

Individualized Web Course Generation in Distance Learning

Timothy K. Shih, Nigel H. Lin, and Yi-Jen Liu

Multimedia Information Network Lab
Department of Computer Science and Information Engineering
Tamkang University
Taiwan, R.O.C.
e-mail: tshih@cs.tku.edu.tw

Abstract

Web courses are widely available on the Internet. However, individualized course content for each student is hard to deal with. In this paper, we propose a new system, as well as a set of algorithms, for the automatic generation of individualized course contents and tests. The algorithms are based on a newly proposed method, called the SPC table. Experience of such a system is based on a data structure course for undergraduate students.

Key words: Distance Learning, Web-based Tutorial, Virtual University, Distance Education, Assessment

1 Introduction

In an earlier paper [1], we propose a Web Navigation Patrol, which keeps track of learning performance of students, via the use of Web objects and popup quizzes. In another paper [3], a mechanism for student performance assessment on Web-based distance learning courses is presented. Now, we discuss a newly implemented system, which generates individualized Web course contents for each student, based on our previous assessment criteria. We propose a new mechanism, called the SPC table, as the base for our computation. The set of algorithms for course and test generation is also proposed. We use this system to design a data structure course to justify our approach.

2 The SPC Table

The navigation messages from the Web Navigation Patrol are used in the assessment process. Before we discuss our approach, we present a mechanism used in the educational literature. The *Student-Problem table* is a two-dimensional table where rows are student numbers and columns are problem (i.e., test question) numbers. In the table, if a particular student answers a particular problem correctly, the cell is filled with a "1". Otherwise, the cell is filled with a "0". Then, the table is sorted by column and by row from the high occurrence of "1"s to the low occurrence. As a consequence, the upper-left triangular is filled with nearly all "1"s. If a "0" is in the upper-left triangular, it indicates an abnormal situation. Theoretically, students with higher scores should solve those problems, which are answered correctly by most students. Similarly, if a problem is solved by most of the students, a good student is able to solve the problem. This concept is incorporated with an indexing mechanism, known as the *caution index*

computing. The abnormal situation of a particular student or a particular problem can receive an index value above 0.5. The computation mechanism demonstrates a realistic method, which is used by some educational professionals.

However, the Student-Problem tabular mechanism is used in traditional education. That is, exams are given as a summative evaluation of student learning performance. In a distance learning environment, as we discussed earlier, the Web navigation behavior should be used in the formative evaluation process, which can suggest students of an abnormal performance, if the tabular mechanism can be re-designed to include navigation behavior. We propose a new tabular mechanism, called the *Student-Problem-Course table* (SPC table). With the Web technology, our Web Navigation Patrol is able to keep track of students' navigation behavior in Web-based distance learning courses. The behavior is recorded as *Navigation Messages* in a database. In a courseware development tool, an instructor is able to provide with each course unit a set of problems. These problems are used as assignments. On the other hand, these problems can be selected and combined so that midterm exams can be automated. The selection of problems from course units forms a relation between the domain of exam problems and the domain of course units. The relation is recorded as the *Test Relation*. When the midterm exam is given, student *scores* with respect to each test problem are recorded as well. Consequently, in an automatic distance learning environment, we can record three types of information: the navigation messages, the test relation, and the scores. The information is powerful, as the automatic tool uses it to help both the instructors and the students.

Consider the example of a SPC table illustrated in figure 1. Three axes are the student axis, the course axis, and the problem axis. Note that, the table is not three-dimensional. Instead, the table is a compounded table. The lower-right matrix is from the student and the problem axes. Essentially, the S-P table can be used. Elements in the matrix are in a discrete domain (i.e., {0, 1} of view A), where '0' indicates a failure and '1' indicates a success of answering a problem. The lower-left matrix contains *Navigation Factors*. Navigation factors are real numbers in between 0.0 and 1.0, which indicates the degree of visiting for a student, with respect to a particular course unit. The computation of navigation factors is discussed in [1]. The upper matrix is made of the test relation. An element in this matrix could be either a 'Y' to indicate that a particular problem is drawn from a particular course unit, or a 'N' otherwise (omitted from the figure). Note that, using a

course develop tool, a problem is selected from a unique course unit. Generally, the content of problem should test the concept presented in the course unit of which the problem is located.

After the basic elements of each matrix are constructed, the SPC table will compute six caution indices. In the S-P table, there are two caution indices: the CS and the CP indices for students and problems, respectively. In the new SPC table, the six indices are summarized as the following with the usage addressed:

- **The CSa index** (caution index of student w. r. t. problem): indicates the outcome of exam. Students of a high index need special attention. On-line tutorial can be generated for these students.
- **The CPa index** (caution index of problem w. r. t. student): indicates the quality of problems. Problems of a high index value can be re-designed.
- **The CSb index** (caution index of student w. r. t. course unit navigation): indicates the degree of student navigation. An abnormal value means that the student either works too hard (too much navigation), or too lazy (low navigation). On-line tests can be generated for those abnormal students.
- **The CCb index** (caution index of course unit w. r. t. student navigation): indicates which course unit is less visited and less effective. Course content can be revised or access paths can be re-constructed.
- **The CPc index** (caution index of problem w. r. t. course unit): a problem is selected for an exam from a unique course unit. The index represents the degree of usage.
- **The CCc index** (caution index of course unit w. r. t. problem): indicates which course unit does not have a problem chosen in an exam. Problems of this course unit can be selected in the next exam with a higher priority.

The computation of the last four indices is different from those of Student-Problem tables. Due to the space limitation, we only give some brief examples here. In figure 1, the third student from the top has a CSa index, which is equal to 0.8. The third student answers problems 9 and 10 incorrectly (also problems 13 to 16). From the SPC table, we know these two problems are from course units 8 and 12. Of these two course units, the third student has navigation factors .2 and .7. Since .2 is relatively low, course unit 8 is used in the automatic generation of on-line tutorial. Course unit 12 is not included. As another example, the CSb index of the eighth student is 0.7. We found that the navigation factor of the 6th course unit is equal to .4. An on-line test can be generated to check if the student actually understands the course unit. For an instructor, one can find the CPa index of problem 14 is 0.9, which is very high. This indicates that problem 14 from course unit 16 needs to be revised. On the other hand, the CCb index of course unit 15 is 0.6, which means that the course unit can be revised or the access paths to such course unit can be altered. The caution indices of SPC table are very powerful. In the next section, we discuss a set of algorithms for tutorial generation and other automations.

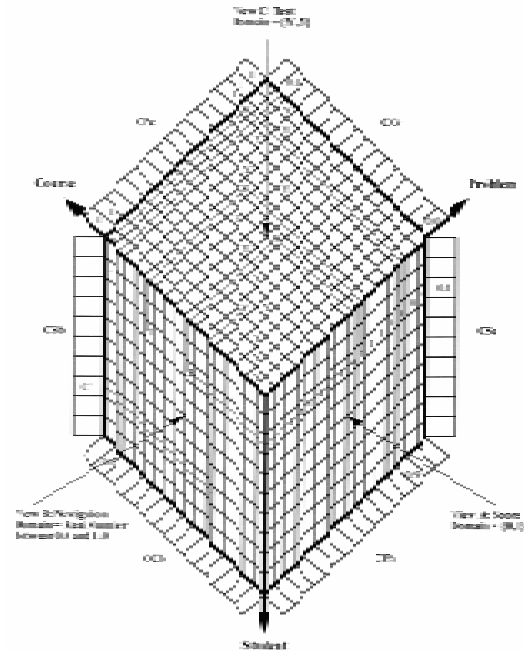


Figure 1: The Student-Problem-Course (SPC) Table

3 Individualized Course Content Generation and Refinement

In section 2, we discussed some examples of the automatic generation of review lectures or tutorials for individual students, who had abnormal performances in navigation and tests. This automation is only one part of the computation. The use of SPC table is diverse. We present some algorithms in this section to demonstrate the powerful computation mechanism of the SPC table.

A Web document for distance learning is an aggregation from a number of course units, which are the basic objects in the process of student assessment and tutorial generation. The organization of these units, from our observation, has the following two types of structure:

- **Web Document Structure:** the hierarchy organization of multimedia objects for the physical representation of a Web document.
- **Web Knowledge Structure:** the navigation sequence of an instructor, who presents the knowledge underlying the Web document structure.

In a traditional lecture, the instructor uses a textbook as the essential reference for a course. When the instructor presents his/her lecture, usually, the structure of textbook is not followed. Indeed, instructors who use the same

textbook may have different topology of presentation. Assuming that, the Web document structure is similar to the structure of a textbook. And, an instructor can use some sort of communication tool to present the content of a Web course. The topology of presentation reflects the knowledge of the instructor, which should be learned by the students. It is the Web knowledge structure that we rely on, to generate review lectures as new Web documents. In the presentation sequence of the instructor (i.e., in the Web knowledge structure), the instructor can set prerequisite course units for a target course unit. Usually, The prerequisites can be some units precede the target in a sequence of presentation. Or, the prerequisites can be the ancestor/sibling units of the target unit in the Web document structure. It is also possible that the two strategies overlap to some extent. The prerequisites can be defined and asserted to the Web course by the instructor via the use of our course development tool. In the following subsections, we discuss the algorithms, which use the prerequisite relation, the navigation sequence of students (obtained by the Web navigation patrol), the content of Web documents, and some thresholds, which are defined for the algorithms.

3.1 Automatic Tutorial Generation

The caution index CS_a of a student is computed based on a set of students who have done a test, which includes a set of problems. If the CS_a is above a threshold, it indicates an abnormal performance of the student with respect to the set of problems. It is possible that the student misses some course content, which leads to the poor test performance. We want to find out which problem in the test the student misses. And, from the problem, we want to find the corresponding course units, which discuss the concept to the problem. If the student did not study hard enough on the course unit, the unit is considered in the initial set of a review tutorial. The following algorithm will traverse the SPC table, from View A to View C and back to View B. The computation requires two thresholds, which are decided by the instructor. And the initial set of course units are computed:

Algorithm: *Tutorial_Set_Initiation*

Preconditions:

Let α_1 be the max threshold of index CS_a

Let β_1 be the min threshold of factor F_b

Input: SPC table

Output: *Tutorial_Init_Set* for each student

Steps:

- For each student S with $CS_a > \alpha_1$
- For each factor F_a in View A of S ,
- where $F_a = 0$
1. Find the factor F_c in View C w.r.t. F_a , where $F_c = 'Y'$
 2. Find the Factor F_b in View B w.r.t. F_c
 3. If $F_b < \beta_1$ then
 - i. Find course C w.r.t. F_b
 - ii. Assert course C into the *Tutorial_Init_Set* for student S

After the initial set is constructed, the prerequisites of each course unit are collected, and linked together according to the Web knowledge structure. Therefore, the tutorial is a

sequence of units (with different portion of reviews), to an individual student. The algorithm is presented below:

Algorithm: *Tutorial_Set_Generation*

Preconditions: None

Input: *Tutorial_Init_Set* for each student

Output: *Tutorial* for each student

Steps:

- For each student S in the *Tutorial_Init_Set*
1. Find the prerequisite course units for course C of the student (i.e., according to the web knowledge structure)
 2. Assert the course units to *Tutorial* and add hyperlinks
- Post *Tutorial* for each student

The generated tutorials are temporary data in the Web course database. The administrator of DBMS maintains these tutorials, until it is time to drop these tutorials.

3.2 Automatic Test Set Generation

It is not necessary to wait until a test is given to the students in order to help them to review the course material. When the student is visiting the Web course, if the caution index, CS_b , of the student is above the threshold, the system uses the navigation factors in View B to identify the course units, which were not traversed carefully. A set of problems is selected randomly from the course units. And, a test is given to the student automatically. The algorithm is presented below:

Algorithm: *Test_Set_Generation*

Preconditions:

Let α_2 be the max threshold of index CS_b

Input: SPC Table

Output: *Test_Set* for each student

Steps:

- For each student S with $CS_b > \alpha_2$
- For each factor F_b in View B,
- where $F_b < \beta_1$
1. Find course C w.r.t. F_b
 2. Select problems of course C
 3. Assert problems to the *Test_Set* for student S
- Post *Test_Set* for each student

The above algorithms generate tutorials or tests for the students. However, it is possible that, the Web course contains inappropriate content. The next subsection discusses how to identify the problematic content.

3.3 Problem and Course Refinement

In a test, theoretically speaking, if a problem is easy, most students will answer the problem correctly, especially the good students. On the other hand, a student below the class average will probably answer a tough problem incorrectly. Abnormal situation occurs if the outcome does not follow the above phenomenon. The strange situation is due to inappropriate or unclear problems in course units. The caution index, CP_a , in View A indicates the situation. If it is above the threshold, we want to find out which course unit is the problem located. And the instructor can improve

the problem. The following algorithm detects these problems:

Algorithm: *Problem_Refinement*

Preconditions:

Let α_3 be the max threshold of index CPa

Input: SPC Table

Output: problem update

Steps:

For each problem P with $CPa > \alpha_3$

1. Find the factor Fc in View C w.r.t. problem P , where $Fc = 'Y'$
2. Find course C w.r.t. Fc
3. Refine problem P in course C

Not only abnormal problems should be refined. Unsuitable course units should be updated as well. Also, if a problem is never used in a test, the system should alert the instructor. The caution index, CCb , indicates the degree of suitability of course units. If the index is too high, the course should be refined, since it is not frequently visited, or cannot be effectively visited due to the erroneous construction of hyperlinks. The algorithm described below finds the problematic course units and the less used problems:

Algorithm: *Course_Refinement*

Preconditions:

Let α_4 be the max threshold of index CCb

Let α_5 be the min threshold of index CCc

Input: SPC Table

Output: course update

Steps:

For each course C with $CCb > \alpha_4$

1. Refine the content of course C and its associated web structure
2. Find the corresponding caution index CCc of course C
 1. If $CCc < \alpha_5$
 1. Suggest instructors to use problems in course C for the test

We have a prototype system running on Microsoft Windows. The system includes two parts. The first is a course development editor allows the instructor to design course contents and problems. It is necessary to use the editor since some of the multimedia objects are stored with their sophisticated data representation, such as their screen coordinates. The data representation helps the Web navigation patrol to detect each multimedia object. The second is the assessment system, which uses algorithms and techniques discussed in this paper.

4 Conclusions and Future Work

One of the most difficulty activities in distance learning is the automatic assessment of student performance [2]. We propose a partial solution, which relies on the SPC table technique and the Web navigation patrol daemon. The proposed system is implemented on the MS Windows. This system is a part of the virtual university software system that we have developed [3]. We use the system in a data structure course. The 40 students have 4 quizzes in the

class. Each quiz has around 20 multiple-choice questions. Individualized tutorials are generated for each student after each quiz. The use of the prototype system shows that, it is possible to have automatic assessment of Web navigation behavior. We believe that, this is a step toward an intelligent system, which can help students to learn better. For instance, from the analysis of student behavior, it is possible to classify students into different groups based on the performance of each individual in a Web course section. For each student group, it is possible to generate a new Web course section, which helps a student to review some of the contents the student did not understand at the first instruction iteration. The review process can be repeated until the student group, or an individual, passes the section. The teaching load of the instructor can thus be reduced. However, the current system did not use the classification and clustering technique to organize student groups. A clustering mechanism can be adopted and used in the next implementation of the system.

Acknowledgement

We want to thank our students, Nigel H. Lin and Yi-Jen Liu for their implementation of the prototype system.

References

1. Chang, Flora Chia-I, "Evaluation Criteria of Student Learning Performance Based on Web Navigation and Popup Quizzes," in Proceedings of the 2000 International Conference on Information Society in the 21 Century: Emerging Technologies and New Challenges, November 5 - 8, 2000, Japan.
2. Klinger, Allen, "Experimental Validation of Learning Accomplishment," in Proceedings of the 1997 Frontiers in Education Conference, pp. 1367 - 1372.
3. Shih, Timothy K.; Chang Shi-Kuo, Ma; Jianhua, and Huang, Runhe, "A Principled Approach for Formative Web Learning Assessment and Adaptive Tutoring," in Proceedings of the Workshop on Web-based Education & Learning (WEL'2000), Hong Kong, June 18 - 20, 2000.