SAC: A Self-Paced and Adaptive Courseware System

Alvin T.S. Chan*, Steven Y.C. Chan and Jiannong Cao
The Hong Kong Polytechnic University, Department of Computing
Hong Kong, SAR China
**cstschan@comp.polyu.edu.hk*

Abstract

This paper presents the design and implementation of a Self-paced and Adaptive Courseware, in short SAC. The main focus of SAC is to formulate a model that encompasses the important requirements of supporting an adaptive learning courseware environment. The paper begins by discussing the issues and design requirements of developing an interactive and adaptive learning system that is able to individualize a student learning style, with the ultimate objective of maximizing his learning experience and effectiveness. The system has been designed and implemented on a three-tier web application architecture, which uses the AHAM approach to structure the domain, user and teaching model for use by the adaptation engine.

1. Introduction

The widespread usage and popularity of the Internet and WWW have shifted information systems development from client-server based on a multi-tier, cross-platform and web-enabled application development environment. Such fundamental shift has also infiltrated to the development of courseware, in which WWW and the Internet provide unique opportunity to deliver distant education with wide accessibility at lower costs. In fact, the flexibility, ubiquity and cost of the technology have often been sited as strong motivations for its use. Despite the potentially significant advantages, web-based courseware packages tend to vary greatly both in quality and educational effectiveness.

Frequently such courseware, although initially novel to the learner, do not hold the learners' attention, fail to enhance his/her higher order learning skills and are not properly integrated within an educationally sound curriculum [1]. In most platforms, course materials are presented primarily in a sequential manner according to the author's perception of the organization of the materials, while the focus is solely on delivering multimedia contents to end users. Although this addressed the needs of some students, others may find it difficult to follow and that the flow structure may differ significantly from the student's preferred learning style. With the diverse backgrounds, knowledge and learning rate, it is reasonable to assume that each student may traverse the course materials according to his or her unique learning style. Unfortunately, most of the courseware available in the market do not readily consider this dimension of learning needs, in which the traditional style of delivering materials to a classroom of students is adopted. In particular, appropriate tools are not made available to students to help them make active choices of what course materials would be most conducive to their learning [4], while taking into consideration of individual needs. Occasionally, some of the courseware do provide limited preferences customization tools, dynamic hyper-linking to different references, but ignoring the most important element of a courseware – an adaptive linking, non-linear and self-paced capability. This paper presents the design and implementation of a Self-paced and Adaptive Courseware, in short SAC. The main focus of SAC is to formulate a model that encompasses the important requirements of supporting an adaptive learning courseware environment. The paper begins by discussing the issues and design requirements of developing an interactive and adaptive learning system that is able to individualize a student learning style, with the ultimate objective of maximizing its learning experience and effectiveness. In section 3, we highlight some of the methods and techniques that are employed to develop adaptive hypermedia system (AHS). Section 4 presents the conceptual design and implementation of the SAC courseware. Finally, the paper ends with a conclusion in section 5.

2. Requirements for web-based educational courseware

A distinct feature of Web-based courseware is in its ability to be accessed in a wide area environment and across heterogeneous systems. In fact, this is the single most important driving factor for the increase awareness and opportunity presented by web-based courseware to provide *anywhere, anytime* and *anyplace* education platform. Unlike other teaching methods, courseware delivery through the WWW is highly heterogeneous, both in terms of the readers' knowledge and learning style, and also the information provided. Unless delivered in a closely monitored and observed environment, the academic standards of students are highly variable, such that no single learning style or program layout is suitable for all the students. In general sense, in fact, students are

0-7695-1013-2/01 $10.00 © 2001 IEEE

their own best guides for the learning process, in which online courseware should readily focus on learner-centered design. Such approach should emphasize on a constructive theory of learning wherein students learn through a process of building their own mental models[3], which is adaptive to their own learning progresses and academic backgrounds. Path-like mechanisms that support personal space organization dynamically evolve to adapt individual student needs for acquiring the ultimate learning objectives, according to their academic backgrounds, learning progresses, and preferences.

A dynamically adaptable educational system should provide personalized navigation guidance, knowledge construction assistance, and courseware analysis tools, which emphasizes on courseware structure and navigational behavior in an educational hypermedia environment. In formulating the personalized learning path, the system should take into consideration the courseware network to determine whether it is well structured, and automatically generate a hierarchical guidance map to help users navigate in a hypermedia environment [2]. The resultant hierarchical learning path allows opportunity for students to explore sequentially in an incremental manner, with adaptive capability to evolve the learning style as new knowledge is acquired.

An important feature of an adaptable educational hypermedia system is in its ability to monitor students' behaviors, learning styles and exploration of the learning resources and materials presented to them. An integrated mechanism is required to provide feedback as to the frequency of use of specific resources, and to identify trends in the student navigation of the course pages. In addition to supporting sequential browsing of the course contents, task and objective-oriented paths should be made available for students to acquire in-depth exposure to the specific subject matters.

The focus of this paper is to investigate the techniques and approaches to developing web-based multimedia courseware package, and to suggest the design and implementation of a highly adaptive, self-paced Internet courseware based on an improved Dexter-based Reference Model [7].

3. Approaches and Techniques for Adaptive Courseware

Adaptive techniques have been widely used by a number of researchers in an attempt to offer navigation guidance and orientation support for rich link structures in courseware systems. A number of techniques have been developed and reviewed in [6], most of which achieve adaptation of hypermedia documents based on user's state or level of expertise. Most techniques offer some variant of conditional text, resulting in variants of text fragments, whole pages, or even clusters of pages. Also, the link structure of a hyper document can be adaptive to offer personalized navigation guidance. Adaptive link structure can be positively guided by means of sorting available links according to user preference or relevance, or negatively guided by means of annotating, dimming or even hiding links which the user should not (yet) follow.

3.1. Structuring of Courseware

According to [5] in the implementation of the Modular Training System, an Internet-Based Courseware can be structured by organizing into three categories:

- *Course Nodes* (CN
- *Course Unit* (CU)
- *Course Material* (CM)

The network of *Course Nodes* defines a course structure and a pre-defined learning goal. While it defines the nodes dependencies and flow, the *Course Node* does not contain any information on how this learning goal can be achieved. Each node delivers a return value such that appropriate event and the next node can be selected. Each node may contain attributes, which indicate the objectives and pre-requisites of the node, as well as the available child nodes that are linked. The contents of each node can be dynamically linked to other child nodes to form a networked course structure according to the profile of individual students. Navigation through the course depends on learners' own path, which may be linked dynamically as a student traverses the course materials.

Course Unit defines the presentation units, exploration units and test units. They are self-contained, and contain references to other course materials. In this sense, *Course Units* form the containers of the course materials that deliver a return value indicating the learner's success when traversing the unit.

3.2. Adaptation Models

The Adaptive Hypermedia Application Model (AHAM) discussed in [7], encompasses most features supported by adaptive systems that exist today. The Adaptive Hypermedia Application Model (AHAM) is based on the *Dexter Model* [7], a widely used reference model for hypermedia. The model achieves personalized adaptation based on a *User Model*, which persists beyond the duration of a session. The system is comprised of three integrative models: *Domain Model, a User Model* and a *Teaching Model.*

The system keeps track of evolving aspects of the user, such as preferences and domain knowledge. This permanent and continuously updated record is captured in the *User Model.* It is used to guide the user towards learning goal and away from information the system considers to be inappropriate or irrelevant for the user. The AHS may do this by dynamically altering the hyperdocument's link structure, *Adaptive Link Annotation*, and/or by dynamically generating or changing the content of the information nodes, *Adaptive Linking*. Users can set certain preferences explicitly or initialize the user model through a registration form or "pre-test" so that the background knowledge can be captured for initial generation of the structure of the hypermedia documents. The *Domain Model* describes how the information is structured and linked together. The *Teaching Model* consists of pedagogical rules that define how the domain model and the user model are combined to provide ways to perform the actual adaptation.

4. Conceptual Design of Self-paced and Adaptive web-based Courseware (SAC)

This section describes the design and implementation of the Self-paced and Adaptive, web-based Courseware (SAC) that is designed to integrate the concepts of Domain, User, and Teaching models described in AHAM for adaptive hypermedia applications [7]. The course structure, materials and learning goals are captured in the abstraction of *Course Nodes*, *Course Units* and *Course Material* as described in the Modular Training System (MTS) [5]. The overall design the SAC is shown in Figure 1.

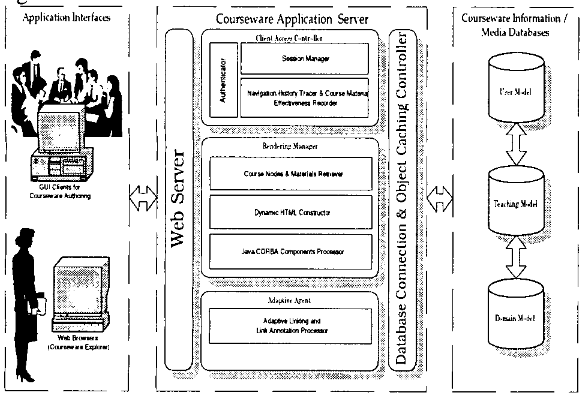
Application Interfaces
GUI Clients for Courseware Authoring
Web Browsers (Courseware Explorer)
Courseware Application Server
Web Server
Client Access Controller
Authenticator
Session Manager
Navigation History Tracer & Course Material Effectiveness Recorder
Rendering Manager
Course Nodes & Materials Retriever
Dynamic HTML Constructor
Java CORBA Components Processor
Adaptive Agent
Adaptive Linking and Link Annotation Processor
Database Connection & Object Caching Controller
Courseware Information / Media Databases
User Model
Teaching Model
Domain Model

Figure 1: SAC Architecture

The courseware is implemented through a 3-Tier Application Model architecture. The first tier is implemented through the browser, which is concerned with presentation of the courseware. The middle tier is implemented and provided by a combination of the web server and an application server. The application server generates XML, DHTML and client-side JavaScript dynamically by using the model information stored in the 3rd tier backend database system.

The adaptive annotation feature of SAC is implemented through a dynamically generated XML document and a browser Java applet. Specifically, the courseware navigation is achieved by providing a tree-based menu, which is generated dynamically by the middle-tier application server in the form of a XML-formatted document. The tree-based menu reflects the current status of the overall learning progress and the adapted learning path for the student. All linked hyper-documents are annotated so that students can easily access the required course materials by surfing through the course nodes. According to student's navigation history and the assessments taken, the navigation menu of the courseware dynamically adapts to his or her learning experience and style. Suggested learning path, as described in the XML document, is parsed by an applet and display as a tree structure, such that student taking the course can view the learning map, with the ultimate objective of maximizing their learning experience.

4.1. Adaptive Mechanisms

The adaptive capability of SAC is provided by the Adaptive Agent. It is comprised of the following adaptive engines: pre-admission Adaptive Engine, Online Individual Adaptive Engine and the Domain Adaptive Engine.

The Admission Manager captures the educational background of individual student via pre-admission tests and SAC enrolment history. It is combined with the pre-defined course inter-node relationships stored in the Teaching Model to form students' academic background and course nodes pre-requisite analysis results, which is then stored into the User Model database.

By using the Student Preferences Customization function provided by SAC, individual student can input their own navigation preferences into the User Model. The Session Manager continuously keeps track of the Online Surfing Information and updates the Navigation History stored in the model.

The Online Individual Adaptive Engine retrieves individual navigation history, academic background and pre-requisites information for analysis. Subsequently, the Course Node Status information is formed which is used as the core information for rendering of the XML documents. The XML rendering and DHTML Generation process combine the Node Status information, user preferences and the Course Inter-Nodes Relationship to produce the Dynamic, Adaptive Link-Annotated web pages and returns to the SAC Browsers.

The batch mode Domain Adaptive Engine uses the historical Navigation Pattern from previous students and the default domain adaptive information to provide revision suggestions to the Teaching Model. The Domain Adaptive information will also be updated during the analysis process.

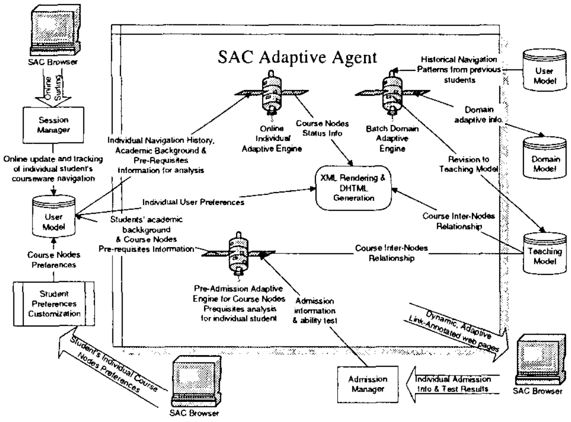
SAC Browser
Online Surfing
Session Manager
Online update and tracking of individual student's courseware navigation
User Model
Course Nodes Preferences
Student Preferences Customization
Student's Individual Course Nodes Preferences
SAC Browser
SAC Adaptive Agent
Individual Navigation History, Academic Background & Pre-Requisites Information for analysis
Online Individual Adaptive Engine
Course Nodes Status Info
Batch Domain Adaptive Engine
Historical Navigation Patterns from previous students
User Model
Domain adaptive info.
Domain Model
Revision to Teaching Model
XML Rendering & DHTML Generation
Individual User Preferences
Students' academic backkground & Course Nodes Pre-requisites Information
Course Inter-Nodes Relationship
Course Inter-Nodes Relationship
Teaching Model
Pre-Admission Adaptive Engine for Course Nodes Prequisites analysis for individual student
Admission information & ability test
Dynamic, Adaptive Link-Annotated web pages
Admission Manager
Individual Admission Info & Test Results
SAC Browser

Figure 2: SAC Adaptive Agent

4.2. Dynamic Site Map

The dynamic adaptation path of the course guidance menu is generated by a Java applet residing within the client-side, which interacts actively with the server-side adaptation engine. Each SAC user is presented with their unique dynamic web content navigation guidance based on their learning style, pace and foreknowledge. Course nodes are presented as leaves and folders in the navigation menu. As the SAC learner surf through the course materials, the client-side Session Manager, implemented as a combination of Javascript and applet, will actively trace his navigation history and record the test results, if any, and finally update the User Model database via the application server. Next time when the user logon again, the generated tree menu will reflect the latest navigation history and test results by re-generating the required XML document as input to the tree applet, according to the Teaching & User Models stored in the backend database.

5. Conclusion

In this paper, we have highlighted the importance of courseware to incorporate the concept of adaptive learning by providing dynamic navigational guidance to students taking online courses. This is in contrast with most existing courseware systems, whose focus is on pure information delivery, which often leads to information disorientation and confusion for learners. As courseware systems are Internet-enabled to ride on the wide accessibility of Web infrastructure, the mix of students' backgrounds, knowledge and learning capabilities are often diverse and heterogeneous in nature. Online courses designed to be delivered across such open platform need to support the desire of students to self-pace their learning rate and style, and to help them to develop quality mental maps of the course delivery structure. This paper describes some of the approaches and development in the area of adaptive hypermedia guidance courseware. We introduce the design and development of a Self-paced and Adaptive Courseware (SAC) system in an attempt to develop an interactive and learning system that is able to individualize a student's learning style using adaptive link guidance approach. The system has been designed and implemented on a three-tier web application architecture, which uses the AHAM to structure the domain, user and teaching model for use by the adaptation engine. In the future, we are intending to develop a formal evaluation test on the effectiveness of the system in enhancing students' learning experiences using the adaptive framework of the SAC system. The results will assist us to better understand the nature and evolution of students' mental maps of the taught materials and to formulate ways in which the adaptation engine can be designed to develop guidance links to maximize their learning experience.

6. References

[1] Vincent P. Wade & Conor Power, Evaluating the Design and Delivery of WWW Based Educational Environments and Courseware, *ITiCSE '98* Dublin, Ireland, 1998 ACM 1-58113-000-7/98/0008

[2] Chuen-Tsai Sun, Chien Chou, and Bing-Kuen Lin, Structural and Navigational Analysis of Hypermedia Courseware, *IEEE Transactions on Education*, Vol. 41, No. 4, Nov 1998

[3] Owen Astrachan, Concrete Teaching: Hooks and Props as Instructional Technology, *ITiCSE '98* Dublin, Ireland, 1998 ACM 1-58113-000-7/98/0008

[4] Curtis A. Carver, Jr., Richard A. Howard, and William D. Lane, SM, Enhancing Student Learning Through Hypermedia Courseware and Incorporation of Student Learning Styles, *IEEE Transactions on Education*, Vol. 42, No. 1, Feb 1999

[5] Taofen Wang & Christoph Hornung, The Modular Training System (MTS) - A System Architecture for Internet-Based Learning and Training, *Fraunhofer-Institute for Computer Graphics (Fh-IGD)*, German, 0-8186-8150-0/97 1997 IEEE

[6] Licia Calvi, Improving the Usability of Hypertext Courseware through Adaptive Linking, *Hyprtes! 97*, Southampton UK @1997 ACM0-89791-866-5

[7] Paul De Bra, Geert-Jan Houbent and Hongjing Wu, *AHAM:* A Dexter-based Reference Model for Adaptive Hypermedia, *Hypertext 99* Darmstadt Germany Copyright ACM 1999 1-58113-064-3/99/2