Intelligence on the Web?

Mia Stern, Beverly Park Woolf, and James F. Kurose

Computer Science Department
University of Massachusetts
Amherst, MA 01003
Phone: (413) 545-0582
Fax: (413) 545-1249
email: {stern, bev, kurose}@cs.umass.edu

Abstract. This paper describes a courseware system delivered over the World Wide Web for transforming existing video-taped courses. The on-line courses take advantage of the interactivity and the individualized instruction of intelligent tutoring systems. They use synchronized audio and HTML slides to present material to the student. An overlay student model records a student’s ability on each topic in the system, determined by which slides the student has seen and quizzes he has taken. This model is then used to help guide the student through the material, to provide interactive and adaptive quizzes, and to prefetch course material before it is explicitly requested, reducing the delays seen by the student. The first goal is accomplished by using adaptive hypertext techniques including adaptive navigation support and adaptive presentation. The second goal is achieved by dynamically constructing quizzes at the correct level of difficulty for the student. The third goal is accomplished by predicting the next actions the student will take, and preparing for them ahead of time. In this paper we discuss initial efforts to implement a student model for the kinds of web-based courses supported in the system. We also discuss how the student model is used to accomplish the three goals. We conclude by providing a brief description of the first student encounters with the system.

1 Introduction

Many web based educational systems are equivalent to computer assisted learning systems, since they do not customize material to meet individual student needs. These systems are usually static and do not reason about the student, and thus are unable to personalize the interaction. MANIC (Multimedia Asynchronous Networked Individualized Courseware), on the other hand, is an intelligent tutoring system (ITS) delivered over the World Wide Web (WWW), which can customize the learning experience for each student. Courses using the MANIC system consist of synchronized audio and HTML slides as well as interactive quizzes. With the MANIC project, we are using existing technologies (such as WWW browsers and plugins, including RealAudio) and are augmenting these technologies with specialized WWW servers and proxies to provide a more individualized learning experience.

Although the audio and slides for a MANIC course are taken from existing video-taped courses, MANIC courses are not simply direct translations of those courses. By designing a course to be delivered over the WWW, it can be considerably more interactive, thus allowing the student to take more control over his learning. Students interacting with the MANIC system have several options for viewing the material. One option is to start the audio playback and let it continue until the end of the lecture. Because we are using the RealAudio client, we have synchronized the playback of the audio with the student’s viewing of the slides. Another option is for the student to control the playback of the audio, stopping and starting whenever he chooses. A third option is to “randomly” traverse the material,
using the table of contents provided as a guide. Additionally, some slides contain links to prerequisite material, which the student can follow if he needs more background information. Finally, at any point, students can elect to take on-line quizzes to test themselves on the material. A more detailed description of the initial implementation of MANIC can be found in [13].

We have added a “tutor” to adapt MANIC courses to customize the material to students’ needs and preferences. The intelligent tutor is used to guide the student through the course material, to provide appropriate quizzes, and to prefetch material before the student requests it. The first two activities help individualize the course material for the student; the last reduces delays experienced due to downloading material across the network.

The MANIC system is currently being used in conjunction with a 3-credit computer science course on computer networks. Additionally, the it was used as part of a 1-credit course taught during the Fall of 1996 on Unix network programming.

The rest of the paper is organized as follows. In section 2, we describe the basic architecture of the MANIC system. Section 3 presents the intelligent aspects of the system. And section 4 concludes the paper.

2 Basic System Architecture

In this section we present the basic system components of MANIC, including the architecture, the domain, and the student model.

2.1 Main components

The MANIC system has five main components: the client side software, the proxy, the HTTP server, the port server, and the student model servers (see Figure 1). The decisions made at the student model server are the main focus of this paper. See [14] for details on the other parts of the architecture.

The client side software consists of a web browser (such as Netscape Navigator) and the RealAudio client. Both of these are “off-the-shelf” software and easy to obtain.

The proxy is an application that runs between a web client and a web server [9]. All requests from the client are processed by the proxy before being sent to the server. Similarly, server responses are processed by the proxy and then sent to the client. The proxy is included in the architecture to enable prefetching; data must be stored in a cache so that it can be served to the client. See [10] for more details on the proxy implementation.

The HTTP server is also standard, but we have augmented it with Common Gateway Interface scripts. These scripts serve as a front end to the ITS; their job is to contact the student model server with a re-
quest and to deliver the results generated by the student model server back to the client. The HTTP server also works with the RealAudio server to provide the audio for playback.

The port server’s role is to control the creation of student model servers (there is one server per student). The student model server is the ITS, making decisions about its student, which it then communicates to the HTTP server. For example, the student model server generates the slide to be seen by the student. There is one student model server for each student using the software.

2.2 Domain organization

The domain is structured in terms of “topics,” with each topic consisting of a set of slides and their associated audio. Topics that have some influence on other topics (e.g. prerequisite topics or related topics) are linked together, with the value on the edge the weight of this relationship. The links are directional; they originate from prerequisite topics. For example, if topic A points to topic B with a weight of 0.75 (on a 0 to 1 scale), then topic A is a fairly important prerequisite to topic B.

Topics that are closely related will have higher weights, while those that are not closely related will have lower weights. This graph can be considered to be a fully connected graph, with most weights at 0. The weights, which represent the ideal way of proceeding through the course, are determined a priori by a domain expert. Unlike traditional ITSs that determine what students see (for example, the LISP tutor [1] and the Cardiac tutor [6]), the MANIC system does not strictly impose presentation of prerequisite topics before a new topic is started.

Additionally, we have extended the existing course by providing alternative ways to present essentially the same material. This is done by providing easy, medium, and difficult versions of each slide and associated audio. The easy versions of the slides provide more detail and background information, while the difficult versions assume the student knows that background information. For example, the easy version of a slide might provide vocabulary terms with their definitions, while the harder version simply presents and uses the terms without defining them.

There is not necessarily a one-to-one correspondence between the number of easy slides to teach some material versus the number of difficult slides. What can be taught with one difficult slide may require more than one easy slide. However, the multiple easy slides are considered as one grouping, which essentially enforces a one-to-one mapping between the versions of the slides.

Each topic is broken down into smaller “concepts” which cover just one or two slides. For example, one topic might be socket system calls, while the bind system call is a concept within that topic. We have elected to use at least these levels of granularity, since students may not be viewing the course in terms of entire topics. Rather, they may simply want to explore one concept for advice on a homework assignment. Thus the concept distinction allows the tutor to focus in on a smaller part of the course.

2.3 The student model

The student model in MANIC is essentially an overlay model [5]. Students must gain a certain degree of knowledge on each topic. No buggy knowledge is tracked, as the system is simply attempting to guide the student into gaining the desired knowledge.

The MANIC student model tracks a student’s “ability” on each topic within the domain. This ability is represented as five scores. The first score records how much of the topic has been viewed. This is determined by how many slides a student has seen.

The second score is determined by which version of the material the student has seen. If the student sees the harder version, his ability on the topic is more highly rated.

The third score records the student’s access patterns for the slides. If he chooses to review some of the slides within the topic, then he probably does not understand it as well as someone who required no review. Also if he did not listen to the associated audio, he may have missed some of the information, thus reducing his comprehension of the material.

The fourth score records whether the student has followed any of the hypertext links on any of the slides to review the topic. If so, then he possibly did not fully understand that background topic.

Finally, the fifth score is the student’s quiz performance on questions associated with the topic. How
many questions the student got right and wrong, and their levels of difficulty, indicate directly how much the student knows about the topic.

3 Decisions at the Student Model Server

The MANIC intelligent tutoring system customizes the teaching material in three ways. In this section, we describe each of these in turn.

3.1 Guiding the student through the material

Work in adaptive hypermedia has identified two methods for adapting the material in a course to each individual student: adaptive presentation [2] and adaptive navigation support [7]. The first technique provides different course content for different students, while the second changes the links that are visible from a given state. We use both techniques in the MANIC system.

When guiding a student through the course, the tutor assumes that the student is viewing the course as we are, in terms of topics. However, this may not be the case. The student may be using the material, for example, as a review. In this case, the student may not view most of the slides for a single topic. Rather, his pattern for viewing the material will be inconsistent and appear somewhat random. Guiding the student in this case may not be possible.

Although we do not want the tutor to control how a student progresses through the material, we do want it to be able to help guide the process. There are a few ways it can accomplish this, including suggesting topics a student should view and deciding which versions of the slides the student should see.

Suggesting topics: When a student finishes a topic, he must choose a new topic on which to start. At this point, the student is presented with a list of all available topics. This list is augmented with suggestions from the tutor indicating which topics the student should view. The tutor thus provides adaptive navigation support. This is similar to the mechanism used in ELM-ART [4] and in the ISIS tutor [3]. As in those systems, the student has the option of either taking or ignoring the tutor’s suggestion.

To accomplish this topic suggestion, the tutor examines the scores the student has on the topic that has just been completed. The scores that are considered, in order of importance, are the quiz grades on the topic, how much of the topic was seen, how the material was viewed, and which version of the material was seen. If these scores are sufficiently high, the tutor suggests the student move on to the next topics. These next topics are the ones that have the highest weights from the current topic in the topic net. Furthermore, when evaluating a possible next topic, the tutor examines other prerequisite topics, checking to see if enough prerequisite knowledge has been gained to suggest this next topic.

If the student should not move on to learn new material, remedial topics are suggested, which are topics that cover material the student did not seem to understand. These topics include those on which the student has low quiz grades as well as those topics the student has seen but has not mastered. The prerequisite topics for these two groups are also suggested.

Finally, the tutor also suggests viable next topics for any topics the student has passed, using the criteria just described. Since there are potentially many paths for traversing the material, this option allows students to view those other paths.

It should be noted that unless students take quizzes, the tutor does not have much data on which to judge them. Therefore, its predictive abilities are limited. In this case, the tutor can suggest prerequisite topics that have not been covered, but it will have no basis to judge if remediation on topics not understood is necessary.

Determining which versions of the material: The student does not control every aspect of the learning. The tutor decides the content of the slides the student sees and the audio he hears, based on his scores on the topics. By doing so, the tutor uses the adaptive presentation technique for customizing the material.

When deciding which version of slides to give, the tutor examines the student’s score on quiz mate-
rial for this topic, which versions of the slides he has seen in the past, which versions of the prerequisite material he has seen, and how many prerequisite topics he has studied. Essentially, the tutor is predicting, from past behavior, the student’s scores on the topic. For example, if a student has been doing well on all of the quiz questions and he has seen the hard versions of all prerequisite topics, then he should see the harder versions of the slides. On the other hand, if a student jumps ahead to a more difficult topic for which there is no evidence that he is ready, he will be provided with the more detailed version of the slides, which provides more information on the prerequisite knowledge.

Furthermore, if a student starts a topic as part of remediation, then he should be given an easier version of the slides, if he had seen the material previously. Remedial work should provide as much support to the student as possible, and giving material that includes more detail accomplishes this goal.

The tutor may not be completely accurate in its assessment of the student’s ability and thus it may choose the wrong versions of the slides. To compensate for this, the tutor is designed to take a conservative approach. Students will generally see slides that are either at the right level of difficulty or easier, with the theory that more detail is better than less detail.

Even with this philosophy, mistakes will be made. For example, if a student has passed the prerequisite topics for a given new topic, he still might not know all of the material. But if he is given the harder versions of the slides, without as much explanatory information, there are still links to retrieve those details. Then if the student chooses to view those links, the tutor must reevaluate its judgment of the student, downgrading its view of his ability.

3.2 Quizzing the student

Students have the opportunity to take on-line quizzes at any time while using MANIC, either following the tutor’s suggestion or choosing to take a quiz on their own.

Types of questions: The MANIC system supports three types of questions: true/false, multiple choice, and short answer. The instructor for the course enters the question text and the answer, and in the case of multiple choice questions, the distractors. Currently the system does not dynamically generate the question text and answers, and we do not see that capability being incorporated any time soon.

Associated with each question is a concept that the question is testing as well as a “level of difficulty.” For example, a multiple choice question with obvious distractors is easier than a short answer question (generating the answer is usually harder than being able to “guess” the right answer).

Creating a quiz: The tutor dynamically decides, from a question database, which questions should be asked. The tutor reasons about: (1) picking questions for topics the student has just seen (and their prerequisites) and (2) picking questions for review. The factors which influence the tutor’s decisions about questions to ask are concerned with the concepts describing the topic on which the student is currently working. These factors include:

- Which versions of the slides have been seen - If the student has seen the difficult versions of the slides, then he should see harder questions than someone who has seen the easier versions.

- How frequently has the material been viewed? - If the student is viewing the slides for the second or third time, based on the suggestion of the tutor, then it is unlikely he has fully mastered the concepts. Therefore, the questions he is asked should not be at the highest level of difficulty. However, if he is simply reviewing the material on his own, then he should be given harder questions, due to his exposure to the material.

- The student’s scores on prerequisite topics - If the student has understood the prerequisite information, then there is a higher probability that he understands the concept being tested. Therefore he should receive harder questions.

- Other questions on this concept - Has the student answered other questions on this concept, either correctly or incorrectly? This should clearly impact which questions the student should see this time around.
• General test performance - If the student, in general, does not perform well on these quizzes, then he should not be challenged as much.

Each of these factors is considered when choosing questions to ask on a particular concept. The tutor gives higher priority to those questions that have not already been answered correctly. However, if there are no new questions on a given concept, one question is chosen as review.

The tutor may also choose to ask questions about topics on which the student has not demonstrated sufficient mastery. In this case, the tutor looks at the scores for all topics. If the quiz score on a topic is below a certain threshold, then the student should definitely be asked questions about the concepts within that topic. The tutor decides which questions to ask using the factors given above, with the additional criteria that they be easier, if possible, than the questions asked previously. The student will be asked review questions, both on material on which he has been quizzed, and on material not yet quizzed but viewed some time in the past.

Likewise, the tutor may elect to do some review work on topics on which the student has done well. If the quiz score on a topic is above a certain threshold, the tutor then looks at how long it has been since the topic had been part of a quiz. If the time has been sufficiently long, then the topic is tested. The time variable is a function of the topic’s score: the higher the score, the longer the time interval allowed. The reasoning for this is that if the score on the topic is very high, there is a good chance the student knows the material, and will thus need less frequent reminders. When testing on a topic that has a high score, the tutor tries to choose questions that are harder than those asked in the past. If none exist, those questions that have not been asked for the longest period of time are chosen.

It should be noted that within a topic, it is possible that not all concepts will be tested. This will happen if it seems that the student has a good grasp of some concepts, and it is not time to review that concept.

Grading a quiz: The student is graded on his quizzes, with correct answers leading to his scores on the corresponding topics being increased. If the question is answered incorrectly, the tutor concludes that the student did not understand the concept nor the corresponding topic. Then the student’s score on the topic is lowered and the question is marked as being answered incorrectly. Furthermore, the tutor suggests that the student review the material corresponding to the concepts he did not understand. If the student elects to see these slides, he will see an easier version, since his knowledge of the topic and concept have been downgraded.

3.3 Prefetching

Many users on the World Wide Web complain of latencies experienced due to downloading a new page [11]. One way to reduce these delays is to download the page ahead of time, before the user makes an explicit request. With a MANIC course, one option is to prefetch large sections of the course, or even the entire course, ahead of time. However, in reality, we cannot do this. There is never infinite cache space nor network bandwidth. Thus we cannot prefetch the entire course, nor can we always prefetch more data than we know the student will use. For these reasons, we must try to intelligently prefetch only small portions of the course.

This prefetching, if accurate, can reduce the delays seen by the user, since the information will already be available before it is requested. However, prefetching that is not accurate will waste network bandwidth and cache space, by storing pages not requested by the user. Thus, the question remains as to how to make the prefetching accurate. Prefetching has been implemented in both file systems [8] and on the WWW [11]. The work presented in these papers involves general population models for guiding the prefetching. Our work, however, uses individual student models to aid with prefetching.

There are three kinds of prefetching that MANIC performs. The first is prefetching while the student is viewing the material on one topic. Prefetching can only be done if the system can determine the pattern with which the student is proceeding through a topic. Currently, only a few patterns can be detected within the MANIC system: (1) letting the audio control the display of slides in a topic and (2) linearly scrolling through a topic, without listening to the audio. If either of these patterns is detected,
the next slide in the sequence is prefetched. We are currently analyzing the access patterns from the first student experiences with MANIC to identify other patterns students use for viewing the material.

The next category of prefetching occurs when the end of a topic is reached. Which new topic should the tutor start to prefetch? Because there is not a clear linear path through the course, our work differs from that presented in [12]. If the tutor predicts the student is most likely going to choose one particular topic, it can start prefetching that one, ignoring the others. The tutor can begin to “know” this information by seeing how accurate its predictions are for what the student will choose as his next topic.

However, another possibility is that the tutor will not “know” with any confidence the next topic the student will elect to see. In this case, the tutor can prefetch the first slide for each topic, essentially “covering all bases.” Once a topic is started, it alone can be prefetched.

A compromise between these two extremes is to prefetch the first few slides of the topics that are most likely to be started by the student next. These topics can be determined by viewing how the student has been progressing through the course and matching this with the weights in the topic net. Those topics with the highest probability from the current topic are candidates for prefetching. To determine how many slides per topic are prefetched, we use the simple rule: the higher the probability, the more slides will be prefetched.

Another category of material that can be prefetched is the hypertext links that appear on some of the slides. The student model tracks if these hypertext links are usually followed. If so, when one appears on a slide, the tutor can prefetch that link.

4 Initial Student Reactions and Conclusions

During the Fall of 1996, 15 students used the MANIC system as part of a one credit course on UNIX network programming and Java, 9 of whom filled out a questionnaire. The version of MANIC at that time did not include any adaptive hypertext techniques or adaptive quizzing, but did include some prefetching techniques. Our main goal was to determine students’ likes and dislikes about the software, and to use this information to change the system.

Eight out of nine students expressed very positive feedback about MANIC. These eight students indicated that they enjoyed the on-line course, with the ability to study at their convenience. Six students liked the linear nature of the course, but the other three would have preferred to have been able to choose what material to see next. Also, seven students said they would have taken advantage of on-line quizzes.

However, not everything was perfect. Students complained about the screen layout, indicating that we need to redesign with their comments in mind. Also, many students had problems logging in, due to the instability of the servers and the proxy, which caused a great deal of frustration.

In the future, we will continue to improve the student model to provide a individualized learning for each student. Since 150 students are using a MANIC course this semester, we will be able to study how they have used the system, and use this information to improve the student model. We will also be adding alternate ways of teaching the material, including video and animations. With these new presentation methods, we must improve MANIC’s decision making mechanism to be able to choose the best way to present the material, based on the student model.

Additionally, we hope to develop more courses using the MANIC software. One goal of this research is to develop general student modeling techniques that can be used when converting any video-taped course. In the Fall of 1997, we will be using MANIC to convert a software engineering course to be available on line.

Overall, we are encouraged by the initial testing of MANIC. Since there was not a consensus amongst the students on how the courseware should be used, we must provide alternate ways for students to proceed. We are convinced that the work presented in this paper will answer many of the criticisms given by the students. With the inclusion of a comprehensive student model, we will be able to individualize the course material for each student, based on his preferences and needs.
Acknowledgments

This work is supported in part by NSF/DARPA and Apple Computer Technology Reinvestment Award (CDA-9408607) and by NSF awards (CDA-9502639 and NCR-9508274).

References


