

Ubiquitous e-Learning With Multimodal Multimedia Devices

Timothy K. Shih, *Senior Member, IEEE*, Te-Hua Wang, Chih-Yung Chang, Tai-Chien Kao, and Douglas Hamilton

Abstract—The Sharable Content Object Reference Model (SCORM) is a set of specifications and guidelines for the representation and operation of asynchronous distance learning. Since it was announced in late 1990s, the reference model has been used by software developers and academics in the development of authoring tools, learning management systems, and repositories for distance learning content. To date, most e-learning systems have been based on multimedia and Web technologies on personal computers. Our project, *Hard SCORM*, advances the field by implementing an integrated system which allows learners to read SCORM-compliant textbooks using multimodal multimedia devices. *Hard SCORM* employs a pen-like optical character reader device (called Hyper Pen) as an input mechanism. A computer, a personal digital assistant, or a cellular phone can be used for user behavior supervision using the *Hard SCORM Machine*. With an authoring tool, specially designed tags are printed in textbooks and recognized by Hyper Pen for user navigation control. In this way, users can read hardcopy textbooks in a traditional manner while the process of reading conforms to the SCORM specification. Part of the implemented system (Pocket SCORM on PDA) received the 2005 Brandon Hall Excellence in Learning Awards. The system has also been used by an airline company for online security checking and a high school for online mobile learning.

Index Terms—Augmented Paper, distance Learning, IEEE LOM, mobile devices, SCORM, web service.

I. INTRODUCTION

IN LINE with the blossoming of Web-based learning and real-time delivery of instruction, distance learning technologies bring to our education society new possibilities for future education. One interesting issue in distance learning is that interactions can be made between the instructor (could be a computer) and students, in many different locations. With advanced communication technology and intelligent agents (i.e., computer programs), students are able to learn in an interactive manner through self-regulation-based distance learning systems. Human-computer interaction then becomes a critical issue. Even so, it is also true that, despite their popularity, not everyone uses computers. People often still prefer reading on printed books [1]. How might it be possible to bridge

the gap between the physical world of printed pages and the virtual world of computers, by creating convenient devices and software for easy distance learning? Essentially, advanced hardware and communication technologies need to be used in conjunction with distance learning platforms. One answer was found in an alternative approach to using ordinary computers for multimedia interaction [2]. The Listen Reader system used radio frequency identification (RFID) tags embedded in an ordinary book. Electronic field sensors are then used to connect the book to a back-end device, for presenting sound tracks. The system was demonstrated in a museum. However, the system may not be sufficiently portable to cope with the needs of mobile learning. Another system uses a co-axial pen with three optional nibs [3]. The system also includes an authoring tool which allows users to create “active” pages. Again, with a back-end computing device, the system supports collaboration between two or three participants. Another system supporting digitally augmented paper via an authoring tool was presented in [4]. The system is able to provide hypermedia access between hard copy papers and multimedia devices. In addition, a multilayered linking technology is developed with digitally augmented papers [5]. Taken together, these systems reveal an important message: hard copy textbooks are required in learning, even though multimedia presentations may provide a higher degree of interaction. Following this trend, several forms of childrens’ story books (e.g., the *LittleTouch LeapPad*) were enabled with augmented sound tracks to attract young children.

Inspired by the touching paper technology, we aim to provide a solution for ubiquitous e-learning. One important issue is we want to deliver instructions on mobile devices that can be accessed from anywhere, any time. The device must be easy to carry. Fortunately, there are consumer products available based on optical character reader (OCR) technology. OCR can be implemented as a pen-like device which allows users to scan through textbooks. Off-the-shelf examples of such pen devices were integrated with an electronic dictionary. In our project, we call this type of pen device “Hyper Pens.” The name Hyper Pen comes from the fact that a hyper jump is performed when moving from using the device in the physical world for reading to a virtual world where an electronic device pronounces vocabulary audibly or even shows motion pictures. One contribution of this paper is to bridge the gap between the physical and the virtual worlds in a seamless manner by providing such a Hyper Pen in conjunction with a distance learning platform. Another contribution of this research project is we introduce an integrated system with multimodal learning devices based on the SCORM [6], with consistent learner records (LRs) maintained for instructional assessment. SCORM is based on HTML and Internet technologies. SCORM-related distance

Manuscript received October 14, 2005; revised July 20, 2006. This work was supported by the National Science Council of Taiwan under Grant 93-2524-S-032-002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jie Yang.

T. K. Shih, T. H. Wang, and C. Y. Chang are with Department of Computer Science and Information Engineering, Tamkang University, Taipei Hsien 25137, Taiwan, R.O.C. (e-mail: tshih@cs.tku.edu.tw; fatty@cs.tku.edu.tw; cychang@mail.tku.edu.tw).

T. C. Kao is with the Institute of Education, National Dong Hwa University, Hualien, Taiwan, R.O.C. (e-mail: mkao@mail.ndhu.edu.tw).

D. Hamilton is with the Academic ADL Co-Lab, University of Wisconsin-Madison, Madison, WI 53703-2793 USA (e-mail: doug@academiccolab.org).

Digital Object Identifier 10.1109/TMM.2006.886265



Fig. 1. Reading a SCORM-compliant hardcopy textbook.

learning systems are widely developed [7]–[12] to assist the e-learning community in standardizing the best approaches to creating, storing, transferring and deploying learning content. However, this is the first time that the integration of multimodal multimedia learning devices has been applied to a SCORM conformant learning environment, with consistent recording of learner events. To further the vision of combining pioneering learning methods and advanced computing technologies, we aim to construct a ubiquitous e-learning environment. We call this new combination of a Hyper Pen-based learning mechanism with multimodal multimedia devices the *Hard SCORM*¹ project. Fig. 1 shows a section of Hyper Pen navigation on a textbook. The student is using a Hyper Pen to read a textbook produced by our authoring tool. A personal digital assistant (PDA) is used for multimedia presentations. The backend PC has audio messages to guide the student. One important contribution is that we allow different devices to keep track of “user reading behavior.” The information recorded includes reading progress, frequency and duration of accessing particular content, and the results of online tests. Reading behavior is guided by our system according to the sequencing definitions (part of SCORM) provided by the instructor. Thus, our approach provides “interaction controls” in addition to using augmented papers to link tags to multimedia records.

Our research proposes a formal definition of functionalities and system responses. The definition uses a set of icons (or tags) with the Hyper Pen. The activation of these tags will change SCORM variables while the Hard SCORM Learning Management System (LMS) is running.

We discuss the SCORM specification in the next section, since terminologies used in this article follow that specification. We then propose our Hard SCORM architecture starting from a brief introduction to the authoring tool and the printed books. The Hard SCORM Tags printed for navigation are then summarized before our definition of a Hard SCORM Machine (HSM). Implementation and technical considerations in Hard SCORM LMS for supporting multimodal learning devices are then addressed. The system was the subject of a field study at a high school and an airline company; this paper includes an analysis and discussion of the results.

II. THE SCORM SPECIFICATION

The Advanced Distributed Learning (ADL) initiative [13] developed SCORM by combining and harmonizing well-accepted specifications from the IMS Global Learning Consortium, Inc. [14], the Aviation Industry Computer-Based Training (CBT) Committee (AICC) [15], the Alliance of Remote Instructional Authoring & Distribution Networks for Europe (ARIADNE) [16], and the Institute of Electrical and Electronics Engineers (IEEE) Learning Technology Standards Committee (LTSC) [17]. The SCORM specification is organized into three parts: the Content Aggregation Model (the format of courseware), the Run-Time Environment (the protocol of running courseware), and the Sequencing and Navigation (with learning status tracking, sequencing rules, and the application program interfaces).

In order to make courseware reusable, a standard representation of contents and structures must be enforced. The Content Aggregation Model (CAM) serves this purpose. CAM can be discussed in three parts: the Content Model, Metadata, and the Content Packaging. The Content Model defines a hierarchy of content components of a learning experience. There are several levels of components in the Content Model according to the SCORM specification.

- **Assets:** basic forms, such as text, images, sound, web pages, assessment objects, or other pieces of data that can be delivered to a Web client.
- **Sharable Content Objects (SCOs):** a collection of one or more Assets. A SCO represents the lowest level of granularity of learning resources that can be tracked by an LMS using the SCORM Run-Time Environment Data Model.
- **Learning Activities (Activities):** an instructional event or events embedded in a content resource or as an aggregation of activities that eventually resolve to discrete content resources with their contained instructional events.
- **Content Organization:** a map (content structure) that can be used to aggregate learning resources into a cohesive unit of instruction (e.g., course, chapter, module, etc.).

An asset does not maintain communication with the SCORM Run-Time Environment. However, with additional control programs, such as JavaScript, the course material can be packed into an SCO, which follows a communication protocol and interacts with the SCORM Run-Time Environment. A learning activity is a collection of instructional events, with or without references to SCOs. For instance, the instruction can give a pretest, a lecture, and a final test in an activity for a particular section of class. Content Organization is a mechanism which gathers different portions of course materials so they can be packed for delivery.

Metadata definition in SCORM contains a set of standard items, which are derived from the IEEE Learning Object Metadata (LOM) specification [18] for describing the components in the Content Model. Metadata provides an efficient and effective mechanism for content search.

Content Packaging is a standard definition allowing the content model and structure to be packed into a standard exchangeable file, known as the Package Interchange File (PIF). PIF files allow course contents to be exchanged in a standard form across

¹Hard SCORM is a project supported by the National Science Council of Taiwan under Grant 93-2524-S-032-002.

different platforms. A content package contains a specific XML document known as the manifest file (i.e., `imsmanifest.xml`) which describes the organization of a course, and points to a set of physical files for the actual contents. Usually, each physical content file is associated with an XML file which contains the metadata. A content package also includes several control programs to maintain the communication between the course object (e.g., SCO) and the SCORM Run-Time Environment.

The CAM provides a standard representation format for courseware content exchange. From the perspective of a software system, courseware should be available on different computers and software platforms. The purpose of the SCORM Run-Time environment is to establish a standard protocol for the courseware to talk to its underlying LMS. The SCORM Run-Time Environment specification includes the procedures and responsibilities for courseware communications with an LMS, a set of standard application program interfaces (APIs) to serve the communication, and a data model which describes the messages passed between the courseware and the LMS.

The Sequencing and Navigation (S&N) specification is another key feature of SCORM, providing support for alternatives in courseware navigation and the ability of courseware to adapt to individual students. S&N allows courseware components to be specified in a relative order, in which conditions and selections are precisely defined. A set of sequencing rules are implemented with respect to a set of course objects. While an individual student is navigating through courseware, their navigation behavior and learning status are recorded and stored by the run-time environment as an activity tree. A portion of the run-time environment, called the Sequencing Engine, is responsible for firing sequence rules and bookkeeping to keep track of learning status (i.e., activity tree). As soon as the sequencing engine receives a request from the courseware, the sequencing rules are fired at each step according to the behavior definitions. The outcomes of the sequencing process may update the status model of each individual student. LMS implementations should provide a mechanism to keep the activity tree of each user.

The short discussion above explains a few terms used in this paper. Our motivation is to design and implement a system which is SCORM-compliant, with additional technologies developed to enable the use of ubiquitous devices. Our project is called “*Hard SCORM*,” which we will discuss in Section III.

III. THE HARD SCORM PROJECT

Can distance learning only use technologies such as HTML and websites? In many cases, these technologies on their own are more than adequate. However, if we consider distance learning solutions from a systematic and ubiquitous perspective, HTML and the Web only provide a partial solution. The Hard SCORM project aims to deliver an integrated system for SCORM-based e-learning on multimodal multimedia devices. The project includes an authoring tool for users to aggregate learning objects and to package courseware for delivery. The courseware can be accessed by different devices including cellular phones, PDAs, PCs, and Hyper Pen, which recognizes a set of Hard SCORM Tags. We discuss the definition of these tags and the design of the authoring tool in the next section.

The mechanism of running the Hard SCORM LMS is then discussed.

A. The Hard Scorm Tags and Authoring Tool

In order to allow Hyper Pen and the LMS to communicate, a special communication mechanism is required. Firstly, we need to define a set of tags, which can be recognized by Hyper Pen. During a reading session these tags are scanned and recognized by the Hyper Pen. The definition of Hard SCORM Tags considers effective interaction and consistency with SCORM’s sequencing and navigation specification. Through the authoring tool, these tags are printed on hardcopy books automatically. Hard SCORM Tags are divided into four categories.

- **Navigation Tags:** User navigation is controlled by using Navigation Tags. These tags are generated automatically by the authoring tool according to the definition of sequencing specification given by the instructor.
 -  — **p Page Tag** is associated with a page number, *p*, which indicates current navigation focus. Activation of the tag changes the status of an activity tree.
 -  — **Next Page Tag** allows a navigation to move forward to the next page and change the status of an activity tree.
 -  — **Previous Page Tag** is similar to the Next Page Tag.
 -  — **p-Page Index Tag** allows users to jump to a particular page.
- **Reference Tags:** Multimedia resources can be used as references, which are triggered by Reference Tags.
 -  **id** — **Video Reference Tag** shows a video.
 -  **id** — **Audio Reference Tag** plays an audio clip.
 -  **id** — **URL Reference Tag** launches a website.
 -  **id** — **Flash Reference Tag** presents a Flash animation.
- **Answer Tags:** Answers in a test can be recorded by a SCORM LMS.
 -  — **Start Quiz Tag** starts a quiz session. All answers need to be recorded before the end of the session.
 -  — **End Quiz Tag** ends a quiz session. The quiz is then evaluated.
 -  **id** — **Question Tag** allows the user to identify a question to give the answer.
 -  1 |  2 |  3 |  4 |  5 — **Multiple-Choice Tag** allows a learner to answer.
 -  Yes |  No |  Y |  N |  True |  False |  T |  F — **True-False Tag** is similar to the Multiple-Choice Tag.
 -  — **Fill-in-Blank Tag** allows a learner to give an answer to LMS. A popup window on a PC or PDA is used to enter a single line text.
- **Auxiliary Tags:** These tags turn on/off or control Hard SCORM LMS.
 -  — **Start Tag** turns on the Hard SCORM LMS.
 -  — **End Tag** turns off the Hard SCORM LMS.
 -  — **Pause Tag** suspends the Hard SCORM LMS.
 -  — **Continue Tag** resumes the Hard SCORM LMS.
 -  — **Learner Status Tag** provides status parameters to learners as an output.

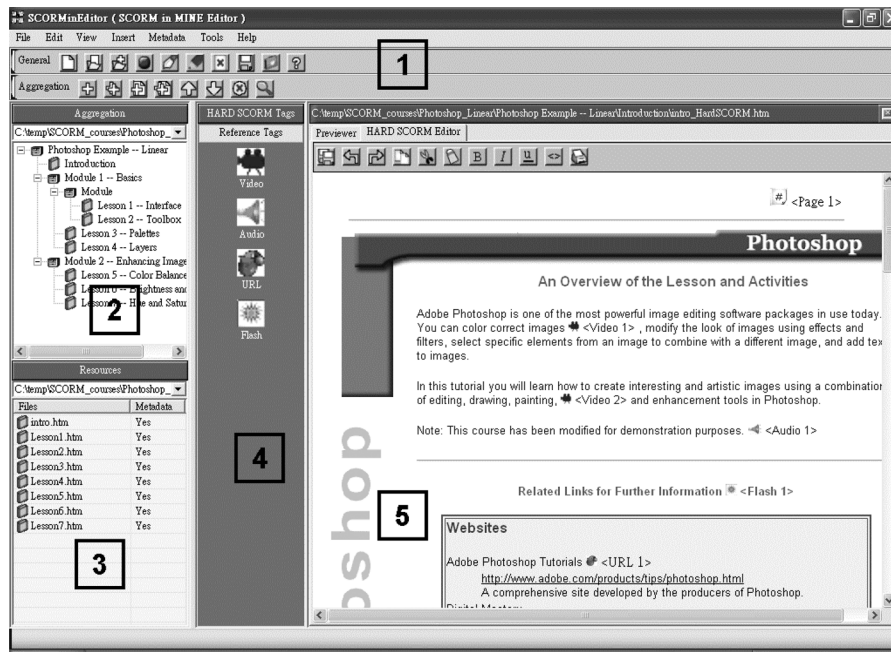


Fig. 2. Hard SCORM Authoring Tool.

An authoring tool was developed to generate Navigation Tags automatically according to the sequencing specification. Reference Tags and Answer Tags are inserted by the instructor when a particular content page is designed. Finally, Auxiliary Tags are printed on a Hard SCORM control panel, which is inserted into each Hard SCORM textbook for navigation controls. Some other authoring tools for SCORM compliant contents are found in the literature [11], [12]. These tools produce e-learning contents on the Web or standalone computers, instead of producing hard copy textbooks with augmented digital links. Generally speaking, the Hard SCORM Authoring Tool is a content aggregation and packaging tool, with special functions to design tags. The user interface is divided into the following five areas, as illustrated in Fig. 2.

- 1) Menu and Tool Bars.
- 2) Content Aggregation Window.
- 3) Resource Pool Window.
- 4) Hard SCORM Tag Bar.
- 5) Content Design Window.

The Menu and Tool Bars area includes a set of rich editing functions, which are similar to ordinary authoring tools. The Content Aggregation Window visualizes SCOs and assets, which can be inserted, deleted, or moved in a Content Aggregation by ordinary drag and drop methods. Multiple Content Aggregations can be loaded to the authoring tool for editing as well. The Resource Pool Window shows the learning resources in the current content aggregation. The Resource Pool Window also displays the available assets, which can be stored in different directories, or searched for on local and remote sites using criteria according to the SCORM metadata definition. The Hard SCORM Tag Bar contains four Reference Tags, which can be added to a Hard SCORM textbook. These Reference Tags, when triggered by a Hyper Pen, can present

multimedia resources such as flash animations, audio, video, or even another Web site as a reference. The Content Design Window shows the final design of assets. The Hard SCORM Authoring Tool can be used with ordinary Web presentation design tools, such as Microsoft FrontPage, which generate HTML files as assets.

In order to compute the organization and layout of a Hard SCORM textbook, a printing wizard is available. In general, the organization of a Hard SCORM textbook is similar to a traditional textbook. We believe that it is easier for the readers to follow if traditional reading behavior is considered. Thus, in the development of the Hard SCORM project, we carefully examine the behavior of ordinary readers. Their behaviors are considered in the design of tags, and in how Hyper Pen should be used. Based on a typical textbook, with limited modification, a Hard SCORM textbook includes the following.

- **The Hard SCORM Control Panel** contains Auxiliary Tags for navigation control.
- **The Table of Contents** of a Hard SCORM textbook enables the “choice” sequencing mode (i.e., to read a particular section). The hierarchy can be extended to an arbitrary level of depth. The LMS is able to recognize the Page Index Tags in the Table of Contents, and guide the user to flip to a specific page.
- **Chapters and Sections** (include tests) represent levels of aggregation. Each chapter, section, or test may contain Navigation Tags, Reference Tags, and Answer Tags.
- **The Index** contains a list of term-to-page pairs. Invocation of a Page Index Tag allows the user to navigate to a specific page which contains such term. The LMS is able to guide the user to flip to a specific page.

The Hard SCORM Control Panel is printed as a separated page for each Hard SCORM textbook. The invocation of tags in

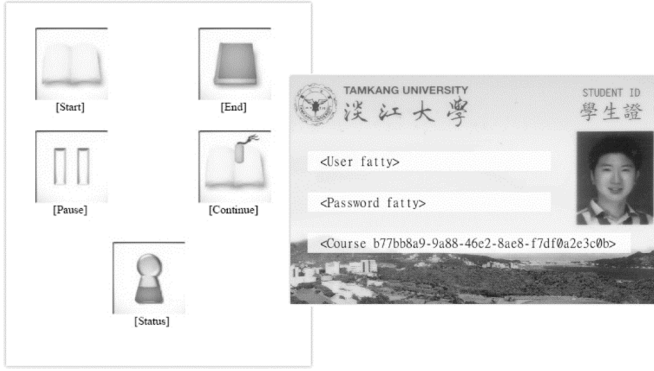


Fig. 3. Example of a Hard SCORM Student ID and Control Panel.

the control panel changes the navigation status of an individual learner. In addition, a Hard SCORM Student ID is printed for each student so they can log in to the LMS. The underlying database keeps student registration records, which are used to produce the IDs. Fig. 3 shows examples of a student ID and the Hard SCORM Control Panel. Different portions of a Hard SCORM textbook are shown in Fig. 4.

The Hard SCORM Authoring Tool and several video demonstrations are available for download at <http://www.mine.tku.edu.tw>. The authoring tool has been used by several e-learning content providers and universities around the world². Content created by the authoring tool can be printed and run on the *Hard SCORM Machine*, which we will discuss in the next section.

B. The Hard Scorm Machine (HSM)

Navigation behavior in a SCORM-compliant interactive session is defined by the sequencing rules associated with each SCO and its immediate descendent SCOs, according to the Sequencing and Navigation (S&N) specification. In a typical on-line learning session, the sequencing engine (part of the LMS) takes control. The engine decides which portion of learning content can be accessed or should be accessed. The SCORM S&N specification discusses the basic control mechanism. However, controlling user behavior during the reading of a Hard SCORM textbook requires a more sophisticated mechanism. It should be remembered that a learner can turn to any page of a textbook without letting the underlying sequencing engine gain control. Thus, user behavior, when a Hard SCORM Tag is accessed, should be controlled with caution. Abnormal behaviors should be accompanied by audio alerts. In some exceptional situations, interaction should be disabled. We make use of a HSM, which is based on the concept of finite state machine (FSM). We refer to FSM as a deterministic FSM in this paper.

A FSM, M , is represented as a 5-tuple

$$M = (Q, \Sigma, \delta, q_0, F)$$

²We continuously receive requests by universities and industrial partners in the SCORM community to receive a trial version of the Hard SCORM Authoring Tool 1.0. As of March 28th, 2006, at least 12 institutions had already installed the authoring tool.

³A finite-state machine can be nondeterministic. That is, the outcome of a nondeterministic finite-state machine results in different state transitions. However, the interaction in Hard SCORM should be deterministic.

where

- Q finite set of **internal states**, for HSM, $Q = \{i, f, \text{Reading}, \text{BC Analysis}, \text{Warning}, \text{Suspending}, \text{Quiz}\}$;
- Σ finite set of symbols called the **input alphabet**, for HSM, $\Sigma =$ set of Hard SCORM Tags;
- $\delta : Q \times \Sigma \rightarrow Q$ total function called the **transition function**, for HSM, is defined in Section III-C as transition tables;
- $q_0 \in Q$ **initial state**, for HSM, $q_0 = i$;
- $F \subseteq Q$ set of **final states**, for HSM, $F = \{f\}$.

A FSM operates as follows. Typically, a FSM starts with one initial state, q_0 , in Q . Within the finite set of input alphabets, when the transition function receives an input symbol, a , triggers a move from state q_0 to state q_1 . The transition function was applied one or many times. If one call to the function results in a state f in the set of final states F , the FSM terminates with an acceptance. If state transitions consume all input alphabets in the input sequence and no final state is reached, the FSM fails (exception condition). The HSM, with its detailed transition table, is defined in Section III-C.

Fig. 5 illustrates the HSM. The unique initial state, i , receives a start signal (when the student logs in to the system) and transits to the Reading state. In addition, the HSM has other states (and a quiz submachine) as described below.

- **Reading:** While a user is reading in a correct range of reading pages, the machine is waiting for a tag to be accessed. In general, a user spends most time in reading state.
- **Behavior and Context (BC) Analysis:** Once a tag is used and a correct destination page is confirmed, the machine stays in the BC Analysis state for action analysis. This state is used in *two-phase transactions* (to be discussed). The BC Analysis state is a special state, which represents a waiting status to be completed by another behavior.
- **Warning:** While a reader is reading in a wrong page (due to an incomplete two-phase transaction), the Warning state will signal audio messages and wait for the reader to provide a correct navigation choice.
- **Suspending:** The reader may suspend the state machine—this also suspends the accumulation of learning time.
- **Quiz Submachine:** The submachine is implemented as an assessment system controlled by a Java program of a SCO. The Quiz Submachine, as illustrated in Fig. 6, has two states:
 - **Question Identification :** As soon as the submachine is triggered (by the Start Quiz Tag), the user needs to scan a Question Tag to identify which question that to be answered.
 - **Answer Recording :** The user provides an answer (true/false, multiple choice, or fill-in-the-blank question types are acceptable to the tag recognition mechanism). Then, the user selects the next question. If no answer is provided and another Question Tag is selected, the newly selected tag will identify a question. Any question that

Table of Contents

- Chapter 1 Introduction to Data Structure.....1
- Section 1.1 Information and Meaning.....1
- Test 1.1.....24
- Section 1.2 Array in C25
- Test 1.2.....40
- Section 1.3 Structures in C.....42
- Test 1.3.....62
- Chapter 2 The Stack.....64
- Section 2.1 Definition and Examples.....64

Test 1.1

Please scan the icon and serial numbers of the question, and use the answer tags to answer the questions. After the Quiz is finished, please use the End Quiz Tag in the bottom of the page to exit.

1.1.1. In computer science, the quantities of information can't be measured.
☐ True <True> | ☐ False <False>

1.1.2. If a device can be in more than two possible states, the fact that it is in a particular state is more than one bit of information.
☐ True <True> | ☐ False <False>

1.1.3. The three bits in computer can be used to represent the integers 0 through 8.
☐ True <True> | ☐ False <False>

1.1.4. How many bits can be used to represent a decimal digit between 0 and 9 in the binary notation described previously?
☐ 1 <1> 3 ☐ 2 <2> 4 ☐ 3 <3> 5 ☐ 4 <4> 6 ☐ 5 <5> 7

1.1.5 The usual method used by computers to represent real numbers is
☐ 1 <1> Floating-point notation ☐ 2 <2> Integer notation ☐ 3 <3> Character Strings

1.1.6 The largest number that can be represented is _____.

☐ End Quiz

Index

- Abstract data type (see ADT).....13, 567, 370
- Abstract general list.....568
- Ackerman's function.....109
- Address.....6
- Address calculation sort.....354
- Adjustment list.....603
- Abstract typedef.....15
- All(x).....516
- Allocated node.....530
- ANSI.....44
- Argument.....137
- Array implementation.....179, 199
- AVL tree.....398
- B-tree.....413, 416, 446
- Backtracking.....136

Chapter 1 Introduction to Data Structures

A computer is a machine that manipulates information. The study of computer science includes the study of how information is organized in a computer. Thus, it is exceedingly important for a lent of computer science to understand the concepts of information organization¹ and manipulation in order to continue study of the field.

1.1 INFORMATION AND MEANING

If computer science is fundamentally the study of information, the first question that arises is, what is information? ¹ Unfortunately, although the concept of information is the bedrock of the entire field, this question cannot be answered precisely. In this sense the concept of information in computer science is similar to the concepts of point, line, and plane in geometry: they are all undefined terms about which statements can be made but which cannot be explained in terms of more elementary concepts. ¹

In geometry it is possible to talk about the length of a line despite the fact that the concept of a line is itself undefined. The length of a line is a measure of quantity. Similarly, in computer science ² we can measure quantities of information. The basic unit of information is the bit ¹, whose value asserts one of two mutually exclusive possibilities. For example, if a light switch can be in one of two positions but not in both simultaneously, the fact that it is either in the "on" position or the "off" position is one bit of information. If a device can be in more than two possible states, the fact that it is in a particular state is more than one bit of information.

Another way of thinking of this phenomenon is as follows. Suppose that we had only two-way switches ² but could use as many of them as we needed. How many such switches would be necessary to represent a dial with eight positions? Clearly, one switch can represent only two positions (see Figure 1.1.1a). Two switches can represent four different positions (Figure 1.1.1b), and three switches are required to represent eight different positions (Figure 1.1.1c). In general, n switches can represent 2ⁿ different possibilities. ³

Fig. 4. Example of a Hard SCORM Textbook.

is never accessed receives a failure score. Finally, the End Quiz Tag is used to end the quiz. Optionally, the test score can be reported to the user.

The HSM has a unique final state *f*. In the Reading state, when an End Tag is used, the machine reaches state *f* and is accepted. This means a particular interactive session is completed.

A *two-phase transaction* requires the user to access two Hard SCORM Tags. For instance, the user may choose a Page Index Tag based on the choice mode of the S&N specification. The two-phase transaction is not completed until the Page Tag on the destination page is accessed. In contrast, a *one-phase transaction* requires only one tag to be accessed. For instance, the user may access the Pause Tag to suspend reading. Or, the author may choose a Reference Tag to trigger a video clip (ended by closing the video window). We summarize these transactions below.

• Two-phase transactions

- **Content Navigation** : The transaction is triggered by a Navigation Tag, including Page Index Tags and Previous/Next Page Tags, and committed by a Page Tag. The transaction is controlled by the Behavior and Context Analysis State in Fig. 5. Depending on the behavior of a user, a different sound signal will be used, especially in the Warning State.
- **Quiz** : The Quiz Submachine handles two-phase transaction for answering a problem in a quiz session. The first phase is responsible for question identification; and the second phase checks for an answer type associated with its answer value, if multiple choice or true-false questions are used. For fill-in-the-blank questions, the second phase is completed with a text answer given in a popup window (on a PC). If two or more Question Tags

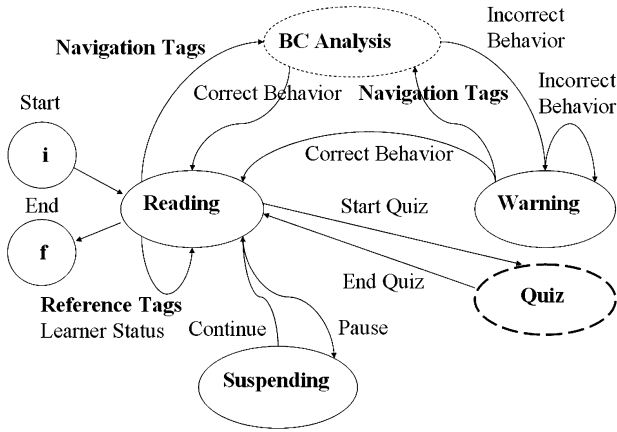


Fig. 5. HSM.

are used, the last tag is used to identify the question to be answered. The machine itself is controlled by another two-phase transaction, which takes a Start Quiz Tag and an End Quiz Tag.

- **One-phase transactions**

- **References:** One phase transactions deal with Reference Tags and Auxiliary Tags. In most situations, the HSM is in the Reading State. When a Reference Tag (video, audio, etc) is selected, an associated multimedia reference will be used and terminated automatically at the end of the particular reference duration. If the user checks his/her learner status, the machine will give a description (either using a popup window or using an audio record). Reading multimedia references or checking learner status occurs in the Reading State. However, if a Pause Tag is triggered, the machine moves to a Suspending State. A Continue Tag resumes the Reading State.

- **Exception Handling**

- **System Alert:** If the user fails to follow audio messages for navigation (i.e., incorrect pages are accessed for more than a predetermined number of times), the HSM forces the user to resume from a particular page, where the last normal navigation was accepted by the machine. Message M15 (see Table III) is used for exception handling.

C. Transition Tables of the HSM

Usually, the transition table of a FSM uses both the row index and the column index to represent state numbers. We use an alternative representation (see Table I). The column index represents states names as usual. However, the row index represents message types. Therefore, a column index and its associated *state actions* (as contents of the table) together represent the transition of a particular state.

A state action item (as the content of table) is represented as an “Actions $\rightarrow$ Next State” pair. “Actions” represents zero or more functions to be performed. “Next State” is a mandatory and unique state name since the HSM is deterministic. Functions in Actions include audio messages to be announced (M1–M14,

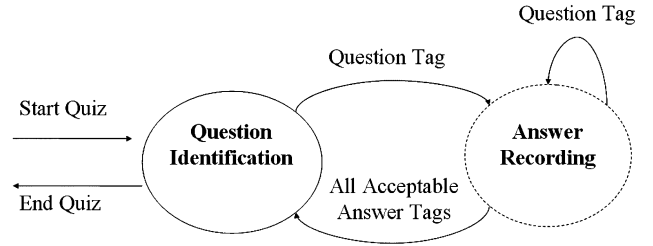


Fig. 6. Quiz Submachine.

summarized in Table III) and changes of state variables such as “current page” visited (e.g., $CP+$, $CP-$, $CP = X$ for increment, decrement, and assignment of current page, respectively). In addition, “Save LR” represents a transaction commit where the LR will be saved, if the current page accessed ends a particular transaction. “Report” indicates a function to announce a learner status or a score of quiz session. In the row index (the first column in Table I), some of the Hard SCORM Tags are used as the conditions of input alphabets. However, since a Page Tag will be compared with a state variable (i.e., CP), we use “[$P = CP$]” and “[$P \neq CP$]” to represent whether a Page Tag accessed meets an intended current page. Finally, a blank cell in the content of Table I represents an exception, in which case no action⁴ will be performed if the tag is selected. Mostly, the user has to select particular tags to operate the HSM. Table II illustrates the transition table of the Quiz Submachine. The representation is similar to Table I.

The implementation of the LMS requires a back-end server and a front-end interface. The front end interface is not limited to Hyper Pen. As we have discussed, we allow the user to use a PDA, cellular phone, PC, or Hyper Pen to access contents, using a consistent bookkeeping mechanism for *LRs*. This mechanism requires a representation and a sophisticated architecture.

D. Learner Records

A *LR* keeps track of the progress of an individual student’s learning experience. In SCORM, an *Activity Tree* represents the status of a particular experience while a student is visiting a SCORM-compliant course. It is not the definition of SCORM which restricts the representation of a particular implementation of an Activity Tree. The definition of SCORM also does not indicate how instructional design methodologies are applied to Activity Trees. Therefore, the representation of *LRs* can be defined as the representation of an Activity Tree plus internal data structures to support a SCORM-based LMS. Since *LRs* are not defined in the SCORM specification, we propose a generic definition to implement our HSM.

A *LR*, LR , is represented as a 4-tuple,

$$LR = \{AT, ALT, CC, LO\}$$

⁴Instead of no action performed, the system may provide an error message “Inadequate tag, please try again.” However, this message may bother the user. Thus, to present the message is optional.

TABLE I
TRANSITION OF THE HSM

	Reading	BC Analysis	Warning	Suspending	Quiz
Next Page Tag	M1, CP + → BC Analysis		M4, CP + → BC Analysis		
Previous Page Tag	M2, CP - → BC Analysis		M5, CP - → BC Analysis		
Page Index Tag	M3, CP=X → BC Analysis		M6, CP=X → BC Analysis		
Page Tag [P=CP]		Save LR → Reading	Save LR → Reading		
Page Tag [P≠CP]		→ Warning	→ Warning		
Reference Tags	M7 → Reading				
Pause Tag	M8 → Suspending				
Continue Tag				M9 → Reading	
Learner Status Tag	M10, Report → Reading				
Start Quiz Tag	M11 → Quiz				
End Quiz Tag					M12, Report → Reading

TABLE II
TRANSITION OF THE QUIZ SUBMACHINE

	Question Identification	Answer Recording
Question Tag [Q]	M13 → Answer Recording	M13 → Answer Recording
Answer Tags		M14 → Question Identification

TABLE III
SUMMARY OF AUDIO MESSAGES

Message ID	Message
M1	"Please turn to the next page"
M2	"Please turn to the previous page"
M3	"Please turn to page X"
M4	"You must turn to the next page"
M5	"You must turn to the previous page"
M6	"You must turn to page X"
M7	"Accessing multimedia reference"
M8	"Please take a break"
M9	"Continue reading"
M10	"Status report" (reports learner status)
M11	"Test begins"
M12	"Test ends" (reports test score if applicable)
M13	"Please answer question number Q"
M14	"Answer acquired"
M15	"System Alert: Please turn to page 'X' to continue"

where

- AT* generic tree representation that follows content aggregation, which contains SCO clusters as elements. Each cluster has a learning time and its associated links to child clusters. Also, each cluster may or may not contain a testing record, which contains two parts: a List of User Answers, and a Quiz Score. Testing records need to be stored if intelligent tutoring is to be used;
- ALT* accumulated learning time, which represents how long a student uses the LMS;
- CC* index to the cluster visited;
- LO* representation of *Learning Objectives*. The SCORM specification separates Learning Objectives from Learning Activities. The associations of contents to learning objectives are not restricted.

LRs are updated by HSM transitions as side effects. These side effects are maintained by a database, which can be used by an instructor for the assessment of student performance. For instance, Learning Objectives and learning time are typically used in performance assessment.

E. Consistency of Learner Records

As we have indicated before, LRs among different devices are maintained in a consistent manner such that when the user uses different devices, his/her learning records are portable and consistent. This is controlled by our backend LMS. Navigation on PCs, PDAs, or cellular phones is similar to the HSM principle in terms of learning experiences. LRs are saved if a particular cluster is completed (i.e., a transaction is completed). However,

PDAs and cellular phones have to deal with the offline situation. A PDA or cellular phone saves LRs and uploads these records to our central server as soon as the mobile devices are online. We discuss how the central server is implemented using the .net architecture in the next section.

IV. IMPLEMENTATION OF THE HARD SCORM LMS

The first implementation of the Hard SCORM LMS was completed in January 2005 and demonstrated at an ADL Conference in February 2005. In order to support new mobile devices, the interface on PDAs and cellular phones was revised in early 2005. We discuss the implementation from different perspectives below.

A. Special Functions in the Implementation

Due to the screen size and the storage limitations, a few important functions are implemented on PDAs and Smartphones. Object reflow technique resizes text, video, and image on PDAs and Smartphones to fit their window sizes. As a result, the reader only needs to use the vertical scroll bar to read contents. The reader can also add personal notes to the content, which can be uploaded to the server. In addition, special caching algorithms are implemented in our LMS and mobile devices. The Sequencing and Navigation specification discussed in SCORM requests learners to follow a predefined navigation topology.

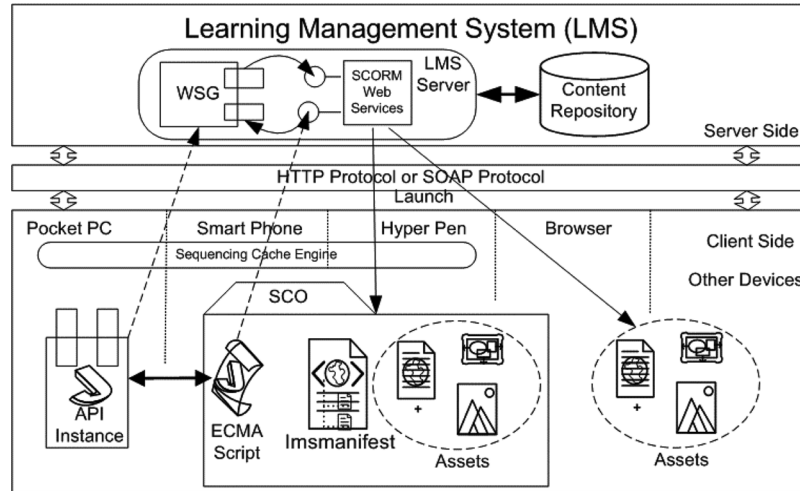


Fig. 7. Revised architecture for multiple devices based on SCORM Run-Time Environment.

Thus, ordinary caching mechanisms are not suitable. Parameters of our caching strategy include control modes of sequencing rules, content sizes, frequency and duration of visits, and network conditions.

On the other hand, Hyper Pen needs to recognize tags. Our first attempt and the current approach are to recognize Hard SCORM Tags in text forms, since OCR-based technology is well-developed. However, text-based tags mix the tags with text instructions. Our next step of implementation will use technologies from content-based image retrieval with a small set of pre-defined shapes and to allow graphical tags to be recognized. Thus, clearness of text-based instructions will be improved.

B. A Revised Architecture

We revised the system architecture discussed in the run-time environment specification of SCORM to serve the needs of accommodating multiple learning devices (see Fig. 7). The Web service framework is suitable to support our implementation since the framework reduces the overhead between service requesters and service providers. The advantages of using Web service architecture in e-learning, in general, have been reported in the literature [19]–[21]. SCORM is XML-based and Web service architecture is also an XML-based message framework. As a result, the data model and the API functions of SCORM are easily integrated with the framework. In Fig. 7, the Original SCORM APIs are converted and used as Web services. The basic operations of the Original SCORM APIs, such as administration, session management, sequencing control, learning content delivery, etc., are supported by the revised architecture. These services are easily extended according to different specific services requested by different end devices (i.e., PCs, PDAs, etc.).

To ease the implementation of SCORM API adopters, a WSG (Web Service Gateway) for the SCORM API is also introduced. With the help of WSG and the revised architecture, traditional invocation of SCORM APIs and other LMS servers which do not support Web services can also use WSG to wrap the original information and send requests to the SCORM web service (part of our LMS). The ECMAScript [22] in the SCORM Run-Time

Environment specification interacts with SCORM API instances (Java Applets) to communicate with the LMS server. In our framework, however, the SCORM API instances are not the only way to communicate with the LMS server. It is also possible to use ECMA script to create messages and send them to the SCORM Web service. This addresses the need to support diverse communication channels. This solution enables platforms which do not support Java Applets.

C. The Transmission Protocol

In general, the transmission protocol between the LMS and end user devices can be either HTTP or SOAP [23]. We use SOAP messages to encapsulate parameters passed in the SCORM Run-Time Environment. However, the underlying transmission protocol for the mobile devices uses either IEEE 802.11 or GPRS, depending on a particular hardware implementation. Thus, parameters are bound in SOAP messages, which are transmitted over IEEE 802.11 or GPRS.

D. Types of Mobile Devices Supported

The Hard SCORM LMS supports different learning devices including PCs, PDAs [24], [25], cellular phones, and Hyper Pen [26]. Figs. 8 and 9 illustrate the user interfaces on PDAs and Smartphones running Windows CE. So far, we have tested a few PDAs (Dopod 700, HP iPAQ 5550, and AnexTEK SP230) and Smartphones (Dopod 565 and Mio8390) in our proposed architecture. In addition, the Hard SCORM LMS is running on Web browsers for PCs. Our demonstrations are available at (<http://scorm.mine.tku.edu.tw>).

V. EXPERIMENTS AND ANALYSIS

A. Experiment Assumptions

The Hard SCORM LMS was developed and used. To compare the effectiveness of the proposed learning environment with traditional learning, three substantial assumptions had to be made beforehand.

Assumption 1: The Hard SCORM project offers learners the dual advantages of multimedia capabilities and nonlinear reading style.



Fig. 8. SCORM LMS on PDA.



Fig. 9. SCORM LMS on Smart (cellular) Phone.

The Hard SCORM LMS was designed to support multimodal multimedia learning devices. Hard SCORM textbooks make possible multimedia presentations which could not be displayed in ordinary textbooks. Furthermore, the features of augmented textbooks facilitate nonlinear learning behavior. The multimedia presentations enhance the richness and diversity of learning content, and the nonlinear learning behaviors provide for the needs for learners to read in a well-organized learning environment.

Assumption 2: The Hard SCORM textbooks are able to solve the problems of information overload and learning disorientation in a hypermedia learning environment.

The Hard SCORM LMS integrates the functionality of multimodal multimedia devices with the advantages of conventional textbooks. Certain issues often arise within a hypermedia learning environment, such as information overload and learning disorientation. Those issues mainly result from the sheer volume of loosely organized information in such an environment. Learners might become easily disorientated and distracted due to the multimodal learning contents, which can negatively affect their learning performance. When we refer to information overload we mean the heavy burden of receiving massive amounts of information in a short time. When we refer to learning disorientation we mean the obscuring of learning objectives and contents in a chaotic learning environment.

Assumption 3: The Hard SCORM project retains and augments the familiar conventional learning behaviors of reading hardcopy books.

Our Hard SCORM LMS extends multimodal multimedia learning devices to facilitate learning on hardcopy books, while retaining the substantiality, stability, and convenience

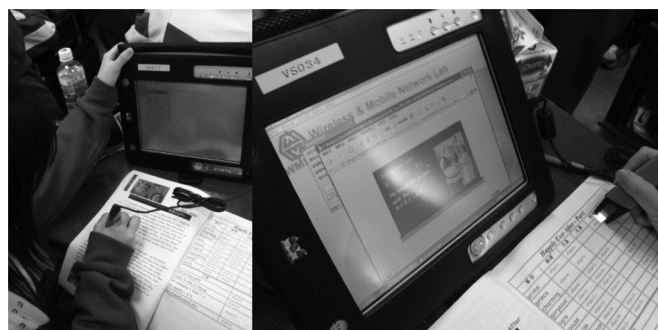


Fig. 10. Using Hard SCORM LMS and Hyper Pen.

of traditional textbooks. Learners are able to perform familiar actions, such as page turning, book marking, and annotation on hardcopy books, thereby enhancing learning efficiency. Our project aims to provide the possible solutions to accommodate future learning styles with advantages in both physical and cyberspace learning environments.

B. Experiment Setup

As part of our Hard SCORM project, testing of the Hard SCORM LMS took place at a local school⁵ for three weeks during the summer of 2005. Students were divided into two groups. The first group (25 students) used tablet PCs equipped with Hyper Pens (see Fig. 10). The second group (seven students) used PDAs. For each student in the first group, a ViewSonic Tablet PCV1100 tablet PC and an Anoto AB C-Pen 20 OCR device (i.e., Hyper Pen) were used. For the students in the second group, seven PDAs were used (three HP iPAQ 5550, two Dopod 700, and two AnexTEK SP230 PDAs). Students used the devices both at home and in school. In addition, all students used the Hard SCORM LMS on a PC while they were in school. The teachers used our authoring tool and designed six units of course material based on content supported by the Ivy League Analytical English, Taiwan. Three research assistants worked closely with the high school teachers and students for system installation and training.

A Likert scale⁶ (as shown in Table IV) questionnaire was developed to probe students' feedback and to verify our assumptions. The questionnaire contained 18 questions which can be categorized into six items relating to three different types of learning platforms: traditional textbooks, Hard SCORM textbooks, and the online LMS.

C. Experimental Result and Analysis

After collecting and analyzing the questionnaire answers, the experimental result from a quantitative perspective has been summarized as Table V. Items 1–1 and 1–2 are designed for verifying assumption 1, and items 2–1 and 2–2 apply to assumption 2. Finally, assumption 3 can be supported by items 3–1 and 3–2. The acronyms *CT*, *OL*, and *HS* stand for the three types of the

⁵The Chun-Li High School, Taiwan is using our system to teach English in a 11th grade class of 32 students.

⁶The Likert technique presents a set of attitude statements. Subjects are asked to express agreement or disagreement of a five-point scale. Each degree of agreement is given a numerical value from one to five. Thus a total numerical value can be calculated from all the responses.

TABLE IV
SCORING OF THE DEGREE OF ACCEPTANCE FOR EACH ITEM

Degree of acceptance	Scores
Strongly agree	5
Agree	4
Neutral	3
Disagree	2
Strongly disagree	1

distinct learning platforms, i.e., the conventional textbooks, the online LMS, and the Hard SCORM textbooks respectively.

Based on our experimental results, we have noted some interesting patterns of response, as summarized and analyzed below. Concerning the diversification of learning contents, although the mean value of using Hard SCORM textbooks is intermediate among the three different learning platforms, the percentage of “strong agreement” responses for HS in item 1–1 (9.52%) is higher than the other two (4.76%, respectively). This interesting result reveals that a few learners agreed the learning contents are more diverse in the augmented learning environment. However, the result in item 1–2 shows the learning contents are less clear and less well organized in HS than in the other two learning platforms. This might be caused by the supplemental learning contents being delivered to the learners via the digital devices instead of using a simplex learning platform.

Regarding assumption 2, the result in item 2–1 shows most learners felt that the information overload was increased while using the augmented learning devices. However, the result also revealed that both the conventional textbooks and the online LMS have the same problem. It should be noted that the Hard SCORM textbooks augment different learning platforms and result in an increased information overload. Concerning item 2–2 from a negative perspective in the questionnaire, the result shows that learning disorientation turns up easily while reading via online LMS. Furthermore, the learners are easily distracted while reading the conventional textbooks. On the other hand, the learning contents are impressive to the learners while learning through Hard SCORM textbooks, and this may be mainly ascribed to the interconnection between hardcopy books and digital multimedia learning contents. Finally, learning performance was maintained in the Hard SCORM learning environment.

Statistical analysis of the results of items 3–1 and 3–2 shows that learning through conventional textbooks fits the major learning behaviors and provides efficiency and convenience. However, the percentage of “strong disagreement” responses for conventional textbooks (14.29%) is the highest among the three learning platforms. This is an interesting result and may reflect the fact that a few learners who are well adapted to the digital learning environment felt that efficiency and convenience were unsatisfactory while reading conventional textbooks. The experimental results also signify that most learners are still accustomed to having learning activities via hardcopy textbooks rather than purely using the electronic learning devices.

Another set of questions was designed by a professor in the school of education to evaluate our LMS and course contents. Out of the 32 sets of answers from the students, 25 included

useful suggestions (20 from the first group and five from the second group). The quantitative result shows that learners agreed that using Hyper Pens augmented the textbooks, facilitated exploring the associated multimedia learning resources, and improved the learning performance during the learning processes (Mean = 3.48, SD = 0.81). The students were highly accepting the stability of the proposed Hard SCORM LMS (Mean = 4.57, SD = 0.51). Below is a summary of learner experience and feedback after using Hard SCORM from a qualitative perspective. Generally speaking, the students were very interested in using the new technology. Some of the notable positive comments include:

- the design of LMS interface is user friendly;
- the user’s menu and online help are clear and easy to read;
- the use of Hyper Pen for learning fit ordinary reading behavior;
- the Hyper Pen records LRs precisely;
- the LMS provides immediate feedback;
- the LMS helps learners to understand their learning performance;
- the LMS can help to find out some critical points in learning;
- the hyperlinks from a textbook to multimedia presentations are useful.

However, there were also some drawbacks to using our LMSs:

- the Hyper Pen does not recognize tags effectively;
- the PDA screen is too small for reading;
- the Tablet PC is too heavy;
- it takes a long time to charge the Tablet PCs.

In addition, we have observed that most young students are interested in using high-tech devices such as PDAs and Smartphones for learning. Learning performance is improved, especially in English vocabulary, due to the fact that learners are able to read the vocabulary on hardcopy books as well as accessing the corresponding pronunciation via audio links. We also asked two professors in computer engineering to evaluate software performance of our system. In their experience, using Hyper Pen was easy and the access time was very short. However, downloading lecture notes to PDAs and Smartphones is still slow due to the transmission rate of GPRS.

D. Use Case in a Domestic Airline Company

Additional testing of the Hard SCORM LMS was carried out by pilots at TransAsia Airways (<http://www.tna.com.tw/>). The PDA version of the LMS, called “*Pocket SCORM*” was supported by a collaboration project with the Institute for Information Industry, Taiwan. The system received the **2005 Brandon Hall Excellence in Learning Awards**. In the customized version of LMS, aircraft security checks were performed by a pilot before taking off. The checklist items were treated as testing questions which are controlled by JavaScript. As soon as the pilot completes the routine check, the record can be uploaded to a central server. In addition, the customized LMS facilitates video presentations on Pocket PCs. Thus, the pilots are able to read news or lecture presentations while they are waiting for missions. A demonstration of the award-winning system is available at <http://www.elearn.org.tw/PocketSCORM/>.

TABLE V
RESULT OF THE QUESTIONNAIRE

Items	Learning Platform	Mean	Std. Deviation (SD)
1-1 Learning contents were provided in diverse forms to facilitate learning activities.	CT	3.62	0.74
	OL	3.48	0.68
	HS	3.57	0.81
1-2 The structure of learning contents was clear and well-organized.	CT	3.34	0.83
	OL	3.48	0.75
	HS	3.24	0.77
2-1 The information overload could be reduced by using this learning platform.	CT	3.38	0.67
	OL	3.33	0.58
	HS	3.00	0.78
2-2 The learning contents are easily forgettable (from a negative perspective).	CT	2.88	0.91
	OL	3.14	0.73
	HS	2.86	0.57
3-1 It is easy to annotate and make notes within the learning contents.	CT	3.71	0.96
	OL	3.00	0.89
	HS	3.10	0.89
3-2 The learning method is efficient.	CT	3.52	0.81
	OL	3.19	0.75
	HS	3.10	0.77

VI. CONCLUSIONS

Our project represents the first time the Hyper Pen device has been used in a SCORM-based LMS. We designed and implemented such an LMS, based on Microsoft's Web service architecture. A few systems on mobile devices have been developed to be used in conjunction with the Hyper Pen. As an integrated system, the LMS supports students in using multimodal learning devices to read the same course material consistently. Individual LRs are maintained in a central server. To allow an instructor to develop SCORM-compliant courseware for supporting the ubiquitous learning environment, an authoring tool (partially addressed in this article) was also developed. It should be noted that the PDA and cellular phone versions of our proposed systems received the 2005 Brandon Hall Excellence in Learning Awards. We are currently seeking a telecommunication company to transfer our technology.

The main contribution of this paper is to bridge the gap between reading in a traditional manner and learning by using electronic devices (such as PDAs). Both SCORM and a Web Service architecture were used to realize this concept. Although the rating statistics of the three distinct learning platforms in the experimental results are not significantly different, however, the learning performance while using our proposed augmented learning environment is improved, and the information overload is not increasing significantly while dealing with the multimodal learning devices. We believe that, with high-tech mobile devices, ubiquitous e-learning will have a positive impact on education, especially for the younger generation who are enthusiastic about using mobile devices and accomplished in their use.

We are looking at two directions for our future research. First, the current Hard SCORM Tags are static. That is, because the navigation topology can vary from student to student, the activation of a tag results in a unique outcome (e.g., accessing a fixed video or jumping to the next page). However, it is possible to use an encoding technique to embed multiple branching for tag activations. Such an encoding technique could be used with an intelligent tutoring system based on a revision of the

sequencing and navigation specification, as well as an assessment model of students' test performance. In conjunction with a printed textbook, supplementary materials might be offered on PCs or PDAs, for example as remedial material for those who need help. We are developing the assessment mechanism based on a technique that uses a so called student-problem table and its associated caution indices. Second, we intend to enhance our Pocket SCORM system (part of our LMS) by using a built-in camera on a PDA or cellular phone. The camera can be used as a tag recognition device, thereby replacing the use of Hyper Pen. However, current Hard SCORM Tags are small in size, making reliable recognition problematical. We have developed a second version of tags, which embed colored shapes printed underneath the ordinary text. The built-in camera captures these tag images and transmits them back to a tag recognition server, which is connected to our LMS. Without interfering with reading, and with the help of GPRS or IEEE 802.11 communication techniques, a student can use a single PDA or cellular phone and a Hard SCORM II textbook in a library to read textbooks and watch supplementary video presentations.

REFERENCES

- [1] K. O'Hara and A. Sellen, "A comparison of reading paper and on-line documents," in *Proc. CHI 97*, Atlanta, GA, 1997, pp. 335-342.
- [2] M. Back, J. Cohen, R. Gold, S. Harrison, and S. Minneman, "Listen reader: an electronically augmented paper-based book," in *Proc. SIGCHI Conf. Human Factors in Computing Systems*, Mar. 2001, pp. 23-29.
- [3] P. Luff, C. Heath, M. Norrie, B. Signer, and P. Herdman, "Only touching the surface: creating affinities between digital content and paper," in *Proc. ACM Conf. Computer Supported Cooperative Work*, Nov. 6-10, 2004, vol. 6, pp. 523-532.
- [4] M. C. Norrie and B. Signer, "Switching over to paper: a new web channel," in *Proc. 4th Int. Conf. Web Information Systems Engineering*, Dec. 10-12, 2003, pp. 209-218.
- [5] B. Signer and M. C. Norrie, "Multi-layered cross-media linking," in *Proc. 14th ACM Conf. Hypertext and Hypermedia*, Aug. 26-30, 2003, pp. 106-107.
- [6] ADL SCORM 2004 Documentation 2005 [Online]. Available: <http://www.adlnet.org>
- [7] N. A. Abdallah, C. Bailey, and H. Davis, "Augmenting SCORM manifests with adaptive links," in *Proc. 15th ACM Conf. Hypertext & Hypermedia*, Aug. 2004, pp. 183-184.

- [8] Y. Atif, J. Berri, and R. Benlamri, "An adaptive multimedia approach to e-learning," in *Proc. 10th IEEE International Conf. Electronics, Circuits and Systems*, Dec. 14–17, 2003, vol. 3, pp. 1224–1227.
- [9] S. A. Kazi, "A conceptual framework for web-based intelligent learning environments using SCORM-2004," in *Proc. IEEE Int. Conf. Advanced Learning Technologies*, Sep. 1, 2004, pp. 12–15.
- [10] S. T. Li and C. H. Lin, "On the distributed management of SCORM-compliant course contents," in *Proc. 19th Int. Conf. Advanced Information Networking and Applications*, Mar. 25–30, 2005, vol. 1, pp. 221–226.
- [11] J. Su, S. Tseng, J. Weng, K. Chen, Y. Liu, and Y. Tsai, "An object based authoring tool for creating SCORM compliant course," in *Proc. 19th Int. Conf. Advanced Information Networking and Applications*, Mar. 25–30, 2005, pp. 209–214.
- [12] J. T. D. Yang, C. H. Chiu, C. Y. Tsai, and T. H. Wu, "Visualized online simple sequencing authoring tool for SCORM-compliant content package," in *Proc. IEEE Int. Conf. Advanced Learning Technologies*, Sep. 1, 2004, pp. 609–613.
- [13] Advance Distribute Learning (ADL) Home Page 2005 [Online]. Available: <http://www.adlnet.org>
- [14] IMS Global Learning Consortium (IMS) Home Page 2005 [Online]. Available: <http://www.imspj.org>
- [15] Aviation Industry CBT Committee (AICC) Home Page 2005 [Online]. Available: <http://www.aicc.org>
- [16] Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE) Home Page 2005 [Online]. Available: <http://www.ariadne-eu.org>
- [17] Learning Technology Standards Committee (LTSC) Home Page 2005 [Online]. Available: <http://ltsc.ieee.org>
- [18] *IEEE Standard for Learning Object Metadata (LOM)*, , 2005 [Online]. Available: <http://ltsc.ieee.org>
- [19] C. P. Chu, C. P. Chang, C. W. Yeh, and Y. F. Yeh, "A web-service oriented framework for building SCORM compatible learning management systems," in *Proc. 2004 Int. Conf. Information Technology: Coding and Computing*, Apr. 5–7, 2004, pp. 156–161.
- [20] X. Liu, A. S. El, and N. D. Georganas, "An implementable architecture of an e-learning system," in *Proc. Canadian Conf. Electrical and Computer Engineering*, May 4–7, 2003, pp. 717–720.
- [21] G. Vossen and P. Westerkamp, "E-learning as a web service," in *Proc. 7th Int. Database Engineering and Applications Symp.*, Jul. 16–18, 2003, pp. 242–249.
- [22] *IEEE 1484.11.2-2003 Standard for Learning Technology—EC-MAScript Application Programming Interface for Content to Runtime Services Communication*, , 2005 [Online]. Available: <http://ltsc.ieee.org/>
- [23] T. K. Shih, W. C. Chang, N. H. Lin, L. H. Lin, H. H. Hsu, and C. T. Hsieh, "Using soap and.NET web service to build SCORM rte and lms," in *Proc. 7th Int. Conf. Advanced Information Networking and Applications*, Mar. 27–29, 2003, pp. 408–413.
- [24] N. H. Lin, T. K. Shih, H. H. Hsu, H. P. Chang, H. B. Chang, W. C. Ko, and L. J. Lin, "Pocket SCORM," in *Proc. 24th Int. Conf. Distributed Computing Systems Workshops*, Mar. 23–24, 2004, pp. 274–279.
- [25] T. K. Shih, N. H. Lin, H. P. Chang, and K. Huang, "Adaptive pocket SCORM reader," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Jun. 27–30, 2004, vol. 1, pp. 325–326.
- [26] T. H. Wang, H. P. Chang, Y. L. Sie, K. H. Chan, M. T. Tzou, and T. K. Shih, "Reading SCORM compliant multimedia courses using heterogeneous pervasive devices," in *Proc. ACM Multimedia Conf.*, Singapore, Nov. 2005.



Timothy K. Shih (SM'02) is a Professor in the Department of Computer Science and Information Engineering, Tamkang University, Taipei Hsien, Taiwan, R.O.C. He has edited many books and published over 350 papers and book chapters, as well as participated in many international academic activities, including the organization of more than 40 international conferences and several special issues of international journals.

Dr. Shih is the Associate Editor of *ACM Transactions on Internet Technology*. He was previously the

Associate Editor of *IEEE TRANSACTIONS ON MULTIMEDIA*. He is the Founder and Co-Editor-In-Chief of the *International Journal of Distance Education Technologies*, published by Idea Group Publishing. He is a member of ACM and of the Educational Activities Board of the IEEE Computer Society.



Te-Hua Wang graduated in 1997 and received the Master's degree in 2001, both from Tamkang University, Taipei Hsien, Taiwan, R.O.C., where he is currently pursuing the Ph.D. degree in the Department of Computer Science and Information Engineering.

He took the alternative service in the Ministry of Education of Taiwan from 2001 to 2003, and then returned to academia. He is the Chief Researcher of the SCORM Research Team in the Multimedia Information Networking Lab, Tamkang University. His current research interests include distance learning and multimedia computing. Recently, he has joined the development of the Hard SCORM Authoring Tool, and is one of the main contributors of the Hard SCORM research project founded by the National Science Council in Taiwan.



Chih-Yung Chang received the Ph.D. degree in computer science and information engineering from National Central University, Jung-Li, Taiwan, R.O.C., in 1995.

He is currently an Associate Professor with the Department of Computer Science and Information Engineering, Tamkang University, Taipei Hsien, Taiwan. His current research interests include wireless sensor networks, mobile learning, bluetooth radio networks, *ad hoc* wireless networks, and mobile computing.

Dr. Chang served as an Associate Guest Editor of the *Journal of Internet Technology* (JIT, 2004), the *Journal of Mobile Multimedia* (JMM, 2005), and as a member of the Editorial Board of *Tamsui Oxford Journal of Mathematical Sciences* (2001–2005).



Tai-Chien Kao received the Ph.D. degree in educational technology from Purdue University, West Lafayette, IN, in 1996.

She joined the faculty of the Institute of Education at Dong Hwa University, Taiwan, R.O.C., and is currently a Professor there, where her work focuses on technology-based scaffolding in the context of web learning and mobile learning environment. Related interests include the design, development, implementation, and evaluation of technology-based instructional strategies that promote students' independent

learning and self-regulation.



Douglas Hamilton has extensive experience as a Designer, Developer, Project Manager, and Consultant to faculty in the areas of learning technology, e-learning, courseware development, and media production. He has been a Member of Staff at the Academic ADL Co-Lab, Madison, WI, since 2001. Previously, he was with the University of Wisconsin-Madison, St. Andrews University, Fife, U.K., Minnesota Public Radio, and Chemeketa Community College, Salem, OR. He has also worked in the private sector as Internet Technologies Manager for

Musicnotes.com, a digital sheet music publisher. He is currently responsible for coordinating the Academic Co-Lab's work with Florida Virtual School. Recent projects include designing and developing online learning materials, section 508 accessibility conversions and training, and producing web-based multimedia. He has created an introductory course on SCORM that has been translated into several languages. He is an elected member of the Wisconsin Public Radio Association Board of Directors.