經營科學
第18卷 第1號
2001年 5月

The People Navigator : 새로운 협동 Chatting 애플리케이션

유상진* · 장영택** · 윤성로***

The People Navigator : A New Collaboration Chatting Application

Sangjin Yoo* · Youngtag Jang** · Seongno Yoon***

■ Abstract ■

We often feel the necessity to communicate with members involved in a project to remote discuss business activities, needs for the fast file transfer, and check the existence of members to talk with. There are some commercial applications in these problems. However, in addition to those existing functions, this study focuses especially on adding couple of functions such as automatic responding function, in which a sender of a message automatically knows if a receiver checks the message. Its application is that a project leader sends an important message and later on can check at his desk the fact that who have checked their messages. The prototype application, build in this study, will check the correctness of every sub-function, and then the interface of inter-sub-functions, finally the integrity of the entire system, so that important information will not be lost among members of a project in the real world situation.

1. Introduction

Nowadays lots of people try to connect to the Internet world and the rate of connected population in the world increased drastically. One of the most famous applications in the Internet

* Department of MIS, Keimyung University, Taegu, Korea
** Department of Management & Marketing, East Tennessee State University, Johnson City, TN. 37614, U.S.A.
*** Department of Management, University of Nebraska-Lincoln, Lincoln, NE. 68588-0491, U.S.A.

space is chatting programs such as Message Messenger, CoolTalk, NetMeeting, and ICQ. However, most of existing commercial chatting programs have limited functions to collaboration working environments in an organization.

In this study, we try to build a prototype collaboration advanced chatting program, named People Navigator, to solving the limitation of existing commercial chatting programs. The People Navigator is a tool that informs the user who is on line at this time and enables one to connect them at will. Two or more people can talk with each other and transfer files and messages. If they are not logged on, People Navigator keeps the messages and sends the messages after they log on, and indicates their status in real time even while the sender works on other applications. If you do not want to talk, you can change your status to log off.

People Navigator also provides a window based graphical user interface (GUI), so users do not need to know Unix commands and try to launch the other application programs in order to talk with their project members and to transfer files. Without People Navigator, users must issue command regularly until their project members log on. People Navigator does that job for them and shows their project members status.

The following sequence of this study contains four sections. Next section briefly describes related products such as ICQ, Microsoft NetMeeting, and Netscape CoolTalk. Third, section three shows the components of the People Navigator and the interactions of those components. Section four describes the experimental test of People Navigator. And finally a conclusion and future work are described in the section five.

2. Existing Commercial Chatting Application

Even though several commercial chatting products have been developed, the checking log on status is not easy in UNIX environment. In Unix environments, users have to type a command to check log on status, and they may use a File Transfer Program (FTP) to send files through complex procedures for a beginner. ICQ, developed by ICQ, Inc., is the windows-based program that provides such functions for even an unskillful user at Unix in Internet. It only shows the log on status of all project members and does not provide a specific group of members. In general, most of the chatting program has integrated GUI, so users do not need to launch other application programs to chat or transfer files. Instead users simply click an ICQ icon, and they can do everything they want in the same application environments. After users confirm the log on status, they can chat with their project members or send an important message to their project members even if they are not logged on at the present time. However, it does not indicate whether their project members get their message or not.

In PC environments, Microsoft NetMeeting and Netscape CoolTalk are widely used to chat with multiple people, but they show neither the log on status of the users nor information of whether the users receive their messages or not.

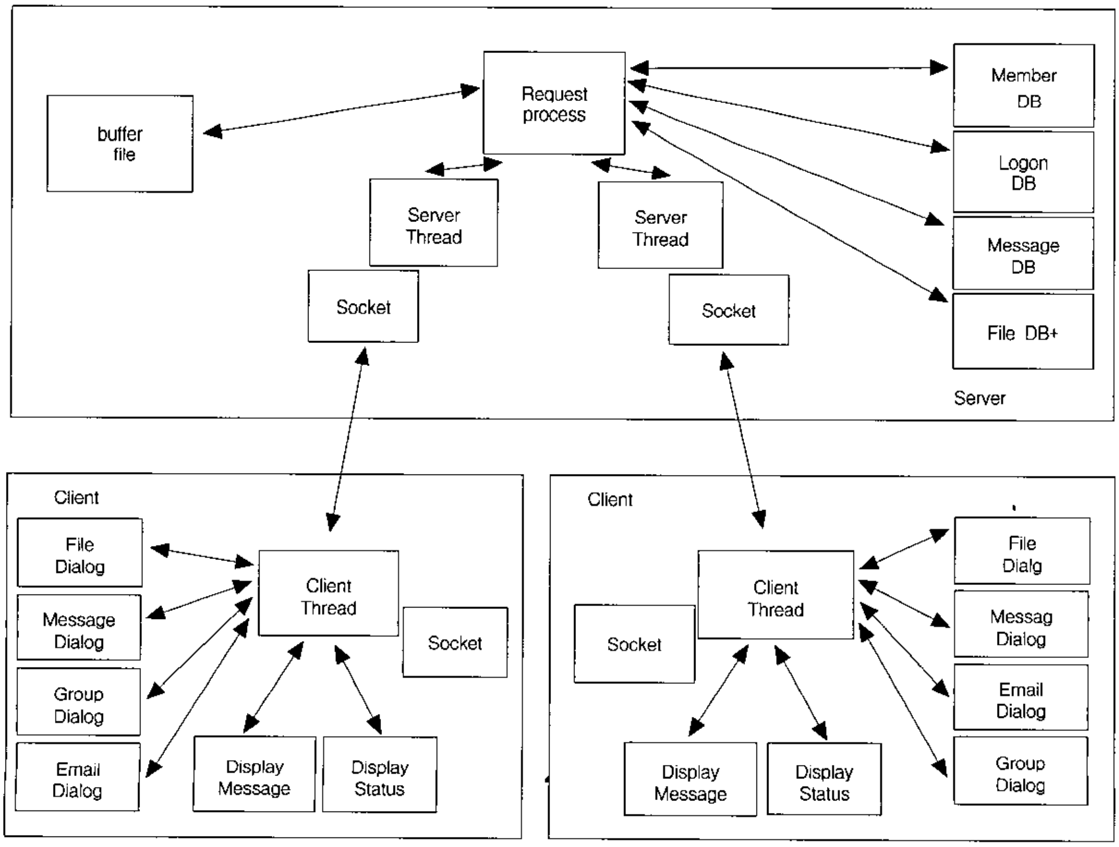
Thus, People Navigator tries to solve those problems. It indicates the status of their group members, and a user can chat with two or more people by just pressing one button. Moreover, the user can know whether project members have received the messages or not.

3. Implementation

People Navigator is implemented with the JAVA language and run on the network environments. JAVA.AWT is used to support the Window User Interface and Sockets will be used for communication. To support multi Clients, Server is implemented by using multi threads.

People Navigator is composed of two major parts : Client and Server. Client communicates with Server by sending login status and transferring files and sending messages. <Fig. 1> shows how People Navigator to operate between Client and Server. Client has six modules : 1) Log on Manager, 2) Chat Manager, 3) File Transfer Manager, 4) Message Transfer Manager, 5) Email Manager, 6) Group Manager. Each of the six modules uses a socket to communicate with Server. Server has four databases (see <Fig. 3>) : Log on DB, Message DB, File DB, and Member DB.

Each time Client sends a request to Server using the following headers in front of the request, the Server gets the request and interprets the request using the headers. <Table 1> shows the detailed meaning of header information. Each message has ##????##receiver : +text : first 8 characters are a request code and the remaining characters are a receiver and text which is separated by a colon.



<Fig. 1> Client / Server Procedures

<Table 1> Header Information

Headers	Description
##name##	Connect to Server with this name.
##only##	Send a message to the specific person only.
##msgg##	Send a message to a person, if he is not logged on, the message will be sent after he logs on.
##mall##	Send a message to all members.
##regg##	Register a person as a member.
##dele##	Delete a person as a member.
##disp##	Display the log on status of members.
##file##	Send a file to a person.
##quit##	Close the connection.
Null	Broadcast a message to all logging Clients.

3.1 Server

The Server manages Logon DB, Message DB and File DB, and Member DB. It checks for new connections from Clients. If it detects a new Client connection, it changes the Logon DB, checks Message DB and File DB. If a user has sent messages to the new Client, it forwards the messages to the Client. If the Client type messages, then Server broadcasts the message to all Clients. If the message is a private, Server sends to the designated user. If the Client sends messages or transfers files, Server checks the receivers and sends the messages or files if the receiver is logged on. If the receiver is not logged on, Server saves the messages or files in Message DB or File DB. The following are the tasks that server must do.

(a) Accept new Clients.
(b) Update the Logon DB.
(c) Check the Message DB or File DB.
(d) Broadcast the current logon status.
(e) Receive request messages.
(f) Process requests and send results to Clients.

Server must support multiple clients, the basic flow of logic in Server is :

```
While (true) {
        Accept a connection;
        Create a thread to deal with the client;
        Send new logging information to all
        Clients;
End while
```

A thread reads from and writes to the Client connection as necessary. The thread reads the request and sends it to Request Processor. Request Processor translates a Clients request and processes the requests. Results will be sent to the Client. Request Processor generally follows the Server procedures in <Fig. 2>.

```
If request is  logon , then update Logon DB and
check the Message DB and File DB.
    If there is a Message, then send the message
    notification to Client;
Else If request is  message send , then check the
user status;
    If the selected user is logged on, then
    send message notification to the user;
    Else save the message in Message DB;
Else If request is  message send  to all members;
If member is logged on, then send the message;
    Else save the message in Message DB;
Else If request is  private message , then send the
message to the selected user;
Else If request is  status  of a member, then send
the status to the client;
    Else If request is  register  , then Insert to
    Member DB and send the results;
Else If request is  delete  r, then Delete from
Member DB and send the results;
Else If request is  file transfer , then check the

user status;
    If the selected user is logged on, then
    send file notification and file to the Client.
    Else save the file in File DB;
Else broadcast the message as chatting;
```

<Fig. 2> Server procedures

The Logon DB is used for showing that the current log-on status of users on the screen.

The Member DB is used to store registered members and used when a user wants to send message to all registered members. The Message DB is used to save messages if the receiver is not logging on now. The File DB is used to store the text files. <Fig. 3> shows the structure of DBs.

```
Logon DB : Save the currently connected Client
           thread.
    Name ;          // Client thread name
Message DB : Save the messages.
    Sender;         // message sender
    Receiver;       // message receiver
    Text;           // message body
    TimeStamp;      // message sending
          time or message receiving time
    Status;//message reading status '0' not read,
              '1' read already
    Type; // message type, '0' : normal, '1' :
              answering message
Member DB : Save the registered users.
    Name;           // member name
    Timestamp;      // log on time or log off time
    Status;         // log on status, On or Off
File DB : Save the files.
    Sender;         // file sender name
    Receiver;       // file receiver name
    Filename;       // filename
    Text;           // file content
    TimeStamp;      // sending or receiving time
        Status;     // reading status
```

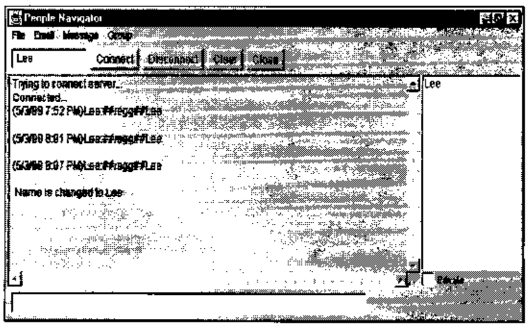
<Fig. 3> The structures of DB

3.2 Client

Client communicates with Server by using Sockets, sending a request to Server and receiving the reply from Server. When a user connects to Server, Server sends the Chat buffer to Client and Client displays the entire contents of Chat buffer having been continued among other users. It contains 4 menus to deal with each function : File, Message, Email, and Group. File menu is used to send a file to a member in spite of his being on-line or off-line. The function of Message menu is similar to that of File, but when a receiver checks his message, the time when he checked his message is returned to a sender of the message. Email menu is used to send a mail to the person who has no People Navigator Client program. Finally, the function of Group menu is to register, delete, and display members. Register members in order to send a message to all the registered members of the group. The detailed Client procedure is shown in <Fig. 4>. And the Client screen of People Navigator is showed in <Fig. 5>.

```
If a request is  Chat ;
        then send a message to Server;
    Else If a request is  Message Send ;
        then send a message with a receiver name
        to Server;
    Else If a request is  File Send ;
        then send a message with a receiver name
                and the file to Server;
    Else If a request is  Group Register ,
        then send a message with a member name
                to Server;
    Else If a request is  Group Delete ,
        then send a message with a member name
                to Server;
    Else If a request is  Group Display ;
        then send a message without a member
                name to Server;
```

<Fig. 4> Client procedures



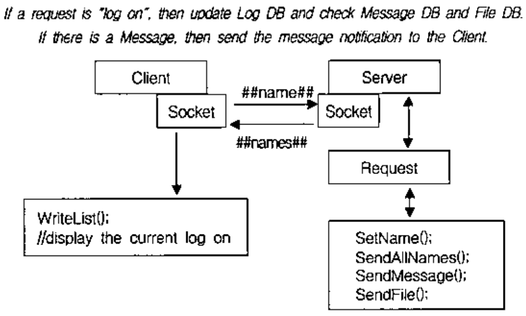
```
Trying to connect server...
Connected...
(5/3/99 7:52 PM)Lee:##regg##Lee

(5/3/99 8:01 PM)Lee:##regg##Lee

(5/3/99 8:07 PM)Lee:##regg##Lee

Name is changed to Lee
```

<Fig. 5> A screen of Client of People Navigator

3.3 Detailed Design of Server and Client Procedures

3.3.1 Log On Manager

Log on Manager module is activated when a new user logs on. It informs Server of the user logging on, and requests Server to send back the entire log on status in LOG DB. In addition, Server explores all the DBs to find the users file and message. If it finds some, it displays those on the users screen. The detailed Log On procedures shown in the <Fig. 6>.


If a request is "log on", then update Log DB and check Message DB and File DB.
If there is a Message, then send the message notification to the Client.

<Fig. 6> The procedures of Log on Manager

3.3.2 Chat Manager

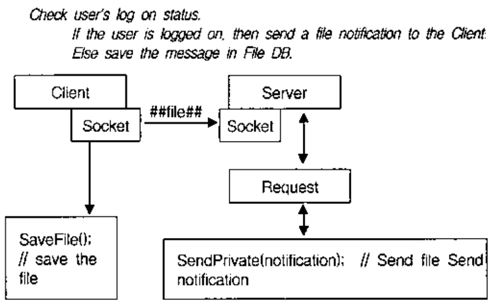
As soon as a user logs on, Chat Manager collects the past chat story in Chat buffer, and displays it on the users screen. It also keeps displaying the current chat story on the users chat box. Any on-line user can take in that session at any time (see <Fig. 7>).

```
Check users log on status.
    If they are logged on, then send a chat
        notification to the Clients;
        Receive messages and save them in Chat
        buffer;
        Forward the message to the Clients;
```

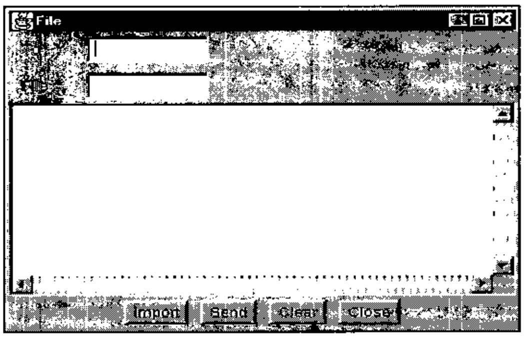
<Fig. 7> The procedures of Chat Manager

3.3.3 File Transfer Manager

File transfer Manager is activated when a user types a receiver and a file name to be sent. Once the File Transfer Manager receives the file name, it seeks that file in the current directory, and displays a content of the file. The file then can be transferred to a receiver being on-line as well as off-line status. The receiver finally read the file by using any kinds of text editors. The detailed above File Transfer Managers processes are described in the <Fig. 8> and the sample screen of it also shown in the <Fig. 9>.


Check user's log on status.
If the user is logged on, then send a file notification to the Client.
Else save the message in File DB.

<Fig. 8> File Transfer Manager procedures



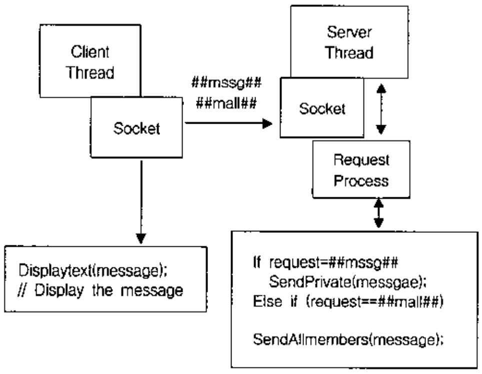
<Fig. 9> A screen of File

3.3.4 Message Manager

A Message Manager is activated when a user selects a receiver or marks a checkbox named To all , writes a message, and clicks a send button. A message can be sent regardless of a receivers status : on-line or off-line. If a receiver is off-line, the message is saved in Message DB, and it will be sent when the receiver logs on.

When the receiver checks his/her message, the sender gets a notification with the time of the receiver having checked the message almost at the same time. The detailed Message Managers processes are described in the <Fig. 10> and the sample screen of it also shown in the <Fig. 11>.

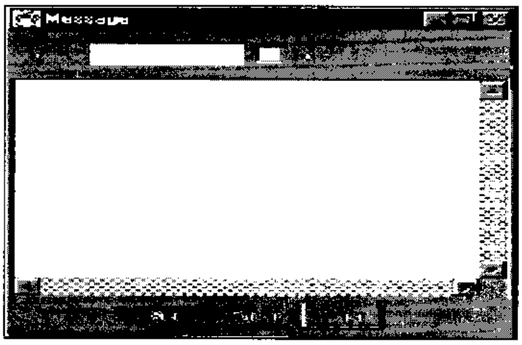
```
Check user's log on status.
    If they are logged on, then send the message notification to the Clients.
    Else save the message in Message DB.
```



```
Displaytext(message);
// Display the message
```

```
If request=##mssg##
   SendPrivate(messgae);
Else if (request==##mall##)

SendAllmembers(message);
```

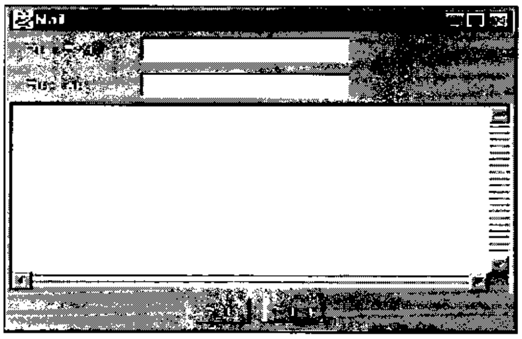
<Fig. 10> Message Manager procedure



<Fig. 11> A screen of File

3.3.5 Mail Manager

The Mail Manager reads a mail when it arrives using below a Mail Manager screen (see <Fig 12>), and sends it with a receivers Email address to SMTP Server. The receiver uses mail tools, such as MS Explorer or Pine in Unix, to read his/her message. In order to use SMTP Server, the application software should define some parameters and a specific format below in the <Fig. 13>.



<Fig. 12> A Screen of File

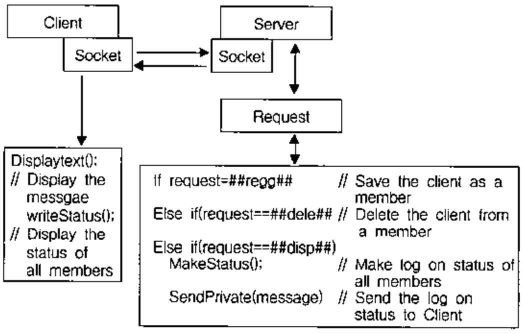
```
SMTP port_no = 25;

Record
     HELO : host_name;
     SENT FROM : sender Email address;
     RCPT TO : a receivers Email address;
     DATA : mail data;

Send the record to SMTP Server;
```

<Fig. 13> Mail procedure

3.3.6 Group Manager

A role of Group Manager is to register, delete, and display the status of all registered members. A user or project manager can send a message

```
If an operation is "register", then send a message with the members name to Server;
Else if it is "delete", then send a message with the members name id to Server;
Else if it is "display", then send just a message to Server;
```

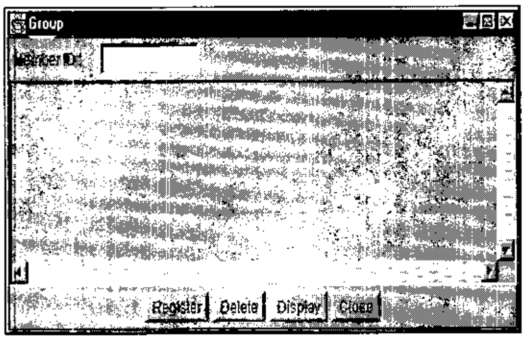


```
Displaytext();
// Display the
   messgae
   writeStatus();
// Display the
   status of
   all members
```

```
If request=##regg##          // Save the client as a
                                member
Else if(request==##dele## // Delete the client from
                                a member
Else if(request==##disp##)
   MakeStatus();             // Make log on status of
                                all members
   SendPrivate(message)  // Send the log on
                                status to Client
```

<Fig. 14> the procedures of Group Manager

to all the members of the group using the Message Manager without selecting every individual member. The detailed procedures of Group Manager are shown in <Fig. 14>. A screen of Group Manager also appears in <Fig. 15>.



<Fig. 15> A screen for Group Register

4. Evaluation

We have tested the correctness of all the sub-systems as well as the integrity of the entire system. We have tested the correctness of each sub-function first, and then the interfaces of inter-functions, finally integrity of the entire systems. As we have mentioned earlier, the primary function of the People Navigator is focus on the small group of peoples or a project team. We have chosen the number of Clients as a variable to test. The items of sub-functions to be checked are as follows.

- Log-on status : When a new user logs into the system and when the user logs off. Is the users id appended and taken out on/from the existing log on Status?
- Chat : When a new user logs on, is the past chat story in Chat buffer displayed on the users screen? Is the current chat story displayed, either?
- File transfer : When a user assigns a file to be sent, is a content of the file displayed on the users screen? Can the receiver read the file by using a text editor?
- Message : When a user intends to send a message to the entire members and clicks

To all check box, is the message sent to all the members regardless of the receivers log on status? When the receivers have checked their message, is the time when he checked his message returned to a sender of the message?

- Mail : Does SMTP server receive a mail sent by Mail Manager? That is, can the receiver use a mail tool, such as Netscape and pine in Unix, to read that mail?
- Group : Is a member registered, deleted, and displayed? Is it displayed the entire registered members?

We call the following matrix as a correctness matrix <Table 2>. It is used to check the detailed sub function one by one.

5. Conclusions and Future Work

People Navigator is programmed by using the JAVA language and is well suited for small groups of a project team or users that need to communicate with each other. It provides the GUI function to simplify communication. System members can use any sub-function just by clicking the icon, rather than striking a series of shell commands in UNIX, or clicking a couple of GUI applications, such as FTP program, Email program and Chat program, which have only a part of all the functions that People Navigator has.

〈Table 2〉 Correctness Matrix

Sub-functions	Check points
Immediately after Clients start	Is the log on status displayed on the right box of Clients screen?
	Is the past chat story displayed on the middle box?
Log on status	Is the user appended to Log on status when a new user logs on?
	Is the user deleted from Log on status when a user logs off
Chat	Is a senders chat displayed on his chat box?
	Can the entire members in a chat session see a senders chat?
	When a private check box is selected, is a message sent only to a specific user?
Group	Is it possible to register and delete a member of a group?
	Can a user display all the registered members?
Message	Is a message sent to a receiver regardless of his log on status?
	Can a sender know that the receiver has checked his message? (when the sender is online)
	Can a sender know that the receiver has checked his message? (when the sender is offline, and later he logs on)
	Is a message sent to the entire members when a sender marks To all box?
File	Is the text displayed when a sender chooses a file to be transferred?
	Can a receiver read that file using a text editor?
Mail	Is a senders mail sent to a SMTP server?
	Can the receiver read that mail using a mail tool (Netscape or Pine)?

People Navigator is comprised of Client and Server, communicating with each other via a socket, and includes five six-functions : Log on Manager, Chat Manager, File Transfer Manager,

Message Transfer Manager, Email Manager, Group Manager. Especially the Message Manager provides a sender with information of whether or not a receiver has checked the message. Using this function, a project leader sends an important message to all the project members, and then he just watches the message panel on People Navigator at his desk if all the members have checked their messages. Also it is convenient for a sender not to have to check by phone or by meeting him directly to determine if a receiver open the message or not.

We have tried to make an integrated GUI application, People Navigator, in which we can perform all the functions. Our systems, however, have some constraints over File Transfer function. This application only deals with text file and not cover a binary type file and confined the size of a file into less than 4Kbytes.

This prototype application will cover other useful functions such as a message scheduling function and a file transfer supporting the scheduled date or time a message or file is automatically transferred to assigned receivers.

References

[1] ICQ, Inc., Home Page URL : http://www.web.icq.com/index

[2] Microsoft NetMeeting and Message Mes-

senger Home Page URL : http://www.microsoft.com

[3] Netscape CoolTalk Home Page URL : http://home.netscape.com/navigator