JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 20, 1057-1077 (2004)

Adaptive Learning Environment to Meet Pedagogical Needs*

JUN-MING SU, SHIAN-SHYONG TSENG+, CHING-TAI CHEN+ AND WEN-NUNG TSAI
Department of Computer Science and Information Engineering
+Department of Computer and Information Science
National Chiao Tung University
Hsinchu, 300 Taiwan
E-mail: {jmsu, tsaiwn}@csie.nctu.edu.tw
E-mail: {sstseng, caster}@cis.nctu.edu.tw

With the vigorous development of the Internet, e-learning systems have become more and more popular. Currently, to solve the issue of sharing and reusing teaching materials in different e-learning systems, several standard formats, including SCORM, IMS, LOM, AICC etc., have been proposed by international organizations. Among these international standards, the Sharable Content Object Reference Model (SCORM) has become the most popular standard in recent years. In the SCORM standard, Sequencing and Navigation (SN), which is based on the concepts of learning activities, defines the course sequencing behavior, which controls the sequencing, selecting and delivering of the course, and organizes the content into a hierarchical structure, namely, an Activity Tree (AT). For a large-scale learning activity, the Activity Tree will become too complex to manage and reuse, as will its sequencing behavior rules. Moreover, the lack of inter-relations among Activity Trees also makes reusing and integrating them hard. This implies that the scalability and flexibility of an adaptive learning system will be limited. Therefore, how to create, represent and maintain an AT and associated sequencing definitions is our concern. In addition, for a personalized learning environment, how to extend the structure of an AT with Pedagogical Theory has also become an important issue.

Therefore, in this paper, we extend and modularize the structure of an AT by applying Pedagogical Theory and the concept of the Object Oriented Methodology, respectively. Thus, we first propose a novel model, the Instructional Activity Model (IAM), which is composed of related AT nodes. Each AT node in IAM is modularized as a learning unit with inter-relations and specific attributes, which can be easily managed, reused, and integrated. We also propose an AT Selection algorithm with a pedagogical strategy for traversing IAM in order to generate dynamic learning content for the learner. IAM with its scalability and flexibility can apply appropriate pedagogical theories to meet specific needs by means of extension scheme. Finally, we propose a systematic approach to fast and easily construction of IAM using traditional course resources.

Keywords: adaptive learning, intelligent tutoring system, SCORM, activity tree, pedagogical theory

1. INTRODUCTION

With the vigorous development of the Internet, in the past ten years, e-learning systems have become more and more popular because they can enable learners to study at

Received March 30, 2004; accepted June 30, 2004.
Communicated by Han-Chieh Chao.
* This work was supported by the National Science Council of the Republic of China under contracts NSC 92-2524-S-009-001 and NSC 92-2524-S-009-002.

any time and any location. However, because the teaching materials in different e-learning systems are usually defined by specific data formats, the sharing of teaching materials among these systems is difficult, making the creation of teaching materials expensive. To solve the issue of uniformizing the teaching materials format, several standard formats, including SCORM [1], IMS [2], LOM [3], AICC [4] etc., have been proposed by international organizations. Based upon these standard formats, the teaching materials in different learning management systems can be shared, reused, and recombined. Among these international standards, the Sharable Content Object Reference Model (SCORM), which integrates IMS, LOM, and AICC, has become the most popular international standard in recent years. Based on the concept of learning objects, SCORM uses metadata to specify the structure of every learning object and proposes a content aggregation scheme to package these objects using the Extensible Markup Language (XML) [5, 6] format.

At present, the Sequencing and Navigation (SN) provided by SCORM 1.3, which adopts the Simple Sequence Specification (SSS) of IMS [2], defines the course sequencing behavior and Content in SN is organized into a hierarchical structure, namely, an Activity Tree (AT). SN is based on the concepts of learning activities, each of which may be described as an instructional event, as an event embedded in a content resource. SN uses information about the desired sequencing behavior to control the sequencing, selection and delivery of activities to the learner. Therefore, by using this standard, we can develop an intelligent approach to (semi-)automatic course or exercise sequencing.

Because SN places no restrictions on the structure, organization, or instruction of the Activity Tree, the tree and the associated sequencing definitions may be static or dynamically created. For a large-scale learning activity, the Activity Tree will become too complex to manage and reuse, as will its sequencing behavior rules. Moreover, the lack of inter-relations among Activity Trees also makes reusing and the integrating them hard. This implies that the scalability and flexibility of an adaptive learning system will be limited. Therefore, how to create, represent and maintain the Activity Tree and associated sequencing definitions is our concern. Existing intelligent tutoring systems often use the grade and learning duration of the learner to evaluate only the learning result and to decide the delivery sequence of the learning content. However, these data may be not sufficient to evaluate the personal learning behavior. Therefore, many researches have used Pedagogical Theory [7-10] to enhance the evaluation of the personal learning characteristic and to generate personalized learning guidance.

Hence, in this paper, the learning characteristics of learners will be taken into consideration. Our approach extends and modularizes the structure of an AT by means of Pedagogical Theory and the concept of the Object Oriented Methodology in order to construct an adaptive learning activity in a personalized learning environment; i.e., one large AT is modularized into several suitable AT nodes which possess several specific attributes and associated inter-relations. By means of these attributes and inter-relations, each AT node can be reused and reintegrated to generate a new organization for a course in order to decrease the cost of designing a learning activity and increase the scalability and flexibility. To achieve specific purposes, e.g., to meet pedagogical needs, the attributes and interrelations of an AT can also be extended. Therefore, we propose a novel model, the Instructional Activity Model (IAM), which is composed of related Activity Tree nodes. Based upon Pedagogical Theory, each AT node in IAM is defined as a learning unit with inter-relations among AT nodes and specific attributes by means of which the IAM can be easily managed, reused, and integrated. In addition, we also propose an AT Selection algorithm, which employs a pedagogical strategy to traverse IAM in order to generate dynamic learning content for the learner. IAM, with its scalability and flexibility can apply different pedagogical theories to meet specific needs by means of scheme extension. Finally, we propose a systematic approach to quickly and easily construct IAM using traditional course resources.

The main contributions of this paper are:

1. We propose a general purposed model, called the Instructional Activity Model (IAM), to generate an adaptive learning course which is compatible with the SCORM standard.
2. We modularize a large Activity Tree into several suitable AT nodes with specific attributes and inter-relations, which can be easily managed, reused, and integrated, based on OO methodologies.
3. We apply Pedagogical Theory in IAM to define the personal learning characteristic and evaluate the learning result of a learner to generate more adaptive learning.
4. We propose a systematic approach to constructing IAM using traditional course resources.

2. RELATED WORK

In this section, we review the SCORM standard and some related works.

2.1 SCORM (Sharable Content Object Reference Model) [1]

Among the existing standards for learning contents, SCORM is currently the most popular one. It is a product of the U.S. government's initiative in Advanced Distributed Learning (ADL). In November of 1997, the Department of Defense and the White House Office of Science and Technology Policy launched the ADL initiative with the goal of providing access to high-quality education and training materials that are easily tailored to individual learner needs and available whenever and wherever they are needed. The SCORM specifications are a composite of several specifications developed by international standards organizations, including the IEEE [3], IMS [2], AICC [4], and ARIADNE [11].

In a nutshell, SCORM is a set of specifications for developing, packaging and delivering high-quality education and training materials whenever and wherever they are needed. SCORM-compliant courses leverage course development investments by ensuring that compliant courses are "RAID": reusable, meaning that they can be easily modified and used by different development tools; accessible, means that they can be searched and made available as needed by both learners and content developers; interoperable, meaning that they operate across a wide variety of hardware, operating systems and web browsers; and durable, meaning that they do not require significant modifications with new versions of system software [12].

2.2 Sequencing and Navigation (SN) [1, 2]

Sequencing and Navigation (SN) in SCORM 1.3 (also called SCORM 2004) adopts the Simple Sequencing Specification of IMS [2] based on the concepts of learning activities, each of which may be described as an instructional event, as an event embedded in a content resource. The content in SN is organized into a hierarchical structure, namely, an activity tree (AT) as a learning map. An example of an AT is shown in Fig. 1. Each learning activity, including one or more child activities, has an associated set of sequencing behaviors, defined by the Sequencing Definition Model, which is a set of attributes used by SN. SN uses information, that is, a specific set of data attributes which are associated with learning activities in the activity tree, about the desired sequencing behavior to control the sequencing, selection, and delivery of activities to the learner.

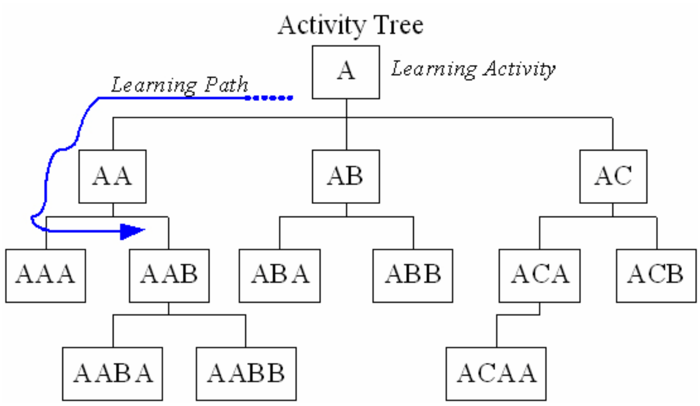

Fig. 1. An example of activity tree.

The sequencing behaviors describe how the activity or how the children of the activity are used to create the desired learning experience. SN places no restrictions on the structure, organization, or instruction of the activity tree. The tree and the associated sequencing definitions may be static or dynamically created. Therefore, how to create, represent, and maintain the activity tree and associated sequencing definition, which is not specified, is an important issue. SN enables us to share not only learning contents but also intended learning experiences. It also provides a set of widely used sequencing methods so that the teacher could do sequencing efficiently. However, the definition of sequencing behavior rules is obviously too simple to satisfy pedagogical needs.

2.3 Other Related Research

Carchiolo *et al*. [13] have proposed adaptive formative paths for e-learning environments. They construct a domain database and student profiles to obtain personalized learning paths. During the learning process, the learning paths can be dynamically modified according to student needs and capabilities. Although this system has some advantages, including consideration of each student's prior knowledge and generation of an adaptive learning path, it dose not take pedagogical theory into account, and it is not yet compatible with the SCORM standard.

Sheremetov and Arenas [14] also proposed a system, called EVA, for developing a virtual learning space at the National Technical Institute in Mexico. EVA consists of five virtual learning spaces: 1. the Knowledge Space, in which all necessary information to exists; 2. the Collaborative Space, in which real or virtual companions get together to learn; 3. the Collaborative Space, in which the teachers or tutors (also real or virtual) guide learning and provide consultation; 4. the Experimentation Space, in which the practical work is done by the students in the virtual environment; and 5. the Personal Space, in which records of users are stored. The model of knowledge is represented in the form of graph, where each node, the basic element of the knowledge structure, is a unit of learning material (ULM). ULMs with a related knowledge concept can be grouped into a POLIlibro (or Multi-Book) along the learning trajectory (path), depending on the students. However, the relations between ULMs are not sufficient to express the structure of the knowledge model, and the attributes of an ULM are sufficient for mining the behaviors of students. The authors also proposed some methods for planning trajectories and scheduling learning activities based on the agent technology. However, how a learning path can be generated was not discussed.

3. INSTRUCTIONAL ACTIVITY MODEL (IAM)

As mentioned above, the SCORM standard defines a hierarchical structure, namely, an Activity Tree (AT), used to sequence the delivery of learning content to the learner. By defining the sequencing behavior rules within an AT, we can develop an intelligent approach to (semi-)automatic course and exercise sequencing. Therefore, how to create, represent, and maintain the Activity Tree and associated sequencing definition is our concern. For a large-scale learning activity, the Activity Tree will become too complex to be managed and reused. Besides, it is hard to reuse and integrate ATs without knowing the inter-relations among ATs. This implies that the scalability and flexibility of an adaptive learning system will be limited. Moreover, for modern personalized learning, many researches have used Pedagogical Theory [7-10] to enhance the evaluation of the personal learning characteristic.

Hence, in this paper, we will first define the interrelation attributes of an AT, e.g., capability, weight, etc. Then, we will extend and modularize the structure of AT by means of Pedagogical Theory and the concept of the Object Oriented Methodology, respectively. As shown on the right side of Fig. 2, a large AT is divided into three small AT nodes with interrelation attributes. Therefore, by means of the interrelation attributes, the small AT nodes can be integrated and further connected with other AT nodes; e.g., AT_1 connects AT_4 and AT_5. Thus, we propose a novel model, the Instructional Activity Model (IAM), which is composed of related Activity Tree nodes. Based upon Pedagogical Theory, each AT node in IAM is modularized as a learning unit with inter-relations and specific attributes, which can be easily managed, reused, and integrated. We also propose an AT Selection algorithm with a pedagogical strategy used to traverse IAM in

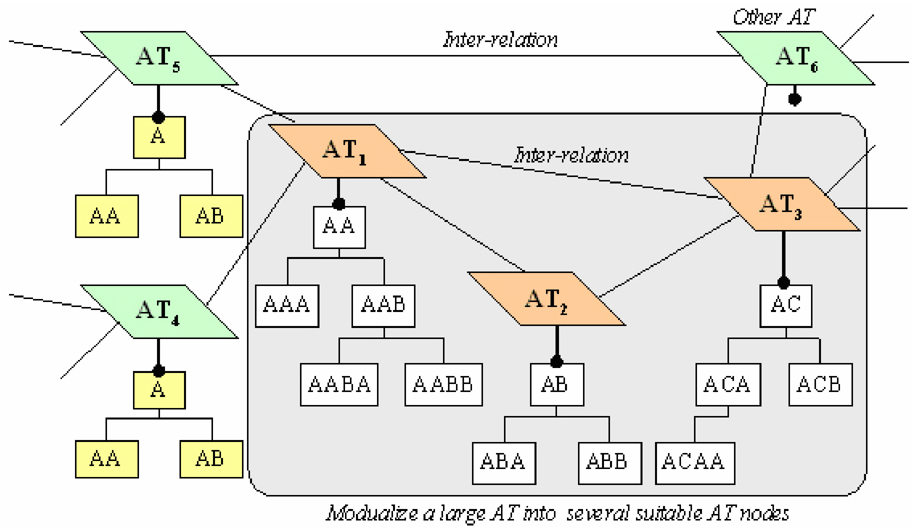

Fig. 2. The concept of modularizing an AT.

order to generate dynamic learning content for the learner. In this section, we will describe the Instructional Activity Model, including its properties, and the AT Selection algorithm.

3.1 Instructional Activity Model

In Sequencing and Navigation (SN), we can create an AT on the fly. As mentioned above, in a large AT, its organization and sequencing rules are hard to manage and reuse. However, a large number of ATs will also make the management of AT nodes and rules complicated. Therefore, to strengthen the scalability and flexibility of AT, we must define a suitable unit of an AT. According to Bloom's Mastery Theory [15], a suitable unit of learning content is a chapter or a section for learning. Thus, in IAM, we define the unit of an AT as a chapter or a section.

Assume there are n ATs. We define an AT set as $\mathbf{AT_{set}} = \{AT_1, AT_2, \ldots, AT_n\}$. According to the formulation of Gagne [16], "*A capability is a knowledge unit stored in a person's long term memory that allows him/her to succeed in the realization of physical, intellectual or professional activity*." Suppose there are m capabilities; we can obtain $\mathbf{C_{set}} = \{c_1, c_2, \ldots, c_m\}$. Before learning an AT, students are supposed to possess some capabilities, called ***Prerequisites***. Similarly, after learning an activity tree, students can acquire further capabilities, called ***Contributions***. Every ***prerequisite*** or ***contribution*** has its own weight representing the significance of learning capabilities before and after learning. Therefore, in IAM, the $\mathbf{C_{set}}$ can be regarded as the union of all ***prerequisites*** and ***contributions***, and an AT, thus, has several capabilities.

A learning activity or a course is composed of several ATs with input/output capabilities. The student learns a suitable AT and gains further capabilities, which enable the student to learn another advanced AT. This learning process is repeated until the student has finished all the learning objectives predefined by teachers. Then, every student will have an individual value of C_{set}. Fig. 3 shows a diagram of IAM.

In Table 1, we define the related attributes of interrelations, measure functions, AT selecting criteria, etc. in IAM as shown in Fig. 3.

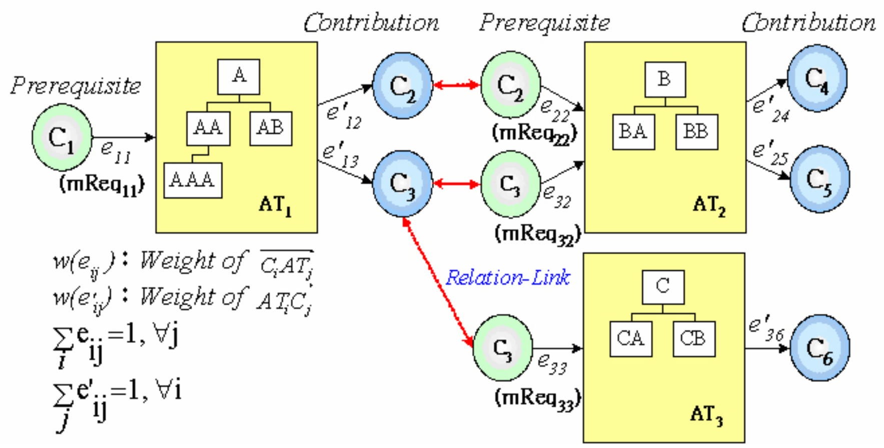

Fig. 3. A diagram of IAM.

Table 1. The definitions of related symbols in IAM.

Symbols	Description
e_{ij}	The edge from c_i to AT_j, called the "*prerequisite edge*," means that before learning AT_j, the student is supposed to possess ability c_i.
e'_{ij}	The edge from AT_i to c_j, called the "*contribution edge*," means that after learning AT_i, the student will gain ability c_j.
$w(e_{ij})$	The weight of e_{ij} denotes the significance of c_i before learning AT_j, where the sum of $w(e_{ij})$ of an AT is 1, i.e., $\sum_i w(e_{ij}) = 1, \forall j$.
$w(e'_{ij})$	The weight of e'_{ij} denotes the significance of c_j after learning AT_i, where the sum of $w(e'_{ij})$ of an AT is 1, i.e., $\sum_j w(e'_{ij}) = 1, \forall i$.
$mReq_{ij}$	The minimum requirement of c_i for learning AT_j is used to determine whether the student is qualified to learn AT_j or not.
$grade(e'_{ij})$	The learning grade after learning AT_i.
$val(c_m)$	The evaluation function of a capability, i.e., $val(c_m) = \frac{\sum_j w(e'_{jm}) \times grade(e'_{jm})}{\sum_j w(e'_{jm})}$.

Table 1. (Cont'd) The definitions of related symbols in IAM.

The Related Measure Functions of AT	
Acquired Capability (AC)	This records the student's learning results; AC = $\cup(c_i, val(c_i))$.
Course Objectives (CO)	This records the student's learning objectives; CO = $\cup(c_i)$.
Potential Capability List (PCL)	Each AT has a PCL recording all the contribution capabilities which can be reached from this AT via edges in IAM. It can be formulated as $PCL_{ATk} = \cup(c_i)$, where c_i can be reached from AT_k by means of connecting edges; e.g., in Fig. 2, PCL_{AT1} equals $\{c_2, c_3, c_4, c_5, c_6\}$.
Student's Grade Prediction (SGP)	SGP denotes the performance prediction of the specific student related to the AT, i.e., $SGP_k = \sum_i (c_i \times w(e_{ik}))$.
Normalized Objective Weight (NOW)	NOW denotes the relationship between an AT and the student's CO. A higher objective weight implies better learning efficiency. Empirically, the selecting function tends to select an AT with a higher SGP and higher NOW for students; $\text{NOW} = \frac{\text{the number of } c_i (c_i \in PCL_{ATj} \ \& \ c_i \in CO)}{\text{the number of } c_i (c_i \in PCL_{ATj})}$.
Chosen Factor (CF)	CF, a linear combination of selection criteria, is used to select a suitable AT for the learner. For example, for AT_i, $CF_i = \alpha NOW_i + \beta SGP_i$, where $\alpha + \beta = 1$, $0 \le \alpha, \beta \le 1$.

In brief, the Instructional activity model (IAM), a graphical representation of a learning activity or course, contains a set of **AT**s; **Capabilities**, including *prerequisites* and *contributions*; a set of **Relations Edges**, including e_{ij} with $mReq_{ij}$ and e'_{ij}; and a set of **Measure Functions**. Assume IAM has *n* ATs and *m* capabilities. Then, it can be formulated as a quadruple, **IAM** = ($\mathbf{AT_{set}}$, $\mathbf{C_{set}}$, $\mathbf{E_{set}}$, $\mathbf{E'_{set}}$),
where

- $\mathbf{AT_{set}} = \{AT_1, AT_2, \ldots, AT_n\}$.
- $\mathbf{C_{set}} = \{c_1, c_2, \ldots, c_m\}$.
- $\mathbf{E_{set}} = \cup(E_j)$, where $\mathbf{E_j} = \cup_i (e_{ij}, mReq_{ij})$, $e_{ij} \in AT_j$.
- $\mathbf{E_{set}}$ is the set of all prerequisite edges with minimum requirement values in an IAM.
- $\mathbf{E'_{set}} = \cup(E'_j)$, where $\mathbf{E'_j} = \cup_j (e'_{jk})$, $e'_{jk} \in AT_j$.
- $\mathbf{E'_{set}}$ is the set of all contribution edges in an IAM.

3.2 Basic Functionalities

Based upon the structure of IAM described above, we can develop several approaches to provide students with a learning environment for a dynamic and adaptive course. The learning process can be simply considered as the sequencing of activity trees in IAM in order to enable students to satisfy the learning objectives. The flowchart and algorithm of AT Selection is shown in Fig. 4 and Algorithm 1, respectively.

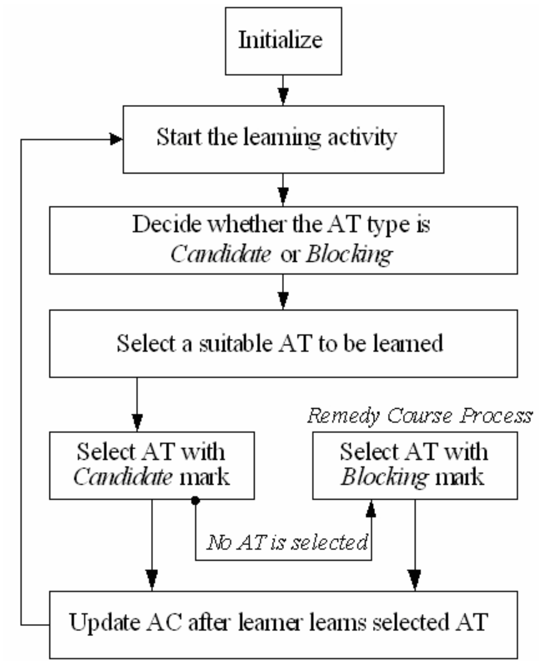

Fig. 4. The flowchart of the AT selection process.

Here, we will explain the AT Selection algorithm of IAM. First, we initialize the learning status by loading AC and CO, evaluate the PCL_{AT} of every AT **(Setp1)**, and then enter the loop of the learning activity **(Step2)**. During the AT selection process, we mark each AT with C*andidate* or *Blocking* after comparing the $mReq(e_{ij})$ with $val(c_i)$ **(Step2.1)**. *Candidate* indicates that this AT will be selected later, and *Blocking* indicates the opposite. Before delivering AT to the learner, we have to execute the selection process to choose a suitable AT. In general, we only use the **CF** value to choose one suitable AT **(Step 2.2.2)**. However, to meet specific needs, e.g., to apply Pedagogical Theory, we can define other selection criteria and a strategy in the *extended selection scheme*, which will be described later in section 4 **(Step 2.2.1)**. After completing the AT selection process, we choose a suitable AT marked *candidate* and deliver it to the learner **(Step2.2)**. However, if no AT marked *candidate* exists, the AT selection process proceeds to the **Remedy Course Process (Step2.2.3-Step 2.2.7)**. In the Remedy Course Process, we select an AT with the largest value of $c_m \in CO$ **(Step2.2.3-Step2.2.4)** and then find a c_i with the *smallest*, *largest*, or *medium* value of $(mReq(e_{ij}) - val(c_i))$, according to the type of ***SelectingPolicy*** **(Step2.2.5)**. In this algorithm, we use three policies to select different capabilities for adaptive learning. The policy "*Easiest First*" tends to select a c_i in which the learner has earned a high grade, but the policy "*Hardest First*" dose the opposite.

After selecting a c_i, we can decide which AT connected with c_i to deliver to the learner by computing **MAX**((mReq(e_{ij}) − grade(e'_{ki})) × w(e'_{ki})), which implies that the progress of the learner is the largest **(Step2.2.6)**. Fig. 5 shows in detail the Remedy Course Process. Finally, when the learner has finished and satisfied all the course objectives (CO), the AT selection process stops.

Algorithm 1. AT Selection Algorithm
Input: **IAM, AC** and **CO** of the learner, and ***SelectingPolicy*** = {*Easiest First*, *Medium First*, *Hardest First*}.
Output: the new **AC** after the learner has finished the learning activity.
Step1: Evaluate the PCL_{AT} of every AT in IAM.
Step2: while(CO ⊄ AC) *//start the learning activity*
// decide whether the type of AT is Candidate or Blocking
2.1: for each c_i with e_{ij} in AC
{if (mReq(e_{ij}) > val(c_i))
then mark the AT_j with *Blocking*
else if (AT_j has not been learned yet)
then (compute CF_j)and (mark the AT_j with *Candidate*)**}**
//select a suitable AT to be learned
2.2: if (∃AT with *Candidate* mark) *// select the AT with Candidate mark*
then
2.2.1: if ∃ *extended selection scheme* of AT **then** do it. *// for specific needs*
2.2.2: Select an AT with the highest CF and deliver it to the learner.
else if (∃AT with *Blocking* mark)
then *//go to Remedy Course Process & select a suitable AT*
2.2.3: for each AT_j marked *Blocking*
{Count the amount of $c_m \in$ CO which is connected by e'_{jm}.}
2.2.4: Select the AT_j with the largest value of $c_m \in$ CO.
2.2.5: for all c_i with e_{ij}
{if *SelectionPolicy* = "*Easiest First*," "*Medium First*" or "*Hardest First*"
then Find the c_i with the *smallest*, *medium*, *largest* value of (mReq(e_{ij}) – val(c_i)), respectively.**}**
2.2.6: for all $e'_{ki} \in E_i$ in c_i,
Select the AT_k with **MAX**((mReq(e_{ij}) – grade(e'_{ki})) × w(e'_{ki})).
2.2.7: Clear the mark of AT_j and deliver AT_k to the learner.
2.3: if the learner passes the selected AT
then mark this AT with *Learned*.
2.4: update AC after the learner learns the selected AT.
Step3: return a new AC.

Example 1: This IAM shown in Fig. 6 can be represented as follows:
IAM = ({AT_1, AT_2 AT_3, AT_4, AT_5,}, {c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9}, {(e_{11}, 0.8), (e_{22}, 0.7), (e_{23}, 0.8), (e_{33}, 0.8) , (e_{44}, 0.8) , (e_{55}, 0.8) , (e_{65}, 0.6)}, {e'_{14}, e'_{15}, e'_{25}, e'_{36}, e'_{47}, e'_{48}, e'_{58}, e'_{59}}).

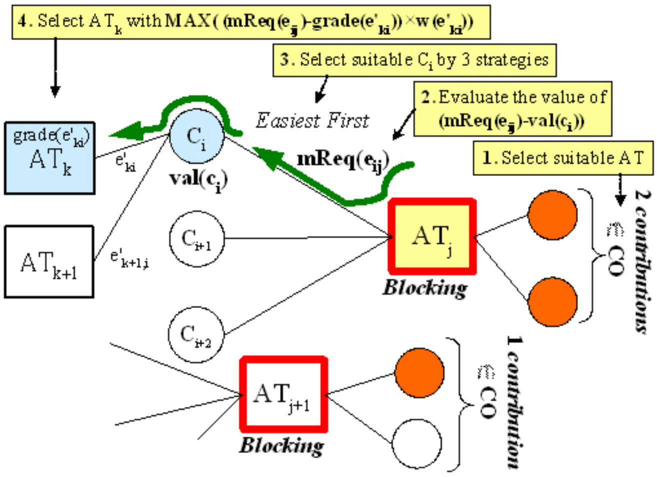

Fig. 5. A diagram of the remedy course process.

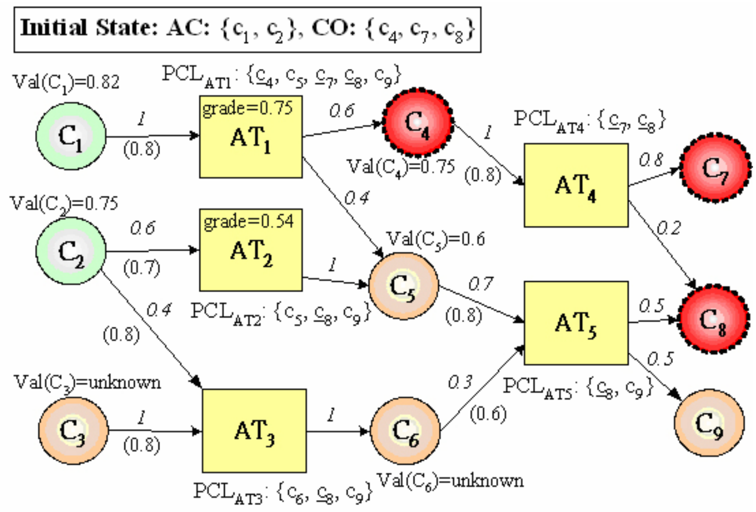

Fig. 6. An example of IAM.

Case1. We assume that **AC** = {(c_1, 0.82), (c_2, 0.75)} and **CO** = {c_4, c_7, c_8}. *Note that the value in parentheses is the val(c_i).*

The PCL_{AT} has been evaluated as shown in Fig. 6. After the first iteration of the **While** loop of Algorithm 1, we can get results as shown in Table 2. Thus, AT_1 will be delivered to the learner because it has the highest CF value.

Case2. We assume that AC = {(c_1, 0.82), (c_2, 0.75), (c_4, 0.75), (c_5, 0.6), (c_6, unknown)}, CO = {c_4, c_7, c_8}, and Blocking AT = {AT_3, AT_5}. The AT selection process has moved on to the **Remedy Course Process**.

Table 2. The related values of AT_1 and AT_2.

	SGP	NOW	CF
AT_1	$SGP_1 = val(c_1) \times w(e_{11}) = 0.82 \times 1 = 0.82$	$Now_1 = \frac{\text{the number of } \{c_4, c_7, c_8\}}{\text{the number of } \{c_4, c_5, c_7, c_8, c_9\}} = \frac{3}{5} = 0.6$	$CF_1 = \alpha \times SGP_1 + \beta \times NOW_1 = 0.5 \times 0.82 + 0.5 \times 0.6 = 0.71$
AT_2	$SGP_2 = 0.45$	$NOW_2 = 0.33$	$CF_2 = 0.39$

Before **Step 2.2.5**, because AT_5 has one $c_m \in$ CO, AT_5 is selected. If the **Selection-Policy** is "*Easiest First*," the c_5 with the smallest value, 0.2, of $(mReq(e_{55}) - val(c_5))$ is selected. Then, by computing $(mReq(e_{55}) - grade(e'_{15})) \times w(e'_{15}))$ and $(mReq(e_{25}) - grade(e'_{25})) \times w(e'_{25}))$, we can decide to deliver the AT_2 with a value of 0.26 to the learner.

4. APPLYING PEDAGOGICAL THEORIES IN IAM

As mentioned above, the Instructional Activity Model (IAM), which is composed of related AT nodes with inter-relations and specific attributes, can be easily managed, reused, and integrated. Our proposed AT Selection Algorithm can then generate the dynamic learning content for the learner by traversing the IAM. In addition, due to strengthened the scalability and flexibility of IAM, appropriate pedagogical theories can be selected and applied to provide personalized learning guidance according to extension schemes for specific needs. Therefore, in this section, we will show how well-known pedagogical theories can be applied in IAM by means of extension schemes.

4.1 Extension Scheme of IAM

We can consider three aspects of pedagogical theories: 1. the **Capability Taxonomy**, 2. the **Learning Style**, and 3. the **Organization of Teaching Material**. We can describe these three aspects as follows.

• **Capability Taxonomy:** By learning different Learning content, the learner will acquire different knowledge or capabilities. Thus, Gagne [7] considered that the learning outcomes of learners can be classified into five types: **Verbal Information**, **Intellectual Skills**, **Cognitive Strategies**, **Motor Skills**, and **Attitude**. Accordingly, we can categorize the learning capabilities in IAM into five types and define each c_i in $C_{set} = \{c_1, c_2, \ldots c_m\}$ as having five dimensions: $<vc_i, ic_i, cc_i, mc_i, ac_i>$, where vc_i denotes **v**erbal **c**apability, ic_i denotes **i**ntellectual **c**apability, cc_i denotes **c**ognitive **c**apability, mc_i denotes **m**otor **c**apability, and ac_i denotes **a**ttitude **c**apability.

• **Learning Style:** The learner's learning style is the way s/he prefers to learn. Therefore, learners have individual learning preferences during learning activities designed for specific instructional approaches or teaching materials. Many articles [17-22] have proved that learners can achieve excellent learning performance if we can give them instruction and teaching materials according to their individual learning styles. Sternberg [23] also collected many taxonomies of learning style based upon different criteria. Thus, we apply three features of learning styles, **V**isual, **A**uditory, and **K**inesthetic, in IAM to generate adaptive learning guidance. To provide a learner with suitable learning contents, we have to define not only the learning style of the learner, but also the learning content of AT. Therefore, we need to select a suitable AT whose learning style is similar to that of the learner. Moreover, we can use existing questionnaires [23, 24] to extract the values of individual learning styles of learners.

• **Organization of Teaching Material:** It is essential to organize suitable teaching materials for students. According to Bassing [25], we can categorize the organization of teaching materials into three types: **(1) Logical Organization**, where the teaching materials are ordered in a systematical fashion as traditional teaching strategies, e.g., teaching the mathematics from basic to advanced concept in a fixed order; **(2) Psychological Organization**, where emphasis is placed on the student's own interest, ability, and needs; and **(3) Eclectic Organization**, which takes both Logical Organization and Psychological Organization into consideration. Therefore, in IAM, the learning guidance and selected AT have to be based on the concepts of Logical Organization and Psychological Organization, respectively. Table 3 shows the related symbol definitions used when applying Pedagogical Theory in IAM.

Table 3. The symbol definitions for applying pedagogical theory in IAM.

Symbols	Description
$LgOrg_i$	This denotes the **Logical Organization** of AT_i. The value of $LgOrg_i$ is mapped to the difficulty of AT_i.
$LnSty_i$	This denotes the value of **Learning Style**, including **V**isual, **A**uditory, and **K**inesthetic in AT_i. The LnSty is represented as a vector, i.e., $<\mathbf{V_{ATi}}, \mathbf{A_{ATi}}, \mathbf{K_{ATi}}>$, where the value is between 0 and 1.
SLS	This denotes the **Student Learning Style** (SLS) for representing the learning style of the student. **SLS** is represented as a vector like $\mathbf{LnSty_i}$, i.e., $<\mathbf{V_s}, \mathbf{A_s}, \mathbf{K_s}>$, where the value is between 0 and 1.

Based upon the symbols shown in Table 3, we can define the **S**imilarity **F**actor, **SF**, and redefine the **C**hosen **F**actor, **CF**, for AT_i as follows:

• $SF_i = SLS \cdot LnSty_i$, where the symbol "•" represents the dot product.
• $CF_i = \alpha NOW_i + \beta SGP_i + \gamma LgOrg_i$, where $\alpha + \beta + \gamma = 1$.

The **SF** is used to compute the similarity of the learning style between the learner and ATs. Thus, we can filter out ATs with low SF values and then select the AT with the highest CF value. Although we have defined the selection formula and strategy according to Pedagogical Theory, teachers also can redefine them by themselves.

4.2 AT Selection Process Using Pedagogical Theories

Therefore, in the **AT Selection Algorithm**, we can compute CF and SF to acquire the psychological organization and logical organization characteristics of every AT **(Step 2.1)**. The SF, which is computed as the dot product of the student's learning style vector (SLS) and the AT's learning style vector (LnSty), can denote the similarity of the learning style between the AT and Learner. Thus, using the value of SF, we can get a suitable AT form IAM **(Step 2.2.1)**. Finally, the CF can be used to determine the most suitable AT for the learner **(Step 2.2.2)**.

Example 2: Learning in IAM using pedagogical theories

We will present a simple example of learning in IAM using pedagogical theories. First, we will define IAM and the related attributes of each AT, and then we will demonstrate the process of the AT Selection Algorithm for a specific student. An example of IAM is shown in Fig. 7.

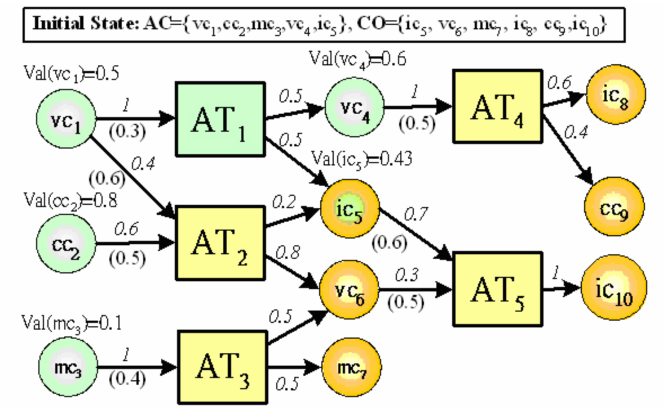

Fig. 7. An example of IAM using pedagogical theories.

IAM as shown in Fig. 7 is represented as follows:

IAM = ({AT_1, AT_2, AT_3, AT_4, AT_5}, {vc_1, cc_2, mc_3, vc_4, ic_5, vc_6, mc_7, ic_8, cc_9, ic_{10}}, {(e_{11}, 0.3), (e_{12}, 0.6), (e_{22}, 0.5), (e_{33}, 0.4), (e_{44}, 0.5), (e_{55}, 0.6), (e_{65}, 0.5)}, {e'_{14}, e'_{15}, e'_{25}, e'_{26}, e'_{36}, e'_{37}, e'_{48}, e'_{49}, $e'_{5,10}$}).

Table 4. Learning style and logical organization of each AT.

	AT_1	AT_2	AT_3	AT_4	AT_5
LnSty	<0.8, 0.1, 0.1>	<0.1, 0.8, 0.1>	<0.6, 0.1, 0.3>	<0.2, 0.1, 0.7>	<0.1, 0.2, 0.7>
LgOrg	0.3	0.3	0.5	0.3	0.7

The Learning Style and Logical Organization used in the AT Selection Algorithm are shown in Table 4. Because the value of LgOrg is mapped to the difficulty of AT, the difficulty of the metadata in SCORM can be used to define the value range, e.g., {*Very Easy*, *Easy*, *Medium*, *Difficult*, *Very Difficult*} corresponding to {0.1, 0.3, 0.5, 0.7, 0.9}. Suppose there is a learner who is learning in this IAM; her/his personal information is as follows:

AC = $\{(vc_1, 0.5), (cc_2, 0.8), (mc_3, 0.1), (vc_4, 0.6), (ic_5, 0.43)\}$;
SLS = $<0.1, 0.2, 0.7>$;
CO = $\{ic_5, vc_6, mc_7, ic_8, cc_9, ic_{10}\}$.

Since s/he has learned AT_1, the AT Selection Algorithm will choose the next AT for her/his learning. CF_i and SF_i are defined as follows:

- $CF_i = 0.2 \times NOW_i + 0.2 \times SGP_i + 0.4 \times LgOrg_i$,
- $SF_i = SLS \bullet LnSty_i$.

The related results obtained by the AT Selection algorithm are shown in Table 5.

Table 5. Selection criteria for each activity tree.

	AT_2	AT_3	AT_4
PCL	$\{ic_5, vc_6, ic_{10}\}$	$\{vc_6, mc_7, ic_{10}\}$	$\{ic_8, cc_9\}$
NOW	1	1	1
SGP	$0.5 \times 0.4 + 0.8 \times 0.6 = 0.68$	$0.1 \times 1 = 0.1$	$0.6 \times 1 = 0.6$
LgOrg	0.3	0.5	0.3
SF_i	0.24	0.29	0.53
CF_i	0.456	0.42	0.44

Then, we can use the following selection strategy: for smart students, select the AT with the highest CF_i value; for other students, select the AT with the highest SF_i value. With this strategy, we select AT_2 for smart students, and AT_4 for other students. In addition, we can revise CF_i and SF_i for specific purposes. For example, some teachers believe that learning style of a student is related to student's grade, and they can modify CF_i and SF_i as $CF_i = 0.5 \times NOW_i + 0.5 \times LgOrg_i$, $SF_i = 0.5 \times SGP_i + 0.5 \times (SLS \cdot LnSty_i)$. If the selection strategy remains the same, we will provide AT_3 for smart students and AT_4 for other students.

4.4 Evaluating of the Expressive Power of IAM

We have shown that it is possible to apply pedagogical theories in IAM for specific need. How many pedagogical theories can be applied in IAM? In this section, we will evaluate that how many different structures IAM can support to meet pedagogical needs.

Educational researchers have proposed various types of course structures to facilitate learning. Posner [26] proposed three types of structures including discrete structure, linear structure, and hierarchical structure. Bruner [27] proposed the concept of a spiral curriculum. Efland [28] also proposed the lattice curriculum. Each structure satisfies certain kinds of pedagogical needs. IAM can be applied to these course structures, as shown in Figs. 8 and 9.

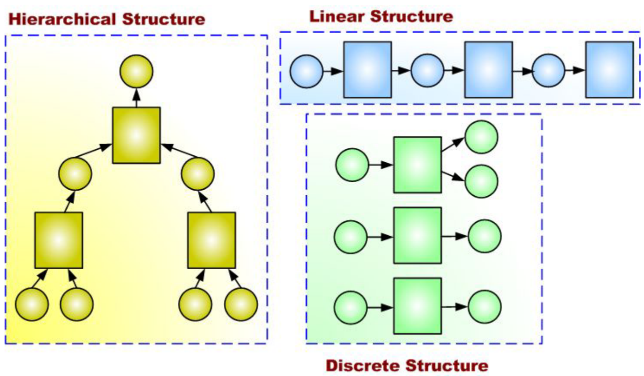

Fig. 8. IAM mapping to a discrete structure, linear structure, and hierarchical structure.

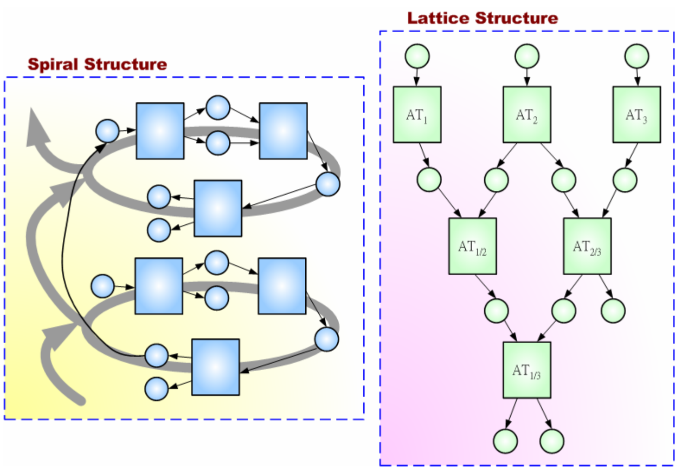

Fig. 9. IAM mapping to the spiral curriculum and lattice curriculum.

5. THE CONSTRUCTION OF IAM

As mentioned in previous sections, based upon the OO Methodology and SCORM standard, we have proposed an Instruction Activity Model (IAM) which is composed of related AT components with inter-relations and specific attributes designed to meet pedagogical needs. However, for teachers and authors, how to apply IAM in real learning environments is also an important issue. Therefore, in this section, we propose a systematic approach to fast and easily construct IAM using traditional course resources.

First, the teacher has to create the *Content-Contribution Relationship Table* denoting the potential concept which will be acquired by learning the learning content. For example, assume that a course, *Introduction to Computers*, includes three chapters as shown in Table 6. According to the content of Chapter A, the teacher can write down its possible contributions, including the related $w(e'_{ij})$ and difficulty level; e.g., $A_1(0.5, 1)$ indicates that the contribution, called *Hardware*, has $w(e'_{ij}) = 0.5$ and *difficulty level* = 1. Then, we use the concept of the Adjacency Matrix to create the Weight Matrix of *Contribution* as shown in Tables 7 and 8. Thus, assuming that there is an $m \times n$ Weight Matrix (M), the weight of m_{ij} in M denotes the significance of c_i before learning c_j. Hence, the teacher can write down the value of m_{ij} to define the related weight between *contribution* c_i and c_j using the following formula:

$$m_{ij} = \begin{cases} x, \text{ if Difficulty } (c_i) \leqq \text{ Difficulty } (c_j), \text{ where } 0 \leqq x \leqq 1 \\ 0, \text{ otherwise} \end{cases}.$$

For example, in Table 7, $A_1(1)$ indicates that the *Contribution* A_1 has difficulty level = 1. The m_{11} between A_1 and B_1 can be written as 0.3 by teachers because the Difficulty(A_1) $\leqq$ Difficulty(B_1). After finishing the Weight Matrix, the teacher can compute the value of $w(e_{ij})$ of every *contribution* using the following equation (*the equation will normalize* $w(e_{ij})$):

$$w(e_{ij}) = \frac{\sum_{1 \le j \le n} m_{ij}}{\sum_{1 \le j \le n} \sum_{1 \le j \le n} m_{ij}}$$

Table 6. The content-contribution relationship table of a course.

	Contributions (*w(e'ij), difficult level*)		
	1	2	3
Chapter A: *Introduction*	A_1 (0.5, 1) (*Hardware*)	A_2 (0.3, 1) (*Software*)	A_3 (0.2, 1) (*Application*)
Chapter B: *Hardware*	B_1 (0.2, 3) (*CPU*)	B_2 (0.4, 3) (*Main Memory*)	B_3 (0.4, 3) (*Auxiliary Memory*)
Chapter C: *Software System*	C_1 (0.5, 4) (*System Software*)	C_2 (0.5, 2) (*Application Software*)	

Table 7. The weight matrix of contribution B.

	$B_1(3)$	$B_2(3)$	$B_3(3)$	$w(e_{ij})$ of B
$A_1(1)$	0.3	0.3	0	0.67
$A_2(1)$	0	0	0.3	0.33
$A_3(1)$	0	0	0	0
$C_2(2)$	0	0	0	0

Table 8. The weight matrix of contribution C.

	$C_1(4)$	$C_2(2)$	$w(e_{ij})$ of C
$A_1(1)$	0	0	0.0
$A_2(1)$	0.8	0.3	0.5
$A_3(1)$	0.2	0.5	0.32
$B_1(3)$	0.2	0	0.09
$B_2(3)$	0.2	0	0.09
$B_3(3)$	0	0	0

Finally, based upon the Weight Matrix, $w(e'_{ij})$, and $w(e_{ij})$, the teacher can construct IAM as shown in Fig. 10.

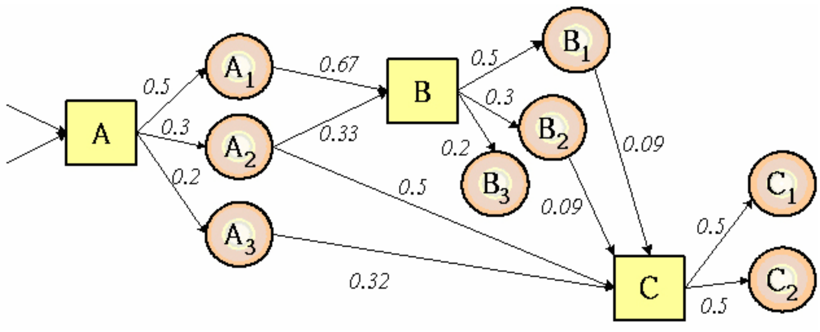

Fig. 10. The design of IAM for "*Introduction to Computers*".

6. CONCLUSIONS

To create, represent and maintain complex Activity Trees (ATs) and associated sequencing definitions in SCORM 1.3, and to apply Pedagogical Theory to provide learners with customized learning contents, we have extended and modularized the structure of AT based on Pedagogical Theory and the concept of the Object Oriented Methodology, respectively. We have also proposed the Instructional Activity Model (IAM), which is composed of related Activity Tree nodes. Based upon the Object Oriented Methodology, each AT node in IAM is modularized as a learning unit with inter-relations and specific attributes so that it can be easily managed, reused, and integrated. An AT Selection algorithm with Pedagogical Theory has also been proposed to generate dynamic learning content for the learner by traversing IAM. Finally, we have also proposed a systematic way to quickly and easily construct IAM using traditional course resources.

IAM with its properties of scalability and flexibility, can apply different pedagogical theories to meet specific need by means of extension schemes. Although several selection formulas and strategies have been defined based upon Pedagogical Theory, they can be redefined according to the preferences or specific needs of teachers and authors. For example, the linear combination of several selection criteria used to compute the similarity (SF) and suitability (CF) of an AT can be revised by adding, deleting, or modifying the attributes and the revised criteria will then influence the selection result of the AT. Therefore, increasing the expressive power of the selection equation according to different pedagogical theories or IT methodologies is important. Now, we are developing an IAM prototype system to manage and dynamically generate a personalized SCORM compliant course by integrating the Inference Engine and Knowledge Base. In the near future, we will enhance our proposed IAM and apply it to some specific domains, e.g., natural science courses for elementary schools, to evaluate its flexibility, scalability and learning performance.

REFERENCES

1. SCORM (Sharable Content Object Reference Model), http://www.adlnet.org/.
2. IMS (Instructional Management System), http://www.imsproject.org/.
3. LTSC (IEEE Learning Technology Standards Committee), http://ltsc.ieee.org/wg12/.
4. AICC (Aviation Industry CBT Consortium), http://www.aicc.org.
5. XML (eXtensible Markup Language), http://www.w3c.org/xml/.
6. W3C, World Wide Web Consortium, http://www.w3.org.
7. R. M. Gagne, "Learning outcomes and their effects: useful categories of human performance," *American Psychologist*, Vol. 39, 1984, pp. 377-385.
8. K. A. Papanikolaou, M. Grigoriadou, G. D. Magoulas, and H. Kornilakis, "Towards new forms of knowledge communication: the adaptive dimension of a web-based learning environment," *Computers and Education*, Vol. 39, 2002, pp. 333-360.
9. S. Chen, "A cognitive model for non-linear learning in hypermedia programmes," *British Journal of Educational Technology*, Vol. 33, 2002, pp. 449-460.
10. E. Triantafillou, A. Pomportsis, and S. Demetriadis, "The design and the formative evaluation of an adaptive educational system based on cognitive styles," *Computers and Education*, Vol. 41, 2003, pp. 87-103.
11. ARIADNE (Alliance for Remote Instructional and Authoring and Distribution Networks for Europe), http://www.ariadne-eu.org.
12. E. R. Jones, SCORM, http://www.scorm.tamucc.edu/scorm/home.html.
13. V. Carchiolo, A. Longheu, and M. Malgeri, "Adaptive formative paths in a web-based learning environment," *Educational Technology and Society*, Vol. 5, 2002, pp. 64-75.
14. L. Sheremetov and A. G. Arenas, "EVA: an interactive web-based collaborative learning environment," *Computers and Education*, Vol. 39, 2002, pp. 161-182.
15. B. S. Bloom, B. B. Mesia, and D. R. Krathwohl, *Taxonomy of Educational Objectives* (*two vols: The Affective Domain and The Cognitive Domain*), David McKay, New York, 1964.
16. R. M. Gagné, L. Briggs, and W. Wager, *Principles of Instructional Design*, 4th ed., Harcourt Brace Jovanovich, 1992.
17. G. Shaw and N. Marlow, "The role of student learning styles, gender, attitudes and perceptions on information and communication technology assisted learning," *Computers and Education*, Vol. 33, 1999, pp. 223-234.
18. S. R. Terrell, "The effect of learning style on doctoral course completion in a

web-based learning environment," *The Internet and Higher Education*, Vol. 5, 2002, pp. 345-352.
19. Y. Shang, H. C. Shi, and S. S. Chen, "An intelligent distributed environment for active learning," *ACM Journal of Educational Resources in Computing*, Vol. 1, 2001, pp. 4-17.
20. E. Trantafillou, A. Poportsis, and S. Demetriadis, "The design and the formative evaluation of an adaptive educational system based on cognitive," *Computer and Education*, Vol. 41, 2003, pp. 87-103.
21. M. Liu, "The relationship between the learning strategies and learning styles in a hypermedia environment," *Computers in Human Behaviour*, Vol. 10, 1994, pp. 419-434.
22. L. Curry, "Patterns of learning styles across selected medical specialties," *Educational Psychology*, Vol. 11, 1991, pp. 247-277.
23. R. Sternberg and E. Grigorenko, *Styles of Thinking in the School*, Branco Weiss Institute, 1998.
24. P. Honey and A. Mumford, *The Manual of Learning Styles*, Peter Honey, Maidenhead, 1986.
25. N. L. Bassing, *Teaching in Secondary Schools*, Houghton Mifflin Company, Boston, 1963, pp. 51-55.
26. G. J. Posner, *Analyzing the Curriculum*, McGraw-Hill, New York, 1992.
27. J. S. Bruner, *The Process of Education*, Cambridge, MA: Harvard University Press, 1960.
28. A. D. Efland, "The spiral and the lattice: changes in cognitive learning theory with implications," *Studies in Art Education*, Vol. 36, 1995, pp. 134-153.

Jun-Ming Su (蘇俊銘) was born in Kaohsiung, Taiwan, on February 18, 1974, and graduated with a B.S. degree from the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan, in 1997. He received the M.S. degree from the Institute of Computer Science, National Chung Hsing University, Taiwan, in 1999. Currently, he is a Ph.D. student at National Chiao Tung University, Taiwan. His current research interests include intelligent tutoring systems, knowledge engineering, expert systems, and data mining.

Shian-Shyong Tseng (曾憲雄) received his Ph.D. degree in Computer Engineering from National Chiao Tung University in 1984. Since August 1983, he has been on the faculty of the Department of Computer and Information Science at National Chiao Tung University, and he is currently a Professor there. From 1988 to 1992, he was the Director of the Computer Center of National Chiao Tung University. From 1991 to 1992 and from 1996 to 1998, he acted as the Chairman of the Department of Computer and Information Science. From 1992 to 1996, he was the Director of the Computer Center of the Ministry of Education and the Chairman of the Taiwan Academic Network (TANet) management committee. In December 1999, he founded Taiwan Network Information Center (TWNIC), and he is now the Chairman of the board of directors of TWNIC. Since 2002, he has been President of the SIP/ENUM Forum, Taiwan. In July 2003, he organized a committee for the Taiwan Internet Content Rating Foundation, and is now the Chair. His current research interests include parallel processing, expert systems, computer algorithms, and Internet-based applications.

Ching-Tai Chen (陳鯨太) was born in Taipei, Taiwan, on September 16, 1979, and he graduated with B.S. and M.S. degrees from the Department of Computer and Information Science, National Chiao Tung University, Taiwan, in 2000 and 2002, respectively. Currently, he is a research assistant in the Institute of Information Science, Academia Sinica, Taiwan. His current research interests include e-learning, data mining, and bioinformatics.

Wen-Nung Tsai (蔡文能) received his B.S. degree from National Chiao Tung University in 1977 and his M.S. degree in computer science from National Chiao Tung University in 1979. He was in the Ph.D. program in Computer Science at Northwestern University between 1987 and 1990. He is now an Associate Professor in the Department of Computer Science and Information Engineering. His current research interests include mobile computing, distributed computing, network security, operating systems, and distance learning.