

## A New Courseware Diagram for Quantitative Measurement of Distance Learning Courses

FLORA CHIA-I CHANG\*, LUN-PING HUNG<sup>†, \*\*</sup> AND TIMOTHY K. SHIH<sup>\*\*</sup>

*\*Graduate Institute of Educational Policy and Leadership*

*\*\*Department of Computer Science and Information Engineering*

*Tamkang University*

*Tamsui, 251 Taiwan*

*E-mail: fcic@mail.tku.edu.tw*

*E-mail: robin@mail.mine.tku.edu.tw*

*†Department of Information Management*

*Kuang Wu Institute of Technology*

*Taipei, 112 Taiwan*

Web-based distance learning provides a flexible environment for students, but there is a lack of interaction between students and the instructor. To overcome this drawback, we propose a systematic assessment mechanism to enhance interaction between both sides and to maximize student learning performance. This mechanism is presented by means of a new courseware diagrams, which combines useful tools provided by the theory of conceptual map and influence diagram. There are two major aspects to our proposed new courseware diagram. First, a course flow chart can be built systematically and students' learning performance can be improved by having them take different levels of remedial courses based on their performance. Second, through the analysis of students' learning performance, the course content can be adjusted to maximize students' learning results. Finally, this mechanism can be easily used by instructors and students because of its user-friendly interface; in addition, allows the instructor to receive prompt feedback from students.

**Keywords:** systematic assessment mechanism, courseware diagram, conceptual map, influence diagram, remedial course

### 1. INTRODUCTION

The use of the Internet has spread quickly around the world as the technology used to construct the Internet has steadily improved. The Internet provides a convenient and reliable environment in which information can be collected and shared. Thanks to the advantages provided by the Internet, the basic infrastructure of distance education has become more solid and sophisticated. While people welcome the more flexible learning environment provided by the Internet, web-based distance education faces numerous challenges. The current pattern for the development of on-line distance education is as follows: building a platform, improving the current language, creating a new language, setting up standards, developing software engineering, and assessing learning performance. In this paper, we strongly argue that the biggest challenge faced by distance learn-

---

Received October 28, 2002; revised January 29, 2003; accepted March 6, 2003.

Communicated by H. Y. Mark Liao.

ing is the lack of a powerful and scalable evaluation mechanism to interaction among students, instructors, and schools, and providing feedback for three parties. Thus, our study fall focuses on the measurement of students' learning performance.

In a traditional classroom, the instructor can gather students' responses while teaching and revise his/her lecturing style or material so as to maximize the teaching quality. However, in an online environment featuring anytime/anyplace learning opportunities, a significant percentage of the teacher's work must be finalized well before the start of the academic term, meaning that teaching and adjusting class material simultaneously is not possible. Before it is published on the Internet, class material has to be perfectly organized and fully understood by students, which implies that the instructor can predict his/her students' learning ability accurately before the class begins. This is not likely because the two sides can not see each other often. On the other hand, for students learning online, their learning performance and knowledge level cannot be fairly and reasonably measured. This makes it difficult to achieve the goal of establishing standardized, simplified distance education and slows the pace of distance learning development.

In order to overcome the drawbacks caused by the lack of an evaluation mechanism, we have considered the systematic design of and situations occurring in a real classroom, and constructed a strategic, organizational, and reasonable system for building up course content in distance education. In our system, we offer an object-oriented tool for designing course content and a method for measuring students' learning performance. Using the course developing tool, the instructor can easily update and redesign course content after receiving responses from students. The instructor can modify minor portions of the course content after obtaining students' learning performance in each evaluation or can reorganize the entire course content after the class is finished. With regard to the evaluation of learning performance, each student's performance can be fairly measured by comparing it with that of the entire class in the middle of the learning process. Just-in-time analysis of students' learning performance allows the instructor to adjust class material to fulfill students' needs and provides made suggestions for maximizing students' learning performance. Our goal is to assure the quality of online distance learning.

This paper is organized as follows. Section 2 analyzes the previous research done in the field of distance learning. The most important part of this paper is section 3, which describes our proposed courseware diagram in detail. In section 4, we explain the advantages of using the courseware diagram and demonstrate with an example how the courseware diagram for an algorithms course can be constructed. Section 5 describes how the method can be used to construct condensed and expanded courseware diagrams. This is particularly useful when different requirements exist in different situations. An evaluation algorithm for computing AD is given in section 6. Conclusions, along with summary and future research directions, are presented in section 7.

## 2. RELATED WORK

Studies done by researchers in the field of distance education are described below. Some researchers have concentrated on developing adequate platform to support flexible distance learning, such as the Intelligent Tutoring System (ITS) [7, 10, 29, 30], the

Learning Space [27], and SEND [8]. Some researchers have devoted themselves to developing group discussion tools. A paper presented at CHEER [20] introduces the concept of a virtual discussion room, which allows users to choose whatever communication software they prefer. As a consequence, different applications have different combinations of communication facilities. In a paper presenting the Virtual University (i.e., VLE [9]), students' classrooms are dynamically located. This discussion also points out that active data is another challenging research issue for distance learning systems. In the MMU project [6], the virtual university structure is divided into three levels: Micro University, Virtual University, and Macro University. Micro University can be a software system, which helps 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 joint project to integrate many existing Virtual University software systems is currently being developed by researchers from the USA, Japan, Taiwan, and other countries. Moreover, in remote areas, distance learning can be carried out by satellite communications. The related technique issues and an evaluation of the educational benefits of a satellite-based distance-learning environment are discussed in [12]. CORAL 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 [5]. Distance learning systems with interactive classrooms and CSCW systems have been proposed [13]. Laboratory-based distance learning systems are discussed in [31]. Discussions of virtual university administration and operation issues can be found in [33]. The benefits and trends of virtual universities are also discussed. Some researchers have concentrated on developing adequate programming languages for Web Browsers, such as SMIL (Synchronized Multimedia Integration Language) [11] and XML (Extension Multimedia Language) [3]. The functions of sharing and reusing course documents can be found in SCORM (Sharable Object Reference Model) [15], IMS (Instructional Management System) [33], IEEE LOM (Learning Object Metadata) [23, 25], ADL (Advanced Distributed Learning) [34]. Electronic Software is available for distance learning, including the RM system (Resource Management) [17] and MITS (Multimedia Interactive Telelearning System) [19]. WebCT [31] was initially developed at UBC, Canada. The system offers a wide range of products and services for high-level education. The available tools and functions of the system include Administrator, Assignments, Calendar, Chat, Languages Tool, Mail, Manage Students, Quiz and Survey, Self Test, Student Presentations, Student Tips, Syllabus, Track Students, Webcourse Builder, and Whiteboard, which are available for different operating systems and hardware platforms. WebCT has strong partnerships with many organizations or companies, which build add-on tools, such as test tools, course material exchange tools, and others. Another famous distance learning system is the Blackboard e-Learning software platform [32]. Hundreds of distance-learning courses have also been implemented. The system can also be integrated with student management systems or other ERP systems, and secure authentication functions. Statistical data shows the learning status to the instructor as well as to the students. Chetty's study on web-based assessment is one example of research on the assessment of distance learning. In brief, some researchers have concentrated on developing communication tools and group cooperation, while others have concentrated on analyzing and evaluating students' learning performance. Our study falls into the last

group-the measurement of students' learning performance. In our study, we refer to two tools and a theory for assessment, decision-making and goal setting. They are concept mapping, the influence diagram and the theory of the learning cycle. We combine the advantages of all these tools in our courseware diagram.

## 2.1 Concept Map

Prof. Joseph D. Novak at Cornell University originally developed the concept mapping technique in the 1960's. Concept mapping is a two-dimensional graphical representation which constructs the records of information in a hierarchical format starting with the most general concepts and proceeding downward in increasingly greater detail [22]. The use of concept maps to assess students' understanding and learning is not a new idea. It was defined as a strategy by Wandersee, [26] and used in a variety of ways for planning by Novak and Gowin [2]. Amaudin (1986) applied concept map in the field of studying and instruction. Anderson (1989) applied it in the field of assessment [1]. There has also been plenty of work done on implementing concept-mapping techniques. [4]

One advantage provided by concept maps is that responses from students can be obtained. The instructor can receive responses from students and take them into consideration in adjusting the teaching material or style in order to maximize students' learning performance. Based on the concept of understanding the interaction between teachers and students through responses received from students, the courseware diagram we propose for distance learning sets out to achieve the same goal.

## 2.2 Influence Diagram

The Influence Diagram [24] was developed to represent decision problems. It grows linearly (as opposed to exponentially in decision trees) so that larger decision problems can be represented. It is a singly connected DAG (Directed Acyclic Graph) without loops. Two types of nodes and links are utilized in the influence diagram – decision nodes, chance nodes, informational links, and conditioning links. A decision node is depicted by a rectangle or a square that represents a variable under the decision maker's control. An oval or a circle that denotes a probabilistic variable represents a chance node. A conditioning link always points toward a chance node and represents a probabilistic dependence. On the other hand, an informational link always points toward a decision node and denotes available information. In addition, the sequence of decision nodes must be fully ordered. This is known as the no-forgetting condition. A decision is made based on all the outcomes of its direct predecessors. Thus, informational links are arranged in chronological order but conditioning links are not. Another importance issue is the direction of conditioning links. In general, the representation of a decision problem is not unique. That is, the chance nodes do not imply orders [14, 21, 29]. We have found that many features of the influence diagram are very useable for designing our mechanism, but not completely suitable for the modeling of distance learning courseware; therefore, it has been modified in our mechanism.

Besides applying concept maps and the influence diagram, we also apply the theory of the Learning Cycle proposed by Kolb, and attempt to apply the processes of experiencing, reflecting, conceptualization, and planning included in the theory behind our

courseware diagram mechanism [28]. According to the theory, learners obtain understanding through experience, learn from it, and modify their behavior to achieve incremental improvement in their future work. Also, Michael Lawrence Slater suggests that progressive feedback is beneficial to academics and students alike and is an opportunity for improvement [18]. Thus, the most important of our objectives in designing the courseware diagram is to improve learners' performance in their work by enabling adequate adjustment of course contents based on progressive feedback collected through evaluation.

### 3. DEFINITIONS OF COURSEWARE DIAGRAM

From the teaching and learning perspectives, constructing a distance learning course is a decision problem: – How does the courseware benefit students and lead to maximal learning? Tools usually used by decision makers to solve decision problems include concept maps and influence diagrams. Therefore, both tools are included and revised in our courseware diagram. Our objectives and a detailed description of our mechanism are given in this section.

#### 3.1 Objectives of the Courseware Diagram

The key objectives of the courseware diagram are listed below:

- to clearly define each component used in the courseware diagram and identify accepted and restricted links between components;
- to indicate the importance of each component to reflect the percentage of the importance of each course unit;
- to evaluate individual student's learning performance compared with the entire class;
- to build up a consistent, equitable, flexible, and implemental tool;
- to reduce the work load of students, teachers, and the school.

#### 3.2 Using Courseware Diagrams to Represent Course Design Problems

Influence diagram are very useful for designing distance-learning courses. Because of the way courses are designed, a few adjustments must be made in order to reach the final goal – maximizing students learning results. The transformation of an influence diagram into a courseware diagram is described in this section.

##### 3.2.1 Nodes and links used in the courseware diagram

Nodes and links utilized in the influence diagram are used in our courseware diagram but with slight changes. Nodes used in our courseware diagram are evaluation nodes and course nodes that are very similar to decision nodes and chance nodes used in the influence diagram. Decision nodes can represent *evaluation nodes* because a decision node in an influence diagram indicates a decision that is under the decision maker's control. Similarly, an evaluation or a test is conducted or controlled by the instructor. An evaluation can be in the form of a test, quiz, homework, presentation, or even report. The results of evaluation units lead to different consequences. Students who receive higher

graded have the instructor's permission to progress to the next course. On the other hand, students who receive lower their grades have to take a remedial course that is designed to enhance their learning performance. No matter which course they take, the instructor will be able to maximize the students' learning performance by teaching remedial courses to students who perform poorly. Thus, we suggest that a decision node can represent an evaluation node. Furthermore, chance nodes can represent *course nodes* because a chance node is associated with a probabilistic variable, meaning the possibility of the occurrence of the chance node. As with a course node, there is a variable attached to it that indicates the relative importance of the course node. Therefore, it is reasonable to say that chance nodes can be used to represent course units. Also, a value node can represent the *final value* of a student's learning performance. At the end of a series of courses and evaluations, each student has obtained knowledge that is measured by means of evaluation units because the only way to know a student's level of absorbed knowledge is to use evaluation units. Therefore, the only predecessor of the final value is the last evaluation unit. If this requirement is violated, the final value cannot be fairly justified. Once a student's final value is attained, his/her grade will be certified based on the final value. Note that the final unit is a sink node. There is no out-going link from the final unit.

Nodes are connected by different links in the courseware diagram. Links indicate the direction of the next node and the characteristics of its instincts. There are six possible connections among three types of nodes. However, not all of them are allowed in our courseware diagram to prevent violation of normal practice. Two types of links are prohibited. First, links from course units to final value units are prohibited because knowledge values can only be measured by means of exams. Students' real performance in acquiring knowledge cannot be calculated based on taking courses alone. Second, links from evaluation nodes to evaluation nodes are constrained because it is not common to take consecutive tests without taking courses in between. Links from course units to both course unit sand evaluation units and links from evaluation units to final units are defined as *informational links* and represented by solid lines with arrowheads. Information can be regarded as a source of knowledge. Our knowledge is the accumulation of enormous amounts of information. Thus, we use the word "information" for links that transfer information from the previous node to the next node. Links from course units to course units fulfill the requirement of transferring information, and links from course units to evaluation units means a certain amount of information has been accumulated for an evaluation. Also, through the informational link between the last evaluation unit and the final value unit in the courseware diagram, we are able to capture students' actual performance. Regarding a link from an evaluation unit to a course unit, it is defined as a *conditioning link* represented by a dashed line with an arrowhead. In order to achieve the goal of suiting the teaching to the ability of the pupil and the goal of maximizing each student's learning performance, we divided students into several groups based on the results of an evaluation and allow the teacher to give lectures to them based on different materials. Therefore, after an evaluation, the material provided by the instructor depends on the student's performance in the evaluation.

The shapes of evaluation nodes and decision nodes are identical, and the shapes of course nodes and chance nodes are also the same. However, in the influence diagram, informational links and conditioning links are solid lines, while in the courseware dia-

gram; they are solid lines and dashed lines respectively.

### 3.2.2 Basic structure of courseware diagram

An example of applying evaluation nodes, course nodes, informational links, and conditioning links to create a courseware diagram is shown in Fig. 1. In addition to the descriptions mentioned above for nodes and links, other important factors are defined as follows.

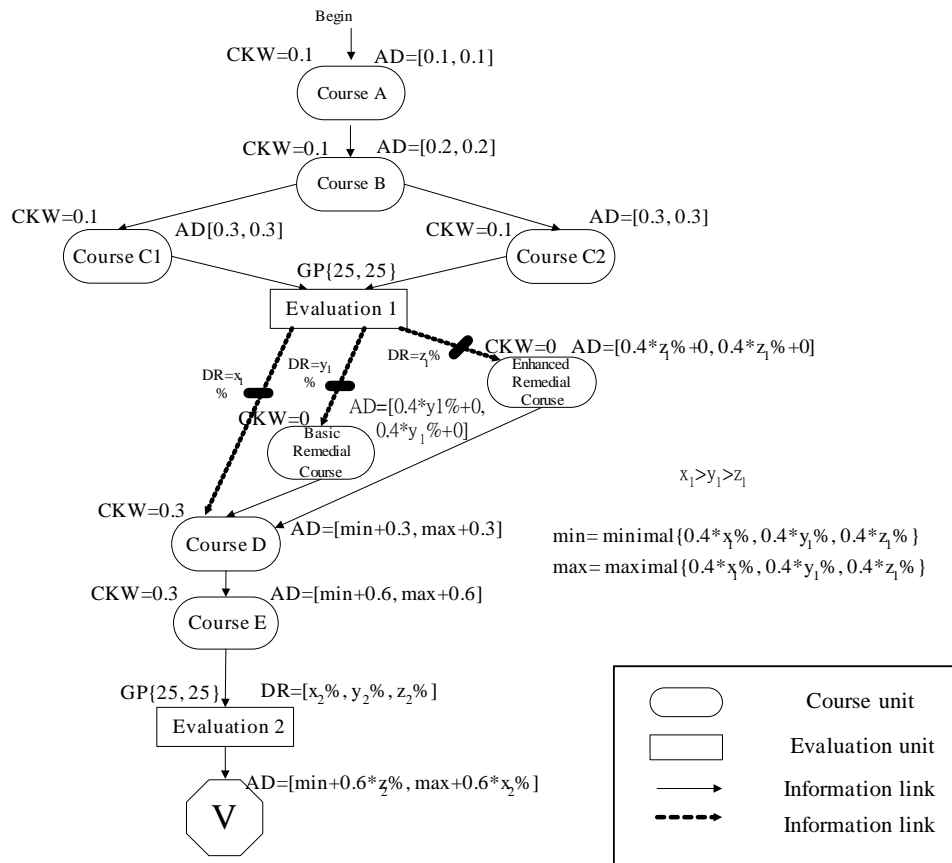


Fig. 1. The basic structure of a courseware diagram.

**Definition 1:** The *Course Knowledge Weight* (CKW) is a value associated with every course unit. A Course Knowledge Weight reflects the importance of the knowledge presented in the course unit. The more important the course unit is, the larger the value of CKW is. The instructor decides the value of CKW. The total value of CKW of all the course units is always equal to one, meaning 100%. The rationale behind this is that we can easily understand the relative importance of each course to the entire course. The CKW value for a remedial course is always zero because new knowledge presented in a

remedial course is nothing. It is reasonable to assign zero to the CKW value for to a remedial course unit. An example of a CKW is “CKW = 0.1”.

**Definition 2:** The *Acquaintance Degree* (AD) is an accumulated course knowledge weight associated with a course unit. It is represented by a pair of numbers. An example of AD is “AD [0.4, 0.6]”. The number on the left side indicates the learning performance of the worst performing group of students or of students who received relatively lower scores in evaluations up to a certain course unit. The number on the right side indicates the learning performance of the best-performing group of students or of students who received relatively higher scores in evaluations up to a certain course unit. The Acquaintance Degree is found by giving an evaluation in order to measure the actual or real learning performance. AD is significant because it reveals the range of the whole class’s actual learning performance. Therefore, an individual student’s learning performance can be compared with the whole class’s learning performance. However, the Acquaintance Degree does not always present the real learning performance because it is possible for the best-performing students in one evaluation to be the worst performing students in the following evaluation. If this situation continues, the Acquaintance Degree does not reflect the real situation. Therefore, we can only say that the Acquaintance Degree is the expected value of a whole class’s learning performance. The actual learning performance depends on various situations that occur.

**Definition 3:** The *Group Percentage* (GP) is a pair of values associated with an evaluation unit and indicates the percentage distribution among students who received lower, average, and higher scores in an evaluation. An example of GP is “GP [25, 25]”, meaning that students are divided into three groups. 25 percent of the students who received relatively lower scores, 25 percent who received relatively higher scores and the rest of the students, 50 percent, who received average scores in an evaluation. The total value of a Group Percentage equals 100. The instructor decides these percentages. Once the percentage distribution has been set at the beginning of the courseware, the rest of the evaluation units have to follow the same percentage distribution for the purpose of continuation. Usually, students are grouped into three different categories, the worst performing students, the average performing students, and the best performing students. The purpose of grouping students is to reinforce students who do not perform well in an evaluation and to calculate the actual learning performance for different groups of students through an evaluation. It is normal to divide students into two groups or to regard them as an entire group, meaning that no remedial course will be given. However, we highly recommend that the instructor embed at least one remedial course. The Group Percentage cannot be omitted no matter if there is a remedial course followed by an evaluation or not. In our courseware diagram, a very special evaluation node is located right in front of the final value node. According to normal practice, no additional regular courses or remedial courses will be given after the last evaluation unit.

**Definition 4:** The *Discount Rate* (DR) is a value associated with every discount bar placed on conditional links. An example of a Discount Rate is “DR = 0.9”. We assume that each student’s score on an evaluation reflects the degree of his/her perception on

previous taught knowledge. Based on the GP assigned by the instructor, every group of students gets an average score. The average score for the group of best-performing students is definitely higher than that of the group of worst performing students. We assume that the average scores for the group are  $X$ ,  $Y$ , and  $Z$  respectively, whereas the relationship among them is  $X > Y > Z$ . The Student Acquaintance Degree before an evaluation multiplied by Discount Rate, which varies among the groups, is the actual learning performance of the students. Thus, the actual learning performance is calculated as AD multiplied by DR.

**Definition 5:** *The Individual Performance (IP)* indicates an individual's actual performance after a series of courses and evaluations. Like the Acquaintance Degree which represents the range of the whole class's performance, the Individual Performance represents a single person's actual performance. This number is accumulated by taking courses and deduced from evaluations taken.

In our courseware diagram, a few restrictions are placed on the design of a course. First, two consecutive evaluation units are not allowed in courseware diagram because the purpose of giving an evaluation is to detect each student's actual learning performance after he/she takes a series of courses. Taking two consecutive evaluations is meaningless. Second, parallel learning is allowed in a courseware diagram. Parallel learning means taking unrelated courses simultaneously within a certain period of time and is a way to differentiate students after they take an evaluation. Therefore, parallel learning may accelerate the learning process. However, one very important feature of parallel learning that cannot be ignored that is every course on the path of parallel learning has to be taken by each student because all the courses together make up the whole class. Parallel learning is indicated in a courseware diagram by multiple paths that may make the courseware diagram which is more complicated. Due to the effect of parallel learning, they can be more than one learning path. This leads to our third restriction. All paths have to merge into the final evaluation node prior to the final value node. The reason for this is obvious. As we stated above, only one evaluation node can exit before the final value node.

Fig. 1 shows the basic structure of a courseware diagram. Each course unit is given a CKW, indicating the importance of each course. The total number of course units is one.

Courses C1 and C2 are not related. Therefore, they are implanted in a parallel learning structure. All course units in a parallel learning structure have to be taken. Therefore, prior to evaluation 1, all courses have to be taken, and AD should be 0.4, which is calculated by adding the CKW of courses A, B, C1, and C2 together. GP [25, 25] is decided by the instructor and stands for all evaluation nodes. The DR values attached to evaluation 1 are  $x_1\%$ ,  $y_1\%$  and  $z_1\%$ , meaning that the average score for the best-performing students is  $x_1$ , the average score for the average performing students is  $y_1$ , and the average score for the worst performing students is  $z_1$ . The AD that is associated with course D is calculated by finding the largest (maximum) and the smallest (minimum) AD among the three different groups after they are discounted by a discount rate and the CKW of course D. The maximum AD is  $0.4 * x_1\% + 0.3$ , and the minimum AD is  $0.4 * z_1\% + 0.3$  for at course D. Between evaluations 1 and 2, there are only two course units, courses D and E. Only material presented in new courses can be included in

the evaluation. Therefore, the total value of CKW (0.6) for courses D and E should be discounted by DR, that is,  $x_2\%$ ,  $y_2\%$  and  $z_2\%$ , respectively, attached to evaluation 2. Therefore, AD after evaluation 2 should be  $[0.4 * z_1\% + 0.6 * z_2\%(0.44), 0.4 * x_1\% + 0.6 * x_2\%(0.66)]$  for the final value node, which indicates that the expected learning performance for the entire class is between the first number and the second number. An individual student's actual performance lies between these two numbers.

#### 4. EVALUATION OF A COURSEWARE DIAGRAM

The most important feature of a courseware diagram is the value of the Acquaintance Degree (AD), indicating the expected best and worst learning performance of the entire class along the path of the courseware diagram. Another important feature is the value of the Individual Performance (IP) indicating the individual student's learning performance along the path of the courseware diagram. The combination of IP and AD indicates one student's performance compared with that of the entire class. The advantages of using the courseware diagram are as follows:

- Evaluation of the individual student's learning performance: Through analyzing the value of IP, the instructor and student are both made aware of his/her performance. From the instructor's point of view, this means being able to measure individual student performance and give each student a fair grade.
- Evaluation of the entire class's learning performance: The Acquaintance Degree attained from the final value gives valuable information to instructor, that is, the value of expected and reachable best and worst learning performance in this class. Although this number does not imply each student's actual learning performance, it does reveal the most possible situation in this class. For example, if the AD attained from the final value node is AD [0.7, 0.9], then the second number (0.9) is the value collected from the best-performing student in every evaluation. However, this person may not exist. For instance, if none of students who received relatively higher scores in the first evaluation falls into the group of best-performing students in the following evaluation, then all the members who fall into the group of best-performing students are students who belong to the group of worst on average performing students. In this case, there is no one who can continuously be the best-performing student in every evaluation. That is why we call AD [0.7, 0.9] the expected value instead of the real value in the courseware diagram.
- Feedback to the instructor: By deploying the IP of an entire class and comparing it with the class AD, the instructor is able to obtain feedback from students.
- Monitoring each student's learning performance along the path of the courseware diagram: Because the process of taking whole class takes a long time, the instructor will not be able to obtain or monitor student's learning ability until after a certain period of time. The design of IP and AD overcomes this problem.
- Maximize the students' learning performance: Due to the design of remedial courses that students take after an evaluation, their learning performance can be reinforced by the studying class material again.

#### 4.1 An Example of Constructing a Courseware Diagram for an Algorithms Course

To show the usage of the courseware diagram, we will give an example in this section. Fig. 2 shows a courseware diagram for an algorithms class. This class starts with a unit on mathematical foundations and ends with the last evaluation unit. For every course unit, there is a Course Knowledge Weight (CKW) attached to it, and the total value of all the course units is one. Due to the topology of parallel learning, unrelated courses can be embedded in a parallel structure. AD is calculated by adding up the previous CKW values on the path. Therefore, at the sorting course, AD is [0.35, 0.35] and at the data structure course unit, AD is [0.25, 0.25]. Before evaluation 1, AD should be 0.45 because there are three courses in total taken by the students. These courses are weighted 0.15 for the mathematical foundations course, 0.2 for the sorting course, and 0.1 for the data structure course, respectively. The result of evaluation 1 is that 25 percent of the students were the best-performing students, who received an average score of 90, 50 percent were the average performing students, who received an average score of 70, and 25 percent of worst performing students, who receive an average score of 50. Therefore, the discount rate are 0.5, 0.7 and 0.9. AD for these three types of students is found to be  $0.45 * 50\%$  for the worst performing students,  $0.45 * 70\%$  for the average performing students, and  $0.45 * 90\%$  for the best-performing students. Before moving on to the next course, the maximum AD is 0.405 ( $0.45 * 90\%$ ), and the minimum is 0.225 ( $0.45 * 50\%$ ). Then, all the students have to continue on to the advanced data structure course (CKW = 0.15), graph algorithm course (CKW = 0.1), number-theoretic algorithm course (CKW = 0.1), computing theory course (CKW = 0.1), and clustering course (CKW = 0.1). However, the graph algorithm course and both the number-theoretic algorithm and computing theory courses can be taken simultaneously. These three courses are designed in a parallel learning structure. The advance data structure course (CKW = 0.15), graph algorithm courses (CKW = 0.1) and number-theoretic algorithm course (CKW = 0.1) which have an accumulated CKW of 0.35, are included in evaluation 2 and should be found using the discount rate (0.5, 0.7, 0.9). Before the computing theory course is taken, the maximum AD is 0.72 ( $0.35 * 90\% + 0.405$ ), and the minimum is 0.4 ( $0.35 * 50\% + 0.225$ ). Then, students move on to the computing theory course. In evaluation 3, the computing theory course (CKW = 0.1) and clustering course (CKW = 0.1) are included and should be determined using the discount rate (0.5, 0.7, 0.9). Before the final value unit is reached, the maximum AD is 0.9 ( $0.2 * 90\% + 0.72$ ), and the minimum is 0.5 ( $0.2 * 50\% + 0.4$ ). Since no more courses exist after evaluation 3, AD at the final value node is [0.5, 0.9]. In conclusion, the expected students' learning performance range between 0.5 and 0.9.

### 5. THE METHOD FOR DEVELOPING CONDENSED AND EXPANDED COURSEWARE DIAGRAMS

In the following section, we will illustrate the transformation process from a condensed diagram to an expanded diagram and the process of reversal transformation. The purposes of the two types of transformation are different, and each transformation leads to different results and advantages.

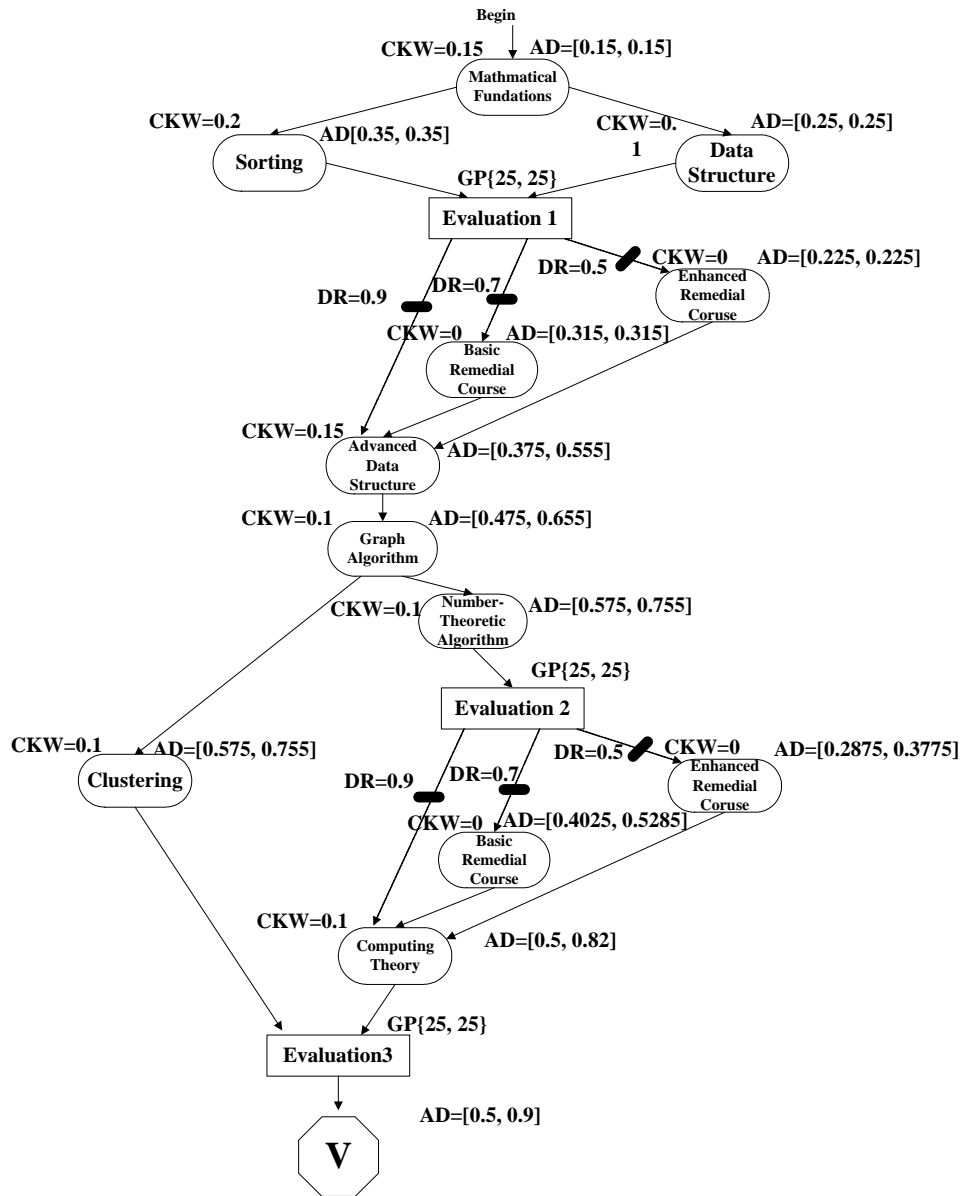


Fig. 2. Courseware diagram for the algorithms class.

### 5.1 Transformation Process From a Condensed to an Expanded Courseware Diagram

The proposed courseware diagram for the algorithms class shown in Fig. 2 originally came from the reduced courseware diagram shown in Fig. 3. The simplest and the very beginning design of the courseware diagram starts with a course unit (CKW = 1)

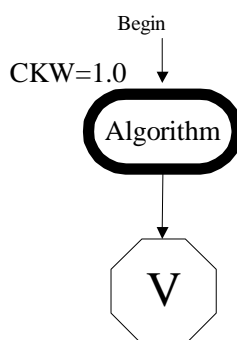


Fig. 3. The reduced courseware diagram for the algorithms class.

leading to the final value node. The purpose of building a reduced courseware diagram is to simplify the design of the courseware diagram and to help the instructor develop a systematic structure. Next, Fig. 4 shows how to develop core courses indicated by bold lined that are divided into two types of courses, related courses (or courses arranged in sequence) and unrelated courses (or courses arranged in parallel). In the algorithms class, related courses are the basic course ( $CKW = 0.45$ ), advanced course ( $CKW = 0.25$ ), and soft computing course ( $CKW = 0.3$ ), and the sequence for taking these courses is depicted in Fig. 4 as well. The instructor decides to teach the students 3 courses in the basic course section, 2 courses in the advanced course section, and 3 courses in the soft computing section. Expanding the structure of Fig. 4 leads to the structure shown in Fig. 5, which show the basic course arranged in a parallel structure, the advance course in a serial structure, and the soft computing in a parallel structure. After the structure for all the courses is plotted, the next step is to decide where to put the evaluation unit. Fig. 6 shows the place where the instructor wishes to put the evaluation unit. According to the requirements of the courseware diagram, an evaluation node has to be placed prior to the final value node in order to measure each student's learning performance. Therefore, evaluation 3 is located in front of the final value. This rule has to be followed or the whole theory will be corrupted. In the course of teaching, the algorithms class, the instructor adds two more evaluation nodes, evaluation 1 and evaluation 2. Where the evaluation nodes will be put is decided freely by the instructor. As long as there is at least one course before an evaluation node, an evaluation can be put anywhere in the diagram. An evaluation node includes two remedial courses that are both weighted 0 since that no new knowledge is provided in remedial courses. The function of the evaluation node is to reduce the value of AD, and it is in a fixed form as shown on the courseware diagram. As shown in Fig. 6, evaluations 1 and 2, which are temporarily represented by extra bold lines, have three conditioning links directed towards the next node, but the evaluation node prior to the final value node has only one conditioning link attached to it. The complete structure of the courseware diagram for the algorithms class is shown in Fig. 2. Another example of expanding the courseware diagram a data structure course is given in Appendix 1.

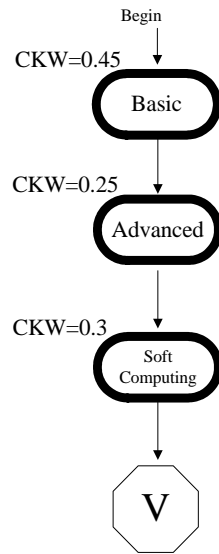


Fig. 4. The expanded structure corresponding to Fig. 3 with the core courses.

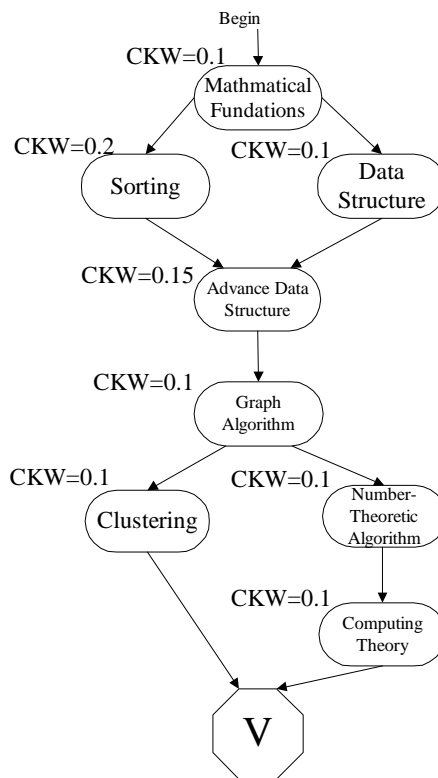


Fig. 5. The expanded structure corresponding to Fig. 4 with all the courses units.

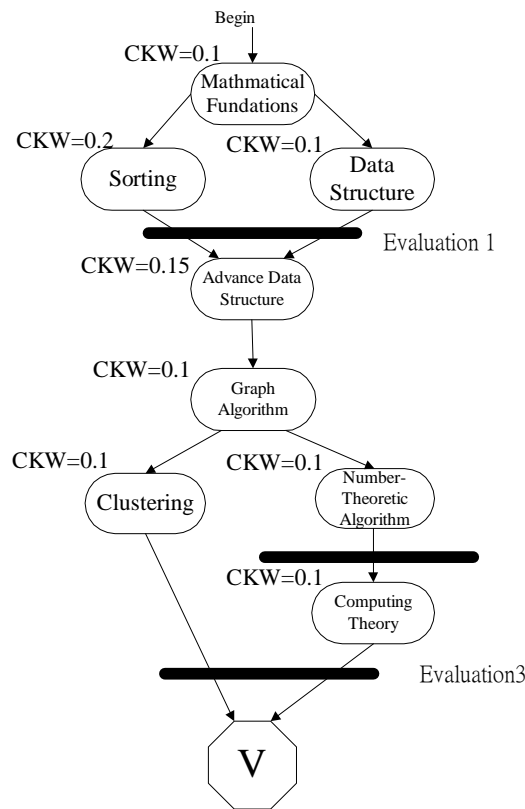


Fig. 6. The expanded structure corresponding to Fig. 5 with all the course and evaluation units.

### 5.2 Transformation Process From an Expanded to a Condensed Courseware Diagram

An expanded courseware diagram can be condensed to its original form by moving in the reverse direction. It is possible to condense a class and leave only the core course units, one evaluation unit and one final value node in the diagram according to the different requirements of the class. A combination of core course units and expanded course units is allowed in the courseware diagram. An example is given in Fig. 7. We assume that if the instructor is asked to design an intensive class for short term training purposes, then he/she only needs combine a few topics belonging to the same core course or to simply combine core course units and expanded core course units together in the same diagram because there is not enough time to finish the whole course. Fig. 7 shows a combination of core course units consisting of an advanced course, a soft computing course, and an expanded core course, which make up a basic course and are depicted in a courseware diagram for an intensive algorithms course. It is possible to change each core course's CKW according to different situations, but we have kept the CKW of each core course the same in this case. In brief, the proposed courseware diagram benefits instructors by offering flexibility in designing the course and the function of evaluation needed to measure each student's learning performance.

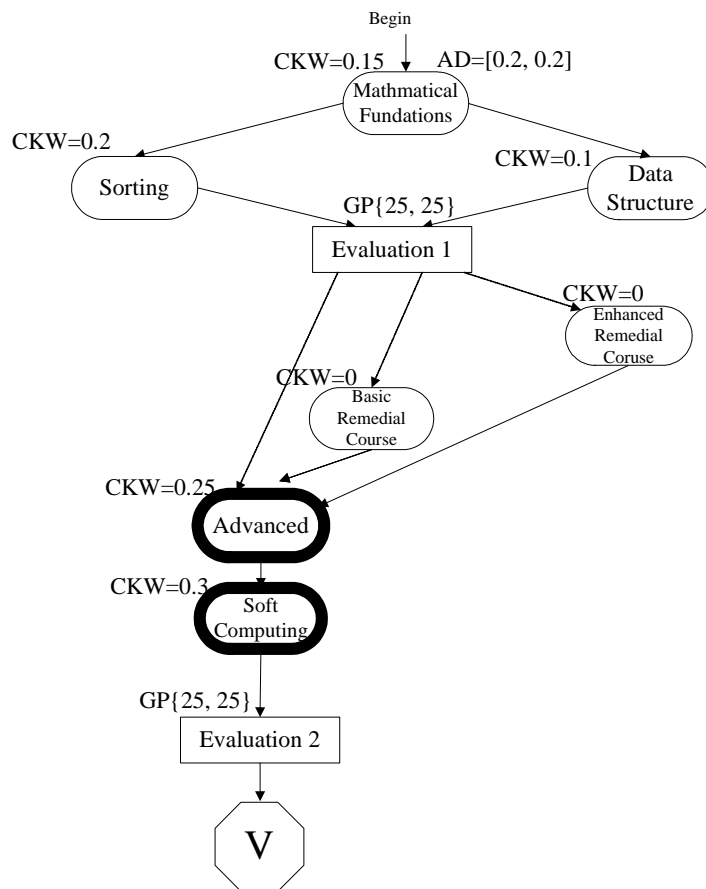


Fig. 7. An example of a courseware diagram for a condensed algorithms class.

### 5.3 Advantages of Transforming the Courseware Diagram

The advantage of building an expanded courseware diagram is that doing so is more systematic. A complete courseware diagram can be designed by following the steps mentioned in subsection 3.4.1: define the core course units, expand core course units, and finally add evaluation units to the diagram. Moreover, the instructor can design a class in a flexible manner because the courses included in each core course are determined based on the instructor's personal judgment and the actual class situation. Any core course unit can simply remain in condensed form without expansion of its contents. This advantage leads to the fulfillment of various requirements involved in designing a courseware diagram for the same class with different timeframes, as show in Fig. 2 and Fig. 7 for the algorithms class, or for based on each student's educational background, such as when designing an algorithm class for both undergraduate students and graduate students. Also, the fact that evaluation units can be placed anywhere in the diagram without restrictions provides even more flexibility for the instructor who designs the courseware diagram.

## 6. AN EVALUATION ALGORITHM FOR COMPUTING AD

An evaluation algorithm for computing AD is illustrated in Fig. 8. In this algorithm, we assume that the evaluation contents include all the course units prior to the evaluation. There are two functions in the following algorithm. First, *CD-Eval* calculates the AD of the final value node. Second, *BFS-Eval* calculates the AD of the core course unit or a compound unit. It is possible to embed condensed units and atomic units together in the same diagram as shown in Fig. 7. Our algorithm can deal with both situations. Also, the calculation of AD for both a parallel structure and a serial structure is included in our algorithm. A more detailed and complicated algorithm that can calculate students' learning performance between two consecutive evaluations is given in Appendix 1.

```

Algorithm CD-Eval(Node, AD)
Require: Node is a pointer, AD is a pair [ADmin, ADmax] of real and Place zero in
        RAD/*RAD is a value of a redundant node*/
if Node is an atomic node then
  if Node is a course node then
    if the child of Node is multiple and n output then
      RAD=RAD+AD*(n-1)
    end if
    ADD CKW to AD
    CD-Eval(Node→child, AD)
  else
    CD-Eval(Node→child, AD)
  end if
else
  if Node is a course node then
    if the child of Node is multiple and n output then
      RAD=RAD+AD*(n-1)
    end if
    place exit_node of condensed node in Exit-Node
    ADD CKW to AD
    BFS-Eval(Node, AD, Exit_Node)
    place sum of CDs of all Exit-Node's parent nodes in AD
    CD-Eval(Exit_Node→child, AD)
  else
    if GP of Node is NULL then
      CD-Eval(Node→child, AD)
    else
      place exit_node of condensed node in Exit_Node

```

Fig. 8. An evaluation algorithm for computing AD.

```

    place the percentages of average score of the I group in  $X_i$ 
  for all child nodes of the test unit do
    place  $[AD_{min} * X_i + CKW, AD_{max} * X_i + CKW]$  in  $AD_i$ 
    BFS-Eval(Evaluation Node  $\rightarrow$  child, AD, Exit_Node)
  end for
  place the minimal ADmin value of parent node of Exit_Node in min
  place the maximal ADmax value of parent node of Exit_Node in max
  place [min, max] in AD CD-Eval(Exit_Node  $\rightarrow$  child, AD)
end if
end if
Return(AD-RAD)

BFS-Eval(Node: pointer,  $[AD_{min}, AD_{max}]$ : Pair of Real, Exit_Node: Pointer) is
  defined as
  if Node = Exit_Node then
    Return
  else
    place AD of Node to  $[AD_{min} + CKW, AD_{max} + CKW]$ 
    for each child node N of Node
      CD-Eval(N,  $[AD_{min} + CKW, AD_{max} + CKW]$ )
    end of BFS-Eval

```

Fig. 8. (Cont'd) An evaluation algorithm for computing AD.

We use “=” as a comparison operator and “set” to represent an assignment statement. The algorithm needs to cope with four types of objects: atomic course units, atomic evaluation units, condensed course units, and condensed evaluation units. For an atomic course unit, its AD is increased by its CKW, and **CD-Eval** is called recursively. For an atomic evaluation unit, the accumulated AD is passed to its descendents. For a condensed course unit, the revised breadth first search function is used to evaluate all the parent nodes of the exit node (i.e., ENode). Then, the AD of ENode is summarized from all of its parent nodes. For a condensed evaluation unit, there are three alternative paths after the evaluation unit. The CKW values of the three alternative course nodes are  $N_x$ ,  $N_y$  ( $= 0$ ), and  $N_z$  ( $= 0$ ), with the percentages of the average scores equal to  $x$ ,  $y$ , and  $z$ , respectively. Breadth first search is applied to the three alternatives again. Finally, the minimal and maximal value of ENode is computed based on all of its parent nodes.

Based on our proposed courseware diagram, we are able to create an authoring tool instructors cause to design a class and to evaluate students' learning performance. Finally, students' grades can be fairly given, and the instructor can carefully examine the design of his courseware and look for possible adjustments by using our courseware diagram.

## 7. CONCLUSIONS

The rapid growth of the Internet around the world has led to substantial numbers of people that are eager to enroll into online distance education courses. However, there are numerous challenges inherent in distance education, especially the assessment of students. Students who enroll in distance education courses tend to have varied educational backgrounds. For the instructor, it is hard to know the students' learning ability before the class starts. However, all of the class materials have to be presented online before the class begins. Therefore, it is crucial for the instructor to determine students' learning performance. Our courseware diagram attempts to deal with this problem. There are two key functions in our courseware diagram. First, the instructor can systematically design the class using drag-and-drop buttons provided by our software system with a user-friendly interface. Second, the instructor can receive just-in-time feedback from students after each evaluation. Based on the results of each evaluation, the instructor will know the students' learning performance and can decide whether or not to make adjustments to his/her class materials. With our software system, students' learning performance can be precisely measured, and the instructor can cope with students with diverse educational backgrounds by constantly adjusting his/her class materials. In conclusion, our proposed mechanism can be used to perform a quantitative analysis based on students' learning performance and provide a flexible course design model for the instructor to use in a distance learning environment.

Continued theoretical and applied investigations along this line of research are needed. We will continue our work on distance learning. This paper has concentrated on evaluating student learning performance. In the future, we will focus on the issue of measuring performance related to the skills, competencies or other characteristics of the instructors and develop a mechanism to implement our idea in order to reach the goal of improving the quality of distance education.

## REFERENCES

1. L. Anderson-Inman and L. Zeitz, "Computer-based concept mapping: active studying for active learners," in *Proceedings of the International Conference on the Computing Teacher*, 1993, pp. 6-11.
2. M. W. Arnaud, J. J. Mintzes, C. S. Dunn, and T. H. Shafer, "Concept mapping in college science teaching," *Journal of College Science Teaching*, Vol. 14, 1984, pp. 117-121.
3. F. Bota, L. Farinetti, and A. Rarau, "An educational-oriented framework for building on-line courses using XML," in *Proceeding of the IEEE International Conference on Multimedia and Expo (ICME)*, Vol. 1, 2000, pp. 19-22.
4. B. Bowen and P. Kocura, "Implementing conceptual graphs in a RDBMS," in *Proceedings of ICCS93*, 1993, pp. 106-125.
5. P. Cadolini, F. Davoli, G. Marescotti, and P. Maryni, "Developing a distance learning system using Java applets," in *Proceedings of the Global Telecommunications Conference (GLOBECOM '96)*, 1996, pp. 63-66.
6. S. K. Chang, E. Hassanein, and C. Y. Hsieh, "A multimedia micro-university," *IEEE*

- Multimedia Magazine*, Vol. 5, 1998, pp. 60-68.
7. S. W. Chou and S. T. Li, "A web-based intelligent learning system," in *Proceedings of the Software Engineering and Multimedia Applications*, Vol. I, 1999, pp. 71-75.
  8. J. S. Correa, D. Fink, C. P. Moraes, and A. A. Sonntag, "Supporting knowledge communities with online distance learning system platform," in *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, 2001, pp. 305-306.
  9. G. Davies and 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.
  10. C. Hegarty, "Statutor: Intelligent tutoring system?" in *Proceedings of the BILETA '96 Conference, Journal of Information, Law and Technology (JILT)*, Issue 3, 1996; <http://elj.warwick.ac.uk/JILT/BILETA/1996/default.htm>
  11. H. Y. Chen, G. Y. Chen, and J. S. Hong, "Design of a web-based synchronized multimedia lecture system for distance education," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Vol. 2, 1999, pp. 887-889.
  12. 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.
  13. J. N. Hwang, S. Deshpande, and 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.
  14. J. K. Kim and S. C. Chu, "Analyzing a class of decision problems: neural network based approach," in *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, Vol. 5, 1998, pp. 36-43.
  15. E. R. Jones, "Implications of SCORM™ and emerging e-learning standards on engineering education," in *Proceedings of the 2002 ASEE Gulf-Southwest Annual Conference*, 2002, pp. 20-22.
  16. M. Khader and 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.
  17. G. Kimovski, V. Trajkovic, and D. Davcev, "Resource manager for distance education systems," in *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, 2001, pp. 387-390.
  18. M. Lawrence-Slater, "Assessing hybrid classes in a flexible learning environment," in *Proceedings of the 31st Annual Frontiers in Education Conference*, Vol. 2, 2001, pp. 1-6.
  19. L. Yuan, Y. Abiza, and A. Karmouch, "Self-guided multimedia courseware system over the Internet," in *Proceedings of the IEEE International Conference on Electrical and Computer Engineering*, Vol. 3, 1999, pp. 1595-1540.
  20. J. Ma, R. Huang, E. Tsuboi, and R. Hayasaka, "A multimedia collaborative environment for distant education," in *Proceedings of the Fifth International Workshop on Distributed Multimedia Systems*, 1998, pp. 175-182.
  21. C. Mudgal and J. Vassileva, "An influence diagram model for multi-agent negotiation," in *Proceedings of the Fourth International Conference on MultiAgent Systems*, 2000, pp. 451-452.

22. J. D. Novak and B. C. Gowin "Concept mapping for meaningful learning," in *Proceedings of the International Conference on Learning*, Cambridge University Press, 1984, pp. 36-37.
23. M. Okamoto, M. Shinohara, Y. Okui, S. Terashima, and M. Hashimoto, "Investigation of learning object metadata and application to a search engine for K-12 schools in Japan," in *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, 2001, pp. 221-224.
24. R. D. Shachter, "Evaluating influence diagrams," *Operating Research*, Vol. 34, 1986, pp. 871-882.
25. D. D. Suthers, "Evaluating the learning object metadata for K-12 educational resources," in *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, 2001, pp. 371-374.
26. J. Wandersee, "Concept mapping and the cartography of cognition," *Research in Science Teaching*, Vol. 27, 1990, pp. 923-936.
27. IBM Software Learning Space, A new version of e-learning (<http://www.lotus.com/home.nsf/welcome/learnspace>)
28. D. A. Kolb, *Experiential Learning: Experience as the Source of Learning and Development*, Englewood Cliffs, N. J., 1984; <http://www.css.edu/users/dswenson/web/PAGEMILL/Kolb.htm>
29. Stottler Henke Associates, Inc, (SHAI), Adaptive Training System™ Projects, (<http://www.shai.com/training.p.htm>)
30. Workshop on Architectures for ITS: Position Paper and Statements of Interest (<http://advlearn.Irdc.pitt.edu/its-arch/papers/>)
31. <http://www.webct.com/>
32. <http://www.blackboard.com/>
33. <http://www.imslobal.org/>
34. <http://www.adlnet.org>

## APPENDIX

### APPENDIX 1.

Algorithm *CD-Eval*(Node , AD)

Require : Node is pointer , AD is a pair  $[AD_{min} , AD_{max}]$  of real

if Node is an atomic node then

if Node is a course unit then

ADD CKW and all immediate predecessors' AD to AD

if Node is synthetic and has n input then

SUBTRACT immediate analytic AD \* (n-1) from AD

end if

*CD-Eval* (Node→child ,  $[AD_{min} , AD_{max}]$ )

else /\*an evaluation unit should pass AD to its successor\*/

if evaluation unit →child is not final value node

if Node is synthetic and has n input then

SUBTRACT immediate analytic AD \* (n-1) from AD

end if

```

if Exit_Node = Null then
    place exit node of evaluation unit in Exit_Node/*exit node is multiple input
    and one output*/
Exit_Node's AD is Separate (evaluation unit)
CD-Eval (Exit_Node , AD)
else
evaluation unit's AD is AD - Exit_Node's AD + Exit_Node's CKW
Temp - AD = Exit_Node's AD - Exit_node's CKW
Place exit node of evaluation unit in Exit-Node
Exit_Node's AD is Separate (evaluation unit)
Exit_Node's AD is Exit_Node's AD+Temp_AD
end if
else
    for DRi I from 1 to n
        place (AD*DRi) in ADi
    end for
        find minimal and maximal value in AD , put into min , max
        AD = (min , max)
    end if
else/*condensed node*/
    if Node is a course unit then
        place exit node of condensed node in Enode
        BFS-Eval (Node , [ADmin + CKW , ADmax + CKW Exit_Node])
        ADD CKW of Exit_node and all predecessor' AD in the condensed node to
        Exit-Node's AD
    if Node is synthetic and has n input then
        SUBTRACT immediate analytic AD * (n-1) form AD
    end if
        CD-Eval (Exit_Node→child , [ADmin of Exit_Node , ADmax of Exit_Node])
else /*Node points to evaluation unit*/
    if GP of Node is omitted then /*pass AD to its successors*/
        CD-Eval (Node→child , [ADmin , ADmax])
    else /*differentiated instruction*/
        if evaluation unit→child is not final value node
            if Node is synthetic and has n input then
                SUBTRACT immediate analytic AD * (n-1) form AD
            end if
        place the percentage of average score of the i group in DRi
        for all child nodes of the evaluation unit do
            place ADmin*DRi + CKWi , ADmax + DRi + CKWi in ADi
BFS-Eval (evaluation unit→child , AD , Exit_Node)
        end for
        place the minimal ADmin value of parent node of Exit_Node in min
        place the maximal ADmax value of parent node of Exit_Node in max
        place (min , max) in AD
        CD-Eval (Exit_Node→child , AD)

```

```

else
  for DRi I from 1 to n
    place (AD*DRi) in ADi
  end for
find minimal and maximal value in AD , put into min , max
  AD = (min , max)
end if

```

```

Separate (evaluation node) /*Set Exit_Node is a exit node*/
  for all child nodes of the evaluation unit do
    place [AD * DRi + CKWi] in ADi
  end for
  find minimal value in ADi , put into min
  find maximal value in ADi , put into max
  AD = (min , max)
  Return AD

```

**BFS-Eval**(Node: pointer , [AD<sub>min</sub> , AD<sub>max</sub>]: Pair of Real , Exit\_Node: Pointer) is defined as

```

if Node = Exit_Node then
  Return
else
  place AD of Node to [ADmin + CKW , ADmax + CKW]
  for each child node N of Node
    CD-Eval( N , [ADmin + CKW , ADmax + CKW])
  end of BFS-Eval

```

**APPENDIX 2.**

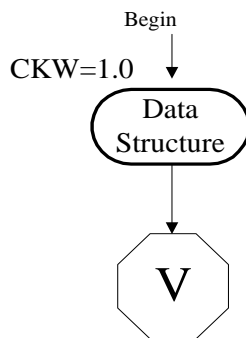


Fig. 9. A reduced courseware diagram for data structure class.

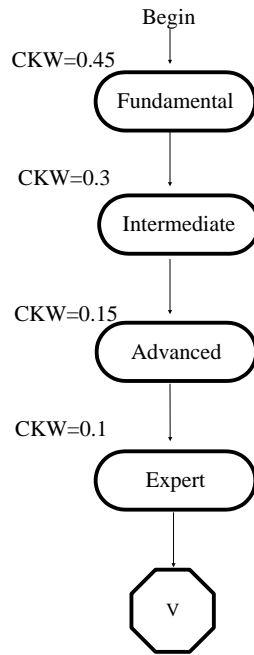


Fig. 10. An expand structure of Fig. 9 with core course.

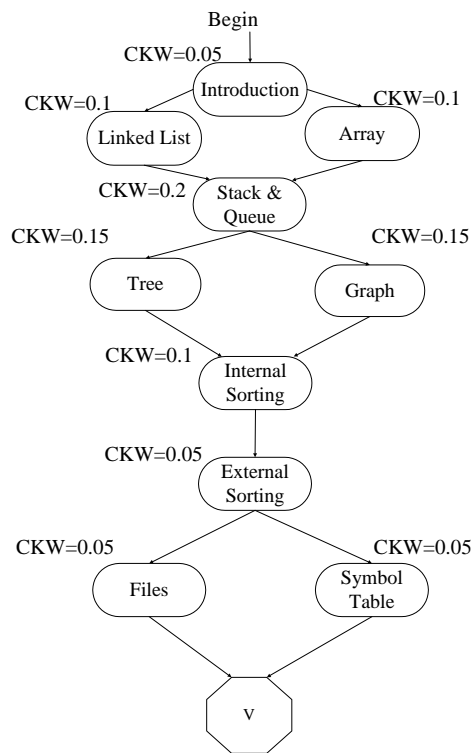


Fig. 11. An expand structure of Fig. 10 with all course units.

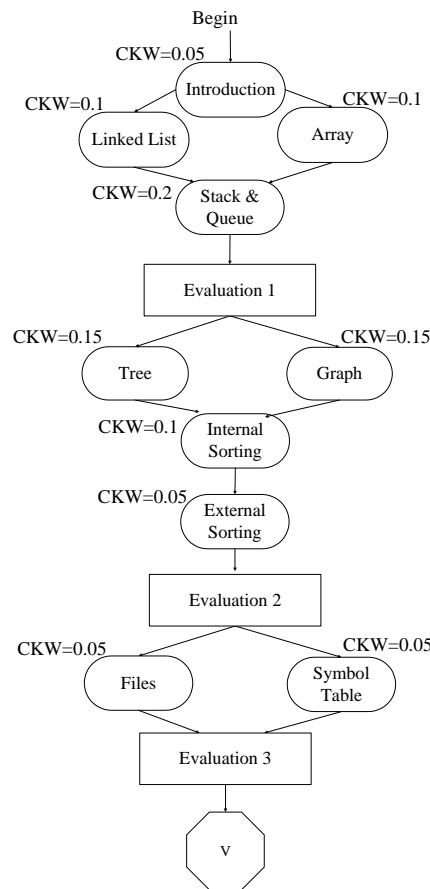


Fig. 12. An expand structure of Fig. 11 with all course and evaluation units.



**Flora Chia-I Chang (張家宜)** received the B.A. degree in Economics from National Cheng Chi University, Taipei, Taiwan, Republic of China in 1979. She received her M.A. degree in Economics from San Francisco State University, and M.A. degree in Administration and Policy Analysis from Stanford University in 1982 and 1983, respectively. She was a Lecturer in Department of Banking and Finance at Tamkang University from 1984-1989. From 1990 to 1995, she attended Stanford University, and received Ed. D. in Higher Educational Administration in 1995. She is currently an Associate Professor in the Graduate Institute of Educational Policy and Leadership at Tamkang University, Taiwan. Her research interests include distance education and total quality management (TQM) in education.



**Lun-Ping Hung (洪論評)** is a Ph.D. student of Information Science and Engineering at Tamkang University in Taiwan. He is also a research member with the Multimedia Information Networking (MINE) Lab and concentrate on the study of distance learning. His research interests include quantitative analysis of distance learning courses, the discovery of customer behavior analysis in mobile commerce and the recommendation technologies for customer relationship management. He has taught at Kwang Wu Institute of Technology for five years and at Tamkang University for two years. Courses lectured are advanced technologies (for MIS students), technical foundation of System Analysis, fundamentals of computer system, as well as management strategies. Prior to entering academia, he worked as a product manager at Shin Lin International Technology Company. He is also a consultant of Show Time Integrated Marketing Company where he is in charge of the development and integration of Electronic Commerce.



**Timothy K. Shih (施國琛)** is a Professor and the Chairman of Department of Computer Science and Information Engineering at Tamkang University, Taiwan, R.O.C.. He is a senior member of IEEE and a member of ACM. His research interests include multimedia computing and networking, distance learning, e-commerce, and content-based multimedia information retrieval. He was a faculty of the Computer Engineering Department at Tamkang University in 1986. In 1993 and 1994, he was a part time faculty of the Computer Engineering Department at Santa Clara University. He was also a visiting professor at the University of Aizu, Japan in summer 1999, and a visiting researcher at the Academia Sinica, Taiwan in summer 2001. Dr. Shih received his B.S. and M.S. degrees in Computer Engineering from Tamkang University and California State University, Chico, in 1983 and 1985, respectively. He also received his Ph.D. in Computer Engineering from Santa Clara University in 1993. Dr. Shih has edited many books and published over 270 papers and book chapters, as well as participated in many international academic activities, including the organization of many international conferences and special issues of international journals. He is the founder and co-editor-in-chief of the International Journal of Distance Education Technologies, published by Idea Group Publishing, U.S.A.. Dr. Shih has received many research awards, including Tamkang University research awards, NSC research awards (National Science Council of Taiwan), and IAS research award of Germany. He also received many funded research grants from both domestic and international agencies. Dr. Shih has been invited frequently to give keynote speeches, tutorials, panels, and talks at international conferences and overseas research organizations.