Design and Implementation of a Web-based Real-time Interactive Collaboration Environment

H. C. Hong and Y. C. Chen
Department of Computer Science and Information Engineering
National Chiao Tung University, Hsinchu, Taiwan
{hchong,ycchen}@csie.nctu.edu.tw

Abstract

WWW becomes the most popular application platform due to its user friendly multimedia user interface. In this work, we design and implement a Web-based Real-time Interactive Collaboration Environment (WRICE) on the Internet. The goal of the work is to design an efficient and friendly network environment to ease the collaboration activities through the Internet. There are three important features implemented in our system, the first one is a user friendly interaction interface through synchronized free drawing and real-time video/audio stream, the second one is a scalable architecture that improves the multicast performance for multimedia content, and the third one is a web-based architecture with a variety of optional features, which are installation free from the standpoint of users.

Keywords: Collaboration, CSCW, distant learning

1. Introduction

Since the development of WWW (World Wide Web) in 1993, the Internet population keeps growing exponentially. Nowadays, WWW becomes the most popular platform for a huge number of Internet applications. Through WWW and browsing mechanism, users could easily search and retrieve information from web-servers located around the world. Applications such as MIME E-mail, browsing and downloading of images, audio or video are nowadays a very common application. Although users may input certain information while they are browsing the Internet, it is still inconvenient for two or more users doing real-time interaction through the network. Therefore, compared with web browsing, technology for doing real-time interaction between web users is still not very mature. Our real-time interactive collaboration environment is based on the traditional CSCW (Computer Supported Cooperative Work) concept [6,7,14]. It could be applied to the following applications:

- Office Automation and Management Information System: Company employees in different branch offices can discuss the work and process the office documents more effectively through the Internet video conference.
- Work at Home: Company employees may work at home, and they could communicate with other colleagues, no matter they are in the company office or at home, through the CSCW system. This working model is very popular in certain well developed countries, and it saves a lot of commuting time, office space, and energy consumption on transportation.
- Tele-medicine: An intern in the rural area could co-diagnosis with an experienced doctor in the large hospital in a city, this could make up the insufficiency of medical care in rural areas.
- Collaboration: It is an effective approach to perform tightly coupled research collaboration through information exchange, audio/video conference and co-authoring among scientists in different locations. [1,9,13].
- Distant Learning: An instructor could give a lecture to his/her class students located in various remote locations through audio/video conference based on certain tools such as whiteboard, co-authoring tool, and bi-directional voice communications [2].

According to our study, the following tools are helpful in building an efficient CSCW system.

- Whiteboard: Some information such as street maps and figures are hard to be described through either voice or text, in such situation hand drawing on a whiteboard should be a more effective communication method.
- Facial video window: To help understanding the main theme in a peer-to-peer talk, it would be more effective if both parties were able to see the face and gesture of the other side. In a synchronous distant learning environment, it is more nature to see the face of the instructor.
- Desktop video window: In certain cases such as calligraphy and painting, the image on the desktop has to be transmitted to the peer party to benefit the understanding and discussion. There are already many so-called 3-D projectors in the market, they need a video window to display the projected objects.
- Audio: Neither text nor drawing is effective for

Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03)
0-7695-1910-5/03 $17.00 © 2003 **IEEE**

teaching a language class through the network, therefore audio would be required in such situation.

- Text talk window: Special symbols such as @, ξ, &, |, ε, ❶ etc. need text mode to display.
- Co-authoring window: This allows collaborating parties to view and to edit the same document simultaneously.
- Co-browsing windows: Both parties could use HTML browser to browse the same document, and the action and its associated information of one party could be sent to the peer party so that the browsing activity can be synchronized.
- Tele-pointer: In a tele-medicine system, a doctor could pin point the suspected spot on an X-ray image, and the mouse pointer in the remote site could be moved synchronously, this would help co-diagnosis the symptom.

Due to the lack of a common standard, it is hard to find a satisfactory CSCW system with the aforementioned features in the market. In this work, we use the common functions of WWW to develop a web-based real-time interactive collaboration environment (WRICE), which can be applied to various Internet applications. In WRICE, we add real-time interaction features to WWW. Firstly we are going to use it as a formal tool for distant learning. This system is different from the common e-learning systems: during the operation, the only system requirement a user needs is a browser, this friendly feature would be very convenient and economic to users.

There are quite a few CSCW-based distant learning systems, also multicast [10,11,12] systems such as Mbone [4,5] is commonly used, but they are all restricted to a specific type of application. These systems are unable to do browsing and interaction simultaneously, also they need specific type of network equipments, so the deployment of these systems is very restricted. Our WRICE system can support distant learning as well as other collaborative applications easily. This paper is organized as follows: Section 2 presents the design architecture; in Section 3, we discuss the implementation issues; Section 4 concludes the work as well as addresses the future works.

2. Design Architecture and Considerations

The first goal of designing WRICE is the friendly user interface during the operation of the system. To allow various user groups to feel the friendliness of the system, we propose quite a few necessary and useful functions. Examples of these functions are to allow users browsing the HTML document synchronously, to build up a whiteboard and a co-editor with on-line file transfer capability, and to support real-time interactive voice/video communications. An overview of some nice functions in WRICE is described as follows:

A. Browser Synchronization

The function for synchronously browsing the HTML documents could be applied to various applications, especially for distant learning. The scenario is, an instructor clicks the anchor of an HTML document with a mouse, and browsers in both the instructor site and student site would display the same web page. In order to support possibly hundreds or even thousands of users browsing the same page simultaneously that may crash the web-server, we design a configuration as shown in Figure 1. When an instructor clicks an anchor, his browser will send out an URL and a sequence number, the URL carries the location of the browsed document, while the sequence number is used for verification purpose. Using this configuration, a WWW server does not have to handle requests directly come from the end-users, instead, it only needs to deal with much fewer requests that are initiated from a proxy or a number of proxies located in different places. These proxies then use the received URL to fetch the associated web page and send it to end-users through multicasting mechanisms.

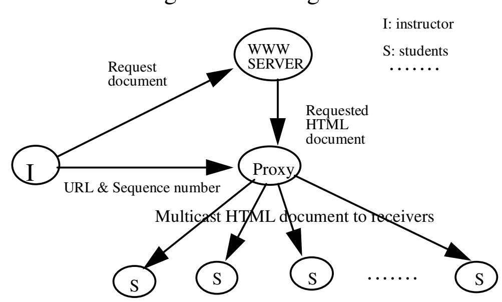


Figure 1. Browser Synchronization.

B. Whiteboard

Whiteboard is an important tool in CSCW, and we believe that “a figure is worth a thousand words”. The whiteboard used in WRICE is called the *cool-board,* it features several nice features which are not available in the traditional whiteboards. The traditional whiteboards use bit map system that supports multiple users, they usually have some deficiency in the control of critical sections. The cool-board in WRICE has the following advantages:

- All the figures in the cool-board are constructed as objects whose color, size and position can be changed independently. Further, these objects can be created and deleted dynamically.
- The content on the cool-board can be saved into files as well as printed out separately.
- It supports multiple users simultaneously and efficiently.

Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS’03)
0-7695-1910-5/03 $17.00 © 2003 IEEE

It is inappropriate to allow all users to access the cool-board in a random manner, otherwise the content on the cool-board may be disrupted without control. We propose three methods to deal with the issue:

(1) Area dividing method: The instructor or the presenter could use the mouse or e1ectronic pen to assign a portion of the cool-board area to a student or a listener who is going to use the cool-board, and the student is allowed to fully control that area. Any access to outside the area is prohibited and a warning message will be prompted.

(2) Object assigning method: If a user tries to modify a specific object on the cool-board, the owner of the object could designate the control of that object to that user.

(3) Cool-board copying method: A user could make a copy of the current cool-board (including the contents on the board), and make modification freely, then he could ask the owner to examine the result after the modification is finished. The modified object may replace the original one if the presenter allows the listener to do that.

Our cool-board has a specific feature and it can be divided into multiple sub-whiteboards. A sub-whiteboard can be further divided into even smaller sub-whiteboards until it is too small to be useful.

Basically, the aforementioned whiteboard is based on the computer screen display. Recently, there is a more advanced electronic whiteboard which is already available in the market. This kind of electronic whiteboard looks like a traditional plain whiteboard, but it features much more functions such as digital recording with JPEG compression, wireless networking, and web-browsing. But most importantly, we could use a nature way to write and draw on the electronic whiteboard. That means we don't even need a camera to capture the writing and drawing on the traditional whiteboard if the talking face of the presenter can be ignored. We are planning to use it as the major component in our WRICE system.

C. Real-time Audio/Video Communications

In WRICE, the audio/video communications can be categorized into three types, one-to-one, one-to-many, and many-to-many. The first type is a typical collaborative work between two parties. This type of application can be easily implemented. Examples for the second type are distant learning systems, in which the instructor sends the course materials to multiple students. This type of application needs a media server that can handle the delivery of various multimedia data. While in the third type, a conference meeting is held among more than two users through the network, and a participant can communicate with other participants under the grant of the chairperson.

D. CSCW Control Language

To facilitate the above functions, we need a CSCW control language. Nowadays it is still lack of a common standard in the application protocol for CSCW. To ease the implementation of WRICE, we developed a CSCW control language, called CCL. As we know, MS-Windows provides many so-called classes or control objects, such as text box, list box, and button. In MS-windows, we usually use SendMessage API to control these objects, through this specification of SendMessage, we know that MS-Windows always use handles to manipulate the objects. The problem is that an object is given different handles while it is in different machines, even in the same machine with different executing times. Therefore we need to solve this issue. The details of our CSCW control language is beyond the scope of this paper.

E. Mutual Exclusive Access of Resources

In WRICE, there are always data contents shared by multiple users, for example, in a co-editor or a whiteboard, it is very likely that two or more users access the same file, or same memory address, or same object simultaneously, and mutual exclusion (critical section) problem occurs.

Three approaches could be used to solve the aforementioned problem: the centralized approach, the distributed approach, and the token ring approach. In the first approach, there is a process that acts as a coordinator, any process which tries to enter the critical section must obtain a permission from the coordinator, and it only allows one process to enter the critical section at any time moment. WRICE uses the centralized algorithm, in which a FIFO queue is associated with a critical section. A request to a critical section must be sent to the coordinator, which then checks the corresponding queue. The request will be handled immediately if the corresponding queue is empty, otherwise the request will be put on the waiting list.

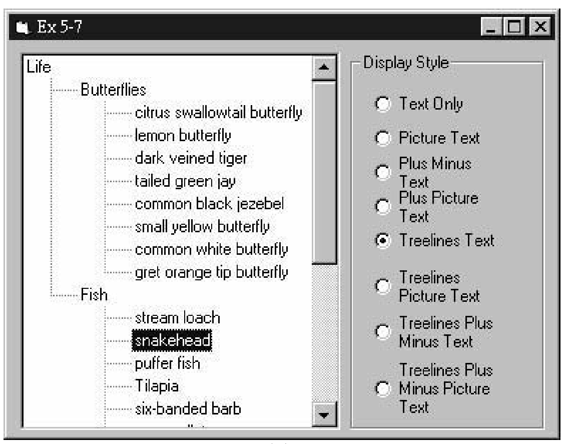
To support the task mentioned above, we design an important process called CSCW manager (or CSCW coordinator), which is in charge of all administration and coordination tasks in the system. The CSCW manager must be activated before any CSCW member enters the system. All new members joining the system must register first and certain information such as IP address must be provided. A member also has to send a quit message to the CSCW manager when he/she is leaving the system.

F. Tree Structure for Enhanced Multicast

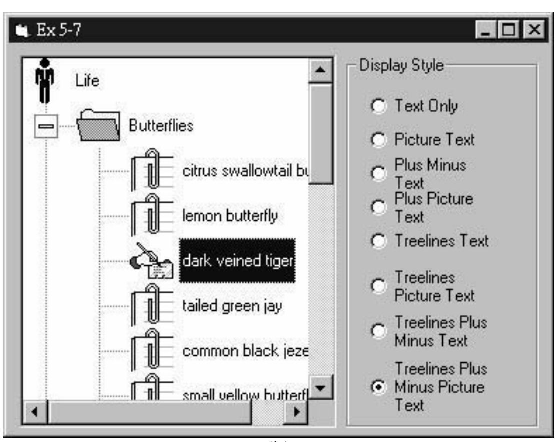
0-7695-1910-5/03 $17.00 © 2003 IEEE

WRICE provides a tree structure that allows users to have a whole layout of the home page structure. Since hyperlinks contained in an HTML document may encounter recursion, for example, a document A links to document B, which links to document C, which links back to A. This is a special characteristic of hyperlink and hypermedia, and it is essential for quickly linking to the requested document. However, we found that courseware in distant learning application always features a hierarchy, for example, a text book always includes chapters, sections, and subsections. To represent the structure of the courseware, we propose two tree structures, as shown in Figure 2. In which part (a) lists the tree using text only, while part (b) uses pictures in addition to the text description.

Basically, the link organization of hypertext and hypermedia has a network-like structure. While the regular book content is organized hierarchically with two or more levels. Even though, when the courseware is converted into homepages, it will become network-like structure because different parts of content have to refer to each other.



(a)



(b)

Figure 2. (a) A treelines text. (b) A treelines plus minus picture text.

There are two typical problems in using hypertext: disorientation and cognitive overload [3, 8]. The former is closely related to the network-like structure, which usually causes a user to get lost and unable to know where he/she is in the hierarchy, hence is unable to quickly get the information he/she wants. This wastes time and network resources. A tree-structured courseware allows homepages to have a hierarchy, so a user could easily understand where he/she is while browsing the homepages.

3. System Implementation

To convert the network-like structure to the tree structure, we firstly defined two tags in HTML documents:

<TREE_CHILD HREF="URL">
<TREE_PARENT HREF="URL">

Here <TREE_CHILD HREF="URL"> points to child nodes, while <TREE_PARENT HREF="URL"> points to parent nodes, our WRICE supports and is able to interpret these two tags. Second, we use programs to solve the recursion problem in tree structures.

Different from the regular browsers, one of the goals of WRICE is to allow a instructor and students at different locations to browse the same document simultaneously and quickly. To improve the system performance, WRICE uses pre-fetching to access the content in advance. Besides, to support a large number of students which tends retrieve the same document at the same time, we use a so-called Server-Coordinator-Client model plus multicasting to replace the traditional Client/Server model, so we can avoid the congestion which are very likely to occur when all students access the same document simultaneously. We have modified the multicast process to improve the system performance. When an instructor is accessing a page, the system in the instructor site can pre-fetch the following part of the current content. In the mean time, this courseware will be multicast to systems in different local media server sites. When an instructor or a student clicks a hyperlink, the WRICE will check whether the selected link is in local server or in remote server site. The system architecture is shown in Figure 3. The reason for doing so is, currently the multicast function of most routers is disabled, because the software-based function is easily crashed if the multicast function is turned on under a heavy traffic load. Therefore we use a local media server to handle the multicast. Although this configuration is not a new idea, however, we implemented it and it saves a lot of traffic between a student site and the remote server. Especially when students are living in the same dorm or in the same computer room, because once a web page is browsed

Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03)
0-7695-1910-5/03 $17.00 © 2003 IEEE

and downloaded, the page may be cashed in the local media server, and the following accesses to the same page can be handled in the local server.

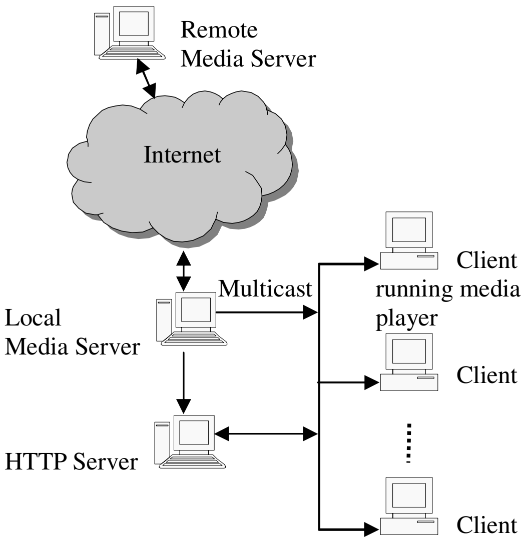


Figure 3. Multicast architecture in WRICE.

Appendix A lists some details of implementation about WRICE. There is another feature in WRICE, that is, we are trying to build up the tree structure of hyperlinks. It is quite straight forward to prove that a network-like structure can not be converted into a tree structure. However, courseware do have certain hierarchy, so we could derive an algorithm that converts network structure into tree structure, which can be easily recognized by most people We call this algorithm the hyper-tree algorithm. The comparison of hyper-tree and original hyperlink network is as follows:

- The number of nodes in a hyper-tree = the number of node in the original hyperlink network.
- The number of links in a hyper-tree ≦ the number of links in original hyperlink network

Our hyper-tree algorithm is as follows：

(1) Starting from the root node, we use BFS (Broad First Search) algorithm to traverse the tree nodes. Since each node has a depth which is measured as the number of levels from the root, we use the smallest one as the depth for a node:

$$l(n_i)=min(l_1, l_2, \ldots, l_k)$$
$$depth(n_i)= l(n_i) - 1$$

Where $l_1, \ldots l_k$ represent alternative paths from the root to node n_i. Given a hyperlink network, we can use Dijkstra algorithm to find out the shortest distance from every node to the root node.

(2) If there are multiple paths from the root to a node, then delete the redundant link(s) with a larger depth.

(3) If there exists a link from n_{i-j} to n_i and a link from n_{i-k} to n_i, then one of these two links must be deleted. This process allows us to convert a hyperlink network into a tree.

Figure 4. A scenario of WRICE.

4. Conclusions and Future Works

In this work, we propose a framework called web-based real-time interactive collaboration environment (WRICE), which could exchange audio, video and hand drawing information between the peer parties for immediate play out. The framework supports real-time bi-directional multimedia communications. The major parts of the system has been implemented, and it can be used to support video conference as well as distant learning effectively. Figure 4 shows a system scenario. Two video windows in the right hand side can be used for video conference. In this scenario users could discuss the map through the network. We are still working on it to improve its features such as multi-parties video conference, large group distant learning, sub-whiteboard, and user friendly GUI. We also plan to use an electronic whiteboard as the major input device, so that a presenter could easily use the system for presentation.

References

[1] R. T. Kouzes et al., "Doing Science on the Internet", IEEE Computer Magazine, August 1996, pp. 40-46

[2] C.T. Sun and C. Chou, "Experiencing CORAL: Design and Implementation of Distant Cooperative Learning", IEEE Transaction on Education. August 1996, pp. 357-366.

Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03)
0-7695-1910-5/03 $17.00 © 2003 IEEE

[3] J. Grudin, "CSCW: History and Focus", IEEE Computer Magazine, May 1994, pp. 19-27.
[4] http://www.savetz.com/mbone/
[5] http://www.ripe.net/ripe/wg/mbone/home.html
[6] http://www.telekooperation.de/cscw/
[7] http://www.ics.uci.edu/~grudin/Papers/IEEE94/IEEEComplastsub.html
[8] J. Grudin. "Groupware and Social Dynamics: Eight Challenges for Developers." Communications of the ACM, 37(1):93-105, 1994.
[9] http://www.hypernews.org/HyperNews/get/www/collaboration.html
[10] http://www.ietf.org/internet-drafts/draft-ietf-idmr-dvmrp-v3-10.txt
[11] http://www.ietf.org/rfc/rfc2201.txt
[12] J. J. Pansiot and D. Grad, "On Routes and Multicast Trees in the Internet," Computer Communication Review, Vol. 28, No. 1, January. 1998, pp. 41-50.
[13] R. Bentley," Architecture Support for Cooperative Multiuser Interfaces," IEEE Computer Magazine, May 1994, pp. 37-47.
[14] W. Reinhard et al., "CSCW Tools: Concepts and Architectures," IEEE Computer Magazine, May 1994, pp. 28-36.

Appendix A. Implemented Features of WRICE

(1) Audio : MPEG 3
(2) Video : MPEG 4
(3) Software: Microsoft Media Server
(4) ASF Play support: YES
(5) Channel Support: YES
(6) Formatted Data Support: YES
(7) Audio/Video Recording：YES
(8) User Management：YES
(9) Real-Time Audio Support : YES
(10) Real-Time Video Support : YES
(11) Iframe = 10
(12) Frames/sec = 15
(13) Interactive = YES
(14) INI File Support: YES
(15) Multicast Support: Yes
(16) Unicast Support: Yes
(17) Distribution Support: YES
(18) Whiteboard Support: YES
(19) Object Whiteboard Support: To be done
(20) Cooperative Support: YES
(21) File Transfer: YES
(22) Student Data Query Support : YES
(23) Mail Support: YES
(24) BBS Support : YES
(25) Web Site Support : YES
(26) Cross Sub-network Distant Learning：YES

Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03)
0-7695-1910-5/03 $17.00 © 2003 IEEE