Courseware Development Using Influence Diagrams Supporting e-Learning Specifications[*]

NIGEL H. LIN, WEN-CHIH CHANG, TIMOTHY K. SHIH
AND HUAN-CHAO KEH
Department of Computer Science and Information Engineering
Tamkang University
Tamsui, Taipei County, 251 Taiwan

Without a systematic assessment mechanism, it is hard for teachers to design e-learning courseware and assess students' on-line learning behaviors. This is a common issue found in most authoring tools available all over the world. Furthermore, some authoring tools are powerful but not standard compliant. To overcome these drawbacks, we propose a courseware design tool with the theory of concept and influence diagram coupled with a user-friendly interface. The transformation algorithm is also included for the conformance with e-learning standards. With the proposed mechanism and tools, the advantages of courseware diagram are preserved. Students' learning performance can be improved by taking different levels of remedial courses based on student performance with a systematically built course flow chart. Furthermore, students' learning results can be maximized by analyzing their learning performance for course content adjustment. More importantly, SCORM compliant courseware can be generated from the courseware diagram directly using the proposed algorithms.

Keywords: SCORM, simple sequencing, content packaging, courseware diagram, conceptual map, influence diagram

1. INTRODUCTION

Internet usage is getting more and more convenient and reliable. High technology and basic infrastructure have promoted Internet usage for many activities. Distance learning or distance education over the Internet provides students with a flexible, convenient and stable environment. Different kinds of course content such as animation, video, audio, and many image file formats are supported in distance learning. While people have accepted the flexible learning environment provided by the Internet, researches of web-based distance education face numerous challenges. The challenges fall into two categories, synchronous education, which means that teaching and learning happen at the same time, and asynchronous education, which provides prepared learning content to users. We focus on asynchronous education. In this paper, we argue that the biggest challenge of distance learning is the lack of a powerful and scalable evaluation mechanism to support interaction among students, instructors, and school. The evaluation mechanism can provide feedback for the three parties. Thus, our study falls into the area of measurement of students' learning performance.

In traditional education the teacher can modify his/her lecturing style or content to maximize the teaching quality with students' responses. However, it is hard for teachers to modify the learning content or style immediately in the distance learning. Teaching materials need to be perfectly organized and fully understood by students before being published on the Internet. Hence an instructor can predict his/her students' learning ability thoroughly before the class begins. In addition to predicting students' learning ability, the instructor can use the students' learning records to make an accurate analysis enabling. Teachers to modify the learning content or style more appropriately and faster.

Nowadays there are many learning management systems (LMS) which provide learning content, member login, a discussion board and content viewer. Each LMS has its own learning content format, so that it is hard for different LMSs to exchange their content with each other. Fig. 1 shows how different e-learning content can be pluged into different learning management systems. With the help of content description, learning content can be packaged as SCOs as defined in SCORM. Similarly, LMSs adopting the SCORM standard can provide SCO APIs. Through SCORM common interface, content developed by different tools can be plugged into different LMSs.
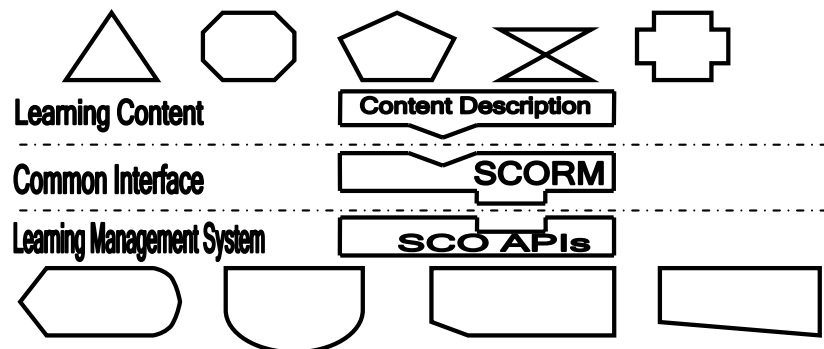

Fig. 1. SCORM servers as a common interface between e-learning content and LMSs.

Some committees proposed infrastructures or specifications of standards for asynchronous distance learning, such as the Learning Technology System Architecture (LTSA), the Sharable Content Object Reference Model (SCORM), and the IMS Simple Sequence Specification. It is very important to have standardized representation of learning content in order to promote the development of e-learning in industry, academia and government. It is a trend to standardize the learning content.

Due to the lack of an evaluation mechanism and distance learning standard, our system is designed using a systematic design, situation occurring in a real classroom, constructed a strategic, organizational, and reasonable system for building up course content in distance education.

We propose an object-oriented tool for designing course content and a method for measuring students' learning performance in our system. Teachers can use our system to make suitable learning content with drag-and-drop operation and linking to related learning materials. After the course content has been created, it is transformed into a

SCORM compliant courseware. The teacher can easily update and redesign course content after receiving response from students by using the course development tool. The instructor can modify minor portions of the course content after obtaining students' learning performance in each evaluation, or reorganize the entire course content after the class is over. With regard to the evaluation of students' learning performance, individual performance can be fairly measured by comparing one performance with the entire class. The course is compatible with SCORM, which is an aggregated specification for asynchronous distance learning called Sharable Content Object Reference Model proposed by Advanced Distributed Learning (ADL) (http://www.adlnet.org/). SCORM has characteristics of accessibility, interoperability, durability, reusability and cost effectiveness. Our target is to provide a system which is able to create courseware according to courserware diagram theory through friendly user interfaces. Futhermore, the produced courseware is compatible to the international e-learning standard SCORM.

Now we briefly introduce the SCORM standard. There are three technical issues:

- The content aggregation model (the format of courseware).
- The run-time environment (the protocol of running courseware).
- The simple sequence specification (with learning status tracking, sequencing rules, and application program interfaces).

In order to make courseware reusable, a standard representation of content and structures must be enforced. The Content Aggregation Model (CAM) serves this purpose. CAM can be discussed in three parts: the Content Model, the Metadata, and the Content Packaging. The Content Model defines the content components of a learning experience. That is the way learning materials are organized into different levels of small potions. The Metadata definition is a set of standard items, which is used to describe the Content Model. Metadata provides an efficient and effective mechanism for content search. The Content Packaging is a standard definition to allow the content model and the structure to be packed into a standard exchangeable file, known as the Package Interchange File (PIF). The purpose of the SCORM run-time environment is to establish a standard protocol for the courseware to communicate with its underlying Learning Management System (LMS) that is both machine independent and operating system independent. Sequencing defined the learning sequencing in SCORM. With the structure of learning content, sequencing strategy and sequencing definition model, learner behavior will be controlled and the system events will be triggered.

Section 1 gives a brief introduction and motivation. Section 2 gives a survey for e-learning learning management systems, distance learning technologies, learning styles and assessment methods. We also introduce the idea of the concept and influence diagram. Section 3 shows the main idea for converting the course into SCORM based content package. Section 4 illustrates the courseware system which we have developed. This section demonstrates the system interfaces, a course example and the authoring procedures. Some famous content authoring tools introduced in section 5. The comparison between our system and others are also discussed. Finally, in the last section we give the conclusion of our work.

2. RELATED WORK

Studies in the field of distance education described below have been done in recent years. These researches can be categorized as multimedia distance learning, distance learning specifications and standards, group learning, CSCW system, web technology used in distance learning, assessment, content development tools, learning management systems (LMS), and learning content management systems (LCMS). We have surveyed some related researches, and their summaries are listed as below.

- Some issues about sharing and reusing course specifications included SCORM (Sharable Content Object Reference Model) [13], IMS [15], IEEE LOM (Learning Object Metadata) [20, 22], ADL (Advanced Distributed Learning) [14].
- Group discussion tools have different combinations of communication facilities. Virtual University (i.e., VLE [5]) mentioned that the students' classrooms are dynamically located. Other researches about virtual university administration and operation issues can be found in [9, 18, 24].
- WebCT [http://www.webct.com/] was initially developed at UBC, Canada. The system offers a lot of tools and functions for learning management, such as administration, calendar, email, student management, self test, and student presentations. Blackboard is another e-learning software platform [http://www.blackboard.com/]. The system provides a secure authentication function. Statistics on learning status is shown to the instructor as well as to students.
- Chang's study of web based assessment [4] is one example of research studies which fall into the field of assessment for distance learning. In brief, some researchers concentrate on developing communication tools and group cooperation while others concentrate on analyzing and evaluating student's learning performance.
- For vast areas, a satellite based distance-learning environment is very useful. [11] discuss the educational benefits and techniques for satellite based e-learning environments.
- Ideas for developing an adequate platform to facilitate flexible distance learning can be found in researches such as Intelligent Tutoring System (ITS) [6, 10], [http://www.shai.com/training/p.htm], [http://advlearn.Irdc.pitt.edu/its-arch/papers/], Learning Space [http://www.lotus.com/home.nsf/welcome/learnspace], and SEND [7].
- SMIL (Synchronized Multimedia Integration Language) and XML (Extensible Markup Language) are being used by some researchers who concentrate on an adequate programming language to use on Web Browsers.
- [1, 2] used WWW techniques and a Java applet for distance learning.
- [8, 12] proposed interactive classroom and CSCW systems in distance learning. Laboratory-based distance learning systems are discussed in [16, 22].
- Electronic Software is available for distance learning, such as LCMS (Learning Content Management System), and MITS (Multimedia Interactive Telelearning System) [19].

2.1 Concept Map and Influence Diagram

The concept mapping technique was developed originally by Prof. Joseph D. Novak

at Cornell University in the 1960's. Concept mapping was first developed as an educational tool for representing general knowledge for organizing and communicating information. Concept mapping is based on the premise that all knowledge can be represented by relationships between more fundamental concepts. Concept mapping is a two dimensional graph which constructs the records of information in a hierarchy format starting with the most general concepts and proceeding downward in increasingly greater detail. The advantage with using a concept map is that responses from students are obtainable. The instructor can receive responses from students and take the information into consideration for adjusting teaching materials or style in order to maximize student learning performance.

The other related research is the Influence Diagram. The Influence Diagram [21] was developed for representing decision problems. It grows linearly (as opposed to growing exponentially for decision trees) so that larger decision problems can be represented. It is a singly connected DAG (Directed Acyclic Graph) without loop. There are two kinds of nodes and links defined in the influence diagram, and they are decision nodes, chance nodes, informational links, and conditioning links. Their descriptions are as follows.

- A Decision node is represented by a rectangle or square that represents a variable under the decision maker's control.
- A Chance node is represented by an oval or a circle that denotes a probabilistic variable.
- A Conditioning link always points toward a chance node and represents a probabilistic dependence.
- 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 with all outcomes of its direct predecessors. Thus, informational links imply a chronological order but conditioning links do not. Another important issue is the direction of conditioning links. In general, representation of a decision problem is not unique. It is not completely suitable for distance learning courseware design; therefore it should be modified in our system. Not only referring to concept map and the influence diagram, but we also refer to the theory of Learning Cycle proposed by Kolb, and attempted to apply the process of experiencing, reflecting, conceptualization, and planning mentioned in the theory in our courseware diagram mechanism [17]. The most important part in our objectives for designing the courseware diagram is to increase learners' achievements through making adequate adjustment of course content based on progressive feedback which is collected during the evaluation.

2.2 Using Course Diagram to Represent Course Design Problem

Chang [3] proposed the courseware diagram which is a combined theory of concept and influence diagram. This mechanism can be easily used by instructor and student because of its user-friendly interface. There are some basic terms applied in the courseware diagram. They are as follows:

Course Knowledge Weight (CKW)

A Course Knowledge Weight (CKW) reflects the importance of knowledge which is a value associated with every course unit. The instructor sets the value of CKW; and the sum CKW of all course units is one.

Acquaintance Degree (AD)

Acquaintance Degree is deduced by taking an evaluation in order to measure the learning performance. In "AD [0.4, 0.6]", for example, 0.4 shows the learning performance of the students who receive relatively lower scores on an evaluation up to a certain course unit. The number 0.6 shows the learning performance of the students who receive relatively higher score.

Group Percentage (GP)

A Group Percentage (GP) is a pair of values associated with an evaluation unit. GP indicates the percentage distribution among students who receive lower, average, and higher score in an evaluation. "GP [25, 25]" means 25 percent of students who receive relatively lower and relatively higher scores and the rest of the students.

Discount Rate (DR)

A Discount Rate (DR) is a value associated with every discount bar placed on conditional links. An example of Discount Rate is "DR = 0.8". The student's score on an evaluation reflects the degree of his/her perception on previously taught knowledge.

Individual Performance (IP)

An Individual Performance (IP) indicates an individual's actual performance after taking a series of courses and evaluations.

The most important factor in the courseware diagram is the value of Acquaintance Degree (AD) which indicates the expected best and worst learning performance of the entire class along the path of the courseware diagram. Individual Performance (IP) indicates an individual student's learning performance along the path of the courseware diagram. A combination of IP and AD shows one student's performance compared with the entire class.

There are some advantages to the inference diagram. The method allows the instructor to receive feedback from learners. The courseware diagram has three aspects. First, the course diagram system can be used to build the flow chart systematically. Secondly, the students' learning performances can be improved by taking varied levels of remedial courses. Thirdly, by analyzing students' learning performances, course content can be adjusted to maximize their learning outcome.

3. IMPLEMENTING COURSEWARE IN SCORM

Although courseware diagram system can be used to build course charts systematically, improve students learning performance by taking different levels of remedial courses based on student performance, and analyze student's learning performances to

adjust course content to maximize students' learning result; it is not a standard courseware development tool. So, it is not compatible with the current wildly accepted e-learning standard known as SCORM. As a result, the courseware designed by the courseware development tool can not be plugged into SCORM based LMS for delivering. This is a major obstacle to the development of authoring tools with powerful functions such as courseware diagram system. For this demand, we propose an algorithm and data model mapping for transforming courseware designed by the courseware diagram system to be SCORM compliant.

3.1 Implementing Content Aggregation

In order to accomplish courseware mapping, we need to consider the data types provided by courseware diagram. In addition, we need to know how those data types can be mapped to the SCORM data model. Courseware diagram system provides three types of nodes, course node, exam node, and final node. Each is represented as a graphical symbol: circle means course node; quadrangle means exam node, and octagon means final node. A course node could be a general course or remedial course. CKW (Course Knowledge Weight) and AD (Acquaintance Degree) are based on these course nodes. The exam node is the evaluation learning content, GP (Group Percentage) is the attribute of the exam node, and the final node is used to analyze the learners' performance for the whole course.

According to the SCORM specification, an Asset is the basic element within the content aggregation. Assets can be text, images, sounds, assessment objects or any other form of media that can be rendered by a web client for learners. An SCO is a collection of one or more Assets that represents a single learning resource with sequencing definition for controlling the navigation of assets within the SCO. SCO can be tracked by learning management systems that support SCORM sequencing and navigation. A content organization is an aggregation that includes all the course content of a single course. This content aggregation is described by an XML Metadata file "imsmanifest.xml" which describes the content aggregation, the course structure, and the resources within the course.

To map nodes in the courseware diagram system, we need to map course nodes, evaluation nodes and the final node to a suitable data model to form a course aggregation with remedial course functionality. General mapping rules for this purpose are described below:

1. The final node becomes the last traversed module and reflects a student's learning behavior by rolling back the navigation behaviors collected during this student's navigation of the course activity. The result determines if the learner can pass.
2. Since course nodes and evaluation nodes are multimedia resources, they are transformed as SCOs or Assets.

With the general mapping rules, the nodes within the courseware diagram can be easily transformed to SCORM compliant course elements.

3.2 Implementing Simple Sequence Specification

The Sequencing and Navigation Specification (SNS) was first introduced in the SCORM 1.3 Sequencing and Navigation (SN) book. It was originally from the IMS simple sequencing specification. The SNS defines the relative order of the components within a courseware by the sequencing definition model. The navigation model of SNS specifies the run-time data model that enables content objects to query the LMS for sequencing state and to indicate the LMS desired navigation requests. The detailed information about students' navigation behaviors is described by the sequencing behaviors of SN book. It is typical to find a set of sequencing rules implemented with respect to a set of course objects which forms a course SCO. While an individual student is navigating through a courseware, his/her navigation behavior and learning status are recorded and stored in a database of the LMS server. A portion of the run-time environment which is responsible for firing sequence rules and bookkeeping learning status is called the Sequence Engine.

For course content to be transformed from courseware diagram to a tree like structure of a SCORM courseware there is a graphics problem. The courseware diagram graph is a directed tree but the format of SCORM course structure, without the sequencing definition model, is a rooted tree structure. A directed tree differs from a rooted tree in structure and arrow edge. As a result, when we transform a courseware diagram, we have to analyze the directed graph within the courseware diagram and transform it to the corresponding sequencing definition.

Fig. 2 shows an example of transforming a courseware diagram into a SCORM course structure with sequencing definition model from level 0 through level 3. In level 0, there are three different representation symbols. The circle indicates the course node, the rectangle indicates the exam node, and the hexagon indicates the final node. This is the general graph which the courseware diagram made. This graphical example illustrates a typical online course which includes a pre-test exam, a midterm exam, and a final exam. As shown in the diagram, there are some possible learning sequences. Each sequence results from a learning behavior and the settings of the courseware diagram. For example, a student may navigate the course with path T1: (PE → L1 → L2 → L3 → L4 → M-TE → L7 → L8 → L9 → FE → END) while another student traverses path T2: (PE → L5 → L3 → L4 → M-TE → L7 → L8 → L10 → FE → END). These two different paths reveal a possible situation which shows a student path T2 might do better on the PE (Pre-test exam) than a student using path T1. Since the total nodes of T2 are less than the total nodes of T1, using path T2 are allowed to skip a course node based on their pre-test exam results. To allow different learning behaviors, according to students' actual learning situation, can be defined in a courseware diagram. There are four other paths that can be found in the courseware diagram shown in level 0. Since a SCORM course structure differs from a courseware diagram, we need to separate the "meaning" of directed graph from the courseware diagram by removing the arrow on the direct edge and transfer the meaning of the directed graph to the sequencing definition.

In order to retrieve the sequencing definition, we analyze the courseware diagram by segmenting the courseware diagram based on each exam within the diagram. As a result, we may get three major activities as shown in level 1. Within each major activity, there are some sub activities. After analyzing each major activity, another two levels of
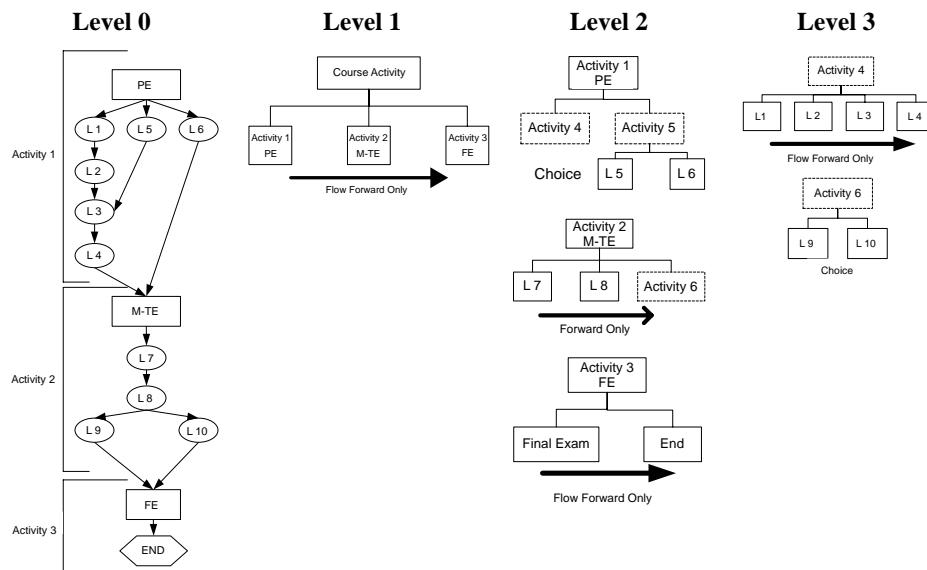
Fig. 2. An example of transforming a courseware diagram into a SCORM course structure.

sub activities can be retrieved which are shown as level 2 and level 3. The meaning of the directed graph can be preserved as a sequencing definition in each generated activity node. These activity nodes become modules with sequencing rules defined as required by SCORM SNS.
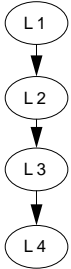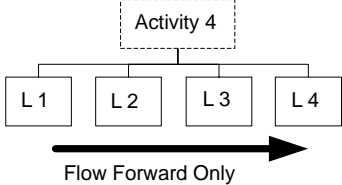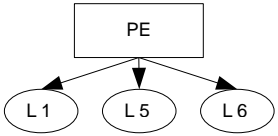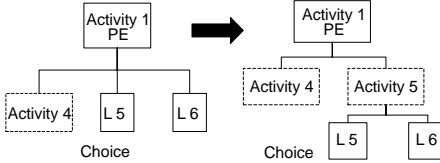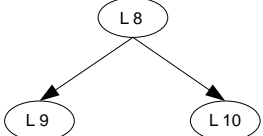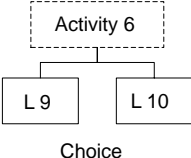
Fig. 2 presents a real example for transforming a directed graph to a tree structure graph. We propose an algorithm to transform the graphs to a SCORM-compliant course structure. There are three major structures in a directed graph of a courseware diagram. These three structures need to be transformed to SCORM sequencing rules. These major structures are as follows:

- [Conditional Choice]: It is similar to [Choice] course structure. However the sequence is decided by predefined conditions. The predefined conditions can be set by instructors in the courseware diagram.
- [Linear]: In the linear structure, course content is retrieved in a fixed sequence.
- [Choice]: When the course sequence has more than one choice, we define this course structure as [Choice].

The transformation for three major structures of courseware diagram is shown in Table 1. The details of our transform algorithm are as follows:

- Transformation of Linear:
  If [L1 → L2 → L3 → L4] then
    Generate a new activity module, ex: "Activity 4", as L1.parent L2.parent L3.parent, and L4.parent
    Set Control mode of "Activity 4" = Flow Forward Only

Table 1. Three structures and the corresponding SCORM based sequencing structures.

Type	Graph	SCORM Activity Tree
Linear		 Flow Forward Only
Conditional Choice		
Choice		 Choice

- Transformation of Conditional Choice:
  If an exam node with more than two child nodes ex. [PE → L1 → L5 → L6] Then
      Process binary tree conversion.
      New nodes generated in the resulting binary tree are new activity modules.
      ex. [Activity 5 → L5, Activity 5 → L6], [Activity 1 → Activity 4, Activity 1 → Activity 5]
  Set Objective Measure Greater then
        Hide from choice

- Transformation of Choice:
  If [L8 → L9] and [L8 → L10] then
      Generate a new activity module, ex: "Activity 6", as L9.parent L10.parent
      Set Control mode of Activity 6 = Choice

3.3 Implementing Data Model

    Due to different structures and specifications, the attributes in courseware diagram system are not supported by the SCORM Run-Time Environment. In order to allow for courseware designed by courseware diagram system can be supported by SCORM compliant Run-Time Environment; the attributes of the courseware diagram need to be

mapped to the SCORM Data Model. Furthermore, the directed graph of the courseware diagram needs to be preserved by sequencing rules transformed from the courseware diagram. With the transformed standard data model and sequencing rule, the original courseware designed by the courseware diagram system is able to be bound with the SCORM standard underlying protocol. Therefore, the courseware can be supported by the SCORM compliant LMS. It can also be invoked or launched by the SCORM API instance.
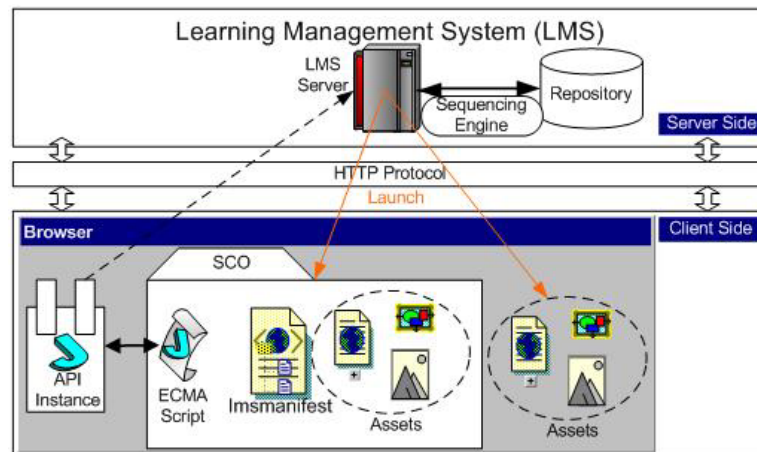

Fig. 3. An illustration of the SCORM run-time environment (revised from http://www.adlnet.org/).

Fig. 3 is a simple illustration of the SCORM run-time environment. Typically, the Learning Management System is installed on a central server with a depository store, where all of the course materials and student profiles are stored. The client's application is normally a web browser. This architecture is also known as a web based architecture. After a student logs in to an LMS server, a series of SCORM API calls can be made through an ECMA script within a SCO. The API instance will connect to the server and launch a SCO or an Asset according to the student's sequencing behavior and navigation model defined in SCORM. The student's sequencing behavior will be updated at each navigation request he/she makes. The standard API functions and their descriptions are listed as follows:

A. Initialize(): The function is used to initiate a communication session. It allows the LMS to handle specific initialization issues of a LMS.
B. Terminate(): The function is used to terminate a communication session. It is called when the communication between a SCO and the LMS is no longer needed.
C. GetValue(): The function requests information from an LMS. It permits the SCO to request information from the LMS.
D. SetValue(): The method is used for requesting the transmission to the LMS of the value of some parameters. It allows the learning content to send information to LMS for storage. It also assists API to store data in some parameters.

E. GetLast Error(): If a learning content calls the method, the API Instance will not alter the current error state, but simply return the requested information.

F. GetErrorString(): This function can be used to return a textual description of the current error state.

G. GetDiagnostic(): The method provides LMS the ability to define additional descriptions or additional diagnostic information through API Instance.

4. INTERFACE DESIGN AND CASE STUDY

SCORM Courseware Editor Version 3.0 is implemented by Java which allows teachers to draw the courseware graph. Button (▶) is used to move the node position. Button (◎) is the course node which includes the general course and the remedial course. Evaluation node is the button (▫). The instructor can choose the test and exam content for the evaluation node. Button (●) is the final node for calculating the final value of the evaluation progress. Button (🖥) is the content package function which transforms the courseware diagram to a SCORM compatible package. Button (✄) is used to clip the selected item. Button (📋) is used to copy the selected item. Button (📋) is used to paste the selected item. Button (🗑) is used to deleted the selected item. Buttons (↩) and (↪) represent recuperate and contrary. Button (💡) is used to calculate the learning performance. Button (🔧) is used to fix the drawing arrow function. Button row (⤢ ⬚ ☒) can minimize, maximize and cancel the window. The tool supports multiple working windows; users can use multiple courseware diagrams in different windows. Fig. 6 shows the individual course node attribute table which provides setting the learning resource and course knowledge weight value. We can set the attribute to general or remedial course. Fig. 6 shows the evaluation course node attribute table. The following is the sequence of how the instructor uses the courseware diagram system

1. Create a new courseware diagram.
2. Add the general course or remedial course with the button (◎). Each course item has two attributes: [Preview source file] sets the real file route and [Property] allows the instructor to set the CWK (Course Knowledge Weight) value.
3. When there are enough course nodes, the instructor can link the course nodes relation and sequence.
4. If the instructor wants to add an evaluation course node (▫), he/she can set the attributes. The attributes are evaluation name, lower group percentage, high group percentage, lower discount rate, higher discount rate and the evaluation source file route.
5. When most of the course nodes are linked and the attributes completed, the instructor can set the final node (●).
6. If the instructor wants to get the evaluation value, he/she can use (💡) to calculate the learning performance.
7. If the instructor wants to package the course as a SCORM compatible course package, he/she can use (🖥) to generate a SCORM content package.

In the whole editing progress, the tool will calculate the related attributes to match the courseware diagram system.
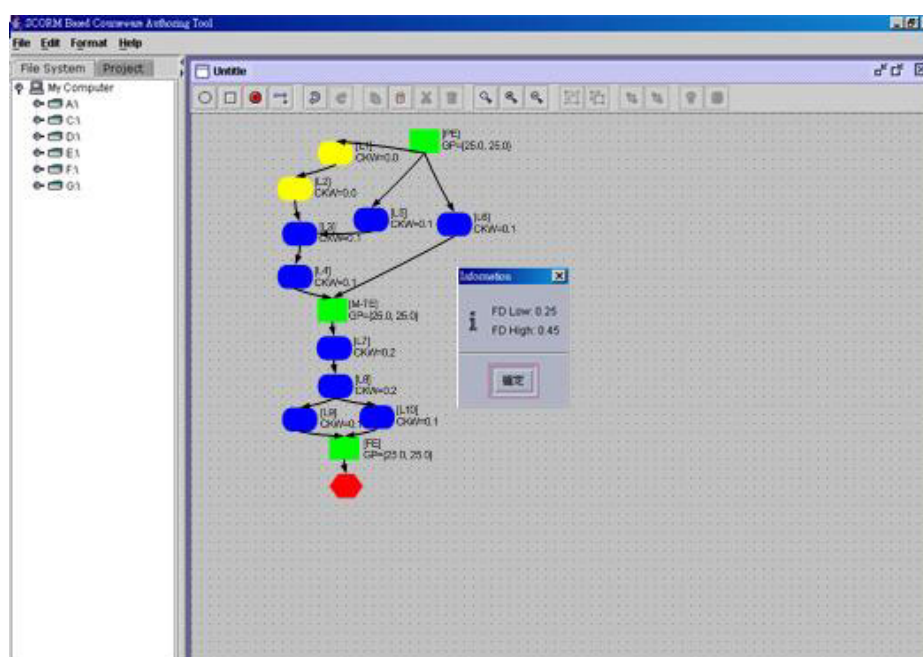
Fig. 4. The evaluation result of the whole course structure.
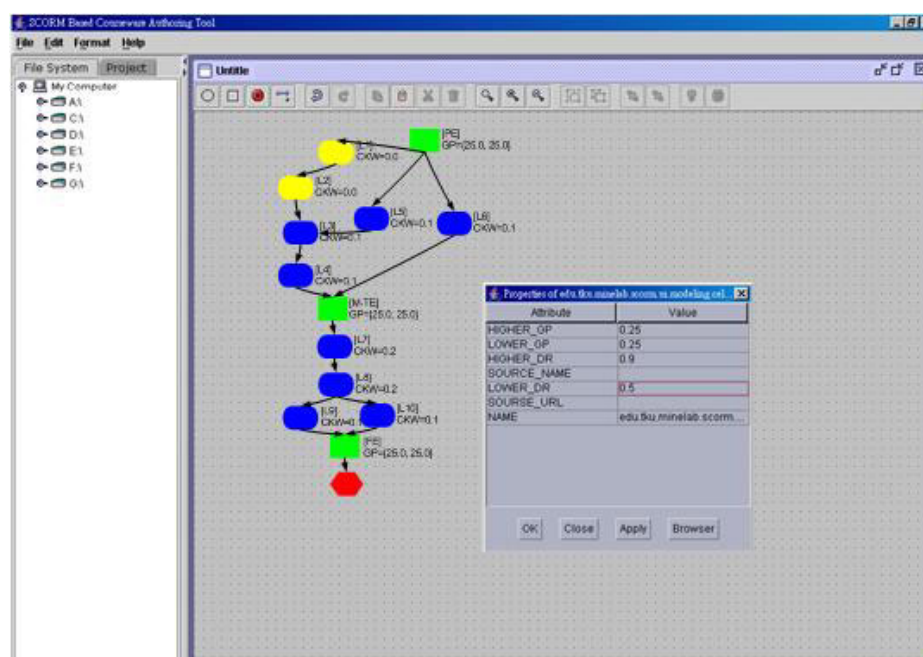

Fig. 5. Evaluation node attribute table. It involves GP (Group Percentage), DR (Discount Rate), course name, course source and the source link.
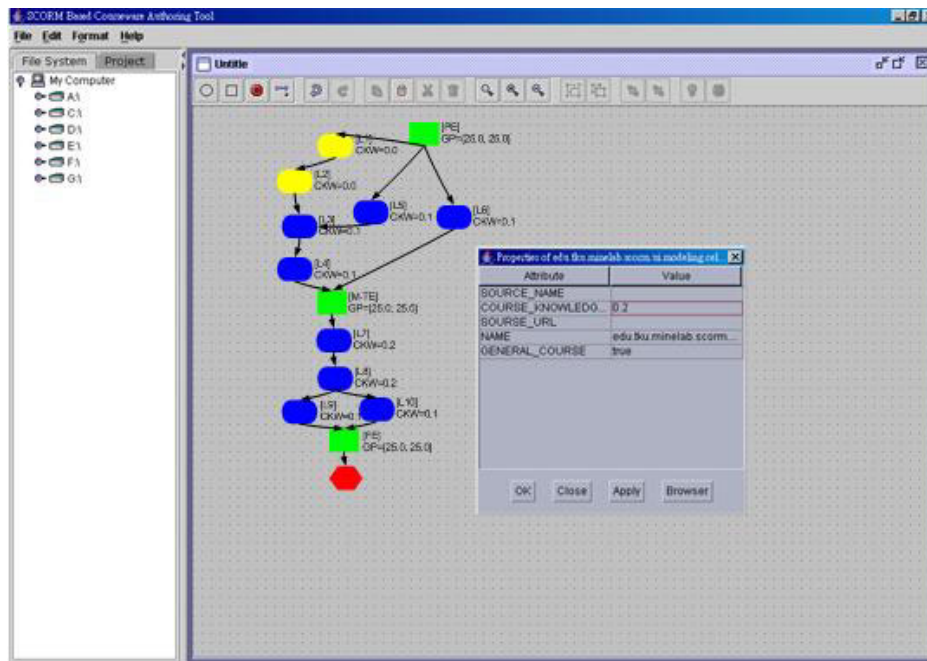
Fig. 6. Individual general course node attribute table. It includes course knowledge weight value, course name, course source and source link.
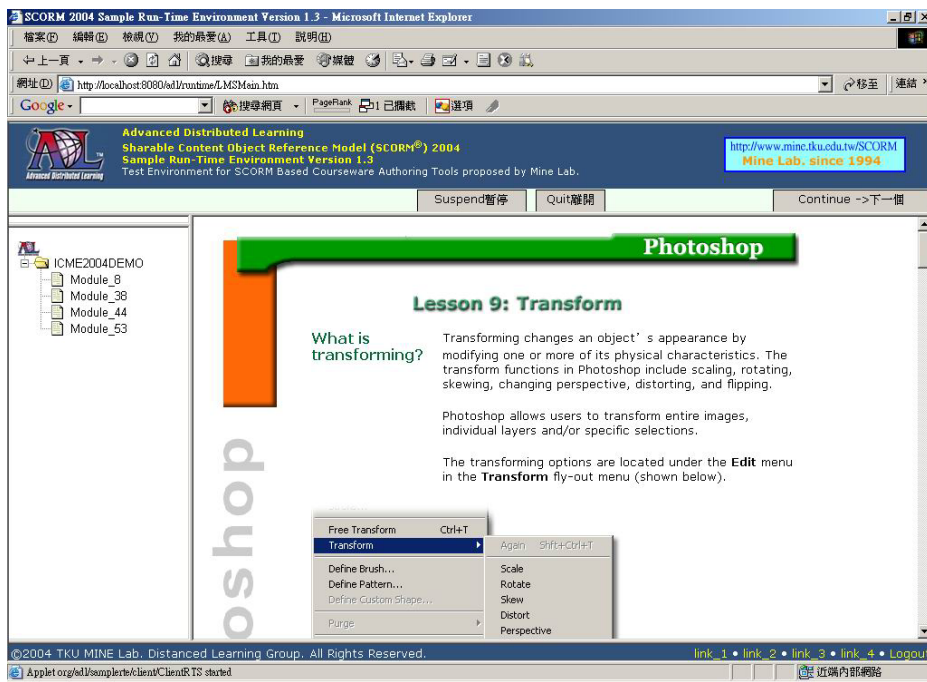

Fig. 7. Our courseware diagram course is successfully running in the SCORM environment.

SCORM Sample Run-time Environment (RTE) Version 2004 was provided by Advanced Distributed Learning. This Sample RTE using HTML, JavaScript, Java Server Pages (JSP), Java Applets and Java Serves is a Web-based client/server application. We put the generated package file in the test environment to prove our packaged file is SCORM compatible. Fig. 7 shows the successful result.

We illustrate a course example of Introduction to Computers as following. In Fig. 8, the first design of courseware diagram starts with a course unit (CKW = 1). A reduced courseware diagram is to simplify the courseware diagram design. Fig. 9 shows how to develop core courses drawn in bold lines. In the Introduction to Computers class, there are some related courses which are basic (CKW = 0.4), and advanced (CKW = 0.6); the learning sequence is depicted in Fig. 9. There are four lessons in the basic course section and four lessons in the advanced course. Fig. 10 shows an expanded version of Fig. 9. After expanding the structure for all courses, the next step is to decide the evaluation unit in the courseware structure. Fig. 11 shows the example which has the evaluation units. An evaluation node has to be set prior to the final value node to measure students' learning performances within the requirement of the courseware diagram. Therefore, evaluation 3 is placed ahead of the final value. This rule has to be followed otherwise the whole theory is corrupted. In the example course, the instructor adds two evaluation nodes (evaluation 1 and evaluation 2). The evaluation node position is set by the instructor. There are two remedial courses that are both weighted 0 for the reason that no new knowledge is provided in remedial courses. The function of the evaluation node is to reduce the value of AD and is in a fixed form shown on the courseware diagram. As shown in Fig. 11, evaluations 1 and 2, which are temporarily represented with an extra bold line, 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.
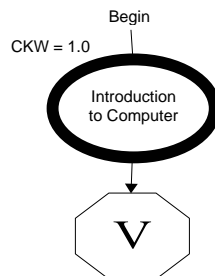
Fig. 8. A reduced courseware diagram for the class.
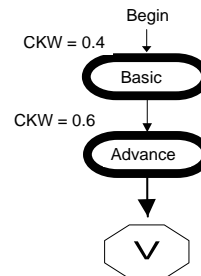
Fig. 9. An expanded structure of Fig. 8 with core courses.

5. DISCUSSIONS

There are many e-learning content and content packaging authoring systems. An authoring tool is an application which supports instructors and developers in authoring their on-line courseware. With these tools, they can edit and author the courses efficiently. However it is also very important to support e-learning specifications. We introduce and compare some authoring tools in this section.
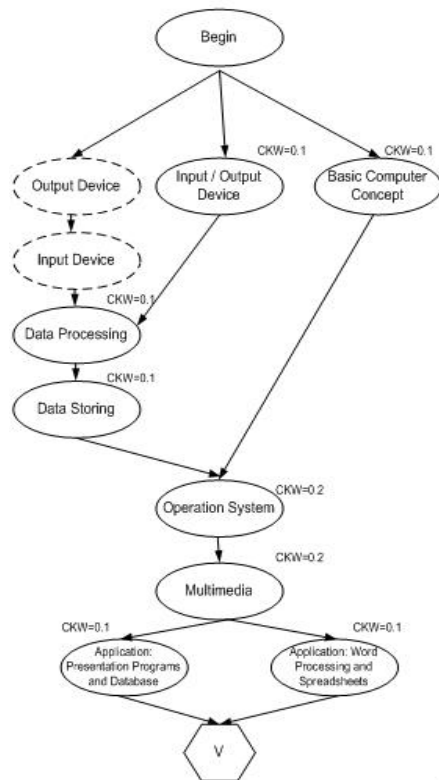
Fig. 10. An expanded structure of Fig. 9 with all course units.
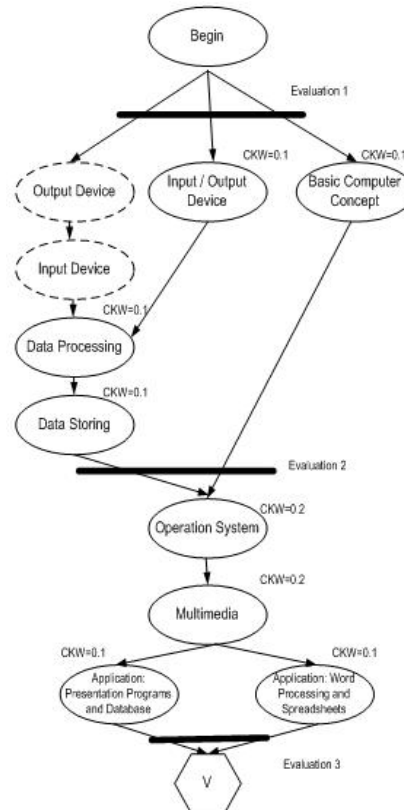
Fig. 11. An expanded structure of Fig. 10 with all course and evaluation units.

- **RELOAD**

    RELOAD provides a metadata editor, learning designer editor and SCORM player. It focuses on the learning object packaging and delivery. It also allows instructional designers to define sequencing rules.

- **FLXeLEARN**

    The FLXeLEARN comprises an editor, player, tracker and tutor. The editor provides standard authoring functions. The player, which is a functionally rich player, can play the content created by the editor. The tracker supports tracking and recording of a remote learner's progress.

- **Macromedia eLearning Suite 2004**

    Authorware 7 can help users producing rich-media courseware for e-learning. Authorware provides an XML-based metadata editor to edit the metadata information. SCORM defines nine categories of metadata to describe and define the learning content. Flash MX Professional 2004 can aid users in developing advanced Flash content, appli-

cations and video experiences for e-learning. Flash provides an exam template for users to set their values and attributes. The provided templates are styles of questions and exams. After the exam is finished, the learning content shows the final number and percentage. Dreamweaver MX 2004 can assist in build professional websites and applications for e-learning.

• **Microsoft LRN and Microsoft Office plug-in tool**

Learning Resource iNterchange (LRN) is based on the IMS Content Packaging and Metadata Specifications. The metadata specification in MS LRN is different from SCORM. Microsoft LRN Toolkit version 3.0 can support MS PowerPoint and Word file formats.

Table 2. Comparison table.

	SCORM Specification Version Compatible	Evaluation Feedback	Visualization Presentation	Learning Performance Simulation
LRN	v1.2	O	◎	O
Reload	2004	O	◎	O
Our system	2004	◎	◎	◎
Macromedia eLearning Suite 2004	2004	◆	◆	O
FLEXeLEARN	v1.2	◎	◎	O
click2learn - Tool-Book II Assistant 8 and Instructor 8.5	v1.2	O	◎	O

◎: Fully Support   ◆: Partial Support   O: None

We compare some characteristics in the following (Table 2).

• SCORM Specification Version Compatible: The generated course package generated by the authoring tool can import and run successfully in a SCORM compatible run-time environment.
• Evaluation Feedback: The tool provides some feedback to the system about the evaluation results. The results and course relations may influence the remaining course sequence.
• Visualization Presentation: It is important to provide a powerful, convenient and user friendly authoring tool for instructors to edit and create learning materials, and use graphics items to represent the course node and learning materials. The instructor can organize the course structure by dragging and dropping.
• Learning Performance Simulation: This feature supports simulation of the students' learning performance simulation for the course.

6. CONCLUSIONS

This paper proposes a system based on courseware diagram theory. It also provides an advanced transform algorithm to generate SCORM compatible course. The instructor can use simple drag and drop operations to build a courseware diagram. The generated course can be packaged as a SCORM compliant content package which can be run on the SCORM based learning management system. One powerful function built into this system is the transforming function. This function is able to convert the original directed graph to SCORM compatible course tree structure with sequencing rules. Our proposed system allows instructors to receive prompt feedback from students. The course diagram system can build the flow chart systematically. With the help of this system, students' learning performances can be improved by taking different levels of remedial courses based on their abilities. With analyzing and assessing student's learning performance, course content can be adjusted to maximize students' learning results. With SCORM compatibility, all course content achieve accessibility, interoperability, durability, reusability and cost effectiveness.

ACKNOWLEDGMENTS

We wish to thank Hsuan-Che Yang and Pei-Ying Wu for their implementation of our system. Without their work, our courseware diagram system could not have been completed.

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 IEEE Southeast-con '97*, 1997, pp. 202-206.
2. P. Cadolini, F. Davoli, G. Marescotti, and P. Maryni, "Developing a distance learning system using Java applets," in *Proceedings of the Global Telecommunications Conference*, 1996, pp. 63-66.
3. C. I. Chang, L. P. Hung, and T. K. Shih, "A new courseware diagram for quantitative measurement of distance learning courses," *Journal of Information Science and Engineering*, Vol. 19, 2003, pp. 989-1014.
4. W. C. Chang, H. H. Hsu, T. K. Shih, and C. C. Wang, "Enhancing SCORM metadata for assessment authoring in e-learning," *Journal of Computer Assisted Learning*, Vol. 20, pp. 305-316.
5. Y. Y. Cheng and J. Yen, "Virtual learning environment (VLE): a web-based collaborative learning system," in *Proceedings of the IEEE 31st Annual Hawaii International Conference on System Sciences*, Vol. 1, 1998, pp. 480-491.
6. 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.
7. 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.

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*, Vol. 1, 1997, pp. 398-401.

9. G. Davies and E. Stacey, "Virtual universities: are dual mode universities the solution," in P*roceedings of the Frontiers in Education Conference*, Vol. 2, 1998, pp. 705-710.

10. C. Hegarty "Statutor: intelligent tutoring system?" in *Proceedings of the BILETA '96 Conference*, *The Journal of Information*, *Law and Technology* (*JILT*), Issue 3, 1996, http://elj.warwick.ac.uk/JILT/BILETA/1996/default.htm.

11. S. M. Hower, "African distance learning via satellite," in *Proceedings of 10th International Conference on Digital Satellite Communications*, Vol. 2, 1994, pp. 532-538.

12. J. N. Hwang, S. Deshpande, and M. T. Sun, "A virtual classroom for real-time interactive distance learning," in *Proceedings of 1998 IEEE International Symposium on Circuits and Systems* (*ISCAS '98*), Vol. 3, 1998, pp. 611-614.

13. E. R. Jones, "Implications of SCORM™ and emerging e-learning standards on engineering education," in *Proceedings of 2002 ASEE Gulf-Southwest Annual Conference*, 2002, pp. 20-22.

14. M. Jovanovic and V. Milutinovic, "An overview of reflective memory systems," *Journal of IEEE Concurrency* [see also *IEEE Parallel & Distributed Technology*], Vol. 7, 1999, pp. 56-64.

15. K. Kaminski, "Transforming an entire institution for online delivery of degree programs: perspectives from experience," in *Proceedings of the Third Annual WebCT Conference*, 2001, pp. 480-491.

16. M. Khader and W. Barnes, "Laboratory based courses in distance learning settings," in *Proceedings of the Frontiers in Education Conference*, Vol. 1, 1996, pp. 55-58.

17. D. A. Kolb, *Experiential Learning: Experience as the Source of Learning and Development*, Prentice-Hall, New Jersey, 1984. http://www.css.edu/users/dswenson/web/PAGEMILL/Kolb.htm.

18. A. Kumar, R. Pakala, R. K. Ragade, and J. P. Wong, "The virtual learning environment system," in *Proceedings of the Frontiers in Education Conference*, Vol. 2, 1998, pp. 711-716.

19. Y. Lei, 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. 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.

21. R. D. Shachter, "Evaluating influence diagrams," *Operating Research*, Vol. 34, 1986, pp. 871-882.

22. 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.

23. 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*, Vol. 1, 1996, pp. 44-46.
24. M. Yazdani and D. Bligh, "Cooperative learning in a virtual university," in *Proceedings of 2nd International Conference on Cognitive Technology*, 1997, pp. 251-255.

Nigel H. Lin (林修名) is a Ph.D. candidate of Department of Computer Science and Information Engineering at Tamkang University, Taiwan, R.O.C. He received both of his M.S. and B.S. degrees of Computer Science and Information Engineering from Tamkang University in 2002 and 2000 respectively. His research is mostly related to distance education, mobile computing, and multimedia technology. He has published several international conference papers. He also joined several NSC (National Science Council of Taiwan) projects since he was an undergraduate student. In 2003, he joined an international cooperative project with NRC (National Research Council of Canada). Nigel has received several awards from Tamkang University. He was elected an Outstanding Young Student by Tamkang University in 2002.

Wen-Chih Chang (張文智) received his B.S. degree in 2001 and Ph.D. degree in 2005 from the Department of Computer Science and Information Engineering at Tamkang University, Taiwan, R.O.C. His research interests include mobile agent, distance learning, SCORM, petri net, and cooperative learning.

Timothy K. Shih (施國琛) is a Professor in the Department of Computer Science and Information Engineering at Tamkang University, Taiwan. He is a member of ACM. As a senior member of IEEE, Dr. Shih joined the Educational Activities Board of the Computer Society. His current research interests include multimedia computing and distance learning. He was a faculty member of the Computer Engineering Department at Tamkang University in 1986. In 1993 and 1994, he was a part time faculty member of the Computer Engineering Department at Santa Clara

University. He was also a visiting professor at the University of Aizu, Japan in summer 1999, a visiting researcher at the Academia Sinica, Taiwan in summer 2001, and an adjunct faculty member of Xidian University, China. Dr. Shih has edited many books and published over 340 papers and book chapters, as well as participating in many international academic activities, including the organization of more than 30 international conferences and several 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. He is also an associate editor of IEEE Transactions on Multimedia and ACM Transactions on Internet Technology. Dr. Shih has received many research awards, including research awards from National Science Council of Taiwan, IIAS research award of Germany, and several best paper awards from international conferences. He also received many funded research grants from both domestic and international agencies. Dr. Shih has been invited to give more than 10 keynote speeches and plenary talks in international conferences, tutorials in IEEE ICME 2001 and ACM Multimedia 2002, and talks at international conferences and overseas research organizations.

Huan-Chao Keh (葛煥昭) is a professor in Information Engineering and Computer Science at Tamkang University since 1991. His research interests include object-oriented methodology, parallel architectures, database systems, and multimedia computing. Dr. Keh received his B.S. in Engineering and M.S. in Computer Science from National Chung Hsing University, Taiwan and from Oregon State University, in 1980 and 1985, respectively. He also received his Ph.D. in Computer Science from Oregon State University in 1991.