DOCUMENT RESUME

ED 427 749 IR 019 300

AUTHOR Wu, Albert; Tam, Lincoln
TITLE An Architecture for Dynamic Courseware Working on the Web.
PUB DATE 1998-11-00
NOTE 6p.; In: WebNet 98 World Conference of the WWW, Internet and Intranet Proceedings (3rd, Orlando, FL, November 7-12, 1998); see IR 019 231. Figures may not reproduce clearly.
PUB TYPE Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Managed Instruction; Computer Software Development; *Courseware; Foreign Countries; Information Retrieval; *Instructional Design; *Instructional Materials; Material Development; *World Wide Web
IDENTIFIERS *Hong Kong Polytechnic

ABSTRACT

This paper presents an architecture for creating, retrieving, and adapting materials on the World Wide Web for learning purposes. There are two major components in the architecture: Authoring Server (AS) and Courseware Server (CS). Associated with the architecture for running dynamic courseware, certain mechanisms are also imposed. These mechanisms involve the manipulation of information on some defined templates. At the moment, there are three standard templates developed: courseware-information-template, curriculum-template, and teaching-style-template. The architecture components have been experimentally implemented in a local server housed at the Hong Kong Polytechnic University. Although more significant results are yet to be seen, the endeavor nonetheless provides a simple yet effective means for building courseware from the pool of teaching materials on the Web. (AEF)

An Architecture for Dynamic Courseware Working on the Web

Albert Wu and **Lincoln Tam**

Department of Computing
Hong Kong Polytechnic University
Hong Kong

Email: csalbert@polyu.edu.hk

Abstract: We have devised an architecture for dynamic courseware working on the World Wide Web. Inherent in the architecture are two components: Authoring Server and Courseware Server. With the associated mechanisms, the approach enables the working of dynamic courseware - composed of widely dispersed learning-related materials on the web, to function for a particular learner.

Introduction

Conventional courseware approaches like CAI tend to be rigid and lack the dynamicity to address individual learners' needs. 'Dynamicity' here means the flexibility of courseware content in meeting the different requirements of different learners. With the emergence of the WWW (World Wide Web) or Web it now became possible for learners to access the myriads of teaching materials available via the Internet. Thus making it feasible for the creation and functioning of dynamic courseware. With the Web, the latest and most updated information can also be gathered. However, the learners have to first face a significant problem: they have to cope a vast amount of materials and to navigate through links which sometimes are not relevant to their learning goals. Very often they get lost in the hyperspace and loose a focused perspective on what they want to learn! Moreover, most of these materials are dispersed and inherently disparate, making their exploitation in an integrated manner difficult [Gruber, Vemuri and Rice 1998]. Searching and retrieving relevant information from current retrieval mechanisms provided by the prevailing search engines like Yahoo, Alta Vista, Infoseek or Webcrawler are also futile as the results are mostly of low precision [Gaines & Shaw 1997].

In this paper, we report on an endeavour towards the realization of dynamic courseware working on the Web. Namely, through the imposition of an architecture for the creation, retrieval and adaptation of materials for learning purposes.

The Architecture

A basic assumption of our approach is that courseware is consisted of modules of units of concept-topics [Gagne 1985]. While a concept may be illustrated via a number of topics, a topic is treated as the most fundamental unit in a courseware piece.

There are two major components in the architecture: AS (Authoring Server) and CS (Courseware Server) (Fig.1).

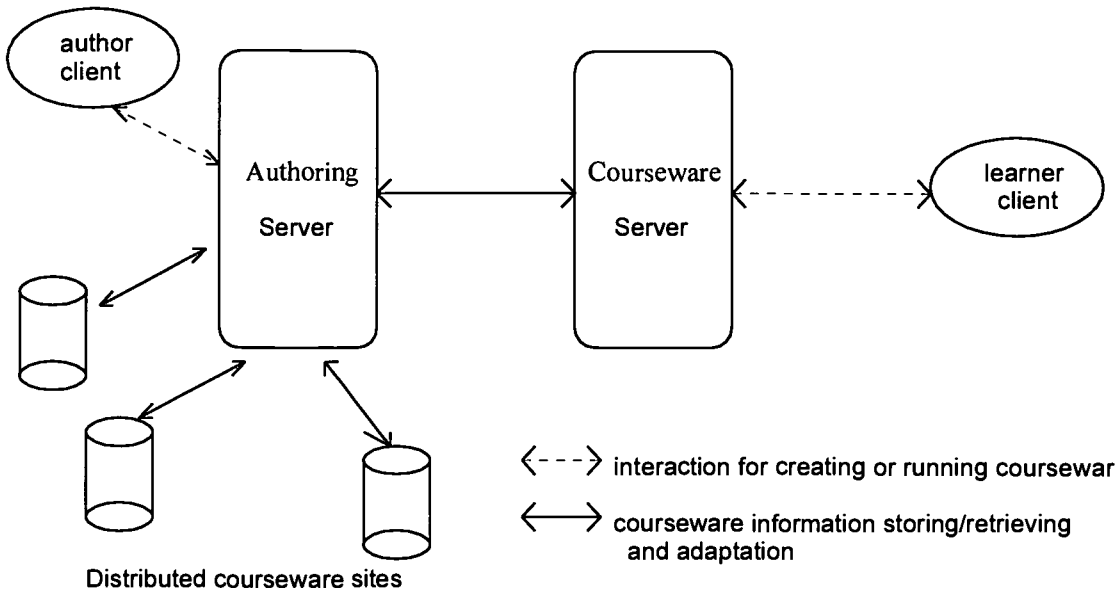

Fig.1: An architecture for operating dynamic courseware

Referring to Fig.1, a client-server approach is adopted in the working of the architecture. With learners and authors/teachers being clients using the services provided by the Authoring Server and Courseware Server respectively. (Here a client is assumed operating with a Web browser.) The teaching materials are expected distributed on various sites in the Web.

Related Mechanisms

Associated with the architecture for running dynamic courseware certain mechanisms are also imposed. Basically these mechanisms involve the manipulation of information on some defined templates. At the moment, there are three standard templates developed, they are:

i) Courseware-Information-Template (CI-Template) containing
- Subject domain information
- Description of relevant topics
- Information related to Uniform Resource Locator (URL/Web Site Address)
- Information related to the medium of teaching
- Information related to pre-requisite level required of the course

ii) Curriculum-Template (C-Template) containing
- Information related to the difficulty level of the courseware
- Topic structure, e.g. relations amongst topics for instructional planning
- Mastery level required of the course
- Teaching example for illustrating the topic
- Exercise for practising understanding of the topic
- Hint for exercise

ii) Teaching-Style-Template (TS-Template) containing
- Possible instructional method(s) used for using the courseware
- Information related to assessment
- The way instruction logic of control is determined
- The way of monitoring the learner's state of learning

Basically the CI-template is coupled with the Authoring Server to provide for authoring services while the C-template and TS-template are associated with the Courseware Server for the composing of dynamic courseware (Fig.2).

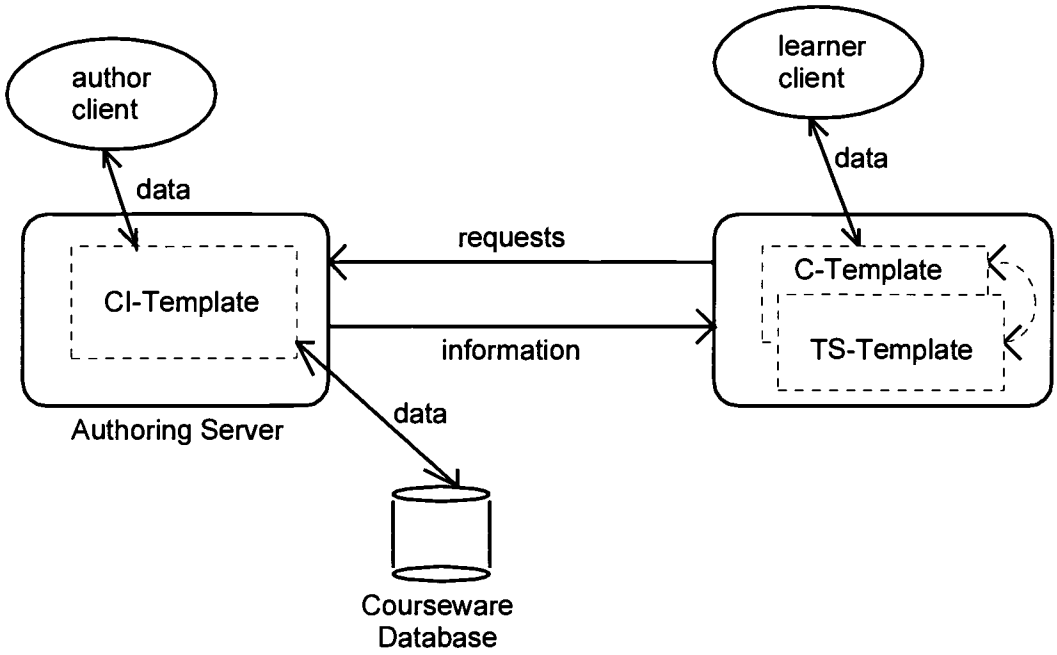

Fig.2: Working amongst the templates

With the CI-template, a standardized guideline or format is provided for the author to 'register' his/her teaching materials in the Web. As for the C-template and TS-template, they help the learner to specify his/her learning particulars so that more specific learning materials can be collected and collated for the courseware. Operating the templates essentially relies on some Java-procedures that carry out the interaction.

Implementation

The architecture components namely AS (Authoring Server) and CS (Courseware Server) and the associated template procedures have been experimentally implemented in a local server housed at the Hong Kong Polytechnic University. In order to facilitate its usage, all system procedures are implemented in Java except those related to courseware databases, which are essentially based on PERL (Practical Extraction and Reporting Language) as the CGI (Common Gateway Interface) script. Java-based and HTML interfaces are also used. Compiled Java applets, which are in the form of bytecode, can be run on any platform with Java's runtime environment present. In the implementation, Java applet is used for displaying the dynamically generated hierarchy of courseware. The applet is embedded inside the hypertext page and be viewed as a table of content page. Three different views are also organised for displaying Modules, Units and Topics. Besides, the implementation also allows users to browse through the table of content of the courseware in hypertext links with web site addresses shown.

Illustration

We have experimented the architecture by building an HTML (Hypertext Markup Language) courseware [Lemay's 1996]. Below we illustrate some of the functioning screens generated by the PERL scripts and Java procedures for developing the courseware (Fig. 3a and Fig. 3b) (with Oracle database used to support the courseware databases). The screens will not be explained as they are self-explanatory.

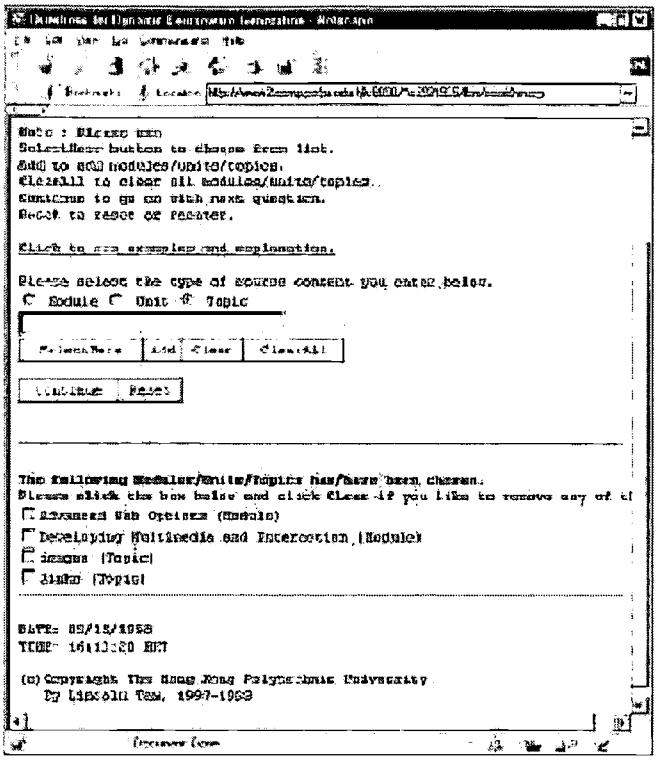

Fig.3a: Sample screen generated by CGI/PERL scripts.

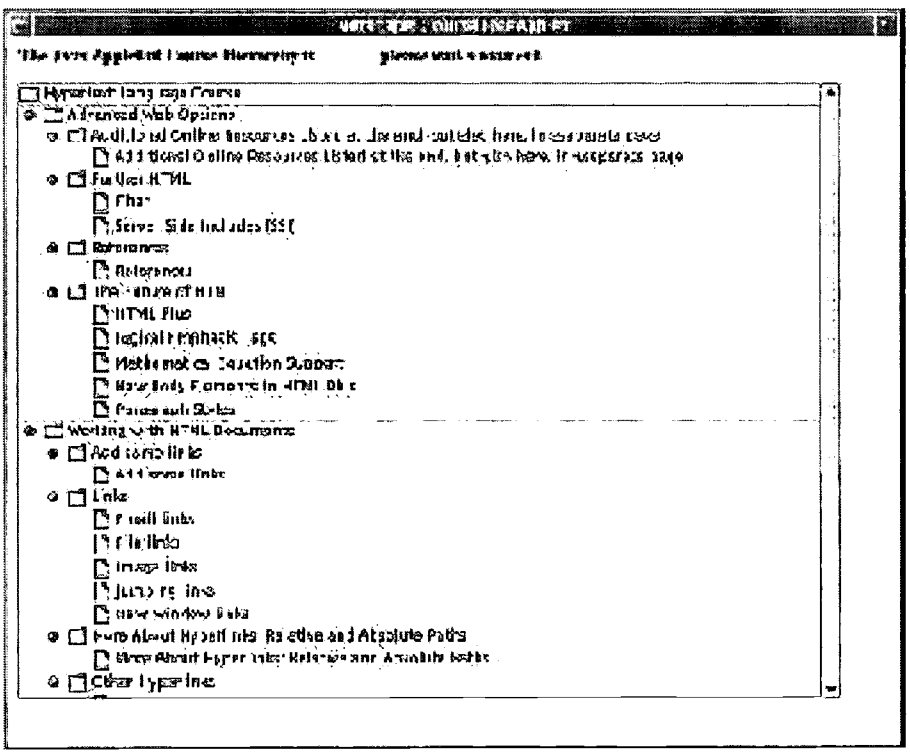

Fig.3b: Sample Java Applet showing course hierarchy.

Discussions

Initial benefits perceived of the approach can be summarized as follows:

- Improvement in relevance of search results in the web.
- Dynamicity in fulfilling different users' needs.
- Efficience in distributing teaching materials.
- Standard web-based user interface.
- Good accessibility of courseware, moreover the approach is easy to use.
- Portablity, as the course is adoptable for multi platforms.

However, similar to others, we also experienced some limitations in our current work [Ibrahim 1997]. They are:

- Our current graphical user interface is limited by the primitive syntax of HTML. Nonetheless, such condition can be improved with the recent development of XML (Extensible Markup Language).
- We currently face some problems related to response time limitation due to heavy network load and server load. It is expected that with improvement in the Internet infrastructure (e.g. bandwidth, etc.) the situation can be alleviated.
- Statelessness of the HTTP protocol. This is a problem faced by the WWW community and is anticipated not easily be solved in the near future.
- In our implementation, we have yet installed any user model. Yet, for truly adaptive to individual, the incorporation of an appropriate user model is mandatory [Kok 1991].

With the recent development of XML (Extensible Markup Language), it is expected that the display aspects can be much improved. An improvement in the Internet infrastructure (e.g. bandwidth, etc.) can also help alleviate the situation. In addition, more visualisation techniques can be applied for the manipulation of templates to aid in the representation of the structure and relationship of courseware contents. Currently we are preparing to implement an accompanying agent to enable further exploitation of the architecture and its associated mechanisms for dynamic courseware generation. Although more significant result is yet to be seen, the endeavour nonetheless provides a simple yet effective means for building courseware from the pool of teaching materials in the web.

References

[Ibrahim 1997] Ibrahim, B. (1997). Use of HTML forms in complex user interfaces for server-side applications. *International Journal on Man-Machine Studies*, **46**, 761-771.

[Gagne 1985] Gagne, R.M. (1985). *The Conditions of Learning (fourth* edition). New York: Holt, Rinchart, and Winston.

[Gaines and Shaw 1997] Gaines, B. R. and Shaw, M. L. G. (1997). Knowledge acquisition, modelling and inference through the World Wide Web. *International Journal on Man-Machine Studies*, **46**, 729-759.

[Gruber, Vemuri and Rice 1998] Gruber, T. R., Vemuri, S. and Rice, J. (1998). Model-based virtual document generation. *International Journal on Man-Machine Studies*, **46**, 687-706.

[Lemay's 1996] *Teach Yourself Web Publishing with HTML 3.2 in 14 Days.* Sams Net Publishing.

[Kok 1991] A formal approach to user models in data retrieval. *International Journal on Man-Machine Studies, 35,* 675-693.

U.S. Department of Education
Office of Educational Research and Improvement (OERI)
National Library of Education (NLE)
Educational Resources Information Center (ERIC)

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").