明治大学科学技術研究所紀要
35(1) : 1-18, 1996

Mem. Inst. Sci. Tech. Meiji Univ.
35(1) : 1-18, 1996

# wwHww : An Application Framework of Multi-organizational Office Network Systems

Takeshi CHUSHO

*Department of Computer Science,*
*School of Science and Technology, Meiji University*
*1-1-1 Higashimita, Tama-ku, Kawasaki, 214-71, Japan*

Received January 8, 1996; Accepted March 22, 1996

Synopsis : 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. Application experts at windows become free themselves from frequently asked questions and can spend much time for creative work since they can teach their expertise to agents in computers by themselves and the agents reply to the questions instead. Without visiting the windows, clients can get application forms, understand how to fill in the form and send them since they can operate terminals by themselves. These benefits are ascertained by developing the prototype program for feasibility study.

Key words : application framework, end-user computing, object-oriented technology, office information system, distributed computing

## 1. Introduction

The number of end-users of computer systems 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 Internet such as GII(global information infrastructure) or NII(national information infrastructure) 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 (Computer-supported cooperative work), computerization of social life is called "CS-life" (computer-supported life) [6] in this paper.

For CS-life, there are at least three different kinds of symbiotic styles between humans and computers:

(1) Computers carry out the job of humans instead of them.

(2) Computers improve the quality of work which humans do.

(3) Computers improve the quality of play which humans enjoy.

Although many controversies about how to change work with computer systems are aroused [11], the first item is paid attention to based on the following philosophy in this paper:

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

Our policy for this purpose is to develop tools for implementing information systems of the end-users, by the end-users, for the end-users because package software may not substitute for various work of various end-users. By liberation from routine work, humans acquire the following benefits:

(1) At office, humans spend more time for creative work.

(2) At home, humans spend more time for their enjoyable life.

In the first step for CS-life, let's consider an application system for windows or counters in banks, city offices, travel agents, mail-order companies, etc. If the window work is automated and is done via computer networks, application

experts at windows will have the benefit of the aforementioned first item, and clients or customers visiting windows will have the benefits of the second item or the both items.

Both friendliness of end-user interfaces for clients and automation of routine work for application experts are still insufficient, although current computer networks have already supported some kind of window work such as mail-order business. 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. Furthermore, a common protocol between the windows and their clients must be developed for avoiding appearance of a number of incompatible interfaces corresponding to the explosive number of combination of the windows and their clients.

As the one solution of these problems, this paper describes an application framework, wwHww, for end-user computing under distributed systems [4]. 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 (OIS), computer-supported cooperative work (CSCW) [9] and object-oriented technology. Some e-mail-based intelligent systems support end-user's operation of the received e-mail with agents or rule-based system [12]. These systems were developed for private use, not for business use. Some workflow systems have already been used in offices. Such a system is developed by the information system support department in a user company, not by the end-user department since incompatible forms are processed among several end-user sections in only one company, not among companies. The wwHww system supports multi-organizational business use for end-user computing.

Object-oriented technology has already been applied to such software fields as programming languages, graphical user interfaces, database and object management systems. The object-oriented analysis and design methods are still

studied [14]. Most of them aim at supporting development of the large-scale software, not at end-user computing. The wwHww system support end-user computing by giving an application framework based on a distributed cooperative model of object-oriented technology.

The details of these previous studies are described in Section 6. This paper presents the philosophy of CS-life in section 2, the requirements of the application framework in Section 3, the design concepts in Section 4 and the software architecture in Section 5.

## 2. CS-LIFE

### 2.1 Process Re-engineering of Society

Recently, business process re-engineering (BPR) attracts notice since information technology may drastically improve efficiency and effectiveness of company activities. Furthermore, information technologies represented by the words "Internet" and "multimedia," will give the power of process re-engineering of human society. Now is the time for shifting our view from work-life to social life because the impact of information technology on social life must be stronger than that on work-life and because the information system market for end-users at home must be bigger than the market for business at office.

By making full use of information infrastructures, CS-life will contribute not only to the amenity to the coming aged society but also to productivity improvement to offices where the number of young workers decrease. Information society should be in the symbiotic relation between humans and computers.

### 2.2 An Example of Application

Let's suppose that you move from some city to any city. How many windows in various organizations do you visit and how many application forms do you fill in there for an address change? For example, two city offices, a telephone company, an electric power company and a gas company for your residence, a post office for mail transfer, the transport bureau for your car life, banks, insurance companies and credit companies for your living, academic societies

for mail, and the account section, the personnel section and the general affairs section in the office where you work.

Although some of these windows receive applications by telephone, most of them require at least for us to fill in application forms and send them the forms. Furthermore, many windows require for us to visit there, get application forms, fill in them and hand them to application experts at windows. If these processes can be executed by using our computer at home or at office, a lot of time will be saved.

As for application experts at windows, they are asked about the same question of how to fill in the form again and again every day and repeat the same explanation and the same check of written applications. If these routine work can be processed by computers, a lot of cost will be reduced. Furthermore, application experts may spend the saved time on service upgrade or some creative work.

### 2.3 Issues of an Explosive Increase of End-user 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 system individually. For example, although some commercial computer network service provider in Japan supports five mail-order book shops, their user interfaces are all different each other.

This issue will become clearer if we consider pre-paid cards. Many kinds of pre-paid cards have already been used in Japan. They are very convenient since we get free of small changes. However, there is a demerit of them, that is, we must always carry many kinds of pre-paid cards. Actually, I have a pre-paid card for telephone, one for monochrome copy machines, one for color copy machines, one for trains, one for car washing machines, one for golf training machines, one for fast food, one for ...

Returning to our subject, 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. Requirements for Application Framework

### 3.1 An Overview of Application Systems

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

(1) Client terminals to send written applications to windows.

(2) Servers at windows to receive written applications.

(3) MOON servers to manage the network system.

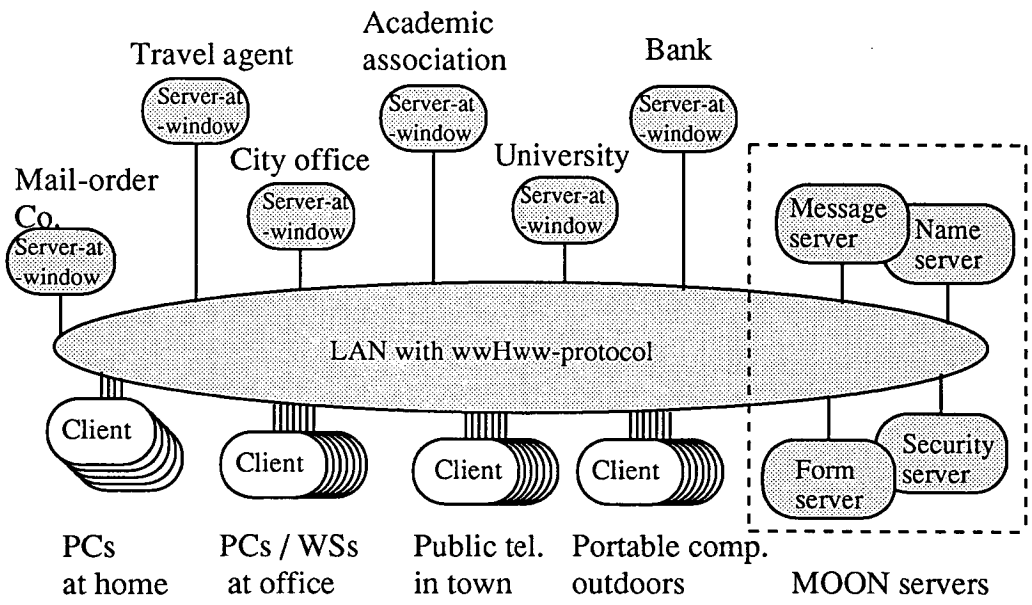



Fig. 1. A MOON (multi-organizational office network) system.

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 many windows 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 as mentioned later.

### 3.2 The Benefits of MOON systems

This system brings 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 easily understand 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 written applications to various windows of different organizations in the uniform manner.

(2) Benefits for application experts and their organizations:

(a) Experts 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 become 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 using 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

#### 4.1.1 The basic form

This paper provides the common protocol which is used for communication between the M kinds of servers at windows and N kinds of client terminals in the MOON system. It is named the wwHww-protocol 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)

4.1.2 Semantics of the protocol

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.

The states of values of these parameters affects semantics of the message. If a value of a message parameter is unknown, the message implies an inquiry about the parameter. Examples are given in the next subsection.

4.2 End-user Interface

4.2.1 External specifications

The actual end-user interface of the common protocol is different from the basic form which is the internal representation in the system. The end-user interface depends on the kind of the end-user's client terminal. 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.

Examples of requests by using the common protocol are given in the basic form for convenience, where the following notation is used:

a, b, ... : A parameter with a known value.

?a, ?b, ... : An inquiry about the parameter with a known value.
The help messages are requested.
x, y, ... : A parameter with an unknown value.
?x, ?y, ... : An inquiry about the parameter with an unknown value.
All possible values are requested for selection.

(1) Examples of sending written applications

(a, b, c, d, e, x)

The written application, b, with the contents, c, is sent to the window, a. The result will be sent by the date, d, to the address, e. A message number will be assigned to the variable, x, by the window receiving this message.

(a, b, c, d, e, f)

The written application, b, of the message number, f, is sent again to the window, a, for change of the parameters, c, d and/or e.

(a, b, , , , ?f)

The state in the process of the written application, b, of the message number, f, is inquired of the window, a.

(a, b, , , , -f)

The written application, b, of the message number, f, which was already sent to the window, a, is canceled.

(2) Examples of inquiries about application forms

(a, b, ?x, , , )

The form for the application, b, to be sent to the window, a, is displayed. How to fill in the form is navigated.

(a.b, ?x, , , , )

The title list of all application forms which the window, b, in the organization, a, receives, is displayed. The system provides the way to specify a window by the nested qualified name such as a.b.

(?x, ?y="k", , , , )

The list of titles of all application forms related to the keyword "k" is displayed with the names of windows receiving them. The system retrieves forms whose titles include the keyword and in which at least one of help messages includes the keyword.

(?x, ?y, , , , )

All windows and all titles of application forms which are received by those windows are listed.

(?a, , , , , )

The explanation on the work of the window, a, is displayed.

(a, ?b, , , , )

The explanation on the application form, b, to be sent to the window, a, is displayed.

4.2.2 An example of operations

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. Fig. 2 shows operations to be done by using his personal computer as a client terminal of the MOON system at home. The operations and the basic form of a common protocol to be sent to the system at each step are described as follows:

(a) The keyword "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. By 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.

(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.

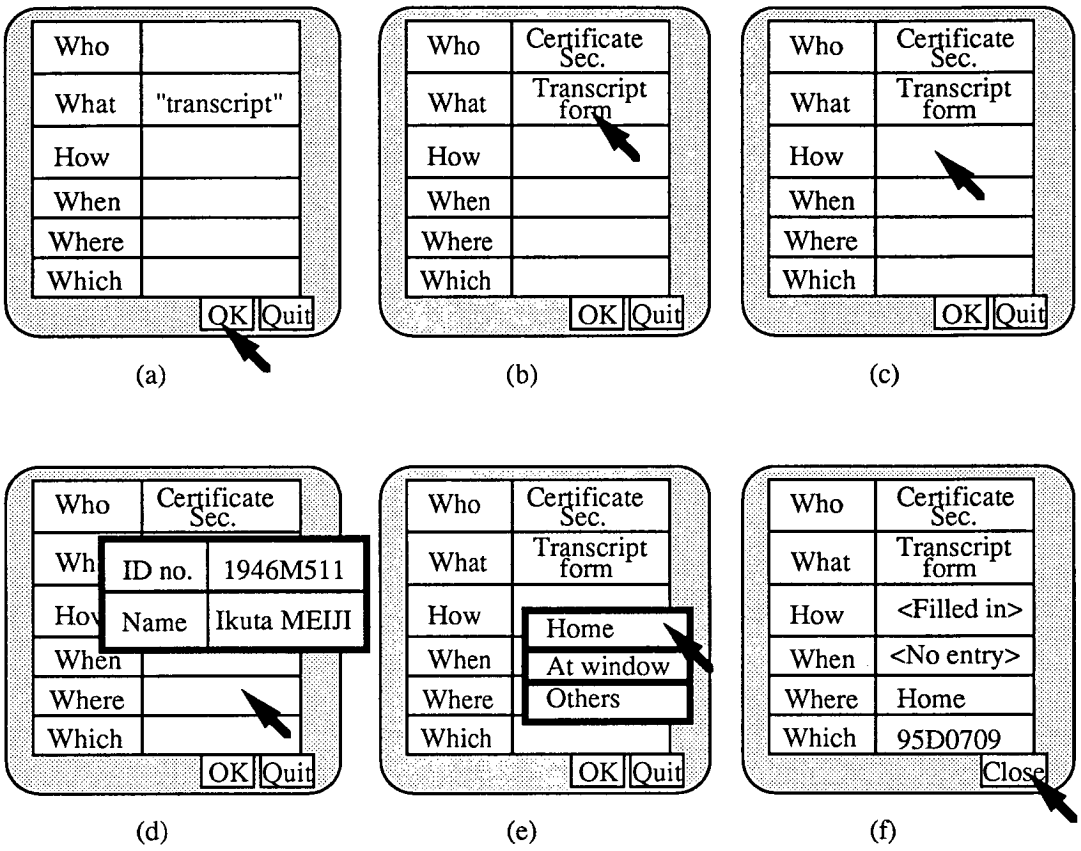



Fig. 2. An example of operations of the MOON system at home.

## 5. Software Architecture

### 5.1 System Configuration

For the purpose of easily constructing the MOON system shown in Fig. 1, the application framework, wwHww, provides three frameworks corresponding to the following three kinds of terminals of the MOON system:

(1) Client terminals to send written applications to windows.

(2) Servers at windows to receive written applications.

(3) MOON servers to manage the network system.

The software architecture of client terminals and window servers is shown

in Fig. 3. The MOON servers have the following roles in the MOON system:

(1) A form server stores all application forms which are defined with help messages and selection menus by application experts, and replies to various inquiries about application forms.

(2) A name server manages addresses of window servers to receive the written applications.

(3) A message server stores written applications received by window servers as messages, manages the states of the process of messages and replies to inquiries about the states.

(4) A security server controls access rights to window servers and the MOON servers, and authenticates clients.

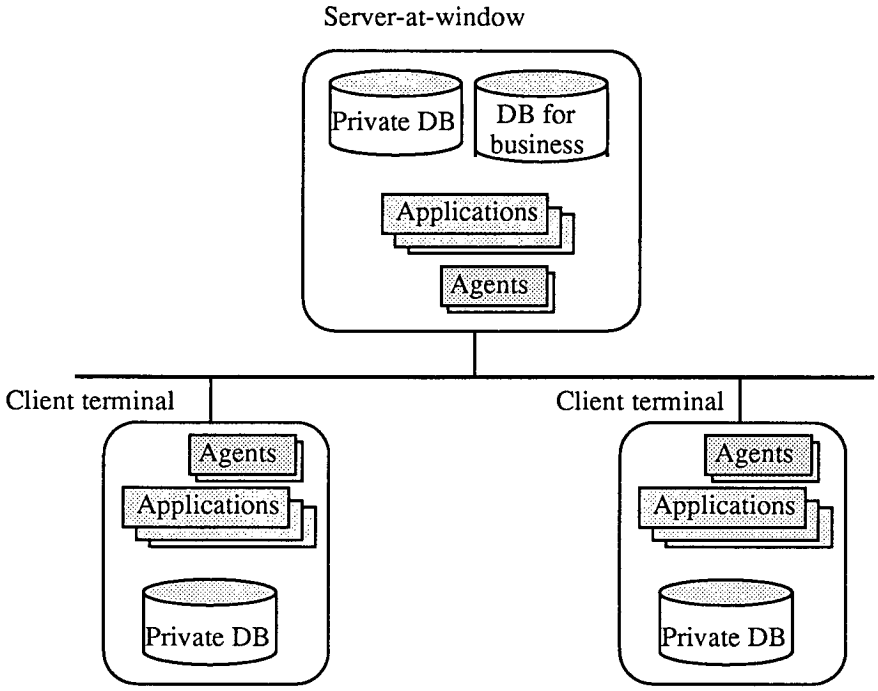
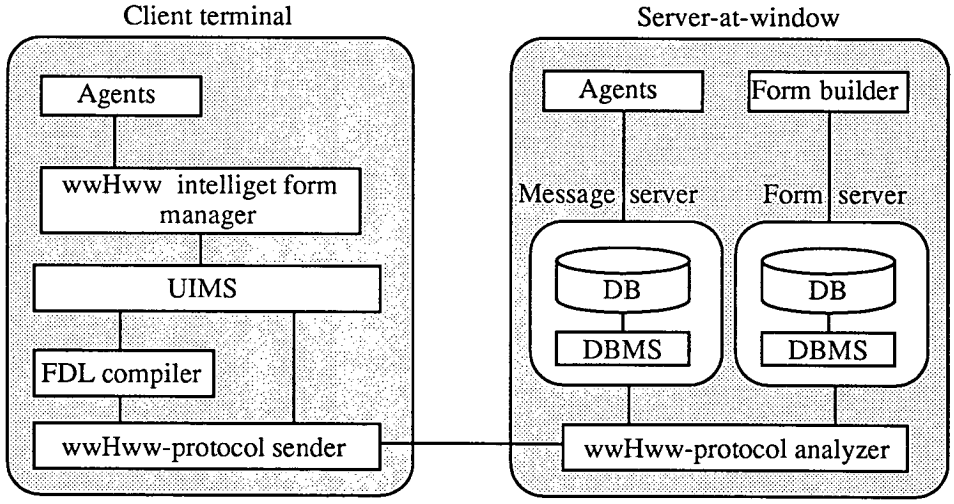



Fig. 3. Software architecture of aserver-at-window and client terminals.

5.2 Current State of the System

We have already developed the prototype program for feasibility study. Unix WSs and DOS/V PCs are connected via LAN. Unix and Windows communicate each other by the TCP/IP protocol. The software architecture of the prototype is shown in Fig. 4. The prototypes of the MOON servers were implemented in the window server. To give an example of Fig. 2, the wwHww intelligent form manager requests an application form to the form server, displays it in a visual form and navigates an end-user to fill in the form. By using the FDL(Form Description Language) compiler, the written form is transformed to a text representing a hierarchical tree structure as each leaf corresponds to a column in the visual form. The wwHww-protocol sender sends this information to the wwHww-protocol analyzer by using an RPC command. Then the receipt number is returned and is displayed on the client terminal.




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

Fig. 4. A prototype of the wwHww system.

## 6. Related work

### 6.1 Distributed Office Systems

From a viewpoint of application systems for end-user computing, the MOON system is considered as the office information system (OIS) or the computer-supported cooperative work (CSCW) system such as an e-mail based system or a workflow system [9, 13].

There are some e-mail-based intelligent systems. Object Lenz and Oval are tools for cooperative work [13]. The template-based user interface is provided and the automatic operations of semistructured objects are implemented by rule-based agents. In PAGES, the electronic secretary is constructed as a rule-base expert system, is connected to e-mail networks and processes the intelligent forms including rules [10]. These systems are given for private use and are flexible since they deal text mail in characters. On the other hand, the wwHww system is developed for business use and supports automation of processing application forms.

There are some workflow systems for supporting form-based work in organizations. The workflow management includes definitions of application data formats and automatic actions to be taken in the routing of the data among user sections. Since such transaction processing is related with several sections in an organization, the application system is developed by the information system support department in the organization, not by any particular user section. The wwHww system is aiming at developing a multi-organizational common protocol for application forms and at end-user computing between application experts at windows and clients visiting the windows for CS-life.

### 6.2 Object-oriented Technology

The meaning of "object-oriented" has not been clearly defined but this technology seems to promote paradigm shift of software for coming generation information systems. Essential concepts of object-oriented technology came out around 1970 and were expanded into programming methodologies in 1970's [3]. Object-oriented programming has already been used in practice into various software fields, especially in middleware such as graphical user interface builders and object management platforms.

Object-oriented analysis and design methodologies come out around 1990 [15, 8].

Most of these methodologies are suitable for large-scale database-centered systems such as banking systems [1, 7, 16], but they are not suitable for application software of such new fields as distributed office systems with end-user computing and cooperative work. This is because these techniques are based on a data model rather than a dynamic behavior model of the whole system. The wwHww system was designed as a distributed cooperative system based on a message-driven model of object-oriented technology.

### 6.3 Our Background

As one solution for an explosive increase of application software for end-user computing on distributed systems, we constructed a model-based software development methodology as "a domain model ≡ a computation model" [5]. The basic policies are : (1) A dynamic model on interactive behavior among objects is focused on prior to an object model. (2) The model is extended to a multiparadigm model including a data-driven model for cooperative work. (3) The model is also extended to an agent model including intelligent activities for personal assistance.

For the first policy, the multiparadigm language, S-Lonli [2], was formerly developed as expertise can be represented in Prolog. For the second and third policies, this language was extended to ES/X90 [5] as expertise can be represented in production rules also. These languages were applied to development of KISS (the knowledge-based intelligent secretary system) and OOO(the object-oriented office system). Cooperative routine work at an office was expressed by using a message-driven model and individual routine work was expressed by using rule-based system called IntelliClon (Intelligent clone). Based on these experiences, the wwHww system is being developed.

## 7. Conclusions

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

framework. That is, the common protocol for communication between the M kinds of servers at windows and the N kinds of client terminals reduced the number of application interfaces for the MOON systems from M*N to M+N.

Furthermore, clients can get application forms, understand how to fill in the form and send them without visiting windows by operating terminals and teaching the fixed operations to their agents by themselves. They don't need to write such plain words as their names and addresses also since their agents do them instead. Application experts become free themselves from frequently asked questions at windows and their agents reply to the questions instead by teaching their expertise to their agents by themselves. Experts spend much time for such creative work as improvement of processes for applications.

These benefits were ascertained by developing the prototype program for feasibility study. The next step of our research is to develop the MOON servers which are independent of window servers.

## REFERENCES

[1] Booch,G., *Object-Oriented Design with Applications*, Benjamin/Cummings, California, 1991.

[2] Chusho,T. and Haga, H., "A Multilingual modular programming system for describing knowledge information processing systems," in *the 10th World Computer Congress IFIP'86* (Dublin), pp.903-908, Sep. 1986.

[3] Chusho, T.,"What makes software tools successful?," *IEEE Software*, vol.10, no.5, pp.63-65, Sep. 1993.

[4] Chusho,T.,"wwHww : An object-oriented model for end-user computing in distributed office systems," (in Japanese), in *Information Processing Society of Japan, SIG on Software Engineering* , vol.94, no.18, pp.33-40, Mar. 1994.

[5] Chusho, T.,"M-base : Object-based modeling of application software as a domain model ≡ a computation model," (in Japanese), in *Information Processing Society of Japan, SIG on Software Engineering* , vol.95, no.55, pp.25-32, May 1995.

[6] Chusho, T., "CS-life," (in Japanese), *Computer Software*, vol.11, no.6, pp.1-2, Nov. 1994.

[7] Coad,P. and Yourdon,E., *Object-Oriented Design*, Prentice Hall, New

Jersey, 1991.

[8] Fichman,R.G. and Kemerer,C.F.,"Object-oriented and conventional analysis and design methodologies," *IEEE Computer*, vol.25, no.10, pp.22-39, Oct. 1992.

[9] Grudin, J.,"Computer-supported cooperative work : history and focus," *IEEE Computer* , vol.27, no.5, pp.19-26, May 1994.

[10] Hammainen, H., Alasuvanto, J., and Arrpe, H.,"Service interface approach in distributed loosely coupled information systems," *Office information systems : the design process*, North-Holland, Amsterdam, pp.183-198, 1989.

[11]Kling, R.,"Controversies about computerization and the organization of white collar work," *UCI Technical Report* (Draft 7a) , 1995.

[12] Maes, P.,"Agents that reduce work and information overload," *Comm. ACM*, vol.37, no.7, pp.30-40, Jul. 1994.

[13] Malone, T., Lai, K. and Fry, C.,"Experiments with Oval : a radically tailorable tool for cooperative work," in *CSCW92*, pp.289-297, 1992.

[14] Monarchi,D.E. and Puhr,G.I.,"A research typology for object-oriented analysis and design, *Comm. ACM*, vol.35, no.9, pp.35-47, Sep. 1992.

[15] Rumbaugh,J.et al., *Object-Oriented Modeling and Design*, Prentice Hall, London, 1991.