B8-3

wwHww : An Application Framework for End-User Computing in Multi-organizational Office Network Systems

Takeshi CHUSHO　Kazuaki KASHIWAGI[*1]　Yasuo KASAMA[*2]
Meiji University, Department of Computer Science
email : chusho@cs.meiji.ac.jp

ABSTRACT

Now is the time for shifting our view of computerization from work-life to social life, which is called "CS-life" (Computer-Supported Life) in this paper since the number of end-users increases on the inside and outside of offices. In the first step for CS-life, this paper describes an application framework of the MOON (multi-organizational office network) systems for window work in banks, city offices, travel agents, mail-order companies etc. based on the philosophy of "All routine work both at office and at home should be carried out by computers." The common protocol for communication between the M kinds of servers at windows and the N kinds of client terminals reduces the number of application interfaces for the MOON systems from M*N to M+N.

1 INTRODUCTION

The number of end-users increases on the inside and outside of offices as computers and computer networks are coming into wide use. In the near future, the spread of the Internet such as GII or NII will accelerate this trend.

The word "end-user" has already implied people in the wider range than those who belong to end-user departments contrasted with information system support departments in user companies. The purposes of information systems have already included social life support in addition to productivity improvement of company activities. Compared with CSCW , computerization of social life is called "CS-life" (computer- supported life) [1] in this paper.

Our philosophy is:

"All routine work both at office and at home should be carried out by computers."

Let's consider an application system for windows or counters.in banks, city offices, travel agents, mail-order companies etc. As computer networks spread widely and rapidly in the near future, the information society will require such new technologies that application experts can automate their own work at windows by themselves and that almost all clients can operate computers at home or at office without the help of others.

As the one solution of these problems, this paper describes an application framework, wwHww, for end-user computing under distributed systems [2]. The name of wwHww is derived from the proposed common protocol of 'who-what-how and when-where' and is pronounced as 'who' for convenience. This protocol is based on a distributed cooperative model of object-oriented technology.

There are many related studies in the research fields of office information systems, CSCW and object-oriented technology. Although most studies support systems for private use or ones for only one company, the wwHww system supports

*1 Sumisho Computer Systems Co. since 1995.4
*2 Yamatake Honeywell Co. Ltd. since 1995.4

multi-organizational business use for end-user computing.

2 An Explosive Increase of Interfaces

It must be very convenient if we can communicate our requests to a window via a computer network without visiting the window. However, as such windows increase, we must learn various user interfaces. Furthermore, as we can use various types of terminals at various places for our requests, we must learn different user interfaces of different types of terminals. This inconvenience is caused by developing each application individually. For example, although some commercial computer network in Japan supports five mail-order book shops, their user interfaces are all different each other.

In order to avoid developing M*N kinds of user interfaces on application software for connecting M servers at windows to N kinds of client terminals, we developed a common protocol between servers and clients for end-user computing, which reduces the number of user interfaces from M*N to M+N.

3 APPLICATION FRAMEWORK

3.1 An Overview of Application

The main purpose of the application framework, wwHww, is the easy construction of a multi-organizational office network (MOON) for window application as shown in Figure 1. The MOON system is based on a client/server model and the terminals are classified into the following three groups:

(1) Client terminals to send applications to windows.
(2) Servers at windows to receive applications.
(3) MOON servers to manage the network system.

In the near future, there will be many types of client terminals such as personal computers and workstations both at home and at office, portable computers outdoors, public telephones with terminals in town, etc. There are already uncountable windows to be servers in mail-order companies, city offices, travel agents, universities, academic societies, caterers, etc. The MOON servers include a form server, a name server, a message server and a security server.

3.2 The Benefits of MOON systems

This system produces the following benefits by standardization of a common protocol for communication between client terminals and window servers, by embedding agents in both client terminals and window servers and by intelligent information retrieval services based on the MOON servers:

(1) Benefits for clients visiting windows:
(a) Clients understand easily to whom, what, and how to request.
(b) Clients get application forms, understand how to fill in the form and send them without visiting windows.
(c) Clients get quick responses or inquire easily about the state of their requests.
(d) Clients don't need to wait in line at windows and to write such plain words as their names, addresses and phone numbers since their agents do them instead.
(e) Clients send applications to various windows of different organizations in the uniform manner.
(2) Benefits for application experts and their

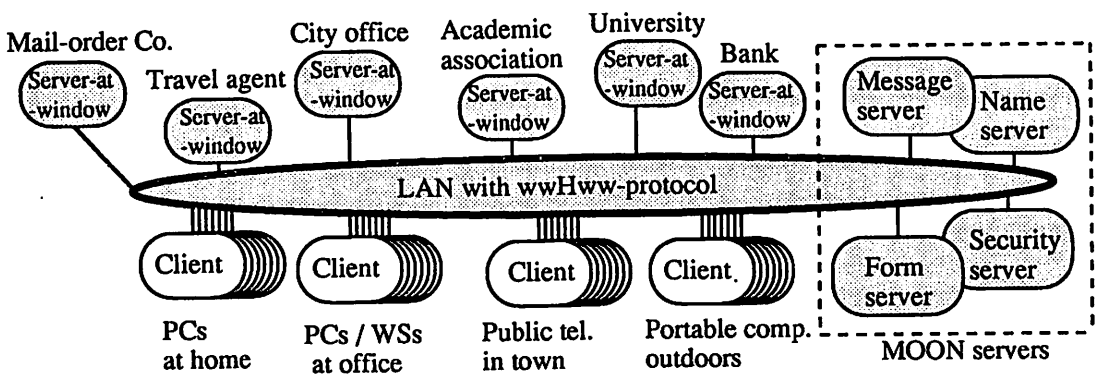


Figure 1. A MOON (multi-organizational office network) system.

organizations:

(a) Experts can free themselves from frequently asked questions at windows since their agents reply to the questions instead.

(b) Experts spend much time for such creative work as improvement of processes for applications.

(c) Expertise which belong to individual experts becomes common property in an organization where they work.

3.3 Essential Requirements

The following requirements are essential in order that the MOON system produces the aforementioned benefits:

(1) Clients can operate terminals and can teach the fixed operations to their agents by themselves.

(2) Application experts can teach their expertise to their agents by themselves.

At present, clients fill in the form at widows by referring to a written example or by asking experts. The first item implies that clients can fill in the form by computers as easily as they do it at windows. The second item implies that experts can express their expertise in understandable words for computers. For these requirements, a new programming paradigm is indispensable.

4 CONCEPTUAL DESIGN

4.1 The Common Protocol

The common protocol for communication between servers at windows and client terminals, named the wwHww-protocol, is derived from the following message components of requests to windows:

>Who receives your request?
>What do you request to the window?
>How do you request it?
>When is the time limit?
>Where is the result sent?

The following component is added to these five components.

>Which is your request?

That is, the basic form of the wwHww-protocol is shown as follows:

(who, what, how, when, where, which)

The semantics of the protocol is based on a message passing concept of object-oriented technology. The six parameters of the protocol correspond to components of a message between objects as follows:

who : A message receiver
what : A method name
how : Parameters of the method
when : A time limit of the reply
where : An address for the reply
which : A message number

In the MOON system, the who-parameter implies a window where a written application is sent. The what-parameter implies the title of the application form. The how-parameter implies contents of the application form. The when-parameter implies a deadline for the reply to the request. The where-parameter implies an address for sending the reply. The which-parameter implies a receipt number stamped on the received application form.

Whether values of these parameters are known or unknown, varies semantics of the message. If a value of a message parameter is unknown, the message implies an inquiry about the parameter. Examples are given next.

4.2 End-user Interface

The actual end-user interface of the common protocol is different from the basic form which is the internal representation in the system. There must be some kinds of end-user interfaces corresponding to the kinds of client terminals. It may be an interface of filling in the form displayed on a screen. It may be the other interface of sequentially inputting answers corresponding to questions displayed in characters. Some of them may display icons or a menu for selection.

Let's consider a student who wants to get his transcript from a university and suppose that he does not know where and how he gets it in the university. The operations and the basic form of a common protocol to be send to the system at each

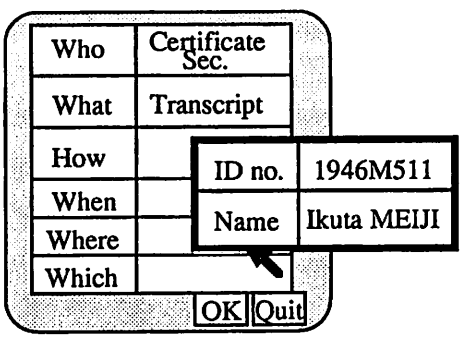
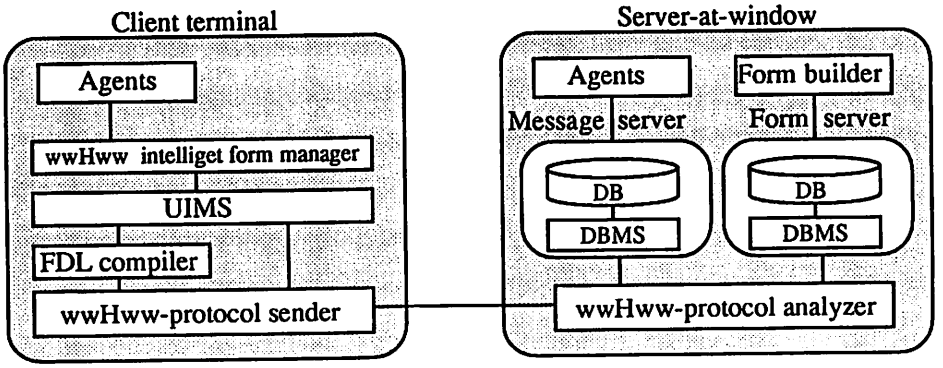


Figure 2. An example of operations.



[Notes] FDL : Form Definition Language
UIMS : User Interface Management System

Figure 3. A prototype of the wwHww system.

step are described as follows:

(a) The keyword of "transcript" is inputted to the what-column in the initial screen, and then the basic form of (?x, ?y="transcript", , , ,) is sent.

(b) The system displays "Certificate Sec." in the who-column and "Transcript form" in the what-column. After clicking the value area in the what-column, the basic form of ("Certificate Sec.", ?"Transcript form", , , ,) is sent.

(c) The system displays the help message on the transcript form. After clicking the value area of a blank in the how-column, the basic form of ("Certificate Sec.", "Transcript form", ?z, , ,) is sent.

(d) The system displays the application form. After filling in the two blanks for the ID no. and the name and clicking the value area of a blank in the where-column, the basic form of ("Certificate Sec.", "Transcript form", ("ID no."="1946M511", "name"="Ikuta MEIJI"), , ? v, w) is sent as shown in Figure 2.

(e) The system displays the menu on the address for sending a certificate. After selecting "Home", the basic form of ("Certificate Sec.", "Transcript form", ("ID no."="1946M511", "name"="Ikuta MEIJI"), , "Home", w) is sent.

(f) The system displays the message number in the which-column, while the value is assigned to the variable, w. Then the system terminates by clicking the close button.

5 SOFTWARE ARCHITECTURE

The application framework, wwHww, is used for the easy construction of the MOON system, and then provides three frameworks corresponding to the following three kinds of terminals of the MOON systems:

(1) Client terminals to send applications to windows.

(2) Servers at windows to receive applications.

(3) MOON servers to manage the network system.

We have already developed the prototype program for feasibility study. The software architecture of the prototype is shown in Figure 3 [3]. The prototypes of the MOON servers were implemented in the window server.

6 CONCLUSIONS

For computerization of social life called "CS-life" in this paper, the application framework, wwHww, was developed. The MOON system is easily developed by using this framework.

REFERENCES

[1] Chusho,T.: wwHww : An Object-Oriented Model for End-User Computing in Distributed Office Systems, (in Japanese), *Information Processing Society of Japan, SIG on Software Engineering*, Vol. 94, No. 18 (1994), pp.33-40.

[2] Chusho, T.: CS-life, (in Japanese), *Computer Software*, Vol. 11, No. 6 (1994), pp.1-2.

[3] Kashiwagi, H. et al.: wwHww : An Application Framework for Distributed Cooperative Systems, *The 50th Annual Conference of Information Processing Society of Japan* (1995), p.5/259-262.