Universal Access for Roaming User via Mobile Agent Technology

Lawrence Y. Deng[1], Timothy K. Shih[2], Teh-Sheng Huang[2], Chun-Hung Huang[2], Ruei-Xi Chen[2] and Yang-Lang Chang[2]

[1]*St. John's & St. Mary's Institute of Technology*
Aletheia University
Tamsui, Taiwan, R.O.C.
E-mail: Lawrence@mail.sjsmit.edu.tw
[2]*Department of Computer Science and Information Engineering*
Tamkang University
Tamsui, Taiwan, R.O.C.
E-mail: tshih@cs.tku.edu.tw

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 is become more and more urgent. In this paper, we proposed a tracking and persistent agent-based mobility management system in the 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, we encapsulated 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 and matching the user’s demands. These integrated technologies are sufficient in distance learning (virtual university) environment.

***Key Word*:** Mobile Agent, Agent Communication Network, Agent Evolution States, Persistence Look-and-feel, Client Agent and Server Agent

1. Introduction

Autonomous agents and multi-agent technology play a control role in network management and systems management communities for several years [1-4]. Besides, many marketing and technical terms under which agent supports in desktop, Internet, Intranet and so on. 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.

The ever-increasing growth of mobile agent applications are encouraging research aimed at the wide spread communication infrastructure. In [4], the mobile agent paradigms and technologies were discussed. The relationships among paradigms and technologies for mobile agent are showed in Table 1.

Table 1. Relationships among paradigms and technologies

Technologies	Paradigms		
	Client-Server	Remote Evaluation	Mobile Agent
Message-based	Well-suited	Code as data Program interpretation	Code and stats as data Program state restoring Program interpretation
Weakly mobile	Code is a single instruction Creates unnecessary execution units	Well suited	States as data Program state restoring
Strongly mobile	Code is a single instruction Creates unnecessary execution units Move state back and forth	Manage migration Move state back and forth	Well suited

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. Therefore, an integrated logical interface to access physical structure via mobile agent application is become more and more important. For flexibility, mobile agents can be accepted as a design paradigm like object-oriented programming or client/server computing. So, we proposed a tracking and persistent agent-based mobility management system in the case of distance learning to illustrate the entire processes.

Students of the virtual university roam from station to station. It is important to provide a persistent environment such 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.

The remaining parts of this paper were organized as follows: The related work was discussed in section 2. A mobile agent system architecture model describes server agent and client agent components are 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 our conclusion and possible extensions in section 5.

2. Related Works

Software agents have a broad appeal across several industries. These agent applications include computing, telecommunication, manufacturing, entertainment and electronic commerce. Practical agent technologies in various businesses are complex technical challenges when the application developers to develop and deploy agent applications. The term *mobile agent* contains to distinct perceptions: agency and mobility. Mobile Agents is an agent encapsulates the program that a receiving server is to execute, the data comprising the program's arguments and state. Figure 1 shows a mobile agent scenario. Software agents that transport themselves form a client computer to various servers for remote execution. In contrast to remote procedure calls (RPCs) that are limited to communicating data to a procedure to be executed on a remote server, mobile agent transport both the data and program acting on data within user's specification/configuration.

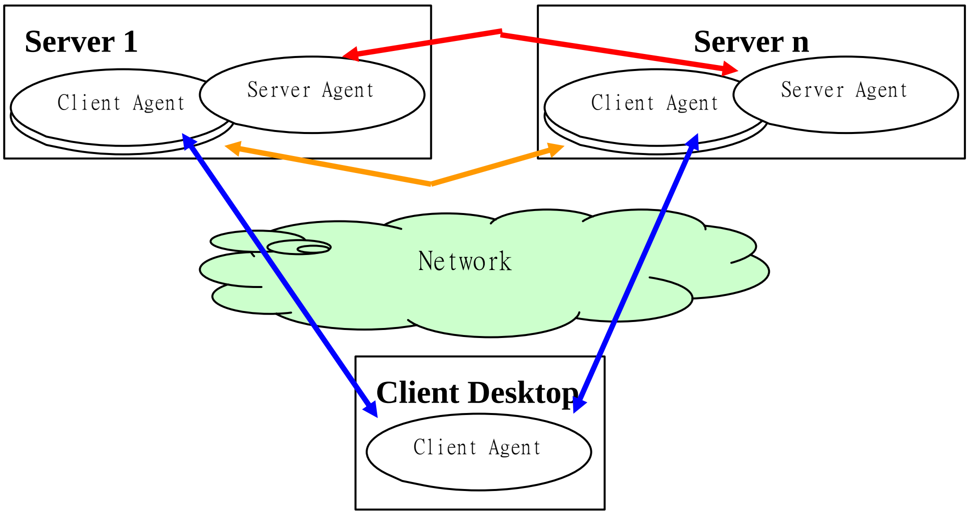


Figure 1. Mobile agent scenario

The concept of agent-based software engineering is discussed in a survey paper [4]. The author presents two important issues: agent architecture and agent communication language. 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 [5]. Agent communication languages allow agents to share information and send message to each other.

Mobile agent needs a programming language that lets users define the role of their agents as they travel across a network. The following languages can be used for specific niches as well.

KQML: A language developed through ARPA knowledge sharing effort aimed at developing sharable large-scale knowledge bases to enable agent-to-agent communication and knowledge. KQML is Suitable for development agent prototype for integration into existing application [6].

Java: Java is an interpreted, multithreaded, and secured platform-independent language suitable for agent application. The multithreading and built-in security features give Java an edge in the implementation of interactive e-commerce applications [7].

ActiveX/OLE: The programming technologies from Microsoft Corporation. Microsoft's answer to Java embracing both Java and Component Object Model (COM) based on Object Linking Embedded (OLE). [8]

Tcl and Safe-Tcl: This is a machine-independent scripting language that uses email to transport agent procedures as contents of mail messages [9].

Telscript: A language proposed by General Magic. Telescript is an object-oriented language and agent-based operating environment designed for mobile agent [11].

Survey of different types of agents can be found in [12-13]. In [12], the author summarized three types of agents: intelligent autonomous agents, mobile agent, and cooperative agents. Agent applications and development are also discussed. The author [12] also indicated that. Web-based intelligent agents are necessary and should be a trend of new engineering applications.

The concept of mobile agent is discussed in several articles [13-16]. The mobile agent architecture, MAGNA, and its platform are presented in [13]. Another agent infrastructure is implemented to support mobile agents [17]. A mobile agent technique to achieve load balancing in telecommunications networks is proposed in [18]. 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 required of a distributed agent environment are discussed in [19]. The evaluation of several commercial Java mobile agents is given in [20-23].

3. Mobile Agent System Architecture

In this paper, we proposed a model to characterize the mobile agent system architecture. As Figure 2 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 constituted by at least one client mobile agents (Pico cells) and one server agent. Several Micro cells construct a Macro cell. The following article describes the architecture of the Micro cell (server side) and the Pico cell (Client side).

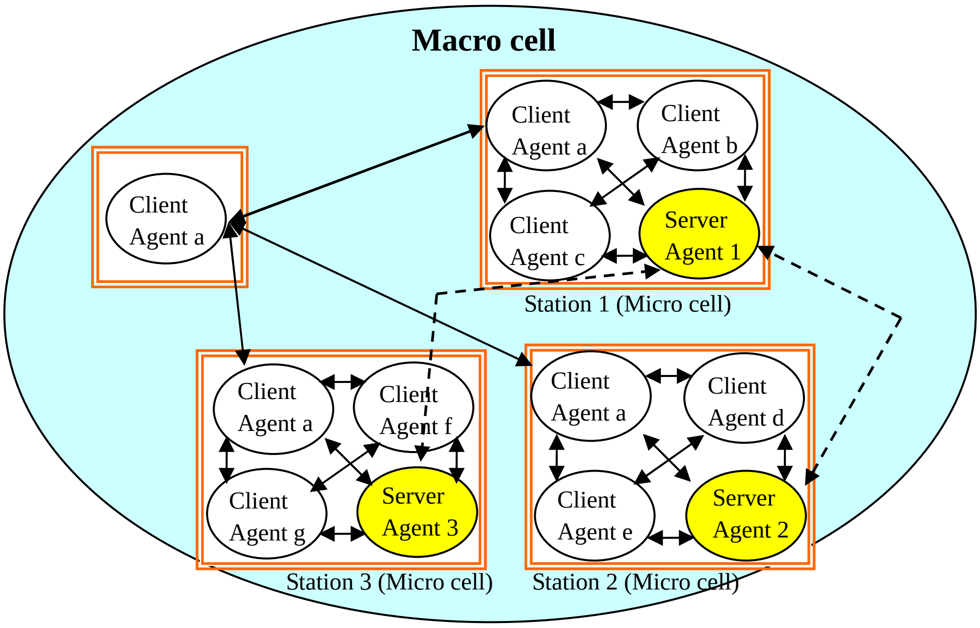


Figure 2. A mobile agent society environment

3.1 Micro Cell

A micro cell is composed of agent profiles/database, server agent and web server shown in Figure 3.

3.1.1 Agent Profile and Database

- User personal environment setting information: This information forms strings, including the items the user chose with their needs for the environment.
- 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.
- User personal information: The user in MMVS may change his/her own personal information via the agent, the modification of the member databases.
- User submitting results: There is some information, which users can submit via the agent architecture, such as the questionnaire system, the pop-up quiz system. Agents won't bring all of the submitted information, and some of those information will be send back to the database with the functions provided in each subsystem.

3.1.2 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: provide the security-minded with agent delegation and authentication, privacy and access control.

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

(3) Management components: provide the system management facilities, such as the administration, resource allocation and agent profiles modification functions.

(4) Roles setting objects provider: provide 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.)

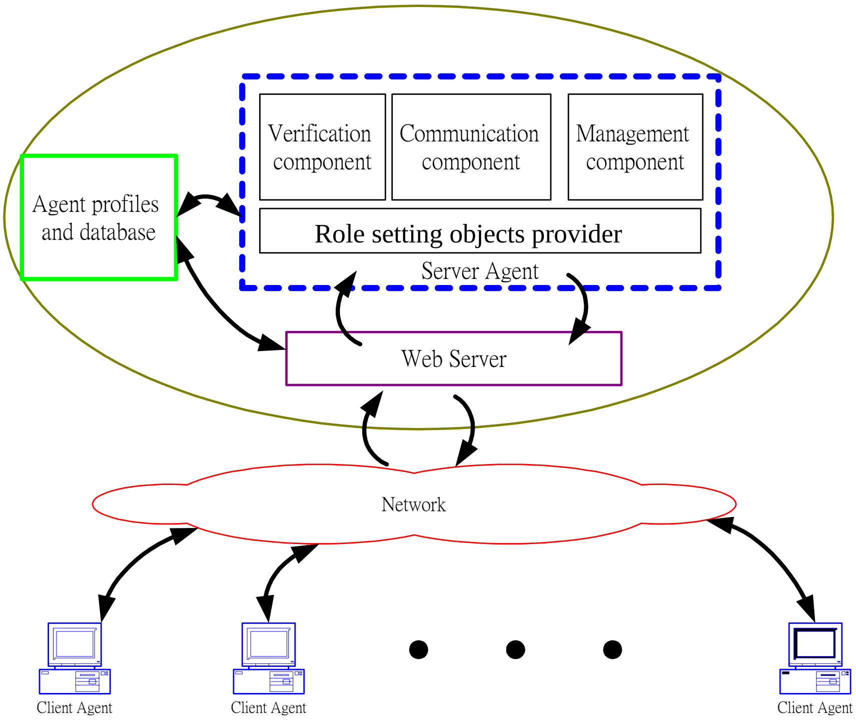


Figure 3. A Micro cell architecture

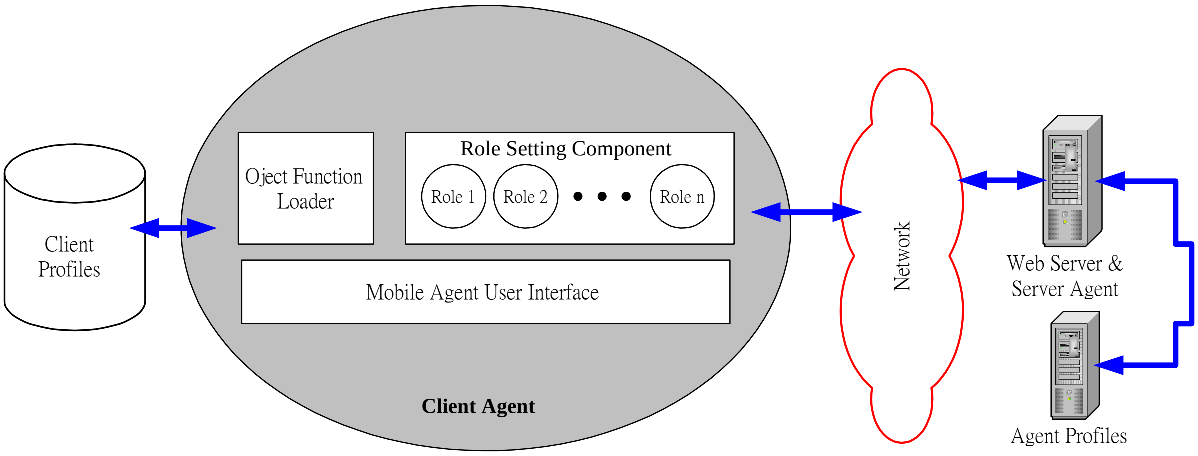


Figure 4. Client agent architecture

3.2 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 shown in Figure 4.

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

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

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

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)

3.2.1 Registration in a New MMVS

When a user wants applying into a new society (MMVS). The client agent is invoked to coordinate the updates between the server and subscriber. Figure 5 shows the operations for a mobile registration to be a new society‘s member.

After the MMVS register system received the use’s apply, it send a registration notification message to the administrator verification component (event 1, Figure 5). 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 send a response back to the user (shown in the Figure 5 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 6 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 apply, it will send the communication tools needed and requests to the communication tool manager (event 2), this invoke message is forwarded to the destined agent (event 3). If the destined agent assents to communicate with the requested one, she will respond to the communication components manager and the desiring service component will 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, and then accomplishing the connection between two agents.

In the event that the client agent applies the roles setting objects, the server provides a set of commensurable application-driven components for the authorized agent. These operations were shown in Figure 7. 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. After the role configuration steps, event 5, 6 and 7 will record the user’s profile and dispatch the desired agent role’s objects to the user.

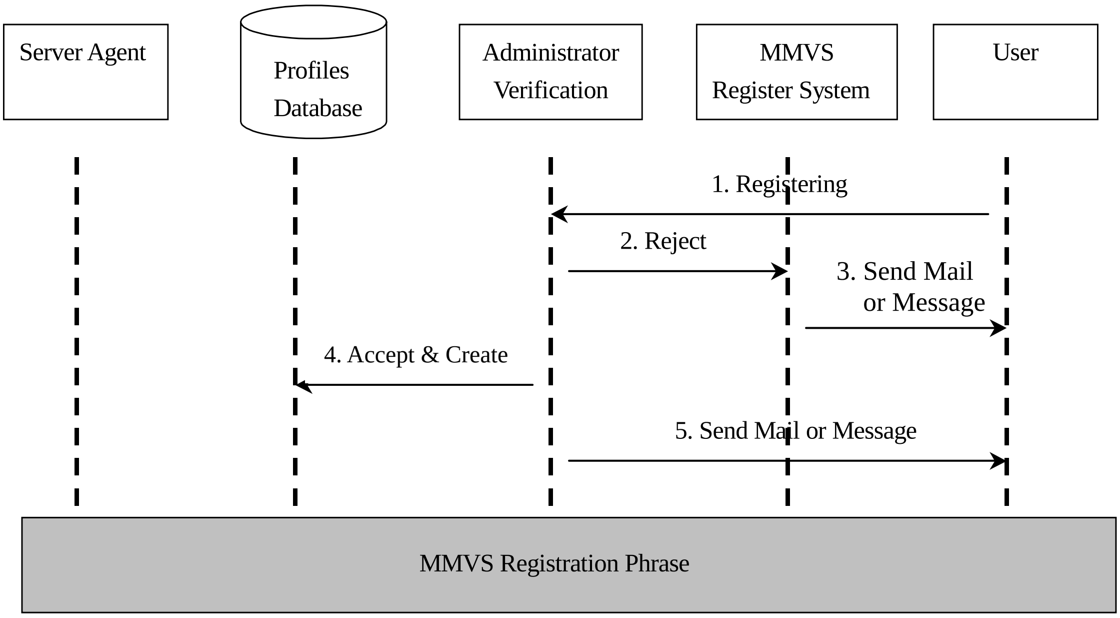


Figure 5. Registration operating phase

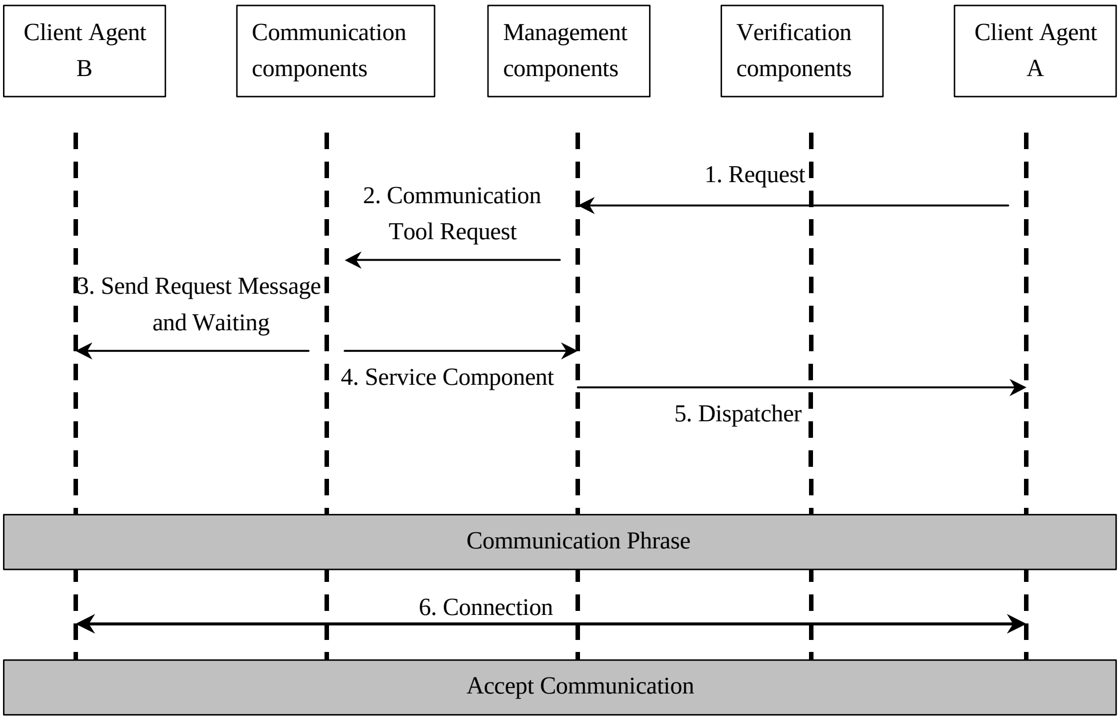


Figure 6. Communication phrase

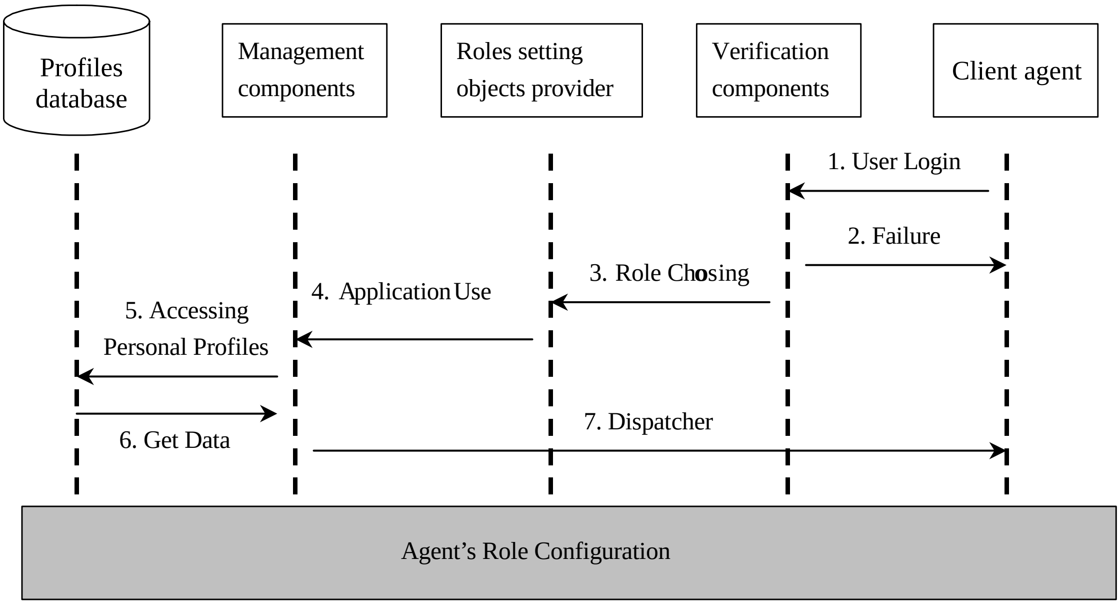


Figure 7. Agent's role setting phase

3.2.2 Formats of the Messages

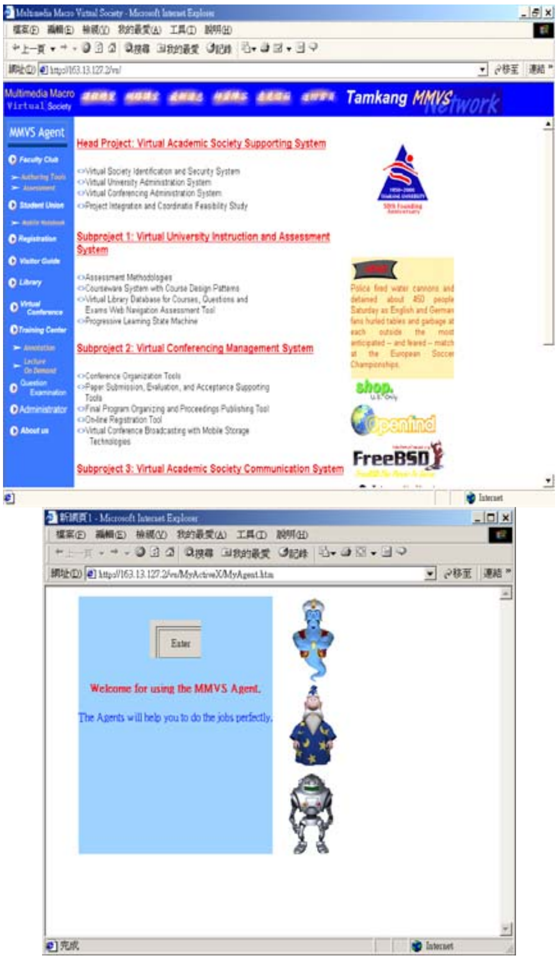
The previous examples are not all of the MMVS operations. Our purpose is to understand the major aspects of MMVS operations, but it should do us no harm to look a bit further. Therefore, Table 2 lists and describes the mobile agent transactions (messages).

Table 2. Mobile Agent Transactions (Messages) in MMVS

Transaction (Message)	Operations		
	From	To	Function
Request_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 set 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 to call “multimedia micro-university” (MMU). In this system, user can apply for admission to be a citizen in virtual society (i.e. MMVS). Then, user can load the multimedia micro-university virtual society agent (MMVS agent). All the applicable agent’s objects/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 such operating systems as MS Windows, Unix and Apple.



Head Project: Virtual Academic Society Supporting System

- Virtual Society Identification and Security System
- Virtual University Administration System
- Virtual Conferencing Administration System
- Project Integration and Coordinatio Feasibility Study

Subproject 1: Virtual University Instruction and Assessment System

- Assessment Methodologies
- Courseware System with Course Design Patterns
- Virtual Library Database for Courses, Questions and Exams Web Navigation Assessment Tool
- Progressive Learning State Machine

Subproject 2: Virtual Conferencing Management System

- Conference Organization Tools
- Paper Submission, Evaluation, and Acceptance Supporting Tools
- Final Program Organizing and Proceedings Publishing Tool
- On-line Registration Tool
- Virtual Conference Broadcasting with Mobile Storage Technologies

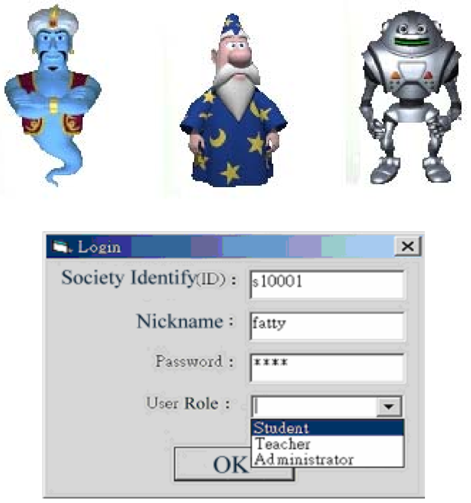
Subproject 3: Virtual Academic Society Communication System

Police fired water cannons and detained about 450 people Saturday as English and German fans hurled tables and garbage at each outside the most anticipated – and feared – match at the European Soccer Championships.

Welcome for using the MMVS Agent.

The Agents will help you to do the jobs perfectly.

Figure 8. MMVS system and the valet mobile agent



Society Identify(ID) : s10001

Nickname : fatty

Password : ****

User Role :

Student
Teacher
Administrator

Figure 9. Agent roles and the client agent login window

In the registration sub-system, user must fill in the more detailed personal information first. The system administrator would check whether you are or not an insider. If the person were legal, he would get the delegation and authentication by host station's supervisor. After finishing registration phase, the MMVS will lead user into the mobile agent system step by step. The MMVS system and the valet mobile agent were shown in Figure 8.

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, as Figure 9 showed.

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 agent signify they will possess different functions/tools respectively. In MMVS, the student will be provided with some application, such as communication tools, section course, e-notebook etc…, the teacher will have authoring tools, communication tools, e-notebook etc… and the system administrator signifies administrator tools, as Figure 10 illustrated. 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, as shown in Figure 11.

The Figure 12 is the Web edit tool, which is designed and implemented by using DHTML (Dynamic HyperText Makeup Language) components. 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.

Searching: The component can display a 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.

User 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 student reviews a teaching material or a discussing topic.

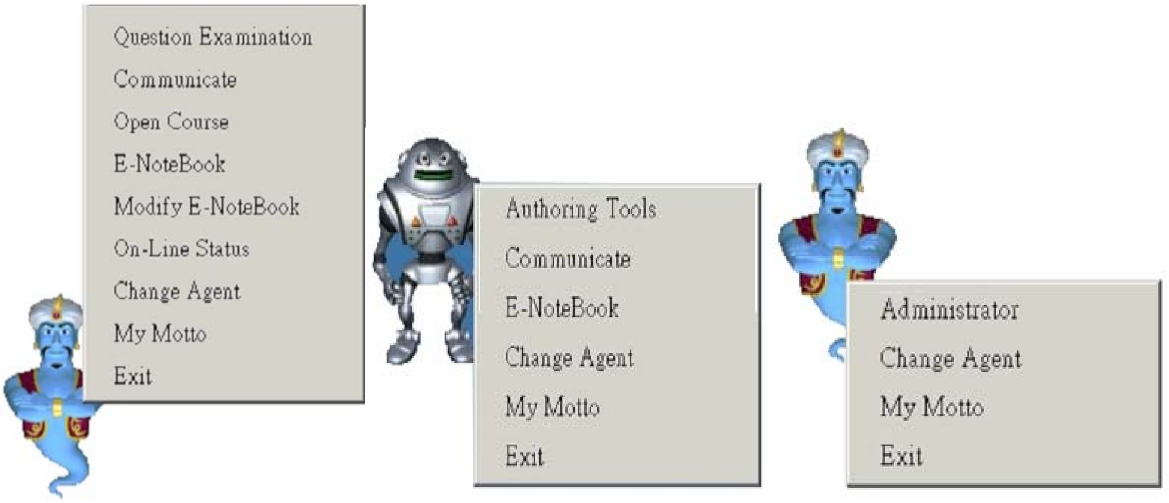
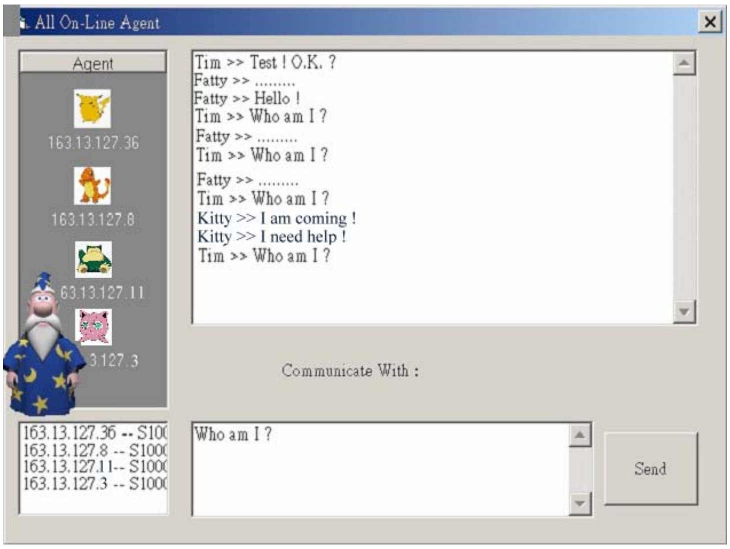


Figure 10. Related functions in roles of the student, teacher, and administrator



Tim >> Test ! O.K. ?
Fatty >> ....
Fatty >> Hello !
Tim >> Who am I ?
Fatty >> ....
Tim >> Who am I ?
Fatty >> ....
Tim >> Who am I ?
Kitty >> I am coming !
Kitty >> I need help !
Tim >> Who am I ?

163.13.127.36 -- S10(
163.13.127.8 -- S100(
163.13.127.11-- S100(
163.13.127.3 -- S100(

Who am I ?

Figure 11. Agent communication tool—chat room

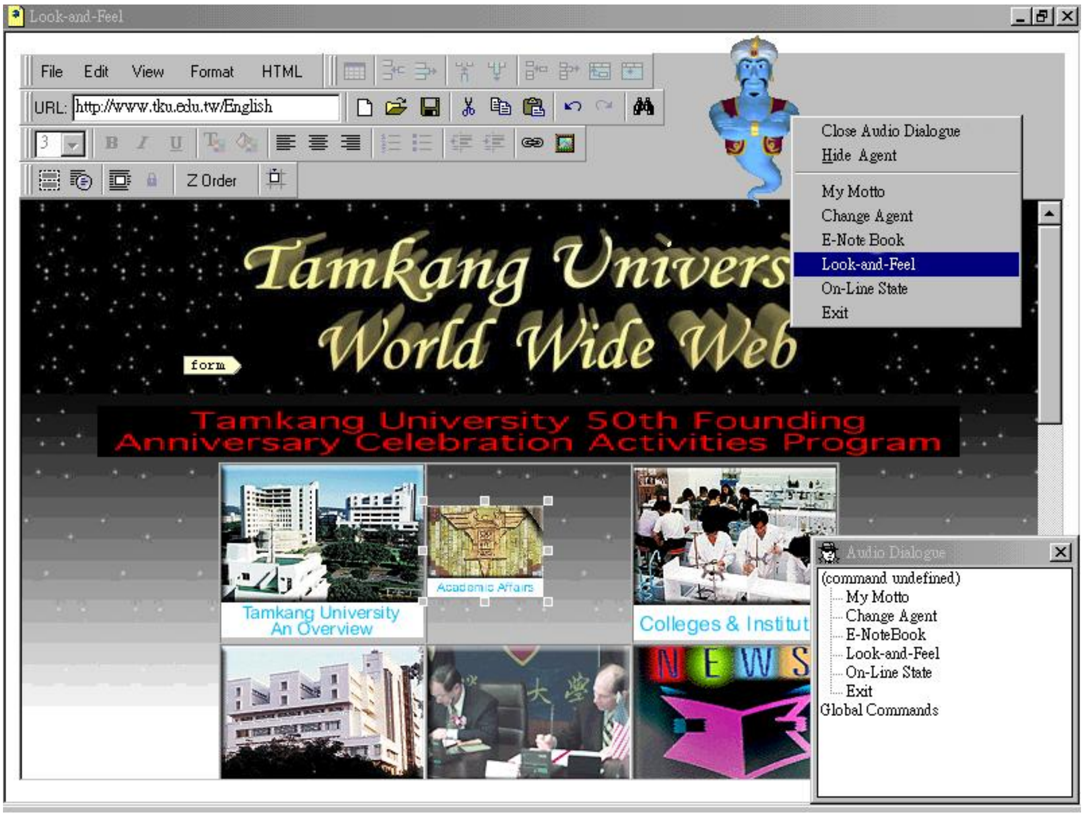


Figure 12. Use mobile E-Notebook tool & edit notebook

5. Conclusion

In this paper, an agent-based mobility management system was introduced and the ways of constructing the application-driven mobile agent was 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 approaches 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 flexibly but also scalable in user's utility tools. In out 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.

References

[1] Software Agents Group at MIT Media Lab: http://agent.www.media.mit.edu/v groups/agents/.

[2] UMBC Intelligent Software Agents Resources: http://www.cs.umbc.edu/agents/.

[3] Charlton, P., Chen, Y., Mamdani, E., Olsson, O., Pitt, J., Somers, F. and Wearn, A., "An Open Agent Architecture for Integrating Multimedia Services," *Proceedings of the Autonomous Agents 97 Conference*, Marina Del Rey, CA, U. S. A., pp. 522-523 (1997).

[4] Chanda, D. and Martin, F., "Information Agents for Automated Browsing," *Proceedings of the 1996 ACM CIKM Conference (CIKM'96)*, Rockville, MD, U. S. A., pp. 296-305 (1996).

[5] Alper, C. and Colin, H., *Agent Sourcebook: A Complete Guide to Desktop, Internet, and Intranet Agents*, Wiley Computer Publishing, (1997).

[6] McKay, D., R, Fritzson, McEntire, and Finin, T., "KQML as an Agent Communication Language," *Proceeding of the Third International Conference on Information and Knowledge Management (CIKM)*, pp. 456-463 (1994).

[7] McGilton and Gosling, "The Java Language Environment: A White Paper," Mountain View, Sun Microsystems, CA, U.S.A., (1995).

[8] "The Component Object Model Specification" http://www.microsoft.com, Microsoft

Corporation and Digital Equipment Corporation, (1995).

[9] Ousterhout, J. K., “Tcl: A Universal Scripting Language,” http://www.smli.com/research/tcl.

[10] White, J. E., ”A Introduction to Safety and Security in Telscript,” http://www.genmagic.com

[11] Charlton, P., Chen, Y., Mamdani, E., Olsson, O., Pitt, J., Somers, F. and Wearn, A., “An Open Agent Architecture for Integrating Multimedia Services,” *Proceedings of the Autonomous Agents 97 Conference*, Marina Del Rey, CA, U.S.A., pp. 522-523 (1997).

[12] Chanda, D. and Martin, F., “Information Agents for Automated Browsing,” *Proceedings of the 1996 ACM CIKM Conference (CIKM'96)*, Rockville, MD, U. S. A., pp. 296-305 (1996).

[13] Robert, S. G., “Agent Tcl,” *Dr. Dobb's Journal of Software Tools for Professional Programmer*, Vol. 22, 18-26 (1997).

[14] Joseph, K. and Daniel, Z., “Hands-on Look at Java Mobile Agents,” IEEE Internet Computing, Vol. 1, pp. 21-30 (1997).

[15] David, K., Robert, G., Saurab, N., Daniela, R., Sumit, C. and George, C., “Agent Tcl: Targeting the Needs of Mobile Computers,” *IEEE Internet Computing*, Vol. 1, pp. 21-30 (1997).

[16] Krause, S. and Magedanz, T., “Mobile Service Agents Enabling Intelligence on Demand in Telecommunications,” *Proceedings of the 1996 IEEE Global Telecommunications Conference*, London, U.K., pp. 78-84 (1996).

[17] Jide, B. O., David, J. K., Stuart, M. W., Nagi, W., Sadanand, S., Chris, G. and JoAnna, G., “SAIRE-A Scalable Agent-Based Information Retrieval Engine,” *Proceedings of the Autonomous Agents 97 Conference*, Marina Del Rey, CA, U.S.A., pp. 292-299 (1997).

[18] Michael, P. and Daniel, B., “Learning and Revising User Profiles: The Identification of Interesting Web Sites,” *Machine Learning*, Vol. 27, pp. 313-331 (1997).

[19] Charles, J. P., “Agent-based Engineering, the Web, and Intelligence,” *IEEE Expert*, Vol. 11, No. 6, pp. 24-29 (1996).

[20] Peter, S. S. and Peter M. B., “WAVE: Mobile intelligence in open networks,” *Proceedings of the 1996 1st Annual Conference on Emerging Technologies and Applications in Communications*, Portland, OR, U.S.A., pp. 192-195 (1996).

[21] Ruud, S., Owen, H., and Janet, B., “Ant-like Agents for Load Balancing in Telecommunications Networks,” *Proceedings of the 1997 1st International Conference on Autonomous Agents*, *Marina Del Rey*, CA, U.S.A., pp. 209-216 (1997).

[22] Joseph, T. and Luis, V., “Mobile Agent Security and Telescript,” *Proceedings of the 1996 41st IEEE Computer Society International Conference (COMPCON'96)*, Santa Clara, CA, U.S.A., pp. 58-63 (1996).

[23] Chirstph, G. T. and Gerhard, F., “Using Agent to Personalize the Web,” *Proceedings of the 1997 ACM IUI Conference (IUI'97)*, Orlando FL, U.S.A., pp. 53-60 (1997).

Manuscript Received: Jun. 22, 2001
Revision Received: Dec. 18, 2001
and Accepted: Aug. 16, 2002