DOCUMENT RESUME

ED 427 668 IR 019 187

AUTHOR	Nakabayashi, Kiyoshi; Maruyama, Mina; Koike, Yoshimasa; Fukuhara, Yoshimi; Nakamura, Yukihiro
TITLE	An Intelligent Tutoring System on the WWW Supporting Interactive Simulation Environment with a Multimedia Viewer Control Mechanism.
PUB DATE	1996-10-00
NOTE	9p.; In: WebNet 96 Conference Proceedings (San Francisco, CA, October 15-19, 1996); see IR 019 168.
AVAILABLE FROM	Web site: http://aace.virginia.edu/aace/conf/webnet/html/321/321.htm
PUB TYPE	Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Computer Interfaces; *Computer Simulation; *Computer System Design; Courseware; Educational Technology; Information Technology; Instructional Effectiveness; *Intelligent Tutoring Systems; Interaction; *Multimedia Instruction; Multimedia Materials; *Systems Development; World Wide Web
IDENTIFIERS	Client Server Computing Systems; *Interactive Systems; Learning Environments; *Multimedia Technology

ABSTRACT

This paper describes the features of the 1996 version of an intelligent tutoring system (ITS) called CALAT. The architecture of CALAT is an extension of conventional World Wide Web systems, consisting of an ITS kernel on the server side and a multimedia viewer on the client side. The viewer control system is designed to achieve both educationally effective multimedia presentation and quick system response time even over a low speed network. As an application of the viewer control mechanism, an interactive simulation environment is implemented. This environment lets the student interact with a simulated target system presented by the multimedia viewer under the control of a state transition machine (STM) running on the CALAT server. These new features make it possible to build an interactive ITS environment available over the network with powerful and effective data presentation capability. The paper highlights design and implementation of the viewer control mechanism and interactive simulation. Three figures present the viewer control mechanism, the pipeline data transfer, and an example of the interactive simulation. (Author/DLS)

An Intelligent Tutoring System on the WWW Supporting Interactive Simulation Environment with a Multimedia Viewer Control Mechanism

Kiyoshi Nakabayashi
Mina Maruyama
Yoshimasa Koike
Yoshimi Fukuhara
Yukihiro Nakamura

An Intelligent Tutoring System on the WWW Supporting Interactive Simulation Environment with a Multimedia Viewer Control Mechanism

Kiyoshi NAKABAYASHI, Mina MARUYAMA, Yoshimasa KOIKE, Yoshimi FUKUHARA, and Yukihiro NAKAMURA

NTT Information and Communication Systems Laboratories
Tokyo 180, Japan
naka@nttaip.min.ntt.jp

Abstract: This paper describes the new features of an intelligent tutoring system(ITS) called CALAT. The architecture of CALAT is an extension of conventional WWW system, consisting of ITS kernel on the server side and multimedia viewer on the client side. A viewer control mechanism is newly developed to achieve both educationally effective multimedia presentation and quick system response time even over the low speed network. As a application of the viewer control mechanism, an interactive simulation environment is implemented. This environment let the student interact with a target(simulated) system presented by the multimedia viewer under the control of a state transition machine(STM) running on the CALAT server representing the internal behavior of the target system. These new features make it possible to build an interactive ITS environment available over the network with powerful and effective data presentation capability.

1 Introduction

Since its capability as a globally distributed hypermedia is so attractive, the World-Wide Web(WWW) [Berners-Lee, 1994] has been widely accepted by not only computer scientists but the people in the various fields. One of the important application of the WWW is for education. There have been many proposals to use the WWW in a educational environment [Kay & Kummerfeld, 1994] [Ibrahim 95] [Schwarz et al, 1996].

We have proposed CALAT [Nakabayashi et al., 1995], an intelligent tutoring system(ITS) on the WWW. The system consists of an WWW server integrated with a tutoring system called CAIRNEY [Fukuhara & Kiyama, 1993] [Fukuhara et al., 1995] which has been developed as a standalone ITS, and WWW clients equipped with a multimedia scene viewer. The CALAT system features the mechanism to identify the user on the client side from the sever side over the stateless protocol of the WWW. This mechanism makes it possible to implement substantial ITS function to present appropriate courseware pages by taking into account of the user's answers to the exercises, progress speed, and other characteristics observed from the interaction history.

Although the first version of CALAT is actually implemented and open to the public on the Internet, there are several limitations and problems:

- Size of courseware page consisting of animation and audio data is too large to be transfered over the low speed network. Sometimes it takes more than a few minutes to display the page, making learning substantially impossible.
- Learning style is limited to explanation and exercise based. There is no support for interactive simulation which, if available, could provide a very effective tutoring environment by the combination with the explanation-based tutoring.
- It is not possible to utilize existing HTML documents and other type of multimedia data distributed over the Internet.

The design goal of new CALAT system is to provide solutions for these issues. This paper concentrates on the first two topics. The third topic is dealt with in another paper [Maruyama et al., 1996]. A viewer control mechanism is newly developed for enabling fast system response possible even for the courseware page of large data size transmitted over low speed network. The mechanism also provides a sophisticated viewer control facility important to realize effective tutoring. Exploiting the viewer control mechanism, an interactive simulation environment is implemented. In this environment, the internal behavior of the target(simulated) system is represented by a state transition machine(STM) running on the CALAT server. The STM fully takes control of the multimedia viewer on the client side by means of the viewer control mechanism so that the student can experience how the target system responds to his/her operation.

The following sections, the design and implementation of the viewer control mechanism and the interactive simulation environment are described.

2 Viewer Control Mechanism

2.1 Design Issues

Multimedia data presentation is quite important for the CAI system. There are several crucial issues to support effective multimedia presentation in the WWW environment.

One issue is the data transfer speed over the network. The amount of multimedia data can be so large that the system's response to the user may be quite slow. In the case of CALAT, one standard courseware page consists of about 10KByte animation data and a few hundred KBytes audio data replayed synchronously. Since these data are packed in one huge archive file [Nakabayashi et al., 1995], the users on the ISDN or telephone line should wait for a few minutes until the requested page is downloaded.

Moreover, current implementation can not take advantage of the data caching mechanism equipped in the WWW proxy server or browser. This is because the each courseware page is selected by the CALAT server responding to the ``same'' URL request from the client saying ``send me the next page'' or ``give me one more hint''. This behavior is essential to implement the adaptive tutoring capability of CALAT with which the courseware pages are dynamically selected based on the student status. However, The proxy server or browser cannot cache the previously downloaded pages since the URL's corresponding to them are all identical and not represent specific data.

Another issue is the viewer control from the server. In the conventional WWW scheme, it is very difficult to make an effective presentation using multiple client window, since there are no means for the server to take control of the external viewers (or helper applications) spawned by the WWW browser. It is important for the effective tutoring since since sometimes it is good for the student to achieve a good understanding by looking at multiple images at a time or sometimes concentration on one scene may result in much progress.

2.2 Implementation

The previous version of CALAT has the mechanism to partially solve the second issue mentioned in the previous section. The new version provides an extended scheme which is designed to overcome the above two issues without modifying the conventional WWW browser and protocol.

The key component of the extended scheme is a control script sent from the CALAT server to the client as the response to the student's courseware page request. On receiving the URL from the client saying ``send me the next page'' or ``give me one more hint'', the CALAT server assembles the control script taking into account of the student status. The WWW browser on the client receives the script and invoke a viewer controller program which interpret the script. The script contains the following information for presenting the courseware page:

- The type and the identifier of the viewer to be used to display the page. If the viewer of the

specified type and identifier is not running on the client, new one will be invoked. If it is already running, its current display data will be thrown away and the new data will be loaded. It is also possible to close the specified viewer if necessary.
- A list of possibly multiple URL's specifying the multimedia data composing the page. The viewer controller requests these URL's by means of the WWW browser's remote control facility such as CCI [NCSA, 1995] for Mosaic or NCAPIS [Netscape, 1995] for Netscape. The received multimedia data will be passed to the viewer specified in the first item.

With the capability described in the first item, it is possible for the server to take control of multiple viewers on the client. For example, the new version of CALAT is able to open a new viewer for the detailed explanation of a certain keyword, or to automatically close every viewer except the browser when the student comes to an exercise page [Fig.1].

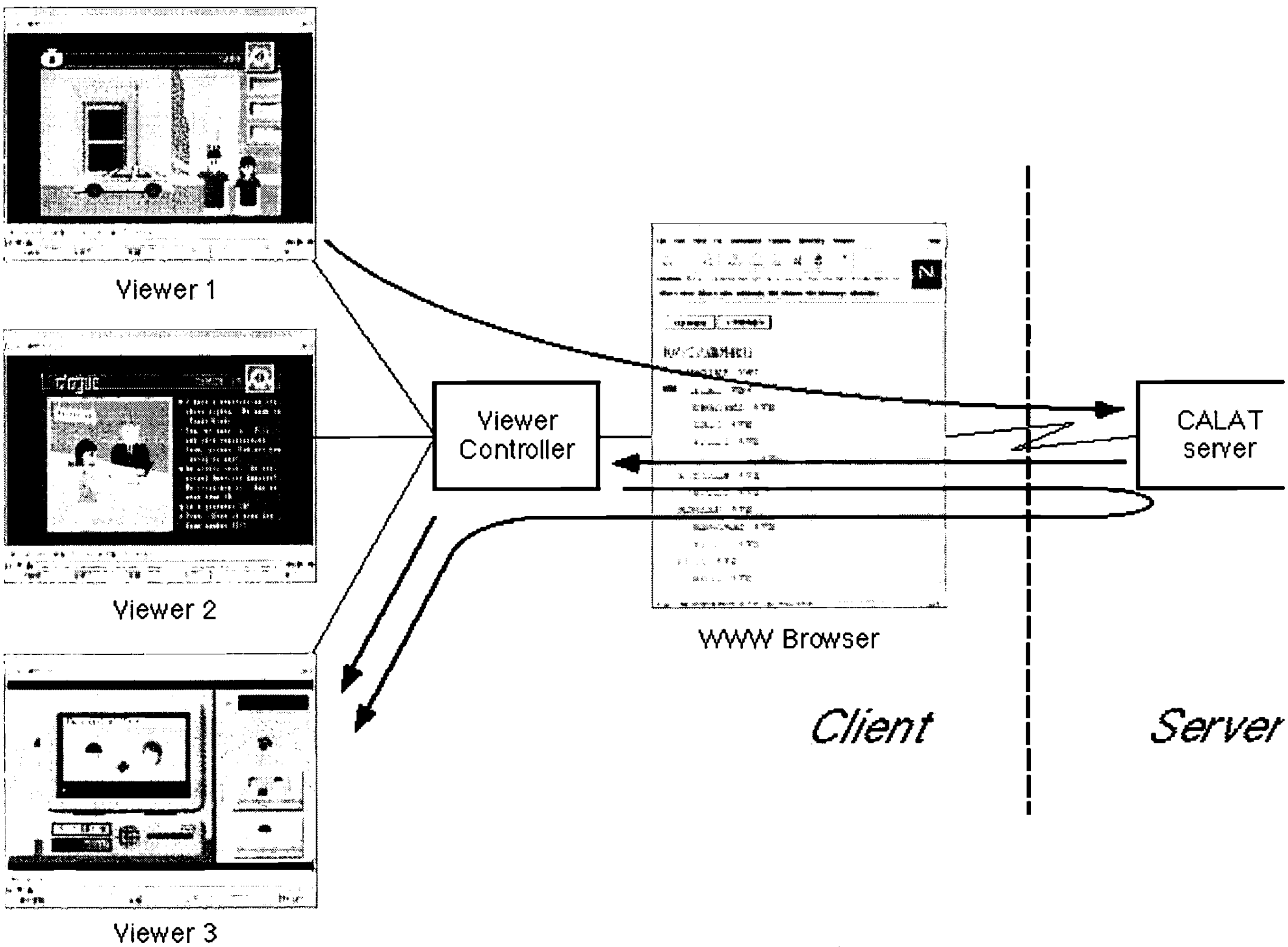

Figure 1. Viewer Control Mechanism

The capability of the second item allows to construct a courseware page consisting of multiple multimedia data. One application of this feature is the interactive simulation described in the following section, in which it is required to display image and sound simultaneously as the response to the user's single action.

Another application of this feature is the ``pipelined processing" mode to make the system response faster[Fig.2]. In the pipeline processing mode, a few hundred KBytes audio data belonging to a CALAT courseware page will be splitted into the small (usually some ten KBytes) chunks. URL's of these audio data chunks together with the animation data will be listed in the control script. The viewer controller accesses these URL's one by one and forward the received data to the viewer. As soon as the viewer receives the animation data and the first chunk of the audio data, it starts to replay them. At the same time the viewer controller keeps accessing the succeeding audio data. Thus the data transfer over the network and replay on the client are performed in a pipelined fashion. With this pipeline processing mode, it is possible to drastically shorten the system response time especially in the case that the network speed is slow and/or the amount of multimedia data in a page is large.

Moreover animation and audio data can be cashed by the WWW proxy server or the browser, since they are accessed with its specific URL's with which the WWW proxy server or the browser distinguish each of them. Thus when the student go back to the previous page, the control script for the previous page comes from the server but the multimedia data composing the page may come from the cache with high probability. This results much more faster response.

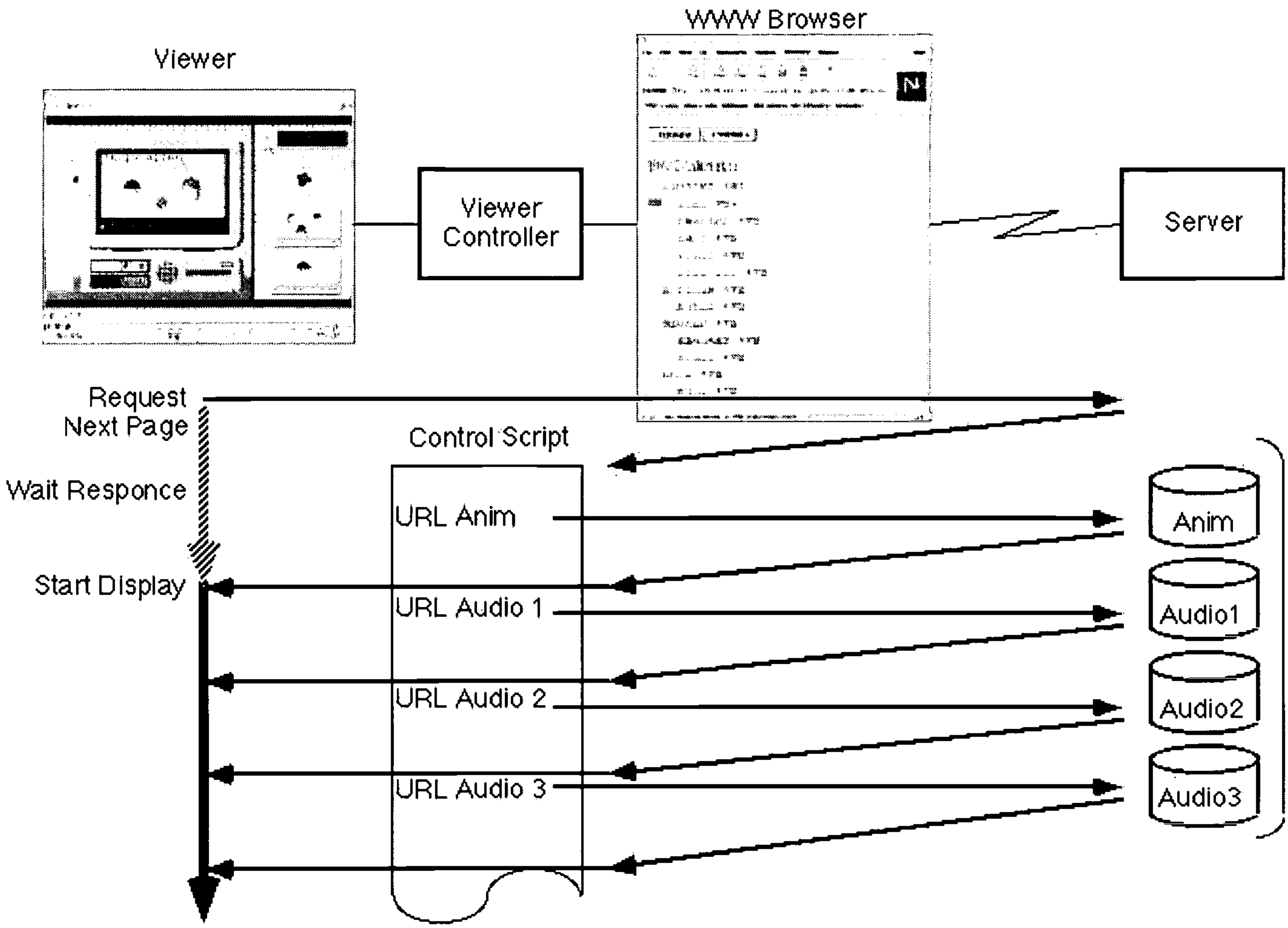

Figure 2. Pipeline Data Transfer

It should be noted that the proposed control script scheme works completely in the conventional WWW environment. Since the WWW browser deals with the control script in a same way as a usual multimedia data displayed using the helper application, the viewer controller program is just registered as a helper application to be spawned when the browser receives the control script. Also the remote control facility used to access URL data by the viewer controller is supported in the commonly used WWW browser such as Mosaic and Netscape. Thus using the control script poses no special modification of the WWW browser or protocol.

3 Interactive Simulation

3.1 Background: CAIRNEY Simulation

An interactive simulation is one of the most important facility to make effective tutoring possible.

CAIRNEY [Kiyama & Fukuhara, 1993], the standalone ITS on which CALAT is based, has the simulation capability mainly for the training of operation procedures to deal with the telecommunication network equipments.

In the CAIRNEY simulation environment, the behavior of the simulated(or target) network equipment is described as a state transition model(STM). The STM consists of several states, the input event which triggers the state transition, and the output actions taking place when certain state transition occurs. The input event and output action of the STM are linked to the user interface of the CAIRNEY simulation environment. The user interface window consists of simple GUI(graphical user interface) objects(widgets) such as buttons or key-input fields, and multimedia outputs such as animated images or sounds. The user's click or type on the widgets causes the event input to the STM, and the output actions from the STM change the color of the widgets, redraw the window image or make sound output.

In the old version of CALAT, above mentioned simulation facility is not implemented because the conventional WWW framework is not adequate to support the required user interface. Standard WWW browsers provide only anchor, form input and clickable map as the ways which user can interact with. There are no concept of ``widget''. It is also difficult or impossible to make effective output such as partially modifying the previously displayed image, or redrawing the image while playing audio simultaneously. (Even recent technology such as VRML does not support ``widget''. Java could be a solution but not very popular at this point of time.)

3.2 Design and Implementation

There are two major design issues to implement CAIRNEY-style simulation on CALAT environment. One issue is how to divide the function composing the entire simulation between the server and client. Another issue is how to implement ``widget'' in the WWW client environment.

Two main alternatives can be considered for the first issue. One is to implement both STM and GUI on the client. It gives maximum response performance for the user's mouse click or key input. The drawback is that we have to prepare STM for every client platform. Moreover the CAI kernel on the server and STM are rather isolated, making it difficult to give pedagogical intervention from CAI kernel. The other alternative is to implement the STM on the server and the GUI on the client. In this alternative, the response speed may be slower than the first one. However it is much much easier to make the simulation environment available on the multiple client platform. Since there are actual requirements to use CALAT system with several client machine, the second alternative is taken for less development effort.

For the second issue on how to implement interactive widgets, we fully take advantage of viewer control mechanism described in the previous section. We developed the multimedia viewer which works together with the viewer controller. The multimedia viewer is able to display the animation/audio data sent from the viewer controller as well as to manage widgets on the window. Mouse click or key input event on the widgets are sent to the viewer controller, then the viewer controller translates it to the URL form and send it to the CALAT server. This will trigger the state transition of the STM on the server. As the result of the state transition, the server generates the viewer control script and send it back to the client making changes on the window and/or playing audio. With this mechanism, CALAT provides an interactive simulation environment available on the network-based server/client system of the WWW. Example of the interactive simulation is shown in [Fig.3]

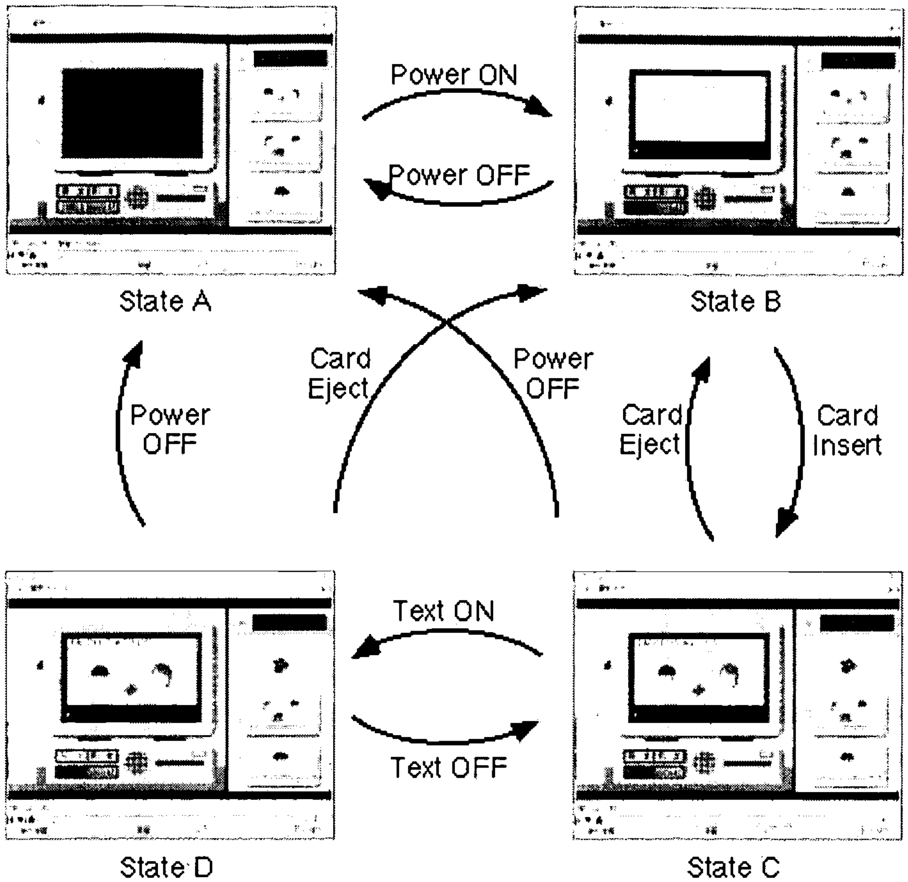

Figure 3. Example of the Interactive Simulation

4 Conclusion

New version of CALAT, an ITS on the WWW is described. It is equipped with several novel features including a viewer control mechanism and an interactive simulation facility. The viewer control mechanism make is possible to achieve a fast response speed even over low speed network as well as to provide sophisticated multimedia presentation. The simulation environment is implemented by fully exploiting the viewer control mechanism. It let the student interact with a target system managed by the STM on the CALAT server over the network with powerful and effective multimedia presentation.

References

[Berners-Lee, 1994]
Berners-Lee, T., et. al. (1994). The World-Wide Web. *Communications of the ACM,* 37(8), 76--82.

[Schwarz et al, 1996]
Schwarz, E., Brusilovsky, P., & Weber, G. (1996) World-wide intelligent textbooks. *Proceedings of the the ED-MEDIA 96.*

[Fukuhara et al., 1995]
Fukuhara, Y., Kimura, F., Kohama, C., & Nakamura, Y. (1995). A Knowledge-based Educational Environment Integrating Conceptual Knowledge and Procedural Knowledge in Telecommunication Service Field. *Proceedings of the ED-MEDIA 95,* 229--234.

[Fukuhara & Kiyama, 1993]
Fukuhara, Y., & Kiyama, M. (1993). A Knowledge-based Educational Environment for Real Business Settings. *Proceedings of the World Conference on Artificial Intelligence in Education.*

[Ibrahim 95]
Ibrahim, B. and Franklin, S.D. (1995). Advanced Educational Uses of the World-Wide Web,

Proceedings of the Third International World Wide Web Conference '95: Technology, Tools and Applications

[Kay & Kummerfeld, 1994]
Kay, J. & Kummerfeld, R.J., (1994). An Individualized course for the C programming language. *Proceedings of the Second WWW Conference.*

[Kiyama & Fukuhara, 1993]
Kiyama, M. & Fukuhara, Y., (1993). An Authoring System for Creating a Practice Environment in the Network Service Field. *Proceedings of the International Conference on Intelligent Computer-Aided Training and Virtual Environment Technology.*

[Nakabayashi et al., 1995]
Nakabayashi, K., Koike, Y., Maruyama, M., Touhei, H., Ishiuchi, S., & Fukuhara, Y. (1995). A distributed Intelligent-CAI System on the World-Wide Web. *Proceedings of the ICCE 95,* 214--221.

[NCSA, 1995]
National Center for Supercomputing Applications (1995). NCSA Mosaic Common Client Interface. http://www.ncsa.uiuc.edu/SDG/Software/XMosaic/CCI/cci-spec.html.

[Netscape, 1995]
Netscape Communications Corporation (1995). Standards Documentation. http://home.mcom.com/newsref/std/.

[Maruyama et al., 1996]
Maruyama, M., Nakabayashi, K., Koike, Y., Fukuhara, Y. & Nakamura, Y.(1996). An Approach to Implement Adaptive Hypermedia for an Intelligent Tutoring System on the World-Wide Web. *Proceedings of the ED-MEDIA 96,* 373.

NOTICE

REPRODUCTION BASIS

☒ This document is covered by a signed "Reproduction Release (Blanket) form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").

EFF-089 (9/97)