ELSEVIER

Information Sciences 140 (2002) 105–125

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

Intelligent assessment of distance learning ☆

Flora Chia-I Chang

Graduate Institute of Educational Policy and Leadership, Tamkang University, Tamsui 251, Taiwan, ROC

Received 17 November 1999; received in revised form 23 December 2000; accepted 18 May 2001

Abstract

In spite of the great success of many distance learning programs, how to evaluate the behavior of student performance in an off-line distance learning course still remains a question. If visiting Web course content is a basic requirement of performance assessment, it is possible to make an automatic tool to help the instructor and the students in terms of Web interaction analysis. This paper proposes a solution to the automation. A newly proposed mechanism, called Student–Problem–Course (SPC) table is discussed. The mechanism is implemented on a Windows-based platform. Instructors and students both benefit from the system of using the performance evaluation to redesign course material and to review automatically generated tutorials.

Keywords: Intelligent distance learning; Virtual university; Distance education; Student–Problem table; Caution index; Web navigation analysis

1. Introduction

Distance learning is a new trend of education. One of the reasons that accelerate this trend is the Internet. The revolution of Internet technology has changed our daily life. From reading news, sending e-mails, to education as well as entertainment, Internet with multimedia technologies provides a new

☆ This project is supported by the National Science Council of the Republic of China, Taiwan, under grand number NSC 89-2511-S-032-012-X3.

E-mail address: tshih@cs.tku.edu.tw (F. Chia-I Chang).

0020-0255/01/$ - see front matter
PII: S0020-0255(01)00183-9

paradise for information exchange. Among these impacts, distance learning seems to be very important and interesting. One of the attractions of distance learning is its flexibility of instruction. Since students and instructors can be separated spatially and temporally from each other, students in a remote area and part-time students can all benefit from this spatial–temporal flexibility as well as a high degree of information sharing. Yet, the revolution of Internet impacts education not only on short courses, or problem-based learning; one of the revolution results from the impact is the realization of virtual university. A virtual university, unlike a traditional university, provides distance learning both from the spatial and the temporal perspectives. Virtual university allows disabled persons or people at a remote location to learn at home. On the other hand, learning can be pursued anytime via information retrieval from the Internet, which benefits working people from continue education programs. In spite of these benefits, there still exist many difficulties of the realization of virtual university, both from the technique and the management perspectives. One of the drawbacks of current virtual university is the lack of an efficient computer network infrastructure and the integrated software system. In this paper, we propose a software environment for the use of virtual university operations, as well as a new mechanism for the automation of distance learning assessment. The system is designed for administrators, instructors, and students of a virtual university.

There are several differences between a virtual university and a traditional university, when an instructor is delivering lectures. For instance, in a traditional university, an instructor provides on-site supervision. In a virtual university, an instructor can only provide off-site supervision due to the separation of the instructor and students. Especially, it is difficult to perform summative evaluation to student performance in a virtual university. Similarly, team teaching is easier in a traditional university, as compared to the inefficiency of a group discussion over the Internet or on the telephone. In a traditional university, lecture plays the most important role. However, in a virtual university, even with a broadband communication system, which provides real-time video lecture, it is hard to have direct eye contact between the instructor and students. Moreover, the degree of interaction descends in a virtual university. These are the drawbacks of a virtual university.

Nevertheless, virtual university has other advantages in addition to its convenience of the seamless geographical difference. Since networks and computers are the essential facilities in a virtual university operation, it is easier to keep track of individual learning records. This advantage results in a better formative evaluation and diagnostic evaluation. Abnormal learning behaviors are easier to be discovered in the middle of instruction delivery. If appropriate guidance is provided, either automatically or semi-automatically, some students can benefit from the virtual university environment. Moreover, virtual university allows remedy instruction. Since lectures and course materials are made available in a virtual library, it is relatively easier for an individual to review missing lectures. However, to realize the above advantages of virtual university, sophisticated computer software is essential. In this paper, we propose a software system for part of the total solution.

We discuss some other research projects of distance learning software systems, before the global view of our system specification is discussed. With a focus on automatic assessment, we discuss how to summarize information from three perspectives: the usability of course content, the performance of tests, and the relation between course units and tests. The proposed mechanism comes with a solution, which suggests the portion of course units that should be improved, as well as which students should be supervised. We discuss some implementation techniques, and our future work and conclusion is presented.

2. Related work

WAILE [2] is a Web-based intelligent learning system, which provides intelligent tools to support distance learning. Group discussion tools are proposed in CHEER [4]. In CHEER, the concept of virtual discussion room is realized by allowing users to choose whatever communication software they need. As a consequence, different applications have different combination of communication facilities. In a paper, present virtual university (i.e., VLE [1]), students' classrooms are dynamically located. The discussion also points out that, active data is another challenging research issue for distance learning systems. In the MMU project [3], virtual university structure is divided into three levels: Micro University, Virtual University, and Macro University. Micro University can be a software system, which assists an individual to learn from his/her digital documents. Virtual University offers such documents to many students via Web technology and digital communications. The aggregation of Virtual Universities is a Macro University. A join project to integrate many existing Virtual University software systems is currently developed by researchers from USA, Japan, Taiwan, and other countries. Distance learning can be carried out by satellite communications in a remote area. The technique issue and the evaluation of educational benefits of a satellite-based distance learning environment are discussed in [5]. CORAL [8] is a distance learning environment for technical communication education. The system provides a course browser and a group of communication tools. Similar approaches using WWW techniques and Java applets are found in [10,11]. Distance learning systems with interactive classroom and CSCW systems are proposed [6,7]. Laboratory-based distance learning systems are discussed in [13,14]. The discussion of virtual university administration and operation issues is found in [9,17,18]. The benefit and trend of virtual university are also discussed.

3. A distance learning environment

How should a virtual university operate? According to a traditional university, instruction delivery is the most important activity. In order to realize the main activity smoothly, administration is required. A traditional university usually has some student activities and organizations, which need to be properly supported by the university's infrastructure. These are some of the important factors of a traditional university. A virtual university also focuses on instruction delivery. Due to the geographical difference, communication tools should be efficient enough to realize instruction. Communication efficiency points out an important factor: the *awareness impact*. Awareness indicates how strong an individual feels the existence of another person in the communication. For instance, when two persons have an eye contact, the awareness is high. When people are located in different cities and are talking on the phone, the awareness is lower. Sending postal mail has the lowest awareness among these three communication channels. Since a virtual university is distributed geographically, how to use computer networks to guarantee a reasonable awareness is one of the considerations. Awareness certainly affects instruction quality. On the other hand, a virtual university needs administration, which includes activities such as registration, course selection, accounting, and so on. Furthermore, a university needs to ensure that students are learning in order to meet some evaluation standard. This step is to guarantee the quality of education. A virtual university is different from a traditional university in that assessment is difficult. For instance, if the instructor is given an on-line test, how to make sure the student is answering the problems is a difficult task, unless a teaching assistant is sent to the remote classroom. Conclusively, we believe that, a well-considered virtual university supporting system needs to meet the following three criteria:

- *The Administration Criterion*. A virtual university environment needs to have administration facilities to keep admission records, transcripts, accounting records, and so on. These administration tools should be available to administrators, instructors, and students (e.g., checking transcript information).
- *The Awareness Criterion*. Distance learning is different from traditional education. Since instructors and students are separated spatially, they are sometimes hard to 'feel' the existence of each other. A virtual university supporting environment needs to provide reasonable communication tools such that awareness is satisfied.
- *The Assessment Criterion*. Assessment is the most important and difficult part of distance education. Tools to support the evaluation of student learning should be sophisticated enough to avoid unbiased assessment.

In Fig. 1, the three criteria are illustrated with some activities occurring in a virtual university. The three criteria are difficult to achieve. However, it is possible to design such a virtual university environment up to a certain degree

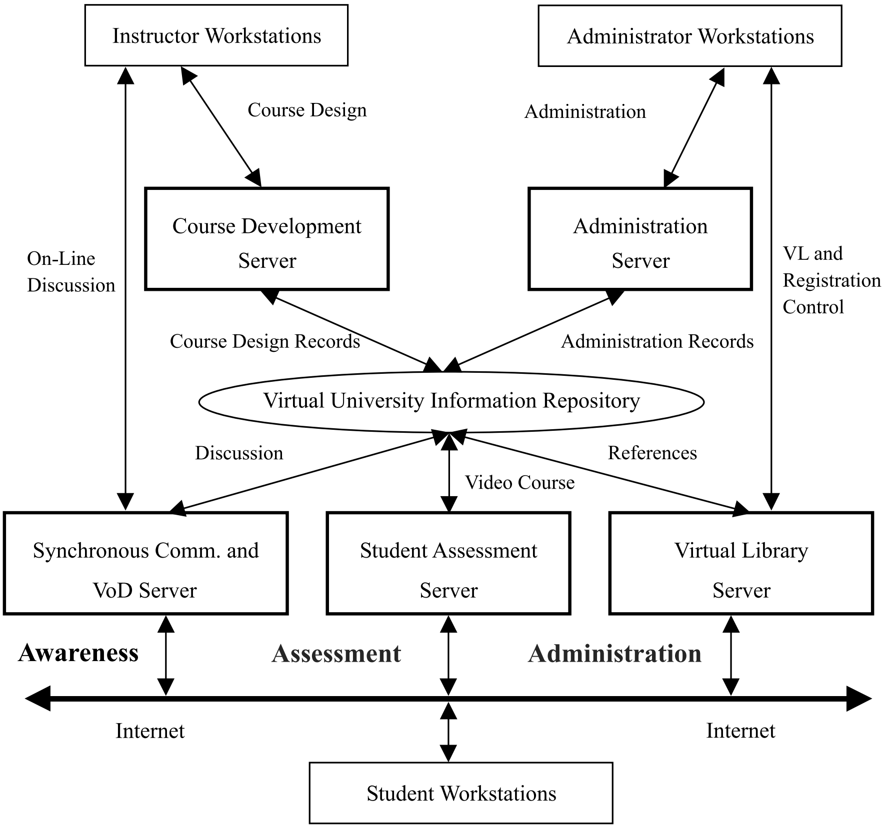

Fig. 1. A virtual university system infrastructure.

of satisfaction. Fig. 1 proposes an infrastructure, with its suggested network logical connections. Assuming that, virtual university operations are running on a single campus. [1] It is feasible to construct several servers on a local area network with a central repository, which is a large scaled database containing all records. The users to such an infrastructure are administrators, instructors, and students. An administrator accesses admission and student records from the help of an administration server. This server runs some tools such as the accounting tool, the curriculum development tool, and others. Instructors use one or more course development servers, which run course design tools, lecture-on-demand tool, and others. The third type of users – the students access a

[1] The definition of a virtual university can be extended to incorporate several campuses on the Internet.

number of servers. The synchronous communication server runs a number of group discussion tools, which are used by instructors and students. The instructor prepares course video, which is digitized and stored as course-on-demand or video-on-demand (VoD) records. These records are video data that can be retrieved by students using different stations on the Internet (or other network infrastructure such as ATM network). A student can also check in or check out references from a virtual library server. Similarly, the student can register for a new course under the registration server. The purpose of different types of server is to balance the load among these machines. The virtual university repository contains several databases, which can be centralized or distributed. Conceptually, the repository should maintain referential integrity in a very high cohesion manner.

The architecture proposed in Fig. 1 only illustrates a general need of such a virtual university environment. One of the most important activities of virtual university operations is to deliver course material. This activity starts from the course development server. With appropriate course content provided by course designer or instructor, Web courses are stored in the Web course database, which is a part of the information repository. Students can access these Web courses from any student workstation. The performance of student learning includes the outcome of test score, as well as the interaction between students and the course material. We believe that, through appropriate and correct interaction, a student can learn better. Hence, formative evaluation is important in the environment. In order to achieve this, we have developed a technique based on Window programming skills. In Section 3.1, we discuss how to keep track of student interactions, which is used as the foundation of automatic assessment.

3.1. The Web navigation patrol

When a student visits our virtual university server, an identification process allows the person's profile to be retrieved from the database. The database includes registration and curriculum development information of the student. When the student visits a particular Web course, the navigation behavior will be recorded. Even though, the recording may involve some privacy violation, the Web site should let the student be aware of this recording process. We suggest that, when a student takes the midterm exam, there is no privacy of his/her answer to the instructor. However, the answers and score of the test should be hidden from other students. In the virtual university system, we treat Web course interaction as part of grading policy. The interaction is part of course requirement. It is not the privacy of the student. However, students are not allowed to access the Web navigation records of others. Before the student is enrolled with a course, the instructor should let him/her know the recording process.

To record the Window navigation messages, we take a similar approach to Cookies. However, in addition to record all URL (i.e., Web hyperlinks) traversed, we need to keep track of each text paragraph and picture visited, as well as the outcome of popup quizzes. The technique uses a mobile agent (the Web navigation patrol, or WNP) to travel to the student workstation, before the station closes its connection to the Web course server. The mobile agent records these navigation messages, after the student station restores the connection, and the messages are sent back to the server. This approach is different from those Web server analyzers, which retrieve navigation from log files. The granularity of our approach is higher, which is required by our assessment analysis. The Web navigation patrol can be implemented using Active Server Pages (ASP) programming techniques, with the support of an underlying database server such as the Microsoft SQL server. The analysis model of performance assessment uses these navigation messages. We discuss the model in the following section.

4. The Student–Problem–Course 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 off-line 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 an SPC table illustrated in Fig. 2. 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 Student–Problem (SP) 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 will be discussed. 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 an 'N' otherwise (omitted from the figure). Note that, using a course development 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 SP 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 redesigned.
- *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

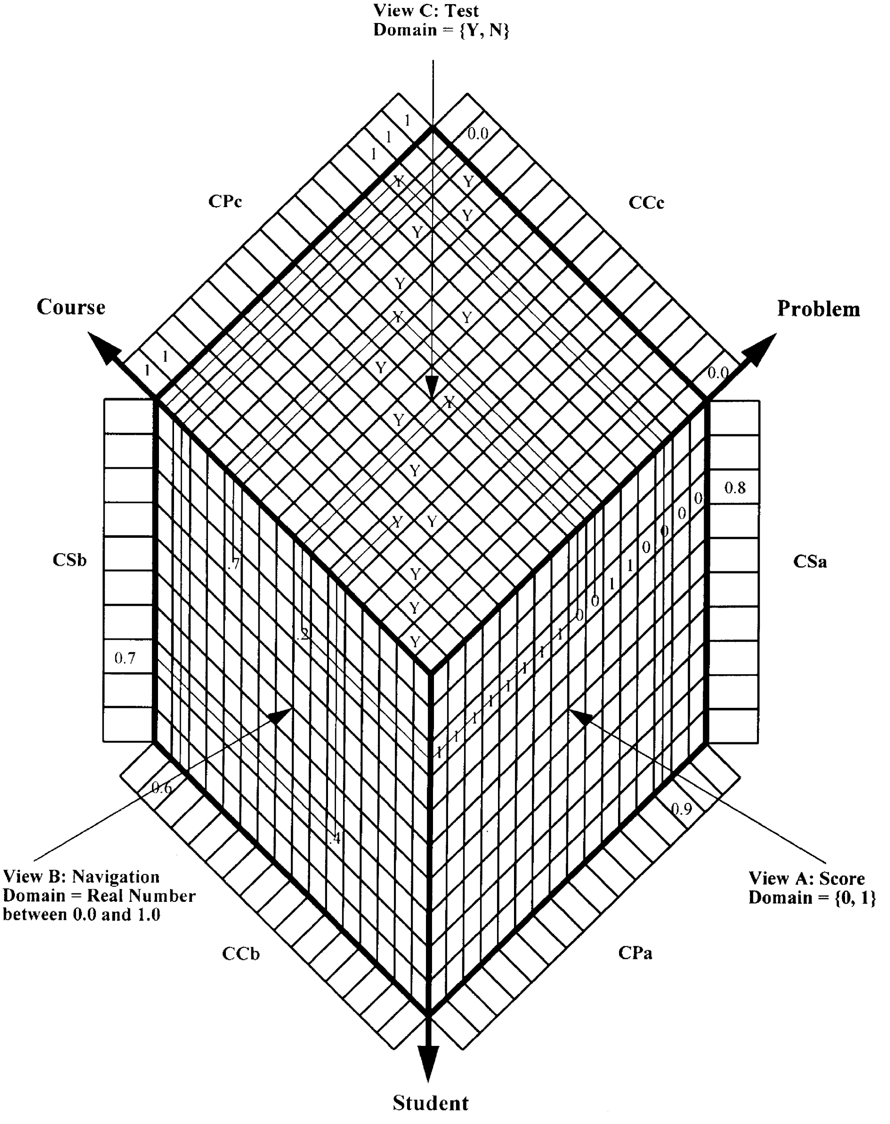

Fig. 2. The SPC table.

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 SP tables. We give some brief examples here. The computation of these indices is discussed in Section 5.

In Fig. 2, 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 0.2 and 0.7. Since 0.2 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 0.4. 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 subsections, we discuss how these values are computed. Also, in Section 6, a set of algorithms for tutorial generation and other automations will be presented.

5. Navigation factor and caution index computation

In Section 4, the SPC table has six types of caution indices (i.e., CSa, CPa, CSb, CCb, CPc, and CCc). The first two types of indices (CSa and CPa) can be obtained according to the SP table. In fact, view A of Fig. 2 is essentially similar to the SP table. However, view A in the SPC table does not sort neither the student records nor the problem records, as the original SP table does. The original SP table sorts these records due to the reason of the analysis of S and P curves. However, if the SPC table is sorted, the order of students records w. r. t. the "problem" and the "course" axes may not be the same. In addition, the sorting process does not help the computation of caution indices, since the computation is based on the values of factors, not the order of these values.

In Sections 5.1 and 5.2, we discuss the computation of *Navigation Factors,* which are elementary components to the computation, and the computation of the last four types of indices (i.e., CSb, CCb, CPc, and CCc).

5.1. Navigation factor computation

As illustrated in Fig. 2, the three views are *score*, *navigation*, and *test*. The score domain is {0,1}, where '1' indicates a correct answer and '0' means a wrong answer, to a particular question answered by an individual student. The test domain represents the association of a question to a course unit. These two domains are self-explanatory. The navigation domain, however, takes some strategic consideration and computation. Before we discuss the computation of navigation factors, we present some considerations here. When an instructor designs a Web course, each course should include some pop-up quizzes, which are HTML like documents. These pop-up quizzes are not for the exam, but to ensure that the student pays attention to the Web course section and to help the student to understand the section. When the student navigates the Web course up to a certain degree of interaction, these pop-up quizzes are presented. These pop-up quizzes also serve as guards so that when a student fails to answer the question, the Web course forces the navigation backtracks to a previous point. When the instructor designs these pop-up quizzes, a condition is attached to each pop-up quiz. Conditions can be the combination of the following:

- *Duration of navigation*: a fixed duration of time, indicating the time difference between two sections, which is defined by the instructor.
- *Frequency of navigation*: a fixed amount of window messages, such as the number of clicks, or the number of objects visited via the mouse pointer.
- *Visiting of a particular object*: objects include buttons, which may indicate the end of a session, or a milestone. Usually, this particular object will trigger a pass/fail quiz before the next section can be visited.

For instance, the conditions of a particular pop-up quiz can be specified as the following. In a Web course section, the student needs to visit at lease 80% of the course material (i.e., multimedia objects), and stay in the section for at least 20 min. When the student meets these requirements, the "next section" button is enabled. As a consequence, the pop-up quiz is presented before the student is able to continue the next section. The pop-up quiz can be a simple multiple choices question, or a simple fill-in-blank question. The outcome of the test decides whether to enable hyperlink to the next section, perform a jump to a specific location, or to backtrack to the beginning of a section.

The purpose of Web navigation patrol also forces the student to follow a predefined navigation sequence. However, the assessment system allows the instructor to turn on or turn off this patrol. As a result, the students either traverse the Web course without guidance, or with guidance. Therefore, the instructor can make a justification of the impact of the Web navigation patrol. The implementation of Web navigation patrol is sophisticated. The mechanism involves parsing a pop-up quiz definition and advanced window programming techniques. We have developed a mechanism to keep track of objects visited by the mouse pointer. These objects include text paragraphs and images. Other objects such as video clips, sound records, and push buttons requires the activation of a mouse click.

The navigation factor considers three elementary messages: the *effective navigation frequency*, the *effective navigation duration*, and the *effectiveness of quiz interaction*. Frequency means how many times each Web object is visited and duration indicates how long a course unit is visited. These two elementary messages need to be "effective". Minimal values and maximal values are designed for these messages. We base on the following two rules to decide the effectiveness:

- If the degree of frequency or duration is less than the minimal threshold, the frequency or duration does not count.
- If the degree of frequency or duration is greater the maximal threshold, it is counted as the maximal value.

Based on the accumulated frequency and duration of each course unit, the effectiveness are computed as:

$$\text{Effective navigation frequency} = \frac{\sum_{\forall \text{Web object}} \text{Effective frequency}}{\sum_{\forall \text{Web object}} \text{Actual Frequency}},$$

$$\text{Effective navigation duration} = \frac{\text{Effective Duration}}{\text{Actual Duration}}.$$

The last elementary message is the effectiveness of quiz interaction, while one is visiting a course unit:

$$\text{Effectiveness of quiz interaction} = \frac{\text{Number of quizzes answered correctly}}{\text{Total number of quizzes}}.$$

Note that, these elementary messages have a value between 0.0 and 1.0. The next step is to compute the compounded navigation factor:

$$\begin{aligned}\text{Navigation factor} = {} & \text{Effective navigation frequency} * W_1 \\ & + \text{Effective navigation duration} * W_2 \\ & + \text{Effectiveness of quiz interaction} * W_3,\end{aligned}$$

where $W_1 + W_2 + W_3 = 1.0$, and $W_1 > 0.0$, $W_2 > 0.0$, $W_3 > 0.0$.

The instructor who designs the course unit also decides these weights. The navigation factor computed is a real number in between 0.0 and 1.0. These factors are the elementary units for caution indices computation.

5.2. Caution indices computation

As we discussed before, the six types of caution indices are important and powerful in the automation of a distance learning assessment system. In this section, the computations of four of these indices are given.

According to Fig. 2, View B, the two axes are student numbers and course unit numbers. As illustrated in Table 1, the navigation factors (i.e., $NF_{i,j}$) are computed according to Section 5.1. In addition, TCN_i is a *Total Course Navigation*, which is the summation of all $NF_{i,j}$'s in the ith row. Similarly, the *Total Student Navigation* (TSN_j) is the summation of all $NF_{i,j}$'s in the jth column. With these summations, the Average Course Navigation (i.e., α) and the Average Student Navigation (i.e., β) are computed according to formulae given in Table 1. The two types of caution indices, CSb, CCb, are also defined. The notation $\lfloor TCN_i \rfloor$ represents that the floor function is applied to TCN_i .

Caution indices CSb and CCb are in between 0.0 and 1.0 in general. When the value of the index is above 0.5, there is an indication of abnormal situation. The higher values point out the worse of behavior. In the assessment system, we have two strategies to raise the problematic students (based on CSb) or course units (based on CCb). The first is according to the percentage of students or course units, which is defined by the course instructor or course material designer. The second is according to an absolute value. The minimal threshold is computed from the minimal index of the percentage, or the absolute value. The values of caution indices higher then the threshold are sources to the generation of warning messages, as well as the automatic generation of individual tutorial for students. The tutorial considers a course unit as the fundamental object. Prerequisite units (relevance course units) are included in the tutorial.

Indices CPc, and CCc represent the degree of association between exam problems and course units. The course-problem table is illustrated in Table 2. Elements in the table are Association Factors (i.e., $AF_{i,j}$). They are in the domain of {**Y**, **N**}, which represents there is an association (i.e., **Y**), and without association (i.e., **N**). Each exam problem is obtained form a course unit. Thus, if we consider each CPc_j as the number of "**Y**" in the column, we have $\forall_j CPc_j = 1$. Index CCc represents the degree of usage of the exam problems w. r. t. a course unit, as computed by the following formulae:

$$\text{Total Number of Problems} = \sum_{i=1}^{m} \#\text{ of problems in Course Unit}_i = \gamma,$$

$$CCc_i = \left(\sum_{j=1}^{n} \#AF_{i,j}\right)/\gamma,$$

Table 1
The Student–Course table

Student number	Course unit number														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	TCN	CSb
1	$NF_{1,1}$	$NF_{1,2}$				$NF_{1,j}$								$NF_{1,n}$	TCN_1	CSb_1
2	$NF_{2,1}$	$NF_{2,2}$												$NF_{2,n}$	TCN_2	CSb_2
3	$NF_{3,1}$													$NF_{3,n}$	TCN_3	CSb_3
4																
5																
6						$NF_{i,j}$								$NF_{i,n}$	TCN_i	CSb_i
7																
8																
9																
10	$NF_{m,1}$													$NF_{m,n}$	TCN_m	CSb_m
TSN	TSN_1	TSN_2				TSN_j								TSN_n		
CCb	CCb_1	CCb_2				CCb_j								CCb_n		

$$\text{Average of course navigation} = \left(\sum_{i=1}^{m} \mathrm{TCN}_i\right) \Big/ m = \alpha,$$

$$\text{Average of student navigation} = \left(\sum_{j=1}^{n} \mathrm{TSN}_j\right) \Big/ n = \beta,$$

where TCN is the total course navigation and TSN is the total student navigation.

$$\mathrm{TCN}_i = \sum_{j=1}^{n} \mathrm{NF}_{i,j},$$

$$\mathrm{TSN}_j = \sum_{i=1}^{m} \mathrm{NF}_{i,j},$$

$$\mathrm{CSb}_i = 1 - \frac{\left(\sum_{j=1}^{n} (\mathrm{NF}_{i,j})(\mathrm{TSN}_j)\right)(\mathrm{TCN}_i)\beta}{\left(\sum_{j=1}^{\lfloor \mathrm{TCN}_i \rfloor} \mathrm{TSN}_j - (\mathrm{TCN}_i)\beta\right)},$$

$$\mathrm{CCb}_j = 1 - \frac{\left(\sum_{i=1}^{m} (\mathrm{NF}_{i,j})(\mathrm{TCN}_i)\right) - (\mathrm{TSN}_j)\alpha}{\left(\sum_{i=1}^{\lfloor \mathrm{TSN}_j \rfloor} \mathrm{TCN}_i - (\mathrm{TSN}_j)\alpha\right)}.$$

Table 2
The Course-problem table

Course unit number	Exam problem number														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	CCc
1	$AF_{1,1}$	$AF_{1,2}$				$AF_{1,j}$								$AF_{1,n}$	CCc_1
2	$AF_{2,1}$	$AF_{2,2}$												$AF_{2,n}$	CCc_2
3	$AF_{3,1}$													$AF_{3,n}$	CCc_3
4															
5															
6						$AF_{i,j}$								$AF_{i,n}$	CCc_i
7															
8															
9															
10	$AF_{m,1}$													$AF_{m,n}$	CCc_m
CPc	CPc_1	CPc_2				CPc_j								CPc_n	

where

$$\#AF_{i,j} = \begin{cases} 1 & \text{if } AF_{i,j} = Y, \\ 0 & \text{otherwise.} \end{cases}$$

The instructors use the degrees of CCc to identify popular exam problems. The system also keeps track of exam problem used. Problems never used before are considered in a new exam.

6. Intelligent tutoring and course refinement

In Section 4, 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. [2] 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 sections, 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.

6.1. Automatic tutorial generation

The caution index CSa of a student is computed based on a set of students who have done a test, which includes a set of problems. If the CSa 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 discusses 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 caution index` **CSa**
 `Let` β_1 `be the min threshold of navigation factor` **Fb**

[2] We have developed a multimedia communication tool, which allows the instructor to use simple line drawing and text paragraphs to annotate a lecture presentation. The communication tool also controls Web browsers. Thus, the Web content displayed on the instructor's station is the same as those on the student stations. An instructor can use this tool to broadcast presentation sequences.

```
Input: SPC table
Output: Tutorial_Init_Set for each student
Steps:
For each student S with CSa > α1
 For each factor Fa in View A of S, where Fa = 0
  Find the factor Fc in View C w.r.t. Fa, where Fc = 'Y'
  Find the Factor Fb in View B w.r.t. Fc
  If Fb < β1 then
    Find course C w.r.t. Fb
    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
   Find the prerequisite course units for course C of the
 student
     (i.e., according to the Web knowledge structure)
   Assert the course units to Tutorial and add hyperlinks
 Post Tutorial for each student on the Web site
```

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.

6.2. 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, CSb, 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 α2 be the max threshold of caution index CSb
```

```
Input: SPC Table
Output: Test_Set for each student
Steps:
For each student S with CSb > α2
  For each factor Fb in View B, where Fb < β1
 Find course C w.r.t. Fb
 Select problems of course C
 Assert problems to the Test_Set for student S Post Test_Set
 for each student on the Web site
```

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

6.3. 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, CPa, 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 α3 be the max threshold of caution index CPa
Input: SPC Table
Output: problem update
Steps:
For each problem P with CPa > α3
 Find the factor Fc in View C w.r.t. problem P, where Fc = 'Y'
 Find course C w.r.t. Fc
 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 suitableness 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 α4 be the max threshold of caution index CCb
 Let α5 be the min threshold of caution index CCc
Input: SPC Table
Output: course update
Steps:
For each course C with CCb > α4
 Refine the content of course C and its associated Web
 structure
 Find the corresponding caution index CCc of course C
  If CCc < α5
   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.

7. Conclusion and future work

One of the most difficulty activities in distance learning is the automatic assessment of student performance. 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 at Tamkang University. The prototype 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.

Another issue of our future work is to study the learning curve of students. The system can be used in a serial of distance learning courses. We need to extend the system to assess student performance among these courses. It is possible to conduct two test sets. One uses the assessment system and another does not. A comparison is required to the outcome of the two test sets. We are also working on other techniques to understand student behavior. For instance, data mining techniques can be applied to the navigation process. The outcome of such a mechanism may help students to identify suitable after-class readings. We hope that, the proposed system and approach will help both the educators and the students.

8. For further reading

[12,15,16]

References

[1] C.Y.Y. Cheng, J. Yen, Virtual learning environment (VLE): a web-based collaborative learning system, in: Proceedings of the 31st Annual Hawaii International Conference on System Sciences, IEEE Press, New York, 1998.
[2] S-W. Chou, S-T. Li, A Web-based intelligent learning system, in: Proceeding of the Software Engineering and Multimedia Applications, August 3–5, vol. 1, 1999, pp. 71–75.
[3] S.K. Chang, E. Hassanein, C.Y. Hsieh, A multimedia micro-university, IEEE Multimedia Magazine 5 (3) (1998) 60–68.
[4] J. Ma, R. Huang, E. Tsuboi, R. Hayasaka, A multimedia collaborative environment for distant education, in: Proceeding of the Fifth International Workshop on Distributed Multimedia Systems, July 23–25, 1998, pp. 175–182.
[5] S.M. Hower, African distance learning via satellite, in: Proceedings of the Tenth International Conference on Digital Satellite Communications, vol. 2, 1994, pp. 532–538.
[6] J-N. Hwang, S. Deshpande, M-T. Sun, A virtual classroom for real-time interactive distance learning, in: Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (ISCAS '98), vol. 3, 1998, pp. 611–614.
[7] T. Dasai, H. Koizumi, K. Yokochi, A multimedia environment in CSCW for interactive distance learning, in: Proceedings of the 1997 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'97), vol. 1, 1997, pp. 398–401.
[8] C. Chou, C.-.T. Sun, A computer-network-supported cooperative distance learning system for technical communication education, IEEE Transactions on Professional Communication 39 (4) (1996) 205–214.
[9] M. Yazdani, D. Bligh, Cooperative learning in a virtual university, in: Proceedings of the Second International Conference on Cognitive Technology, 1997, pp. 251–255.
[10] P. Cadolini, F. Davoli, G. Marescotti, P. Maryni, Developing a distance learning system using Java applets, in: Proceedings of the Global Telecommunications Conference (GLOBECOM'96), 1996, pp. 63–66.
[11] A. Aguilar, R. Eaglin, I. Batarseh, N. Bakir, Distance learning and use of the Internet and the World Wide Web in education, in: Proceedings of the IEEE Southeastcon'97, 1997, pp. 202–206.

[12] J.B. Parekh, Distance learning for enhancing senior productivity, in: Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics, vol. 5, 1998, pp. 4805–4808.
[13] K.D. Taylor, J.W. Honchell, W.E. DeWitt, Distance learning in courses with a laboratory, in: Proceedings of the Frontiers in Education Conference (FIE'96), vol. 1, 1996, pp. 44–46.
[14] M. Khader, W. Barnes, Laboratory based courses in distance learning settings, in: Proceedings of the Frontiers in Education Conference (FIE'96), vol. 1, 1996, pp. 55–58.
[15] J. Vaupel, M. Sommer, Multimedia education, distance learning and electronic commerce applications, in: Proceedings of the International Conference on Virtual Systems and MultiMedia (VSMM'97), 1997, pp. 174–175.
[16] M.T. Davis, Technical communication and distance learning – natural partners, in: Proceedings of the 1997 IEEE International Professional Communication Conference (IPCC'97), 1997, pp. 37–43.
[17] A. Kumar, R. Pakala, R.K. Ragade, J.P. Wong, The Virtual Learning Environment system, in: Proceedings of the Frontiers in Education Conference (FIE'98), vol. 2, 1998, pp. 711–716.
[18] G. Davies, E. Stacey, Virtual universities: are dual mode universities the solution, in: Proceedings of the Frontiers in Education Conference (FIE'98), vol. 2, 1998, pp. 705–710.