符合教學需求之適性化學習環境
Adaptive Learning Environment for Pedagogical Needs

Ching-Tai Chen
Department of Computer and Information Science, National Chiao-Tung University
caster@cis.nctu.edu.tw

Shian-Shyong Tseng
Department of Computer and Information Science, National Chiao-Tung University
sstseng@cis.nctu.edu.tw

Chang-Zhuo Chen
Department of Computer and Information Science, National Chiao-Tung University
muirthemne.cis89@nctu.edu.tw

Jun-Ming Su
Department of Computer Science and Information Engineering, National Chiao-Tung University
jmsu@csie.nctu.edu.tw

Huan-Yu Lin
Department of Computer and Information Science, National Chiao-Tung University
huan.cis89@nctu.edu.tw

Yi-Li Liu
Department of Computer and Information Science, National Chiao-Tung University
yili875.cis89@nctu.edu.tw

Abstract

With vigorous development of the internet, e-learning system has become more and more popular. Currently, to solve the issue of sharing and reusing of teaching materials in different e-learning system, several standard formats including SCORM, IMS, LOM, and AICC, etc. have been proposed by international organizations. Among international standards, Sharable Content Object Reference Model (SCORM) has become the most popular standard. In SCORM standard, the Simple Sequence Specification, which relies on the concept of learning activities, defines the course sequencing behavior, which controls the sequencing, selecting and delivering of course, and organizes the content into a hierarchical structure, namely Activity Tree (AT). For a large learning activity, the Activity Tree will become too complex to manage and reuse, much less its sequencing behavior rules. Moreover, the lack of inter-relations among Activity Trees also makes the reusing and the reintegrating hard. These defects will limit the scalability and flexibility of adaptive learning system. Therefore, how to create, represent and maintain AT and associated sequencing definition are our concerns. Besides, for more personalized learning environment, how to extend the structure of AT with applying Pedagogical Theory also becomes an important issue.

Therefore, in this paper, we extend and modualize the structure of AT by applying Pedagogical Theory and concept of Object Oriented Methodology, respectively. Thus, we first propose a novel model, Instructional Activity Model (IAM), which is composed of related AT nodes. Each AT node in IAM is modualized as a learning unit with inter-relations and specific attributes, which can be easily managed, reused, and integrated. We also propose two heuristic algorithms with general and specific domain heuristic to traverse IAM to generate the dynamic learning content to the learner. IAM with scalability and flexibility can apply different pedagogical theories for specific need by extension mechanisms. In addition, based on SCORM 1.3 RTE (Run Time Environment), we develop an IAM system to manage and dynamically generate personalized SOCRM compliant course. Based upon the results of experiment, we may conclude that the IAM system is workable and compatible with SCORM standard.

Keywords: Adaptive Learning, Intelligent Tutoring System, SCORM, Activity Tree, Pedagogical Theory.

1. Introduction

With vigorous development of the internet, in the past ten years, e-learning system has become more and more popular because it can make learner study at any time and any location conveniently. However, because the teaching materials in different e-learning systems are usually defined in specific data format, the sharing of the teaching materials among these systems becomes difficult, resulting in increasing the cost of creating teaching materials. To solve the issue of uniform teaching materials format, several standard formats including SCORM [2], IMS [3], LOM [4], and AICC [5], etc. have been proposed by international organizations. By these standard formats, the teaching materials in different learning management systems can be shared, reused, and recombined. Among international standards, Sharable Content Object Reference Model (SCORM), which integrates IMS, LOM, and AICC, has become the most popular international standard. Based on the concept of learning object, SCORM uses the metadata to specify structure of every learning object and proposes the content aggregation scheme to package it with Extensible Markup Language (XML) [6][7] format.

At present, SCORM 1.3 adopts the Simple Sequence Specification (SSS) of IMS [3] to define the course sequencing behavior. The SSS relies on the concept of learning activities. A learning activity may be loosely described as an instructional event or events embedded in a content resource, or as an aggregation of activities to describe content resources with their contained instructional events. Content in simple sequencing is organized into a hierarchical structure, namely Activity Tree (AT). The Simple Sequencing process uses information about the desired sequencing behavior to control the sequencing, selecting and delivering of activities to the learner. Therefore, by this standard, we can develop an intelligent approach to (semi-)automatic course or exercise sequencing.

Simple Sequencing makes no requirements on the structure, organization or instruction of the Activity Tree. The tree and the associated sequencing definitions may be static or dynamically created. However, how to create, represent and maintain the Activity Tree and associated sequencing are our concerns. For a large learning activity, the Activity Tree will become too complex to manage and reuse, much less its sequencing behavior rules. Moreover, the lack of inter-relations among Activity Trees also makes the reusing and the reintegrating hard. These defects will limit the scalability and flexibility of adaptive learning system. Existing intelligent tutoring systems often use the grade and learning duration of learner to evaluate the learning result and to decide the delivering sequence of learning content. However, to evaluate the personal learning behavior, the grade and learning duration may be not sufficient. Therefore, many researches used Pedagogical Theory [14][17][20][22]to enhance the evaluation of the personal learning characteristic and to generate more personalized learning guidance.

Hence, in this paper, the learning characteristic of learner has been taken into consideration. Our approach is to extend and modualize the structure of AT by applying Pedagogical Theory and concept of Object Oriented Methodology respectively to construct a personalized learning environment. One large AT is modualized into several suitable AT nodes which possess several specific attributes and associated inter-relations. By these properties, each AT node can be reused and reintegrated to generate new organization of course for decreasing the cost of designing learning activity, and increasing the scalability and flexibility. For the needs of specific purpose, e.g. pedagogical needs, the attribute and interrelation of AT can be extended, too. Therefore, we propose a novel model, 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 which the IAM can be easily managed, reused, and integrated by. In addition, we also propose two heuristic algorithms with general and specific domain heuristic to traverse IAM to generate the dynamic learning content to the learner. IAM with scalability and flexibility can apply different pedagogical theories for specific need by extension mechanisms. In addition, based on SCORM 1.3 RTE (Run Time Environment) [2], we develop an IAM system to manage and dynamically generate personalized SOCRM compliant course. Based upon the results of experiment, we may conclude that the IAM system is workable and compatible with SCORM standard.

The main contributions of this paper are:

1. Propose a general purposed model, called Instructional Activity Model (IAM), to generate adaptive learning course which is compatible with SCORM standard.
2. Modualize a large Activity Tree into several suitable AT nodes with specific attributes and inter-relations which can be easily managed, reused, and reintegrated, based on OO methodologies.
3. Apply Pedagogical Theory in IAM to define the personal learning characteristic and evaluate learning result of learner for generating more adaptive learning.
4. Develop an IAM system, which integrates Simple Sequencing Specification, inference engine, DRAMA, and SCORM RTE 1.3, to deliver individual learning course.

2. Related Work

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

2.1 SCORM (Sharable Content Object Reference Model) [2]

Among those 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, IMS, AICC and ARIADNE.

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: easily modified and used by different development tools, Accessible: can be searched and made available as needed by both learners and content developers, Interoperable: operates across a wide variety of hardware, operating systems and web browsers, and Durable: does not require significant modifications with new versions of system software [8].

2.2 Simple Sequencing Specification [2][3]

The Simple Sequencing specification of IMS [3], which was adopted by SCROM 1.3, relies on the concept of learning activities. A learning activity may be loosely described as an instructional event or events embedded in a content resource, or as an aggregation of activities to describe content resources with their contained instructional events. Content in simple sequencing is organized into a hierarchical structure, namely activity tree (AT) as a learning map. The examples of AT are shown in Figure 1. Each 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 Simple Sequencing. The Simple Sequencing process uses information about the desired sequencing behavior to control the sequencing, selecting and delivering of activities to the learner. The intended sequence is described by a specific set of data attributes, which are associated with learning activities in the activity tree to describe the sequencing behavior. Moreover, the activity tree can be considered as the learning map.

The sequencing behaviors describe how the activity or how the children of the activity are used to create the desired learning experience. Simple Sequencing makes no requirements 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. Simple Sequencing enables us to share not only learning contents, but also intended learning experiences. It provides a set of widely used sequencing method so that the teacher could do the sequencing efficiently. However, the definition of sequencing behavior rules is obviously too simple to satisfy pedagogical needs.

2.3 Other Related Research

Carchiolo [12] has proposed adaptive formative paths for e-learning environment. They constructed domain database and students' 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 student's priori knowledge and generation of adaptive learning path, it didn't take pedagogical theory into account and it could not be compatible with SCORM standard yet.

Sheremetov [19] also proposed a system, called EVA, for developing a virtual learning space in the National Technical Institute in Mexico. EVA consists of five virtual learning spaces: 1. Knowledge Space: all necessary information to learn, 2. Collaborative Space: real or virtual companions that get together to learn, 3. Collaborative Space: the teachers or tutors (also real and virtual) who give the right direction for learning and consult doubts, 4. Experimentation Space: the practical work of the students in virtual environment to obtain, and 5. Personal Space: records about users' information. The model of knowledge is represented in the form of graph, and each node is a unit of learning material (ULM) which is the basic element of knowledge structure. ULMs with a related knowledge concept can be grouped into a POLIlibro (or Multi-Book) along the learning trajectory (path) depending on students. However, the relations between ULMs are not sufficient for expressing the structure of the knowledge model and the attributes of an ULM is not enough for mining the behaviors of students. They also proposed some methodologies for planning of trajectories and scheduling of learning activities based on the agent technology. However, they didn't discuss how to generate a learning path.

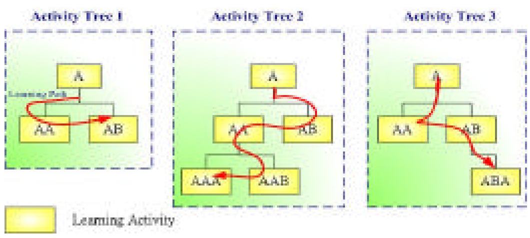

Figure 1: Examples of Activity Tree

3. Instructional Activity Model (IAM)

As mentioned above, SCORM standard defines a hierarchical structure, namely Activity Tree (AT), to sequence the learning content to the learner. By defining the sequencing behavior rules within AT, we can develop an intelligent approach to (semi-)automatic course and exercise sequencing. However, how to create, represent and maintain the Activity Tree and associated sequencing definition are our concerns. For a large learning activity, the Activity Tree will become too complex to manage and reuse. It is hard to reuse and reintegrate ATs without the inter-relations. These defects will limit the scalability and flexibility of adaptive learning system. Moreover, for more personalized learning, many researches have used the Pedagogical Theory [14][17][20][22] to enhance the evaluation of the personal learning characteristic. Hence, in this paper, we extend and modualize the structure of AT by applying Pedagogical Theory and concept of Object Oriented Methodology, respectively. Thus, we propose a novel model, Instructional Activity Model (IAM), which is composed of related Activity Tree nodes. Based upon Pedagogical Theory, each AT node in IAM is modualized as a learning unit with inter-relations and specific attributes, which can be easily managed, reused, and integrated. We also propose two heuristic algorithms with general and specific domain heuristic, to traverse IAM to generate the dynamic learning content to the learner. In this section, we will describe the concept of Instructional Activity Model including its properties and the learning process algorithms.

3.1 Concept of Instructional Activity Model

In Simple Sequencing Specification (SSS), we can create an AT on the fly. As mentioned above, in a large AT, its organization and sequencing rules definition are hard to manage and reuse. However, too many small ATs also result in that the management of AT nodes and rules definition is complicated. Therefore, for the scalability and flexibility of AT, how to define a suitable unit of AT becomes an important issue. According to Bloom's Mastery Theory [10], 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,…,AT_n}. According to the formulation of Gagne [15]: "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." Therefore, to suppose there are m capabilities, we can also obtain $\mathbf{C_{set}}$={c_1, c_2,…,c_m}. Before learning an AT, students are supposed to possess some capabilities, called ***Prerequisite***. Similarly, after learning an activity tree, students can acquire further capabilities, called ***Contributions***. Every ***prerequisite*** and ***contribution*** has its own weight representing significance 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 has several capabilities.

A learning activity or a course is composed of several ATs with input/output capabilities. The student will learn easy AT he can qualify and gain further capabilities which enable student to learn another advanced AT. This learning process will be repeated until the student has finished all learning objectives. Then, every student will have an individual value of C_{set}. The Figure 2 shows the diagram of IAM.

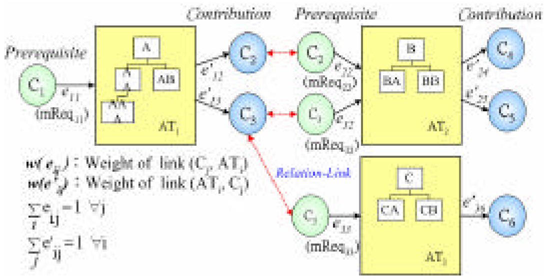

Figure 2: The Diagram of IAM

In Figure 2, we can define these terms as follows:

- $\mathbf{e_{ij}}$ **:** is an edge from c_i to AT_j, called *"prerequisite edge"*. It means that before learning AT_j, the student is supposed to possess this ability c_i.
- $\mathbf{e'_{ij}}$ **:** is an edge from AT_i to c_j, called *"contribution edge"*. It means that after learning AT_i, the student will gain the ability c_j.
- $\mathbf{w(e_{ij})}$ **:** is the weight of e_{ij}. It represents the significance of c_i before learning AT_j. Note that the sum of $w(e_{ij})$ of an AT is 1, that is, $\sum_i w(e_{ij}) = 1, \forall j$.
- $\mathbf{w(e'_{ij})}$ **:** is the weight of e'_{ij}. It represents the significance of c_j after learning AT_i. Note that the sum of contribution weights of an AT is 1; that is, $\sum_j w(e'_{ij}) = 1, \forall i$.
- $\mathbf{mReq_{ij}}$ **:** is the minimum requirement of c_i for learning AT_j, which is denoted as $mReq_{ij}$. We use it to determine whether the student is qualified to learn AT_j or not.
- $\mathbf{mFun_i(AT)}$ **:** is the measure function of AT_i. It is used to evaluate whether the student is qualified to learn AT_i or not. We can define Measure Function (*mFun*) for each AT; e.g., the *mFun* of AT_2 can be defined as $mFun_2 = val(c_2) \times w(e_{22}) + val(c_3) \times w(e_{32})$.
- $\mathbf{grade(e'_{ij})}$ **:** represents the learning grade after learning AT_i.
- $\mathbf{val(c_m)}$ **:** the value of capability m ($\mathbf{c_m}$).

$$val(c_m) = \frac{\sum_j w(e'_{jm}) \times grade(e'_{jm})}{\sum_j w(e'_{jm})},$$

- **Student's Acquired Capability (AC):** records his/her learning results in the form of

$$AC = \bigcup (c_i, val(c_i)),$$

- **Student's Course Objectives (CO):** records his/her learning objectives in the form of

$$CO = \bigcup c_i.$$

- **Potential Capability List (PCL):** Each AT has a potential capability list recording all the contribution capabilities which can be reached from this AT via edges in IAM. It can be formulated as $PCL_{ATj} = \bigcup c_i$, *//C_i can be reach* *form AT_j by connected edges*//. For example, in Figure 2 the PCL_{AT1} equals $\{C_2, C_3, C_4, C_5, C_6\}$.

- **Student's Grade Prediction (SGP):** represents the performance prediction of the specific student related to the activity tree. We use measure function (mFun) and acquired capabilities (AC) to compute SGP value of each AT, For example, $SGP_2 = mFun_2 = val(c_2) \times w(e_{22}) + val(c_3) \times w(e_{32})$
- **Normalized Objective Weight (NOW):** represents the relativity between an activity tree and the student's course objectives. Higher objective weight implies better learning efficiency. Empirically, selecting function tends to select the activity tree with higher SGP and NOW for students. We propose two following approaches to compute the value of NOW.

1. **Normalized Objective Weight Computing by Heuristic1 ($NOWC^{h1}$):**

$$NOW_j^{h1} = \frac{\text{the number of } c_i \ (c_i \in PCL \ \& \ c_i \in CO)}{\text{the number of } c_i \ (c_i \in PCL)}$$

2. **Normalized Objective Weight Computing by Heuristic1 ($NOWC^{h2}$):**

Step1: $\forall c_i$ in PCL_j, use DFS to find the path between AT_j and c_i, $path_i$ is of the form $e'_{j1}, e_{12}, \ldots, e'_{ni}$, where each set is a directed link in IAM

Step2: $\forall c_i$ in PCL_j, compute its path donation (PD_i) where PD_i = product of every link donation (LD) = $LD(e'_{j1}) \times LD(e_{12}) \times \ldots \times LD(e'_{ni})$ where

$$\begin{cases} \text{if link is of the form } e'_{xy}, \ LD(e'_{xy}) = \dfrac{w(e'_{xy})}{\sum_i w(e'_{iy})} \\ \text{if link is of the form } e_{xy}, \ LD(e_{xy}) = w(e_{xy}) \end{cases}$$

Step3: $$NOW_j^{h2} = \frac{\sum_{i, c_i \in CO} PD_i}{\sum_i PD_i}$$

- **Chosen Factor (CF):** is a linear combination of selecting criteria, NOW and SGP.

$CF_i = aNOW_i + ßSGP_i$, where $a+ß=1$, $0 \le a, ß \le 1$.

In brief, Instructional activity model (IAM), a graphical based representation of a learning activity or course, contains several **AT**s, **Capabilities** including *prerequisite* and *contribution*, **Relations Edges** including e_{ij} with $mReq_{ij}$ and e'_{ij}, and several **Measure Functions**. It can be formulated as **IAM=(AT_{set}, C_{set}, E_{set}, E'_{set}),**
where

- $\mathbf{AT_{set}} = \{AT_1, AT_2, \ldots, AT_n\}$.
- $\mathbf{C_{set}} = \{c_1, c_2, \ldots, c_n\}$.
- $\mathbf{E_{set}}$ is the set of all prerequisite edges with minimum requirement value in an IAM. It is formulated as $E_{set} = \bigcup E_j$, $\mathbf{E_j} = \bigcup_i (e_{ij}, mReq_{ij})$, $e_{ij} \in AT_j$.
- $\mathbf{E'_{set}}$ is the set of all contribution edges in an IAM. It is formulated as $\mathbf{E'_{set}} = \bigcup E'_j$, $\mathbf{E'_j} = \bigcup_j e'_{jk}$, $e'_{jk} \in AT_j$.

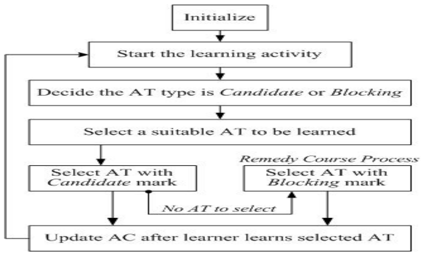

Figure 3: The Flowchart of Learning Process

3.2 Basic Functionalities

Based upon the structure of IAM described above, we can develop several approaches to provide students a learning environment with a dynamic and adaptive course. The learning process can be simply considered as sequencing activity trees in IAM in order to let students satisfy learning objectives. The flowchart and algorithm of learning process is shown in Figure 3 and Figure 4, respectively.

Here, we explain the learning process algorithm of IAM. First, we initialize the learning state by loading AC and CO, and evaluate the PCLAT of every AT (Setp1-Step2), and then enter the loop of learning activity (Step3). During learning process, we mark each AT with *Candidate* tag or *Blocking* tag by comparing the $mReq(e_{ij})$ with $val(c_i)$ (Step3.1). The *Candidate* type represents that this AT will be selected and the *Blocking* type is opposite just. Then, we select a suitable AT with *candidate* type first and deliver it to the learner (Step3.2). If there exists no AT with *candidate*, the learning process goes into the Remedy Course Process (Step3.2.1-Step 3.2.5). At the Remedy Course Process, we will select an AT with the largest amount of $C_m \in CO$ (Step3.2.1) and then find a c_i with the smallest or largest value of $(mReq(e_{ij}) - val(c_i))$ according to **SelectingPolicy** (Step3.2.2-Step3.2.3). In this algorithm, we provide three policies to select different capability for adaptive learning. The policy of "*Easiest First*" trends to select a c_i which the learner has gained more high grade in but policy of "*Hardest First*" is opposite. After selecting a c_i, we can decide which AT connected with c_i to deliver to learner by computing $MAX((mReq(e_{ij}) - grade(e'_{jk})) * w(e'_{jk}))$ which implies that the progress of learner is the largest (Step3.2.4). When learner has finished and satisfied all course objectives, the learning process will be stopped.

Table 1: The Related Value of AT_1 and AT_2

	SGP	NOW	CF
AT_1	$val(c_1) \times w(e_{11}) = 0.82 \times 1 = 0.82$	**NOW^{h1}**: $\frac{\text{the number of } \{c_4, c_7, c_8\}}{\text{the number of } \{c_4, c_5, c_7, c_8, c_9\}} = \frac{3}{5} = 0.6$	**CF^{h1}**: $a \times SGP_1 + b \times NOW^{h1} = 0.5 \times 0.82 + 0.5 \times 0.6 = 0.7$
		NOW^{h2}: $\frac{PD_4 + PD_7 + PD_8}{PD_4 + PD_5 + PD_7 + PD_8 + PD_9} = \frac{1 + (1 \times 1 \times 0.67) + (1 \times 1 \times 0.25)}{1 + 1 + (1 \times 1 \times 0.67) + (1 \times 1 \times 0.25) + (1 \times 0.7 \times 0.56)} = 0.46$	**CF^{h2}**: $0.5 \times 0.82 + 0.5 \times 0.82 = 0.64$
AT_2	$SGP_2=0.45$	$NOW^{h1}{}_2=0.33$	$CF_2^{h1} = 0.39$
		$NOW^{h2}{}_2=0.152$	$CF_2^{h2} = 0.301$

Algorithm 1: Learning Process Algorithm
Input:
IAM: instructional activity model,
AC: acquired capabilities with normalized grade of student
CO: course objectives of the student.
SelectingPolicy: {*Easiest First*, *Medium First*, *Hardest First*}
Output:
AC: new AC after learner has finished learning activity.
Algorithm :
Step1: Load the learner's **Acquired Capabilities (AC)** and **Course Objectives (CO)**.
Step2: Evaluate all AT ∈ IAM with PCL AT
Step3: while(CO ⊄ AC) *//start the learning activity*
Begin *// decide type of AT is Candidate or Blocking state*
3.1: for all 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
3.2: if (∃ any AT with *Candidate* mark) *// select the AT with Candidate mark*
then (Select an AT_j with the highest CF_j and delivery it to the learner.)
else if (∃ any AT with *Blocking* mark)
then *//go to Remedy Course Process & select a suitable AT*
3.2.1: for all AT_j {Count the amount of $C_m \in CO$ which is connected by e'_{jm}.}
3.2.2: Select the AT_j with the largest amount of $C_m \in CO$.
3.2.3: for all c_i with e_{ij}
{ **if** *SelectingPolicy*="*Easiest First*",
then Find c_i with the smallest ($mReq(e_{ij}) - val(c_i)$).
else if *SelectingPolicy* ="*Hardest First*",
then Find c_i with the largest ($mReq(e_{ij}) - val(c_i)$).
else if *SelectingPolicy* ="*Medium First*",
then Find c_i with the medium ($mReq(e_{ij}) - val(c_i)$).}
3.2.4: Select AT_k with ∀i,
$MAX((mReq(e_{ij}) - grade(e'_{jk})) * w(e'_{jk}))$, $e'_{jk} \in E_j$.
3.2.5: Clear mark of AT_j and delivery AT_k to learner.
3.3: if learner pass the selected AT
then mark this AT with *Learned*.
3.4: update AC after the learner learns selected AT.
End
Step4: return AC.

Figure 4: Learning Process Algorithm

SelectingPolicy (Step3.2.2-Step3.2.3). In this algorithm, we provide three policies to select different capability for adaptive learning. The policy of "*Easiest First*" trends to select a c_i which the learner has gained more high grade in but policy of "*Hardest First*" is opposite. After selecting a c_i, we can decide which AT connected with c_i to deliver to learner by computing $MAX((mReq(e_{ij}) - grade(e'_{jk})) * w(e'_{jk}))$ which implies that the progress of learner is the largest (Step3.2.4). When learner has finished and satisfied all course objectives, the learning process will be stopped.

Example 1:
This IAM in Figure 5 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'_{26}, e'_{47}, e'_{48}, e'_{58}, e'_{59}}).

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 parenthesis is the $val(c_i)$.

The PLC_{AT} has been evaluated as shown in Figure 4. After first iteration of the **While** loop of algorithm, we can get results as shown in Table 1. Thus, AT_1 will be delivered to learner because it has highest CF value.

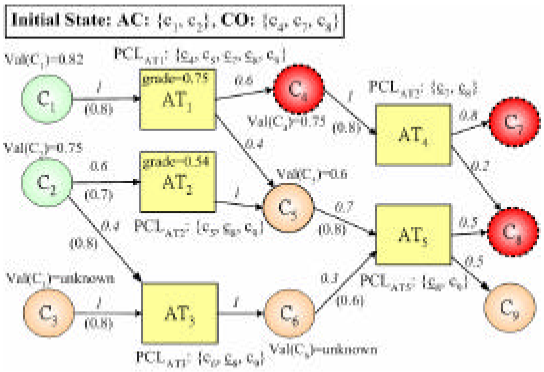

Figure 5: The Example of IAM

Case2: we assume that AC={ c_1(0.82), c_2(0.75) , c_4(0.75) , c_5(0.6) , c_6(unknow) }, CO={ c_4, c_7, c_8}, and Blocking AT = {AT_3, AT_5}. The learning process has gone into **Remedy Course Process**.

Before the Step 3.2.3, because AT_5 has one c_m ∈ CO, AT_5 is selected. If the **SelectingPolicy** is "*Easiest First*", the C_5 with the smallest value 0.2 of ($mReq(e_{55}) - val(c_5)$) is selected. Then, by computing the ($mReq(e_{55}) - grade(e'_{15})) \times w(e'_{15})$)

and (mReq(e_{25})–grade(e'_{25})) × w(e'_{25})), we can decide to deliver the AT_2 with value 0.26 to learner.

4. Applying Pedagogical Theories of IAM

We have proposed an Instructional Activity Model with several algorithms to provide scalability and flexibility of course sequencing in a dynamic way. This general model can be specialized by utilizing pedagogical theories. Now we are going to map several well-known pedagogical theories to IAM using extension mechanisms.

4.1 Pedagogical Theories

We consider pedagogical theories in three parts. In each part, we apply a well-known pedagogical theory to IAM in an attempt to show its applicability in practice. The three parts are 1. Organization of IAM, 2. Learning style of students, and 3. Capability taxonomy, which are described as follows.

1. Organization of IAM

It is essential to organize suitable teaching materials for students. According to the proposal of Nelson L. Bassing [9], we can categorize organizations of teaching materials into three types: **1).logical organization:** e.g., teaching from basic idea to advanced idea in mathematics, or teaching from ancient times to modern times in history, sequences teaching materials in a systematical fashion, **2).psychological organization:** places emphasis on the student's own interest, ability, and needs, and **3).eclectic organization:** does not take side with either logical organization or psychological organization but takes both of them into consideration.

2. Learning style of learner

There are lots of classification taxonomies collected in [21] based on different criteria. However, most of them are not designed for e-learning environment. We will simply utilize VAK learning style in IAM. V implies visual, A implies auditory, and K implies kinesthetic. Existing questionnaire [21] is used to extract learners' learning style. After each learner finishes the questionnaire, we can obtain his sensibility in visual level, auditory level, and kinesthetic level. We utilize this information in activity tree selection.

3. Capability taxonomy

Capabilities in IAM can be classified into several types based on different needs. We use what proposed by Gagne [14] to do the classification. Gagne considers that learning outcomes can be classified into five types: **verbal information**, **intellectual skills**, **cognitive strategies**, **motor skills**, and **attitude**.

We will categorize capability into five types based upon Gagne's theory. Related definitions and modifications will be discussed in this section.

4.2 Definitions and Properties

We are going to modify definitions corresponding to the model we proposed according to those pedagogical theories. First we categorize capability into five types: **verbal information**, **intellectual skills**, **cognitive strategies**, **motor skills**, and **attitude**. Suppose there are *n* capabilities in a specific domain, each capability is one of the five types. We obtain C_{set} by grouping these capabilities together.

$C_{set} = \{c_1, c_2, \ldots c_n\}$, where each c_i has five dimensions: <vc_i, ic_i, cc_i, mc_i, ac_i>, and vc_i denotes verbal capability, ic_i denotes intellectual capability, cc_i denotes cognitive capability, mc_i denotes motor capability, and ac_i denotes attitude capability. With this new definition of C_{set}, our proposed algorithms are capable of searching five types of capabilities without further modifications. We give a few definitions for pedagogical need as follows.

Definition 1: $LnSty_i$ of AT_i

Each activity tree is accompanied with LnSty as a vector to represent its learning style in visual degree, auditory degree, and kinesthetic degree. Each value in this vector is between 0 and 1.

Definition 2: $LgOrg_i$ of AT_i

LgOrg, e.g. the default teaching order of the activity tree, indicates the logical organization of an activity tree. Teaching order is labeled in consecutive number in traditional course. We discretize the teaching order and map it to interval [0,1]. For example, LgOrg of activity tree is either one of {0.3, 0.6, 0.9}.

Definition 3: Student's learning style (SLS)

We create a vector, called student's learning style (SLS), for each student to characterize his/her learning style.

Step 3.1:

…

else if (AT_j has not been learned yet)
then (compute $CF_j^{(1)}$ and $CF_j^{(2)}$) and (mark the AT_j with *Candidate*)

…

where:

$$\begin{cases} CF_i^{(1)} = a\ NOW_i + ß\ SGP_i + g\ LgOrg_i, & a, b, g \text{ are constants.} \\ CF_i^{(2)} = SLS \bullet LnSty_i \end{cases}$$

Step 3.2:

if (∃ any AT with *Candidate* mark) *// select the AT with Candidate mark*
then
1: sort $CF_j^{(2)}$ in descending order, delete AT whose $CF_j^{(2)}$ is in the second half.
2: Select an AT_j with the highest CF_j and deliver it to the learner.

….

Figure 6: The Pedagogical Learning Process Algorithm

The first element represents level of visual characteristics, the second one represents level of audio characteristics, and the third one represents level of kinesthetic characteristics. We utilize this vector in activity tree selecting function to determine if the activity tree suits the student's learning style.

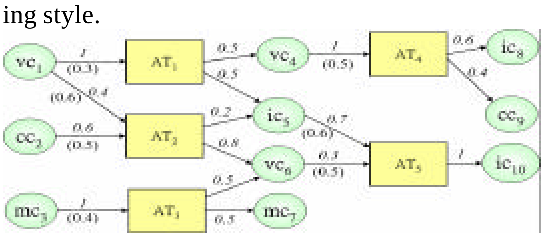

Figure 7: An Example of IAM with Pedagogical Theories.

4.3 Learning Process with Pedagogical Theories

We are going to make a few modifications to accommodate the algorithms to pedagogical theories. The modified learning process algorithm has only slight difference than the previous one. We use learner' s learning style as a criterion for selecting suitable activity tree for student's learning.

4.3.1 Learning Process Algorithm with Pedagogical Theories

In the section 3.3, Learning Process Algorithm selects activity tree based on SGP and NOW, now we propose a Pedagogical Learning Process Algorithm, which is extended form Learning Process Algorithm, to select AT by the criteria of pedagogical theories, i.e. logical organization and psychological organization. From our viewpoint, logical organization is characterized by the order of activity trees and psychological organization is characterized by student' s learning style, grade prediction and course objectives. The four characteristics are used as criteria for selecting AT. Pedagogical Learning Process Algorithm is to extend the Step3.1 and Step3.2, which are shown in Figure 6, of Learning Process Algorithm.

In Step 3.1, we compute $CF_j^{(1)}$ and $CF_j^{(2)}$ to represent characteristics of psychological organization and logical organization. We regard these two as partially, linearly dependent. Normalized Objectives Weight (NOW), Student' s Grade Prediction (SGP), and logical organization are mutually linearly dependent and can be computed by a simple linear function. Learning style, non-linearly dependent with the three, is computed by product of student' s learning style vector (SLS) and activity tree' s learning style vector (LnSty). The more similar these two vectors are, the higher the product is. Step 3.2 represents a selecting strategy, first we sort $CF_j^{(2)}$ in descending order and prune the latter half. Then we select the activity tree whose $CF_j^{(1)}$ is the highest. This strategy implies we consider psychological organization first and prune those invalid activity trees, and then we consider the physical organization to select the most suitable activity tree.

Table 2: Learning style and logical organization of each AT.

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

Since pedagogical objectives differ from learners, courses, teachers, etc., we can modify Step 3.1 to Step 3.2, if needed, to fulfill pedagogical needs. Different selecting strategies generate different sequencing results.

Example 2: Learning in IAM with pedagogical theories

We are going to show a simple example of learning in IAM with pedagogical theories. First we define IAM and related attributes of each activity tree; thereafter we' re going to demonstrate the process of Learning Process Algorithm for a specific student. The example of IAM is shown in Figure 7.

IAM in Figure 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}, $e'_{5,10}$})

The learning style and logical organization, being used in Pedagogical Learning Process Algorithm, are shown in Table 2. Note that the logical organization (LgOrg) of each activity tree is represented in numerical format. Numbers 0.3, 0.5 and 0.7 can be regarded as basic, medium, and advanced teaching resources, respectively. Suppose there' s a learner learning in this IAM, his/her personal information is

AC={(vc_1,0.5),(cc_1,0.8),(mc_1,0.1),(vc_2,0.6),(ic_2,0.6)},
SLS =<0.1, 0.2, 0.7>,
CO={(vc_3,0.3), (ic_2,0.4), (ic_3,0.6), (ic_4,0.5), (cc_2,0.6), (mc_2,0.5)}.

Since he/she had learned AT_1, Learning Process Algorithm will choose the next activity tree for his/her learning. $CF_i^{(1)}$ and $CF_i^{(2)}$ in Pedagogical Learning Process Algorithm are defined as

$CF_i^{(1)}=0.2\times NOW_i+0.2\times SGP_i+0.4\times LgOrg_i$,
$CF_i^{(2)}= SLS \bullet LnSty_i$, where the symbol " ● " represents the dot product.

To execute Learning Process Algorithm, we can compute NOW, SGP, $CF_i^{(1)}$, and $CF_i^{(2)}$ and other information as shown in Table 3.

Table 3: Selecting Criteria for Each Activity Tree.

	AT_2	AT_3	AT_4
PCL	{ic_2,vc_3,ic_4}	{mc_2,vc_3,ic_4}	{ic_3, cc_2}
NOW^{h1}	1	1	1
SGP	0.5×0.4+ 0.8×0.6=0.68	0.1×1=0.1	0.6×1=0.6
LgOrg	0.3	0.5	0.3
$SLS \bullet LnSty_i$	0.24	0.29	0.53
$CF_i^{(1)}$	0.456	0.42	0.44
$CF_i^{(2)}$	0.24	0.29	0.53

Then, we use the following strategy for selection: for smart students, select the AT with the highest $CF_i^{(1)}$ value; for remaining students, select the AT with the highest $CF_i^{(2)}$ value. With this strategy, we select AT_2 for smart students, AT_4 for remaining students. In addition, we can revise $CF_i^{(1)}$ and $CF_i^{(2)}$ for specific purposes. For example, some teachers believe that learning style have relations with student's grade, they can modify $CF_i^{(1)}$ and $CF_i^{(2)}$ as $CF_i^{(1)} = 0.5 \times NOW_i + 0.5 \times LgOrg_i$, $CF_i^{(2)} = 0.5 \times SGP_i + 0.5 \times (SLS \bullet LnSty_i)$. If the selecting strategy remains the same, we will provide AT_3 for smart students, and AT_4 for remaining students.

4.4 Estimation of the Power of IAM

We have shown that it is possible to apply pedagogical theories to our IAM for specific need. However, how many pedagogical theories can be applied to IAM? In this section, we estimate that how many different structures IAM can support for pedagogical needs.

Educational researchers proposed various types of course structure to facilitate student's learning. Posner [18] proposed three types of structures including discrete structure, linear structure, and hierarchical structure. Bruner [11] proposed concept of spiral curriculum. And Efland [13] proposed the lattice curriculum. Each structure satisfies certain kinds of pedagogical need. IAM we proposed can be applied to those course structures, as can be seen in Figure 8 and Figure 9.

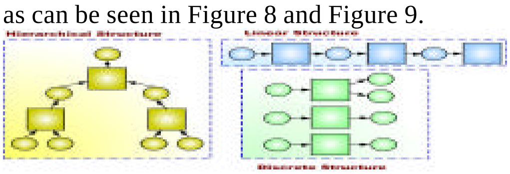

Figure 8: IAM Mapping to Discrete Structure, Linear Structure, and Hierarchical structure

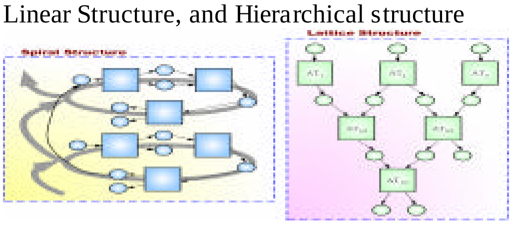

Figure 9: IAM Mapping to Spiral Curriculum and Lattice Curriculum

5. Implementation

As mentioned in previous sections, based upon concept of OOP and SCORM standard, we have proposed an Instruction Activity Model (IAM) which is composed of related AT components with inter-relation and specific attributes for pedagogical needs. In this section, we want to verify that proposed IAM is workable and compatible with SCORM standard, so we develop an IAM system which is based on SCORM 1.3 RTE [2]. As shown in Figure 10, the IAM system consists of 2 modules: 1. IAM module including IAM Controller, IAM Model, and IAM database, and 2. AT Tree Inference Engine Module including AT Parser and DRAMA Inference Engine. In IAM Module, the IAM Controller mainly manages the IAM Model and communicates with Controller and AT Tree IE Module. While learner logins system, IAM system will assign a Controller, which will manage all events during learning, to learner.

In AT Tree IE Module, the Inference Engine, called DRAMA, is the Object-Oriented Rule Base Management System (OORBMS) which was proposed based on Drama Model (DM) for constructing a reusable, sharable, and modifiable knowledge base [1][23][24][25]. The model manages rules under object-oriented paradigm. Moreover, the rule format of Simple Sequencing Specification possesses the general rule format as: if condition set Then action. This format can make us to easy adopt Drama Model to generate the sequencing rule set. During learning, the learner requests a course to learn and then the SCORM RTE 1.3 will send this request to the IAM Controller via Controller. The IAM Controller acquires learner's information and personal IAM Model and then selects an AT to inform AT Parser to extract and deliver rules to DRAMA Inference Engine. Finally, SCORM RTE receives the inference result of DRAMA and then deliver suitable course to learner.

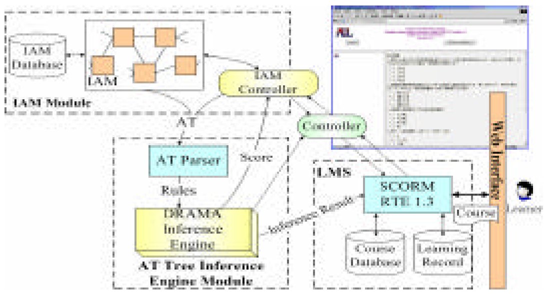

Figure 10: The Architecture of IAM System based on SCORM RTE 1.3

6. Conclusion

With vigorous development of the internet, e-learning system has become more and more popular. Currently, to solve the issue of sharing and reusing of teaching materials in different e-learning system, several international standard including SCORM, IMS, LOM, and AICC, etc. have been proposed by international organizations. Among international standards, Sharable Content Object Reference Model (SCORM) has become the most popular standard. In the Simple Sequence Specification (SSS) of SCORM, which relies on the concept of learning activities, defines the course sequencing behavior and organizes the content into a hierarchical structure, namely Activity Tree (AT). The Simple Sequencing process uses information about the desired sequencing behavior to control the sequencing, selecting and delivering of activities to the learner. Hence, how to create, represent and maintain the Activity Tree and associated sequencing are our concerns. For a large learning activity, the Activity Tree will become too complex to manage and reuse. It is hard to reuse and reintegrate ATs without the inter-relations. These defects will limit the scalability and flexibility of adaptive learning system. Moreover, for more personalized learning, many researches have used the Pedagogical Theory to enhance the evaluation of the personal learning characteristic.

Therefore, in this paper, we extend and modualize the structure of AT by applying Pedagogical Theory and concept of Object Oriented Methodology, respectively. Thus, we propose a novel model as graph, Instructional Activity Model (IAM), which is composed of related Activity Tree nodes. Based upon Pedagogical Theory, each AT node in IAM is modualized as a learning unit with inter-relations and specific attributes, which can be easily managed, reused, and integrated. We also propose two heuristic algorithms with general and specific domain heuristic to traverse IAM to generate the dynamic learning content to the learner. IAM with scalability and flexibility can apply different pedagogical theories for specific need by extension mechanisms. In addition, based on SCORM 1.3 RTE (Run Time Environment), we develop an IAM system to manage AT nodes and dynamically generate personalized SOCRM compliant course. Based upon the results of experiment, we may conclude that the IAM system is workable and compatible with SCORM standard. In the near future, we will enhance continuously our proposed IAM and apply it to some specific domains to evaluate its flexibility, scalability and learning performance.

7. Reference

[1] S. S. Tseng, G. J. Hwang, M. F. Jiang, C. J. Tsai, and Y.T. Lin, ***專家系統 導論/工具/應用***, **文魁資訊股份有限公司**, Taiwan, 2002.

[2] SCORM (Sharable Content Object Reference Model), http://www.adlnet.org/

[3] IMS (Instructional Management System), http://www.imsproject.org/

[4] LTSC (IEEE Learning Technology Standards Committee), http://ltsc.ieee.org/wg12/

[5] AICC (Aviation Industry CBT Consortium), http://www.aicc.org

[6] XML (eXtensible Markup Language), http://www.w3c.org/xml/

[7] W3C, World Wide Web Consortium, http://www.w3.org

[8] Edward R. Jones, SCORM, http://www.scorm.tamucc.edu/scorm/home.html

[9] Nelson L. Bassing, *Teaching in Secondary Schools*, Boston: Houghton Mifflin Company, pp. 51-55. 1963.

[10] Bengamin S. Bloom, Bertram B. Mesia, and David R. Krathwohl, *Taxonomy of Educational Objectives (two vols: The Affective Domain & The Cognitive Domain)*, New York. David McKay, 1964.

[11] J. S. Bruner, *The Process of Education*, Cambridge, MA: Harvard University Press, 1960.

[12] Vincenza Carchiolo, Alessandro Longheu, and Michele Malgeri, "Adaptive Formative Paths in a Web-based Learning Environment, " *Educational Technology & Society*, 5(4), 2002.

[13] A.D. Efland, "The Spiral and the Lattice: Changes in Cognitive Learning Theory with Implications," Studies in Art Education, 36(3), pp. 134-153, 1995.

[14] R.M. Gagne, "Learning Outcomes and Their Effects," American Psychologist, No.4, 1984.

[15] R.M. Gagné, L. Briggs and W. Wager, *Principles of Instructional Design*, 4th ed. Harcourt Brace Jovanovich, 1992.

[16] H. Gardner, *Frames of Mind*, New York: Basic Books Inc., 1983.

[17] K. 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, pp. 333-360, 2002.

[18] G. J. Posner, *Analyzing the Curriculum*, N.Y. : McGraw-Hill, 1992.

[19] Leonif Sheremetov and Adolfo Guzman Arenas, "EVA: An Interactive Web-based Collaborative Learning Environment," *Journal of Computers and Education*, February 2002.

[20] Sherry Chen, " A Congitive Model for Non-Linear Learning in Hypermedia Programmes," British Journal of Educational Technology, 33(4), pp.449-460, 2002.

[21] R. Sternberg and E. Grigorenko, *Styles of Thinking in the School*, Branco Weiss Institute, May 1998.

[22] Evangelos Triantafillou, Andreas Pomportsis, and Stavros Demetriadis, "The design and the formative evaluation of an adaptive educational system based on cognitive styles," Computers and Education, 41(1), pp. 87-103, August, 2003.

[23] C. J. Tsai, S. S. Tseng, and Y. C. Wu, "A New Architecture of Object-Oriented Rule Base Management System," *Proc. of Tools Asia'99*, Nanjing, China, Sep., 1999.

[24] C. J. Tsai, and S. S. Tseng (2002), "Building A CAL expert system based upon two-phase knowledge acquisition, " *Expert Systems with Applications* 22,pp. 235-248.

[25] Y.C. Wu (1999), *An Approach to Object-oriented Rule Base Management System*, master thesis, Department of Computer and Information Science, National Chiao Tung University.