Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Standards & Interfaces 28 (2006) 336–355

COMPUTER STANDARDS & INTERFACES

www.elsevier.com/locate/csi

Constructing SCORM compliant course based on High-Level Petri Nets

Jun-Ming Su[a], Shian-Shyong Tseng[b,*], Chia-Yu Chen[b], Jui-Feng Weng[b], Wen-Nung Tsai[a]

[a]*Department of Computer Science and Information Engineering, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu, Taiwan 300, ROC*

[b]*Department of Computer and Information Science, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu, Taiwan 300, ROC*

Received 30 December 2004; received in revised form 11 April 2005; accepted 11 April 2005
Available online 4 June 2005

Abstract

With rapid 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, Sharable Content Object Reference Model (SCORM) is the most popular standard among existing international standards. In SCORM standard, the Sequencing and Navigation (SN) defines the course sequencing behavior, which controls the sequencing, selecting and delivering of a course, and organizes the content into a hierarchical structure, namely Activity Tree (AT). However, the structures with complicated sequencing rules of Activity Tree (AT) in SCORM make the design and creation of course sequences hard. Therefore, how to provide a user-friendly authoring tool to efficiently construct SCORM compliant course becomes an important issue. However, before developing the authoring tool, how to provide a systematic approach to analyze the sequencing rules and to transform the created course into SCORM compliant are our concerns.

Therefore, in this paper, based upon the concept of Object Oriented Methodology (OOM), we propose a systematic approach, called *Object Oriented Course Modeling* (OOCM), to construct the SCORM compliant course. High-Level Petri Nets (HLPN), which is a powerful language for system modeling and validation, are applied to model the basic sequencing components, called *Object-Oriented Activity Tree* (OOAT), for constructing the SCORM course with complex sequencing behaviors. Every OOAT as a middleware represents a specific sequencing behavior in learning activity and corresponding structure with associated sequencing rules of AT in SCORM. Thus, these OOATs can be efficiently used to model and construct the course with complex sequencing behaviors for different learning guidance. Moreover, two algorithms, called *PN2AT* and *AT2CP*, are also proposed to transform HLPN modeled by OOATs into a tree-like structure with related sequencing rules in Activity Tree (AT) and package the AT and related physical learning resources into a SCORM compliant course file described by XML language, respectively. Finally, based upon the OOCM scheme, a prototypical authoring tool with graphical user interface (GUI) is developed. For evaluating the efficiency of the OOCM approach compared with existing authoring tools, an

* Corresponding author.
E-mail addresses: jmsu@csie.nctu.edu.tw (J.-M. Su), sstseng@cis.nctu.edu.tw (S.-S. Tseng), gis91520@cis.nctu.edu.tw (C.-Y. Chen), roy@cis.nctu.edu.tw (J.-F. Weng), tsaiwn@csie.nctu.edu.tw (W.-N. Tsai).

0920-5489/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2005.04.001

experiment has been done. The experimental results show that the OOCM approach is workable and beneficial for teachers/instructional designers.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Adaptive learning environment; High-Level Petri Nets (HLPN); SCORM; Course sequencing; Learning activity

1. Introduction

With rapid development of the Internet, in the past 10 years, e-learning system [18,19,28,30,31] has become more and more popular. However, because of the non-uniform formats of teaching materials in different e-learning systems, 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 uniformizing the teaching materials format, international organizations have proposed several standard formats including SCORM [13], IMS [6], LOM [8], AICC [1], etc. Based upon these standard formats, the teaching materials in different learning management systems can be shared, reused, and recombined. Among these international standards, 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 object, SCORM uses the metadata to specify the structure of every learning object and proposes the content aggregation scheme to package these objects with Extensible Markup Language (XML) [16,17] format.

At present, the Sequencing and Navigation (SN) [15] provided by SCORM 2004 (version 1.3) adopts the Simple Sequence Specification (SSS) of IMS [6] to define the course sequencing behavior, and Content in SN is organized into a hierarchical structure, namely Activity Tree (AT). The SN relies on the concept of learning activities, each of which may be described as an instructional event, events embedded in a content resource, or an aggregation of activities to describe content resources with their contained instructional events. The SN uses information about the desired sequencing behavior to control the sequencing, selecting and delivering of activities to the learner. Therefore, by this standard, the instructional experience of teachers can be shared and the intelligent approach for (semi-) automatic course or exercise sequencing can be developed.

However, it is hard to understand the complicated sequencing rules in SN much less using it to construct a SCORM course with desired learning guidance. Recently, although many SCORM authoring tools have been developed by commercial companies, unfortunately, these tools support SCORM 1.2 only, for example, the Authorware 7 of Macromedia [10], Click2learn Unveils SCORM 1.2 Resource Kit [3], Seminar Author of Seminar Learning System [14], Elicitus Content Publisher [4], and more other SCORM 1.2 compliant authoring tools found in [5].

The learning guidance of a course can be represented as a graph which is easier to be understood and created for teachers/authors. Accordingly, if we can provide an authoring tool for teachers/authors to edit the structure of course with sequencing behavior rules by graph representation and transform it into SCORM compliant file automatically, the teachers/authors will be willing to use and create the SCORM compliant course. Thus, how to provide a user-friendly authoring tool to efficiently construct SCROM compliant course becomes an important issue. However, before developing the authoring tool, how to provide a systematic approach to analyze the sequencing rules and to transform the created course into SCORM compliant are our concerns.

Therefore, in this paper, based upon the concept of Object Oriented Methodology (OOM), we propose a systematic approach, called *Object Oriented Course Modeling* (OOCM), to construct the SCORM compliant course. High-Level Petri Nets (HLPN), which is a powerful language for system modeling and validation [20–27], are applied to model the basic sequencing components, called *Object-Oriented Activity Tree* (OOAT), for constructing the SCORM course with complex sequencing behaviors. Every OOAT as a middleware represents a specific sequencing behavior in learning activity and corresponding structure with associated sequencing rules of AT in SCORM. Thus, based upon HLPN theory, these OOATs can be easily used to model and construct the course with complex

sequencing behaviors for different learning guidance. Moreover, because every OOAT represents a basic sequencing building block as a cluster in AT, we thus propose two algorithms, called *PN2AT* and *AT2CP*, to transform HLPN modeled by OOATs into a tree-like structure with related sequencing rules in Activity Tree (AT) and package the AT and related physical learning resources into a SCORM compliant course file described by XML language, respectively. Finally, based upon the OOCM scheme, a prototypical authoring tool with graphical user interface (GUI) is developed. For evaluating the efficiency of the OOCM approach compared with existing authoring tools, an experiment has been done. The experimental results show that the OOCM approach is workable and beneficial for teachers/instructional designers.

The main contributions of this paper are:

(1) Propose a systematic approach, called *Object Oriented Course Modeling* (OOCM), to generate adaptive learning course which is compatible with SCORM standard.
(2) Model the basic sequencing components as the middleware, called *Object-Oriented Activity Tree* (OOAT), which can be easily managed, reused, and integrated, based upon High-Level Petri Nets (HLPN) theory.
(3) Construct a SCORM compliant course by user-friendly graphic user interface (GUI), which can be executed on the SCORM RTE system, according to the proposed OOCM approach.

2. Related work

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

2.1. SCORM (Sharable Content Object Reference Model)

Among those existing standards for learning contents, SCORM, which is proposed by the U.S. Department of Defense's Advanced Distributed Learning (ADL) organization in 1997, is currently the most popular one. The SCORM specifications are a composite of several specifications developed by international standards organizations, including the IEEE LTSC [8], IMS [6], AICC [1], and ARIADNE [2]. 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 *Reusable*, *Accessible*, *Interoperable*, and *Durable* (*RAID*) [9]. At present, the Sequencing and Navigation (SN) [15] in SCORM 1.3 (or called SCORM 2004) adopting the Simple Sequencing Specification of IMS relies on the concept of learning activities, each of which may be described as an instructional event, events embedded in a content resource, or an aggregation of activities to describe content resources with their contained instructional events. Content in SN is organized into a hierarchical structure, namely activity tree (AT) as a learning map. The example of AT is shown in Fig. 1. Each activity in the Activity Tree includes two data models: Sequencing Definition Model (SDM) including an associated set of desired sequencing behaviors of content designer and Tracking Status Model (TSM) including the information about a learner's interaction with the learning objects within associated activities. The SN uses information in SDM and TSM to control the sequencing, selecting and delivering of activities to the learner. The sequencing behaviors describe how the activity or how the children of the activity are used to create the desired learning experience. SN enables users to share not only learning contents, but also intended learning experiences. It also provides a set of widely used sequencing method so that the teacher could do the sequencing efficiently. However, how to create, represent and maintain the activity tree and associated sequencing definition is an important issue.

2.2. Other related research

Because the complicated sequencing rule definitions of SN in SCORM 2004 make the design and creation of course hard, the article in Ref. [11] has proposed several document templates to construct SCORM compliant course according to the sequencing definitions of SN. Teachers/authors can design their desired learning activities by modifying the sequencing definitions in document templates. Then, the SCORM course with sequencing definitions can be created by programming. However, for teachers/

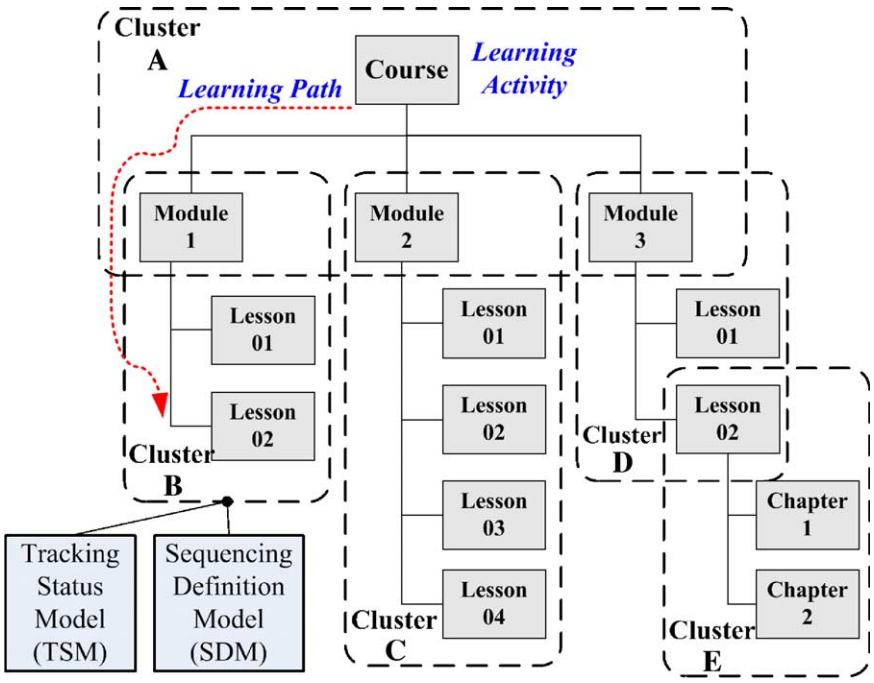

Fig. 1. An example of Activity Tree (AT) with clusters.

authors, creating the SCORM course with sequencing behavior rules by document templates is still hard. Moreover, it is time consuming and costs much to create SCORM course by programming.

Moreover, an open source tool, called Reload Editor, developed by Ref. [12] can be used to create the SCORM 2004 course. For setting the learning guidance, users have to edit the sequencing rules by clicking in the comboBox of sequencing rules. Although it offers the graphical user interface (GUI) to create SCORM course, the sequence of final course is hard to image and creating course is also time-consuming. Timothy et al. [29] also proposed a collaborative courseware authoring tool to edit the SCORM compliant course which can support collaborative authoring and suggest an optimal learning sequence. They analyzed the metadata of SCA in SCORM 1.3 to design the activity rules which can be used to generate lecture sequencing. This tool also offers users the sequencing rules definition page to define the sequencing behavior of courseware. Besides, Yang et al. [32] developed a web-based authoring tool, called Visualized Online Simple Sequencing Authoring Tool (VOSSAT), to provide an easy-to-use interface for editing existing SCORM-compliant content packages with sequencing rules. Nevertheless, the disadvantages in Refs. [29,32] are the same as Reload Editor [12].

Lin [25] applied Petri Nets theory to model online instruction knowledge for developing online training systems. Two-level specialized Petri nets including TP-net, which represents goal-oriented training plans, and TS-net, which represents task-oriented training scenarios, are proposed. A Goal-Oriented Training Model Petri net (GOTM-net), which is combined by a TP-net and all TS-nets, is converted as a set of "*if–then*" rules representing the behaviors a learner may perform and the corresponding responses. However, GOTM-net may not be compatible with SCORM standard. Based on SCORM 1.2, Liu [26] discussed meta-data structure which makes a base for reusing and aggregating learning resources in e-learning, and provided an aggregation model, called Teach net, based on High-Level Petri Nets (HLPN). Several routing constructs in workflow are also modeled by HLPN for flexible navigation. However, the Teach net is mainly used to model the content aggregation without considering course sequencing. Besides, the mod-

eled routing constructs may not be sufficient for modeling sequencing definition in SCORM 2004.

3. Object Oriented Course Modeling (OOCM)

As mentioned above, the structures with complicated sequencing rules of activity tree in SCORM make the design and creation of course sequences hard. Therefore, how to provide a user-friendly authoring tool, which can represent the course as a graph and transform it into SCORM compliant course file automatically, to efficiently construct SCROM compliant course becomes an important issue. However, before developing this kind of authoring tool, how to provide a systematic approach to analyze the sequencing rules and transform the created course into SCORM compliant are our concerns. Therefore, in this paper, we apply the High-Level Petri Nets (HLPN), which is a powerful language for system modeling and validation, to model the basic sequencing components as the middleware, called *Object-Oriented Activity Tree* (OOAT), for constructing the SCORM course with complex sequencing behaviors. Thus, according to these OOATs, we can model a complex structure of course with different learning guidance. Then, two transformation algorithms are also proposed to transform the created course into SCORM compliant one described by XML language. Fig. 2 shows the idea of Object Oriented Course Modeling (OOCM) approach.

3.1. The scheme of OOCM

Based upon the concept of Object-Oriented Methodology (OOM) and High-Level Petri Nets (HLPN) theory, we can model several basic sequencing components with specific sequencing behaviors in SN, which can be easily used to model complex structure of course. Therefore, in Fig. 3, the OOCM process includes four processes as follows:

(1) *OOAT modeling with HLPN*: Apply HLPN to model five basic sequencing components as the middleware with corresponding structure of AT and specific basic sequencing behaviors, called *Object-Oriented Activity Tree* (OOAT).
(2) *Course construction with OOAT*: Use these basic sequencing components (OOAT) to model complex structure of course with different learning guidance based upon the HLPN theory.

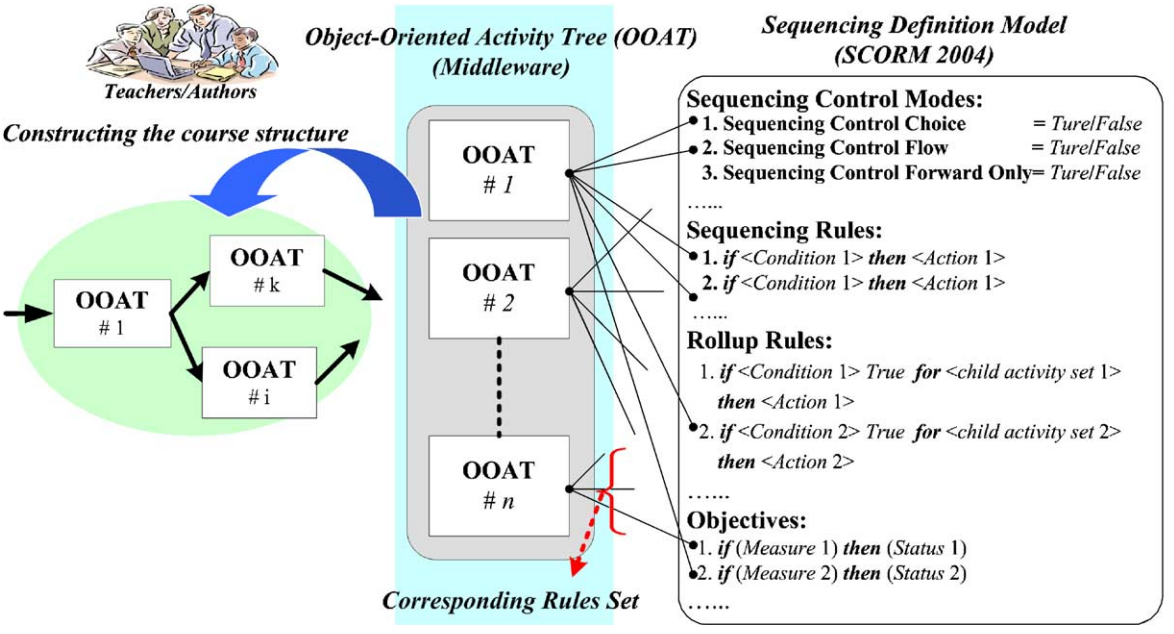

Fig. 2. The idea of Object Oriented Course Modeling (OOCM).

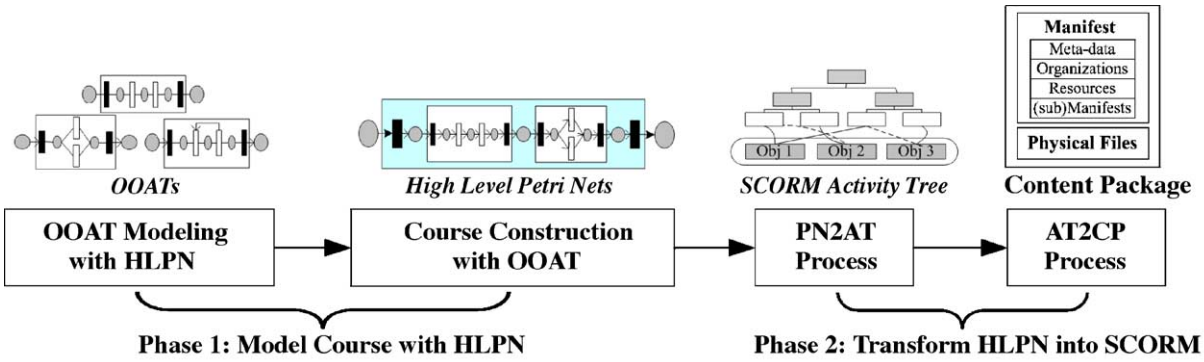

Fig. 3. The flowchart of Object Oriented Course Modeling (OOCM).

(3) *PN2AT process*: Transform the modeled course structure into tree-like SCORM-compliant AT with sequencing definition of SN.
(4) *AT2CP process*: Package the transformed AT structure with corresponding physical learning resources and then generate the content packaging course of SCORM.

3.2. The OOAT modeling with High-Level Petri Nets (HLPN)

As shown in Fig. 1, an AT in SCORM 2004 is structured by a set of clusters. A cluster, the basic sequencing building block, is an organized aggregation of activities consisting of a single parent activity and its first level children, but not the descendants of its children. The parent activity of a cluster will contain the information about the sequencing strategy for the cluster. The status information of all child activities will be collected and used to sequence these activities in the structure. Each cluster has a *Sequencing Definition Model* (SDM) to define a set of elements that can be used to describe and affect various sequencing behaviors. In this paper, we only take six out of ten rule definitions in SDM into account, that is, 1) Sequencing Control Modes, 2) Sequencing Rules, 3) Rollup Rules, 4) Objectives, 5) Objective Map, and 6) Delivery Controls, because these six rule definitions can perform the most of sequencing behaviors in SN. Therefore, we apply HLPN to model several basic sequencing components as a cluster with corresponding structure of AT and specific basic sequencing behaviors, called OOAT, which can be used to model a complex structure of a course. Thus, based upon these OOATs and OOCM approach, the remaining rule types in SDM could be analyzed and modeled in a similar way. Here, an OOAT can be represented as a Chapter or Section. For modeling the sequencing behaviors in SN, firstly, the OOAT in HLPN is defined as follows:

Definition 1. The HLPN of Object-Oriented Activity Tree (OOAT) is a 6-tuple
OOAT=(P, T, Σ, A, G, E), where

1. $P=\{p_1, p_2, \ldots, p_m\}$ is a finite set of places. P includes five types of places: P_G denotes the global objectives, P_L denotes the local objectives, P_M denotes the connector between transitions, P_R checks whether the transition executes the Rollup process or not, and P_W checks whether the transition defines the global objective (P_G) or not. Besides, in connective places (P_M), we use P_{in} and P_{out} to represent the starting place and ending place of an OOAT component. P_G and P_L contain tokens recording the information in Tracking Status Model (TSM).
2. $T=\{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions ($P \cap T=0$). T includes four types of transitions: T_A denotes a learning activity or a sub-OOAT component, T_M denotes the connector between OOAT components, T_R rolls up all learning status of its children, and T_O will set the global objective (P_G) of an activity according to its local objective (P_L).
3. $\Sigma=<C_{TSM}, C_O>$is the non-empty finite color sets of tokens. C_{TSM} represents the *Tracking Status Model* (TSM) in SN, which records the learning information of *Activity Progress Information*, *Attempt Progress Information* and *Ob-*

Table 1
The arc expression function $E(a)$ and its related token color

Arc expression function	Token
$E(\overrightarrow{P_G T_A})$, $E(\overrightarrow{P_G T_M})$	$<C_O+C_{TSM}>$
$E(\overrightarrow{T_A P_L})$, $E(\overrightarrow{T_R P_L})$, $(\overrightarrow{P_L T_R})$, $E(\overrightarrow{P_M T_R})$, $E(\overrightarrow{T_R P_M})$, $E(\overrightarrow{T_O P_G})$, $E(\overrightarrow{P_L T_O})$, $E(\overrightarrow{P_L T_A})$	$<C_{TSM}>$
$E(\overrightarrow{T_A P_M})$, $E(\overrightarrow{P_M T_A})$, $E(\overrightarrow{T_A P_G})$, $E(\overrightarrow{T_M P_G})$, $E(\overrightarrow{P_M T_M})$, $E(\overrightarrow{T_M P_M})$, $E(\overrightarrow{P_W T_O})$, $E(\overrightarrow{P_R T_R})$	$<C_O>$

jective Progress Information of learners. C_O denotes the *ordinary color*, corresponding tokens without information, which is applied to initialize or trigger a learning process.

4. $A \subseteq (P \times T) \cup (T \times P)$ is a finite set of directed arcs. $\overrightarrow{PT}$ is the arc from a place to a transition; $\overrightarrow{TP}$ is the arc from a transition to a place.
5. G: is *a guard function*. The firing rule $G(t)$ of a transition ($t \in T$) is defined as "*if–else*" form in SDM. The guard function can generate specific sequencing behaviors. In OOAT, we define the following guard functions:
   - $G(T_A)$: define the sequencing rules of SDM and specify whether a learner is ready or not to learn the activity according to her/his learning results in previous activity.
   - $G(T_R)$: control the rollup process of an activity based upon the Rollup rules definition of SDM.
   - $G(T_O)$: set the learning status of the global objective according to local objective of activity (T_A). In SDM, teachers can define how to read/write a global objective for different course sequencing.
6. E: is an arc expression function. $\Sigma(a)$, $\forall a \in A$, denotes how many and which kinds of token colors should be removed from the input places and added to the output places. In OOAT, we define the expression functions as shown in Table 1.

In addition, Fig. 4 shows the basic diagram of HLPN of OOAT. As mentioned in Definition 1, the connectors, P_M and T_M, pass the token only, T_R enabled by P_R with ordinary token $<C_O>$ executes the rollup process according to the token $<C_{TSM}>$ carrying the learning information. Besides, in the right part of Fig. 4, T_O will change the type of a place, e.g., P_{out}, into P_G if P_W has ordinary token $<C_O>$.

According to the sequencing behaviors in SN specification, we propose five OOAT components, 1. *Linear*, 2. *Choice*, 3. *Condition*, 4. *Loop*, and 5. *Exit*, to model different learning strategies. Fig. 5 shows these five basic sequencing components of OOATs with its corresponding structures of courses and related definitions of Guard functions, and Table 2 shows their related *Sequencing Definition Model* (*SDM*) including *Sequencing Control Mode (SCM)* which controls the navigation behaviors, *Objective* which defines the requirements of evaluated conditions, and *Sequencing Rules* which define the evaluated conditions of course sequencing during learning activity. Here, every *guard function* of OOAT can be mapped to

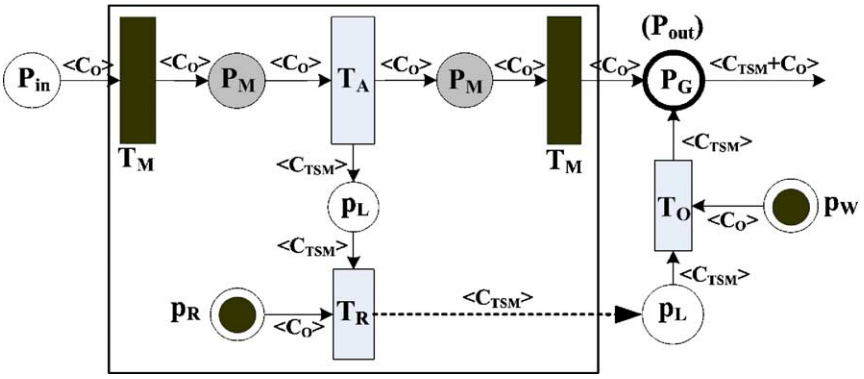

Fig. 4. The diagram of HLPN of OOAT.

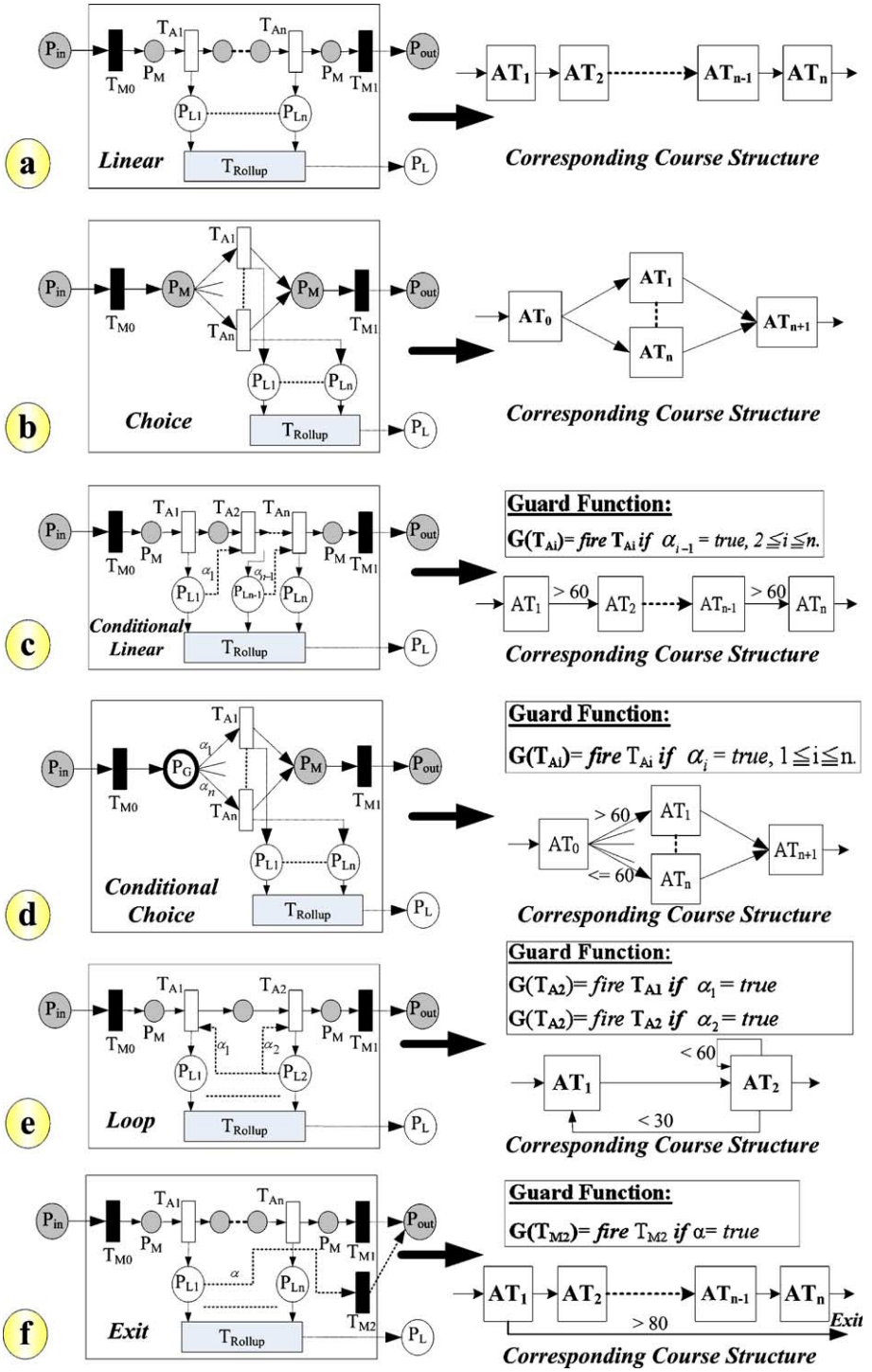

Fig. 5. The five sequencing components of OOATs.

Table 2
The related SDM definition of OOAT

OOAT types	Sequencing control mode	Objective	Sequencing rules
Linear	Flow=true Forward only=true Choice exit=true		
Choice	Choice=true Choice exit=true		
Conditional Linear	Flow=true Forward only=true Choice exit=true	Objective: • Satisfied by measure=true • Minimum satisfied Normalized measure=α_i	Postcondition Rule: • If α_i=true *then* continue *else* retry, $1 \leqq i \leqq n-1$
Conditional Choice	Flow=true Choice=true Choice exit=true	Objective: • Satisfied by measure=true • Target objective ID=OBJ $\mathbf{P_G}$ • Read satisfied status=true • Read normalized measure	Precondition rule: • T_{A1}: *Read* OBJ P_G (global objective) • If $\alpha_1 \neq$true *then*Hidden From Choice, $1 \leqq i \leqq n$
Loop	Flow=true Choice exit=true	Objective: • Satisfied by measure=true • Minimum satisfied normalized measure=α_1/α_2	Postcondition rule: • T_{A2}: if α_1/α_2=true *then* previous / retry *else* continue
Exit	Flow=true Forward only=true Choice exit=true		Postcondition rule: • T_{A1}: *if* α=true *then* Exit Parent / Exit All

corresponding *sequencing rules* in SDM, which record the sequencing behaviors of learning activity in SCORM AT. In Fig. 5, the *Linear* OOAT (5a) denotes that the learners can learn the activity (transition) straightforward. Therefore, "*Sequencing Control Flow*" in SCM is set as true. The Rollup transition (T_{Rollup}) will collect the status information of related local objective places (P_L) in included child transitions (activities) to evaluate the value of P_L in parent transition. The *Condition* OOAT includes *Conditional Linear* (5c) and *Conditional Choice* (5d). The former is a *Linear* OOAT with conditional criteria (α) that checks whether an activity will be assigned to a learner or not according to his/her learning result in previous activity. For example, in Fig. 5c, the token, $<C_{TSM}>$, will be delivered to the local objective (P_{L1}) after learning the activity (T_{A1}). Then, according to the activity's tracking information (TSM) and related guard function in T_{A2}, the next transition (T_{A2}) may be accessible (fired) if the condition α_1 is true. The latter is similar to the *Choice* component. According to the previous learning status stored in global objective P_G, an activity (T_{Ai}) can be selected by learners if its conditional criterion (α_i) is true. Fig.

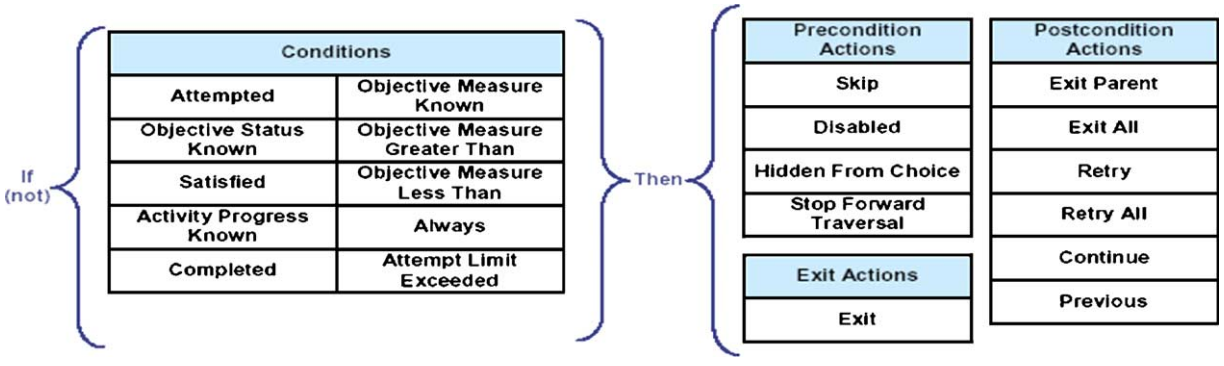

Fig. 6. The structure of sequencing rules.

Table 3
The action types and corresponding OOATs of *precondition* in sequencing rules

Action element	Description	OOATs
Skip	This action will omit an activity to be learned.	Conditional Choice
Disabled	This action will block an activity to be learned.	Conditional Linear
Stop forward traversal	This action will terminate learners to continuously navigate learning activity forward.	Conditional Linear
Hidden from choice	This action will stop the choice of activity.	Conditional Choice with "Sequencing Control Choice" is false.

5e shows the *Loop* OOAT which can control the learners to study continuously the same activity or previous one according to the conditional criteria (α_1 and α_2). In addition, in Fig. 5f, the *Exit* OOAT controls the termination of learning process. For example, after learning the T_{A1}, the token,<C_{TSM}>, will be delivered to P_{L1}. Then, according to the tracking information of T_{A1}, learners will finish the component if the condition α is true.

3.3. Sequencing rules modeling of SDM

In SDM, each Sequencing Rule consists of a set of conditions and a corresponding action in *if [condition_set] then [action]* format. A sequencing behavior of activity associated with the rule's action will be executed if the rule's condition-set evaluates to True. Thus, different definition of sequencing rules will result in different learning guidance. However, how to define the appropriate sequencing rules within course is an important issue. Therefore, in this section, we define these sequencing conditions as tokens used to determine whether an activity is accessible or not, e.g., symbol "α_i" in Fig. 5. Besides, the OOATs are used to model the rule's actions for modeling the sequencing behaviors of SCORM course. The structure of a sequencing rule is shown in Fig. 6.

In SN specification, the sequencing rules of SDM include the following rule's actions:

(1) *Precondition actions*: decide whether an activity will be selected or not for learning. These actions will be executed while an activity will be selected. Its action elements and corresponding OOATs are shown in Table 3.
(2) *Postcondition actions*: control the sequencing flow according to learning result of learners after learning an activity. These actions will be executed while an activity has been finished. Its action elements and corresponding OOATs are shown in Table 4.
(3) *Exit actions*: will be executed after a descendant activity has been finished or some condition is satisfied. It is controlled by a SCORM complaint learning management system (LMS). Thus, we can set the system commend, *Exit*, to inform LMS for finishing the whole course.

Fig. 7 shows the example of *Skip* action modeled by *Conditional Choice* OOAT, which represents that if the rule condition α is false, the activity T_{A1} will be skipped and then the T_{A2}, which does not execute any learning activity, will be triggered according to the definition of guard function. The *Disabled* Action can also be modeled by *Conditional Linear* OOAT as

Table 4
The action types and corresponding OOATs of *postcondition* in sequencing rules

Action element	Description	OOAT
Exit Parent	This action terminates an activity.	Exit
Exit All	This action terminates whole activity tree (course).	Exit
Retry	This action makes learner to relearn some previous activities if its condition is evaluated as true.	Loop
Retry All	This action makes learners to relearn all previous activities if its condition is evaluated as true.	Loop
Continue and Previous	This action makes learners to learn next or previous activity, respectively.	Conditional Linear and Loop

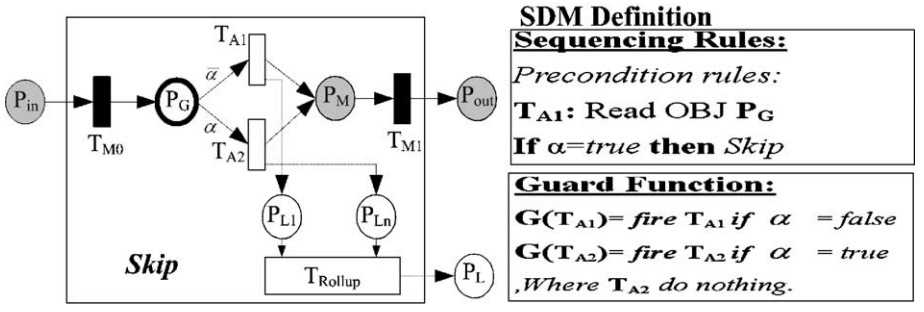

Fig. 7. An example of modeling *Skip Action* in sequencing rules by *Conditional Choice* OOAT.

shown in Fig. 5c. In postcondition actions, the *Exit Parent* action can be modeled by *Exit* component shown in Fig. 5f. For *Retry* action in Fig. 5e, the token<C_{TSM}>of T_{A2} is delivered to P_{L2}. Then, T_{A2} will be relearned if condition α_2 is true according to its learning status of local objective (P_{L2}).

3.4. Objective modeling

In SN, each activity has many associated learning objectives which include two types: *local objectives* and *global objectives*. The local objective which can only be referred by its associated activity and the global objective which can be shared between activities for the more complex instructional designs define how to evaluate an activity's *objective progress information*. Therefore, in OOATs, each transition (activity) has one local objective (P_L) and global objective (P_G) which will be defined if necessary. As shown in Fig. 8, in general, the transition (T_{A1}) only has one local objective (P_L) and no global objective. Here, the "Minimum Satisfied Normalized Measure=0.6" means that the score of learner must exceed 0.6. After learning T_{A1}, a Token<C_{TSM}>with *Objective Progress Information* of T_{A1} is delivered to P_L for recording the related learning information. Then, if P_w is assigned an ordinary Token<C_O>, the T_O will set the connector transition (P_M) as a global transition (P_G) for sharing the learning results with another transition (T_{A2}). Then, according to guard function

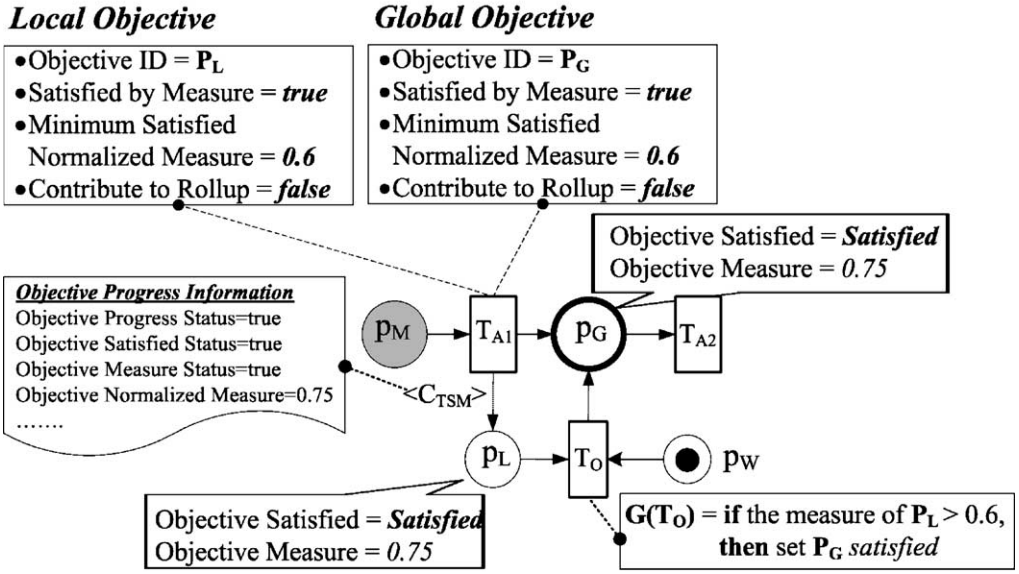

Fig. 8. The process of objective reference.

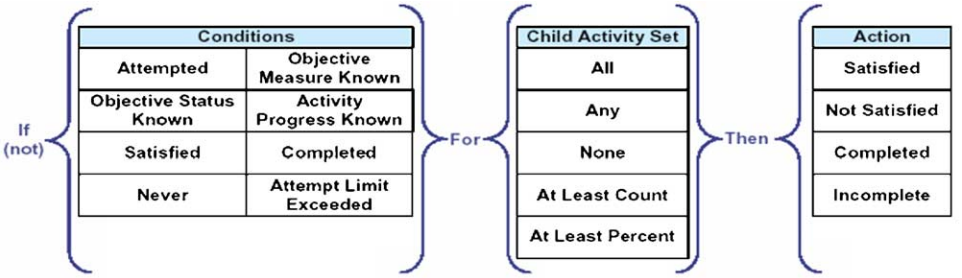

Fig. 9. The structure of rollup rules.

$G(T_O)$, the P_G will be set as *satisfied* because the score (0.7) is greater than 0.6.

3.5. Rollup rule and delivery control modeling

In SN, cluster activity, which is the basic sequencing building block, can be applied with a set of zero or more Rollup Rules which are evaluated during the overall Rollup Process. Each Rollup Rule is defined as "*if [condition_set] True for [child activity set] then [action]*" format, which denotes that if the set of conditions (*condition_set*) evaluates to True from the tracking information of included child activities (*child activity set*), corresponding action (*action*) will set the cluster's tracking status information. Fig. 9 shows the structure of a Rollup Rule.

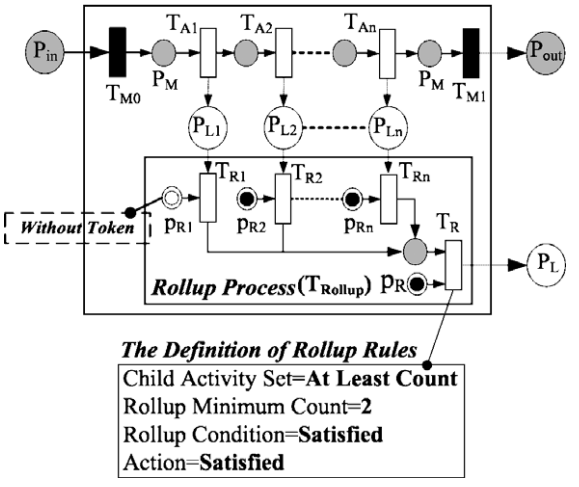

Fig. 10. The rollup model in OOATs.

As mentioned above, in OOATs, we use the T_{Rollup} transition to process the Rollup rules for evaluating the learning results of learners in a cluster. The T_{Rollup} transition can be modeled by HLPN as shown in Fig. 10. Here, in T_{Rollup}, each T_R transition will evaluate the learning status recorded in associated local objective (P_L) if its P_R transition is marked by an ordinary token<C_O>, where P_R transition enables or disables the Delivery Controls, which is used to manage the activity's tracking status information, in SDM. For example, in Fig. 10, because the P_{R1} of T_{A1} is not marked by a token<C_O>, T_{A1} will not be triggered but others with Tokens will be triggered to execute the rollup process. Moreover, according to the definition of Rollup Rules, the learning status of OOAT will be set as *satisfied* in P_L if at least two activities (transitions) within it are *satisfied*.

4. Activity tree transformation process

In Section 3, we have described how to model the HLPN model of course sequences in SCORM by our proposed OOATs. Therefore, in this section, how to transform the HLPN model into SCORM compliant course will be described. In this paper, we propose two algorithms, called *PN2AT* (Petri Nets to Activity Tree) and *AT2CP* (Activity Tree to Content Package), to do the activity tree transformation process.

4.1. PN2AT process

In OOAT, each transition with included child transitions can be represented as a cluster of AT in

SCORM. Thus, an algorithm, called *PN2AT*, transforms each non-terminated transition into a cluster with associated sequencing definitions in SDM and integrates them to construct the structure of AT. For example, in Fig. 11, an HLPN model of course can be decomposed as a hierarchical structure. In every level, a non-terminated transition, e.g., T_{A1}, will be represented as a root-node (AA) and included sub-transitions (T'_{A1} and T'_{A2}) will be represented as the child nodes (AAA and AAB), which form a tree-like structure as a cluster with associated sequencing definition of SDM in AT. Then, we can recursively transform all non-terminated transitions by the same process.

Algorithm: PN 2AT Algorithm

Definition of symbols:
AT_F: denote the final AT with XML code.
C_i: denote a tree-like cluster.
Input: The HLPN model of a course
Output: AT_F

Step 1: For each $T_i \in$ HLPN model
1.1: If T_i is a non- terminated transistion *then* create a tree-like cluster C_i
1.2: Insert T_i as root node and its included sub-transitions T_k as child nodes into C_i
1.3: Generate the corresponding XML codes according to its structure type of OOAT and the sequencing definitions including *Sequencing Control Mode*, *Sequencing Rules*, *Rollup Rules*, and *Objective definitions* in SDM for C_i into appropriate position of AT_F.
1.4: If $\exists\ T_k \in C_i$ is a non-terminated transition *then* execute recursively the same processes as Step 1.1.
Step 2: Output the AT_F

4.2. AT2CP process

After transforming the HLPN model of a course by PN2AT process, the structure and sequencing definitions of SCORM course without physical learning resources can be generated. Therefore, according to content packaging scheme of SCORM, an algorithm, called AT2CP, will be used to package the structure of AT and its related physical learning resources into a SCORM compliant course file described by XML language. The AT2CP process is also shown in right side of Fig. 11.

Algorithm: AT2CP Algorithm

Definition of symbols:
PF: denote a temporary place which collects related physical learning resources of AT.
CP: denote the contents package file of SCORM.
Input: Activity Tree (*AT*) generated by PN2AT algorithm.
Output: Content Package (*CP*)

Step 1: For each leaf node in AT
1.1: Retrieve the related physical learning content to store in PF according to its information of learning resource.
1.2: Generate the corresponding XML code including <resource>, <file>, etc. to integrate the leaf node and its learning resources.
Step 2: Generate the *manifest* file which describes the structure of course and related learning resources.
Step 3: Package the *manifest* file and *PF* into the *CP*;
Step 4: Output the *CP*

4.3. Example of Object Oriented Course Modeling (OOCM)

In this paper, we use the course "Photoshop" as experimental example, which is released by ADL SCORM organization, to show the process of Object Oriented Course Modeling (OOCM).

Fig. 12a is the HLPN Model of Photoshop course created by 6 OOATs and Fig. 12b shows its corresponding AT structure transformed by PN2AT and AT2CP processes. Its creating steps are described as follows:

Step 1: Select a *Linear* $OOAT_1$ for creating a course structure with 4 learning activities (node).

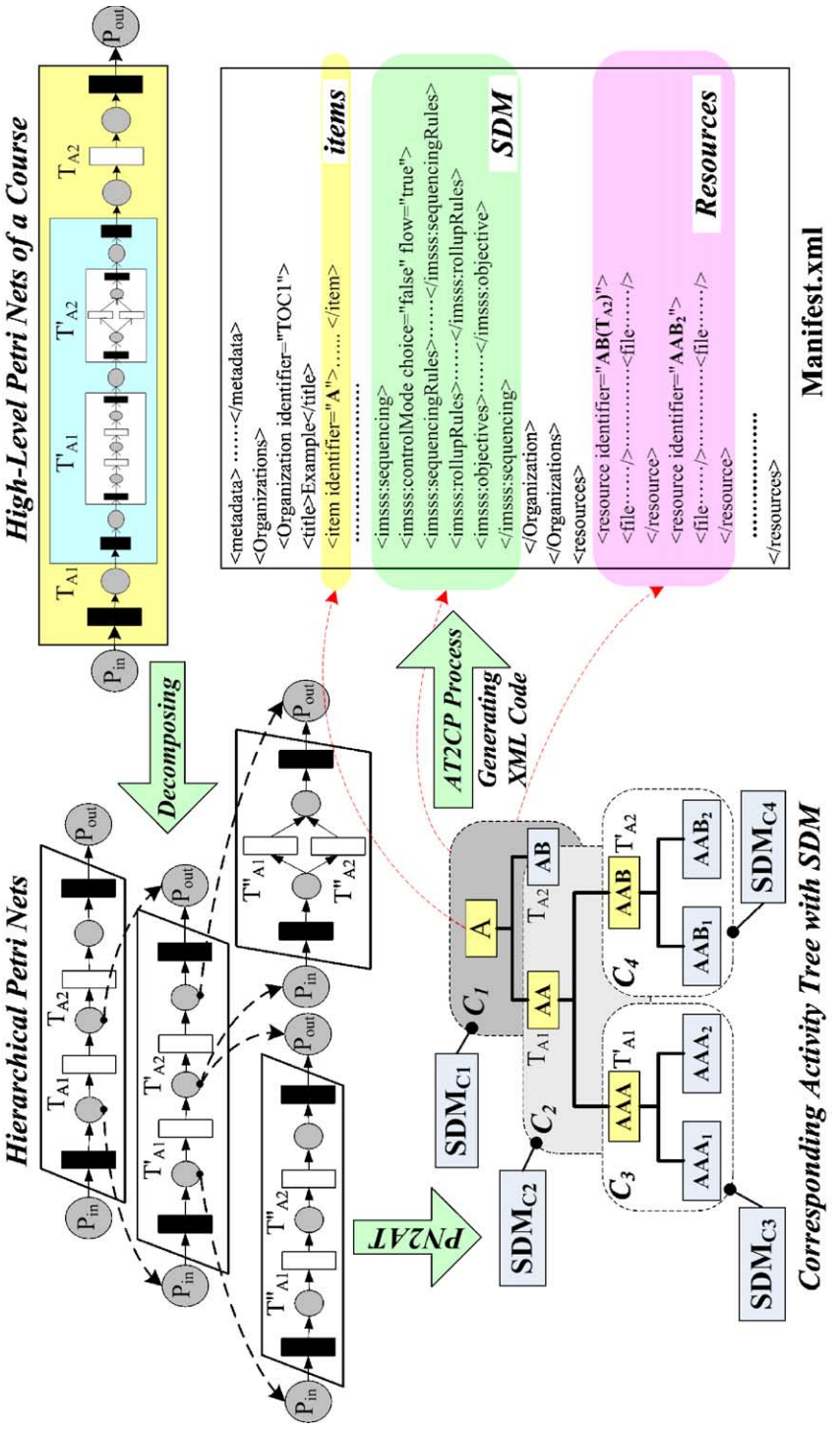

Fig. 11. An example of PN2AT process and AT2CP process.

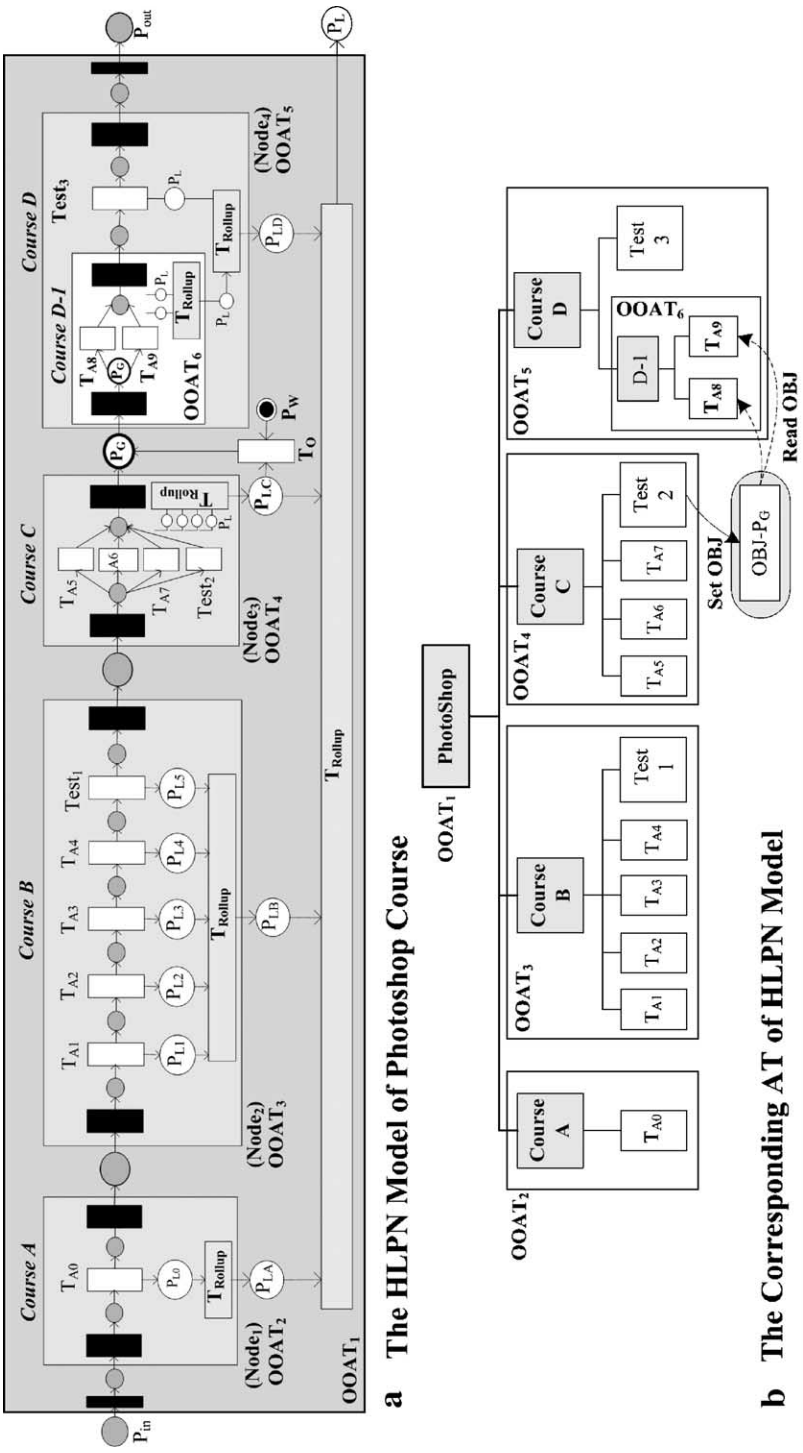

Fig. 12. The HLPNs Model and AT structure of course "PhotoShop".

Step 2: Insert a *Linear* $OOAT_2$ into *Node* 1 and *Linear* $OOAT_3$ into *Node* 2 in $OOAT_1$ for creating the *Course A* and *B*, respectively.
Step 3: Insert a *Choice* $OOAT_4$ into *Node* 3 in $OOAT_1$ for creating the *Course C* and then set that the $Test_2$ node will write its testing result into *Global Objective* P_G so the P_W with token C_O enables the T_O to set P_G according to the learning result in P_{LC}.
Step 4: Insert a *Linear* $OOAT_5$ into *Node* 4 in $OOAT_1$ and then insert a *Conditional Linear* $OOAT_6$ into $OOAT_5$ for creating the *Course D*-1. The $OOAT_6$ will read *Global Objective* P_G and then select different learning activities (T_{A8} or T_{A9}) for learners according to the testing result of *Course C*.

5. Implementation of the OOCM authoring tool

In this section, based upon the OOCM scheme, a prototypical authoring tool is developed. It can provide users with graphical user interface (GUI) to efficiently construct the learning activity structure with desired sequencing behaviors and then transform learning activity into SCORM compliant course.

5.1. The prototypical framework of OOCM authoring tool

As shown in Fig. 13, for constructing a SCORM compliant course, the OOCM authoring tool including 3 functional components, an OOATs Library, and a Learning Object Pool are described as follows:

(1) Learning object importer: import the existing learning resource within SCORM course or user-defined learning objects into the learning object pool.
(2) Course sequencing constructor: provide the teacher/instructional designer to construct a complex graph based course structure by inserting OOAT selected from OOATs Library.
(3) SCORM content package transformer: transform the graph based course structure into Activity Tree with related sequencing rules and then package its related learning resources into SCORM compliant course, based upon PN2AT and AT2CP algorithms.

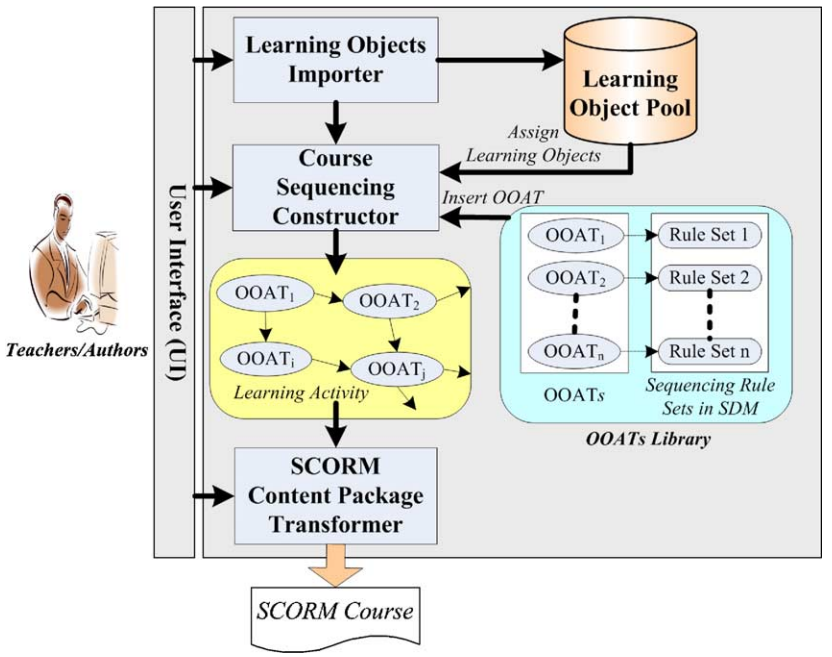

Fig. 13. The prototypical architecture of OOCM authoring tool.

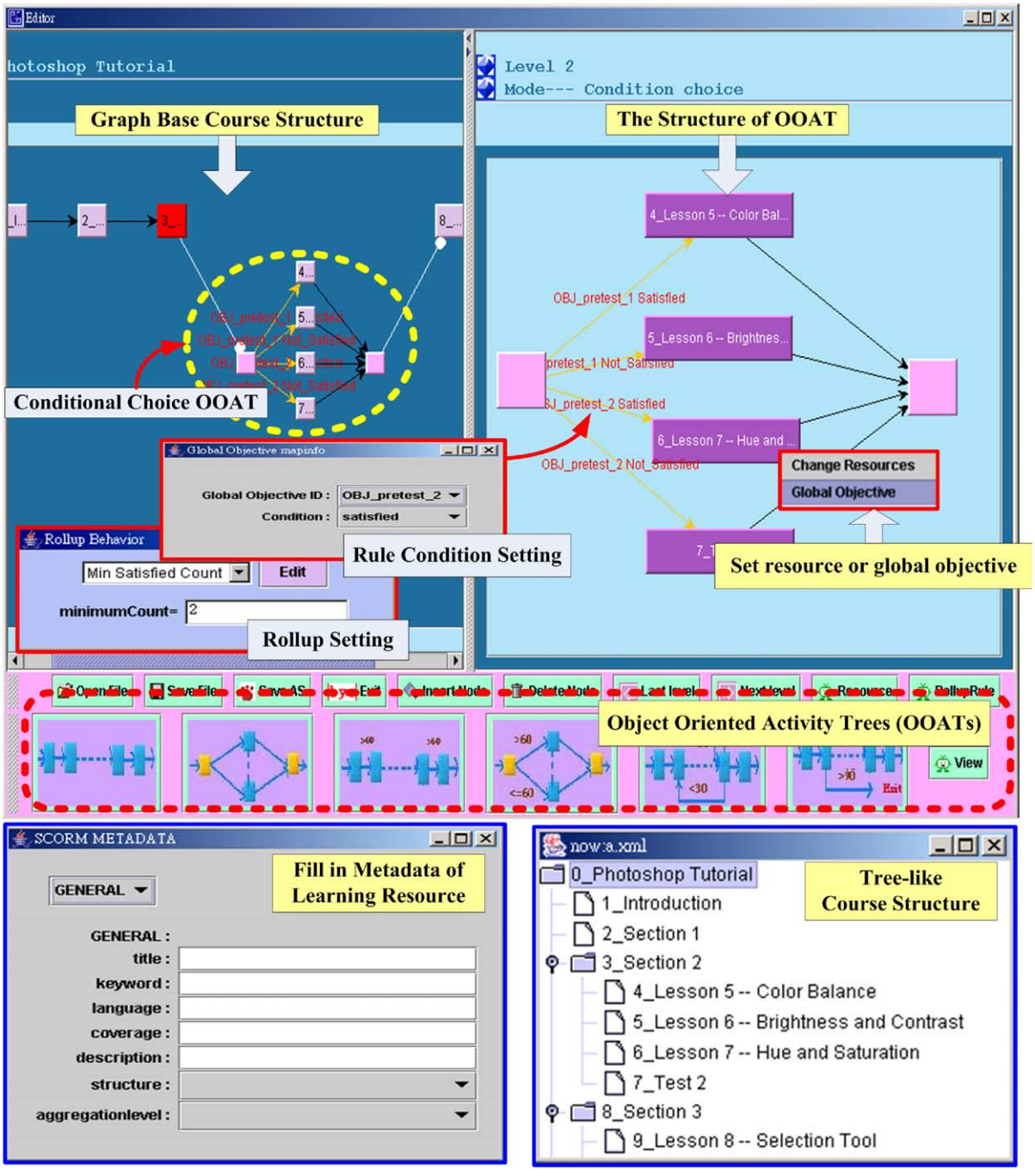

Fig. 14. The screenshot of the OOCM authoring tool.

Here, we describe and show the screenshot of OOCM authoring tool for constructing a SCORM compliant course by OOATs. The Authoring Tool is developed based on Java language and JGraph graphic tool [7] running on Windows operation system. The Fig. 14 is the screenshot of OOCM authoring tool. The

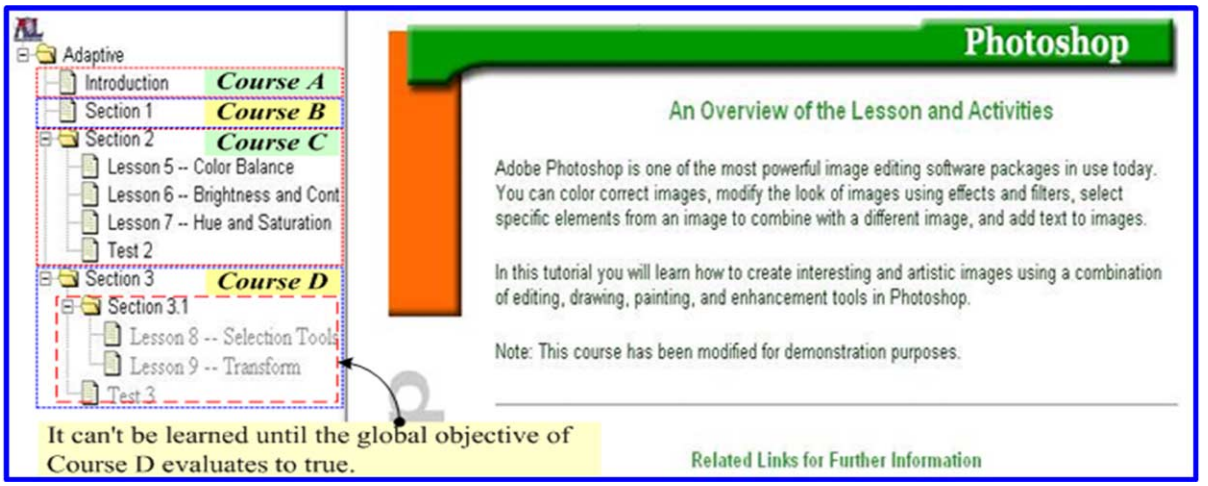

Fig. 15. The screenshot of course "PhotoShop" executed on SCORM RTE 1.3.

example course of "Photoshop" described in Section 4.3 was created by this OOCM authoring tool and executed on the SCORM RTE 1.3 as shown in Fig. 15. As we see, the table of content in the left side of Fig. 15 is consistent with the sequencing definition of HLPN in Fig. 12a. For example, the Course D (Section 3) cannot be selected until the test result in Course C satisfies the objective measure in *Global Objective* P_G.

5.2. The evaluation of OOCM approach

For evaluating the efficiency of the OOCM approach compared with Reload Editor, an experiment has been done. The participants of experiment are eight Master students in educational college, which were divided into two groups: one (Experiment Group) used the OOCM authoring tool and the other (Comparison Group) used the Reload Editor. To begin with, everyone in two groups was given 30 min to be familiar with these tools and then given the same learning activity with desired sequencing behaviors to create the SCORM course by assigned tool for evaluating the *time cost*. Finally, two groups interchanged the assigned tool to create the same SCORM course for evaluating the *satisfaction degree* by questionnaire. The evaluation results are shown in Fig. 16. The average time of using OOCM authoring tool is 14 min while the average time of using Reload Editor is 32 min. Moreover, according to the questionnaire, 1) learning the tool easily, 2) constructing the course without setting the complicated sequencing rules and 3) imagining the final course structure easily are the main advantages of OOCM authoring tool compared with Reload Editor. This shows that the OOCM approach is workable and beneficial for teachers/instructional designers.

6. Conclusion

In this paper, we propose a systematic approach, called *Object Oriented Course Modeling* (OOCM),

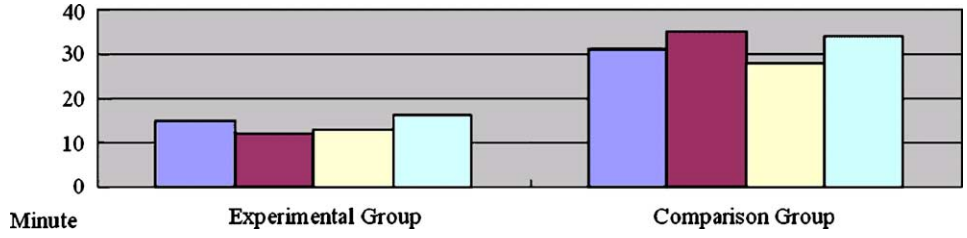

Fig. 16. The histogram of the time cost.

to construct the SCORM compliant course. High-Level Petri Nets (HLPN) is applied to model the basic sequencing components, called *Object-Oriented Activity Tree* (OOAT). Every OOAT as a middleware represents a specific sequencing behavior in learning activity and corresponding structure with associated sequencing rules of AT in SCORM. Thus, these OOATs can be easily used to model and construct the course with complex sequencing behaviors for different learning guidance. Moreover, two algorithms, called PN2AT and AT2CP, are also proposed to transform HLPN modeled by OOATs into a tree-like structure with associated sequencing rules in Activity Tree (AT) and package the AT with related physical learning resources into a SCORM compliant course file described by XML language, respectively. Finally, based upon the OOCM scheme, a prototypical authoring tool with graphical user interface (GUI) is developed. For evaluating the efficiency of the OOCM approach compared with existing authoring tools, an experiment has been done. The experimental results show that the OOCM approach is workable and beneficial for teachers/instructional designers. Therefore, in the near future, we will improve the OOAT models to model the remaining definitions in SN for enhancing its scalability and flexibility, e.g., the Limit Conditions, Selection Controls, etc. The OOCM prototypical authoring tool will be enhanced to import new OOAT models which are modified and created based upon new purposes or version of SCORM. In addition, for developing the personalized learning course, applying the Educational Theory to OOATs modeling will also be investigated.

Acknowledgement

This work was partially supported by National Science Council of the Republic of China under contracts NSC 93-2524-S-009-001 and NSC 93-2524-S-009-002.

References

[1] Aviation Industry CBT Committee (AICC), AICC-Aviation Industry CBT Committee, 2004, http://www.aicc.org.

[2] Alliance for Remote Instructional and Authoring and Distribution Networks for Europe (ARIADNE), ARIADNE: foundation for the European knowledge pool, 2004, http://www.ariadne-eu.org.

[3] Click2Learn, 'Click2learn unveils SCORM 1.2 resource kit', sum total, 2004, (Retrieved 4 September 2004 from). http://www.sumtotalsystems.com/archived/pressrelease_c2l.html?cid=40.

[4] Elicitus Content Publisher, (ECP), Elicitus content publisher—create more learning with less authoring, 2004, (Retrieved 4 September 2004 from) http://www.elicitus.com/home.htm.

[5] e-learning Centre, e-learning centre: course authoring tools, 2004, (Retrieved 4 September 2004 from) http://www.e-learningcentre.co.uk/eclipse/vendors/authoring.htm.

[6] Instructional Management System (IMS), IMS global learning consortium, 2004, http://www.imsproject.org/.

[7] JGraph, Open-source graph component available for Java, 2004, (Retrieved April 21, 2004 form) http://www.jgraph.com/.

[8] IEEE Learning Technology Standards Committee (LTSC), IEEE LTSC|WG12, 2004, http://ltsc.ieee.org/wg12/.

[9] E.R. Jones, Dr. Ed's SCORM course, 2004, http://www.scormcourse.jcasolutions.com/index.php.

[10] L. Kause, C. Fallon, 'Creating e-learning content in authorware 7 for SCORM1.2-compliant LMSs and LCMSs', macromedia-solutions:creating e-learning content in authorware 7 for SCORM1.2-compliant LMSs, 2004, (Retrieved 4 September 2004 from) http://www.macromedia.com/resources/elearning/article/lo_packager01/.

[11] LSAL, SCORM Best Practices Guide for Content Developers, Learning Systems Architecture Laboratory, Carnegie Mellon 2003, (LSAL http://www.lsal.cmu.edu/lsal/expertise/papers/index.html).

[12] Reload Editor (Reload), Reload project, 2004, http://www.reload.ac.uk.

[13] Sharable Content Object Reference Model (SCORM), Advanced distributed learning, 2004, http://www.adlnet.org/.

[14] Seminar Learning System, (SLS), Seminar learning system-information transfer, 2004, (Retrieved 4 September 2004 from). http://www.seminar.co.uk/author/author.cfm.

[15] Sequencing and Navigation (SN), 'Sharable Content Object Reference Model (SCORM) Sequencing and Navigation (SN) version 1.3', advanced distributed learning, 2004, http://www.adlnet.org/index.cfm?fuseaction=DownFile&libid=648&bc=false.

[16] W3C, World wide web consortium, (updated 9 Jun. 2004) http://www.w3.org.

[17] eXtensible Markup Language (XML), Extensible Markup Language (XML), (updated 26 Mar. 2004) http://www.w3c.org/xml/.

[18] P. Brusilovsky, J. Vassileva, Course sequencing techniques for large-scale web-based education, Journal of Engineering Education and Lifelong Learning 13 (2003) 75–94.

[19] J.E. Gilbert, C.Y. Han, Adapting instruction in search of A significant difference, Journal of Network and Computer Application 22 (1999) 149–160.

[20] K. Jensen, Coloured Petri Nets. Basic concepts, analysis methods and practical use, Monographs in Theoretical Computer Science, Springer-Verlag, 1997.

[21] K. Jensen, An introduction to the theoretical aspects of coloured Petri Nets, in: J.W. de Bakker, W.-P. de Roever, G. Rozenberg (Eds.), A Decade of Concurrency, Lecture Notes in Computer Science, vol. 803, Springer-Verlag, 1994, pp. 230–272.
[22] K. Jensen, G. Rozenberg, High-level Petri Nets, Theory and Application, Springer-Verlag Publishers, 1991.
[23] J. Lee, L.F. Lai, A high-level Petri Nets based approach to verifying task structures, IEEE Transactions on Knowledge and Data Engineering 14 (2) (2002) 316–335.
[24] X. Li, W. Yu, Object oriented fuzzy Petri Net for complex knowledge system modeling, Proceedings of the 2001 IEEE International Conference on Control Applications, 2001, pp. 476–481.
[25] F.H. Lin, Modeling online instruction knowledge for virtual training systems using Petri Nets, Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, vol. 1, Victoria, B.C., Canada, ISBN: 0-7803-7080-5, 2001, pp. 212–215.
[26] X.Q. Liu, et al., Knowledge aggregation and navigation high-level Petri Nets—based in E-learning, Proceedings of the First International Conference on Mache Learning and Cybernetics, Beijing, China, 2002, pp. 420–425.
[27] T. Murata, Petri Nets: properties, analysis and applications, Proceedings of the IEEE 77 (4) (1989) 541–580.
[28] L. Sheremetov, A.G. Arenas, EVA: an interactive web-based collaborative learning environment, Computers and Education 39 (2) (2002) 161–182.
[29] K. Timothy, J. Shih, C.S. Hung, W.C. Ko, W.C. Chang, N.H. Lin, Collaborative courseware authoring based on SCORM metadata, Proceedings of the IEEE International Conference on Multimedia and Expo 2003 (ICME 2003), Taipei, Taiwan, 2003 July, Retrieved 4 September 2004 from http://www.mine.tku.edu.tw/scorm/.
[30] E. Triantafllou, et al., The design and the formative evaluation of an adaptive educational system based on cognitive styles, Computers and Education 41 (1) (2003) 87–103.
[31] J. Vassileva, R. Deters, Dynamic courseware generation on the WWW, British Journal of Educational Technology 29 (1) (1998) 5–14.
[32] J.T.D. Yang, C.Y. Tsai, T.H. Wu, Visualized online simple sequencing authoring tool for SCORM-compliant content package, Proceedings of the 4th IEEE International Conference on Advanced Learning technologies (ICALT 2004), Finland, 2004 August.

Jun-Ming Su was born in Kaohsiung, Taiwan, on February 18, 1974, he graduated with a BS degree from the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan in 1997. He received his MS degree from the Institute of Computer Science, National Chung Hsing University, Taiwan in 1999. Currently, he is a PhD student at National Chiao Tung University, Taiwan. His current research interests include intelligent tutoring system, knowledge engineering, expert systems, and data mining, etc.

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

Chia-Yu Chen was born in Taichung, Taiwan, on March 20, 1980. She graduated with BS and MS degrees from the Department of Computer and Information Science, National Chiao Tung University, Taiwan in 2002 and 2004, respectively. Currently, she is a research assistant in ASUS company, Taiwan. Her research interests include e-learning, data mining, etc.

Jui-Feng Weng was born in Taichung, Taiwan, on June 25, 1978. He graduated with BS and MS degrees from the Department of Computer and Information Science, National Chiao Tung University, Taiwan in 2000 and 2002, respectively. Currently, he is a PhD student at National Chiao Tung University, Taiwan. His current research interests include e-learning, knowledge engineering, expert systems, and data mining, etc.

Wen-Nung Tsai received his BS degree from National Chiao Tung University in 1977 and his MS degree in computer science from National Chiao Tung University in 1979. He was in the PhD 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 com-puting, network security, operating system, and distance learning.