



# Quantitative Analysis of Distance Learning Courseware

FLORA CHIA-I CHANG

*Graduate Institute of Educational Policy and Leadership, Tamkang University, Taiwan, Republic of China*

**Abstract.** Web-based courseware are widely developed for distance learning programs in continue education, employee training center, and e-learning portals. Usually, a Web-based courseware contains course contents and on-line tests. However, most Web document development tools are not incorporated with a strategic evaluation mechanism, to allow a quantitative analysis of distance learning courseware. We propose an evaluation mechanism and a multimedia tool, based on our Courseware Diagram, to allow a quantitative justification of courseware. Courseware produced by our development system allows an instructor to choose different instruction sequences based on the outcomes of an exam. Alternatively, the courseware may allow self-guided study in a Web-based distance learning program. This paper explains the courseware diagram, the evaluation algorithm, and the design of our courseware development system.

**Keywords:** distance learning, influence diagram, conceptual graph, automatic assessment, courseware diagram, virtual university

## 1. Introduction

Recent research interests of multimedia and communication include tele-medicine, tele-conference, E-Commerce, and distance learning. Among these new research areas, it is possible that distance education and E-commerce systems rely on operating research and decision support methods, such that strategic suggestions can benefit both the service providers and the customers. For instance, in E-commerce, decision support technologies are used in intelligent negotiation of autonomous agents, which helps Web shoppers to seek for best buy goods. In addition, recommendation systems for E-commerce may use statistic or neural network methods to find out potential customers and non-popular products such that the overall revenue will be increased. The success of some E-commerce cases benefit from the new decision support technologies. Thus, decision support and operating research will be very important for these new research directions. On the other hand, distance learning became one of the most important revolutions in college education. Several advantages of distance learning programs include flexibility, convenience, content precision, and the automatic assessment of student performances. Distance learning systems require good communication facilities such as broadband networks and video conference systems. On the other hand, course contents and student performance assessment are as important as the instruction deliver facilities. Distance education includes the distance learning programs in traditional universities, as well as E-Learning portals. Conventional universities uses distance learning programs to attract students in continuous education. E-learning portals are Internet-based companies/organizations with practical distance education programs to help

small and middle size companies in employee/customer training or adult education. However, distance learning contents are hard to build. In addition, current authoring techniques are not designed only for distance learning. It is hard to use current authoring systems, such as Authorware, Director, or the like to control the progress and performance of students. Thus, many distance learning platforms, such as Blackboard,<sup>1</sup> LearningSpace,<sup>2</sup> and WebCT<sup>3</sup> were developed. These systems hardly allow intelligent mechanisms to allow an instructor to monitor student performance. Even some of the systems provide statistic information for the instructors, it is hard to find an automatic method which guides students from time to time in their learning process. In addition, student assessment by the above systems is nearly related to course contents in most cases, as it should be in typical classroom. Therefore, it is important to investigate a strategic assessment method, based on the content of courses, to help both the instructors and the students.

Contents of distance learning courses use multimedia resources, such as text, pictures, audio/video clips, and animations. HTML, as a standard representation language, was used to organize course materials on the Web. Recent standards also include the control of media synchronization in SMIL (Synchronized Multimedia Integration Language), and the enhancement over HTML (i.e., the XML language). Standard formats of distance learning contents including SCORM (Sharable Course Object Reference Mode) were also proposed. Among these standards, formats, and mechanisms, it is hard to find techniques to evaluate the effectiveness of a courseware. Also, it is hard to design strategic methods based on current approaches, to help the instructor in terms of measuring the effectiveness of course contents. In this paper, we propose a revised influence diagram, for designing course structure, which organizes both the instructional materials, as well as tests. Influence diagram is superior to decision tree for many reasons. The diagramming method can be used to build a decision support system. We argue that, the design of a courseware as well as its utilization (i.e., the instruction sequence) is a decision problem. A well-constructed courseware should maintain an effective course structure, with an appropriate number of tests. It is important to maintain this course structure based on our strategic method such that the instruction can proceed in a maximum efficiency.

## **2. Related work**

Distance Learning research project has different focuses. Some emphasis on communication and group collaboration tools, while others uses intelligent techniques. WAILE [7] is a Web-based Intelligent Learning System, which provides intelligent tools to support distance learning. Group discussion tools are proposed in CHEER [14]. 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 presents Virtual University (i.e., VLE [5]), 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 [4], 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 joint 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 [10]. CORAL [6] 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 [1, 3]. Distance learning systems with interactive classroom and CSCW systems are proposed [8, 11]. Laboratory-based distance learning systems are discussed in [12, 16]. The discussion of virtual university administration and operation issues is found in [9, 13, 17]. The benefit and trend of virtual university are also discussed.

In addition to research papers, distance learning systems or software products were developed.<sup>1-3</sup> WebCT<sup>3</sup> was initially developed at UBC, Canada. The system offers a wide range of products and services for high level education. 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 on different operating systems and hardware platforms. WebCT has a strong partnership 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.<sup>1</sup> Hundreds of distance learning courses are also implemented. The system can also be integrated with student management system or other ERP system, and a secure authentication function. Statistic data shows learning status to the instructor as well as to students. LearningSpace<sup>2</sup> is a complete learning management and delivery system. As a part of the IBM Mindspan Solutions Services, the LearningSpace system supports self-based, collaborative, and virtual classroom learning. A comprehensive tracking and reporting function is also implemented to record student performance.

Unlike traditional textbooks, a distance learning courseware is usually represented as hypermedia documents. The most common hypermedia representation relies on HTML, which is widely used by Web browsers. The advantages of HTML include its portability and simplicity. HTML-based Web browsers (such as IE and Netscape) are available in different platforms. Portability accelerates distance learning courseware. Also, since HTML is text-based, it is small in its size. However, references to multimedia resources increases the size of a distance learning courseware.

However, HTML has its drawback. It is hard to control media synchronization. For instance, in a HTML program, it is hard to embed dynamic media, such as video and audio, and to control inter-stream synchronization. To overcome the problem, SMIL (Synchronized Multimedia Integration Language) was developed. Similar to HTML, SMIL is a text-based language allows the description of media synchronization, as well as properties such as the hypertext feature of HTML. SMIL was used in the applications of E-Commerce and distance learning. HTML, SMIL, and their extension (i.e., XML) are general purpose Web languages. Even distance learning courseware can be implemented on Web using these

languages, course documents are different. For instance, course units are reusable and sharable objects. And, there is a strong relation between course content and tests (such as quizzes and assignments). Fortunately, the Sharable Course Object Reference Model (SCORM) was developed. SCORM defined three types of standards: the course structure format (CSF), the run-time environment (RTE), and the meta-data (MD). CSF is based on the XML standard. CSF is similar to a blueprint, which is portable in different platform. Reusable units of the CSF can be shared among its users. RTE defines the requirement of course contents. The requirement allows these contents to be adapted in different distance learning platform. Thus, exchange of course units among different platforms is possible. Meta-data (MD) defines the properties of a courseware. MD allows the system designers to develop mechanism for searching and retrieving course units. Thus, MD and course units are co-existence objects. In addition to SCORM, the Universal Learning Format (ULF) was developed. ULF is based on the SCORM standard, as well as other international formats such as IMS (Instructional management System), IEEE LOM (Learning Object Metadata), ADL (Advanced Distributed Learning), Dublin Core, and vCard.

Representation of courseware, or standard formats are practical solutions to distance learning platforms. From the theoretical perspective, educational theory also contributes to distance education. Conceptual Graph (or Conceptual Map) [2] can be used as a theoretical representation of the structure of course contents. Dependency relations (known as the epistemological order) and course units are used in the graph with weighted values. Thus, the aggregation of dependency relations forms a topology, called the Conceptual Graph. Conceptual Graph helps courseware designers to organize course units. Each of these units has a primary concept to be presented.

### 3. The proposed courseware diagram

Decision support system is one of the success examples that computer programs can help a manager to make a systematic analysis, and lead to a good decision for overall revenue. Decision problem consists of sub-problems, with influences or dependencies to each other. From a knowledge acquisition prospective, the organization of a distance learning courseware is a decision problem—how does the courseware benefit students in a maximal learning capacity. Thus, we consider several approaches on decision science that could benefit our courseware design. Conceptual graph was initially considered. The graphical representation of conceptual graph and *influence diagram* are similar, but influence diagram has different types of nodes, including decision nodes. Conceptual graph is primitive. It is worthy to consider influence diagram as a base for our proposal, as we should explain the reasons in the following sections.

#### 3.1. *Influence diagram as a decision support tool*

The most common method to formulate a decision problem is the *decision tree*. A decision tree contains controllable and uncertain variables. It is easy for the decision maker to explicitly see the value of each possible outcome. Decision tree was used in the construction

of decision support systems. However, in spite of its popularity, decision tree has several drawbacks:

- Independent relations are hardly exploited.
- A large decision problem requires a highly redundant tree.
- If redundant sub-trees are eliminated, the tree will lead to losses of information.
- Tree structure forces decision maker to think forward. But, backward reasoning seems to be more attractive.
- Un-experienced practitioners confuse decision with probabilistic expansions. This may lead to a wrong structure of the formulated problem.

Consequently, for large decision problem, a symmetric decision tree is hard to construct and visualize, even with computers. Therefore, decision tree can only be used in smaller decision problems in general. Nevertheless, *influence diagram* was developed with theoretical and practical advantages over decision trees. Clear and smaller size of the diagram, with a clear distinction between informational and probabilistic nodes, allows exploited independent relations. Influence diagram grows linearly (as opposed to growing exponentially of decision trees) so that larger decision problem can be represented.

An influence diagram is a singly connected DAG (directed acyclic graph) without loop. Two types of nodes are used in an influence diagram—the *decision nodes* and the *chance nodes*. A decision node is represented by a rectangle or a square, which represents a variable under the decision maker's control. A chance node is represented by an oval or a circle, which denotes a probabilistic variable. Links in an influence diagram are divided into types—the *conditioning links* and the *informational links*. A conditioning link always points to 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 to the above definitions, the set of decision nodes must be fully ordered. This is known as the no-forgetting condition. A decision is made with all outcomes of its direct predecessors. Thus, informational links imply a chronological order. However, conditioning links do not imply a chronological order. Another important issue of conditioning link is its direction. In general, representation of a decision problem is not unique. That is, the chance nodes do not imply orders. A conditioning link between two chance nodes can have a reversed link for the same decision problem. Another important feature of influence diagram is, an influence diagram may have a number of redundant links, which can be deduced from the structure of an influence diagram. Without loss of generality, while representing a decision problem, these redundant links can be omitted.

Nodes (decision or chance) with no successors or predecessors are special in the diagram. Typically, an influence diagram has a single sink node, which is called the *value node*. Links toward the value node are also conditioning. A value node can be represented as an octagon, which represents the decision maker's value on the overall decision outcome. Chance nodes without direct predecessors are called *border nodes*. A border node usually represents an independent variable and is important in the construction of an influence diagram.

However, influence diagram is not completely suitable for the modeling of a distance learning courseware, as we should discuss the differences in a revised diagramming technique in the next section.

### 3.2. Courseware diagram as an authoring tool

As we discussed before, the design of a courseware will benefit to the students to receive a maximal learning efficiency. This design can be regarded as a decision problem. Thus, the use of influence diagram in courseware design is natural. However, proper alternation to the diagram is required. We argue that, the following two propositions hold:

- *Decision nodes can be used to represent* test units: In a typical instruction procedure, tests such as quizzes and exams will be given to students to monitor learning performance. According to the outcomes, an instructor needs to suggest appropriate course units for the next lecture, or to design a remedial lecture. Thus, the topology of a courseware can be designed to fit multiple needs, which depend on the decision the instructor makes after a test. Therefore, we believe decision node can be used to represent a test and its evaluation event, before the next step of instruction can be proceeded.
- *Chance nodes can be used to represent* course units: A chance node is associated with a probabilistic variable, which represents the possible influence of the node. When a course unit is learned by a student, there is a degree of understanding associated with the unit, as the probabilistic variable. Therefore, chance nodes can be used to represent course units.

Conclusively, we have two types of nodes, the course units and the test units in the diagram. We call the new definition the *courseware diagram*. In addition to the two types of nodes, the value node (call the *final unit* in courseware diagram) can be treated as a summative evaluation. Summative evaluation gives the semester grade to a student, which justify the learning performance of an individual. Nodes are connected by different links in the courseware diagram. There are six types of connections:

1. From course unit to course unit: *accumulation link*
2. From course unit to test unit: *knowledge link*
3. From course unit to final unit: *knowledge link*
4. From test unit to course unit: *conditioning link*
5. From test unit to test unit: *prohibited*
6. From test unit to final unit: *prohibited*.

Note that, the final unit is a sink node. There is no out-going link from the final unit. In addition, we forbid the use of the last two kinds of connections, for sack of simplicity. Moreover, it is strange to have two consecutive tests in a sequence of class activities even it is possible in the classroom. Thus, the last two types of links are not recommended. Consecutive course units will accumulate degree of understanding (via accumulation links). Links from a course unit to a test unit or the final unit will be regarded as knowledge information (knowledge links) required for the test. Links from a test unit to a course unit will be a conditioning link. In addition, there are strategic suggestions to the use of these links, as summarized below:

- *Conditioning links*: From a test unit, the number of conditioning links is three usually. As we should discuss the Percentage Factor in the next section, the three conditioning links

reflects the outcomes of tests, which subdivide the class into three groups for strategic evaluation. However, it is possible to have a singular conditioning link out of a test unit. As such, there is no subdivision.

- *Accumulation links and knowledge links*: From a course unit, there is one or many outgoing links (accumulation or knowledge links). Multiple links indicate that it is possible to organize the learning topology into parallel course units. However, these parallel course units should be embedded in multiple paths that are merged into a single node (a test unit or a course unit), before the final unit.

As a visualization mechanism, course units are represented as ellipses or circles. Test units are represented as rectangles or squares. And, the final unit is represented as an octagon. These shapes are following the convention of influence diagram. However, it is different from the conceptual graph. In a conceptual graph, concept or course nodes are represented as boxes. But, course units in a courseware diagram are ellipses. Knowledge links are represented as solid lines with arrowhead. Accumulation links and conditioning links are denoted as dash lines with arrowhead. Figure 1 illustrates an example of courseware diagram, as we should explain in the next section.

### 3.3. Evaluation of a courseware diagram

In general the topology of a courseware contains two types of parallelism—concepts that can be learned in parallel and parallel learning sequences for different levels of studies. The first results in multiple accumulation or knowledge links from a course unit. The second is based on the test outcomes. Courseware parallelism also results in an important consequence—the differences of learning impacts. The evaluation of these learning impacts is the most important contribution of our proposed courseware diagram. After a courseware diagram and its contents (course units and test units) are designed by an expert, the courseware can be used by different instructors and students through different topologies. Topology parallelism of a course diagram can be used in different ways, with a quantitative guidance from our system:

- *Differentiated instruction (use norm-referenced evaluation)*: While an instructor is teaching a course, feedbacks from students are gathered from test outcomes. The instructor may choose a suitable learning path, which is recommended by our system. The final path used in the instruction can be compared against the path of an optimal (maximal) learning impact. The comparison can be used as a reference if the same course is taught again.
- *Self-guided study (use criterion-referenced evaluation)*: For life-long learning and employee training, a courseware using our system will help students to find out a suitable learning path. The topology of learning can be designed to include different speed of learning paths. Some of the longer paths may cover details for students who are lack of particular background. Other paths may represent a short cut or an overview. Students who visit these types of Web site can have a self-guided study.

We need to present a few definitions before our evaluation algorithm can be discussed.

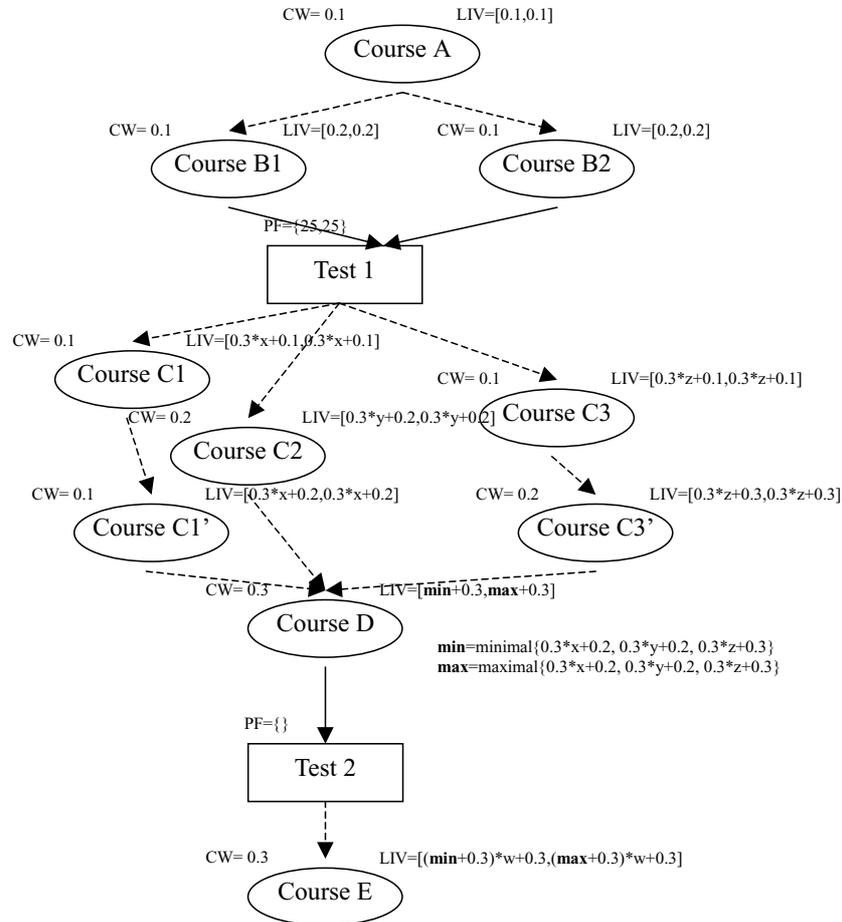


Figure 1. Evaluation of a courseware diagram.

*Definition 1.* A Concept Weight (CW) is a value associated with each course unit. A Concept Weight reflects the importance of knowledge presented in the course unit.

*Definition 2.* A Learning Impact Value (LIV) is a value factor accumulated according to Concept Weights. A LIV reflects the effectiveness of learning in a particular learning sequence w.r.t. a check point (as a course unit). Thus, the accumulated value does not reflect the impact of an individual course unit. LIVs are reduced by test outcomes and CWs.

In general, a Concept Weight is given by the courseware designer, who should decide how much effort a student should spend in order to understand the course unit. But, Learning Impact Values are deduced from the diagram. Since courseware diagram allows alternative sequences of instruction and test outcomes for each student (or class average) are different,

it is possible that the accumulated concept weights (i.e., LIVs) are different. However, the upper and lower bounds of LIVs are computable.

*Definition 3.* A Maximal LIV ( $LIV_{\max}$ ) represents the largest LIV w.r.t. a course unit. A Minimal LIV ( $LIV_{\min}$ ) indicates the smallest value. Differences of LIVs are deduced by test outcomes.

In figure 1, Course A is the first course unit. Its CW is 0.1, which is the  $LIV_{\max}$  and also the  $LIV_{\min}$  of Course A. Similarly, Course B1 and Course B2 accumulate CWs in their LIVs. Note that, B1 and B2 are independent course units that can be learned in parallel. After that, Test 1 is given to the students. As a test unit, Test 1 may result in different outcomes, which can be used as a base for student assessment.

*Definition 4.* Percents of Higher Group (PHG) is the percentage of students who received a relatively higher score in a test. Percents of Lower Group (PLG), on the other hand, is the percentage of lower score students. A Percentage Factor (PF) is a pair of values from PHG and PLG associated with a test unit. These percentages are decided by the instructor. A PF divides the outcome of a test into three consequences (i.e., higher group, middle group, and lower group). If a PF is omitted, there exists only a single consequence from the test unit. As such, the entire class is a group. However, singular consequence of a test unit is not recommended.

In figure 1, Test 1 has its PF equals to  $\{25, 25\}$ . These two numbers are set by the instructor who teaches the class. The PF means that the top 25% of class students are in the higher group, the last 25% of class students are in the lower group, and the rest 50% are in the middle group. Thus, the outcomes of Test 1 result in three alternatives (lower, middle, and upper), as shown in the three conditioning links from the left to the right after Test 1. The three alternatives point to C1, C2, and C3. Assuming that, the lower, the middle, and the higher groups in Test 1 have their average score percentages shown as  $x\%$ ,  $y\%$ , and  $z\%$ , respectively. These percentages may reflect the effectiveness of learning. Thus, the accumulated LIV (i.e., 0.3, accumulated from A, B1, and B2) are reduced by the corresponding percentages, plus their individual CWs (i.g., 0.1 for C1, 0.2 for C2, and 0.1 for C3). Note that, the upper bound and the lower bound of LIV of C1 are still the same (the same for C2 and C3). The LIVs of C1' and C3' are computed in the same manner.

The accumulation links from C1', C2, and C3' are gathered at Course D. Depending on the computed values, the  $LIV_{\min}$  and the  $LIV_{\max}$  of Course D are assigned (shown as **min** and **max** in figure 1). In Test 2, there is no definition of the percentage factor. Thus, a singular conditioning link points to Course E, which as its LIV shown in figure 1. However, singular test outcome is not recommended in the courseware diagram. The use of percentage factor brings the flexibility of a courseware.

**3.3.1. Self-guided study.** Percentage factor examples in figure 1 are designed by the instructor to differentiate student performances, based on norm-referenced evaluation. Alternatively, for self-guided study, the instructor may use these numbers for the percentages of test questions answered correctly for a student (i.e., criterion-referenced evaluation) in

order for him/her to continue the next course unit. Thus, our mechanism is particularly useful for an off-line distance learning course.

### 3.4. A courseware diagram example for data structure course

As an example to show the usage of courseware diagram, we illustrate an example in this section. Figure 2 is a courseware diagram for data structure class. The topology has two main themes: to explain tree structures and to explain sorting algorithms. The first theme

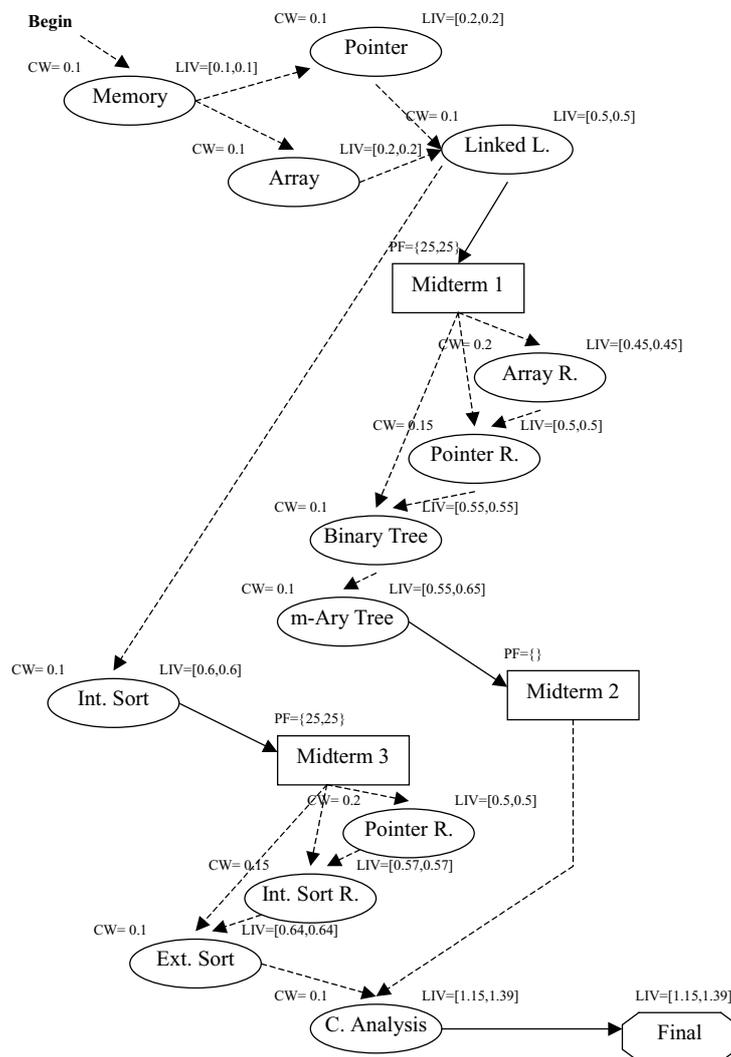


Figure 2. Courseware diagram for data structure class.

is tested in Midterm 1 and Midterm 2. The sorting algorithms are tested in Midterm 3. The common prerequisites are course units Memory, Pointer, Array, and Linked Lists. In Midterm 1, we have  $PF = \{25, 25\}$ , which is the same as Midterm 3. However, Midterm 2 has its PF omitted (since the instructor does not have enough time for the evaluation). We assume that, the top 25 percents of students have an average score percentage of 90%. The middle group has a percentage of 70%, and the lower group has 50%. These numbers reflects the learning impacts of students.<sup>4</sup> The  $LIV_{\min}$  and the  $LIV_{\max}$  of course unit Array Review are  $0.5 * 50\% + 0.2 = 0.45$ . Similarly, the  $LIV_{\min}$  and the  $LIV_{\max}$  of course unit Pointer Review are  $0.5 * 70\% + 0.15 = 0.5$ . And, the  $LIV_{\min}$  and the  $LIV_{\max}$  of course unit Binary Tree are  $0.5 * 90\% + 0.1 = 0.55$ . Moreover, the  $LIV_{\min}$  of course unit m-Ary Tree is taken from the minimal of 0.45, 0.5, and 0.55, plus 0.1 (the CW). And the  $LIV_{\max}$  is the maximal of 0.45, 0.5, and 0.55, plus 0.1. Midterm 2 does not differentiate students. Thus, the accumulated LIV in course unit Complexity Analysis (i.e., C. Analysis) is  $[0.55, 0.65]$  (as a learning impact from the theme of tree structures). In the second path (i.e., the sorting algorithms), LIV is accumulated in course unit Internal Sort (i.e., Int. Sort). Midterm 3 differentiates students into three groups again. The differences of outcome result in the two review units (Pointer R. and Int. Sort R.), and the Ext. Sort unit. The LIVs of there three course units are computed in the similar manner. And, finally, the LIV of course unit Complexity Analysis (C. Analysis) is accumulated (i.e.,  $LIV = [0.5 + 0.55 + 0.1, 0.64 + 0.65 + 0.1] = [1.15, 1.39]$ ). The LIV of the single and the last course unit (i.e., C. Analysis) will be used as the LIV of the final unit.

The evaluation of a courseware diagram involves traversal of Directly Acyclic Graphs (i.e., DAGs). A graph reduction to retrieve DAGs from the diagram and to compute the upper and lower bounds is presented in the next section.

### 3.5. Diagram reduction and LIV computation

In the courseware diagram, course weights and percentage factors are given by the instructor. However, learning impact values are computed. Courseware diagrams are acyclic graphs, with directed edges (i.e., DAG). The first step is to reduce the diagram to a multi-level DAG structure. A *Basic DAG* is a compound object, with a singular entry node, and a singular exit node. For instance, in figure 2, Midterm 1 is an entry node and Binary Tree is an exit node. Similarly, Memory is an entry node and Linked L. is an exit node. The reduced courseware diagram shown in figure 3 has the above two Basic DAGs shown in bold lines. In addition, the whole reduced courseware diagram is a Basic DAG. Essentially a courseware diagram is a recursive structure. It is possible that, in a Basic DAG which represents an exam and its alternative outcomes, there is an embedded Basic DAG, either for another test, or for a compound course unit sequence (such as the Basic in figure 3). Thus, the computation algorithm for LIVs should be able to cope with this recursion.

Note that, LIVs are accumulated from the top to the bottom of the diagram. Thus, even a test unit does not have a LIV, for the sack of completeness, Midterm 2 in figure 3 has a LIV, which is equal to  $[0.5, 0.56]$ . As a matter of fact, a Basic DAG also has a LIV, which represents the accumulated LIV value.

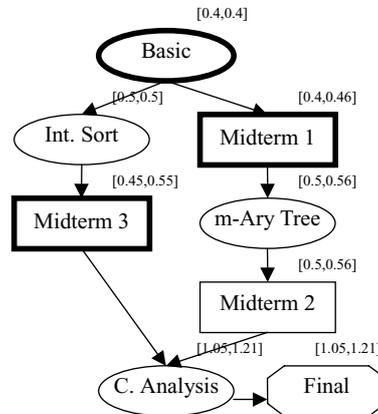


Figure 3. A reduced courseware diagram shown as a DAG.

In figure 4, we present the recursive algorithm to evaluate a courseware diagram. To compute the LIV of **Final**, which is the expected LIV of the diagram, the system calls **CD-Eval** (Memory, [0.0, 0.0]). Note that, the initial LIV is shown as [0.0, 0.0]. **CD-Eval** has two formal parameters. Node is a pointer to any node. And,  $[LIV_{\min}, LIV_{\max}]$  is a pair of real numbers. We use an auxiliary function, **BFS-Eval**, which take as input parameters a Node pointer, a LIV, and an ENode pointer. The ENode pointer points to an exit node of a Basic DAG. The **BFS-Eval** function is a revised breadth first search algorithm. However, the search terminates at the exit nodes. Also, **BFS-Eval** and **CD-Eval** are mutually recursive functions.

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

Based on the method discussed, we designed an authoring system, which allows the instructor to design distance learning courseware using the courseware diagram. In the next section, we should present our design.

#### 4. The tamkang courseware tool

Recent software architecture uses a three-tier approach. That is, the interface of a system is usually run on one computer (possibly with a Web browser). The computation (or the Web

```

CD-Eval(Node: Pointer, [LIVmin, LIVmax]: Pair of Real) is defined as
{
  if Node is an atomic node then
    if Node is a course unit then {
      set LIV of Node to [LIVmin + CW, LIVmax + CW]
      CD-Eval(Node-->child, [LIVmin + CW, LIVmax + CW])
    }
    else { /* a test unit should pass LIV to its descendant nodes */
      set LIV of Node to [LIVmin, LIVmax]
      CD-Eval(Node-->child, [LIVmin, LIVmax])
    }
  else /* compound node */
    if Node is a course unit then {
      set ENode be the exit node in the compound node
      BFS-Eval(Node, [LIVmin + CW, LIVmax + CW], ENode)
      set LIV of ENode to the sum of LIVs of all ENode's parent nodes
      CD-Eval(ENode-->child, [LIVmin of ENode, LIVmax of ENode])
    }
    else /* Node points to a test unit */
      if PF of Node is omitted then { /* pass LIV to its descendant nodes */
        set LIV of Node to [LIVmin, LIVmax]
        CD-Eval(Node-->child, [LIVmin, LIVmax])
      }
      else { /* differentiated instruction */
        set LIV of Node to [LIVmin, LIVmax]
        set ENode be the exit node in the compound node
        set x, y, z be the percentages of average scores of the 3 groups
        set Nx, Ny, Nz be pointers to the first node in the 3 groups
        BFS-Eval(Nx, [LIVmin * x + CW, LIVmax * x + CW], ENode)
        BFS-Eval(Ny, [LIVmin * y + CW, LIVmax * y + CW], ENode)
        BFS-Eval(Nz, [LIVmin * z + CW, LIVmax * z + CW], ENode)
        set min to the minimal LIVmin values of parent nodes of ENode
        set max to the maximal LIVmax value of parent nodes of ENode
        set LIV of Node to [min, max]
        CD-Eval(ENode-->child, [min, max])
      }
    }
  } /* end of CD-Eval */

BFS-Eval(Node: Pointer, [LIVmin, LIVmax]: Pair of Real, ENode: Pointer) is defined as
{
  if Node = ENode then
    return
  else {
    set LIV of Node to [LIVmin + CW, LIVmax + CW]
    for each child node N of Node
      CD-Eval(N, [LIVmin + CW, LIVmax + CW])
    }
  } /* end of BFS-Eval */

```

Figure 4. An evaluation algorithm of courseware diagram.

site) is installed on another machine, while the database management system is running on the third machine. Web browser is a convenient user interface. It is suitable for end users to exploit interesting information on the Internet, including distance learning materials. We take this approach so that all courseware materials are accessible by students on the Web. However, for the instructor, to implement drag-and-drop user interface, which is a basic function to design a courseware diagram, is hard to be implemented on the Web. Even it is possible to use mobile agent technology, to embed these editing functions in Applets, such that course editing is enabled on the Web, it is still difficult for the instructors to operate the system in an efficient manner since mobile agents need to be downloaded to individual computers. Thus, the software system that we designed uses a sophisticated window programming mechanism. The authoring tool for the courseware diagrams, the course units, and the test units is a traditional window program. However, the courseware produced is automatically uploaded to a distance learning Web site, which is accessed by conventional Web browsers.

## 5. Conclusions

Distance learning seems to be a trend of future education. Or, at least distance learning tools will be able to help high level education to move toward another dimension of instruction. Current distance learning tools focus on communication and multimedia presentation technologies. These improvements enable Web-based course materials, video conferencing, video-on-demand lectures, and others. Several standard formats, including platform format and content format were proposed. Yet, assessment of distance learning is still weak. In this paper, we think of courseware design as a decision problem. We studied conceptual graphs and influence diagrams, and proposed a courseware diagram method. The method can be used in a software system, which allows an instructor to design a courseware as making a decision, which can be computed to justify the maximal efficiency. We are still developing a prototype under windowing system. Assessment of distance learning did not get much attention in the past, especially the systematic mechanisms to evaluate the quality of a courseware. We hope that, the assessment criteria, or standard, can be realized by educators, engineers, and policy makers. Thus, future distance learning will provide better courseware and a more accurate control of education quality.

## Notes

1. <http://www.blackboard.com/>
2. <http://www.lotus.com/home.nsf/welcome/learnspace>
3. <http://www.webct.com/>
4. The percentages could also be explained as the effectiveness of learning, which can be decided and justified by an instructor.

## References

1. A. Aguilar, R. Eaglin, I. Batarseh, and 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.

2. B. Bowen and P. Kocura, "Implementing conceptual graphs in a RDBMS," in Proceedings of ICCS93, Canada, 1993, pp. 106–125.
3. 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.
4. S.K. Chang, E. Hassanein, and C.Y. Hsieh, "A multimedia micro-university," IEEE Multimedia Magazine, Vol. 5, No. 3, pp. 60–68, 1998.
5. C.Y.Y. Cheng, and 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, 1998.
6. C. Chou and Chuen-Tsai Sun, "A computer-network-supported cooperative distance learning system for technical communication education," IEEE Transactions on Professional Communication, Vol. 39, No. 4, 205–214, 1996.
7. Shih-Wei Chou and Sheng-Tun Li, "A web-based intelligent learning system," in Proceeding of the Software Engineering and Multimedia Applications, 1999, Vol. 1, pp. 71–75.
8. T. Dasai, H. Koizumi, and 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), 1997, Vol. 1, pp. 398–401.
9. G. Davies and E. Stacey, "Virtual universities: Are dual mode universities the solution," in Proceedings of the Frontiers in Education Conference (FIE'98), 1998, Vol. 2, pp. 705–710.
10. S.M. Hower, "African distance learning via satellite," in Proceedings of the Tenth International Conference on Digital Satellite Communications, 1994, Vol. 2, pp. 532–538.
11. Jenq-Neng Hwang, S. Deshpande, and Sun Ming-Ting, "A virtual classroom for real-time interactive distance learning," in Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (ISCAS'98), 1998, Vol. 3, pp. 611–614.
12. M. Khader and W. Barnes, "Laboratory based courses in distance learning settings," in Proceedings of the Frontiers in Education Conference (FIE'96), 1996, Vol. 1, pp. 55–58.
13. A. Kumar, R. Pakala, R.K. Ragade, and J.P. Wong, "The virtual learning environment system," in Proceedings of the Frontiers in Education Conference (FIE'98), 1998, Vol. 2, pp. 711–716.
14. J. Ma, R. Huang, E. Tsuboi, and R. Hayasaka, "A multimedia collaborative environment for distant education," in Proceeding of the Fifth International Workshop on Distributed Multimedia Systems, 1998, pp. 175–182.
15. R.D. Shachter, "Evaluating Influence Diagrams," in Operations Research, Vol. 34, No. 6, 871–882, 1986.
16. K.D. Taylor, J.W. Honchell, and W.E. DeWitt, "Distance learning in courses with a laboratory," in Proceedings of the Frontiers in Education Conference (FIE'96), 1996, Vol. 1, pp. 44–46.
17. M. Yazdani and D. Bligh, "Cooperative learning in a virtual university," in Proceedings of the Second International Conference on Cognitive Technology, 1997, pp. 251–255.



**Flora Chia-I Chang** 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.