CORAL*:A Multi-server Distant Cooperative Learning System

Ying-Dar Lin Yuan-Cheng Lai Ming-Horng Chang
Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan

E-mail: ydlin@cis.nctu.edu.tw
Phone: +886-3-5731899
Fax: +886-3-5721490

Abstract: CORAL(CoOperative Remote Access Learning) system, a WWW-based distant and cooperative learning system, was originally developed with a single-server architecture. Currently some organizations and schools may own self CORAL servers to provide the distance learning for their employees and students. However, the original design of the single-server architecture cause no linkages between these CORAL servers, and result in the narrow learning environment; i.e., fewer course materials and fewer learning partners.

In this paper, we present a *multi-server tight-coupled* CORAL* system. In order to deploy CORAL across multiple organizations and schools, we develop multi-server and user management procedures and extend single-server communication channels to multi-server communication channels in our CORAL* system,. In addition, we add four new functions into this system: the guided navigation system, the mail system, the audio system and the learning activities query system. Currently the system provides an environment for (1) users to access remote class materials, find learning partners, and conduct cooperative learning; (2) instructors to monitor and study learning activities; (3) managers to operate and maintain their local servers independently.

Introduction

Many applications (like News, BBS and WWW [1]) are available in the computer networks due to the quick progress of the technology. On the News, people can read articles which are ASCII text. On the BBS, people can read the ASCII articles and chat with another persons. On the WWW, people can get their desired multimedia information by browsing the materials through browsers. Various server sites are located in the large and sporadic places to provide these services.

According to the communication between the servers and the linkage between the users, there are three degrees of connection strength; *independent*, *loosely-coupled*, and *tightly-coupled*. On the News, the articles are exchanged by the connection (NNTP) between the News server. However, the people can't communicate with each other in the same or different server. Hence the News application can be classified into the *loosely-coupled* system. On the BBS, there are no connection between the BBS server, the articles are just exchanged by the mail forwarding. The users can chat with another in the same server, but can not communicate with users in the other site. Hence the BBS application also is *loosely-coupled* system. On the WWW, the WWW browsing is a solitary activity because users are not aware of each other and can not communicate with each other. Also, WWW servers operate independently with hyperlinks being the only connections between them. The WWW application is classified into a *independent* system.

Distance learning [2-6] is becoming a popular application that helps people to get education and training for no spatial and temporal restriction. Since the content of course must be fancy to attract the interests the learner, WWW application is the best choice as the base of distance learning system. Nevertheless, Due to the independent characteristic of current WWW application, it must to modified to fit the need the distance learning system. The CORAL (CoOperative Remote Access Learning) system [7,8] was developed in NCTU, Taiwan. It is a WWW-based system which supports class material access, learning activities monitor and query, and communication channels between learning partners.

However, The CORAL still is a *loosely-couple* system. When deploying CORAL to a group of educational institutions, each CORAL server is likely to be managed by a single institution independently. This arrangement partitions and limits the student population for cooperative learning. The learner only retrieve the course materials and find the learning partners in the same server. Hence integrating multiple CORAL servers, so that cooperative learning can be done across all involved institutions, is necessary. Whatever functions are in the single server system should now be executed across multiple servers. Thus, we extend the original *loosely-coupled* CORAL system to a *multi-server*, *tightly-coupled* CORAL* system. Our CORAL* provides wider learning environment. Users can retrieve more course materials and discuss with more learning partners, which are not only at the local site but also at remote sites. Also CORAL* provides the better expansion capability. A new site can join the CORAL* environment anytime and anywhere, and form a larger learning environment Finally some additional features are also added into this system, for example, the mail system, the guided navigation system and the learning activity query system.

Distance Cooperative Learning Environment

With the aids of computer networking, teachers and students are not confined to physical classrooms because courses and training performed electronically over computer networks. Through distance learning systems, the limitations of traditional teaching can be eliminated. Distance education through computer networks is a new promising direction in education nowadays.

• Virtual Classroom

Virtual classrooms[9] are in the form of courses provided across the network. In a physical classroom, the relationship between teachers and students is direct and face-to-face. Students must attend schools for learning at the school time. This kind of learning is limited in the domains of space and time. But in virtual classrooms, geographic and time limitations are broken.

To build the environment of a virtual classroom, some equipment must be provided. In a physical classroom, equipment like a chalkboard, projection machines, textbooks and so on should be available. In a virtual classroom, tools like audio-visual equipment should be used in the form of network-based software applications. The following functions may be provided in a virtual classroom: on-line class material, shared whiteboard, video/audio system, notes, notes, communication channels and so on.

Issues for a virtual classroom range from technical considerations to sociological aspects. Here, we focus only on technical issues as discussed below.

(1)Class Size:

In the traditional workshops or classrooms, the number of attendants is limited by the amount of space (physical classrooms). For distance learning, the size of a class will be affected by the computing and networking power.

(2)Course Production

In most of the current learning systems, the courseware is in the form of hypertexts. For experienced instructors or non-experienced instructors, the production of well-structured courseware is a challenging task[10,11].

(3)Learning Environment:

Interactivity plays an important role in learning. By providing communication channels like a chatroom, e-mail, or a shared whiteboard, an effective learning environment can be built.

(4)Network Bandwidth and Resource Limitation

Network bandwidth and resources are important factors in designing a distance learning system. A network with low bandwidth slows down the transfer of class materials and reduces the quality of some applications, like video conferencing, that need high bandwidth.

• Learning Environment on WWW

WWW(World Wide Web) is a candidate environment for distance learning. WWW is a resource sharing technology based on HTTP (HyperText Transport Protocol) [12], which can carry text, picture and sound all together. In this environment, the materials are represented in the form of hypertexts. Hypertexts are edited in HTML(Hypertext Makeup Language), and URL(Uniform Resource Locator) represents its hyperlink. This provides readers an attractive and easy-reading way to browse the environment.

The development of WWW and real-time applications over the Internet opens another door for distance education. The course materials can be organized in hypertexts. The inclusion of Internet applications(for example I-Phone, chatroom) makes WWW-based distance learning systems more attractive and applicable.

Learning in distance education can be divided into two categories: interactive and non-interactive. In interactive learning, users not only read the materials but also get help from the systems or send questions to the systems for answers. In non-interactive learning, users just read the content of the courses. In distance learning systems, providing interactive learning tools significantly improves the environment.

• CORAL* System

In this section, we introduce CORAL* system and its functions. CORAL* system is based on the WWW and follows the multi-server architecture. It is designed to be a distant cooperative learning environment. The goal of CORAL* system is to provide an environment where students, over a network, can (1) learn certain subject materials stored either locally or remotely, (2) find learning partners, discuss with other students or instructors, (3) query learning activities. The instructors of CORAL* system can inspect students‘ status and provide help as needed.

The whole system can be viewed in four aspects (1) placement and access of hypertexts, (2) distributed anagement , (3)communication and learning activity query and (4) communication channels for learning partners.

(1) Placement and Access of Hypertexts

CORAL* hypertexts are placed in a distributed way on many servers scattered over the Internet. An organization which joins CORAL* system may setup a server and place the materials on it. These materials are writeable by the local manager, but read-only by remote servers. The activities of hypertexts access are recorded. The learning activities record is used to search for learning partners or learning information. Users in CORAL* system can query and find learning partners. This is one of the key features of CORAL* system.

(2) Distributed Management

Distributed management is implemented in the multi-server CORAL* system. A management protocol is defined to exchange monitoring and management information between (1) a client and its local server , and (2) a local server and a remote server. The registration of users and local servers is also handled. A local server maintains its own user accounts and a centralized user manager is in charge of managing all the users in the whole system. A centralized registration server is in charge of handling registration of local servers.

(3) Communication and Learning Activity Query

Activities query plays an important role in a distant cooperative learning environment. It helps users in the system to find suitable learning partners, group discussions and so on. The queries in CORAL* system can be classified into two types: (1) communication activities queries and (2) learning activities queries.

Communication activities queries are used when users want to use the communication channels provided by CORAL* system. It helps users to get the information about communication groups that are in progress in the system. We list some communication activities queries below:

◦ List all chatrooms that are in progress.

◦ List users in a given chatroom.

Learning activity queries are used when users want to find learning partners. In CORAL* system, the activities of materials access are recorded and placed in local servers. We list some learning activity queries below:

◦ List in-line users.

◦ List all local servers in this system.

◦ List users who have accessed a specific hypertext file.

◦ List hypertexts that a specific user has accessed.

◦ Trace a user's learning path.

(4) Communication Channels for learning partners

Several communication facilities are provided in CORAL* system: (1) guided navigation tool, (2) chatroom, (3) shared whiteboard, (4) audio conferencing, (5) mail, and (6) discussion board. The first four are real-time functions across the multi-server architecture, while the other two are non-real-time functions:

(1) The guided navigation tool is a new feature by which WWW browsing is no longer a

solitary activity. A teacher can lead students to read the hypertexts on WWW. An audio tool is provided to enable audio discussion during guided navigation.

(2) In the chatroom system, users can create virtual discussion classrooms and communicate with each other by texts. It is extended from the single-server architecture to the multi-server version. Users scattered over the local servers can chat even if they are not at the same site.

(3) The shared whiteboard system is used to discuss with each other by graphs and texts. It is also extended from the single-server to the multi-server architecture.

(4) The audio conferencing system is used to communicate by audio speech. It is also a multi-server version.

(5) The mail system is another new function. It is designed for the purpose of communication and convenience for users. Again, it is also a multi-server version.

(6) The discussion board is similar to the board on the widely used BBS. It is imbedded in the browser developed in our project. Users post to the local server on some specific topic. The posts on the server can be read by users associated with the local server but not readable by remote users . This is done by keeping URL links to the posts on the local server.

Design and Implementation

In this part, we describe each CORAL* subsystem and examine the implementation details.

• CORAL* Implementation Architecture

CORAL* system is a multi-server, web-based system that follows pure client-server architecture. There can be more than one local server in this system. A local server is installed in a workstation and it is in charge of providing local services to the local users. The client program is runnable on MS windows in PC. In addition, there are two servers: “Server Manager” and “User Manager”. Both of them are also installed in workstations. User Manager is connected with the Sybase system. The overview of CORAL system is shown in figure 1.

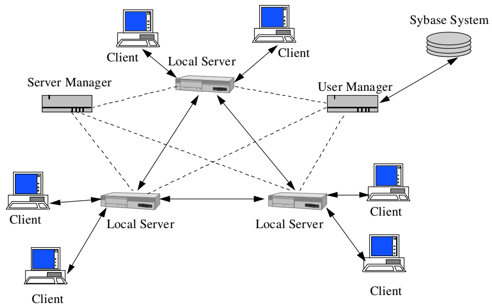


Figure 1. Overview of CORAL*.

Next, we take a look at the local server and the client. The architecture of a local server and a client is shown in figure 2.

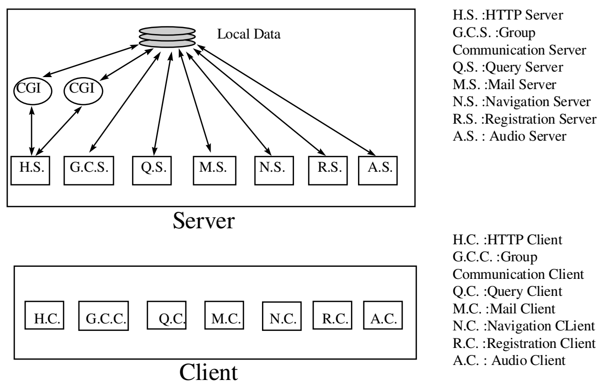


Figure 2. Architecture of a Local Server and Clienr

In the following, all the components of a local server will be described.

(1)HTTP server and http client:

The HTTP server is a standard Web server, but the HTTP client is the browser developed in our project.

(2)CGI scripts:

The CGI scripts are used for the recording of learning activities and the handling of discussion boards.

(3)GC server and GC client:

The GC(Group Communication) server is in relation to GC clients. Both of them forms a group communication system. In this system, a chatroom, a shared whiteboard and communication queries are provided.

(4)Query server and query client:

The query servers are in relation to the query clients. They form a query system. In CORAL* system, cooperative learning is emphasized. In order to achieve this demand, we provide learning activity queries.

(5)Navigation server and navigation client:

Navigation servers are in relation to navigation clients. They form a navigation system. In this system, a teacher can navigate for students to browse the hypertexts in WWW.

(6)Mail server and mail client:

Mail servers are in relation to mail clients. Both of them form a mail system. In order to let CORAL* system be a more friendly system, we implement the mail system.

(7)Register server and registration client:

The registration server is used when a new local server joins or drops from CORAL* system, or when a user who registers in CORAL* server. The registration server is connected with the Server Manager and the User Manager.

(8)Audio server and audio client:

The audio servers are in relation to audio clients. They form an audio conferencing system. It provides users real-time voice communications.

(9)Local data:

The local data is in relation to HTTP servers, GC servers, query servers, mail servers, navigation servers and registration servers. The local data includes: (1) on-line materials, (2) information about local user accounts, (3) a record of learning activities, (4) the articles of discussion boards, (5) information of group discussions, (6) the mail, and (7) the server lists.

• Server Manager

In the multi-server CORAL* system, a server manager is in charge of the management of all the local servers. A protocol is defined for the management of local server. In CORAL* system, there can be more than one local server, but there is only one server manager. It is designed to be a centralized server.

There are two operands between the Server Manager and local servers:(1)joining, and (2)withdrawing. The operand of joining is performed when a new local server wants to join the CORAL* system. The new local server sends a joining request to the Server Manager. The server manager processes the request and returns a server list to the new local server. Then the server manager informs other servers that a new one has joined this system.

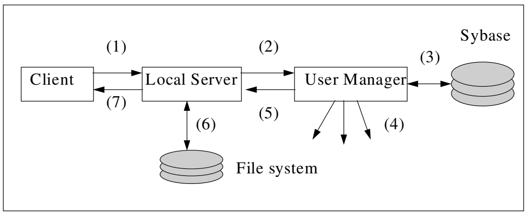
The operand of withdrawing is performed when a local server wants to withdraw from the CORAL* system. The local server sends a withdrawal request to the server manager, the server manager accepts the request and send an ACK back to the local server for confirmation. Then the server manager informs the other local servers that this local server has left the system.

That is, the local servers and the server manager keep IP addresses of all the local servers. When a new one wants to join or leaves the system, the server manager is responsible for the message transmission.

The server manager is implemented to be a centralized server. We adopt centralized control for the management of all the local servers in CORAL* system. Two reasons contribute to the need for centralized architecture of the server manager. The first one is about the actual implementation. To design and implement a centralized server manager is easier than for a decentralized server manager. The second one concerns the management. The management of local servers in a centralized server manager is better than that in the decentralized server manager.

- **User Manager**

"User Manager" in the multi-server CORAL* system is responsible for the management of all the users in the system. It is used to identify users who have multiple accounts in more than one local server. The identification of multi-account users helps with the search for learning activities. The architecture of User Manager is shown in figure 3.



(1)Client sends request of registration
(2)Local Server connects to User Manager.
(3)User Manager connects with Sybase system.
(4)User Manager informs other local servers the user has registered
(5)User Manager replys to Local Server.
(6)Local Server places information in the file system.
(7)Local Server replies to Client.

Figure 3. Architecture of User Manager.

1. A new user asks for the registration. He fills in a registration form and sends it to his local server.

2. The local server receives the form, makes a connection with the user manager.

3. The user manager receives the form, makes a connection with the Sybase system and then sends the form to the Sybase system. The Sybase draws out the ID and the IP of the local server, checks and then inserts it into the table. Then it returns a server list and a name list to the user manager.

4. If the registration succeeds, the user manager informs other local servers listed in the server list that the user has registered. It sends the new local server IP and account name to them.

5. The user manager returns the server list and the name list to the local server.

6. The local server puts the user's server list and name list in its local disk.

7. The local server sends a message to the user for confirmation.

Generally, the user manager maintains all the users in the system and processes the registration of new users. Every local server maintains its own users and keeps users' server lists and name lists received from the user manager.

In the multi-server CORAL* system, the user manager is combined with the Sybase system for the management of all users in this system. The relationship between the user manager and the Sybase system is the client-server architecture. Imbedded SQL (Structured Query Language) is supported in C language[13].It is used as the interface for a program to communicate with the Sybase system.

• Mail System

The mail system in CORAL* system is implemented for the purpose of communication and convenience to users. It provides a communication channel to users for the message exchange ,and provides a friendly environment for learning. Four basic functions are provided in the mail system: send mail, read mail, delete mail and list mail.

• Navigation System

In the Web environment, students may get lost because they are not familiar with the hypertexts on WWW. The navigation system is designed to solve this problem. A teacher can navigate for students by the browsing of hypertexts. Figure 4 shows the architecture of the navigation system.

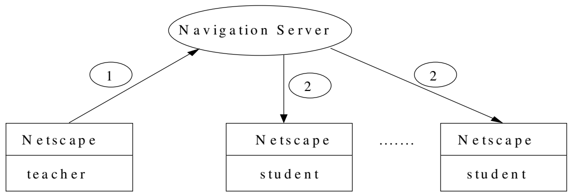


Figure 4. Architecture of the Navigation System

1. The navigation client❶, the teacher ,gets the current URL in Netscape and sends it to the navigation server.

2. The navigation server receives the URL and then sends it to the navigation clients❷and ❸,the students. Then the navigation clients import the URL into Netscape.

Netscape provides a SDI(Software Development Interface), which is an API (Application Programming Interface) used for the DDE(Dynamic Data Exchange) implementation that Netscape Navigation for MS Windows follows. Netscape's DDE is a kind of IPC (InterProcess Communication) that follows the client-server architecture.

Two topic names are used in Netscape's DDE implementation of the navigation system. They are "WWW_GetWindowINfo" and "WWW_OpenUrl". The former is used by the teachers to get the current URL in Netscape. The latter is used by the students to import a given URL into Netscape.

• Communication Channels

In CORAL* system, communication channels are provided for the purpose of group communication. Three communication tools are available:(1) chatroom, (2) shared whiteboard and (3) audio conferencing. In the following, we will described these tools in detail.

(1) Chatroom

Unlike the chatroom in BBS, the chatroom in the CORAL* system is implemented based on the architecture of multi-servers. Every user in the CORAL* system can enter any chatroom that is taking place. The logical diagram of a multi-server chatroom is shown in figure 5.

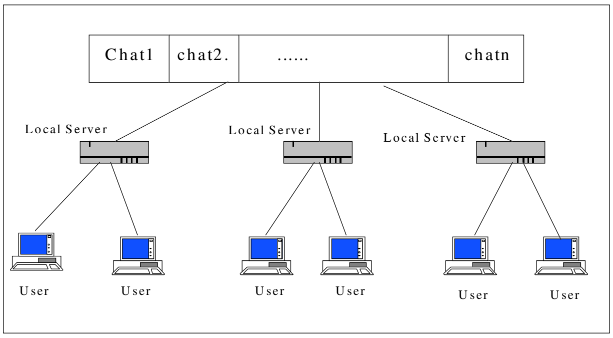


Figure 5. Logical Diagram of multi-server Virtual Classrooms

(2) Shared Whiteboard

The shared whiteboard in the CORAL* system is implemented based on the multi-server architecture, also. The logical diagram of multi-server chatroom is shown in figure 5. In the shared whiteboard, users can draw lines, circles rectangles, ellipses and so on to communicate with each other.

(3) Audio Conferencing

Audio conferencing is provided in CORAL* system. Users in CORAL* system can communicate with each other by speech. The speech transportation is one-to-many. A user in a virtual classroom can transmit his speech to all the members in this virtual classroom. Hardware equipment, like microphones and earphones should be available when users use this system.

• Learning Information Query

In a distant and cooperative learning environment, it is necessary to provide users information about learning partners. In CORAL* system, all the learning activities are recorded. Then users can query the learning activities to find learning partners. Two kinds of queries are provided:(1) query according to users, and (2)query according to material.

In the multi-server CORAL* system, a learning activity search mechanism is shown in figure 6. Each local server maintains its own users, including their server lists and name lists. When the queries are performed to collect information concerning a specific user, the local server just sends queries to those local servers listed in the user's server list.

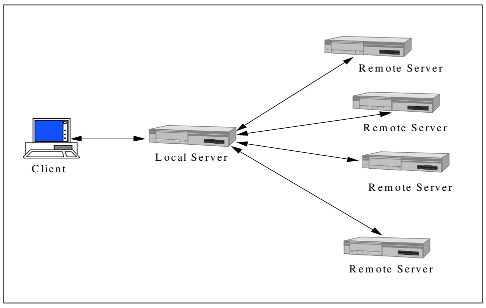


Figure 6. Diagram of Search Mechanism.

Implementation Issues

In this part, some important issues are identified and the approaches to them are described.

• Adoption of HTTP Proxy Server

There are two kind of web servers. One is called “Proxy Server”, developed by CERN(European Center for Nuclear Research); the other is a non-proxy server, developed by NCSA(National Center of Supercomputer Application). In the following, we will describe what a proxy server is and why we choose HTTP proxy server as our local server.

A HTTP proxy server has two characteristics: (1)caching and (2)proxy. Picture 7 shows the structure of HTTP proxy servers.

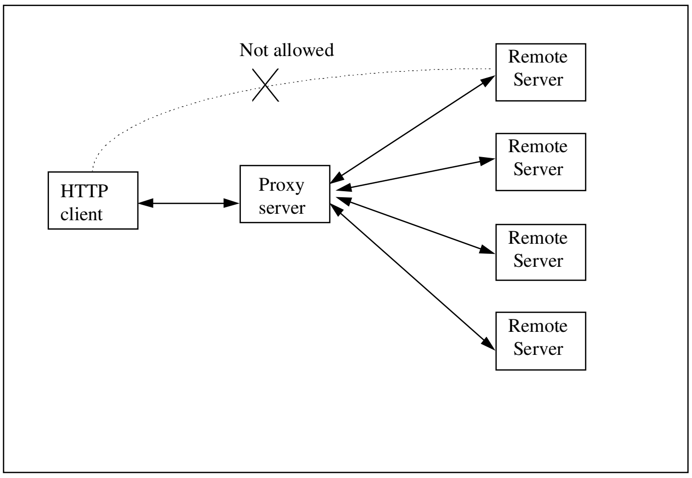


Figure 7. HTTP proxy server

The adopting of an HTTP proxy server greatly simplifies CORAL* implementation. All HTTP and management requests from clients have to go through the client's proxy which is usually set to be the local CORAL* server. That is, all the requests from clients must be handled by their local servers.

However, a suitable cache size of proxy servers is an important factor that will affects the performance of the whole system. With a big cache size, the performance is improved, but the resources will be exhausted and the problem of hypertext location transparency arises apparently. With a small cache size, the performance degrades. Adjusting cache size plays an important role in using of HTTP proxy servers as our local servers.

• Policies of Recording Learning Activity

In the CORAL* system, the activities of hypertext access are recorded. Users can query the learning information for search for learning partners. Users navigate around a system composed of many local servers. One issue arises when we want to record users' learning activities. It concerns where to record the access activities. There are two policies: (1) recording at the user's home site and (2) recording at the site storing the hypertext.

(1)Recording at the user home site:

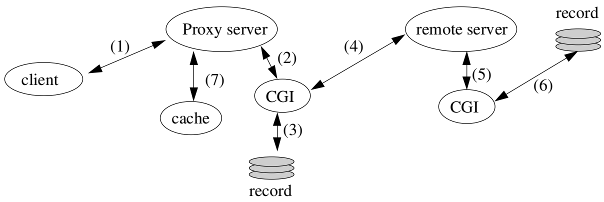
In this policy, the learning activities of hypertext access is recorded in the local server where the user registers. By setting of local servers to be a proxy server, all the requests from the user are handled by the local server. When the user accesses materials stored, either in a remote server or in the local server, the recording is done by the local server.

(2)Recording at the site storing the hypertext:

In this policy, the learning activity of hypertexts access is recorded in the server that the hypertext belongs to. When users navigate the whole CORAL* system, the learning activities are recorded by the servers that these hypertexts belong to.

We choose the policy "recording at the site storing the hypertext" as our implementation approach. We have two reasons for this choice. One is that the local server with the course materials has the right to know who access its hypertexts. This kind of policies allows the recorded data to be well-maintained. The other reason is that it is easier to implement. When the users want to access materials remotely, they must have accounts on that server. After accessing hypertexts on the remote server, the learning activities are recorded and the recorded data is just placed on that remote server.

However, a problem arises when we set a local server to be a proxy server. A cache mechanism is provided in HTTP proxy server. When the user want to access remote materials that are currently in the local cache, hypertext access is performed in its local server instead of the remote server. In this situation, it violates our recording policy. Our solution to this problem is shown in figure 8. When materials in the cache are accessed, a CGI script is activated to contact the CGI in the remote sever indicated by the URL. Then the remote CGI script performs the recording and puts the recorded data in its server.



(1) hypertext is accessed
(2) if URL points to local server
(3) CGI records the accessing activity
(4) if URL points to remote server
(5) remote server pass it to CGI
(6) CGI record the accessing activity
(7) return hypertexts in cache to client

Figure 8. Recording in HTTP proxy server

• Mechanisms of Recording Learning Activities

In CORAL* system, recording learning activity plays an important role in cooperative learning. In the following, we introduce three methods to do this and address their issues.

(1)Recording by Web server:

By modifying Web servers, we can have the activities of hypertext access recorded directly by web servers. However, some problems arise when using this approach. The first one is that modifying web servers may take a lot of effort. Besides this, there are many web servers around the world. Modifying web servers may cause them to be incompatible with the others. Thus, it should be the last resort.

(2)Recording by CGI:

CGI(Common Gateway Interface) is designed for the extension of web servers to satisfy customers' requirement. To achieve the function of learning activity recording, we can use CGIs to help us.

Some problems also arise when using by this approach. The most troublesome thing we need to face is that every hypertext placed on the web server must be modified. By adding some URLs in the hypertexts, when the hypertext selected by the URL is clicked on, the specified CGI script is called to perform the recording.

(3)Recording by Browser and CGI:

In figure 9, we show the architecture where the browser and CGI are combined to perform learning activity recording.

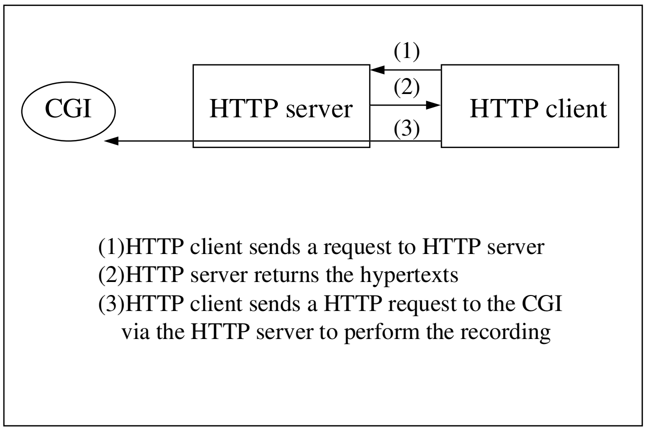


Figure 9. Record by using browser and CGI

When a user accesses some hypertexts, the browser sends a HTTP request to a specified CGI via Web server to record the activity. The CGI writes the record into the file system. In CORAL* system, we choose this method because the HTTP browser developed in our project has the basic recording function.

• Adoption of Databases

In the multi-server CORAL* system, a database system is adopted to store management data. Three types of management data should be maintained in the multi-server CORAL* system: (1)users information, (2)servers information, and (3)query information. We use a combination of a database system and a file system to handle them.

We have a centralized management architecture for all servers and users. We let each local server maintain its local users, on-line materials and query information. In the centralized management, we use the Sybase system to manage the data. In the local server, the learning activities and information of the local users are handled by the local file system.

Because database systems may not be available in some environments, the adoption of Sybase system in local servers may cause serious problems. So we choose the file system as an alternative. For better management and query support, we choose Sybase for the centralized management of users and servers.

• Server Management

In general, we have two ways to manage all the servers in this system: centralized management and distributed management. The management on local servers has to take care of the joining or dropping of a local server. When a new local server wants to join CORAL*, a message should be transmitted to all the servers in this system. When a local server drops out of CORAL*, a message also should be transmitted to the existing local servers in this system. That is, each local server should keep a server list of the system.

If the above process is to be done in a distributed way, it is more difficult because the new server doesn't know where to register and get the server list. So we set up a centralized server that is responsible for the joining or dropping of a local server. This server is well-known and the new server which wants to join into our system can connect to it for registration and to get the server list.

• ID Mapping

In the multi-server CORAL* system, a user can register in more than one local server by different account names. In such a situation, the problem of user identification arises.

A centralized user manager is in charge of handling unique ID mapping. When a user wants to register at a local server, he or she must give detailed information to the system for registration, especially the globally unique ID. We use the globally unique ID to identify the user who have different account names in different local servers. For example, a user has an account in local server "A" with the user name "x" and also has a account in local server "B" with the user name "y". The system must be able to know that user "x" in server "A" and user "y" in server "B" are the same individual.

When the problem of ID mapping is solved, the system appears to be more tightly-coupled. A user can query a specific user at a local server and the system can return all the information about the specific user the local server in and remote servers.

Conclusion and Future Works

In order to provide a cooperative and distant learning environment, we extend the original single-server, loosely-coupled CORAL system to a multi-server tightly-coupled CORAL* system. In our CORAL* system, materials and learning partners are not limited to the local server . Users have a wider learning environment; i.e., the course materials and learning partners in the remote server also can be accessed and conversed

A protocol for the management of multiple servers is presented and some additional features are added into out system. These new features includes the navigation tool, mail system, query system, and audio system. Some implementation issues are addressed for the better performance of our system.

The next thing to do in the future is the diffusing of CORAL* system. Experiments will be performed in some courses to collect the problems and experiences. Then we can solve whatever problems arise and improve CORAL* system. Guidelines to adopt CORAL* will be produced.

Another thing is to port CORAL* system to heterogeneous networks, including ISDN[15], PPP modem, ATM[16], and CATV. For users that do not use the Internet, there are multiple platforms for them to choose from. Current CORAL* implementation is IP-based and can run over any network that supports IP transport.

Since all ATM products support IP-over-ATM, CORAL* is now able to run over ATM. However, to utilize the Quality of Service(QoS) guarantee feature of ATM, some network protocols in CORAL* need to be modified to use the ATM API (Application Programming Interface) directly.

References

[1] Torkington, N., WWW Primer, http://www.vum.ac.nz/~gnat/ideas/www-primer.html/,1994.

[2] Eleonora Bilotta, Mariano Fiorito, Dario Iovane, Pietro Pantano," An Educational Environment Using WWW", Computer Netwroks and ISDN Systems, 27, pp. 905-909, 1995.

[3] Bertrand Ibrahim, Stephen D. Franklin, "Advanced Educational Uses of the World-Wide Web", Computer Networks and ISDN Systems, 27, pp. 871-877, 1995.

[4] Squires, D., “Learning Through Multi-user Network CAL”, Computers and Education, Vol. 16,

No. 1, pp.87-91, 1991.

[5] Barnard, Y. and J. Sandberg, “Open Learning Environments: What Support Should They Offer?”, Proceeding of International Conference on Computers in Education, pp. 156-161, Taipei, Taiwan, December 1993.

[6] Jacobson, M. J. and J. A. Levin, “Hypertext and Network-based Learning Environments: Technology for the Construction of Personal and Shared Knowledge Spaces”, Proceedings of International Conference on Computers in Education, pp. 282-287, Taipei, Taiwan, December 1993.

[7] Ming-Chih Lai, Bih-Horng Chen, Shyan-Ming Yuan, “Toward a new educational environment”, World Wide Web Journal, Issue One:Conference Proceedings, O‘Reilly & Associates, Inc., pp.221-230, Nov. 1995.

[8] Ming-Chih Lai, Hsiou-Mien Lien, Shyan-Ming Yuan, “Navigation Control in Educational Hypermedia”, Proceedings of ED-MEDIA 95, pp.773, Jun. 1995.

[9] Dan Dwyer, Kathy Barbieri, Helen M. Doerr, "Creating a Virtual Classroom for Interactive Education on the Web", Computer Networks and ISDN Systems, 27, pp. 897-904, 1995.

[10] J.K. Compbell, S. Hurley, S.B. Jones, N.M. Stephens, "Constructing Educational Courseware Using NCSA Mosaic and the World-Wide Web", Computer Networks and ISDN Systems, 27, pp. 887-896, 1995.

[11] Robert E. Wray III, Ronald Chong, Joseph Philips, Seth Rogers, William Walsh, Joan Laird, "Organizing Information in Mosaic: A Classroom Experiment", Computers Networks and ISDN Systems, 28, pp. 167-178, 1995.

[12] Internet Engineering Task Force (IETF), Hypertext Transfer Protocol Specifications, Internet Working Draft.

[13] C.J. Date, "An Introduction to Database Systems", Vol. 1, Fifth Edition, Addision Wesley, 1990.

[14] Rebecca Eurich-Fulcer, Janet Warn Schofield, "Wide-Area Networking in K-12 Education: Issues Sharping Implementation and Use", Computers Educ. Vol. 24, No. 3, pp. 211-220, 1995.

[15] Stallings, W., “ISDN and Broadband ISDN”, second edition, Macmillan, New York, 1993.

[16] Handel, R., M. N. Huber, and S. Schroder, “ATM Network: Concepts, Protocols, Applications”

2nd edition, Addison-Wesley, 1994.

[17] Clifford A. Lynch, "Networked Information Resource Discovery: An Overview of Current Issues", IEEE Journal on Selected Areas in Communications, Vol. 13, No. 8, October 1995.