An Ontological Model for SCORM-Compliant Authoring Tools

JIN-TAN DAVID YANG, WEN-CHIH CHEN*, CHUN-YEN TSAI** AND MEI-SHENG CHAO+
Department of General Literacy
**Graduate Institute of Information and Computer Education*
***Graduate Institute of Science Education*
National Kaohsiung Normal University
Kaohsiung, 802 Taiwan
E-mail: yangdav@nknucc.nknu.edu.tw
+Department of Education
National Taiwan Normal University
Taipei, 106 Taiwan

This paper models authoring course practices on the SCORM-compliant Content Repository Management System (CRMS) and presents three new invaluable components: the Ontology-Based Outline Authoring Tool (OBOAT), the Visualized Online Course Authoring Tool (VOCAT), and the Visualized Online Simple Sequencing Authoring Tool (VOSSAT). The pivotal role that ontologies play in authoring SCORM-compliant learning objects is again reconfirmed based on their ability to provide an outline of precisely-defined terms that can be communicated among teachers by means of visualized presentations. Teaching materials that require outlining most likely rely on conceptual models in the form of ontologies that open up new ways for teachers to move from a learning object-oriented view to a content framework-oriented view. In the same vein, VOCAT and VOSSAT both offer user-friendly interfaces on the Web. While the former allows teachers to compile teaching materials, the latter enables them to devise their own instructional strategies. The prime contributions of this study are newly-devised innovative tools that teachers can easily use to author teaching materials that will suit their teaching contexts, including their students' needs and backgrounds. Finally, we discuss the implications of this study for education.

Keywords: SCORM, content repository management system (CRMS), RDF/RDFS, ontology, IMS SS

1. INTRODUCTION

For some time, the WWW has been regarded as the ideal platform on which to develop Internet applications; this is by virtue of its powerful communication paradigm, which is based on multimedia, its browsing capabilities, and its open architectural structure, which, to a great extent, collectively facilitate the integration of different types of content and systems. More to the point, learners can simply open their browsers and immerse themselves in a learning environment to obtain the knowledge they seek. At the same time, this universal interface also offers an efficient, convenient way for teachers to author their own teaching materials as long as the learning objects have been provided on the Web.

Received July 7, 2004; revised February 1, 2005; accepted May 18, 2005.
Communicated by Robert Lewis.

To further upgrade the ability of users to share the teaching materials available in these systems, various international organizations have proposed standard formats, including SCORM 2004 [1-3, 10, 12]. Given such standard formats, teaching materials in different learning management systems can now be not only shared, but also reused, and even recombined. Among these international standards, the Sharable Content Object Reference Model (SCORM), which integrates IMS, LOM, and AICC, has perhaps emerged as the most popular international standard in recent years. ADL SCORM 2004, in particular, aims at facilitating adaptive learning through declaratively rule-based descriptions. Nevertheless, it must be acknowledged that structures with complicated sequencing rules in the Activity Tree (AT) in SCORM 2004 make the design and creation of course sequences rather difficult, at best. Bearing this in mind, to help teachers and instruction designers structure course units based on sequencing behavior rules, we propose newly-devised authoring tools which can be represented as graph modes, thus providing a considerably more user-friendly platform than those currently available.

While learning and course materials are typically composed of tree-like structures or directed-graphs, clearly annotating a SCORM-Compliant content package with metadata (i.e., data describing its contents/functionalities) is essential for authoring purposes (courses/materials). In this regard, proposals for the standardization of annotations used in various languages have previously been put forward by the W3C [23], the most notable of which are the Resource Description Framework (RDF) and the RDF Schema, which are currently being used both to represent information and to exchange knowledge on the Web. Such annotations, however, are of limited value unless, of course, there is a commonly-held understanding as to their precise meaning.

As metadata constructed from a set of ordered terms for the discovery and filtering of learning objects (LOs) from a content repository, ontologies can indeed help to fulfill this requirement by providing "a representation of a shared conceptualization of a particular domain" that can be easily handled by teachers [5]. Moreover, in terms of authoring, the outlines of materials are more apt to be based on conceptual models in the form of ontologies that open up new avenues for individuals to move from a learning object-oriented view to a content framework-oriented view. What is clear, however, is that learning objects are structured, interlinked, combined, and used, thereby facilitating interaction among teachers. In this way, the objectives of the content framework of the teaching materials can be achieved. This kind of semantic web, containing ontologies and machine-processable relational metadata, can greatly facilitate the construction of course units.

The RDF Schema (RDFS) itself is already recognized as an ontology, that is, a knowledge-based language, since at its core, it focuses on classes and their properties (i.e., the binary relations between them), on the range of these properties and the domain constraints placed on them, and on subclass and sub-property (sub-sumption) relations [17]. Being a relatively primitive language in terms of its ability to describe resources, RDF/RDFS provides ontological support for the authoring of course outlines by filtering out learning objects (LOs). For teachers, the benefits of filtering out LOs might be mostly related to their teaching requirements. Unlike the eXtensible Markup Language (XML), which offers a simple hierarchical structure in the form of a tree-like presentation [23, 25], RDF/RDFS is represented as a graph, which makes its application both easier and faster.

To reduce the time required for searching and assembling, such as that required for LOs, for sharable content objects (SCOs), or for a Content Package (CP) [1] within a content repository, an ontology can lay the groundwork for authoring courses though it does only offer a static outline. The shortcoming here is that it is more often than not quite difficult for teachers to adapt this type of outline to meet their students' preferences or needs. This is why, in this paper, we propose semi-dynamic outlining. In essence, making an outline is a pragmatic way of evaluating whether a set of vocabulary items is adequate, consistent, or valid, since teachers are able to assess whether the vocabulary selected can serve as a framework for a conceptual map in a higher-level knowledge abstraction [13]. To be sure, a semi-dynamic authoring environment is preferable for devising a teaching context that is suitable in light of the students' backgrounds.

In this study, we apply RDF/RDFS techniques as we explore how to best author an aggregation of metadata as web-based ontologies for the mathematics domain at the secondary school level in Taiwan. More specifically, we focus on joint-authoring course practices based on a specific content repository, namely, the Content Repository Management System (CRMS), which consists of SCORM-Compliant learning objects, such as the SCO, Asset, and a Content Package. CRMS offers all of the learning resources available in the SCROM 2004 format for the compulsory education system in Taiwan.

We start with a literature review in the next section and follow it with a discussion of the framework of the methodology we used in the present study. In sections 4 and 5, we demonstrate selected modeling authoring practices based on various scenarios and also include some recommendations for future study.

2. LITERATURE REVIEW

In this section, we identify four themes which we regard as fundamental to this study. They are 1) CRMS, 2) the RDF/RDFS and ontology, 3) an example of a domain ontology with RDF/RDFS, and 4) visualized online authoring tools. We present a brief review of the literature for each.

2.1 CRMS

CRMS, a SCORM 2004-Compliant content repository, consists of a collection of such teaching materials as Learning Objects or Content Aggregations [26]. CRMS is a web-based application which is designed for teachers who wish to easily and quickly assemble or create their own course materials from existing sharable LOs and CPs. In other words, teachers and instruction designers can author courses by searching and editing in a very effective way using CRMS.

CRMS is also a platform with extensible functionalities. As shown in Fig. 1, when viewed horizontally, the CP upload, the authoring tools, and even the ADL 2004 RTE (Run Time Environment) modules can be added in their entirety. In addition, new functionalities, like Lego's plug-ins, can also be integrated into CRMS. On the other hand, when viewed from vertically, in the infrastructure layer, in the intermediate layer, and in the highest layer, data, services and applications, respectively, are available. The data layer consists of a content repository, which contains LOs or CPs. The service layer

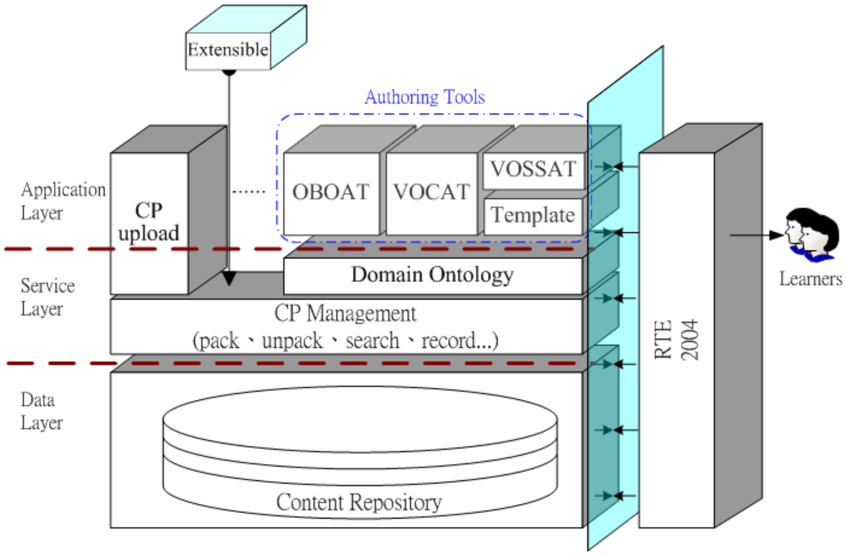


Fig. 1. Framework of the CRMS.

includes a variety of services, such as the domain ontology and CP management. The application layer provides Ontology-Based Outline Authoring Tools (OBOAT), the Visualized Online Course Authoring Tool (VOCAT) for course editing or aggregation, and the Visualized Online Simple Sequencing Authoring Tool (VOSSAT) for setting IMSSS rules.

At present, the IMS Simple Sequencing (SS) Specification in SCORM 2004 describes a method that can be used to represent the intended behavior of an authored learning experience such that any LMS will sequence discrete learning activities in a consistent way. The IMS SS is not labeled "simple" because the specification itself is simple, but rather because it defines a limited number of widely used sequencing behaviors [1]. Furthermore, the IMS SS recognizes only the role of the learner and does not define any sequencing capabilities that utilize or are dependent on other actors, such as instructors, mentors, or peers.

Sequencing and Navigation [20] in SCORM 2004, which adopts the Simple Sequencing Specification (SSS) of the IMS, relies on three concepts regarding learning activities, each of which may be described either as an instructional event, as events embedded in a content resource, or as an aggregation of activities to describe content resources with their contained instructional events [1]. The contents of the SN are organized into a hierarchical structure, that is, in the form of an activity tree (AT) in a learning map, as shown in Fig. 2. Each learning activity which includes one or more student activities has an associated set of sequencing behaviors, as specified in the Sequencing Definition Model (SDM), which is a set of attributes used by the SN.

The SN uses a specific set of data attributes which are associated with learning activities in the activity tree as the required sequencing behaviors in order to control the sequencing, selection, and delivery of activities to the learner. The sequencing behaviors describe how the activity or how the students' participation in the activity can be

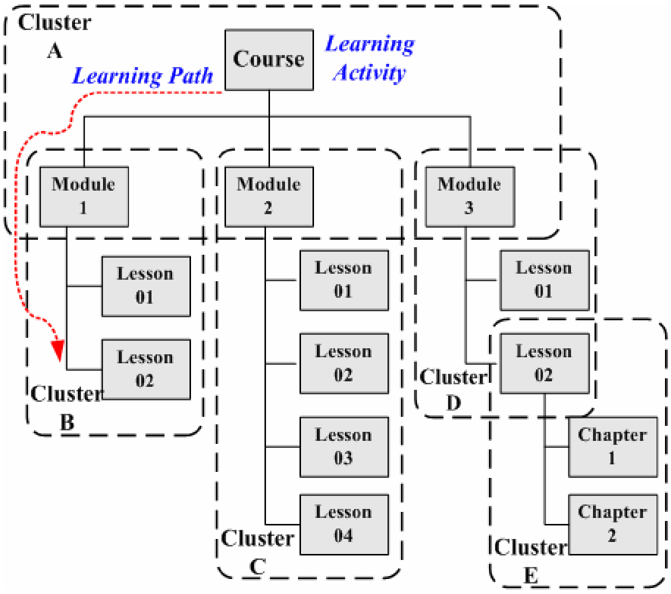


Fig. 2. Example of an activity tree (AT) with clusters.

modified to create the desired learning experience. The SN enables teachers to share not only the learning contents but also the intended learning experiences. It also provides a set of widely used sequencing methods so that teachers and instruction designers can arrange the sequencing efficiently. However, the best way to create, to represent, and to maintain an activity tree and the associated sequencing definition of the authoring tools is of prime concern.

The IMS SS Authoring tools generally organize LOs or SCOs into a hierarchy of courses, modules, lessons, presentations, etc., which are related to each other by means of a prerequisite, by another part, and by some other relationships. Each instructional unit typically has its own instructional objectives. In addition, some systems might even include SCOs that address misconceptions or provide remedial materials. The contents are usually stored on a web page, and the above tools can substantially help teachers and instruction designers effectively create courses for web-based learning.

Authoring tools for the IMS SS excel both at representing diverse teaching strategies using rule-based descriptions and encoding fine-grained strategies often used by teachers and instructional experts. These authoring tools tend to focus on the "macro" level of instruction, i.e. the sequencing of topics or modules. At the same time, they address the "micro" level of instruction, particularly as it concerns decision-making. This includes when and how to best give explanations, summaries, examples, and analogies; precisely what type of hinting and feedback to give; and what types of questions and exercises to provide students with. Somewhat ironically, authoring tools for the IMS SS vis-`a-vis teaching strategies have the most sophisticated sets of primitive tutorial operations. In addition, these tools have the capability of representing multiple tutoring strategies and "meta-strategies" that can facilitate the selection of the most appropriate tutoring strategy within a given context. Perhaps the greatest contribution of the IMS SS is that it offers teachers and instruction designers a simple interface, where they can fill in specific teaching templates simply by clicking a menu or options on the web, a technology described by educators since the advent of intelligent CAI in the early 1970s.

With the assistance of the IMS SS authoring tools, teachers and instruction designers can edit their own teaching materials, while learners can proceed to learn from the screen with minimal interactivity. Simply put, students can learn by reading and thinking rather than by merely doing since they must choose one of the previously established teaching strategies. Once students have made their choices, whether they are active or simply responding to interactive environments, owing to the availability and intelligent interjection of small grain-sized SCOs, such as explanations, multiple levels of hints, and analogies, teachers and instruction designers have the opportunity to be much more responsive.

2.2 Ontology and the RDF/RDFS

The Semantic Web aims to provide semantic mark-up for resources on the Web and to make these resources machine-readable and interpretable [6, 8]. When computers understand the semantics of data, the cost of searching and extracting information is considerably reduced, the precision of information-searching is greatly enhanced, and a multitude of potential applications of the Semantic Web are brought to the forefront. Ontologies can play the role of metadata and, as such, provide a set of well-defined vocabulary items to explicitly define all the data on the Web. Given the sharing of a common ontology, people and computers can precisely and effectively communicate with each other using the same syntax and semantics. This means that the success of the Semantic Web relies on whether ontologies can be readily agreed upon and quickly deployed. Many researchers [5, 9] assume that there will be plenty of small, distributed ontologies on the Semantic Web in the future, and that these ontologies will largely consist of pointers to each other, thus forming the backbone of the main ontology of the Semantic Web. Should this be the case, then authoring such a plethora of ontologies for various courses is bound to become an issue of major concern.

The term “ontology” was first used to describe the philosophical study of the nature and organization of reality. From the AI perspective, an ontology is an explicit specification of a conceptualization [5]. In this regard, a conceptualization is an abstract view of the world. Genesereth and Nilsson’s definition of a conceptualization is a tuple, <*D*, *R*>, where *D* is the domain of a discourse and *R* is a set of relations on *D*. An ontology associates vocabulary terms with entities identified in the conceptualization and provides definitions to constrain the interpretations of these terms.

Contrast this with the description offered by Guarino and Giaretta (1998) [7], who claimed that a conceptualization should actually be a triple structure, <*W*, *D*, *R*>, where W is a set of possible words; *D* is the domain of a discourse; and *W* is the max state set in *D*. For example, the domain *D* is (*S*1, *S*2, *S*3, *S*4, and *S*5), and these are divided into two piles, (*S*1, *S*2, and *S*3) and (*S*4 and *S*5). Each of the piles is a state. When we put *S*1 on *S*4, we obtain a new state. The collection of all states is called the max state set *W*. *R* is the full set of all conceptual relations in <*W*, *D*>, and *n* dimensional conceptual relations ρn is the full function: ρn: $W \rightarrow 2D^n$. Guarino refined this model and went on to define an ontology as described below.

An ontology is a logical theory which explains the intended meaning of a formal vocabulary item, i.e., its ontological commitment to a particular conceptualization of the world. The intended models of a logical language using such a vocabulary item are constrained by the ontological limitations of the item.

The architecture shown in Fig. 3 consists of three layers. The lowest layer can be viewed as a repository of raw data. The intermediate layer is a repository of organized data which are constructed from the raw data. In the highest layer, there are some applications which can extract the organized data from the intermediate layer or sometimes from the lowest layer. For example, a chapter or a page of a book can be considered the raw data in the lowest layer. Once that part of the book is read, the organized data therein are stored in the intermediate layer of the brain. Then, when information is retrieved from the brain, the information is obtained from the application layer. This concept is visually presented in Fig. 3. The ontology serves as the intermediate layer used for information- or knowledge-retrieval [14].

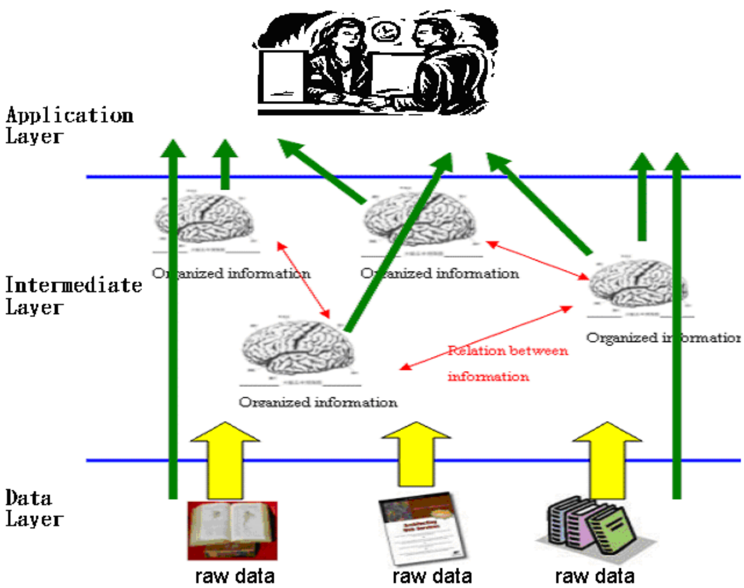


Fig. 3. Ontology representing the intermediate layer of the human brain.

The core technology involved in implementing a semi-dynamic ontology is the encoding of an LO embedded in compliance with the specifications of an XML-file. Also required are Resource Description Framework/Schema (RDF/RDFS) files for storing the relations among LOs that use standardized Java API programming. The benefits of this approach are twofold: the RDF files can be parsed as components of the framework for outlining purposes, while the JavaScript for Web-based programming offers a user-friendly interface on the user-side.

Reasoning using terms from the deployed ontologies is important when authoring a course outline in the ontology-design phase, as this makes it possible to detect logically inconsistent classes and to discover implicit subclass relations. This in turn calls for a more descriptive approach in the ontology-design phase and also requires that the reasoner infer part of the sub-sumption lattice. Reasoning is particularly beneficial when ontologies are large and are authored multiple times. In addition, it facilitates ontology-sharing, ontology-merging and ontology-integration. In other words, reasoning with terms from deployed ontologies is more powerful provided that the operations of the ontologies are combined in such Boolean operations as join (+), difference (-), or/and not (~).

The simplest framework of an outline structure is probably one that can be represented as a tree-like structure in a visualized form using an XML-file. More specifically, the nature of scientific concepts is embedded within logical and structural piles. Then, the domain-specific terms can be used as a device to filter out LOs from the content repository in the tree-like structure.

In this study, a group of experienced teachers and university math professors designed and constructed a mathematics-domain ontology. They then used this ontology to set up a course outline. Experienced and novice teachers could then add, delete, or restructure the course outline to suit their students' backgrounds or classroom practices at different school levels.

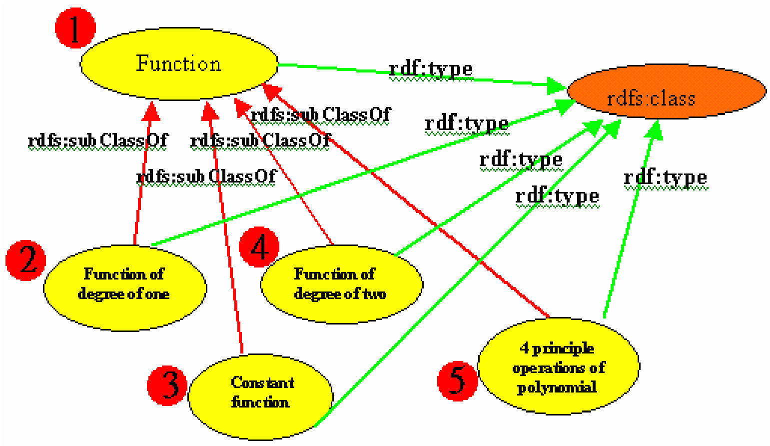
2.3 Example of a Domain Ontology with RDF/RDFS

To parse a domain ontology using RDF/RDFS, a general RDF-aware browser should be able to interpret the resulting course-item descriptions. From this point of view, representing of a mathematics course template consisting of subparts along with their own sets of descriptors is cumbersome. To obtain a domain ontology like that shown in Fig. 4, an indirect link from a course instance to a descriptor triple of the RDF representation is needed. The root-structure is the "Function." Also, the whole structure can be described in the RDF, as shown in Table 1. With ontological support, the contents of secondary school mathematics can be constructed in a similar format to that shown in Fig. 4. Within the root-level shown in Fig. 4 is the "Function" of secondary school mathematics courses in Taiwan. It is worth noting that RDFS allows for multiple inheritances [27].

We, therefore, refrain from using a "part-of" style of organization of descriptors. Instead, we define a meta-class descriptor, with descriptor groups representing subclasses. Subsequently, we define course slots as instances of the appropriate course-descriptor subclass. One of the reasons we prefer to adopt Protégé 2000 [19] as the RDFS editor is that it supports the treatment of instances as classes and vice versa. Martin (1997) [16] claimed that class/instance flexibility is a central requirement for adequate conceptual modeling. In addition to the course descriptors and their value sets, there is also a considerable amount of mathematical knowledge about the relationships among the descriptor values.

2.4 Visualized Online Authoring Tools

It is true that commercial multimedia authoring systems on desktop computers are somewhat superior in providing the instruction designer with the necessary tools to



❶ Function
❷ Function of the degree of one
❸ Constant function
❹ Function of the degree of two
❺ 4 principle operations of the polynomial

Fig. 4. Using an RDFS graph to present the relations of the “Function” ontology.

Table 1. Sample code of the RDF/RDFS.

```
<rdf:RDF
xmlns:rdf = ‘http://www.w3.org/1999/02/22-rdf-syntax-ns#’
xmlns:rdfs = ‘http://www.w3.org/2000/01/rdf-schema#’>
<rdfs:Class rdf:ID = “Function_of_degree_of_one”>    /*❷*/
<rdfs:subClassOf rdf:resource = “#Function”/>    /*❶*/
</rdfs:Class>
<rdfs:Class rdf:ID = “Constant_function”>    /*❸*/
<rdfs:subClassOf rdf:resource = “#Function”/>    /*❶*/
</rdfs:Class>
<rdfs:Class rdf:ID = “Function_of_degree_of_two”>    /*❹*/
<rdfs:subClassOf rdf:resource = “#Function”/>    /*❶*/
</rdfs:Class>
<rdfs:Class rdf:ID = “4_principle_operations_of_polynomial”>    /*❺*/
<rdfs:subClassOf rdf:resource = “#Function”/>    /*❶*/
</rdfs:Class>
```

produce visually appealing and highly interactive screens, but actually, behind the screens, lies only a shallow representation of the content and pedagogy [18]. Researchers have been investigating authoring tools since the early 1970s, but tools for authoring SCORM- Compliant course units are still lacking on the Web [21]. With this in mind, therefore, we proceeded to develop SCORM-compliant authoring tools for use on the Web. These authoring tools, coupled with a user-interface, allow non-programmers to formalize and visualize their course contents. Because of the advantages they offer with regard to elegance and cost effectiveness, these visualized online authoring tools are very appropriate for writing reusable software modules.

Unquestionably, visualized authoring tools for authoring LOs on the Web can help teachers and instruction designers visualize the relationships among SCO elements (such as topics, courses, concepts, and procedures) and obtain a bird's eye view of the subject-matter. In this way, course authors are able to prepare a course using their browsers instead of having to revert to an already-complete programming environment, such as Visual Basic. Authoring tools should, needless to say, allow for more free-formed representations. Therefore, the nodes should be layout in a hierarchical representation automatically although teachers and instruction designers will still have to position the nodes themselves for hierarchical representations on the Web.

Three issues generally arise when a new programming paradigm moves from the desktop to Web-based applications [26]. These are security-awareness, user scaling, and sharing among communities. As for the security issue, though it does not exist with desktop applications, it is present in a web-based programming language. With regard to scaling, this feature must be available within a Web-based environment although nobody can actually predict how many users might log in simultaneously. To cite one example, the number of ICQ users is highly related to its reliability. While a small number of users might easily be accommodated, this may not be the case with a large number of users. Finally, with regard to sharing among communities, two advantages of a Web-based environment are the development of user communities through BBS or online-chat rooms, and the provision of course units anytime and anywhere. In the former case, authors need to share their authoring experiences either synchronously or asynchronously. In the latter case, authors just need a browser, such as MS Internet Explorer, that can be used across time and location boundaries.

Fig. 5 illustrates a SCORM 2004-compliant authoring tool for implementation on a desktop platform [22]. It offers a Graphics User Interface (GUI) which teachers and instruction designers can use to author rule-based descriptions that comply with IMS SS [11]. While some teachers might have no interest in using it, many others at the secondary high school level might lack sufficient knowledge or skills to install or maintain a programming language in their own computers [22]. For this reason, we have developed VOSSAT with Java programming even though, at this time, the WWW is becoming more and more popular and is being widely adopted in the field of educational.

In this study, we also considered issues pertaining to security-awareness and scaling. More specifically, we developed two specific aspects of authoring tools: an innovative format of a visualized presentation and web-based as opposed to desktop applications.

3. METHODOLOGY

The Visualized Online Authoring Tool (VOAT) model, as illustrated in Fig. 6, consists of four stages that are employed in making courseware. As platforms for course delivery, we consider OBOAT, VOCAT, VOSSAT, and ADL RTE 2004. Beyond the

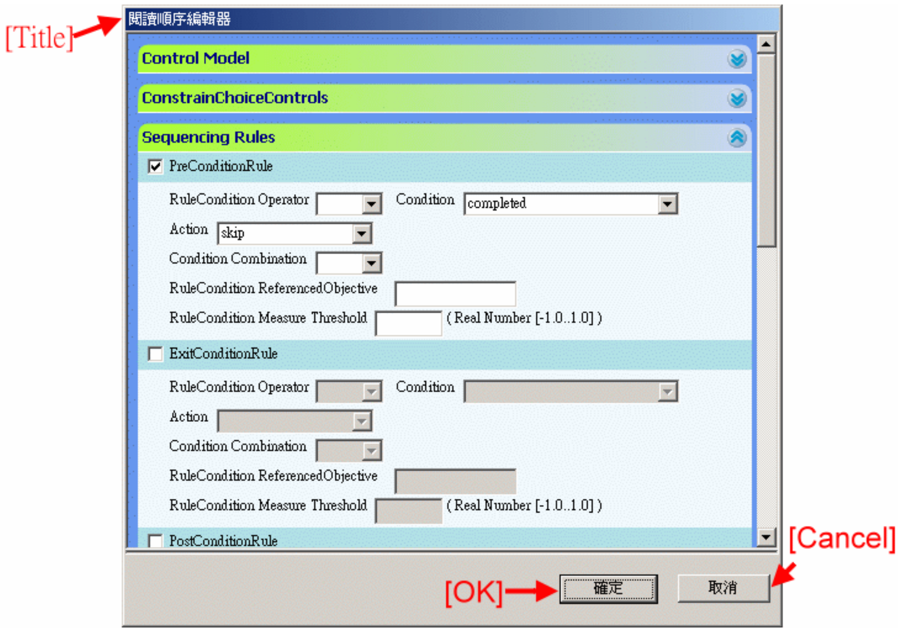


Fig. 5. Desktop for the IMS SS authoring tool designed by Shih (2004).

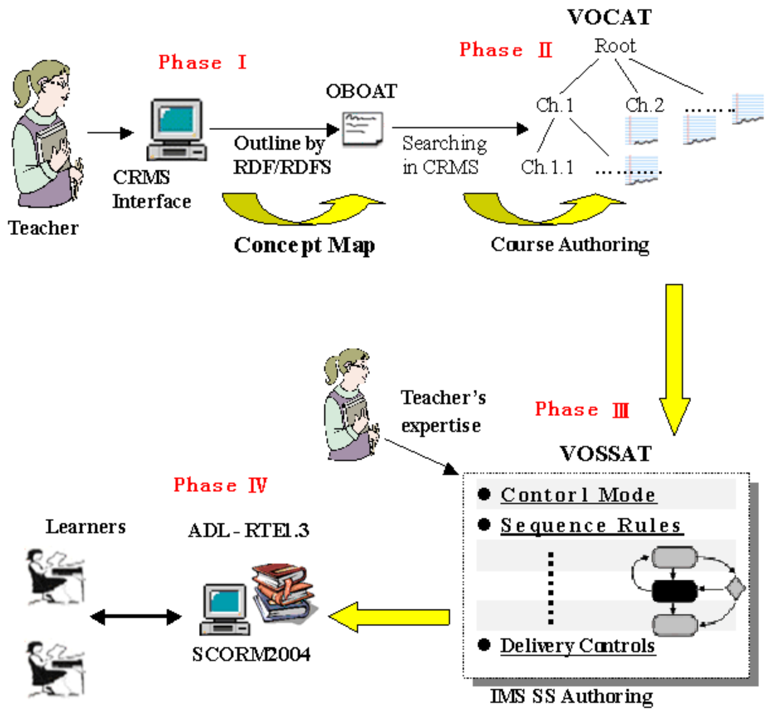


Fig. 6. VOAT: flow chart of the procedure for making courseware.

domain model, the factors associated with producing a course outline might include teacher beliefs, teacher preferences, the opinions of external experts, the characteristics of the students, and so on. A static outline editing environment, which mostly relies on the domain model, is far too often insufficient to motivate teachers to author their own courses. To overcome this barrier, we propose a semi-dynamic ontology, which can serve as the scaffolding. The advantage of this approach is that the integration of a course to suit a particular teaching context can be easily aggregated or assembled on the CRMS.

In Phase I, we provide OBOAT. This means teachers or instruction designers can construct an outline and be guided by the domain ontology. This is based on the premise that teachers have the best knowledge with which to choose detailed information for their own teaching context. They are also able to revise the outline since OBOAT offers a number of functions for assembling domain ontologies.

In Phase II, we provide VOCAT, which offers many visualized tools, like “add,” “delete,” and “re-shuffle” nodes, to change the order of LOs or CPs, thereby enabling teachers to assemble course units that accommodate different teaching contexts. In other words, teachers can choose to only revise or modify certain LOs instead of authoring from scratch.

In Phase III, we provide VOSSAT, which is designed for adaptive learning and makes rule-based descriptions available. With a user-friendly, graphic-user interface, VOSSAT allows teachers and instruction designers to select options by clicking on buttons or options in the window menu. This means that teachers and instruction designers do not need to have any knowledge of the IMS SS requirements. Our design for the GUI of VOSSAT, therefore, is suitable for many teachers and instruction designers. At the end of Phase III, they produce SCORM 2004-Compliant content packages by adding IMS SS tags to imsmanifest.xml.

In Phase IV, teachers and instruction designers deliver the final courses to ADL RTE 2004, provided that they have constructed the CPs as SCORM 2004-compliant units.

3.1 Tools

The major tools that we use are the following:

a) Resin: a web-server for JSP;

b) Jena 2.0: a RDF/RDFS tool designed for parsing a course outline;

c) Java script: a programming language used for user-interface design at the users’ end;

d) Protégé 2000: a tool used to create a mathematical ontology for junior high school levels. Fig. 7 shows a snapshot of the “Function” that is generated by Protégé 2000. Protégé 2000 offers an ontology editor which we can use to define classes, a class hierarchy, slots, slot-value restrictions, relationships between classes, and the properties of these relationships.

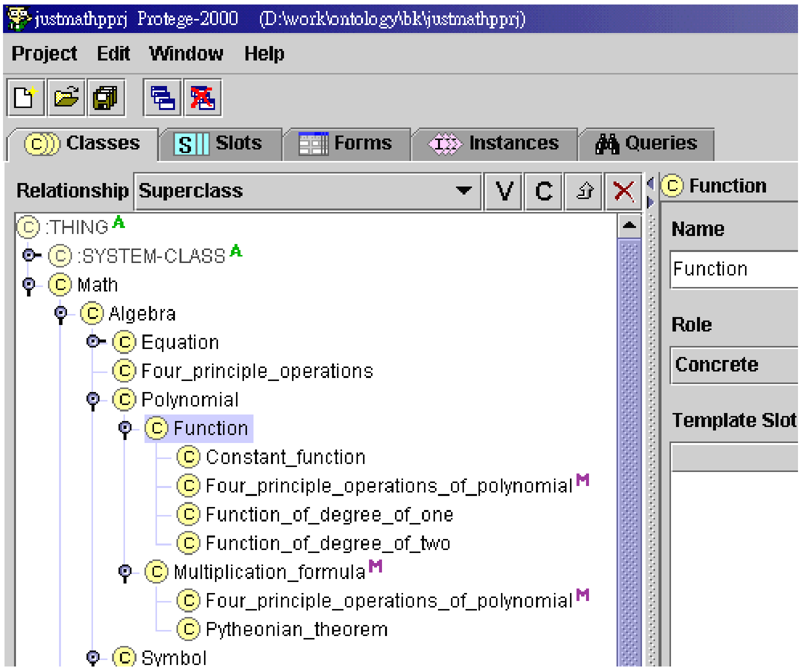


Fig. 7. Snapshot of a mathematics ontology for the junior high school level on Protégé 2000.

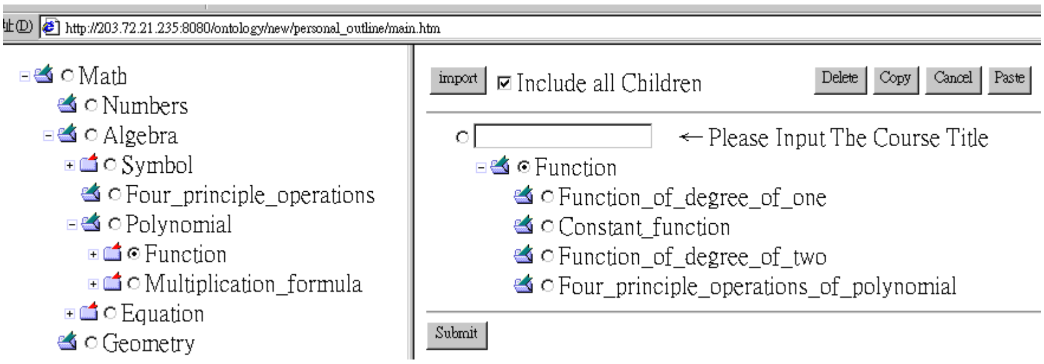


Fig. 8. Snapshot of the authoring "Function" on OBOAT.

4. IMPLEMENTATION AND RESULTS

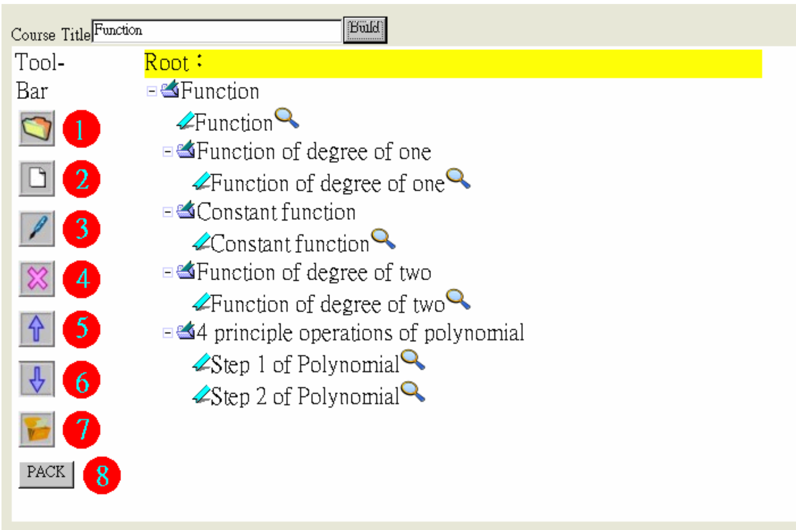
In the following, we will describe a series of scenarios that illustrate the process of modeling our newly-developed authoring tools.

We will begin with a mathematics teacher who intends to author a SCORM 2004-Compliant course unit for use at the secondary school level. The teacher can finish his/her task in three steps. The title in this scenario is "Introduction to the function." A dialogue is initiated to allow teachers and instruction designers to edit some of their preferred nodes, as shown in Fig. 8.

Scenario-1

It has been increasingly recognized that ontological principles can be effectively applied in the broader field of information systems [7], a phenomenon that has led to the notion of "ontology-driven information systems." Ontological abstractions of artifacts lead to improved communications through common vocabularies, enhanced reuse of LOs, and systems interoperability. In this study, we apply ontological support for the semi-dynamic construction of an ontology for course outlining. Compared with static ontologies that serve as scaffolding for course outlining and are currently available, OBOAT allows for the semi-dynamic construction of outlines, since most teachers and instruction designers have their own individual teaching contexts, which include teacher beliefs, teacher preferences, student characteristics, and so on.

As shown in Fig. 8, the framework has left and right sides. The former is used to access outlines from the Math domain ontology on the CRMS, whereas the latter provides a visualized space for re-structuring the outline of a course unit. So that teachers and instruction designers can design outlines visually, we provide six buttons, namely "Import," "remove," "copy," "cancel," "paste," and "submit." Fig. 8 shows the "Import" function node with all students starting in the authoring area. This then moves them to "Math" when the "Function" node at the top on the left-hand side is clicked.



❶ Creating a directory ❷ Adding a new LO
❸ Modifying an existing LO
❹ Deleting an existing LO
❺ Moving up an LO ❻ Moving down an LO
❼ Adding a new CP from "My package"
❽ Packing new teaching materials

Fig. 9. Snapshot of authoring a course on VOCAT.

Scenario-2

Similarly, VOCAT provides ontological support for course authoring. Once teachers finish the authoring process on OBOAT, they send the new outline to VOCAT, which is used to author a course. Fig. 9 shows the eight functions of VOCAT. By using VOCAT, teachers can author their courses within a shorter time period since the procedure for each action is very well designed. The final step in this scenario is packing all the LOs into a zip file.

Scenario-3

We designed VOSSAT so as to enable teachers and instruction designers to choose actions by clicking options or by completing certain texts, such as parameters or chapter names. Once they fill the IMS SS slots, they save the IMS SS tags in imsmanifest.xml automatically, as shown in Fig. 10. VOSSAT offers a GUI that can reduce the work that teachers and instruction designers have to do to set up teaching strategies for rule-based descriptions. This differs from the "Template" that has previously been proposed by Carnegie Mellon University [15]. Important here is that VOSSAT offers greater flexibility than does the "Template."

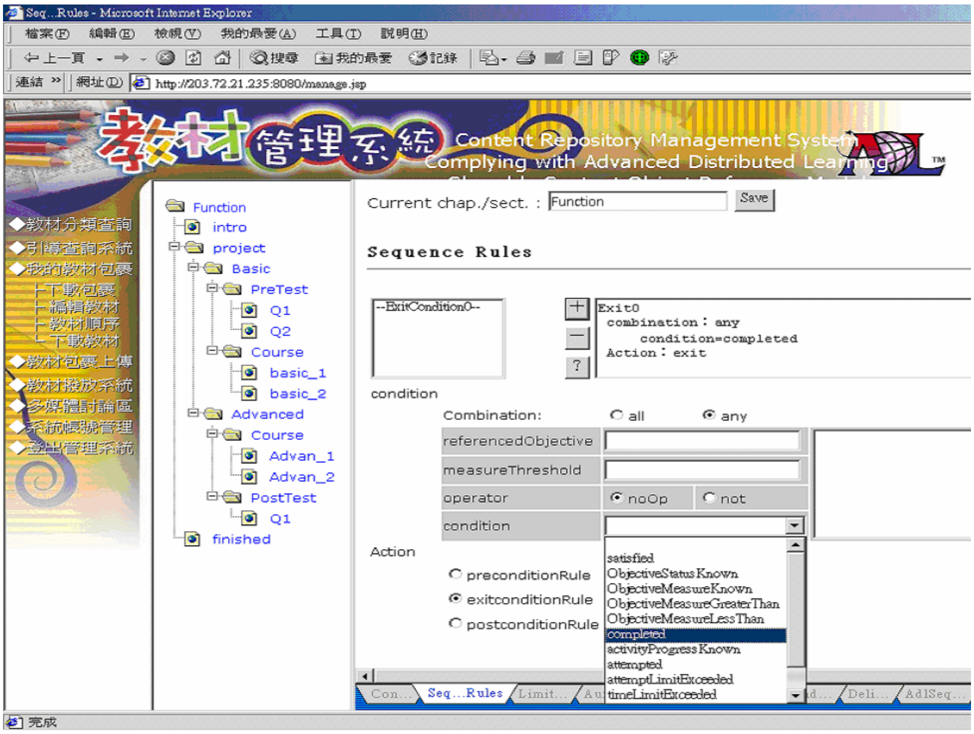


Fig. 10. Snapshot of delivering a course.

Scenario-4

The results that are generated in Scenario-3 make up a SCORM 2004-Compliant content package. Therefore, it can be delivered to ADL RTE 2004. This means that in the last scenario, students can browse the course based on the teaching strategies set up by the teachers and instruction designers. Fig. 11 shows a snapshot of a course on ADL RTE 2004.

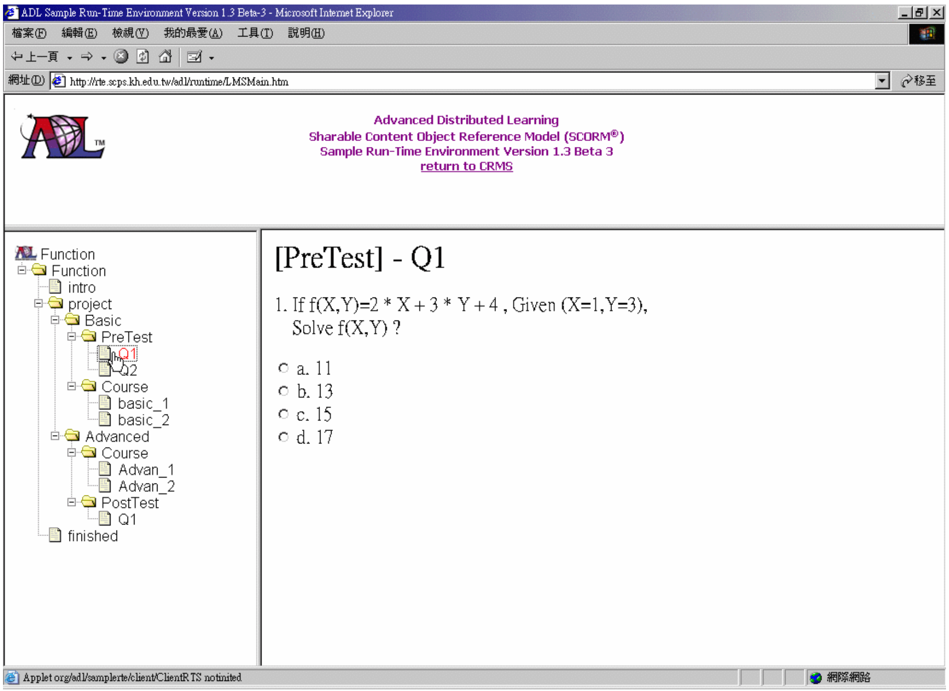


Fig. 11. Snapshot of browsing a course.

5. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE STUDY

This study is the first one to develop a systematic approach called the Visualized Online Authoring Tool Model (VOAT), with which teachers and instruction designers can construct SCORM 2004-compliant courses with sequencing behaviors in the application layer of the CRMS. To make this possible, we have developed three authoring tools, namely, the Ontology-Based Outline Authoring Tool (OBOAT), the Visualized Online Course Authoring Tool (VOCAT), and the Visualized Online Simple Sequencing Authoring Tool (VOSSAT). We have also used a series of scenarios to illustrate the processes involved in modeling newly-developed authoring tools. These applications have the capability of directly calling on the service layer to represent a specific course for the assembling or sequencing behaviors in a learning activity and the corresponding structures along with their associated sequencing rules in an activity tree. Thus, these content packages can be easily generated.

This study can be credited with making ontologies so that they are visualized, and it allows teachers to play the role of scaffolding guides. This is of immeasurable benefit to teachers who lack the skills needed to author SCOs or Content Packages. In this sense, the ontologies and Semantic Web offer teachers easier ways in which to use in SCOs or Content Packages besides merely searching keywords.

Equally important, based on the scenarios described above, teachers and instruction designers are able to easily create effective SCORM 2004-Compliant courses within a much shorter period of time in a CRMS environment. The newly-developed framework that we have proposed here consists of a set of systematic approaches to analyzing, organizing, and developing better, faster ways of authoring courses. Without this type of framework, it may very well be too difficult for teachers and instruction designers to integrate the SCORM 2004 into their teaching practices.

In future work, we intend to further explore the potential applications of ontologies and the Semantic Web, such as merging ontologies, differentiating between ontologies, and reasoning in a VOAT. We also plan to improve upon the VOAT model by enhancing its scalability and flexibility. In addition, we plan to develop a personalized learning course or applying educational theories to the VOAT model.

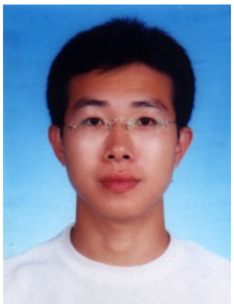
REFERENCES

1. ADL: Advanced Distributed Learning, Retrieved May 8, 2004, http://www.adlnet.org.
2. AICC: AICC – Aviation Industry CBT Committee, Retrieved June 11, 2004, http://www.aicc.org.
3. ARIADNE: Alliance for Remote Instructional and Authoring and Distribution Networks for Europe, *Foundation for the European Knowledge Pool*, Retrieved June 11, 2004, http://www.ariadne-eu.org.
4. S. Decker, *et al.*, "The semantic web on the respective roles of XML and RDF," *IEEE Internet Computing*, Vol. 4, 2000, pp. 63-74.
5. T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, Vol. 5, 1993, pp. 199-220.
6. T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International Journal of Human and Computer Studies*, Vol. 43, 1995, pp. 907-928.
7. N. Guarino, "Formal ontology and information systems," in Guarino, (ed.), in *Proceedings of 1st International Conference on Formal Ontologies in Information Systems* (*FOIS '98*), 1998, pp. 3-15.
8. C. W. Holsapple and K. D. Joshim, "A collaborative approach to ontology design," *Communications of the ACM*, Vol. 45, 2002, pp. 42-47.
9. M. Huhns and M. Singh, "Ontologies for agents," *IEEE Internet Computing*, Vol. 1, 1997, pp. 81-83.
10. LTSC: Learning Technology Standards Committee, *IEEE LTSC WG12*, Retrieved June 11, 2004, http://ltsc.ieee.org/wg12/.
11. IMS, Learning Resource Meta-data Specifications, Version 1.2, 2000, http://www.imsproject.org/metadata/index.html.

12. IMS, IMS Global Learning Consortium, Retrieved June11, http://www.imsproject.org/.
13. R. Kishore, H. Zhang, and R. Ramesh, "A helix-spindle model for ontological engineering," *Communications of the ACM*, Vol. 47, 2004, pp. 69-75.
14. C. M. Liu, "On the design of an ontology-based knowledge management system with a concept map," Unpublished Master thesis, Chung Cheng University, Chiayi County, 2003.
15. LSAL, "SCORM best practices guide for content developers," *Learning Systems Architecture Laboratory*, Carnegie Mellon LSAL, Retrieved June 11, 2004, http://www.lsal.cmu.edu/lsal/expertise/papers/index.html.
16. J. Martin, *Object-Oriented Methods – A Foundation*, UML ed., Prentice Hall, Upper Saddle River, New Jersey, 1997.
17. D. McGuinness and A. Borgida, "Explaining sub-sumption in description logics," in *Proceedings of International Joint Conference on Artificial Intelligence* (*IJCAI '95*), 1995, pp. 816-821.
18. T. Murray, "Authoring intelligent tutoring systems: an analysis of the state of the art," in *Proceedings of International Journal of Artificial Intelligence in Education* (*IJAIED*), Vol. 10, 1999, pp. 98-129.
19. N. F. Noy, R. W. Fergerson, and M. A. Musen, "The knowledge model of Protégé-2000: combining interoperability and flexibility," in *Proceedings of 12th European Workshop on Knowledge Acquisition, Modeling and Management* (*EKAW 2000*), LNCS 1937, 2000, pp.17-32.
20. SN, "Sharable content object reference model (SCORM) sequencing and navigation (SN) version 1.3," *Advanced Distributed Learning*, Retrieved June 11, 2004, http://www.adlnet.org/index.cfm?fuseaction=DownFile&libid=648&bc=false.
21. "Sharable content object reference model (SCORM)," *Advanced Distributed Learning*, Retrieved June 11, 2004, http://www.adlnet.org/.
22. T. Shih, "Developing SCORM- compliant software tools – from the perspective of an ADL academic co-lab partner," Taiwan, 2004.
23. W3C: World-? Wide Web Consortium, Retrieved June 11, 2004, http://www.w3.org.
24. D. A. Wiley, "Connecting learning objects to instructional design theory: a definition, a metaphor and a taxonomy," in D. Wiley, (ed.), *The Instructional Use of Learning Objects*, Bloomington: Association for Educational Communications and Technology, 2000.
25. XML: Extensible Mark-Up Language (XML), Retrieved June 11, 2004, http://www.w3c.org/xml/.
26. J. T. Yang and C. Y. Tsai, "An implementation of SCORM-compliant learning content management system – Content repository management system," in *Proceedings of 3rd IEEE International Conference on Advanced Learning Technologies* (*ICALT 2003*), 2003, pp. 453-457.
27. J. T. Yang, P. T. Yu, N. S. Chen, C. Y. Tsai, C. C. Lee, and T. K. Shih, "Using ontology as scaffolding for authoring teaching material – the case of secondary school mathematics courses," *Journal of Distance Education Technology*, Vol. 3, 2004, pp. 81-96.

Jin-Tan David Yang (楊錦潭) is Professor at National Kaohsing Normal University. He also holds ICALT 2005 as co-chair. He received his Ph.D. from University of Oregon in 1993. He has been involved in large-scale national research projects on SCORM 2004-compliant Learning Content Management System. He has published over 100 research papers in international refereed journals, conferences and books. He is also a senior member of Association of e-Learning, Taiwan Association (AEL) of Artificial Intelligence (TAAI), and IEEE.

Wen-Chih Chen (陳玟志) is a master student at National Kaohsiung Normal University (NKNU) in Taiwan. He is studying at Institute of Information, Computer and Education (ICE) at NKNU as well as working at Chien Chen Senior High School. In 1998, he attained bachelor diploma from college of education of National Taiwan Normal University. His research interests focus on e-learning content management system, ontology-based knowledge management, and data mining.



Chun-Yen Tsai (蔡俊彥) is a Ph.D. student at National Kaohsiung Normal University (NKNU) in Taiwan. His major is in science education at the Graduated Institute of Science Education (GISE). Before entering GISE at NKNU, he received the Bachelor of Education in Mathematics and Science Education at National Taichung Teachers College in Taiwan, and M.S. degree in Information and Computer Education at NKNU. His research interests aim at e-learning system, computer-assistant instruction, and the media assisted learning in science education.

Mei-Sheng Chao (趙美聲) is Professor of Education Department at National Taiwan Normal University. Previously, she was director of the Center of Teachers' Education at National Chung Cheng University in Taiwan. Her research interests focus on the development of a set of indicators concerning knowledge management capabilities in the web-based learning environments.