same task. Hence, the collaborating information can be recovered automatically if any failure occurs and the Info Managers are capable of serving agents continuously while these mobile agents are unaware of the failures.

Regardless of the crash of Info Managers themselves, the Info proxies shield the mobile agents from this mishap before the Info Managers recover. For instance, one collaborating agent of private InfoGroup may be delegated to accomplish a significant sub-task. If this agent is terminated prematurely, other sub-task agents may not continue their assignments without referring to the data which the prematurely terminated agent would contribute. The Info Manager solves this circumstance by reloading the prior cached information reference from Persistency Repository and then spawning the agent again. Consequently, Persistency Repository assists Info Manager to recover agents, maintain agent membership, and store collaborating data. In additions, similar to support the Info Manager, the Persistency Repository supplies the Agent administrator to hoard the relative information into independent database.

4.4 Agents Administrator

Since the agent is mobile and autonomous, after agents were released, the system must have mechanisms to control them. In our infrastructure, the fundamental component, Agent Administrator, is responsible for providing controls over agents. The Agents Administrator have to monitor actively the progress of mobile agents or passively track the agents itinerating throughout networks, requested by users [3, 17]. In order to capture recent state of the mobile agent, mobile agent may be asked to periodically report its location and state for administration purpose. The detail of the agent location methods is beyond the scope of this paper and the reader is referred to [23, 24].

In a collaborating infrastructure, the administrative mechanisms are more sophisticated and tricky even though the executing operations (such as terminate, suspend, resume, etc.) are similar to those in a single mobile agent system. An operation on a mobile agent may influence the other processing mobile agents if they have collaborating relationship and are in the same InfoGroup. The Agent Administrator has no idea about the content of mobile agents and how they are to collaborate although it had received the agent reference from the Mobile Supporting Server as the they were launched. Agent Administrator can solve this kind of problems by cooperating with the Info Manager. Overall, the Agent Administrator provides the remote administration through a central location.

4.5 Broker

Broker facilitates to assist in locating the services/resources for the mobile users or mobile agents and allows the services/resources providers to advertise for network users. As mentioned in section 1, the variety and amount of data sources are increasing and the dynamic nature of the information services and resources are changing constantly. If a services/resources server can support mobile agents or its services/resources are changed, it registers what they providing and where it locating to the broker. The agent firstly interacts with broker to receive the up-to-date location information of the service/resources where the task can be fulfilled before starting it or travel to collect information or deliver requests, so that the agent is able to avoid the unavailable serves and continue the task.

The broker can migrate the request of mobile agent to another broker if it cannot satisfy the request, or.
periodically exchange the local information of registered services/resources with another to supply better quality of services. At the same time, the broker can maybe bear the responsibility of agent administrator to manage the services/resources servers supporting the mobile agents.

4.6 Service/Resource Server

This is the location where the services and resources are provides for the user. The agent server is prerequisite for the services/resources server and is served as the point of contact on the services/resources server. The service/resource server is wrapped in the agent server which is to be an interface hiding the complexity in order for the agent programmers to develop agents without clearly knowing the network. The agent server also has some mechanism to handle the heterogeneous database for mobile agents to information supply. When the mobile agent arrives in the Service/Resource server, the contacts firstly with the agent server. The agent server receives the requests from mobile agent, executes the embedded assignments belonging to him, embeds the executed results into mobile agent, and then transmits the mobile agent to his next destination again.

5 Implementation of the Collaborating Multi-Agents Infrastructure

We have presented in the previous section the components of collaborating multi-agents infrastructure for information gathering. In this section, we describe our implementation of the prototype infrastructure. We briefly describe the content of mobile agent and then account the platform for the agent execution.

5.1 Collaborating Mobile Agent

The encapsulation of the collaborating mobile agent in the ATP format is as shown in Figure 2. The collaborating mobile agent in our infrastructure not only consists of code, itinerary, intermediate results, and related arguments (as our previously proposed infrastructure [12, 19]) but also add a new field, registered location, which recorded the location of the registered Info Manager. Note that in the original ATP specification, the dispatched message format only specifies a vector for carrying agent-related information. We further refine the data structure to include UserID, Arguments, AgentID, Source, Itinerary, as well as Registered Location and GroupID. The UserID and AgentID keep the unique IDs of the user and the collaborating mobile agent, respectively. The Source filed is to record the address of the server where is the agent originally created (to send back the final results). The Itinerary filed keeps the location information of the Agent Servers that the collaborating mobile agent should visit to fulfill the assigned task. In particular, the Registered Location filed stores the registered Info Manager location and the GroupID keeps the InfoGroup ID in Info Manager respectively on behalf of the agent collaboration. So, as those launched agents need to collaborate with each other to accomplish their tasks, they can contact others via the meeting point on the registered location (The registered Info Manager have allocated a private InfoGroup for this batch of collaborating mobile agents as mentioned in Section 4.2).

Figure 2. Encapsulate a Mobile Agent of the Extended ATP

5.2 Architecture of Collaborating Multi-Agents Platform

It seems to be impossible for a mobile agent to directly contact the service/resource servers to retrieve the information: a mobile agent cannot foreknow what kinds of services or which data format the service/resource servers providing. On the other hand, the service/resource servers itself don’t include the functions to support the mobile computing. We need a agent server to support a execution platform that shields the mobile agents from the complexity of directly contacting the service/resource servers and also facilitates the tasks on behalf of mobile agents. In the following, we introduce the architecture of agent platform and the illustration is given as Figure 3. We use the Extended ATP (Agent Transfer Protocol) [12, 19], which is modified from the IBM ATP [9] by adding the control actions, as the communication protocol between agent servers.

Agent Context

Agent context serves as the interface between the mobile agent and agent server. Agent server prohibits the mobile agents directly accessing the service/resource systems without through the agent context. All activities of mobile agents have to be processed on the agent context and through this context to access.
Agent Security Manager
The agent security manager on the top of the agent server defends the service/resource server from the malicious mobile agents. Agent security manager will firstly mediate the operations on the service/resource systems from mobile agents if legal to avoid damage. Furthermore, it will check the information mobile agent access even though the mobile agent is not malicious.

ATP Handler
ATP handler fetches the agent from the network and then distinguishes if this agent belonging to an ATP agent. Likewise, the ATP handler wraps the outgoing data as an ATP agent and then transfers it to the next destination.

ATP Parser
According to the ATP communication protocol, ATP parser parses the agent content from ATP handler, gets the arguments etc. and then instantiates this agent.

ATP Loader
ATP loader generates the agent context and arguments according to the instantiated agent from ATP parser. Then, ATP loader loads the agents to be executed on agent context.

ATP Creator
According to the ATP communication protocol, ATP creator generates the ATP agent including ATP agent header, agent content and arguments, etc.

Figure 3. Architecture of Agent Platform

Agent servers provide the executable environment and resources to assist the mobile agents in fulfilling their assigned tasks. ATP handler firstly distinguishes if the coming agent belongs to an ATP agent when a agent from network comes in. If true, the ATP parser in turn parses this ATP agent content and accordingly has three different implementations as the Figure 4 depicting. First, if the content of this ATP agent consists of mobile agent class, the ATP parser interprets the content and arguments, then calls the ATP loader to instantiate these classes and executes this agent. After finishing this execution, the ATP creator packs the resulted data and related class as a ATP agent and then ATP handler transfers it to the next agent server if the agent wants. Second, if the coming agent is a control agent, the ATP parser interprets the operations (response, retract, resume, etc.) and then executes the assignments of control agent. Finally, the agent server may be also served as a mobile supporting server so that it can receive the user agent from network user asking to launch mobile agents. ATP parser interprets the content and arguments of the user agent; ATP creator accordingly generates this agent in turn; ATP handler launches it to the destination.

In the agent server architecture, the ATP loader can not only loads the mobile agent class but also can load directly service agent class from the file system. Like Java, the Agent loader is de facto a class loader: we utilize the java.lang.ClassLoader class in Java to construct this Agent loader. Agent itself is a "executable class": after loading and reifying, it can be executed. This facility is a key point in mobile agent system.

6 Agent Communication for Information Gathering
In traditional modus, a single agent launched for information gathering task travels throughout networks together with the data and code. During the mobile agent traveling, the scheduler routine is sequential that says it is transmitted to the next agent server together with the data obtained from last agent servers. The payload of single agent may increase gradually as migrating from one resource server to another; therefore, the ponderous body postpones the single agent migrating, reduces the agent mobility under mobile network. Unfortunately, the huge amount of unorganized information and disparate format information lead the single mobile agent into a predicament. Likewise, the sequential design is brittle and inflexible and often requires numerous iterations through what can be an expensive and time-consuming set of steps [16] Contrarily, using collaborating multi-agents to retrieve or gather information can alleviate this embarrassment. Mobile agent server possesses the ability to divide information gathering task into several sub-task and can leasing collaborating multi-agents instead of leasing only one single agent according to user's demand as mentioned in section 4.1.

6.1 Agents Traits for Information Gathering
A mobile agent is of mobility across networks with embedded data and code while released. Once a mobile agent is launched, it should be persist, not reliant on the mobile supporting server launching it.
and also be autonomous with the instructive information describing the delegated task, not intervened by the mobile user. That says, during the task, the mobile agent is self-control and even to be adaptive, like with "intelligence", according information obtained from the network. This is our guideline of designing this infrastructure enabling mobile agents to be collaborative. Besides, these characteristics are indispensable for mobile agents to be practical and are convenient for network users to obtain distributed information and services to fulfill their tasks.

Historically, a mobile agent is viewed as an independent object of idiosyncrasies and should perform well in certain application domains. However, as discussed before, applying a single mobile agent technology in large-scale network applications seems to be insufficient and ineffective. The infrastructure we proposed in this paper offers the equipment to enable mobile agents to collaborate together information gathering, that is, concretely mobile agents can share or exchange information via the Info Manager, described in Section 4.2, as an "meeting point" to achieve the interoperability.

6.2 How agents obtain information

The variety and amount of data sources are increasing and the dynamic nature of the information services and resources are changing constantly. In order to assist in locating services/resources servers, user agents first contact with the Broker to obtain recent advertisements. Then, network users fill in the pre-formatted forms to describe the agent travel and send these forms to mobile supporting server.

6.2.1 local information

Most mobile agent systems use this mode of obtaining information. A single agent is launched for information gathering task and travels throughout networks with the data and code. During the mobile agent itinerary, the mobile agent sequentially visits each services/resources server, contacting agent servers to retrieve their local information.

6.2.2 global information

The variety and amount of the information sources are increasing dramatically day by day and at the same time the availability and reliability of information services are changing constantly. In this infrastructure, we allow mobile agents to retrieve global information with more flexibility via two approaches, interested and grouped information:

* Interested information:
Mobile agent registers the Info Manager with the interesting types and the location address where it wishes its interesting data to be sent. Info Manager filters all kinds of publicly and accessibly passing data and then forward it to the proper registered location addresses.

* Grouped information:
Similar to the registration of retrieving interested data, the agent can participate certain InfoGroups to share unfiltered information which members in this InfoGroup contribute. InfoGroup forwards information received from its members to other members. That is to say, the agents can join the special InfoGroups if they have the common interest in some certain topic.

6.2.3 collaborating information

A mobile agent server divides the large or complicated information gathering task into several sub-tasks according to mobile user's will and the delegates sub-task to collaborating mobile agents as well as sends registration references of collaborating mobile agents to Info Manager. The Info Manager enables the agent server (including the Mobile Supporting Server) to do complicated information gathering and to apply privilege InfoGroups where the batch of collaborating mobile agents in the same task will participate and partake the data on behalf of agents.

6.3 Collaborating Mobile Agents
Conventionally, a mobile agent is launched to solely fulfill an assigned task, but lots of legacy constraints and problems decrease the mobile computing power. We introduce the collaborating mobile agents to address these issues.

6.3.1 Constructing a Collaborating Mobile Agent

In our prototype implementation, the mobile supporting server provides the network users a familiar interface, Java-enable WWW browser, to enter the fixed network. The mobile user first connects to the WWW server that resides in a Mobile Supporting Server via the HTTP protocol and then requests a form (in HTML format) of services. The WWW server returns the mobile user a Java applet to fill in the instructive parameters and related information that are needed for creating a mobile agent and specifying the agent task. The WWW browser then transforms this form into an ATP message and submits it to the mobile supporting server. As mentioned in Section 5.2, the mobile supporting server reifies the ATP message to be mobile agents. In addition, the Mobile Supporting Server sends the registration and relationship of mobile agents to Info Manager so that the mobile agents are enabled to collaborate with each other through the meeting point in Info Manager as they executing assigned tasks.

6.3.2 Interaction between Collaborating Mobile Agents

We now suppose that content of user agent asks mobile supporting server to spawn a batch of mobile agents to collaborate with. Mobile supporting server accordingly comparts the task into several sub-tasks, allocates these sub-tasks to this batch of mobile agents and then launched them. At the same time, it also sends the registration of each mobile agent and relationship to Info Manager and asks the Info Manager to create a private InfoGroup which serves as the meeting point of the batch.

Each launched collaborating mobile agent includes a registered location filed which indicates where the registered Info Manager locates, as shown in Fig. 2. The agent servers can send ATP messages to registered Info Manager to exchange or share the data contributed by other collaborating mobile agents. With the GroupID of the agent, the Info Manager could look up the InfoGroup to find out the corresponding data and then replicate or modify the data. The Info Manager may send back the inquired data to the agent to continue its assigned task. Fig. 5 illustrates the cooperative work of mobile agents.

Mobile agents can collaborate with each other through the Info Manager, relaxing the embarrassment of a single mobile agent. Furthermore, the Info Manager may include a Info proxy can provide transparency and reliability to mobile agents. The Info proxy guards the mobile agents from directly retrieving data during the effects of failures of InfoGroup or Info Manager. With the Persistency Repository storing all the prior registration information, as well as the administrative reference from Agent Administration, the recovery of InfoGroup or Info Manager or even Agent Administration becomes possible.

6.4 Scenario: Cooperative information gathering

The multi-agents technology has been applied to a lot of applications but most of them are specialized for certain purposes as well as built-in §st dligence of knowledge”, such as decision making, industrial procedure analysis, etc. The following is a scenario (depicted in Fig. 6) for the application of agents.

Charles, studying in Taichung, has a rendezvous with Michelle in Taipei. He has no idea how long it takes and how much it costs from Taichung to Taipei, but he has a mobile computer.

He fills in a predetermined-format form.

In this form, Charles specifies what kinds of transportation (airplane, train, or bus), starting city (Taichung) and destination city (Taipei), the
maximum amount of money willing to pay, the upper bound of travel time.

and then delivers this form to the mobile supporting server.

Mobile supporting server receives the form from the user; according to the content of the form, the mobile supporting server requests the broker to provide the services/resources location references of airline, train and bus transport companies.

After obtaining the references, the Mobile Supporting Server launches three batches of collaborating agents and each batch is obligated to distinct transport.

At the same time, the Mobile Supporting Server registers the three batches of collaborating agents in the Info Manager and each batch of agents joins the distinct private InfoGroup.

Each collaborating agent in the same batch sends sharing data (pay, transport time) from different company to InfoGroup, and then Info Manager forwards it to other agents. If the received data is better, the collaborating agent is terminated. After all batches accomplished their assignment, the optimal option is derived.

Charles finally receives the optimal choice and goes to Taipei happily.

7 Conclusion

Mobile agent technology is affecting the world of network computing and the multi-agents technology is well suited for the network applications. We expect the agents to be mobile and be able to do collaboration, integrating these two technologies, to facilitate network user in retrieving information. The advantages of collaboration between mobile agents include enhancing throughput, improving quality of information, flexibility, reliability, etc..

In this paper we have proposed an infrastructure that supports support mobile agents to access information or services available in a large-scale network environment, across heterogeneous networks, platforms, and protocols as well as data represented in diverse data formats. The agent server shields mobile agents from these complexities. The design of Info Manager and the facility of Persistency provide the transparency of collaborating mobile agents communication, agent persistency, and reliable transmission on behalf of network users. The Broker hides the network users from the dynamic nature of services/resources servers and allows them to capture the service information updates.

The security issues are not addressed in the paper. In addition, the structure of collaborating agents does not carry much §§intelligence§§ per se. We are currently working on the issues.

References


[5] ¡ObjectSpace Voyager Core Package Technical
Overview \(\text{\textregistered}\) Version 1.0; "December 199


