

A Message Board System on WWW with a Visualizing Time Function for On-Door Communication

Yuko Murayama * Hiromi Gondo * Yasunari Nakamoto † Norihisa Segawa *

Masatoshi Miyazaki *

1: Iwate Prefectural University

2: Hiroshima City University

Abstract

We have brought a message board on the door of a room in a student hall of residence into the WWW environment as a novel type of asynchronous communication system. The idea of this research came from the experience of an extensive use of such a message board in a postgraduate student hall of residence. We call the type of communication through this type of a message board "on-door communication." We have tried and developed such a system on the network as a message board system on WWW for asynchronous communication. The system provides users with simple tools for drawing. Letters are coded as a collection of lines. Users write messages by hand making use of a mouse and a tablet. We incorporate a feature for time visualization into the system, so that messages in past would be fading out of the board gradually. It is this feature that makes our system significantly different from the other whiteboard systems on the network; the latters are implemented usually for real-time communications. This paper reports our design and implementation of a prototype system as well as its operation. The evaluation of drawing function is reported briefly.

1 Introduction

This paper presents our work on bringing a message board on the door of a room in a student hall of residence into the WWW environment as a novel type of asynchronous communication system.

The motivation of this research is based on the experience of an extensive use of such a message board in one of the postgraduate student halls of residence. The board was a sheet of paper on the door with a pencil attached without any eraser. It was meant to be a message board for friends to leave messages only

to the resident of the room. On the contrary, the room with the board was located in front of an elevator entrance, so that the board became soon to be known to many and used extensively as a message exchanging media by the other residents for many to many asynchronous communication with anonymous writers. The hall had about 500 residents from overseas countries. Among them around four or five people were suspected to be active writers; sometimes they started a quiz game which involved more writers who tried and answered.

The objective of this research is to bring such a message board onto the network, and to examine the difference from the real one in both terms of system structure and user-level application; we have investigated the former, however, we need more work on the latter.

We designed and implemented such a system on the WWW, which provides users with simple tools for drawing. Letters are coded as a collection of lines. On this board, any message can be written by hand making use of a mouse and a tablet.

The original user-level application of our system is to provide the users with a tool for leaving a short note to the owner of the board. Our ultimate goal of this research is to observe the use of the board and to investigate more applications. For example, it could be used for discussion as in chat systems [1][2], although communications would be asynchronous with our message board; this was the way in which the message board in the student hall was used. Alternatively a board could be owned by a group of people and used as a community board for group communications. Moreover, a single user can own several boards for different purposes. Our current prototype message board server deals with only one board, and we are working on an extended function so that it can manage multiple message boards. In future, we are planning to open our source code so that more different types of use would be investigated in various environments.

Although we have not had enough research on the usability of the message board, one of the authors had an experience that wherever she went on trip over the world, when she had an access to WWW, she could read and write short notes on the message board. This indicates that our message board could be used as a ubiquitous communication tool particularly for emergency use.

Traditional Internet applications such as email and text-based message boards provide asynchronous communication tools based on text. In such communications, senders and receivers need to agree on what language to be used for coding. This point is critical especially when you use a language other than English. Moreover, in email communications users use quite often some non verbal expressions such as smile marks “:-)” and “;-).” Communications would be easier if we used figures as well as text on the same media, and our system offers this.

What makes our system significantly different from the other whiteboard systems[3][4][5][6] on the network is its use for asynchronous communication, whereas the most whiteboard systems are implemented for real-time communication. From this view point, we add a feature so that messages in past would be fading out gradually. We discuss only the design issue of this feature, as we are still working on its implementation.

This paper reports on the design of our system as well as the implementation of a prototype system and its operation. Finally we report our discoveries on the difference from the real one in terms of a system structure, and discuss the issues to be dealt with in future.

2 A Model of on-door communication

We call the type of communication through a message board on the door of a room in a student hall of residence “on-door communication.” Figure 1 shows our model with three types of objects; the *resident* of a room, a *message board* and a *user*. A user leaves a message for the resident. Anyone including the resident can read the message. The resident, the original receiver of the messages, is a special user and manages the message board such as clearing all the messages out of the board.

The message board functions as a communication media and has the following characteristics:

1. Messages on the board are short and written by hand.
2. It provides asynchronous communication.

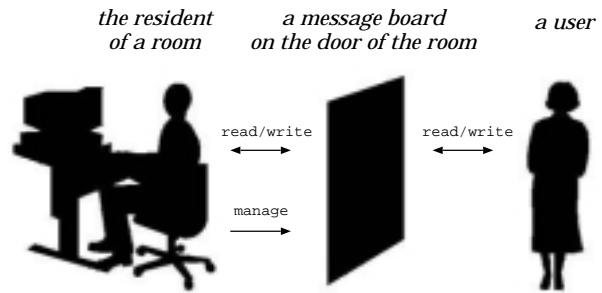


Figure 1. The Model of On-Door Communication

3. Anyone can read and write; anything can be written on the board, including a message addressed to the others than the resident.
4. The board is write-only and has no eraser for the ordinary users; this is merely due to the fact that the original message board in the student hall did not have any eraser.
5. There is no authentication and users stay anonymous.
6. There has to be a way for the others to know of the existence of the message board; i.e. a feature of awareness is needed.

The motivation for anonymity, 5 in the above, is this. The original message board in the student hall was used in such an extensive way that not only for friends left a message to the resident of the room, but also some other residents started leaving messages presumably to the room resident but anyone else enjoyed reading and replied. The writers stayed anonymous with code names such as Doracula and the board worked as an entertaining media for many-to-many communications.

3 The design issues of the message board on WWW

We design a message board on WWW so as to satisfy the six characteristics listed in the previous sections. The board system is reached through a hyperlink from the owner’s WWW page. The system provides the users with simple drawing tools so that they write messages by hand, making use of a mouse and a tablet. messages are encoded into a set of line commands. The users can write and read messages whenever they feel

like. Neither access control over writing and reading nor authentication is enforced. No eraser is provided.

The existence of the board is known to the users by means of personal notice from the owner of the board. Moreover we presume that the Web page is open to the global Internet, some may come across the board by chance. We take such a passive approach to this awareness aspect for the moment. Alternatively we could try and locate the board at more popular site, so that any user would know of its existence easily.

Moreover, we add a feature such that the messages in past would be fading out gradually. The idea is to bring the time axis into the message board space. The message board in the real world has no such feature.



Figure 2. An On-Door Communication System on WWW

4 The system design

4.1 The server and client systems

Our on-door communication system on WWW has a client-server structure (see Figure 2.) A server provides the *message board* functions and manages the drawing information. It provides the *resident*, the owner of the board with tools to manage the message board. Client systems get the drawing information from a server system, and show the drawings on the board to the *users*. A *user* has access to the server through the hyperlink within the owner's Web page, and imports the client system from the server system automatically. We presume the use of JAVA Applet, but if there is any more practical way, we shall use it in future. The idea is that a user site does not need to deal with the installation of the client system. *Users* write messages on the client system, which encodes the messages into the drawing information and send it to the server. A server can connect to multiple clients simultaneously.

4.2 The communication protocol

A server and a client use our original connection-oriented protocol, the On-Door Communication Protocol(ODCP) for exchange of drawing information. ODCP runs over a TCP connection. Therefore, there are two TCP connections between a user site with a client system and a server site. One is used by the Hyper Text Transfer Protocol(HTTP) for the WWW connection, and the other is used by ODCP for the message board transactions.

ODCP is used as follows:

1. for a server to send a probe to monitor the state of a client

2. for a server and a client to exchange the drawing information

3. for an owner, a special user, to manage a server

Figure 3 shows the list of ODCP packets. MSG and MAXDATA packets are used by the server to send the drawing information to a client. A LOAD packet is for a client to start a session and request the server for the initial drawing information. LINE and BOX packets are used by a client to send the additional drawing information on one stroke to the server. PING and PONG packets are for the server to check keep track of the session with a client. BYE and QUIT packets are used by a server and a client to close the session.

An SU packet is used by the owner to begin the management session with the server, and a Genocide packet is used to finish it. A Crumble packet is used to clear out the board. A Store packet is used for storing the drawing information on the current board into a file as an archive, and a Restore packet is used to resume the message board in past.

When a user obtains access to the message board on the server through a link from the board's owner's Web page, a client system is loaded automatically and evoked. The client at the user site sends a request for an ODCP connection to a server. When the connection is established, the client receives the drawing information on the messages currently on the board from the server, decodes it and shows it to the user (see Figure 4).

A user has to wait until the client completes redrawing, before s/he begins writing. When the user adds a message, the client encodes it into the drawing information, one stroke at a time, and sends it to the



Figure 4. Redrawing process

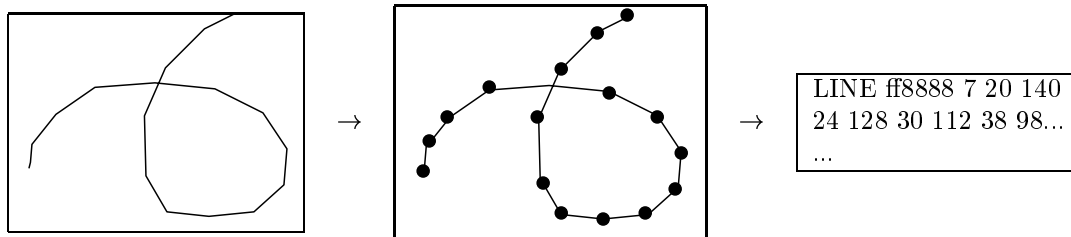


Figure 5. The coding process

client-server packets	management packets
MSG	SU password
LOAD	Genocide
MAXDATA	Crumble
LINE	Store
BOX	Restore
PING	
PONG	
BYE	
QUIT	

Figure 3. The types of the ODCP Packets

server.

When the server received the drawing information from a client, it shall add a time stamp to the information and keep it. If the server has the other clients, the received information will be distributed to them.

4.3 The drawing information

Messages are encoded into a collection of lines and sent to the server. We use our own coding which is a type of vector drawing. We employ the vector drawing because its size of information is relatively small compared to pixel-oriented coding.

When a user picks up a pen and start writing a message, the system starts sampling the points and encoding as a collection of straight lines (see Figure 5.) A stroke is encoded as one LINE command with several points. In other words, it might take two strokes to draw a letter "A" which would be encoded into two LINE commands, while it would take one stroke to draw "C", which would result in one LINE command.

The format of a LINE command is as follows:

```

LINE <color> <thickness> <the X and Y
coordinates of a point>...
    
```

The color is represented in the red, green, and blue (RGB) scheme.

Our message board is expected to be used mainly for asynchronous communications, and the system does not have any distinct lock mechanism for concurrency control. When two or more users started drawing at the same area of the message board, local drawings would appear on the board at each site. Those drawings would be encoded one stroke at a time and sent to the server. The server would store the incoming one stroke drawing information in its drawing information base as well as send it over the ODCP connections to the other clients. One ODCP packet carries one stroke drawing information. When those clients received the drawing information, they would show it onto the board. This way, a user would notice that the others were writing at the same area when their first stroke information arrived at the user's site and appear over the user's local drawings.

In such a case, however, in the main drawing information base maintained by the server has the information for those drawings sequentially according to the time when one stroke information was received. Therefore, when the user tries access to the message board next time, those past concurrent drawings will appear in this sequence which might be different from the one which user saw at the time when the concurrent drawings was happening.

5 Visualizing time

5.1 Overview

We design a feature so that old messages are presented in a faded color according to the time when they were written. Figure 6 shows the sample of what we expect.

We locate this function at the client sites, so that it will be left for users to control fading as they like; some may be interested only in the messages written in a few days, whereas other may like to see the message written in a few weeks.

A client maintains a cache of drawing information with time stamps, which has been obtained from the server. The client provides a user with a function to control the lightness of the color of the messages according to how long ago they were written.

This feature has not been implemented yet, and we present the design issues in the following subsections.

The current implementation of a prototype provides users with seven colors to choose; red, green, blue, pink, orange, black, and white. Those colors are expressed in RGB for each LINE object of the drawing information, whereas for visualizing time we use another expression form of color, HSB(Hue Saturation and Brightness); in fact, hue, saturation, and brightness are aspects of color in RGB. According to [7], higher saturation value, one would get more *pure* color, and lower saturation, more *washed-out* color.

We use the saturation parameter to fade a message as follows. A message is composed of a set of LINE objects. Each LINE object has been time-stamped by the server. The color information of LINE objects is transformed into HSB from RGB at a client and the saturation parameter is recalculated according to the time stamp of a LINE object.

Users specify the time duration to change the saturation. If one specifies a few days as the time duration, only those messages which were written in a few days would appear in a faded fashion. Those which had been written before the few days ago would disappear. Figure7 shows the calculation of the saturation for each LINE object. In Figure7, the oldest time to display, T_{old} , is specified by a user. This parameter is used to specify the user's preference in the oldest message to be shown on the message board. The default time would be the generation time of the oldest message.

$$S = (S_{max} - S_{min}) \times \frac{T_{gen} - T_{old}}{T_{new} - T_{old}}$$

S_{max} : the maximum saturation value (Faded:i.e. oldest)
 S_{min} : the minimum saturation value (Pure:i.e. latest)
 T_{gen} : the generation time of a LINE object
 T_{old} : the oldest time to display
 T_{new} : the generation time of the latest LINE object

Figure 7. Calculating the saturation, S , of a LINE object

This feature is to be implemented near future. Users could specify the maximum and minimum values of brightness. Since this may cause the user interface

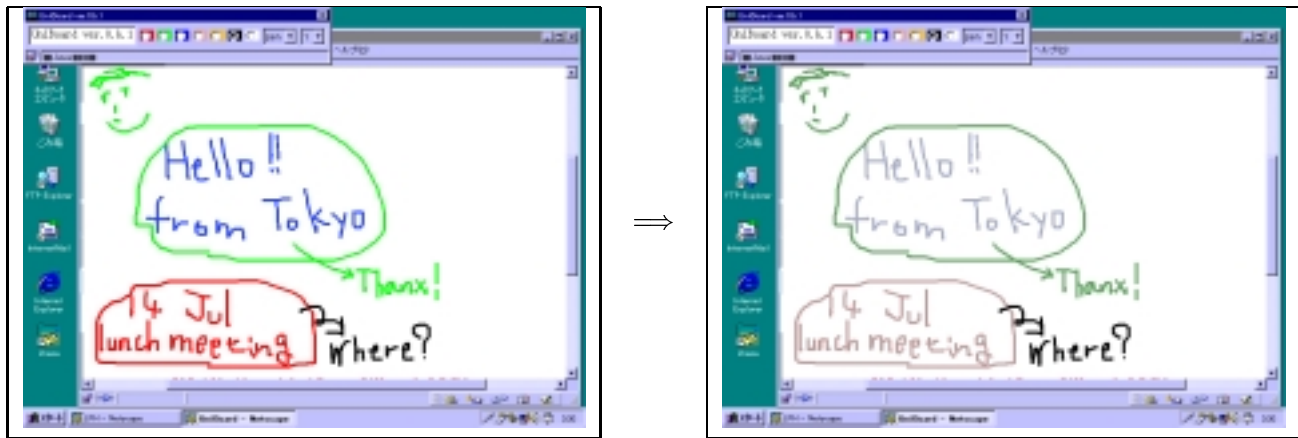


Figure 6. Fading messages

more complicated, we may not include this in the initial implementation.

6 Prototype: the Uni Board

6.1 The description on the system

The prototype system is called the Uni Board¹ due to nickname of one of the authors, Yasunari Nakamoto who implemented the system originally.

Not all of the functions have been fully implemented for the moment. The management functions of ODCP are implemented locally in the server, but cannot be used through the ODCP; one has to be logged onto the server machine to use them.

The current system uses RGB for the color space representation. The feature to fade old messages has not been implemented yet.

We used Java for a server, and Java Applet for a client. The size of the server software is 10Kbytes, and the client one is 31Kbytes. At the moment a server deals only with one message board. The original prototype server used to run on the SUN Sparc Ultra-1, however, the current one runs on a PC with the CPU of AMD K6 166MHz and a 64Kbyte-long memory. The operating system is Vine Linux 1.0 (kernel 2.0.36) and the Java virtual machine is JDK1.2-prel. We use Apache for the HTTP server.

The client runs with such Web browsers as Netscape Communicator with the version 4.5 and the higher ones, and Mictosoft Internet Explorer with the version 5.0 and the higher ones.

¹ The “U” of Uni is pronounced as the “wo” in “wolf.”

A *resident*, the owner of the board, has a link to the message board server on his/her home page[8] on WWW. Through the link to the server, client systems are generated automatically with the use of Java Applet. When client system starts operating, it sends the server a request for an ODCP connection. In practice, the server can have as many connections as possible as long as its system resource is available.

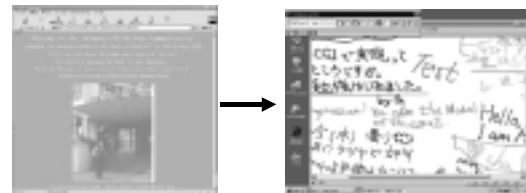


Figure 8. The owner's Web page and the message board

Figure 8 shows the message board attached by the control panel the pen selection. The size of the current message board is fixed as 1000 × 1000pixels. A user selects a pen with the favorite color and thickness, and start drawing with an input devices such as pentabret and a mouse. When a user starts drawing, the client system starts sampling points of a line. Sampling is performed whenever the client system gets a mouse event which is a standard function of JAVA. A mouse event is provoked whenever a mouse is moved. The sampling time may vary according to the JAVA Applet environment. Using a PC with the Pentium 133MHz CPU with Windows98 and Internet Explorer

5.0, it was observed about 60msec.

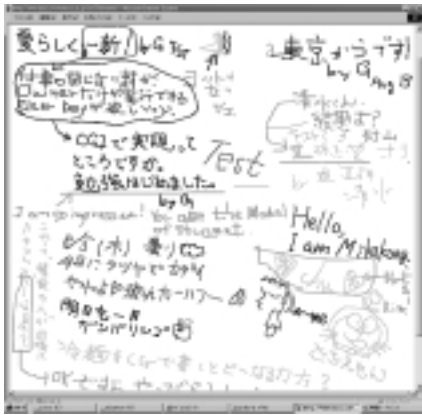


Figure 9. The message board with 100Kbyte drawings

6.2 The operation of the Uni Board

We have been running the board since June 1999. The board is cleared out by the owner user nearly monthly. The average size of the drawing information at the end of each month is 1320Kbytes long. The number of user access varies, but on average we have had 60 to 180 times each month. The Uni Board in Figure 9 has 100Kbyte drawing information.

We also measure the time for a client to read the initial drawing information from the server. The measurement took place over 10Mbps Ethernet on which only one server and one client systems are connected. Figure 10 shows the experiment environment.

It took 35 seconds to read a file with 3000 LINE objects, whose size was 384Kbytes through the network. Figure 11 shows the time required to read those objects initially. Moreover, when a user obtained the first access to the link to the Uni Board, it took about 30 seconds to set up the Java Virtual Machine (VM) over the Web browser and download the Java Applet for the client.

Once the client system got started, the user was not aware of any delay in drawing because the local drawing information is shown to the user locally and then sent to the server.

6.3 Evaluation of the drawing function

We have tried and compared the drawing function of our prototype system with the other drawing systems.

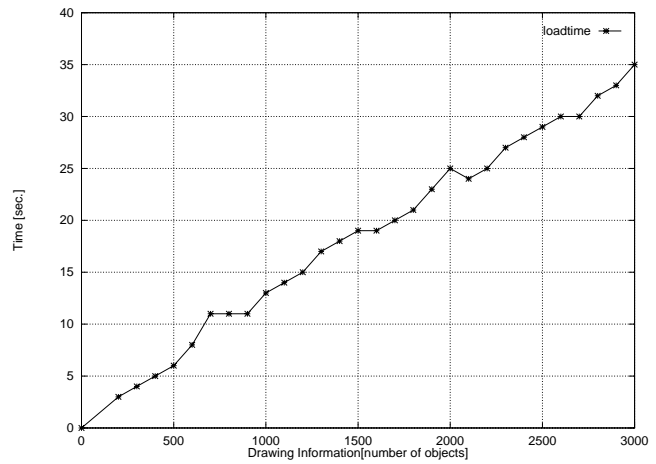


Figure 11. The measured time for a client to get the 3000 LINE objects (384Kbytes)

The idea is to see how easy to write a message by hand using our current prototype system. More detailed results are reported in [9].

We examined the differences in the following aspects:

1. operating systems
2. input devices
3. application software

We performed the experiment to see differences in drawing in the above aspects using eight different environments (see Figure 12.) Experiments 1, 2, and 5 use our Uni Board for application. Experiment 4 uses one of the popular Japanese word processing software for application. In Experiment 8 a user writes a message on a sheet of paper using a fiber-tipped pen. The other experiments use a paint tool provided by Windows 95.

In each one of the above eight systems users were asked to do the followings:

1. to try and use a system for one minute to get used to operation.
2. to write a message, "Good Morning," in Japanese as follows:

おはようございます

We recorded the user operation in video and analyzed it later.

a server system	
CPU:	Celeron 333 MHz
O/S:	RedHat Linux 5.2 (kernel 2.0.36)
memory	128 Mbyte
Java VM	JDK1.2 prel

a client system	
CPU:	Pentium 166MHz
O/S:	Windows95
memory	96 Mbyte
Java VM	Internet Explorer 5.0

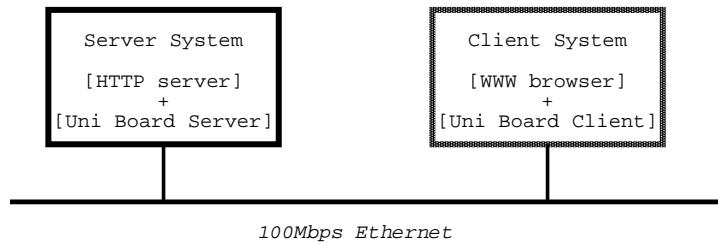


Figure 10. The experiment environment

No.	O/S	Input device	Application
1	Windows 95	mouse	Uni Board
2	Windows 95	pen tablet	Uni Board
3	Windows 95	mouse	paint tool
4	Windows 95	mouse	Ichitaro
5	UNIX	mouse	Uni Board
6	Windows CE	LCD tablet	paint tool
7	PalmPilot	LCD tablet	paint tool
8	a sheet of paper	a fiber-tipped pen	writing by hand

Legend:
 LCD: Liquid Crystal Display

Figure 12. The system environments of the experiments

According to the result, the easiest way is to write with a pen on a sheet of paper, and it takes 1.25 to 2.32 times longer using any other tools on computer systems; the LCD tablets were considered to be the second best. We may say that those drawing tools are not suitable for writing long sentences, however, they could be used for short messages. This is exactly what we need in on-door communications.

As for the comparison between a pen on display and a pen tablet, examiners drew faster with the former than with the latter. This is because one can see the drawings directly at the tip of the pen with a pen on a display, whereas one has to see the drawing outputs on a separate display as one draws with a pen on a tablet.

The average time to draw with our Uni Board running on the system with a pen tablet (Experiment 2) varies a lot according to individual examinee's ability,

and it results in the big value of the standard deviation due to an irregular user who preferred a mouse to a pen tablet. The others preferred a pen tablet.

Apart from measuring the time with video, we asked the users to give a rank to each system based on their preference.

Our system with mouse was evaluated by the users as bad as the paint tool with mouse. On the other hand, our system with pen tablet was considered as good as the system with the LCD pen.

Our discoveries from the experiment are as follows:

- the easiest way is to write with a pen on a sheet of paper.
- it is easier to draw with a pen-type interface than to use a mouse.
- As far as the Uni Board is concerned, although its handwriting function was not evaluated as good as a paper system, it is considered almost as easy as the other drawing tools in

7 Discoveries and future work

There are several differences between our message board on WWW and the one in the real world.

As we put it in one of the designed features of the system, the aged messages could be faded away. This feature of time visualization would not be possible on the board in the real world.

In the real world, the owner of the board manages the board, whereas in our system, the owner of the board is not necessarily the manager of the board server. We need to extend our prototype system so that a server can generate more than one boards on demand from a user. This work is essential due to many users'

requests that they would like own their own message board; some users requested that they would need multiple message boards for different purposes. In this way, in the future, we would have more complicated situations, and what is needed to construct the management structure. Moreover, one may wish to purchase a board from a dealer in future. We may well need to model the management structure in terms of business as well.

We need to specify inter-server communication as well so that the whole system would operate in a distributed manner.

We also need to deal with security issues on how one can guarantee that a user who leaves a message stay anonymous. Writing figures could give us some indication of writers. We are working on a feature so that we can hide our individual writing characteristics in handwriting if a user wishes so.

The real message board in the student hall was located at a popular place, so that everyone can come across the message board easily. For this awareness issue, our work did not implement fully. We took a passive approach that the board owner notifies a group of friends personally of the existence of the board. We need more research on this aspect.

We shall also start looking at the use of the LCD panel on the real door of a room, which will display the web page on the network as well as to let the users to write the messages. Moreover, we have started connecting a real whiteboard with a special pen holder and a sensor which detects the movement of the pen holder. The sensor is controlled by a PC on the network, so that messages written on this board appear on our message board on WWW. This will enable people who had no access to the Internet to leave messages on our system.

The current size of the message board is fixed as 1000×1000 pixels at the moment. In future we would like to let a user to adjust to a desirable size.

We would like to investigate if any other types of communications among users would be possible through the use of our system than our original intention to use the message board as the one on a door of an individual room.

8 Comparison with previous work

Most of whiteboard systems[3][4][5][6] and NetMeeting[10] provide users with a tool for handwriting. Those applications are intended to provide users with real-time communication for computer-supported cooperative work(CSCW), and focused on synchronous communication. They share the writing space during

the period of a session. They are not interested particularly in the messages in past.

On contrary, our work is focused on asynchronous communication. It is important for us to share the messages in past with the others. The users share the writing space as long as weeks for instance. A sequences of messages on a discussion taken place at one period of time would be shown to the people in some other time; they can restart the discussion. In other words, they share the time axis as long as the owner does not clear out the board. We have brought time visualization with our work from this point of view. The idea is to fade the messages according to the time when they are written, so that users would see what messages were written more recently. In this way, users share the time axis more practically.

For time visualization, Rekimoto presented the idea on time-machine computing[11]. a time centric approach to organizing information so that a system provides a user with a tool to move back and forth between the past and the current states of computers. The aim is to resume the state in past. He also presented the idea on visualizing time by showing the information according to the time axis, which is exactly what we are doing with our work on visualizing time, although the aim is different. We are concerned only with presenting the information on the time axis to let the user know what messages are more recent ones.

Flatland by Mynatt[12] is a computer-augmented whiteboard system for asynchronous communication, however, it is designed to support informal work in a personal office environment in which the board owner leaves some memo and ideas. It has a feature of semi-public use for visitors and office meetings in which people share a physical whiteboard. It also has a feature of time visualization in which a user can attach a time slider to a segment of the whiteboard and can view a snapshot of the drawing in past. Time visualization in their use is originally designed to see some memo and ideas in past, whereas the one in our system has a different objective to see all the messages on the board at a glance with their freshness; that is why they are washed out gradually rather than a snapshot.

Finally the most attractive point with our system is its simplicity and lightness in terms of the code length and drawing data size; they are all in order of some Kbytes.

9 Conclusions

We have presented our work to bring the message board on the door of a room in the student hall of residence into the WWW environment as a simple drawing

system. It provides an asynchronous communication among people.

We have reported on our prototype system and its operation. A feature to visualize time is discussed, however, has not been implemented in our prototype yet. It will be implemented near future.

Although we reported some usability in terms of system environments, we need more work on user-level usability such as what type of communication could or could not be performed with our system.

The differences from the board in the real world, include time visualization and the need to structure of the management. We shall work on sorting out those issues as well as to try and extend the overall communication environment to include the real world, by making use of LCD panels. Moreover, we would like to investigate more interesting use of our board system for communication among people than just a board for message exchange, if any.

References

- [1] Fernanda B. Viegas and Judith S. Donath: Chat circles, Proceeding of the CHI 99 conference on Human factors in computing system, pp. 9-16 (May 1999)
- [2] Marc A. Smith, Shelly D. Farnham and Steven M. Drucker: The social life of small graphical chat spaces, Proceedings of the CHI 2000, pp. 462-469 (April 2000)
- [3] T. Moran, P. Chiu, S. Harrison, G. Kurtenbach, S. Minneman and W. Melle: Evolutionary engagement in an ongoing collaborative work process: a case study, pp. 150 - 159, Proceedings of the ACM 1996 CSCW, (1996)
- [4] G. Abowd, C. Atkeson, A. Feinstein, C. Hmelo, R. Kooper, S. Long, N. Sawhney and M. Tan: Teaching and learning as multimedia authoring: the classroom 2000 project, pp. 187 - 198, Proceedings of ACM Multimedia, (1996)
- [5] T. P. Moran, W. van Melle, P. Chiu: Tailorable domain objects as meeting tools for an electronic whiteboard, Proc.the ACM 1998 conference on Computer supported cooperative work (CSCW '98.), pp.295-304 (1998)
- [6] J. Rekimoto: A multiple device approach for supporting whiteboard-based interactions, Proc.Conference proceedings on Human factors in computing systems(CHI '98), pp.344-351, (1998)
- [7] whatis.com (The IT specific encyclopedia): hue, saturation, and brightness (HSB), <http://www.whatis.com/hsb.htm>, (last accessed on May 30, 2000) http://www.whatis.com/WhatIs_Definition_Page/0,4152,212262,00.html
- [8] The Web page of a board owner: <http://www.comm.soft.iwate-pu.ac.jp/murayama/LPH-ODC>, (last accessed on September 28, 2000)
- [9] N. Segawa, H. Gondo, Y. Nakamoto, Y. Murayama, M. Miyazaki: Evaluation of the handwriting function of an on-door communication system, will be presented at HICSS-34, Emerging Technology Track, Non-traditional Computer Support for User Evaluation mini trak
- [10] NetMeeting, <http://www.microsoft.com/windows/netmeeting/>
- [11] J. Rekimoto: Time-machine computing: a time-centric approach for the information environment, ACM UIST'99, pp. 45 - 54 (1999)
- [12] E. Mynatt, T. Igarashi, W. Edwards and A. LaMarca: Flatland: new dimensions in office whiteboards, pp. 346 - 353, Proceeding of the CHI 99 (1999)