



Synchronous navigation control for distance learning on the Web

Ping-Jer Yeh ^{*}, Bih-Hong Chen ¹, Ming-Chih Lai ², Shyan-Ming Yuan ³

Institute of Computer and Information Science ⁴, 1001 Ta Hsueh Rd., Hsinchu, Taiwan30050, ROC

Abstract

Most of the research on “distance learning on the Web” is learner-centric. That is, learners play an active role in their learning process, while instructors are passive or even invisible. Unable to treat the instructor/learner relationship as a whole causes some problems. Therefore, we introduce a new concept of instructor-oriented navigation control on the Web, and then propose and implement a protocol for it. In this model, an instructor (technically, *master*) can setup a learning group, and those who wish to be guided distantly (technically, *slaves*) during the study may have their navigation activities controlled synchronously by joining the group. We believe that instructors will become more active and positive with the help of our proposed model in a WWW-based educational environment.

Keywords: Distance learning; Synchronous navigation control; Master/slave model; Instructor/learner relationship; Albatross

1. Introduction

The *Albatross* [1] is a WWW-based distance education system that aims to provide an integrated and distributed hypermedia information environment for the academic community. In this environment, instructors can construct course material and prepare some learning assistance for learners; learners can access course material and participate in on-line and off-line discussion. Learners can use ordinary WWW browsers or Albatross clients, though with the latter they can use several unique functions easier.

After pilot experiments, we discover that it fails to treat the instructor/learner relationship as a whole. Our system, current WWW-based courseware as well, strives to solve navigation problems that learners conceive, but ignores more positive aids that instructors can provide. Consequently, not only is instructor’s potential underestimated in the domain of distance education, but also are learner’s navigation problems handled incompletely.

We re-examine the WWW model, and find that such a client/server model fails to support direct interaction between instructors and learners. Therefore, we propose a concept of master/slave models and synchronous navigation control so that instructors can help learners more in distance learning environment.

1.1. Background

Distance education exists for more than one hundred years, and there are various theories, media, and

^{*} Corresponding author. Email: gis84530@cis.nctu.edu.tw, <http://plato.cis.nctu.edu.tw/~william/>.

¹ Email: gis84509@cis.nctu.edu.tw, <http://plato.cis.nctu.edu.tw/~toto/>

² Email: gis83502@cis.nctu.edu.tw, <http://cissun51.cis.nctu.edu.tw/~gis83502/>

³ Email: smyuan@cis.nctu.edu.tw, <http://plato.cis.nctu.edu.tw/smyuan.html>

⁴ <http://www.cis.nctu.edu.tw/>

approaches in this field [2]. Whatever technology it uses, a virtual classroom is always the aspiration. Students in this virtual classroom need not gather at the same physical place at the same time, while the same learning effects may be obtained, or even better.

As the World Wide Web (WWW) grows rapidly, researchers begin to incorporate their pedagogical ideas into it. In this way educational system can benefit from the WWW technology, such as multimedia documents, hypertext/hypermedia capability, openness of the WWW world, and popularity of WWW client/server software and HTML authoring tools [3].

However, the WWW has several drawbacks from pedagogical points of view, because it was not designed originally for this field. Some, commonly known as navigation problems, are probably inherent in hypertext systems, as discussed by Conklin [4], Wright [5], and many others:

- *Disorientation*. Too much nonlinear information makes users “get lost in space”.
- *Cognitive overhead*. Multi-path and nonlinear structure require more cognitive concentration and decision.

There are still other important issues not supported directly by current WWW-based educational systems. For example, interaction and communication among instructors and learners are required to aid learning, problem solving, and experience sharing; logging of learners’ behavior is required for statistical and quantitative analysis; assessment functionality is also required to evaluate learning effects and system usability.

Various techniques have been proposed to handle these problems. Book-like overview maps, concept maps, fisheye views [6], etc. can help users understand the overall logical structure of large hyperdocuments. Guided tours, path mechanism [7], Footsteps [8], etc. can guide users to navigate through large hyperdocuments without too much cognitive overhead. In addition, many educational environments are equipped with text-mode, audio, and/or video facilities to improve interaction and communication.

They all help, but most of them are biased in favor of learners and consequently does not address the whole instructor/learner relationship well. With these systems, learners play an active role in the

learning process: they fetch, read, and select material independently. On the other hand, instructors are passive or even invisible: they merely construct course material, put them on the WWW, and then go behind the scenes, providing little positive aids to learners. Therefore, instructors are absent from the actual learning progress of learners. We call this phenomenon *student-centric*.

However, to address the whole instructor/learner relationship well, instructors must play a more positive and active role. With such initiative in hand, instructors may guide learners’ learning progress through various visual and vocal hints, just as they do in traditional classroom. Without such consideration, a pedagogical system on the WWW is no more than a fashionable bulletin board.

1.2. Our solution

We implemented an WWW-based educational environment, the Albatross, as described in our previous paper [1]. In addition to a standard WWW browser, it has several new facilities to address navigation problems. The Overview Map provides an organized road map of the whole course material to address the disorientation problem. The Guide Tool provides logging and prerequisite constraints; when combined with Navigation Hints, it can address partially the cognitive overhead problem. In addition, the Notebook provides public and private notes associated with individual documents so that each user may share ideas and experience. They all start to give positive aids beyond simple information presentation.

However, they are intrinsically prepared *before*, not dynamic and discretionary *during* the learning progress of learners. What we need is not only the former, but also the latter, so that the quality of learning process can be improved.

In this paper, a new mechanism for instructor-oriented navigation control is further proposed and implemented. An instructor can setup a learning group. Those who wish to be guided distantly during the study may join a specific group, and then their navigation behavior, such as document retrieval and scrolling, can be guided synchronously by the instructor. With this addition, some problems may be

partially resolved and are listed as follows:

- disorientation is decreased because the instructor can focus the group participants on a limited set of documents;
- cognitive overhead is eased because the learners can follow the instructor’s on-line guidance if they wish;
- interaction is improved because the instructor can make real-time visual hints and guidance;
- adaptability is enhanced because the instructor may adapt the learners’ navigation paths to fit in different situations.

The remainder of this paper is organized as follows. Section 2 introduces the concept of synchronous navigation control in the context of WWW-based distance education. Section 3 describes our implementation of such architecture and protocol in detail. Section 4 discusses possible application of this synchronization concept. Section 5 enumerates related work. Section 6 summarizes the paper and offers some future directions.

2. Concepts of synchronous navigation

This section explains some features not supported on current WWW system. It then defines terminol-

ogy, roles and activities involved, and finally discusses the mechanism and policy of synchronous navigation control.

2.1. Limitations of HTTP

Some strength and weakness of WWW can be traced back to its underlying transmission protocol: the Hypertext Transfer Protocol (HTTP). According to current HTTP/1.0 specification [9], participants in any HTTP connection are merely a pair of one client and one server of various kinds (original server, proxy, gateway, or tunnel). Normally, the client is the user agent requesting resources identified by Universal Resource Identifiers (URIs), while the server is the site holding or transferring resources.

This client/server model is adequate for basic data search and retrieval but inadequate for our needs:

- Inter-client communication is not taken into account in HTTP.
- HTTP is stateless, but synchronous control requires state-awareness.
- An HTTP connection is established at the request of the client and then closed as the server replies, but synchronous control performs better on a permanent connection.

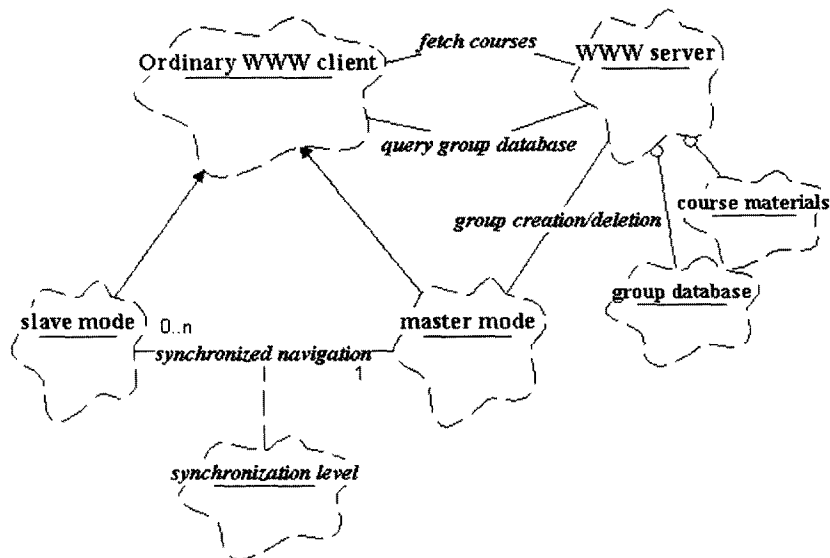


Fig. 1. Roles and relationships (represented in Booch notation).

Because of these restrictions, we propose an additional master/slave model which enables synchronous navigation control over learners on the WWW. This section describes new roles and relationships of participants in this model.

2.2. Roles

Participants' roles and capabilities in our new model differ from that in a traditional client/server model on which current WWW bases. Recall our motivation for synchronous navigation control. The instructor who wishes to guide others distantly may setup a learning group, and those who wish to be guided may join an existing group. In this way, the instructor may guide learners synchronously through navigation, visual and vocal hints. As the scenario goes, several new roles and relationships appear on the scene. To formalize and broaden the range of problem domain, we define the following terminology.

- *Client*: A program that establishes connections for the purpose of sending requests, just as defined on HTTP specification.
- *Server*: A program that accepts connections in order to service clients' requests for resources, as defined on HTTP specification.
- *Group*: A logical collection of on-line clients browsing on the same set of documents with synchronous navigation behavior.
- *Master*: A client that acts as a group leader; the role is often taken by instructors, moderators, leaders, or peer-tutors.
- *Slave*: A client that acts as a passive group participant; the role is often taken by learners, followers, or onlookers.

Fig. 1 illustrates the roles and relationships in the master/slave model. Among them, servers are repositories of course contents, logs, and other educational facilities, while clients are user agents used to fetch, read, and navigate through that hypermedia information. These roles remain unchanged as in a traditional client/server model.

Moreover, we introduce several new roles to support synchronous navigation. A client may turn into a master by registering a specific group on the server. Other clients can query on the server to find out existing group lists, something about group mas-

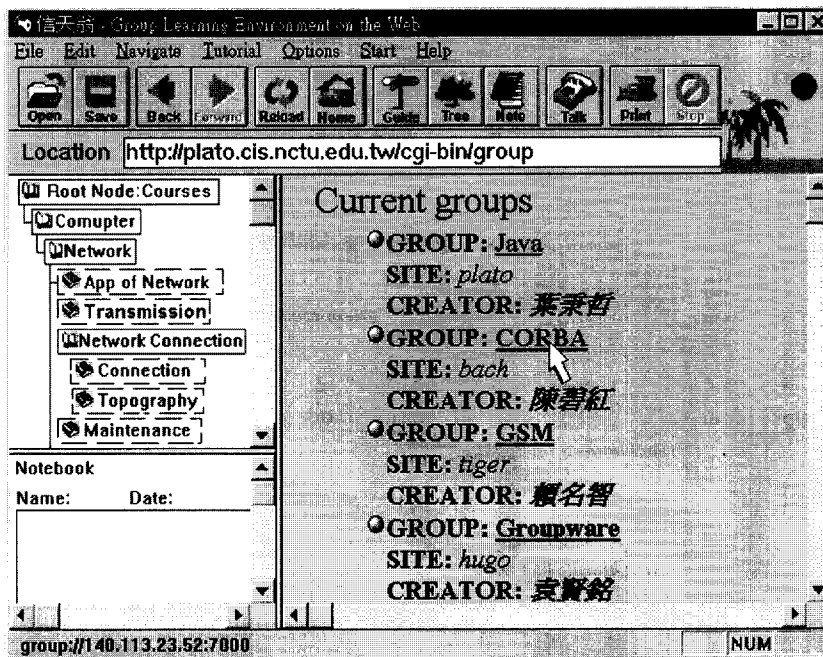


Fig. 2. A query of existing groups.

ters, and other related information, as shown in Fig. 2. If a client wishes to become a slave in an existing group, it can select any group of interest to join in by requesting the group master. When a master/slave connection is granted and established successfully, the client turns into a slave and its navigation is synchronized with the master.

After a master/slave connection is established, all control activities for synchronous navigation occur in this connection only, irrelevant to any server. Therefore, servers require no modification and they will not be the bottleneck.

Two issues about the master/slave connection deserve attention. First, since a master guides many slaves in the same group, the cardinality of relationship is one-to-many. Second, we model their relationship as an attributed association rather than simple multicast, because there may be different levels of synchronization according to their negotiations.

Being an attributed association gives individual variation and more flexibility for the whole system. We discuss the “mechanism versus policy” issue regarding this relationship in Section 2.4.

2.3. Which to synchronize?

The main purpose of synchronous navigation is for a master to guide on-line slaves immediately. Synchronous activities vary diversely in forms, ranging from full control of hardware to simple notification of software display. Instead of implementing plenty of synchronization, we focus only on those that are important for distance education.

From the learner’s perspective, hypermedia documents are obviously the objects and media of learning materials on the WWW, so they are naturally the target of navigation control. If we observe the learning behavior of WWW users, typical types of docu-

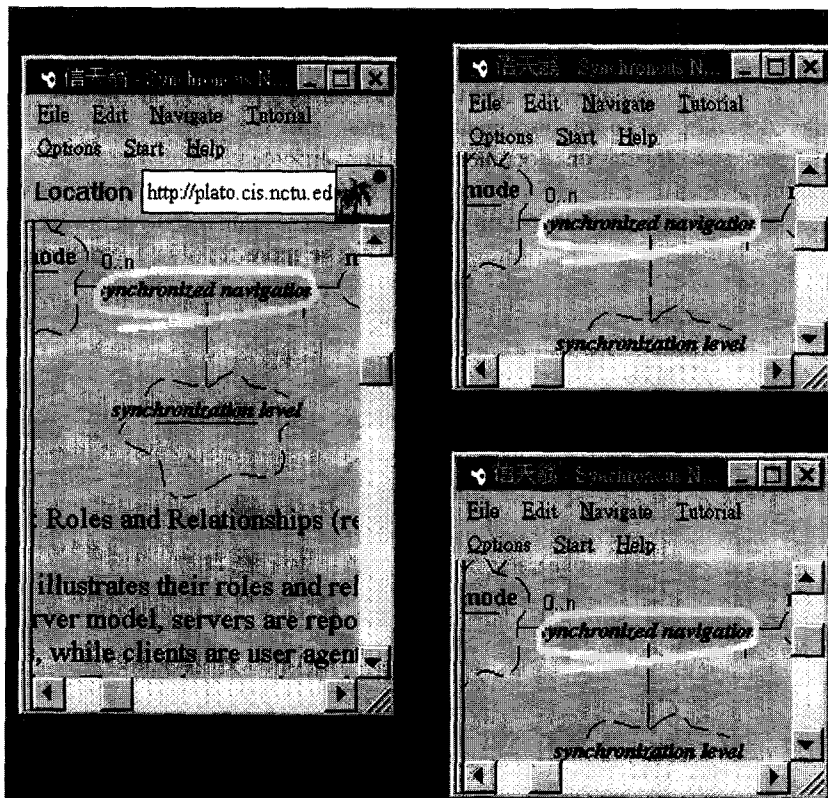


Fig. 3. Synchronous navigation between master and slaves.

ment navigation are easily identified:

- Access to a new object identified by an URL (Uniform Resource Locator). This may be done by keying in a new URL, by clicking on a hyperlink, or by selecting a bookmark/hotlist item from within the browser.
- Access to a visited object. This may be done by clicking on back/forward toolbar icons or menu items.
- Horizontal and vertical scrolling. This may be done by using function keys or scroll bars.
- Highlights on special texts or graphs. This is often done by positioning or dragging the mouse.

Therefore we implement the following synchronization activities:

- retrieval and display of common documents,
- synchronous document scrolling,
- mouse positioning, and
- common highlights and/or handwriting on documents.

Fig. 3 illustrates these synchronous navigation activities. Here, master A, slave B, and slave C are in the same group. Master A reads a document, and so do slaves B and C. Master A scrolls its document view horizontally and vertically, and slaves B and C scroll as well. Master A highlights on the document, and the same highlights also appear on document views of slaves B and C.

The underlying protocol is discussed in Section 3.

2.4. Mechanism versus policy

The mechanism of navigation control is better separated from the policy. The mechanism concerns protocols and procedures, while the policy concerns enforcement of synchronization rules between a pair of master and slave. Failure to distinguish between them will lead to the loss of flexibility and autonomy.

Pedagogical theories change from earlier behaviorism to recent cognitivism and constructivism. Course representations change from linear textbooks to nonlinear hypertexts. Distributed computing and distance education grow rapidly. These all reflect the trends of decentralization and autonomy [2,10]. Distance education values more at learner autonomy in its nature. Therefore, if a master (e.g., an instructor) puts exceeding synchronous control over its slaves (e.g., learners), the motivation and advantages of navigation control will be defeated by compulsion.

Separation of mechanisms allows more flexible learning behavior and makes self-paced learning possible, while on-line guidance still coexists. A master/slave pair may agree on an appropriate “synchronization level” according to the slave’s situation, and the policy decides the choices, contracts, and enforcement of various synchronization levels. Thus, the guidance with navigation control becomes assistance, rather than confinement.

The more flexible a synchronization policy is, the

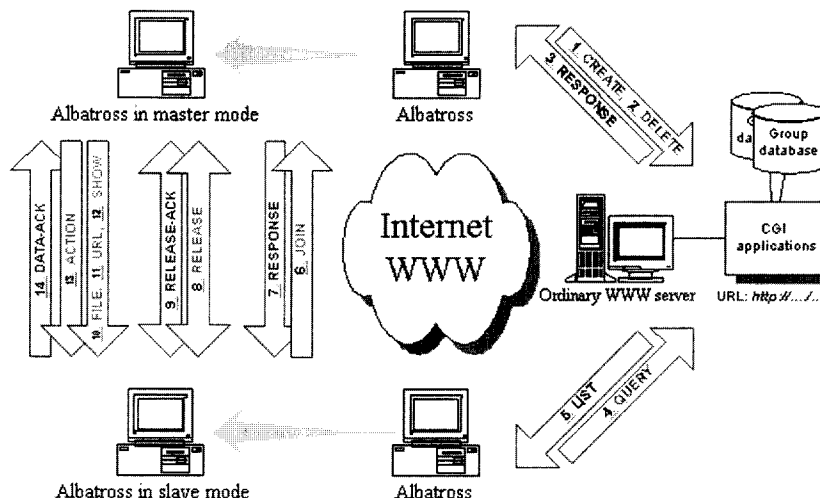


Fig. 4. Illustration of architecture.

more freedom learners may enjoy. An experienced learner, for example, often at will scrolls the document view and jumps to other URLs. If synchronized too much, he or she may suffer annoying “scroll wars” [11]. For such people, we may specify in the rule that their document scrolling and retrieval are synchronized only when the master forces it explicitly. In contrast, it is reasonable to synchronize most navigation activities of a novice.

“Which set of synchronization policies is more flexible, reasonable, and suitable for distance education?” Perhaps more experiments and analysis are required so we may answer the question.

3. Architecture and protocol

This section describes the overall architecture and detailed protocol of Albatross implementation.

3.1. Overview

The overall architecture and protocol of this system are depicted in Fig. 4. Note that each operation is associated with a number. The numbers correspond to the same numbers in Section 3.2 where we will discuss the operations in detail.

We can divide the protocol into three categories: group repository interface, per-group interface, and synchronous navigation interface.

Group repository interface (No. 1–5 in Fig. 4) concerns the creation, deletion, query, and other maintenance issues of the group database. The group database is stored in a public WWW server for WWW clients to query. If a client wants to be a master, it can register itself in the database; when it no longer wants to be, it can cancel the entry. All work on servers is done by server-side programs compliant with the Common Gateway Interface (CGI). Since communication in this category involves WWW servers, HTTP is used for underlying message transmission.

Per-group interface (No. 6–9 in Fig. 4) concerns connections between individual master/slave pairs. After query on group data (No. 4 and 5 mentioned above), any client who wishes to be a slave may

select a registered group and issue the connection request to the group master. The connection can also be released by either participant: the master or the slave.

Synchronous navigation interface (No. 10–14 in Fig. 4) concerns all activities of synchronous navigation control. This is the most important part of the 3 categories, so many alternatives to the same function are provided. For example, all requests contain an optional *WITH-ACK* field (as shown in Section 3.2 which enables assured synchronization mode between a pair of master and slave. Again the judgment is left as policy; we provide mechanisms only.

As for retrieval and display of common documents, three methods are provided for various transmission conditions:

- Direct data transmission (No. 10). The master packages up hypermedia data and transfers them directly to slaves.
- URL notification (No. 11). The master tells slaves the URL with which slaves can retrieve common documents. If transmission time is shorter for slaves to retrieve documents directly from the server, this method performs better.
- URL notification with proxy location (No. 11 with *PROXY* field filled). The master has fetched documents via the proxy, and tells slaves to fetch them via the same proxy.

In addition, three types of synchronization are provided to aid slaves with on-line and real-time guidance:

- synchronous document scrolling (No. 13, using *SCROLLTO* field);
- mouse positioning with text comments (No. 13, using *POSITION* field);
- common highlights and/or handwriting on documents (No. 13, using *BEGINDRAW/ NEXTDRAW* fields).

3.2. Protocol

We explain communication details as follows. To be consistent, we use the augmented Backus-Naur Form (BNF) and basic nonterminals specified in HTTP 1.0 specification [9] to define our protocol for synchronous navigation control.

1. Create a new group.

Message direction: from a client to a server.

Message format:

```
<HTTP headers>
  "CREATE" SP <group name> LF
  "SITE" SP <master site 's host:port> LF
  [ "ALIAS" SP <master site 's alias> LF ]
  "ID" SP <user ID creating the group> LF
  [ <miscellaneous> ]
```

2. Delete an existing group.

Message direction: from a master to a server.

Message format:

```
<HTTP headers>
  "DELETE" SP <group name> LF
```

3. Response to previous *CREATE/DELETE* requests.

Message direction: from a server to a client.

Message format:

```
HTTP-version SP Status-Code SP Reason-Phrase CRLF
  "Content-Type: text/html" CRLF
  *( <other full-response headers> )
  CRLF
  [ <ordinary data in "text/html" format> ]
```

Note: The client knows the status of its previous request from the *Status-Code* field in the HTTP message.

4. Query the group list.

Message direction: from a client to a server.

Message format:

```
<HTTP headers>
  "QUERY" LF
  [ <miscellaneous> ]
```

5. Return current group list.

Message direction: from a server to a client.

Message format:

```
<HTTP headers>
<ordinary data in "text/html" format>
```

6. Join in a group.

Message direction: from a client to a master.

Message format:

```

"JOIN" SP <group name> LF
"SITE" SP <host:port> LF
"ID" SP <user ID joining in the group> LF
[ <miscellaneous> ]

```

7. Response to previous *JOIN* request.

Message direction: from a master to a client.

Message format:

```

[ "OK" | "ERROR" ] LF
[ <reasons> ]

```

Note: If the response is *OK*, a master/slave connection is established, and the client will turn into a slave.

8. Release a master/slave connection.

Message direction: from a slave to a master, and vice versa.

Message format:

```

"RELEASE" SP <host:port> LF

```

9. Acknowledge previous *RELEASE* request.

Message direction: from a master to a slave, and vice versa.

Message format:

```

"RELEASE-ACK" SP <host:port> LF

```

10. Transmit the contents of a file identified by the URL.

Message direction: from a master to a slave.

Message format:

```

"FILE" SP <file 's URL> TAB <block size> [ TAB "WITH-ACK" ] LF
<a block of data>

```

11. Transmit "URL" only.

Message direction: from a master to a slave.

Message format:

```

"URL" SP <URL that the slave itself is told to fetch> LF
[ "PROXY" SP <WWW proxy that the slave is told to talk with> LF ]
[ "WITH-ACK" LF ]

```

12. Command the slave to show the data it has received so far.

Message direction: from a master to a slave.

Message format:

```

"SHOW" SP <URL which the slave was told to display> LF
[ "WITH-ACK" LF ]

```

Note: It is used when there is too much transmission delay on the slave.

13. Synchronize other navigation activities.

Message direction: from a master to a slave.

Message format:

```
"ACTION"      LF
[ "WITH-ACK"  LF ]
<action>
```

The *<action>* is enumerated as follows:

```
"SCROLLTO"    SP <point-x> TAB <point-y> LF
"POSITION"    SP <point-x> TAB <point-y> LF
[ "MESSAGE"   SP <message string> ]
"BEGINDRAW"   SP <point-x> TAB <point-y> LF
"NEXTDRAW"    SP <point-x> TAB <point-y> LF
```

14. ACK response to previous No. 10–13 requests.

Message direction: from a slave to a master.

Message format:

```
"DATA-ACK"    LF
{ "OK" | "ERROR" } LF
[ <reasons>   ]
```

4. Applications

Distance learning is certainly the motivation for our synchronous navigation control. Equipped with such facilities, knowledge exploration on distance education environment will not merely uni-directional and learner-biased. Instructors can play a more active and positive role in distance instruction.

The idea of navigation control applies not only to education, but also to many fields. It can be used, for example, to give a manual hypermedia presentation. It can also be used to do an automatic demonstration if navigation scripts are provided. It can even act as a part of an Intelligent Tutoring System (ITS) that is capable of adapting navigation paths to individual's learning behavior and circumstance.

5. Related work

Ideas about common display or, more broadly, about remote control are not entirely new. Some use special electronic equipment to get the total control at hardware level; some require specific operating

systems to achieve the desired effects; still others use proprietary applications that focus on a certain domain of problems.

Groupware is perhaps the most famous and mature application of this kind. Teleconferencing (e.g., the Colab [11]) and co-authoring (e.g., the DUPLEX [12]) have similar features of common display and annotation. However, they focus more on content collaboration, while our problem domain (distance education) focuses mainly on instructor-oriented navigational synchronization. Consequently, problems such as concurrency control and data serializability are not that serious.

Since the WWW is intrinsically distributed, collaboration over it becomes a hot topic [13]. There are also several conferences and workshops (e.g., the "ERCIM Workshop on CSCW and the Web" held on Feb. 1996) dedicated to it. We state here only typical work similar to ours.

The Yarn Web [14] extends an existing electronic meeting system to work with NCSA Mosaic. Its "common document display" feature is similar to ours. However, it uses functions not generally avail-

able on other platforms: remote display capabilities in X Window System, and remote control (though we think the term “forced document loading” is more appropriate) in X version of Mosaic. In addition, the Yarn server forks a child process on the server side for each client connection (execution of the child process can be displayed by remote sites through the X protocol), so higher load and resource consumption are inevitable.

Virtual Places(TM) [15] provides a virtual place where WWW users gather together, take on-line guided tours, and communicate with each other. It does have similar features to ours. However, its target is Internet on-line services, and therefore it lacks synchronization *within* documents and annotation found on most teleconferencing systems.

The WebDesk [16] is a collaborative environment that aims to support synchronous interaction (e.g., teleconferencing), asynchronous message exchange (e.g., electronic mail), and distributed document publishing homogeneously and integratedly. It looks promising judging from its scope and framework, though the project just begins.

Collaboration was not considered in original design of the WWW (as discussed in Section 2.1, so extensions to WWW clients, servers, protocols, or their combinations are required in order to support synchronous navigation. We embed synchronization mechanism directly into the WWW client, while most systems (e.g., the Yarn Web and Virtual Places) hook themselves on WWW clients. With the former, all synchronization mechanism is performed by the augmented WWW client; with the latter, synchronization and inter-client communication are performed by the hooked programs. We choose the former because currently there are no standard, portable, and powerful solutions with the latter.

Three popular extensions provide general mechanisms that had to be hard-coded into WWW clients before. Common Client Interface (CCI) by NCSA Mosaic [17] and Netscape Client APIs (NCAPIs) by Netscape Navigator [18] both allow external programs to command a WWW client to do something desired in a standard (or de facto) way. However, the commands are initiated by external programs, not by WWW clients. Therefore, communication between WWW clients, which occurs frequently during synchronization and collaboration, cannot be done ac-

tively by themselves, but by external programs only. On the other hand, Java by Sun Microsystems [19] does provide inter-client communication (which is impossible in JavaScript) through the `java.net.Socket` class family. However, event triggering occurs only on the frame windows of Java applets, not on those of a WWW client. Consequently, Java cannot synchronize the whole WWW client.

We may try to implement some functions with these extensions if more powerful they become.

6. Conclusion and future directions

In this paper, we stated several limitations of current WWW-based hypermedia systems for distance education, discussed the need for an instructor-oriented and master/slave model, and introduced a new idea of synchronous navigation control. We then proposed architecture and protocol for such functions, outlined possible applications of this concept, and surveyed related work.

We plan to support the model of problem-based cooperative learning in the context of distance education. Since our Albatross system is currently limited to a single master per group, we plan to turn it toward a more cooperative model. In this model, group participants take turns to synchronize others' navigation, so that they can share ideas and experience more conveniently and efficiently. In addition, facilities for shared and individual knowledge construction are also required from pedagogical points of view.

We believe that the aspiration of virtual classroom is further approaching with these additions.

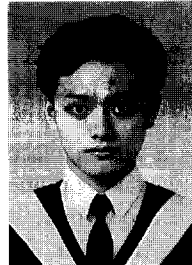
Acknowledgements

We would like to thank Hsin Pan and Shih-Ping Wang for reviewing this paper. We would also like to thank the anonymous referees whose helpful comments and suggestions improve the presentation of this paper.

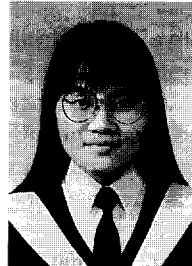
This work was partially supported by the National Science Council of Taiwan, R.O.C. under Contract NSC84-2511-S009-004CL.

References

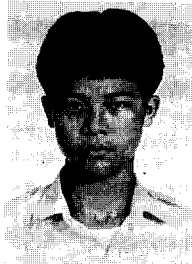
- [1] Ming-Chih Lai, Bih-Horng Chen and Shyan-Ming Yuan. Toward a new educational environment, in: Proc. 4th Internat. World Wide Web Conf., *World Wide Web J.* 1 (1995) 221–230. <http://www.w3.org/pub/Conferences/WWW4/Papers/238/>
- [2] Chao-Hsiu Chen and Chien Chou, The definitions, theories, and technology uses in cooperative distance learning, in: *Proc. 4th Internat. Conf. on Computer Assisted Instruction* (1995) 11–16 (In Chinese).
- [3] Bertrand Ibrahim and Stephen D. Franklin, Advanced educational uses on the World-Wide Web, in: Proc. 3rd Internat. World-Wide Web Conf., *Computer Networks and ISDN Systems* 27(6) (1995) 871–877.
- [4] Jeff Conklin, Hypertext: An introduction and survey, *Computer (IEEE)* 20(9) (1987) 17–41.
- [5] Patricia Wright, Cognitive overheads and prostheses: Some issues in evaluating hypertexts, in: *Hypertext '91 Proc.*, ACM, New York (1991) 1–12.
- [6] Emanuel G. Noik, Exploring large hyperdocuments: Fisheye views of nested networks, in: *Hypertext '93 Proc.*, ACM, New York (1993) 192–205.
- [7] Polle T. Zellweger, Scripted Documents: A hypermedia path mechanism, in: *Hypertext '89 Proc.*, ACM, New York (1989) 1–14.
- [8] David Nicol, Calum Smeaton and Alan Falconer Slater, Footsteps: Trail-blazing the Web, in: Proc. 3rd Internat. World-Wide Web Conf., *Computer Networks and ISDN Systems* 27(6) (1995) 879–885.
- [9] Tim Berners-Lee, Roy T. Fielding and Henrik Frystyk Nielsen, Hypertext Transfer Protocol – HTTP/1.0, Internet Draft (February 1996). <http://www.w3.org/pub/WWW/Protocols/HTTP/1.0/spec.html>
- [10] Martin Ryder, Instructional Technology Connections, (March 1996). <gopher://oasis.cudenver.edu/h0/UCD/dept/edu/IT/ryder/itcon.html>
- [11] Mark Stefik, Gregg Foster, Daniel G. Bobrow, Kenneth Kahn, Stan Lanning and Lucy Suchman, Beyond the chalkboard: Computer support for collaboration and problem solving in meetings, *Comm. ACM* 30(1) (1987) 32–47.
- [12] François Pacull, Alain Sandoz and André Schiper, Duplex: A distributed collaborative editing environment in large scale, in: *Proc. ACM 1994 Conf. on Computer Supported Cooperative Work* (1994) 165–173.
- [13] Karen MacArthur, Collaboration, knowledge representation and automatability, <http://www.w3.org/pub/WWW/Collaboration/>
- [14] Tak K Woo and Michael J Rees, A synchronous collaboration tool for World-Wide Web, in: *Electronic Proc. 2nd World Wide Web Conf. '94: Mosaic and the Web*. <http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/CSCW/rees/SynColTol.html>
- [15] Virtual Places(tm). <http://www.ubique.com/>
- [16] Peter Parnes, Dick Schefström and Kåre Synnes, WebDesk: The tele-conferencing service of MATES, Position paper for ERCIM Workshop on CSCW and the Web (February 1996). http://mates.cdt.luth.se/papers/SMAC/Webdesk_SMAC.html
- [17] Dave Thompson, Common Client Interface Protocol Specification (1995). <http://yahoo.ncsa.uiuc.edu/mosaic/ccii.spec.html>
- [18] Netscape Client APIs. <http://home.netscape.com/newsref/std/index.html>
- [19] Java. <http://java.sun.com/>



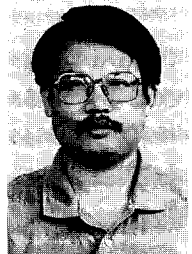
Ping-Jer Yeh is a Master student at the Institute of Computer and Information Science, National Chiao-Tung University⁵, Taiwan. He translated the comp.lang.c++ FAQ into Chinese, and is a regular contributor to local journals. His research interests include object-oriented technology, distributed systems, and programming languages.



Bih-Horng Chen is a Master student at the Institute of Computer and Information Science, National Chiao-Tung University, Taiwan. Her research interests include distributed and educational hypermedia systems. She is currently developing an Internet problem-based learning environment based on the Albacross system.



Ming-Chih Lai is an information officer in the Chinese army. He received his Master degree in Computer Science from National Chiao-Tung University, Taiwan. His current interests include distance learning, distributed information management, and educational hypermedia.



Shyan-Ming Yuan is a professor at the Institute and Department of Computer and Information Science, National Chiao-Tung University, Taiwan. He received the Ph.D. degree in Computer Science from the University of Maryland, College Park in 1989. His current research interests include distributed system design, fault-tolerant computing, CSCW, multimedia environments and ICAL in distance cooperative learning environment. Dr. Yuan is a member of ACM and IEEE.

⁵ <http://www.nctu.edu.tw/>