A SCORM-based e-Learning Process Control Model and Its Modeling System

Hyunah Kim, Eunjung Lee, Junchul Chun and Kwanghoon Pio Kim
Department of Computer Science, Kyonggi University
Suwonsi, Gyeonggido, 443-760, South Korea
[e-mail: {hakim, ejlee, jcchun, kwang}@kgu.ac.kr]
*Corresponding author: Kwanghoon Pio Kim

Received December 16, 2010; revised April 18, 2011; revised June 2, 2011; revised September 22, 2011; accepted November 10, 2011; published November 29, 2011

Abstract

In this paper, we propose an e-Learning process control model that aims to graphically describe and automatically generate the manifest of sequencing prerequisites in packaging SCORM's content aggregation models. In specifying the e-Learning activity sequencing, SCORM[1] provides the concept of sequencing prerequisites to be manifested on each e-Learning activity of the corresponding tree-structured content organization model. However, the course developer is required to completely understand the SCORM's complicated sequencing prerequisites and other extensions. So, it is necessary to achieve an efficient way of packaging for the e-Learning content organization models. The e-Learning process control model proposed in this paper ought to be an impeccable solution for this problem. Consequently, this paper aims to realize a new concept of process-driven e-Learning content aggregating approach supporting the e-Learning process control model and to implement its e-Learning process modeling system graphically describing and automatically generating the SCORM's sequencing prerequisites. Eventually, the proposed model becomes a theoretical basis for implementing a SCORM-based e-Learning process management system satisfying the SCORM's sequencing prerequisite specifications. We strongly believe that the e-Learning process control model and its modeling system achieve convenient packaging in SCORM's content organization models and in implementing an e-Learning management system as well.

Keywords: e-Learning process, e-Learning activity sequencing prerequisites, SCORM content package's menifest

[1] Sharable Content Object Reference Model by the Advance Distributed Learning (ADL) Initiative that aims to foster the e-Learning specifications and standards. The scope of the paper is based on SCORM Version 1.2.

This work was supported by the GRRC program of Gyeonggi Province. We would like to extend our appreciation to our fellow researchers in the MIC Lab, Graphics Processing Lab and the Collaboration Technology Research Lab (CTRL) at Kyonggi University, and transfer also thanks to the GRRC colleagues and the officers of Gyeonggi Province. Additionally, the implementation of the e-Learning process control modeling system has been performed on the BEIT's Grid Computing Facility.

DOI: 10.3837/tiis.2011.11.014

1. Introduction

The e-Learning technology has been developed for many years to improve efficiency and increase cost-effectiveness in learning environments. Furthermore, based upon the outcomes of a lot of empirical studies, it is generally acceptable that individually tailored instruction using information technology sometimes offers ideal learning outcomes. This is because, in contrast to classroom learning, information technologies can adjust the pace, sequence, content and method of instruction so as to better fit each student's learning style, interests and goals [1]. However, even though the most current and cutting-edge instructional technologies, such as web-based instruction, interactive multimedia instruction, intelligent tutoring systems and SCORM-based e-Learning management systems, are actively adopted into the educational environments, realizing the promise of improved learning efficiency still depends on the ability of those technologies, and simultaneously in order to sufficiently and easily tailor the quality of learning, it is necessary to provide the appropriate learning process and experience according to the needs of individuals. Also, the one-on-one individualization capabilities (or tailoring capabilities of the learning process) of technology-based instruction, in contrast to one-on-many classroom-based instruction, may approximate and perhaps exceed the effectiveness of one-on-one tutoring.

These technology-based instructional capabilities of individual learners and their needs can be seen in several categories of e-Learning products. The typical standardized e-Learning product is the ADL[2]'s SCORM that aims to foster creations of reusable learning contents for computer and web-based instructions. Particularly, the SCORM 2004 Third Edition [1][3][4][5], which has been recently released, newly announced the sequencing and navigation (SN) standards [5] so as to define a set of the rules and behaviors for the sequencing and ordering of learning activities, which is the main concern of this paper. According to the sequencing and navigation standards, the learning sequencing rules and presentation strategies for e-Learning activities are encoded in XML to be embedded into the items of the tree-structured content organizations. Nevertheless, the details of the XML-based SN specifications are extremely complicated, and it is very hard for contents developers to grasp and compose their e-Learning content packages. The previous version of SCORM 1.2 also does not define any sequencing information that the learner is able to choose which learning object to use and in which order. Instead, SCORM 1.2 provides an optional means to describe conditional sequencing behavior in the organization section of the manifest using the concept of prerequisites, which might not be a convenient and easy way from the users' point of view, either. Therefore, it certainly needs to be supported by a certain way of convenient means in defining e-Learning activity sequencing and navigation specifications at this moment.

In this paper, we propose a process-driven e-Learning content organizing approach, which is called the e-Learning process control model. In contrast to the SCORM's approach, our approach is able to graphically define the e-Learning activity sequencing model and to automatically generate a set of sequencing prerequisites associated with the items' elements constituting the corresponding content organization. In other words, we try to conceive a graphical and formal modeling approach to conveniently compose all of the XML-based manifests (all items' specifications) of the e-Learning content aggregation model. Particularly,

[2] The Advanced Distributed Learning Initiative launched by the Department of Defense and the White House Office of Science and Technology Policy.

we focus on the sequencing prerequisites to be embedded into the item-elements of the SCORM's content organization model. Note that the prerequisites are the script language specifying the sequencing and ordering of e-Learning activities by a set of specific operators. Therefore, we do not need to keep the hierarchical structure (or the activity tree) anymore in aggregating contents. At the same time, we are able to easily represent an e-Learning process by jointly using the sequential-style of learning activity flow control, the conjunctive-style (parallel-style) and disjunctive-style (alternative-style) of learning activity flow control [8]. Additionally, we are able to define diverse and multiple e-Learning processes on a single content organization model, and these multiple e-Learning processes can be effectively and efficiently applied to educating diverse levels of students and trainees for the same e-Learning content package.

The remainder of this paper is organized as follows. The next section briefly describes about the motivation, related work and scope of this paper. Section 3 explains the concept of the e-Learning process control model by defining its graphical and formal notations. Further, we describe how to apply the model to compose a SCORM content aggregation model, and we show how the model is automatically packaged into the manifests. Finally, the paper implements the e-Learning process control modeling system with an operational example, and summarizes the implications of the proposed model with our future works.

2. Related Work and Scope

Studies on the e-Learning process control model originated in [12] that was firstly presented in the conference[3], where we just introduced the concept of e-Learning activity sequencing approach as a substitute means of the conventional SCORM's sequencing and navigation model. Based upon [12], we are able to refurbish the concept and rename it the e-Learning process control model. That is, the primary motivation of this paper is on the structure of the content organization package of the SCORM content aggregation model.

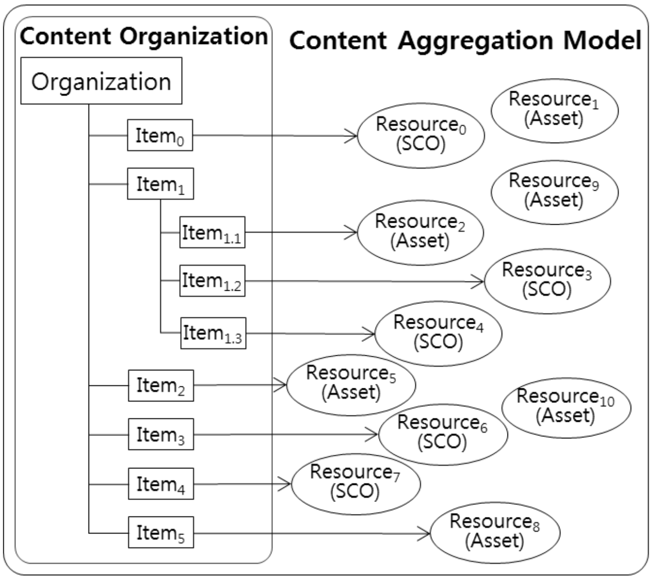


Fig. 1. A SCORM Content Aggregation Model

[3] The IEEE Fourth International Conference on Semantics, Knowledge and Grid, pp. 322 - 329, 2008

As shown in **Fig. 1**, the basic structure of a content aggregation model [2][3] consists of content organizations and learning objects. The content organization is built by tree-structured items, and the learning objects are represented by resources, each of which is linked to a SCO (Sharable Content Object) or an asset chosen from a silo of learning objects. Moreover, these elements of the content aggregation model are precisely specified in a SCORM's package. That is, LMS's (e-Learning Management System) packaging specification uses SCORM as a foundation for the packaging and organization of learning objects. A package may contain more than one content organization for the same learning objects, which implies that several e-Learning processes covering the same subject at different levels of depth or different audiences can be defined in a single SCORM package. Therefore, LMS can take advantage of the learning flexibility of this feature, which enables learners to choose more appropriate e-Learning content organizations. As an example, we are able to represent the package specification of the content aggregation model of **Fig. 1** as follows.

```
<organizations default="TOC">
    <organization identifier="TOC1" structure="hierarchical">
    <title>default</title>
          <item identifier="ITEM0" idref="RESOURCE0" isvisible="true">
            <title>Lesson 0</title>
          </item>
          <item identifier="ITEM1" isvisible="false">
            <title>Lesson 1</title>
            <item identifier="ITEM1.1" idref="RESOURCE2" isvisible="true">
                 <title>Introduction 1</title>
            </item>
            <item identifier="ITEM1.2" idref="RESOURCE3" isvisible="true">
                 <title>Content 1</title>
            </item>
            <item identifier="ITEM1.3" idref="RESOURCE4" isvisible="true">
                 <title>Summary 1</title>
            </item>
          </item>
          <item identifier="ITEM2" idref="RESOURCE5" isvisible="true">
                 ....

    </organization>
</organizations>
          ....
<resource identifier="RESOURCE0" type="webcontent"
                       adlcp:scormtype='sco' href="sco1.html">
    <file href="sco1.html"/>
    <file href="sco2.html"/>
    <file href="sco3.html"/>
</resource>
<resource identifier="RESOUCE2" type="webcontent"
                       adlcp:scormtype='asset' href="pics-sigs-add.jpg">
    <file href="pics-sigs-add.jpg"/>
</resource>
          ....
```

After organizing the content aggregation model with tree-structured items as shown in **Fig 1**, we need to supplement some extensions for controlling learning activities (SCOs) that a run time environment or LMS will operate with the learner. SCORM does specify how a LMS uses some optional features of the package, such as multiple content organizations, multiple learning sequences, and various navigation controls, through some special rules. However, these special rules are too complicated to effectively compose the learning content package. Particularly, in terms of defining the learning sequences, SCORM version 1.2 [2] provides a special rule that is a very basic form of prerequisites, because in principle, it does not define how to sequence SCOs, but it allows learner-choice sequencing of SCOs. In other words, a prerequisite refers another element in a content organization tree that must be completed or mastered. However, because the SCORM specification does not clearly define the behavior associated with prerequisites, as the only way to specify them, the principle is to let them use prerequisites at their own risk. Therefore, we try in this paper to conceive a formal and graphical approach of defining the learning sequences with prerequisites, which we term "e-Learning processes[8]", based on the SCORM specifications.

3. e-Learning Process Control Model

The e-Learning process control model is based on a mathematical formalism to graphically design a prerequisite-driven learning process for the SCORM sequencing specifications, and it aims at eventually being a theoretical basis for developing a prerequisite-driven SCORM sequencing modeling system or a graphical e-Learning content aggregation modeling system, which is able to automatically generate the SCORM-based e-Learning content specifications for a SCORM-based e-Learning management system. In this section, we describe a fundamental motivation of the model and introduce the metamodel and the descriptions of its components. Finally, we give the graphical notations and their formal definitions of the model, and we explain how to generate the SCORM's sequencing prerequisites from the model.

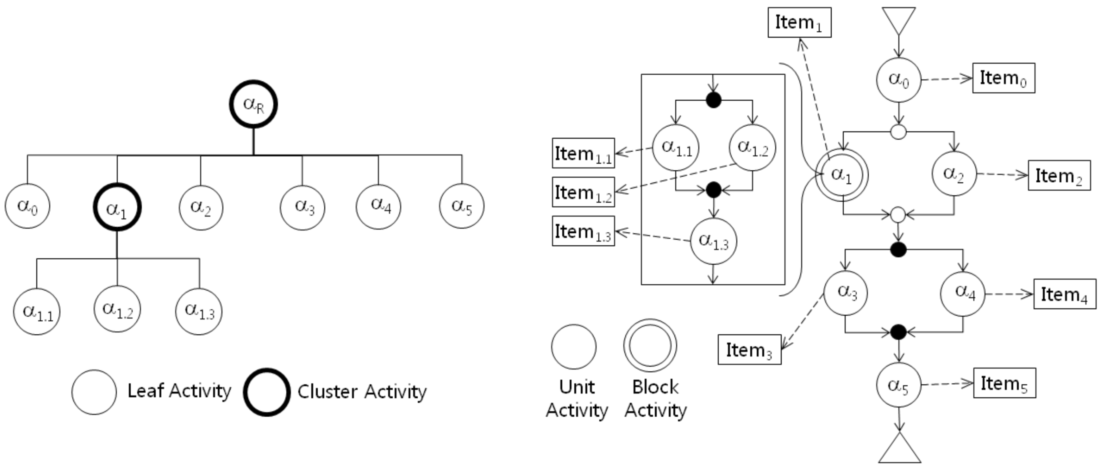


Fig. 2. The e-Learning Activity Tree ***vs***. The e-Learning Process Control Model

3.1 Motivation: Advanced e-Learning Activities Sequencing

In the current SCORM, the content organization is a map that represents the intended use of the content through tree-structured units of instructions (Activities). The map shows how activities relate to one another as shown in the left-hand side of **Fig. 2**. Each of the activities corresponds to an item in the content organization of the content aggregation model shown in **Fig. 1**, and it is conceptually possible for a certain type of activity to contain other activities, which is called a cluster activity in the SCORM's activity tree. There is also no limit on the number of levels of nesting for activities. Only the leaf activity has an associated learning resource (SCO or Asset) that is used to perform the corresponding item. In terms of the representation of e-Learning process, SCORM specifies the extended element, Prerequisite, of item's manifest and the sequencing and navigation (SN) model in Release 2002 (SCORM 1.2) [2] and Release 2004 (SCORM 1.3) [5], respectively. The sequencing information such as Prerequisites or sequencing elements of the SN model, which we call an e-Learning process, is embedded in each item's manifest as the internal specifications of each activity.

Summarily speaking, the SCORM content aggregation model provides a way of representation for the e-Learning process through the activity tree or the Prerequisite element, while on the other it might be provided a certain type of automated means to define and visualize a set of e-Learning processes running over its e-Learning content organization. Without such an automated means, it is extremely hard and inconvenient to define and visualize its e-Learning processes as well as to adopt some advanced sequencing-behavior features like dynamic changes, alternative and parallel sequencing, looping sequencing, e-Learning process status reporting and auditing, and so forth. Therefore, in this paper, we propose a formal and graphical representation approach to resolve the problem, which is called an e-Learning process control model. The right-hand side of **Fig. 2** shows a possible e-Learning process that can be defined by our approach; The unit activity and block activity in the e-Learning process control model define the leaf activity and the cluster activity in the SCORM's e-Learning activity tree. Also, it shows that a unit activity is associated with an item of the content organization. The model proposed in this paper is surely applicable to both SCORM 1.2's Prerequisites approach and SCORM 1.3's sequencing and navigation model approach. Particularly, in this paper we focus on the sequencing Prerequisites of SCORM 1.2, which can be handled by the e-Learning activity control model.

3.2 Prerequisites: SCORM 1.2 Sequencing Behaviors

In order to describe the e-Learning sequencing behaviors, SCORM 1.2 provides an optional means in the organization section of the manifest by using the concept of prerequisites. That is, in the manifest file of the content aggregation model, `<item>` elements can include an optional sub-element called `<adlcp:prerequisites>`, which is reserved for a field to be used to represent a conditional logic for the learner's navigation over a course structure of the content aggregation model. This element can also be used for the lesson-status data of each `<item>`, which is tracked at runtime by an LMS.

The run-time environment provides a means for learning resources (objects) to report to an LMS when a particular part of the learning resource is "completed" or "Incomplete". Using the tracked information available to it, an LMS can evaluate the logical statements inserted into the `<adlcp:prerequisites>` element of an `<item>` to determine what learning resource should be delivered to the learner next. The `<adlcp:prerequisites>` element defines, in conditional logic, what other parts of the content must have been completed before the learner may start the `<item>` within which it is contained. The lesson-status value is assigned based on SCO basis and has a preserved term that is one out of the following values: Passed, Completed, Browsed, Failed, Not attempted and Incomplete.

The following describes some of the operators[4] of a prerequisite scripting language called "aicc-script" [5] [2], which is used to encode the conditional logic statements in an "`<adlcp:prerequisites>`" tag element:

---

[4] This paper uses the mathematical notations for the sake of clear understanding. In fact, SCORM uses the script's typesetting notations.
[5] Used in the older content organization model through which the Aviation Industry CBT Committee (AICC) made its courses' structure, AICC CMI001 Guidelines for Interoperability.

- **AND** All elements separated by a "∧" must be compete for the expression to be evaluated as completed. For example, (S1 ∧ S2 ∧ S3) implies that SCOs numbers S1, S2 and S3 must all be complete ("passed" or "completed") for the group to be considered as completed.
- **OR** If any of the elements separated by a "∨" are complete ("passed" or "completed"), then the expression is considered true. For example, (S4="passed" ∨ S5="passed" ∨ S6="passed") implies that if any one of the SCOs, S4, S5, or S6, is passed, then the group is considered as completed.
- **NOT** An operator (~) that returns incomplete (false) if the following element or expression is complete, and returns complete (true) if the following element or expression is incomplete (false). (Element Identifier: S7, Requirement: ~S8) implies that the learner may enter SCO S7 as long as SCO S8 has not been completed (that is, the status of S8 must be "incomplete", "failed", or "not attempted"). If SCO S8 is complete, the learner may not enter SCO S7.
- **EQUAL** An operator (=) that returns true when representations on both sides of the symbol have the same values. (Element Identifier: S10, Requirement: S9="passed") implies that the learner may enter SCO S10 if he has passed SCO S9.
- **NOT EQUAL** An operator (≠) that returns true when elements on both sides of the symbol have different values. (Element Identifier: S12, Requirement: S11≠"passed") implies that the learner may enter SCO S12 as long as he or she has not passed SCO S11. Note that this expression is different from the example for the **NOT** operator. The equivalent of S11 is (S11≠"passed" ∧ S11≠"completed").
- **X*** **X** is an integer number. An operator that **X** or more numbers of the set that follows must be complete for the expression to be complete (true). For example, (Element Identifier: S25, Requirement: 3*{S21, S22, S23, S24}) implies that any three or more of the SCOs S21, S22, S23 and S24, must all be complete ("passed" or "completed") before the student can enter SCO S25.
- **Precedence** The expression inside the parenthesis ( ) must be evaluated before combining its results with other parts of the logical statement. Parentheses may be nested. The operator precedence is the same as that in the C programming language including the use of parenthesis. (Element Identifier: S13, Requirement: S4 ∧ S5 ∨ S6) implies that, in this statement, the last unit S36 by itself is enough to be evaluated in order to enable the learner to enter S13. On the other hand, (Element Identifier: S13, Requirement: S4 ∧ (S5 ∨ S6) implies that, by adding the parenthesis, it becomes necessary to complete ("passed" or "completed") at least two units (S6 all by itself is no longer enough) to enter unit S13.

Based upon the prerequisite scripting language, the e-Learning content developer is able to define a learning sequencing behavior, which we call an e-Learning process, that describes what other SCOs of the learning content must have been completed before starting the item. This allows an LMS to compute multiple paths through the learning content. As an example, the following is to describe the detailed specification of the item, 'Item1', in **Fig. 1**. We can interpret this as "In order to access the Item1.3, the learner must pass (or complete) both of the items, Item1.1 and Item1.2":

```
<item identifier='Item1'>
    <item identifier='Item1.1' identifierref='R_I1.1'/>
```

```
      <item identifier='Item1.2' identifierref='R_I1.2'/>
      <item identifier='Item1.3' identifierref='R_I1.3'>
            <adlcp:prerequisites type='aicc_script'>
                  Item1.1 ∧ Item1.2
            </adlcp:prerequisites>
      </item>
   </item>
```

As presented in the previous example, SCORM 1.2 provides a systematic means for specifying an e-Learning process, which is the prerequisite scripting language. It might be a reasonable approach from the LMS developer's point of view. However, it ought to be an inappropriate approach from the e-Learning content developer's standpoint. Therefore, we propose a new way for content developers to graphically define an e-Learning process and automatically generate its sequencing prerequisites to be embedded into the specifications of the corresponding items of the SCORM content aggregation model. We call it the process-driven content organizing approach with respect to the explicit sequencing definition of the e-Learning activities, and it is realized as an e-Learning Process Control Model. This approach is based on the strong belief that the e-Learning process is the most effective factor in the e-Learning instruction, and through this approach, we are also able to realize the interactive e-Learning activity flow control and management on LMS's runtime environments. The details of this approach are described in the following contiguous sections.

3.3 Meta-Model of the e-Learning Process Control Model

The SCORM-based e-Learning arena has a vast published literature, and sometimes a confusing array of terms and meanings. Next we will define the basic terminology such as e-Learning process, e-Learning activity, e-Learning item, e-Learning resources consisting of SCO (Sharable Content Object) and Asset, e-Learning-case, educator including group, role and so on. Based upon the terminology we build the meta-model of the e-Learning process control model as shown in **Fig. 3**.

An **e-Learning process** is a predefined or intended set of e-Learning steps, called activities, and a partial ordering of these activities. An e-Learning management system helps to organize, control, and execute the e-Learning instances, which are called **e-Learning-cases**, instantiated from the e-Learning process. The control aspect of the e-Learning process is specified by the ordered combinations of sequential (before entering on e-Learning activity A, complete activity B), conjunctive (before entering on e-Learning activity A, complete activities B and C), disjunctive (before entering on e-Learning activity A, complete activities B or C) prerequisite logics. Note that each member of the prerequisite is specified by the prerequisite conditions, such as Passed, Completed, Browsed, Failed, Not attempted and Incomplete. An **e-Learning activity** is a basic unit of learning behavior, and consists of two types, elementary activity and block activity, containing another e-Learning process. That is, the elementary activity and the block activity are realized by a unit item and a compound item, respectively. Only the unit item (elementary activity) is materialized through an e-Learning resource (SCO or Asset) that is used to perform its activity in one of three modes, manual, automatic, or hybrid. Except for the prerequisite, each activity may additionally define the AICC/CMI inspired extensions for the sequencing-related behaviors, such as time-related element (max time allowed, time limit action), data from eLMS element, and mastery score element. The time-related element allows a maximum time limit for the activity and specifies an action determining what happens if the student does not respect within the time limitation; The data from the eLMS element is able to specify how the eLMS terminates the sequence without message. The mastery score element is used to establishing the passing score, which should be a value between 0 and 100, for the activity.

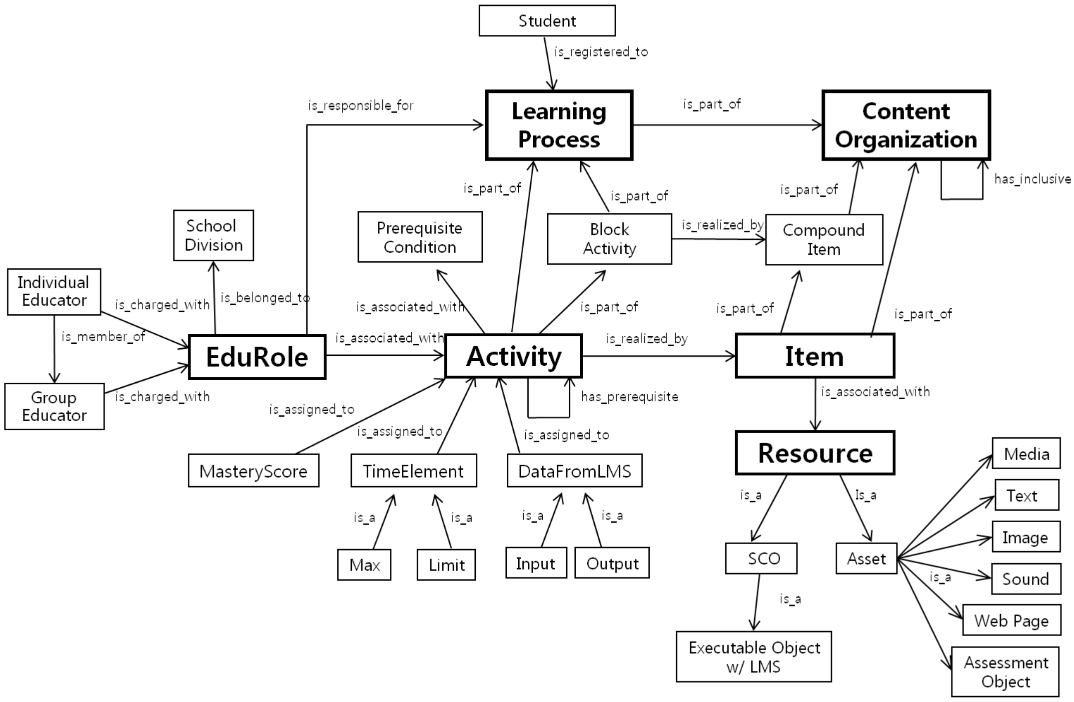


Fig. 3. The Meta-Model of the e-Learning Process Control Model

Typically, one or more participants (educators) are associated with each activity via roles. An **EduRole** is a named designator for one or more educators, which conveniently acts as the basic responsibility of the corresponding activity. An **educator** is an individual or a group that can fulfill the associated role to execute, to be responsible for, or associated in some way with activities and e-Learning processes. Multiple copies, so-called e-Learning-cases, of the same e-Learning process, may be in various stages of execution. Thus, the e-Learning process can be considered as a class, and each e-Learning-case can be considered an instance of the class, each of which is taken by a student. An e-Learning-case is thus defined as the locus of control for a particular execution of an e-Learning process. If an e-Learning activity is executed in automatic or hybrid mode, then this means that the whole/part of the SCO (or Asset) resource associated with the activity is automatically launched by an e-Learning enactment service.

3.4 Representation of the e-Learning Process Control Model

Based on the meta-model described in the previous section, we are able to define its formal and graphical representations of the e-Learning process control model[6]. In the following subsections, we introduce them and show how the model may be applied to automatically generate a SCORM's content aggregation model embedding the concept of e-Learning process with the sequencing prerequisites through building a typical e-Learning course as an example.

[6] The origin of which is the information control net [6][7][9] that has been widely used for modeling workflows and business processes in an organization.

3.4.1 Formal Representation

A basic e-Learning process control model is formally defined by 9-tuple, $\mathbf{\Gamma} = (\delta, \varphi, \xi, \gamma, \varepsilon, \pi, \kappa, \mathbf{I}, \mathbf{O})$ formation over an e-Learning activity set, **A** (including unit/block activities), a set **S** of items, a set **R** of resources implemented by SCOs or Assets, a set **C** of prerequisite conditions, a set **D** of relevant-data from/to eLMS, a set **P** of educational roles, and a set **T** of educators (including individual and group educators), where

- **I** is a finite set of initial input relevant-data, assumed to be loaded with information by some external e-Learning process before executing the model;
- **O** is a finite set of final output relevant-data containing information to be used by some external e-Learning process after executing the model;
- $\delta = \delta_i \cup \delta_o$

  where, $\delta_o : \mathbf{A} \rightarrow \wp(\mathbf{A})$ is a multi-valued mapping function of an e-Learning activity ($\alpha \in \mathbf{A}$) to its power-set of immediate successors with post-requisites in the e-Learning sequencing behavior, and $\delta_i : \mathbf{A} \rightarrow \wp(\mathbf{A})$ is a multi-valued mapping function of an activity to its power-set of immediate predecessors with pre-requisites in the e-Learning sequencing behavior;
- $\varphi = \varphi_{item} \cup \varphi_a$

  where, $\varphi_{item} : \mathbf{A} \rightarrow \wp(\mathbf{S})$ is a single-valued mapping function of an e-Learning activity ($\alpha \in \mathbf{A}$) to a power-set of e-Learning items defined in the content organization model, and $\varphi_a : \mathbf{S} \rightarrow \wp(\mathbf{A})$ is a multi-valued mapping function of an e-Learning item to a power-set of e-Learning activities;
- $\xi = \xi_{resource} \cup \xi_{item}$

  where, $\xi_{item} : \mathbf{R} \rightarrow \wp(\mathbf{S})$ is a single-valued mapping function of a resource ($\sigma \in \mathbf{R}$) to a power-set of e-Learning items defined in the content organization model, and $\xi_{resource} : \mathbf{S} \rightarrow \wp(\mathbf{R})$ is a single-valued mapping function of an e-Learning item ($\beta \in \mathbf{S}$) to a power-set of resources;
- $\gamma = \gamma_i \cup \gamma_o$

  where, $\gamma_o : \mathbf{A} \rightarrow \wp(\mathbf{D})$ is a multi-valued mapping function of an e-Learning activity to its power-set of output relevant-data, and $\gamma_i : \mathbf{A} \rightarrow \wp(\mathbf{D})$ is a multi-valued mapping function of an e-Learning activity to its power-set of input relevant-data; Where, the type of relevant-data is classified into mastery-score, maximum time allowed, time limit action, and data from/to eLMS;
- $\varepsilon = \varepsilon_a \cup \varepsilon_p$

  where, $\varepsilon_p : \mathbf{A} \rightarrow \mathbf{P}$ is a single-valued mapping function of an e-Learning activity to one of the educational roles, and $\varepsilon_a : \mathbf{P} \rightarrow \wp(\mathbf{A})$ is a multi-valued mapping function of an educational role to a power-set of e-Learning activities;
- $\pi = \pi_p \cup \pi_t$

  where, $\pi_t : \mathbf{P} \rightarrow \wp(\mathbf{T})$ is a multi-valued mapping function of an educational role to a power-set of educators, and $\pi_t : \mathbf{T} \rightarrow \wp(\mathbf{R})$ is a multi-valued mapping of an educator to a power-set of educational roles;
- $\kappa = \kappa_i \cup \kappa_o$

  where, $\kappa_i : \mathbf{A} \rightarrow \wp(\mathbf{C})$ is a power-set of prerequisite-conditions, **C**, on each arc, $(\delta_i(\alpha), \alpha)$, $\alpha \in \mathbf{A}$; and $\kappa_o : \mathbf{A} \rightarrow \wp(\mathbf{C})$ is a power-set of postrequisite-conditions, **C**, on each arc, (α, δ_o(α)), α ∈ **A**; Where, each element's value of the set of pre/post-requisite conditions is one of the status values: {Passed, Completed, Browsed, Failed, Not attempted and Incomplete}.

Starting and Terminating e-Learning Activities. Additionally, the execution of an e-Learning process control model commences by a single λ transition-condition. So, we always assume without loss of generality that there is a single starting e-Learning activity (α_I). At the commencement, it is assumed that all input relevant-data in the set, **I**, have been initialized with data by the external system:

$$\{ \exists \alpha_I \in \mathbf{A} \mid \delta_i(\alpha_I) = \varnothing \wedge \kappa_o(\alpha_I) = \{\lambda\} \}.$$

The execution is terminated with any one λ output transition-condition. Futher, we assume without loss of generality that there is a single terminating e-Learning activity (α_F). The set of output repositories, **O**, is data holders that may be used after termination by the external system:

$$\{ \exists \alpha_F \in \mathbf{A} \mid \delta_o(\alpha_F) = \varnothing \wedge \kappa_i(\alpha_F) = \{\lambda\} \}.$$

Sequencing Behaviors with Structured Properties. Given the formal definition, the sequencing behavior of an e-Learning process control model can be interpreted as follows.

For any e-Learning activity $\alpha \in \mathbf{A}$, in general,

$$\delta_i(\alpha) = \{ \{ ß_{11}, ß_{12}, \ldots\ldots, ß_{1m(1)}\}^s, \{ ß_{21}, ß_{22}, \ldots\ldots, ß_{2m(2)}\}^s, \ldots\ldots, \{ ß_{n1}, ß_{n2}, \ldots\ldots, ß_{nm(n)}\}^s \}$$

means that before entering to the e-Learning activity (α), it ought to be pre-conditionally satisfied with one of the following situations. Where, ***s*** is the selective-number, the value of which can be 0, 1, or x ($1 < x < n$).

- $n = 1 \wedge m(n) = 1 \wedge s = 0$: **Sequential prerequisite**, which is related with a sequential behavior. As an example, $\delta_i(\alpha) = \{\{ß\}^0\}$ represents that *ß* is assciated with a prerequsite of *α*. That is, upon the completion of *ß*, the activity, *α*, is able to lunch its own e-Learning items.
- $n > 1 \wedge \forall m(i) = 1\ (1 \le i \le n) \wedge s = 1$: **Conjunctive(AND) prerequisite**, which is related with a parallel behavior (parallel-join). As an example, $\delta_i(\alpha) = \{\{ß_{1j}\}^1, \{ß_{2j}\}^1, \ldots\ldots, \{ß_{nj}\}^1, 1 \le j \le m(j)\}$ represents that all of the e-Learning activities, $\{ß_{1j}\}^1, \{ß_{2j}\}^1, \ldots\ldots, \{ß_{nj}\}^1, 1 \le j \le m(j)$, must be simultaneously completed before launching the e-Learning activity, *α*.
- $n = 1 \wedge m(n) > 1 \wedge s = 1$: **Disjunctive(Exclusive-OR) prerequisite**, which is related with an alternative behavior (alternative-join). As an example, $\delta_i(\alpha) = \{\{ß_{i1}, ß_{i2}, \ldots\ldots, ß_{im(i)}\}^1, 1 \le i \le n\}$ represents that only one out of the e-Learning activities, $ß_{i1}, ß_{i2}, \ldots\ldots, ß_{im(i)}, 1 \le i \le n$, must be execlusively selected and completed before launching the e-Learning activity, *α*.
- $n = 1 \wedge m(n) > 1 \wedge s = x$: **Selective-Conjunction (X*: Selective-AND) prerequisite**, which implies a selective parallel bebavior (selective-join). As an example, $\delta_i(\alpha) = \{\{ß_{11}, ß_{12}, \ldots\ldots, ß_{1m(1)}\}^x\}$ with setting the selective-number represents that multiple

selective-choices, $\boldsymbol{x}$ $(1 < x < n)$, out of the e-Learning activities, $ß_{i1}, ß_{i2}, \ldots\ldots, ß_{im(i)}, 1 \leq i \leq n$, must be simultaneously completed before launching the e-Learning activity, α.

Also, for any e-Learning activity $\alpha \in \mathbf{A}$,

$$\delta_o(\alpha) = \{ \; \{ ß_{11}, ß_{12}, \ldots\ldots, ß_{1m(1)}\}^s, \; \{ ß_{21}, ß_{22}, \ldots\ldots, ß_{2m(2)}\}^s, \; \ldots\ldots \; \{ ß_{n1}, ß_{n2}, \ldots\ldots, ß_{nm(n)}\}^s \; \}$$

means that upon completing the e-Learning activity (α), it ought to be post-conditionally satisfied with one of the following situations. Where, $\boldsymbol{s}$ is the selective-number, the value of which can be 0, 1, or x $(1 < x < n)$.

- $n = 1 \wedge m(n) = 1 \wedge s = 0$: **Sequential post-requisite**, which is related with a sequential behavior. As an example, $\delta_o(\alpha) = \{\{ß\}^0\}$ represents that $ß$ is related with a post-requsite of α. That is, after the completion of α, the e-Learning activity, $ß$, is able to lunch its own e-Learning items.
- $n > 1 \wedge \forall m(i) = 1$ $(1 \leq i \leq n) \wedge s = 1$: **Conjunctive(AND) post-requisite**, which is related with a parallel behavior (parallel-split). As an example, $\delta_o(\alpha) = \{\{ß_{1j}\}^1, \{ß_{2j}\}^1, \ldots\ldots, \{ß_{nj}\}^1, 1 \leq j \leq m(j)\}$ represents that all of the e-Learning activities, $\{ß_{1j}\}^1, \{ß_{2j}\}^1, \ldots\ldots, \{ß_{nj}\}^1, 1 \leq j \leq m(j)$, must be simultaneously begun after completing the e-Learning activity, α.
- $n = 1 \wedge m(n) > 1 \wedge s = 1$: **Disjunctive(Exclusive-OR) post-requisite**, which is related with an alternative behavior (alternative-split). As an example, $\delta_o\alpha) = \{\{ß_{i1}, ß_{i2}, \ldots\ldots, ß_{im(i)}\}^1, 1 \leq i \leq n\}$ represents that only one out of the e-Learning activities, $ß_{i1}, ß_{i2}, \ldots\ldots, ß_{im(i)}, 1 \leq i \leq n$, must be execlusively selected and launched after completing the e-Learning activity, α.
- $n = 1 \wedge m(n) > 1 \wedge s = x$: **Selective-Conjunction (X*: Selective-AND) post-requisite**, which implies a selective parallel bebavior (selective-split). As an example, $\delta_o(\alpha) = \{\{ß_{i1}, ß_{i2}, \ldots\ldots, ß_{im(1)}\}^x, 1 \leq i \leq n\}$ with setting the selective-number represents that multiple selective-choices, $\boldsymbol{x}$ $(1 < x < n)$, out of the e-Learning activities, $ß_{i1}, ß_{i2}, \ldots\ldots, ß_{im(i)}, 1 \leq i \leq n$, must be simultaneously started right after completing the e-Learning activity, α.

Particularly, in forming the structured types (a pair of split and join) of seqencing behaviors, such as AND, Exclusive-OR, and Selective-AND, in an e-Learning process control model, they must be properly structurized by preserving the structured properties, the proper-nesting property and the matched-pairing property. Note that we use the prerequiste conditions in modeling an e-Learning process control model.

3.4.2 Graphical Representation

An e-Learning management system supporting e-Learning process control models provides a graphical modeling tool to design, analyze, and evolve e-Learning processes' specifications. An e-Learning process control model is graphically represented by interconnecting its associated e-Learning activities (Large circles) to each other through the four basic types of primitive sequencing behaviors. Eventually, the graphically defined model can be translated into a XML-based specification embedding prerequisites' transitions to form a content aggregation model with the SCORM 1.2's sequencing information.

In **Fig. 4**, we present the graphical notations of the four types of primitive sequencing behaviors formally defined in the previous subsection, and a possible precedence case of structural sequencing behavior. The followings are about each of the graphical notations:

- Sequential sequencing behavior: represented by a directed arc between nodes
- Exclusive-OR sequencing behavior: represented by a pair of small-open circles
- AND sequencing behavior: represented by a pair of small-filled circles
- Selective-AND, X*(3* if **X** = 3) sequencing behavior: represented by a pair of X*

If e-Learning activity α_A leads to e-Learning activity α_B, (i.e., (α_A, α_B) is an edge in the graph), then the e-Learning activity α_A must have a sequential priority over the e-Learning activity α_B, which means that α_A becomes a prerequisite-condition for entering α_B in the underlying e-Learning process model. Naturally, we need to define graphical notations of the remaining entities of the meta-model, such as item, resource, educational role, sco, asset, and so on. However, in this paper, we just focus on the graphical notations of the sequencing behavior to define a prerequisites (or post-requisites) menifest that is used to produce the sequencing information of e-Learning activities.

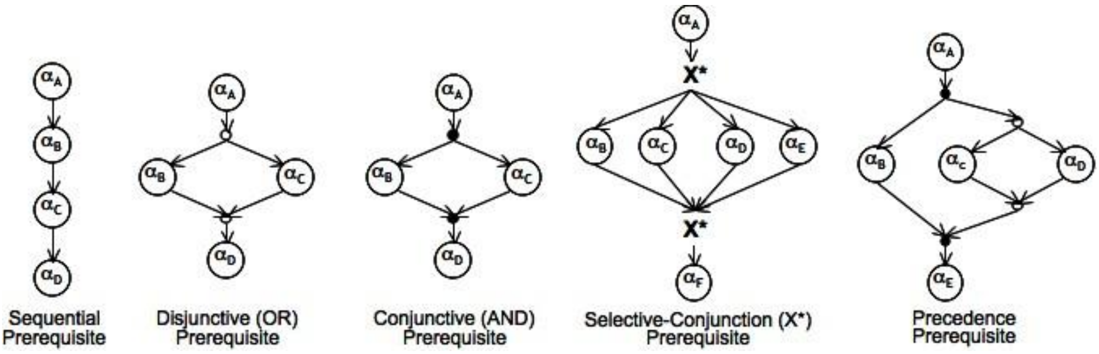


Fig. 4. Graphical Notations of the Primitive e-Learning Sequencing Behaviors

3.4.3 Translation Rules from Graphical Prerequisites to Script Prerequisites

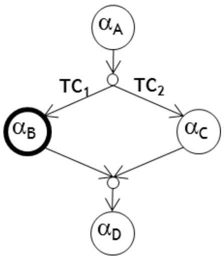
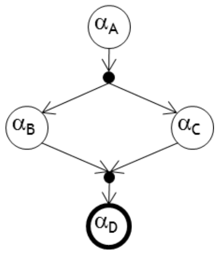
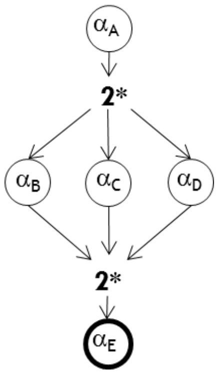
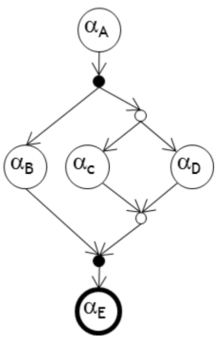
Table 1 shows the translation rules of the four types of primitive e-Learning sequencing behaviors. That is, the graphcial representations of the sequential, Exclusive-OR, AND, and Selective-AND sequencing behaviors can be translated into formal representations with the pre-conditional pre-requisites, respectively. And, each of these formal representations can be automatically translated into a form of script prerequisites by properly applying a proper-combination of the prerequiaite operators[7] defined in the prerequisite scripting language termed "aicc-script"[8] [2]. Ultimately, these script prerequisites are used to encode the conditional logic statements in a pair of "`<adlcp:prerequisites>`" and "`</adlcp:prerequisites>`" to form a prerequisite-manifest of the SCORM's content aggregation model.

---

[7] In the previous section, we introduced the operators, AND, OR, NOT, EQUAL, NOT EQUAL, X*, precedence ( ).

[8] Used in the older content organization model through which the Aviation Industry CBT Committee (AICC) made its courses' structure, AICC CMI001 Guidelines for Interoperability.

Table 1. Prerequisites' Translation Rules

Graphical Prerequisite	Formal Prerequisite	Script Prerequisite
	$\delta_i(\alpha_B) = \{\{\alpha_A\}^0\}$; $\kappa_i(\alpha_B) = \{\{= \text{"completed"}\}\}$;	<item identifier="α_B" … isvisible="true"> <title> α_B </title> <adlcp:prerequisites type="aicc_script"> <![CDATA[α_A="completed"]]> </adlcp:prerequisites> </item>
	$\delta_i(\alpha_B) = \{\{\alpha_A\}^0\}$; $\kappa_i(\alpha_B) = \{\{= \text{"TC}_1\text{"}\}\}$;	<item identifier="α_B" … isvisible="true"> <title> α_B </title> <adlcp:prerequisites type="aicc_script"> <![CDATA[α_A=**TC_1**]]> </adlcp:prerequisites> </item>
	$\delta_i(\alpha_D) = \{\{\alpha_B\}^1, \{\alpha_C\}^1\}$; $\kappa_i(\alpha_D) = \{\{= \text{"completed"}\},$ $\{= \text{"completed"}\}\}$;	<item identifier="α_D" … isvisible="true"> <title> α_D </title> <adlcp:prerequisites type="aicc_script"> <![CDATA[α_B= "completed" $\wedge$ α_C = "completed"]]> </adlcp:prerequisites> </item>
	$\delta_i(\alpha_E) = \{\{\alpha_B, \alpha_C, \alpha_D\}^2\}$; $\kappa_i(\alpha_E) = \{$ $\{= \text{"completed"},$ $= \text{"completed"},$ $= \text{"completed"}\}$ $\}$;	<item identifier="α_E" … isvisible="true"> <title> α_E </title> <adlcp:prerequisites type="aicc_script"> <![CDATA[2*{α_B = "completed", α_C = "completed", α_D = "completed"}]]> </adlcp:prerequisites> </item>
	$\delta_i(\alpha_E) =\{\{\alpha_B\}^1, \{\alpha_C, \alpha_D\}^1\}$; $\kappa_i(\alpha_E) = \{$ $\{= \text{"completed"}\},$ $\{= \text{"completed"},$ $= \text{"completed"}\}$ $\}$;	<item identifier="α_E" … isvisible="true"> <title> α_E </title> <adlcp:prerequisites type="aicc_script"> <![CDATA[α_B = "completed" $\wedge$ (α_C = "completed" $\vee$ α_D = "completed")]]> </adlcp:prerequisites> </item>

3.5 An Example of the e-Learning Process Control Model

As an example, let's consider a typical e-Learning course for the reading and comprehension lecture of TOEFL. Basically, by using the e-Learning process control model, we are able to build a content aggregation model of a course. **Fig. 5** shows the e-Learning course graphically modeled by the e-Learning process control model. It consists of five e-Learning activities from the placement test activity to the student evaluation activity. Assume that the detailed e-Learning activities are as follows.

- **Placement Test** (α_1, $item_1$): Learners take the placement test before joining the lecture. Its resource is a SCO_1 of a web-based exam sheet.
- **Reading & Comprehension Lecture** (α_2, $item_2$): Learners being classified into an expert group take this learning path. Its resource is a SCO_2 of a R/C lecture.
- **Vocabulary Lecture** (α_3, $item_3$): Learners being classified into a novice group take a vocabulary lecture prior to the R/C lecture. Its resource is a SCO_3 of a vocabulary lecture note.
- **Reading & Comprehension Lecture** (α_4, $item_4$): After taking the vocabulary lecture, learners take the R/C lecture. Its resource is a SCO_2 that is the same as the previous one.
- **Student Evaluation** (α_5, $item_5$): Learners are evaluated after the R/C lecture. Its resource is a SCO_4 of a web-based R/C problem sheet.

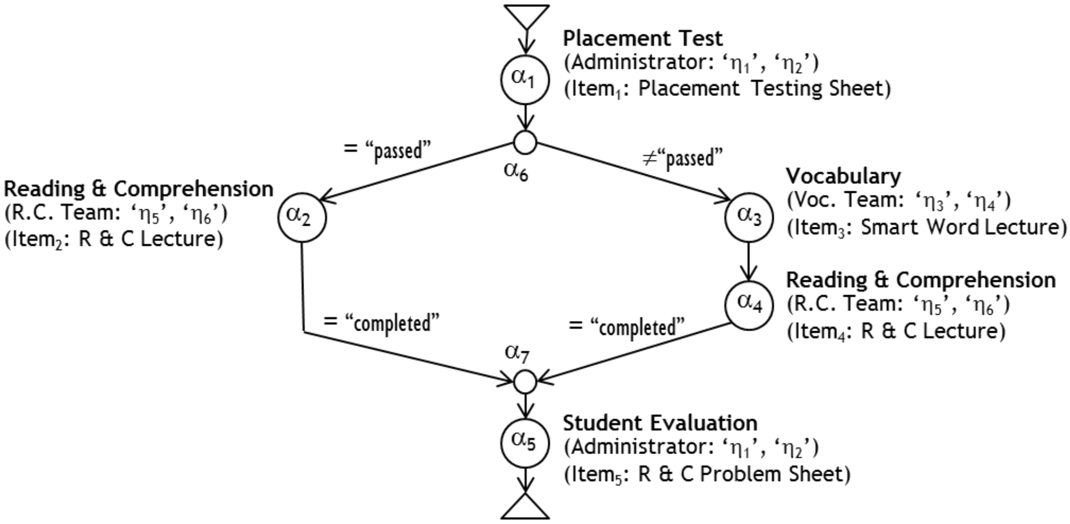


Fig. 5. An e-Learning TOEFL Course Using the e-Learning Process Control Model

Based on the graphical model of the e-Learning TOEFL course, we are able to define its formal model as shown in **Table 2**. In the table, we present just the pre-requisites and post-requisites of each e-Learning activity of the course , not all of the elements that have to be specified in the formal representation of the e-Learning process. Particularly, in order to extract SCORM 1.2's sequencing information from the e-Learning process control model of the TOEFL's e-Learning course, we need only the prerequisite e-Learning activities and their conditions in the table. Fortunately, we can use both of them to extract the sequencing information of SCORM 1.3 with a little modification of the e-Learning process control model, because the SCORM 1.3 sequencing rules defines the precondition rules and the postcondition rules as well.

Table 2. Formal Representation of the e-Learning TOEFL's Course

$\Gamma = (\delta, \vartheta, \xi, \gamma, \varepsilon, \pi, \kappa, I, O)$		over A, S, R, C, D, P, T
$\mathsf{A} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_I, \alpha_F\}$		$//Activities$
$\mathsf{S} = \{item_1, item_2, item_3, item_4, item_5\}$		$//Items$
$\mathsf{R} = \{SCO_1, SCO_2, SCO_3, SCO_4\}$		$//Resources$
$\mathsf{C} = \{(=\text{"}passed\text{"}), (\neq \text{"}passed\text{"}), (=\text{"}completed\text{"})\}$		$//Prerequisite - Conditions$
$\mathsf{D} = \{\emptyset\}$		$//Relevant - Data$
$\mathsf{P} = \{administrator, voc - team, rc - tem\}$		$//EducationalRoles$
$\mathsf{T} = \{\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6\}$		$//Educators$
$I = \{\emptyset\}$		$//InitialInput$
$O = \{\emptyset\}$		$//FinalOutput$
	Pre-requisites	**Post-requisites**
$\delta = \delta_i \cup \delta_o$	$\delta_i(\alpha_I) = \{\{\emptyset\}^0\};$	$\delta_o(\alpha_I) = \{\{\alpha_1\}^0\};$
	$\delta_i(\alpha_1) = \{\{\alpha_I\}^0\};$	$\delta_o(\alpha_1) = \{\{\alpha_2, \alpha_3\}^1\};$
	$\delta_i(\alpha_2) = \{\{\alpha_1\}^0\};$	$\delta_o(\alpha_2) = \{\{\alpha_5\}^0\};$
	$\delta_i(\alpha_3) = \{\{\alpha_1\}^0\};$	$\delta_o(\alpha_3) = \{\{\alpha_4\}^0\};$
	$\delta_i(\alpha_4) = \{\{\alpha_3\}^0\};$	$\delta_o(\alpha_4) = \{\{\alpha_5\}^0\};$
	$\delta_i(\alpha_5) = \{\{\alpha_2, \alpha_4\}^1\};$	$\delta_o(\alpha_5) = \{\{\alpha_F\}^0\};$
	$\delta_i(\alpha_F) = \{\{\alpha_5\}^0\};$	$\delta_o(\alpha_F) = \{\{\emptyset\}^0\};$
	Prerequisite-Conditions	**Postrequisite-Conditions**
$\kappa = \kappa_i \cup \kappa_o$	$\kappa_i(\alpha_I) = \{\emptyset\};$	$\kappa_o(\alpha_I) = \{\{=\text{"}completed\text{"}\}\};$
	$\kappa_i(\alpha_1) = \{\{=\text{"}completed\text{"}\}\};$	$\kappa_o(\alpha_1) = \{\{=\text{"}passed\text{"}\}, \{\neq \text{"}passed\text{"}\}\};$
	$\kappa_i(\alpha_2) = \{=\text{"}passed\text{"}\};$	$\kappa_o(\alpha_2) = \{=\text{"}completed\text{"}\};$
	$\kappa_i(\alpha_3) = \{\neq \text{"}passed\text{"}\};$	$\kappa_o(\alpha_3) = \{\{=\text{"}completed\text{"}\}\};$
	$\kappa_i(\alpha_4) = \{\{=\text{"}completed\text{"}\}\};$	$\kappa_o(\alpha_4) = \{=\text{"}completed\text{"}\};$
	$\kappa_i(\alpha_5) = \{\{=\text{"}completed\text{"}\}, \{=\text{"}completed\text{"}\}\};$	$\kappa_o(\alpha_5) = \{\{=\text{"}completed\text{"}\}\};$
	$\kappa_i(\alpha_F) = \{\{=\text{"}completed\text{"}\}\};$	$\kappa_o(\alpha_F) = \{\emptyset\};$

Ultimately, the e-Learning TOEFL course based on the e-Learning process control model is encoded in a XML-based specification forming a content aggregation model with the SCORM 1.2's sequencing information. The e-Learning process control model can straightforwardly transform the e-Learning TOEFL course into a content aggregation model, as followings.

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"  identifier="SAMPLE1" version="1.2"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
http://ltsc.ieee.org/xsd/LOM lom.xsd http://www.adlnet.org/xsd/adlcp_v1p3
adlcp_v1p3.xsd http://www.imsglobal.org/xsd/imsss imsss_v1p0.xsd
http://www.adlnet.org/xsd/adlnav_v1p3 adlnav_v1p3.xsd">
    <metadata>
              <schema>ADL SCORM</schema>
              <schemaversion>Version 1.2</schemaversion>
    </metadata>
    <organizations default="TOC0">
       <organization identifier="TOC1" >
          <title>English (TOEFL) eLearning Course</title>
              <item identifier="ITEM1" identifierref="RESOURCE1" isvisible="true">
                     <title>Placement Test</title>
                     <adlcp:masteryscore>85</adlcp:masteryscore>
              </item>
              <item identifier="ITEM2" identifierref="RESOURCE2" isvisible="true">
                     <title>Reading & Comprehension</title>
                     <adlcp:prerequisites type="aicc_script">
                            <![CDATA[ITEM1="passed"]]>
                     </adlcp:prerequisites>
              </item>
              <item identifier="ITEM3" identifierref="RESOURCE3" isvisible="true">
                     <title>Vocabulary</title>
                     <adlcp:prerequisites type="aicc_script">
                            <![CDATA[ITEM1≠"passed"]]>
                     </adlcp:prerequisites>
              </item>
              <item identifier="ITEM4" identifierref="RESOURCE2" isvisible="true">
                     <title>Reading & Comprehension</title>
                     <adlcp:prerequisites type="aicc_script">
                            <![CDATA[ITEM3]]>
                     </adlcp:prerequisites>
```

```
              </item>
              <item identifier="ITEM5" identifierref="RESOURCE4" isvisible="true">
                     <title>Student Evaluation</title>
                     <adlcp:prerequisites type="aicc_script">
                            <![CDATA[ITEM2 = "completed" | ITEM4 = "completed"]]>
                     </adlcp:prerequisites>
                     <adlcp:masteryscore>90</adlcp:masteryscore>
              </item>
       </organization>
    </organizations>
    <resources>
       <resource identifier="RESOURCE1"
           type="webcontent" adlcp:scormtype="sco" href="Placement_Testing_Sheet.htm">
                     <file href="assets/image/image01.gif" />
                     <file href="Placement_Testing_Sheet.htm" />
       </resource>
       <resource identifier="RESOURCE2"
           type="webcontent" adlcp:scormtype="sco" href="R_C_Lecture.htm">
                     <file href="R_C_Lecture.htm" />
                     <file href="Reading.htm" />
                     <file href="Comprehension.htm" />
       </resource>
       <resource identifier="RESOURCE3"
           type="webcontent" adlcp:scormtype="sco" href="Smart_Word_Lecture.htm">
                     <file href="assets/image/image02.gif" />
                     <file href="Smart_Word_Lecture.htm" />
                     <file href="Vocabulary_Example.htm" />
       </resource>
       <resource identifier="RESOURCE4"
           type="webcontent" adlcp:scormtype="sco" href="R_C_Problem_Sheet.htm">
                     <file href="R_C_Problem_Sheet.htm" />
       </resource>
    </resources>
</manifest>
```

4. An e-Learning Process Modeling System

Fig. 6 illustrates the overall system architecture of a typical SCORM-based e-Learning process management system (eLPMS), which will be implemented in the near future by collaborative works of the graphics and image processing lab and the collaboration technology research lab of Kyonggi University. The system is based upon the SCORM-based e-Learning process control model stated in the previous sections, which is also becoming a theoretical basis for implementing a SCORM-based e-Learning process modeling system. Conclusively, the e-Learning process control model has got to affect not only the SCORM content aggregation model but also the SCORM runtime environment so as for SCORM to adapt the process-driven content aggregating approach. In this section, we simply introduce the architectural components of the system and a modeling system supporting the proposed approach and its model.

4.1 Architectural Components of the eLPMS

We briefly describe an achievable architecture of an e-Learning process management system, which is based upon the e-Learning process control model proposed in this paper. As shown in **Fig. 6**, the system is composed of three types of software components and one external product. The three types of software components are a graphical modeling tool, an enacting engine and a runtime-client, whereas the external product is a SCO authoring tool with SCO silos (SCORM-based e-Learning contents). These components collaboratively enable us to accomplish two major functionalities provided by the conventional e-Learning management tools; one is the e-Learning process enacting functionality, and the other is the e-Learning content aggregating functionality based upon the e-Learning process control model. In summary, the system's physical components and the service interfaces among them are as followings.

1) The physical components
   - Content aggregation modeling component
   - e-Learning process enacting component
   - Learner's, educator's and supervisor's runtime clients component
2) The service interfaces
   - Interface 1: between the modeling component and the enacting component
   - Interface 2: between the runtime client component and the enacting component
   - Interface 3: between the SCO authoring and archiving component and the enacting component
   - Interface 4: Interoperability between the enacting component and the enacting components in other systems

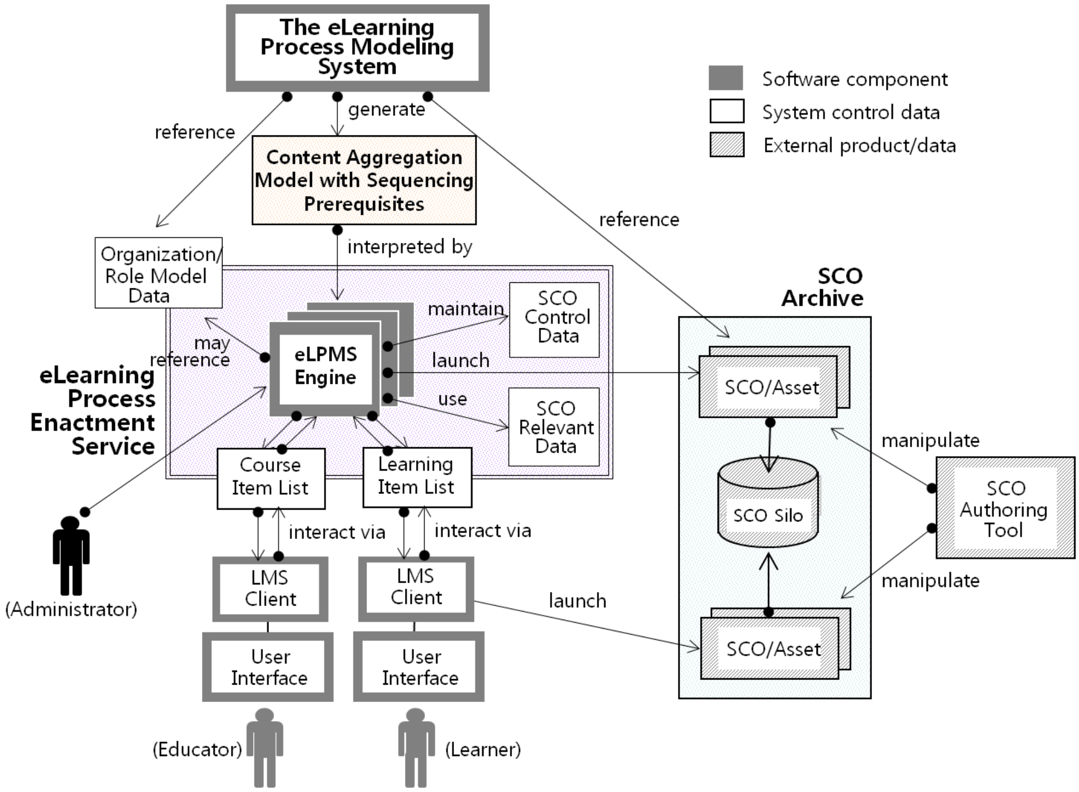


Fig. 6. System Architecture of the SCORM-based e-Learning Process Management System

The scope of this paper is to implement the e-Learning contents aggregating functionality supporting the proposed model, the e-Learning process control model. In the next section, we give an example of the implemented modeling system showing how to automatically aggregate e-Learning contents with manifesting the sequencing prerequisites. **Fig. 7** shows the schema structure of the SCORM's e-Learning package manifest. As you see in the schema, the organization model consists of e-Learning items and their structural relations, and each e-Learning item has several elements to specify its properties. The prerequiste element is one of the most important elements to represent an e-Learning sequencing behavior of the corresponding e-Learning process.

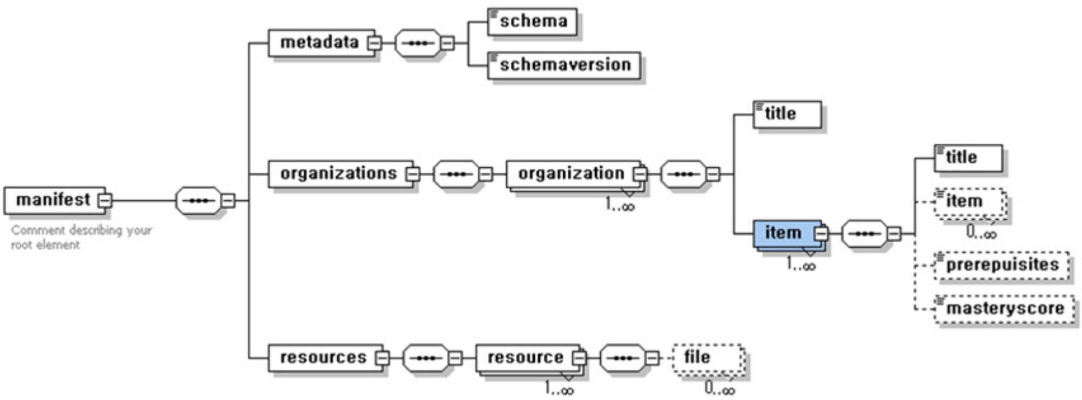


Fig. 7. Schema Structure of the SCORM's e-Leaning Package Manifest

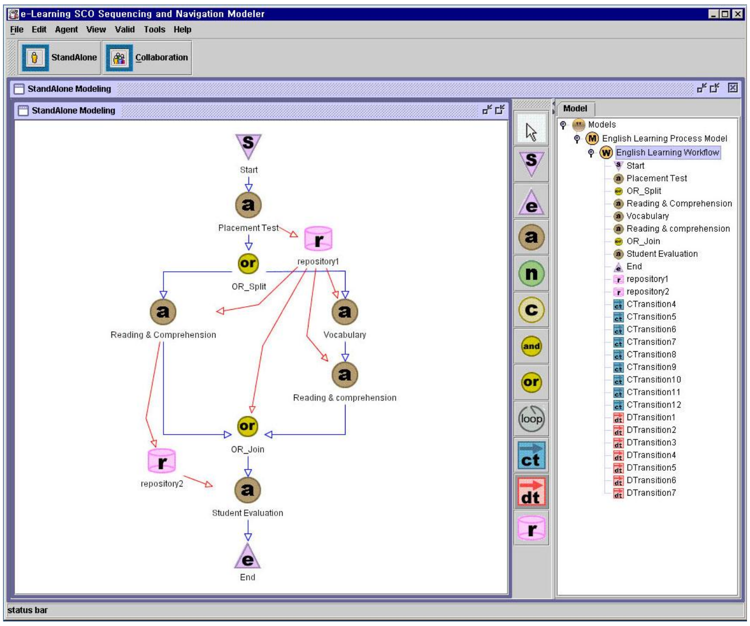


Fig. 8. Snapshot of the e-Learning Process Modeling System

4.2 The e-Learning Process Modeling System

We will not state all of the components and the service interfaces of the system, here. This paper's main concern is about the e-Learning process control modeling component. That is, we newly formalize a theoretical basis for implementing the system that is based upon the model of the sequencing prerequisites approach. The model is possibly incorporated into not only the SCORM content aggregation model, but also the SCORM runtime environment in order to enhance the efficiency and flexibility of e-Learning management systems. Also, if the current SCORM can be supplemented with the e-Learning process control model in specifying the content aggregation model, it can support a much more sophisticated and convenient way for content developers to define the intended sequencing and ordering of e-Learning activities.

Fig. 8 shows a snapshot of the e-Learning process control modeling system that is currently being developed by the collaborative research group, while **Fig. 9** shows the output of the modeling system for the e-Learning TOEFL course examplified in the previous section.


```
<?xml version="1.0" encoding="UTF-8" ?>
- <manifest>
  - <metadata>
      <schema>ADL SCORM</schema>
      <schemaversion>Version 1.2</schemaversion>
    </metadata>
  - <organizations default="Package1275986580475">
    - <organization identifier="Process-1275986649968">
        <title>English Learning Process Model</title>
      - <item identifier="SCORM1275986771702" identifierref="Resource1" isvisible="true">
          <title>Placement Test</title>
          <prerepuisites>SCORM1275986740385</prerepuisites>
          <masteryscore>5</masteryscore>
        </item>
      - <item identifier="SCORM1275987310939" identifierref="Resource2" isvisible="true">
          <title>Reading & Comprehension</title>
          <prerepuisites>SCORM1275986771702</prerepuisites>
          <masteryscore>6</masteryscore>
        </item>
      - <item identifier="SCORM1275987313205" identifierref="Resource3" isvisible="true">
          <title>Vocabulary</title>
          <prerepuisites>SCORM1275986771702</prerepuisites>
          <masteryscore>7</masteryscore>
        </item>
      - <item identifier="SCORM1275987338974" identifierref="Resource4" isvisible="true">
          <title>Reading & Comprehension</title>
          <prerepuisites>SCORM1275987313205</prerepuisites>
          <masteryscore>8</masteryscore>
        </item>
      - <item identifier="SCORM1275987373212" identifierref="Resource0" isvisible="true">
          <title>Student Evaluation</title>
          <prerepuisites>(SCORM1275987310939 Or SCORM1275987338974)</prerepuisites>
          <masteryscore>8</masteryscore>
        </item>
      - <item identifier="SCORM1275987379557" identifierref="" isvisible="true">
          <title />
          <prerepuisites>SCORM1275987373212</prerepuisites>
          <masteryscore>1</masteryscore>
        </item>
      </organization>
    </organizations>
  - <resources>
    - <resource identifier="Resource0" type="webcontent" scormtype="SCO" href="..\scorm\advanced">
        <file href="..\scorm\advanced\10.html" />
      </resource>
    - <resource identifier="Resource1" type="webcontent" scormtype="SCO" href="..\scorm\img">
        <file href="..\scorm\img\1-1.jpg" />
        <file href="..\scorm\img\1.jpg" />
        <file href="..\scorm\img\10-1.jpg" />
        <file href="..\scorm\img\10-2.jpg" />
        <file href="..\scorm\img\10-3.jpg" />
        <file href="..\scorm\img\2.jpg" />
        <file href="..\scorm\img\3.jpg" />
        <file href="..\scorm\img\4.jpg" />
        <file href="..\scorm\img\5.jpg" />
        <file href="..\scorm\img\6.jpg" />
        <file href="..\scorm\img\7-1.jpg" />
        <file href="..\scorm\img\7.jpg" />
        <file href="..\scorm\img\8.jpg" />
        <file href="..\scorm\img\9.jpg" />
```


Fig. 9. The SCORM Contents Aggregation Model for theTOEFL's e-Learning Process Control Model Automatically Generated by the Modeling System

5. Conclusion

In this paper, we have presented the e-Learning process control model for realizing a content aggregating approach with sequencing prerequisites. Particularly, in this paper we pointed out the limitations of the current SCORM sequencing and navigation model and proposed a feasible solution for resolving the limitations, as well. Especially, our approach is able to easily represent the sequencing behavior of e-Learning activities and experiences by using the four types of sequencing prerequisites, such as sequential, conjunctive (parallel-style), disjunctive (alternative-style), and **X*** (selective-conjunction) prerequisites, without any further restrictions. Through the proposed model, it is possible to easily define diverse and multiple e-Learning processes on a single content organization model, so these multiple e-Learning processes can be effectively and efficiently applied to educating diverse levels of students and trainees for the same e-Learning content package. We would point out that providing the easy way of defining multiple e-Learning processes on a single e-Learning content organization model is the most important contribution of the paper.

In summary, before the advent of the SCORM and the shift toward an interoperable development strategy, it was extremely difficult to share contents between different authoring environments and equally difficult to reuse contents in other contexts containing different sequencing requirements. Within SCORM, those difficulties were eliminated by defining the sequencing information based on those Activities, which are represented in the Content Organization Model and external to the learning resources associated with the Activities, as well. However, SCORM still has some limitations and difficulties in representing the sequencing information. Therefore, we strongly believe that our approach be a remedy for eliminating the SCORM's limitations as simply pointed out in the earlier section.

References

[1] “SCORM 2004 Overview,” Advanced Distributed Learning, 2004

[2] “SCORM Content Aggregation Model, Version 1.2,” Advanced Distributed Learning, 2002

[3] “SCORM Content Aggregation Model, Version 1.3,” Advanced Distributed Learning, 2004

[4] “SCORM Run-Time Environment, Version 1.3,” Advanced Distributed Learning, 2004

[5] “SCORM 2004 3rd Edition: Sequencing and Navigation, Version 1.0,” Advanced Distributed Learning, 2006

[6] C.A. Ellis, G.J. Nutt, “Office Information Systems and Computer Science,” *ACM Computing Surveys*, vol. 12, no. 1, 1980. Article (CrossRef Link)

[7] C.A. Ellis, “Formal and Informal Models of Office Activity,” in *Proc. of the 1983 Would Computer Congress*, Paris, France, 1983.

[8] K.-H. Kim, et al., “SCO Control Net for the Process-driven SCORM Contents Aggregation Model,” in *Proc. of ICCSA 2005*, LNCS 3483, pp. 38-47, 2005. Article (CrossRef Link)

[9] K.-H. Kim, C.A. Ellis, “ICN-Based Workflow Model and Its Advances,” *Handbook of Research on Business Process Modeling*, IGI Global, 2009. Article (CrossRef Link)

[10] H. Zhuge, “The Knowledge Grid,” World Scientific Publishing Co., Singapore, 2004.

[11] H. Zhuge, “Communities and Emerging Semantics in Semantic Link Network: Discovery and Learning,” *IEEE Transactions on Knowledge and Data Engineering*, July 2008. Article (CrossRef Link)

[12] N. Park, et al., “An e-Learning Activity Control Model for SCORM's Sequencing Prerequisites,” in *Proc. of IEEE Fourth International Conference on Semantics, Knowledge and Grid*, pp. 322 - 329, 2008. Article (CrossRef Link)

[13] H. Kim, et al., “A Prerequisite-driven e-Learning Activity Modeling System,” in *Proc. of KSII First International Conference on Internet (ICONI 2009*, pp. 331-338, 2009.

Hyunah Kim is a post-doctoral member of research staff in the Gyeonggi Regional Research Center, Contents Convergence Software Research Center, at Kyonggi University. She received a B.S. degree in computer science from Nazarene University in 2001. She received M.S. and Ph.D. degrees in computer science from the Graduate School of Kyonggi University, South Korea, in 2003 and 2009, respectively. Her research interests include workflow systems, BPM, SCORM, and e-Learning management systems.

Eunjung Lee is a professor of the computer science department in Kyonggi University, since 2001. She received her Ph.D. degree in computer science from KAIST in 1994. At KAIST, she studied compilers and parsing theories. Recently, she is interested in programming language-related research topics on internet and web services technologies.

Junchul Chun is currently a professor in the department of computer science and a principal investigator in the Graphics and Image Processing Research Laboratory at Kyonggi University, Suwon South Korea. He received a B.S. degree from Chung-Ang University majoring in computer science. He also received M.S. and Ph.D. degrees from the department of computer science and engineering at the University of Connecticut, CT., U.S.A, in 1992 and 1995, respectively. He was a visiting research scholar in PRIP (Pattern Recognition and Image Processing) at Michigan State University and University of Colorado at Boulder, 2001 and 2009 respectively. He is currently a chief director of CCSRC (Contents Convergence Software Research Center). His major research fields are augmented reality, vision-based animation, and human computer interaction.

Kwanghoon Pio Kim is a professor in the computer science department and the supervisor of the collaboration technology research laboratory at Kyonggi University, South Korea. He received a B.S. degree in computer science from Kyonggi University in 1984. And he received a M.S. degree in computer science from Chungang University in 1986. He also received his M.S. and Ph.D. degrees in computer science from the University of Colorado at Boulder, in 1994 and 1998, respectively. He had worked as a researcher and developer at Aztek Engineering, American Educational Products Inc., and IBM in USA, as well as at Electronics and Telecommunications Research Institute (ETRI) in South Korea. In present, he is a vice-chair of the BPM Korea Forum, a vice-chair of the Workflow Management Coalition, and a vice-president of the Korean Society of Internet Information. His research interests include groupware, workflow systems, BPM, CSCW, collaboration theory, Grid/P2P/Cloud distributed systems, workflow intelligence, workflow knowledge discovery and rediscovery, and e-Learning process models and systems.