A METHOD OF DESIGNING WEB-BASED EOD SYSTEM

P. Venkataram, R.Rajavelsamy and Anandi Giridharan

Department of Electrical Communication Engineering
Indian Institute of Science
Bangalore - 560 012, India
pallapa@ece.iisc.ernet.in, rajvel@protocol.ece.iisc.ernet.in, anandi@ece.iisc.ernet.in

ABSTRACT

It is generally recognized that the Internet can be used as a platform to deliver distance education services and the Internet information systems can be leveraged to enhance education. In this paper we describe online design and delivery of subject contents to various categories of learners using WESLA (Web based Education System for Learning and other Academic activities), an Education-On-Demand model. In particular we describe, the implementation of the "WESLA" for the Communciation Protocol course.

1. INTRODUCTION

Cyberage technologies permit interaction asynchronously, between students and instructor. We will herein describe the successes and challenges that we have experienced in developing courses with Education-on-Demand facility with WESLA model. All the lectures are presented as study pages in HTML format to the learners. These are synchronized in time with the activities in the video window so that the slides change automatically as the lecture progresses. In addition, a lecture is also classified into major topical areas (as shown in the MENU window in Figure 1) so that the learners can go to a specific topic of interest. In our proposed WESLA model, different learners student, faculty and Expert are considered.

2. THE WESLA DESIGN

Architecture of WESLA with various components and their inter-connectivity are shown in figure 2. WESLA system works as a client/server model to deliver the learner's requested contents. Clients are geographically distributed with various categories of learners, who access the course material from centralized server [1]. Various components of WESLA architecture are described below.

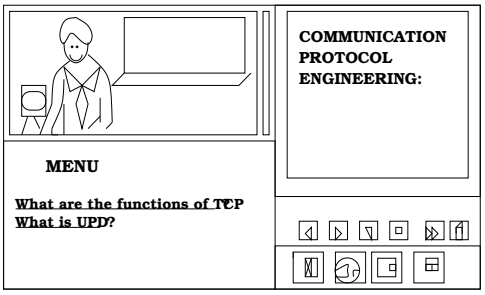

Figure 1: WESLA based EOD Lectures

2.1. WESLA Server

2.1.1. WESLA Server Interface

Server Interface interacts with Client Interface to authenticate the learner. The course modules are sent to the learners (clients) according to their requirements and needs.

2.1.2. Subject(s) domain

In WESLA system, subject domain is a conceptual model representation of subject material for the courses. A subject material is organized as a set of modules [2]. Modules are classified into its associated concepts and again concepts are classified into subconcepts. Inter-module relation and inter-conceptual links have been set up like Module-Module

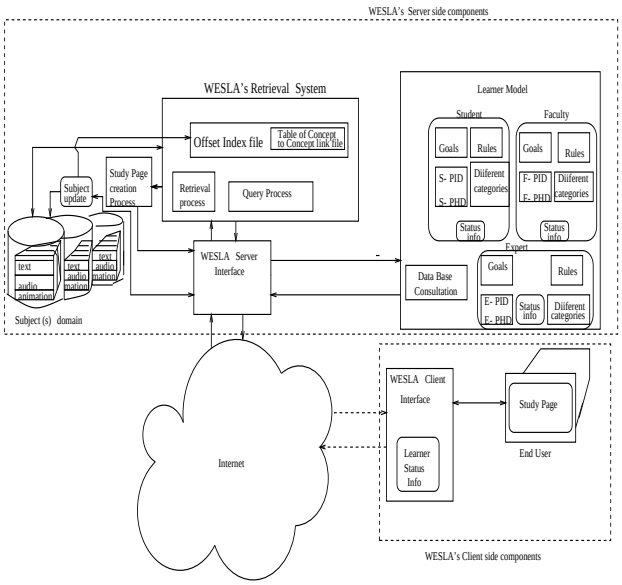

Figure 2: Architecture of WESLA EOD Design

relation, Module-concept relation, Concept-Concept relation based on the order in which subject material is to be presented to the learners.

2.1.3. Learner Model

Learner model is an explicit representation of properties of a particular learner. It describes different categories of learners like student, faculty and expert with their objectives/goals, and also with their Database containing Personal Information Database (L-PID) and Performance History Database (PHD). A system that constructs and consults learner model can adapt diverse aspects of its performance to individual learners. Learner models are important within computer-based systems intended to promote learning because they provide the means to support intelligent individually-adapted instruction . It is a process of inferring the learner's knowledge by analyzing his/her behavior.

2.1.4. Learner's Interface

WESLA system, interacts with all the categories of learners through learner's interface and collects the information on type of activity to be conducted by the learner.

2.1.5. Offset Index-File

To locate and retrieve the exact module or concept from the subject domain, Offset Index-File is created. Subject domain-file is constructed using delimiters to indicate the physical location of each concept on the secondary storage (Hard disk or CDROM).

2.1.6. Table of Concept to Concept

Whenever a learner likes to have depth understanding about a single concept, which may be distributed among various modules, to access such concept, we have created concept to concept link. This will be separately stored in table of Concept to Concept link file.

2.1.7. Subject Update

Subject update can be done by adding additional information to the existing concepts or introducing new concepts to the existing modules. Offset Index file will be updated accordingly.

2.1.8. Database consultation

After Interacting with learners, model checks for the status of the learners from their appropriate Databases.

2.1.9. Query processing

Queries are accepted from different learners like student, faculty and expert. Based on the purpose and objectives of respective learner, different types of queries are created.

Algorithm: Learner Query Processing

Nomenclature
Study Page: *File created with the requested concepts.*
Data Base : *Student records are stored*
Module: *Both logical and physical grouping of course material*
Concept: *Each Module as a set of associated concepts*
fp: *file pointer*

1. IF Student as Learner
BEGIN
    Read student Login and Password
    Authenticate the learner
    OPEN Data Base of Log on Learner
    READ the learner info from the DB

```
    List the Module/Concepts to be retrieved.
    CALL Study Page Creation Algorithm
    DISPLAY the study page;
END

2. IF Faculty or Expert as Learner
BEGIN
   Read user Login and Password
   Authentication Check
   DISPLAY faculty queries
   READ Choice
   IF (Choice = Paper setting)
      DISPLAY index of concepts & set questions.
   IF (Choice = Creating Assignments)
      DISPLAY index concepts of assignments.
   IF (Choice = Framing Exercise)
      DISPLAY index of concepts to check exercise
   IF (Choice = Conducting course)
      DISPLAY index of concepts regarding lectures.
   IF (Choice = Deep level knowledge of a concept)
    BEGIN
          DISPLAY index of concepts;
          READ the concept topic;
          OPEN Concept to Concept links file;
          READ List_of_Concepts to the topic
    END
    CALL Study Page Creation Algorithm
    DISPLAY Study Page
END
```

2.1.10. Study page creation process

After retrieval process, according to the requirements of the learner, the Study page creation process converts the fetched data in to HTML format. This converted HTML page is sent to the WESLA Client side.

Algorithm: Study page creation Processing

```
BEGIN
   READ the list of concepts to be retrieved;
   CREATE a new Study_Page;
   FOR all listed Concepts
   BEGIN
      OPEN Offset_Index File;
      READ Offset_Begin and Offset_End;
      CLOSE Offset_Index File;
      OPEN Subject domain_file;
      SEEK fp to offset_begin in Subject domain_file;
      WHILE (fp ≤ Offset_End)
          WRITE the contents to Study_Page;
      CLOSE Subject domain_file and Study_Page;
   END
END
```

Algorithm: Computing Performance Factor

```
   FOR all concepts in the study page;
   BEGIN
      DISPLAY question;
      READ answer;
      Evaluate and assign Performance_Factor;
      IF ( Performance_Factor ≤ 0.4)
          Add to the list of failed Concepts;
   END
```

2.2. WESLA Client

2.2.1. WESLA Client Interface

WESLA Client Interactive Interface is used to get status information about the learner accessing the course.

2.2.2. Learner's Status information

Whenever a learner logs on, status information process at client side will collect the current information about the learner.

3. EXPERIMENTS AND RESULTS

WESLA has been implemented as a server with several clients geographically distributed. The WESLA Model is developed in C programming language on a Intel workstation, running RedHat Linux OS 7.2 . We have simulated a hypothetical Learner Model consisting of Student, Faculty and Expert Model. We have conducted several experiments using WESLA Model, for retrieving the subject material designed for Communication Protocol course for various learners like Student, Faculty and Expert. The course consists of ten modules and the number of concepts in each module varies from 3 to 10. We have also conducted experiment for classifying students into different categories like excellent, good, average and poor student.

3.1. Experiment I: Student as a Learner

When a student access the course as a learner, he/she is made to register the course by assigning him/her with login identification, password and registration number. The student also informed about the way subject pages

are presented to him. Towards end of each module, student is checked for his knowledge/understanding of the course concepts in a question/answer session [3]. His performance level is evaluated and according to his performance, system will decide for his further access of the subject material.

Consider a student, s1 accessing the course, he will be presented with modules of chapter 1 , along with its associated concepts, in a step by step process. Student goes through the concepts of the module 1 completely and towards end of module 1, he is given a question/answer session to check his performance factor.

PF obtained by S1 for Module 1 are:
PF in concept 1 is 0.743680,
PF in concept 2 is 0.178667,
PF in concept 3 is 0.794714,
PF in concept 4 is 0.623167,
PF in concept 5 is 0.797608

Failed in one concept, (i.e.,) Concept 2

He is given two more attempts to complete his failed concept and go to next module 2. All failed concepts are retrieved using retrieval technique, and student's study page containing concept 2 is displayed to the student in HTML format.

Again test is given to student on that particular concept

PF in concept 2 is 0.642532 (second attempt for Module 1)
PF_{avg} for module 1 is computed as,

$$\frac{0.74+0.79+0.62+0.79+0.64}{5} = 0.7$$

The system declares that, student has cleared module 1. After this, second module is presented to him, and same procedure is carried out until he finishes his course.

Simulation was conducted on 100 students, to check their performance, while completing the entire course. Figure 3 shows the graphical plot of total number of attempts excluding first attempt (in terms of concepts) taken by each student to complete the course. This statistics is used to classify the students in different categories. Based on student's performance in the test, they are categorized as poor, average, good and excellent student according to the class performance as shown in Figure 4.

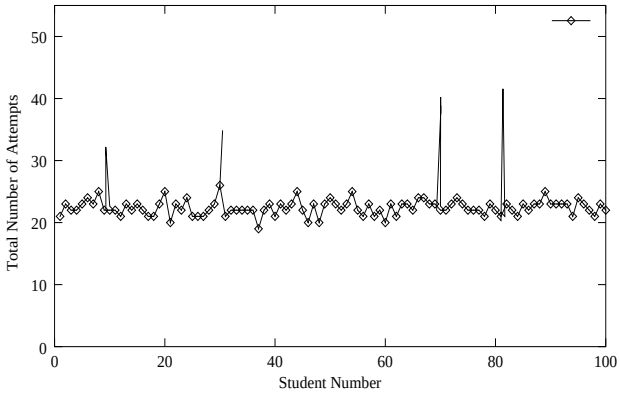

Figure 3: Simulation plot of students vs their no. of attempts taken to complete course

No. of attempts taken	No. of students	student category
<= 20	5	Excellent
21 to 30	91	Good
31 to 40	3	average
41 to 50	1	Reg. Cancelled

Figure 4: Classification of students based on their performance into different categories

3.2. Experiment II: Faculty as a Learner

Faculty is initially given list of modules along with its associated concepts and subconcepts [4]. Simulation results of a faculty F1, with registration number F3344, who access the course for referring to question paper model is shown below.

Enter the User ID : F1
*Password=*******
After authentication, server sends his particulers like
Name = F5
Reg. no F3344, Reg. on 25.6.2001
‘ Faculty can ask for any module and concept according to his requirements and needs.
Looking at the concepts index, faculty specifies the concepts required like
Specify Concepts :M1.C1, M4.C2 or give titles of concepts or modules as query.
All these modules are retrieved and presented to the faculty in HTML format.

3.3. Experiment III : Expert as a Learner

Experts requests concepts to either abreast his knowledge, or to update latest developments in technical fields, or to reshape curriculum by giving suggestions[5]. Simulation results of a Expert E1, with registration number E3340 , who access the course for authoring a book is as shown below.

Enter the User ID : E1,
*Password=********
After authentication, server sends his particulars like
Name = E4
Reg. no E3340, Reg. on 25.6.2001
Expert can ask for any module and concept according to his requirements and needs.
Specify the concepts: , M3.C1, M2.C3, M5.C5

Requested modules along with concepts are retrieved , when expert wants to have surface level knowledge and deep knowledge on a topic.

4. CONCLUSION

Delivering online courses represents revolutions both in the way students learn and in the methods used to deliver asynchronous courses. Patterns set by the model are sure to be standardly employed in many effective distance-education programs in the near future. This channel of teaching represents new horizons to working engineers, offering them a chance to improve and enhance their knowledge. Thus, we feel that we are building one more bridge between the academic community and industry.

REFERENCES

[1] Robin Mason, ” Models of Online Courses”, *ALN Magazine*, Volume 2, issue 2 - October 1998.

[2] Peter Brusilovsky, ”Concept-Based Courseware Engineering for Large Scale Web-based Education”, In: G. Davies and C. Owen (eds.), ”Proceedings of WebNet’2000”, World Conference of the WWW and Internet, San Antonio, TX, Oct. 30 - Nov. 4, 2000, AACE, pp. 69-74.

[3] Yujian Zhou, Martha W. Evens, ”A Practical Student Model in an Intelligent Tutoring System”, *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*. Chicago, IL, 1999, pp. 13-18.

[4] Virvou, M., Moundridou, M, ”Student and Instructor Models: Two Kinds of User Model and their Interaction in an ITS Authoring Tools”, In: Bauer, M., Gmytrasiewicz, P., Vassileva, J. (eds.): *User Modeling 2001: Proceedings of the 8th International Conference UM2001*, Lecture Notes in Artificial Intelligence, 2109. Springer, Berlin (2001) 158-167.

[5] Dr Martin Oliver, ”New professionals’ as Expert Learners: a model for professional development”, *5th Annual Staff and Educational Developers Conference.*