Maintaining Persistent Look-and-Feel for Roaming User with Mobile Agent in Distributed Environment

SHIH Kuo-chen(Timothy)[1], DENG Yu-kuang(Lawrence)[1], HUANG Teh-sheng[1], MA Jian-hua[2], HUANG Run-he[2]

[1](*Department of Computer Science and Information Engineering, Tamkang University, Taibei, China*);

[2](*Faculty of Computer and Information Science, Hosei University, Tokyo* 184-8485, *Japan*)

E-mail: tshih@cs.tku.edu.tw

http://www.mine.tku.edu.tw

Received June 20, 2001; accepted December 13, 2001

Abstract: Mobile agents refer to self-contained and identifiable computer programs that can migrate along the network and can act on behalf of the user or other agents. Mobile agents often work on heterogeneous network and operating system environment. Therefore, an integrated logical interface to access physical structure via mobile agent application has become more and more urgent. In this paper, we proposed a tracking and persistent agent-based mobility management system in case of distance learning. The main purpose of our system is addressed to achieve the universal access objective. In order to let the whole mobility management system full play, firstly, we proposed a mobile agent communication network model and the mobile agent evolution states. The mobile agent communication network model facilitates the agent computation, and the mobile agent evolution states assist agent supervisor to monitor client agents. Secondly, we encapsulate the utility tools to be a role-setting object. The role-setting object is an application-driven component, which can provide customization benefits for user to match user's demands. These integrated technologies are sufficient in distance learning (virtual university) environment.

Key words: mobile agent; agent communication network; agent evolution states; persistence look-and-feel; client agent; server agent

Autonomous agents and multi-agent technology play a control role in network management and systems management communities for several years. Desktop agent includes the operating system agent (e.g., MacOS Open Sesame, Windows 9x) and the application agent (e.g., Micrsoft Office applications Wizard)[3]. On the MacOS, Open sesame agent could find repetitive user patterns and provide to automate the customary tasks for the user. On Window 9x, the system wizard executes in the background and carry out the programs through the task scheduler browser; the maintenance wizard automatically organizes the files on your hard disk for efficiency, fixes disk and file errors and frees up disk space; the Add New Hardware Wizard resolves the device problems. In Microsoft Windows desktop application agent such as Office wizard, the web-publishing wizard make it easier than ever to create an office documentation and to transfer files and publish Web pages to a remote server. With the growing of network, likewise, the application-driven agents also provide many specialized facilities such as information retrieval agent, Mobile agent, Process automation agent, collaborative customization and database agent.

In this paper, we devoted our attention to the mobile agent issues especially in universal access objective. Since, mobile agents often work on heterogeneous network and operating system environment. An integrated logical interface to access physical structure via mobile agent application has become more and more important. For flexibility, mobile agents can be accepted as a design paradigm like object-oriented programming or client/server computing. Based on these wherefores, we proposed a tracking and persistent agent-based mobility management system in case of distance learning to illustrate the entire processes.

© 中国科学院软件研究所 http://www.jos.org.cn

Students of the virtual university roam from station to station. It is important to provide a persistent environment so that students will always obtain their personal profile. The solution of such a roaming service involves mobile agent technique. A mobile agent can travel from station to station, with its execution status attached. To implement such a mobile agent involves station privilege control. Usually, a mobile agent platform needs to be installed in each student workstation. When the mobile agent travels to the station, the agent platform accepts it and invokes a child process to run the mobile agent. The mobile agent will retrieve personal information of the student from a mobile profile. The profile will contain the information such as personal notebook, learning status, and the personal look-and-feel setup. A student notebook tool allows the student to cut and paste Web course content objects into a personal notebook. The objects include test paragraphs, pictures, animations, audio clips, and possibly video records. The learning status of each course taken by the student will be recorded. The student should be able to continue from a previously visited point in each Web course. In addition, each student can setup some look-and-feel personal data, which includes the resolution and name of Web browser (IE or Netscape), generic look-and-feel setup, personal communication list, etc.

The mobile person agent serves as a front end of virtual university access. Students will talk to this agent anytime anywhere. Similarly, instructors will have another agent. Agents will communicate with each other. For instance, a student can look at other agents, which represent their owners in an on-line course. Via clicking on an agent, the student can talk to each other.

A mobile agent, in general, can be more than just a search program. For instance, a mobile agent can serve as an emergency message broadcaster, an advertising agent, or a survey questionnaire collector. A mobile agent should have the following properties:

- Automation: It can achieve a goal automatically.
- Delegation: It should be able to perform a set of tasks on behalf of a user(or other agents)
- Communication: It should be able to communicate with other agents.
- Actuation: It should be able to affect its environment via an actuation mechanism for autonomous operation.
- It has evolution states, including a termination state.

We propose a logical network for agent connections/communications called Agent Communication Network (or ACN). ACN is dynamic. It involves as agent communication proceeds. It also serves as a graph theoretical model if agent evolution computing. Our research purposes include:

- Provide a model for agent evolution and management facility.
- Construct simulation facilities to estimate agent evolution.
- Suggest guidelines to construct mobile agent environment.
- Suggest strategies to construct Characteristic-Oriented (application-driven or Role & Function) Mobile Agent, and build fundamental security (Delegation and Authentication).
- Tracking and Persistence Agent-based Mobility Management in Mobile Networks Web Agent Systems.

Given an ACN, the model finds which agent evolution policy produces the maximum throughput (i.e., the goal of agents achieved). Or, changing the structure of an ACN, the model is able to find out how to adjust the agent evolution policy in order to recover form the changes (or how is the throughput affected).

The remaining parts of this paper were organized as follows: The related works are discussed in Section 1. The mobile agent communication network and an agent evolution based on a state transition diagram are given in Section 2. A mobile agent system architecture model describing server agent and client agent components is illustrated in Section 3. Implementation (case in distance learning and called Multimedia Macro Virtual System) is then discussed in Section 4. And finally, we discuss the conclusion and possible extensions in Section 5.

© 中国科学院软件研究所 http://www.jos.org.cn

1 Related Work

The concept of agent-based software engineering is discussed in a survey paper[2]. The author presents two important issues: agent communication language and agent architecture. Agent communication languages allow agents to share information and send messages to each other. Agent architecture, on the other hand, includes network infrastructure and software architecture that ensures agent computing. An open agent architecture for kiosk-based multimedia information service is proposed in Ref.[1].

Survey of different types of agents can be found in Refs.[3,4]. Web-based intelligent agents are necessary and should be a trend of new engineering applications. However, application designers need to resolve the conflict between the client-server-based designs and the peer-to-peer protocol. The Java Agent Template (JAT) is proposed, which allows application designers to write Java programs runnable on heterogeneous computer environments. Agent programs developed in this technique can send KQML messages to each other. In our simulation, we use JATLite, which is a version of JAT, on Windows NT based environment. In Ref.[4], the author summarized three types of agents: intelligent autonomous agents, mobile agent, and cooperative agents. Agent applications and development are also discussed. A personalized agent system, BASAR (Building Agents Supporting Adaptive Retrieval), maintaining Web links based on personal bookmarks is proposed in Ref.[5]. The system is able to support information updating, providing that the content of Web site where the user marked is changed. The system also reduces the number of links by deleting seldom-used ones. And, new links can be added by using search engines.

A Web-based information agent is proposed in Ref.[6]. Using KQML as the communication language, the proposed system significantly reduces network load by pushing the computation to the server Structured Meta information is also used to decrease the complexity of browsing. A multi-agent system supporting intelligent information over the Internet is proposed in Ref.[7]. The system incorporates technologies from intelligent software agents, natural language understanding, and conceptual search mechanisms to access Earth and Space Science Data.

The concept of mobile agent is discussed in several articles[8~11]. Agent Tcl, a mobile-agent system providing navigation and communication services, security mechanisms, and debugging and tracking tools, is proposed in Refs.[12~14]. The system allows agent programs to move transparently between computers. A software technology called Telescript, with safety and security features, is discussed in Ref.[15]. The mobile agent architecture, MAGNA, and its platform are presented in Ref.[10]. Another agent infrastructure is implemented to support mobile agents[11]. A mobile agent technique to achieve load balancing in telecommunications networks is proposed in Ref.[16]. The mobile agent programs discussed can travel among network nodes to suggest routes for better communications. Mobile service agent techniques and the corresponding architectural principles as well as requires of a distributed agent environment are discussed in Ref.[17]. The evaluation of several commercial Java mobile agents is given in Ref.[18]. The survey above shows the glimmer of agent functionality that can be built into a service application. Anyway, designing a mobile agent system is a technical challenge. In order to make the mobile agent service application to be accomplished, we built the agent communication network first. Then, we used this logical network framework model to construct a tracking and persistence agent-based system.

2 Mobile Agent Society

2.1 Agent communication network

Agents communicate with each other since they can help each other. For instance, agents sharing the same search query should be able to pass query results to each other so that redundant computation can be avoided. An Agent Communication Network (ACN) serves this purpose. Each node in an ACN (shown in Fig.1) represents an

© 中国科学院软件研究所 http://www.jos.org.cn

agent on a computer network node, and each link represents a logical computer network connection (or an agent communication link). Since agents of the same goal want to pass results to each other, they are modeled as a complete graph. Therefore, an ACN of agents holding different goals is a graph of complete graphs. In Fig.1, there are six types of agents (i.e., species **A** to **F**). Since agents can have multiple goals (e.g., searching based on multiple criteria), an agent may belong to different complete graphs. For instance, agents **a** and **b** both belongs to the complete graphs of species **A** and **B**. On the other hand, agents may have a unique goal (e.g. agents of species **E** or **F**). Agent e of species F is the only one of its kind.

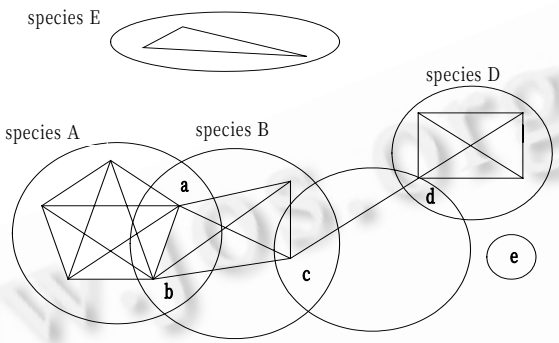

Fig.1 An agent communication network

We define some terminology, which were used in this paper. A host station (or station) is a networked workstation on which agents live. A query station is a station where a user releases a query for achieving a set of goals. A station can hold multiple agents. Similarly, an agent can pursue multiple goals. An agent society (or society) is a set of agents fully connected by a complete graph, with a common goal associated with each agent in the society. A goal belonging to different agents may have different priorities. An agent society with a common goal of the same priority is called a species. Since an agent may have multiple goals, it is possible that two or more societies (or species) have intersections. A communication cut set is a set of agents belonging to two distinct agent societies, which share common agents (e.g., {**a,b**}, {**c**}, and {**d**} in Fig.1).

The removing of all elements of a communication cut set results in the separation of the two distinct societies. An agent in a communication cut set is called an articulation agent (e.g., agent **a, b**, **c**, and **d** in Fig.1). Since agent societies (or species) are represented by complete graphs and these graphs have communication cut sets as intersections, articulation agents can be used to suggest a shortest network path between a query station and the station where an agent finds its goal. Another point is that an articulation agent can hold a repository, which contains the network communication statuses of links of an agent society. Therefore, network resource can be evaluated when an agent checks its surviving environment to decide its evolution policy.

2.2 Mobile agent evolution states

An agent evolves. It can react to an environment, respond to server agent, and communicate with other agents. The evolution process of an agent involves some internal states. As shown in Fig.2, an agent is in one of the following states after it is born and before it is killed or dies of natural:

Ministering: the agent is serving for a user.

Suspending: the agent is waiting for enough resource or connection in its environment.

Dangling: the agent loses connection with server agent, it is waiting for a new admission.

Mutating: the agent is changed to a new identity and admitted to a virtual society.

© 中国科学院软件研究所 http://www.jos.org.cn

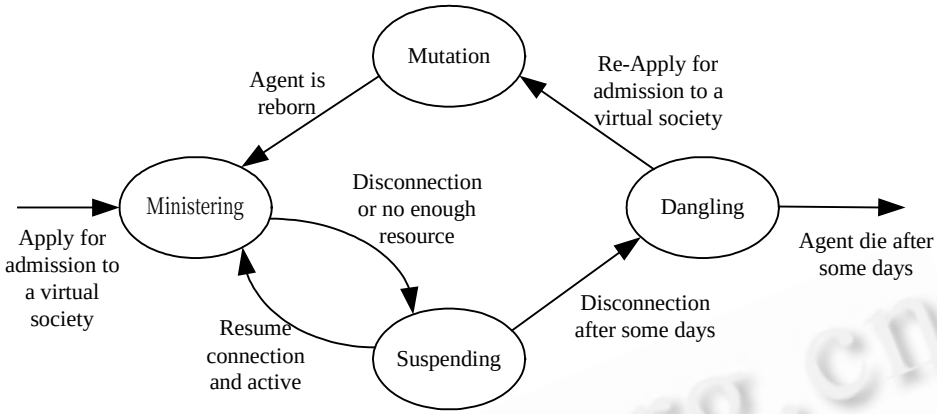

Fig.2 Mobile agent evolution states transition diagram

An agent is born to a ministering state to serve for user (i.e., live in a virtual society). All creatures must have goals (e.g., accomplish one's mission). However, if its surviving environment (i.e., a host station) contains no enough resource, the agent may transfer to a suspending state (i.e., hibernation of a creature). The missioning process will be resumed when the environment has better resources. But, if the environment is lack of resources badly (i.e., natural disasters occur), the agent might be killed. When an agent disconnects from host station (or server agent) for some days. Agents will abort their mission and transfer to a dangling state. An agent in a dangling state cannot survive for a long time. It will die after some days (i.e., a duration of time). Or, it will re-apply for a new admission with a possible new host station, which is a new destination where the agent should travel. In this case, the agent is in a mutating state and is reborn to minister for the new goal.

3 Mobile Agent System Architecture

In this paper, we proposed a model to characterize the mobile agent system architecture. As Fig.3 illustrated, the mobile agent virtual society was composed within three cells: Pico cell, Micro cell and Macro cell. The Pico cell represented the client mobile agent. The Micro cell was constituted by at least one client mobile agents (Pico cells) and one server agent. Several Micro cells construct a Macro cell. The following article will describe the architecture of the Micro cell (server side) and the Pico cell (Client side).

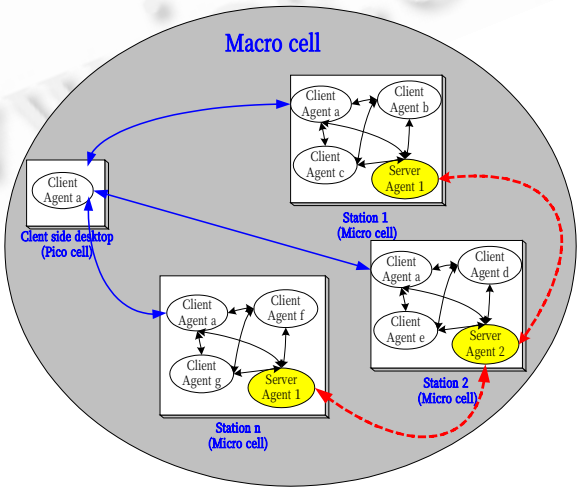

Fig.3 A mobile agent society environment

© 中国科学院软件研究所 http://www.jos.org.cn

Micro cell

A micro cell is composed of agent profiles/database, server agent and web server (Fig.4).

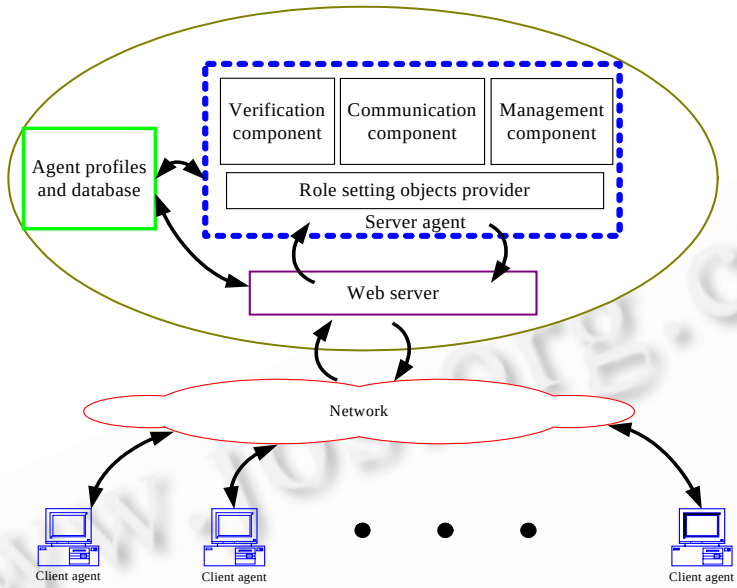

Fig.4 A micro cell architecture

Agent profile and database:

1. User personal environment setting information: This information forms strings, including the items the user choses with their needs for the environment.

2. User Log Files: The log file plays an important part while agent carried back to agent server. The agent server will parse the log files with different catalogs, such as the course participations, the shopping experience, and so on.

3. User personal information: The user in MMVS may change his/her own personal information via the agent. And the changed need to be updated in the member databases.

4. User submitted results: There is some information, which users can submit via the agent architecture, such as the questionnaire system, the pop-up quiz system. Agents will not bring all of the submitted information, and some of the information will be sent back to the database with the functions provided in each subsystem.

Server agent:

The server agent contains four components: (1) verification components, (2) communication components, (3) management components, (4) Roles setting objects (agent characteristics) provider. As with non-mobile agent, the primary requirement is a method of delegating authority to the mobile agent.

1. Verification components: provided the security-minded with agent delegation and authentication, privacy and access control.

2. Communication components: provided the universal communication tools, such as the chat (text), audio, video and windows message (annotation) application tools.

3. Management components: provided the system management facilities, such as the administration, resource allocation and agent profile modification functions.

4. Roles setting objects provider: Provided the application-driven/characteristic objects to the client agent to download those objects. (e.g. E-notebook, Authoring tool (for course design), questionnaire sub-system, lecture-on-demand and so on.)

© 中国科学院软件研究所 http://www.jos.org.cn

Pico cell:

The client agent is the base unit (Pico cell) of the mobile agent society. The main elements of a client agent include the client profiles, object function loader and role setting components selector. (Fig.5)

1. Client profiles: User personal environment setting information, User Log Files, User personal information and User submitted results. (Those are same as the server side's profiles/database).

2. Role setting components selector: this component provided several of role templates for user, user can choose the agent's role which represents what application tools the agent will possess and use in the society.

3. Object function loader: after users had selected their role sets, the object function loader will download the related objects from server agent (Roles setting objects provider).

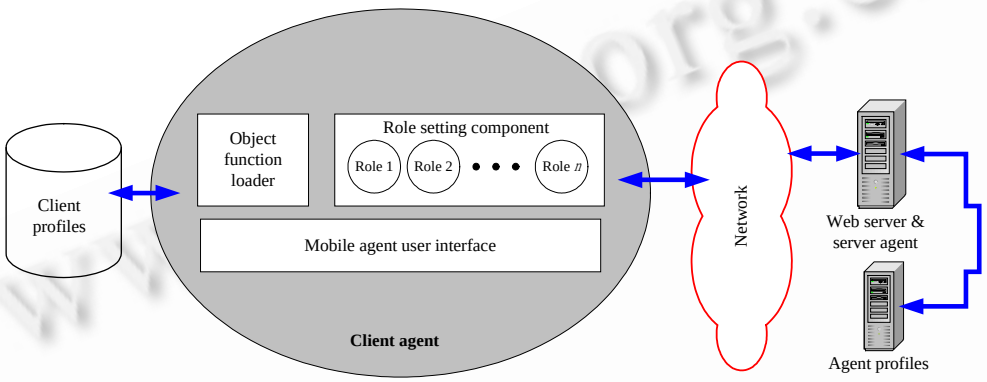

Fig.5 Client agent architecture

Operations:

This section shows several examples of Mobile Agent operations. The examples illustrated three main phrases:

(1) registration

(2) communication

(3) calling a virtual society (Multimedia Macro Virtual Society (MMVS) sub-systems)

Registration in a new MMVS

When a user wants to apply into a new society (MMVS). The client agent is invoked to coordinate the updates between the server and subscriber. Figure 6 shows the operations for a mobile registration with a new society's affiliate.

After the MMVS register system receives the use's application, it sends a registration notification message to the administrator verification component (event 1, Fig.6). In this example (event 2), the administrator verification component sends a reject notification to the MMVS register system. Upon receiving response form administrator verification, the MMVS register system sends a response back to the user (shown in the Figure as event 3).

In event 4, the administrator verification records the registration acceptation message into the profile database. Event 5 is the acceptation response back to the user.

Figure 7 illustrated an example of an invoking communication between two mobile agents. Event 1 is the request message to be sent to the management component of the server station. In turn, after the system's management component agrees this application, it will send the needed communication tool request to the communication tool manager (event 2), this invoking message is forwarded to the destined agent (event 3). If the

© 中国科学院软件研究所 http://www.jos.org.cn

destined agent assents to communicate with the requested one, she will respond to the communication components manager and the desirous service component will be sent to the system's management component (event 4). Thereafter, a connection pre-processing is established as depicted in this figure. The management components will dispatch the service component to original agent then accomplish the connection between two agents.

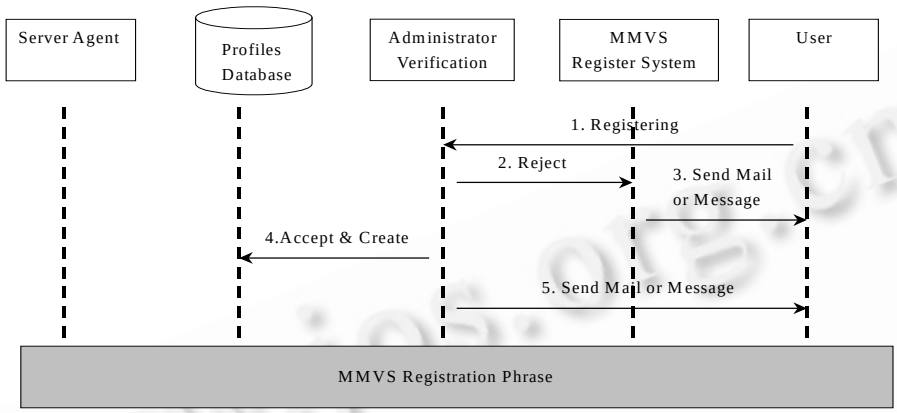

Fig.6 Registration operating phase

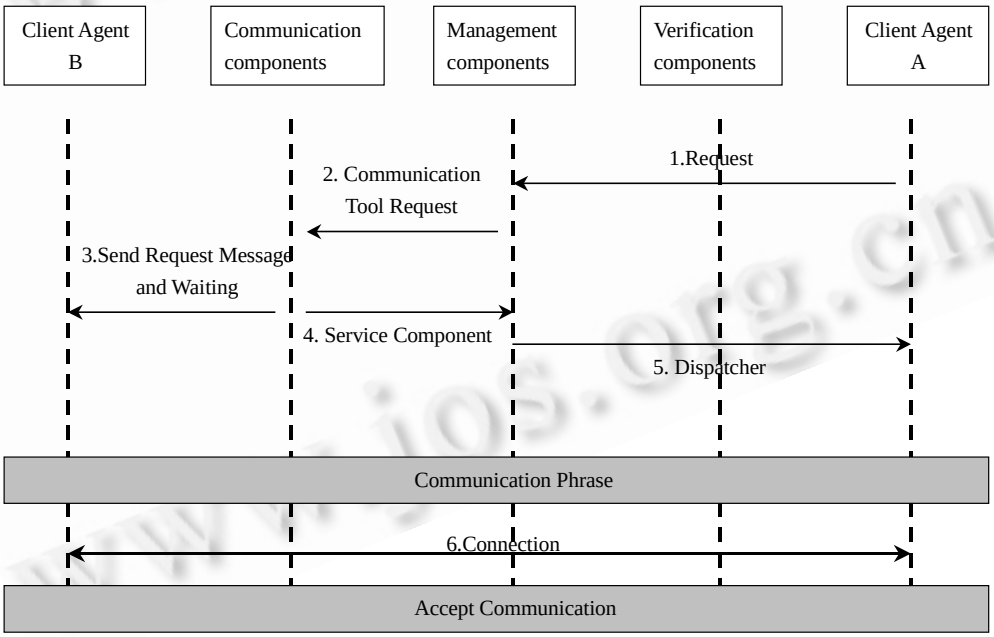

Fig.7 Communication phase

In the event that the client agent applies for the roles setting objects, the server provides a set of commensurate application-driven components for the authorized agent. These operations were shown in Fig.8. The first two events handled the testing and verifying the agent status, and rejected the unauthorized login agents. The function of these events provided the delegation and authentication. Event 3 and 4 provided the role configurations. According the role configuration, event 5, 6 and 7 will record the user's profile and dispatch the desired agent role's objects to the user.

© 中国科学院软件研究所 http://www.jos.org.cn

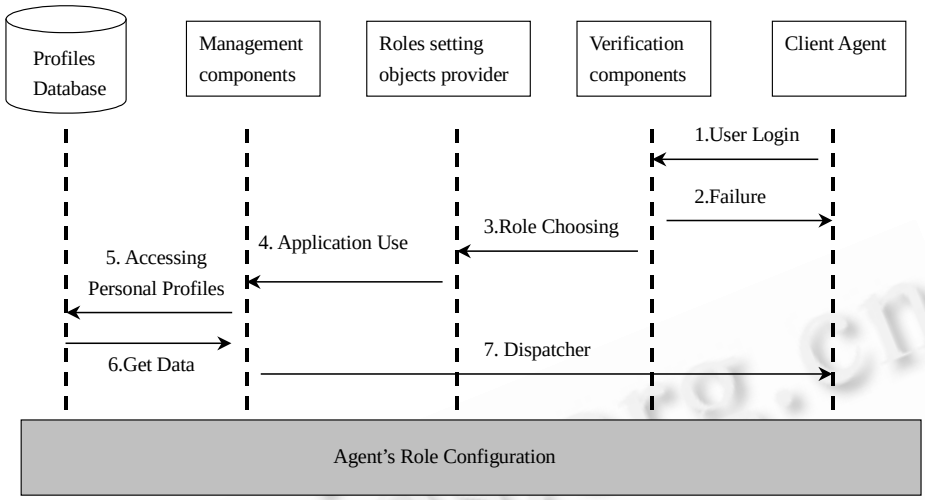

Fig.8 Agent's role setting phase

Formats of the messages

The previous examples are not all of the MMVS operations. Our purpose is to understand the major aspects of MMVS operations, but it should do us no harm to delve a bit further. Therefore, Table 1 lists and describes the mobile agent transactions (messages).

Table 1 Mobile agent transactions (messages) in MMVS

Transaction (Message)	Operations		
	From	To	Function
Reguest_ID	User	MMVS register system	Request for a new ID
Reject_Message	Administrator	MMVS register system	Reject request to illegal
SendMail_ Message	MMVS register system	Client agent	Alarm message to user
Accept_Create	Administrator	Profiles & Database	Create a new user
User_Login	Client agent	Verification components	Enter MMVS agent system
RoleChoose_ Message	Verification components	Roles setting objects provider	Select characteristics or roles
Used_Application	Roles setting objects provider	Management components	Request object-oriented services
Profile_ Message	Management components	Profiles & Database	Find User Personal Data
Dispatcher_List	Management components	Client agent	Dispatcher user object
Communication_Request	Management components	Communication components	Request communication connection
SendRequest_Message	Communication components	Client agent	Waiting for other agent accept
ServiceComponent_List	Communication components	Management components	Send object service component
Identification_ID	Client Agent	MMVS homepage verification	Apply to enter MMVS subsystem
Session_ID	MMVS homepage verification	MMVS subsystem	Open session & enter system

4 Implementation: Mobile Agent in Distance Learning

With the growing popularity of World Wide Web, teachers and students are able to conduct their teaching/learning activity in a virtual classroom via Internet. In addition to access Web multimedia documents, the virtual classroom should have a sets of integrated tools to support customized manipulations of Web multimedia documents created by students or instructors. These integrated tools should further facilitate information exchanges among teachers and students. Thus, we developed an agent-based virtual system and called "multimedia micro-university" (MMU). In this system, users can apply for admission to be a citizen in virtual society (i.e.

© 中国科学院软件研究所 http://www.jos.org.cn

MMU). Then, users can load the multimedia micro-university virtual society agent (MMVS agent). All the applicable agent's objects and programs are defined by Interface Definition Language (IDL). The IDL defines interfaces between client and server programs. IDL can also be used to create client and server programs for heterogeneous network environments that include operating systems such as MS Windows, Unix and Apple.

In the registration sub-system, users must fill in more detailed personal information for the system administrator would check whether you are or not an insider. If the user passed muster with the identity, the user would get the delegation and authentication by host station's supervisor. After finishing registration phase, the MMVS will lead the user into the mobile agent system step by step. The MMVS system and the valet mobile agent were shown in Fig.9.

Fig.9 MMVS system and the valet mobile agent

The valet mobile agent provides the function of the role setting. The role-setting object is an application-driven component, which can provide customization benefits for the user and to match the user's demand. In our system, we provided three roles: student, teacher and system administrator. MMVS system provides the easy administration facilities through the use of server agent. The server agent can make into a list of the agent's information such as user's IP, role and the number of the existent agents.

The role-setting service provides the application-driven components. Different roles of the mobile agents signify they will possess different functions/tools respectively. In MMVS, the student will be provided with some applications, such as communication tools, section course, e-notebook, and the teacher will have authoring tools, communication tools, e-notebook and the system administrator signifies administrator tools. The more detail description of the mentioned application will be introduced in following paragraphs.

When a user wants to chat with other participants, he can input his message into the Chat Input window and this message will be displayed on the Chat Output Window of each participant's chat board. Notice that, in the chat room before the user enters a virtual world, the Chat Input Box is labeled as "Input User Name". Figure 10 is the Web Edit tool, which is designed and implemented using DHTML (Dynamic HyperText Makeup Language) components.

© 中国科学院软件研究所 http://www.jos.org.cn

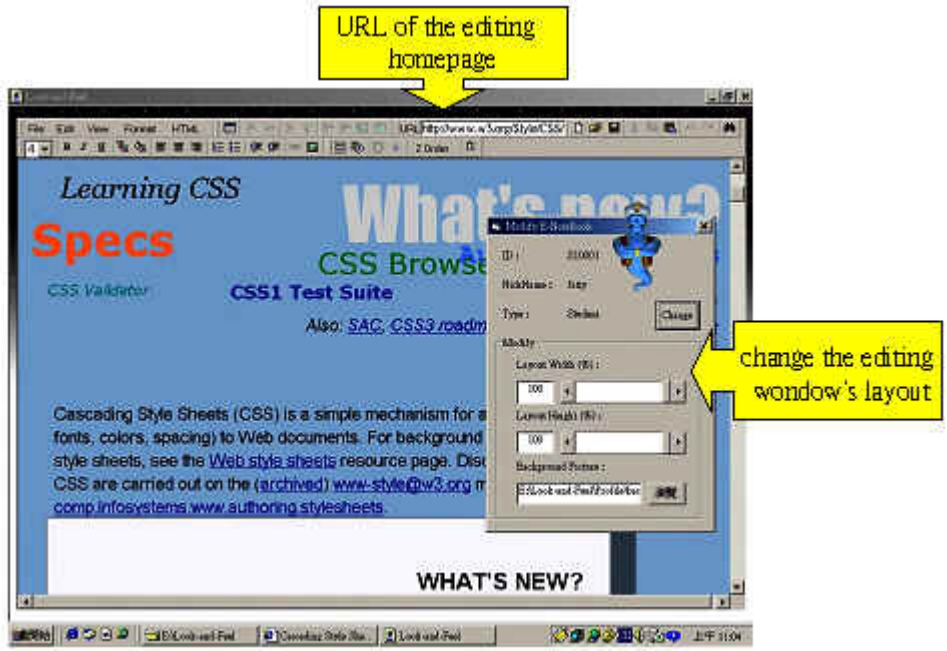

Fig.10 Web editor tool

The DHTML component includes editing features that allow even novice HTML users to create sophisticated Web pages. Editing features include:

Text formatting: Users can set character formats, including font style, face, size, and color. Paragraphs can also be justified and indented.

Editing: The editing function includes multilevel undo and redo commands. Users can also use the standard Cut, Copy, and Paste commands. Elements can be moved or resized by dragging. The component supports a number of keyboard accelerators as well.

Drag-and-drop capability: Users can drag any object, text, or element anywhere on the page.

Absolute positioning: Elements in the document can be absolutely positioned. That is, users can use CSS style attributes to set their location on the page with the equivalent of *x* and *y* coordinates. For more details, see Working with Absolutely Positioned Elements.

Searching: The component can display a find dialog box to allow users to search for text.

Hyper-linkage: Users can define links and bookmarks in the text.

Images: Users can insert images into the document.

Table editing: Users can insert tables and can add and delete columns, rows, and cells.

File management: Users can open existing HTML files from disk or from the Web, save them to disk, and print them. The document being edited can also be loaded from and saved to memory, allowing you to create custom client-server applications.

Context menus: The component allows you to create a context menu that users can display by right button clicking on the document.

The DHTML Editing control provides a variety of means by which you can load documents for editing and save them. Reading documents using URL is useful to get documents from a Web server. Reading and writing string containing the contents of the document is useful, if you need to develop custom solutions for loading and saving documents.

© 中国科学院软件研究所 http://www.jos.org.cn

Users can use the Web Edit tool as a “Notebook” tool. Via only using a Web browser (Internet Explorer4.0 or later) without any other installed applications, user can load and save documents of what he wanted. This tool provided a very important service when students review teaching material or discussing topics.

The authorization system is designed for the system administrator to use. In the system, the lists are presenting the information about new teachers in the MMVS registration. In the authorized process, the system administrator just checks the identity of the people in the lists and decides whether they have the rights or qualifications to be new teachers or not. If the user has the qualifications, the system administrator clicks the “YES, Accept” button. The administrator can just click the “NO, Reject” button if the identification is not valid. Then the authorized process will be finished.

5 Conclusion

In this paper, an agent-based mobility management system was introduced and the ways of constructing the application-driven mobile agent were addressed. We suggested a framework to model the mobile agent virtual society, which contained both mobile agent communication network and mobile agent evolution states. This approach aspired to provide to the software developers who could get advantages in the agent computing and the management routine work. Also, the role-setting components are object-oriented approach. This approach not only gives the flexibility but also scalability to user’s utility tools. In our experiments, the application-driven mobile agent actually improved the persistent look-and-feel for roaming student in distance learning environment. We hope that, this study should be prolonged and applied to future communication network environment.

Acknowledgement We thank Mr. Wang Jung-Huang for implementing the mobile agent system and providing the distance learning system for Tamkang University.

References:

[1] Caglayan, A., Harrison, C. Agent Sourcebook: A Complete Guide to Desktop, Internet, and Intranet Agents. Wiley Computer Publishing, 1997.

[2] Dharap, C., Freeman, M. Information agents for automated browsing. In: Proceedings of the 1996 ACM CIKM Conference (CIKM’96). Rockville, MD, 1996. 296~305.

[3] Pazzani, M., Billsus, D. Learning and revising user profiles: the identification of interesting web sites. Machine Learning, 1997,27: 313~ 331.

[4] Annunziato, J.G. A review of agent technology. In: Proceedings of the International Conference on Multi-Agent Systems. San Francisco, California, 1995.

[5] Tardo, J., Valente, L. Mobile agent security and telescript. In: Proceedings of the 41st IEEE Computer Society International Conference (COMPCON’96). Santa Clara, CA, 1996. 58~63.

[6] Charlton, P., Chen, Y., Mamdani, E., *et al*. An open agent architecture for iIntegrating multimedia services. In: Proceedings of the Autonomous Agents 1997 Conference. Marina Del Rey, California, 1997. 522~523.

[7] Magedanz, T., Eckardt, T. Mobile software agents: a new paradigm for telecommunications management. In: Proceedings of the 1996 IEEE Network Operations and Management Symposium (NOMS’96). Kyoto, Japan, 1996. 360~369.

[8] Lingnau, A., Drobnil, O. Making mobile agents communication: a flexible approach. In: Proceedings of the 1st Annual Conference on Emerging Technologies and Application in Communications. Portland, OR, USA, 1996. 180~183.

[9] Petrie, C.J. Agent-Based engineering, the web, and intelligence. IEEE Expert, 1996,11(6):24~29.

[10] Krause, S., Magedanz, T. Mobile service agents enabling intelligence on demand in telecommunications. In: Proceedings of the 1996 IEEE Global Telecommunications Conference. London, UK, 1996. 78~84.

© 中国科学院软件研究所 http://www.jos.org.cn

[11] Krause, S., de Assis Silva, F.M., Magedanz, T. MAGANA—a DPE-based platform for mobile agents in electronic service markets. In: Proceedings of the 3rd International Symposium on Autonomous Decentralized Systems (ISADS'97). Berlin, Germany, 1997. 93~102.

[12] Kiniry, J., Zimmerman, D. Hands-On look at Java mobile agents. IEEE Internet Computing, 1997,1(4):21~30.

[13] Genesereth, M.R. Software agents. Communications of the ACM, 1994,37(7):48~54.

[14] Gray, R., Kotz, D., Nog, S., *et al*. Mobile agents: the next generation in distributed computing. In Proceedings of the 2nd Aizu International Symposium on Parallel Algorithm/Architecture Synthesis. Fukushima, Japan, 1997. 8~24.

[15] Schoonderwoerd, R., Holland, O., Bruten, J. Ant-Like agents for load balancing in telecommunications networks. In: Proceedings of the 1st International Conference on Autonomous Agents. Marina Del Rey, California, 1997. 209~216.

[16] Sapaty, P.S., Borst, P.M. WAVE: mobile intelligence in open networks. In: Proceedings of the 1st Annual Conference on Emerging Technologies and Applications in Communications. Portland, OR, 1996. 192~195.

[17] Kotz, D., Gray, R., Nog, S., *et al*. Agent Tcl: targeting the needs of mobile computers. IEEE Internet Computing, 1997,1(4):58~67.

[18] Gray, R., Kotz, D., Nog, S., *et al*. Mobile agents: the next generation in distributed computing. In: Proceedings of the 2nd Aizu International Symposium Parallel Algorithms /Architecture Synthesis. Aizu, Japan, 1997. 8~24

移动代理人编程技术与移动用户管理系统探讨

施国琛[1], 邓有光[1], 黄德胜[1], 马建华[2], 黄润和[2]

[1](淡江大学 资讯工程研究所,台北,中国);

[2](法政大学 资讯科学系,日本)

摘要: 有自我控制与辨别身份能力的移动代理人编程技术,不但能够在网路上移动,也能够与其他代理人编程沟通及执行任务.由于移动代理人常常运作于异质的网路或作业系统(operating system)环境下,因此,一个具有整体逻辑性操作界面来存取实体结构的管理系统就显得越发重要.提出了一个能够追踪和维持移动代理人的管理系统,并且应用于远距教学(distance learning).此系统的主要目的在于提供移动代理人通用的存取环境.为了能使此系统完全运作,论述了移动代理人通信网路模组、移动代理人演化阶段来支援管理代理人与用户端代理人.另外,也将移动代理人所使用的工具编程整合成为角色扮演(role-setting)物件;此角色扮演物件是符合所谓应用编程驱使元件(application-driven component)的,因此,此系统也能够符合专业化利益的使用者的需求.

关键词: 移动代理人;代理人通信网路;代理人演化阶段;观感;用户端代理人;伺服器端代理人

中图法分类号: TP391 文献标识码: A

© 中国科学院软件研究所 http://www.jos.org.cn