

# SCORM Sequencing Testing for Sequencing Control Mode

Hsuan-Pu Chang, Chun-Chia Wang\*, Kuen Han Jan and Timothy K. Shih

*Department of Computer Science and Information Engineering*

*Tamkang University, Taiwan*

*\*Northern Taiwan Institute of Science and Technology*

*Beitou, Taipei, Taiwan 112*

*musicbubu@gmail.com*

## Abstract

*In SCORM 2004 defines the sequencing information that describes how SCORM-conformant content may be sequenced to the learner through a set of learner or system-initiated navigation events. It provides users the ability to prescribe the intend learning sequencing strategy by themselves, but quit many completed definitions and lacking the testing mechanism for these authored sequencing information results in usual developers probably design the unreasonable or careless settings of SCORM sequencing.*

*The detecting mechanism focuses on detecting improper setting of Sequencing Control Mode elements applied to learning activities. An assistant truth table derived from the definitions of Sequencing Control Mode elements and experiments verified with latest ADL runtime environment will be introduced.*

**Keywords:** *ADL, SCORM, Learning Sequencing, Sequencing Control Definition, Sequencing Control Model, Detecting Algorithm, Truth Table*

## 1. Introduction

Advanced Distributed Learning (ADL) proposes SCORM (Shareable Content Object Reference Model) that aims to provide the specifications necessary to enable content developers with the ability to produce content that is sharable, accessible, reusable, and most importantly interoperable. Parts of SCORM Sequencing and Navigation[1] are originally derived from the IMS[2] Simple Sequence Specification (SSS) which was first introduced and developed in the SCORM 1.3 specification. The IMS SSS allows components in a courseware to be specified in a relative order, in which conditions and selections are precisely defined. The SCORM Sequencing extends from IMS SSS defines the required behaviors and functionalities for SCORM-conformant LMSs to be

implemented to process sequencing information at run time. More specially, it describes the branching and flow of learning activities in terms of an Activity Tree, based on the result of a learner's interactions with launched content objects and an authored sequencing strategy.

SCORM sequencing and navigation spec. defines quit intricate modes and behaviors. Sequencing Control Mode is one of these defined modes which allows the content developers to affect how navigation requests are applied to a cluster and how the cluster's activities are considered while processing sequencing requests. According to our proposed architecture and algorithm, authors can trigger the sequencing detecting function to verify the authored values of Sequencing Control Mode elements during editing the learning content and sequencing strategy. As a result, the robust sequencing information can be interoperably transferred between systems and reduce the cost of revising improper sequencing strategies.

## 2. Related Works

With the rapid development of e-learning technology and the popularity of SCORM, more and more tools are developed to facilitate the practical usage for e-learning applications. Some related SCORM sequencing tools such as providing an user-friendly interface to assist teachers/instructors efficiently construct SCORM compliant courses with learning sequencing information [4] and a visualized online Simple Sequencing authoring tool to help instructors edit existing SCORM-compliant content packages[5]. The study tries to decrease the complexity of the sequencing definition by applying the features of Petri net can be found at [6]. Academic ADL Co-Lab also develops a Sequencing Visualizer Tool[7] that using graphical representation of the imsmanifest XML file and programmers are able to step through content

to ensure all rules are working as desired. The rest content of this section will introduce the cluster definition and elements defined in Sequencing Control Mode.

## 2.1 Cluster definition

As shown in Figure 2.1, an activity tree can be separated into several smaller cluster units which considered as a basic building block of learning activity. SCORM sequencing is especially applied to clusters. The parent activity of the cluster will contain the learning sequencing information. The child activities in the cluster will have associated learning content and objects. The cluster is defined for learning sequencing strategy.

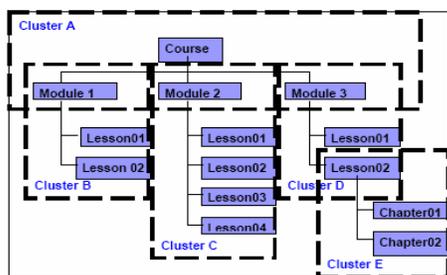


Figure 2.1: Cluster Example

## 2.2 Sequencing Control Modes

The control modes can be used in the following ways:

- During processing of a navigation request to determine whether or not the request will translate into a valid sequencing request.
- During various sequencing request for sub-processes to affect the activities for delivery.
- During various sequencing behaviors to affect how tracking status information is managed.

There are six elements defined in Sequencing Control Modes includes Sequencing Control Choice, Sequencing Control Choice Exit, Sequencing Control Flow, Sequencing Control Forward Only, Use Current Attempt Objective Information and Use Current Attempt Progress Information. The last two elements are used for Tracking Model which will not be discussed in this paper. The following short descriptions of the other elements in Sequencing Control Model would directly affect the learner's learning sequencing.

### 2.2.1 Sequencing Control Choice

The Sequencing Control Choice element indicates

that the learner is free to choose any activity in a cluster in any order without restriction. If a parent activity has Sequencing Control Choice element with default value; true, then its child activities are all valid targets for a Choice navigation request. The Sequencing Control Choice control mode has no affect when defined on a leaf activity.

### 2.2.2 Sequencing Control Choice Exit

Sequencing Control Choice Exit indicates whether a Choice navigation request can target activities that are not descendants of the affected activity, and causes the affected activity to terminate. *ChoiceExit* only can be applied to an active activity. The default value for Sequencing Control Choice Exit, if not explicitly defined for the activity, is True.

### 2.2.3 Sequencing Control Flow

In Sequencing Control Flow, the system directed sequencing through the child activities of a cluster is supported. The default value for Sequencing Control Flow, if not explicitly defined for an activity, is False.

### 2.2.4 Sequencing Control Forward Only

The Sequencing Control Forward Only indicates that system directed sequencing through the child activities of the cluster is constrained to disallow previous navigation request and to disallow Choice requests that would move in a backward direction.

## 3. Sequencing Errors Detecting Concept

The improper sequencing information would lead a learner to attend unanticipated learning activities, or causes unexpected situations such as learner's attempt is stuck in a certain activity or some relevant activities are not able to be identified for delivery always. According to the experiences of designing sequencing information, the following unexpected situations probably may occur during run-time.

- Learning attempt is stuck in a certain activity
- Learning attempt is fallen into a vicious learning circle.
- Deserved activities are not identified for delivery

Of course, these unexpected situations could have reciprocal effect to each other.

### 3.1 Reduce the problem and build the truth table

We reduce the detecting problem to the simplest situation, a single cluster, which has a parent activity with a single child activity or a parent activity with multi-child activities.

As mentioned before, there are four Sequencing Control Model elements directly affect the learning sequencing navigation requests. Therefore for the simplest case, there are totally  $2^8$  combinations to construct the truth table. According to the different characteristics of Sequencing Control Choice, Sequencing Control Flow and Sequencing Control Forward have no effect when they are defined on a leaf activity and Sequencing Control Choice Exit can be only applied to active activities. Therefore all possible authored cases can be simplified to sixteen cases as shown in table 3.1. The result of each case in table 3.1 is derived from the experiments tested in the latest version of ADL runtime environment and analyzing the reciprocal effect between the sequencing navigation behaviors defined in Sequencing Spec.

**Table 3.1**

Case	Parent Activity			Child Activity	Block
	Flow	Forward Only	Choice	Choice Exit	
1	False	False	False	False	True
2	False	False	False	True	True
3	False	False	True	False	True
4	False	False	True	True	False
5	False	True	False	False	True
6	False	True	False	True	True
7	False	True	True	False	True
8	False	True	True	True	False
9	True	False	False	False	False
10	True	False	False	True	False
11	True	False	True	False	False
12	True	False	True	True	False
13	True	True	True	False	False
14	True	True	True	True	False
15	True	True	False	False	False
16	True	True	False	True	False

We take two single cluster examples to explain the blocking cases in table 3.1. In figure 3.1, the sequencing setting of this single cluster corresponds to blocking cases includes case 5, case 6 and case 7. In this cluster the Parent Activity has Flow defined as false. LMS will not provide any mechanism for the learner to indicate their desire to “Continue” to the next activity or to go back to a “Previous” activity. But the child activities are all valid targets for learners to

choose due to the Sequencing Control Choice defined as true. If a learner chooses Lesson2, the learner’s attempt will always be stuck in this activity. The reason is that when a learner’s attempt is on Lesson2, the parent activity of Lesson2 has Sequencing Control Choice defined as true so every sibling of Lesson2 is a valid target for a Choice navigation request. But allow one of Lesson1 or Lesson3 or Lesson4 to be identified for delivery would result in Lesson2 terminating, violating the intention of Choice Exit control. At the moment, the learner can neither trigger Flow nor Choice navigation requests, so there is no activity could be identified for delivery. The figure 3.2 shows the real content package which has the same sequencing information with Block Example 1 is running in ADL runtime environment. After a student clicks the Lesson2, the content aggregation tree and all navigation requests button are disappeared except suspend and quit request buttons for this learning process.

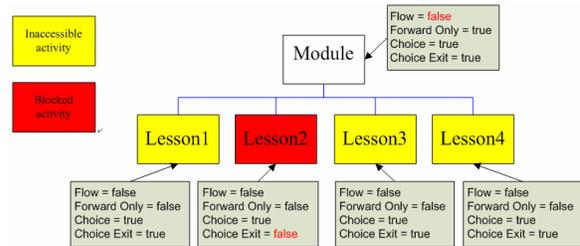


Figure 3.1 Blocking Example 1

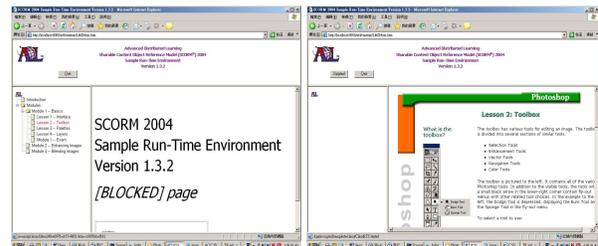


Figure 3.2 Blocking Example 1 tested on ADL runtime environment

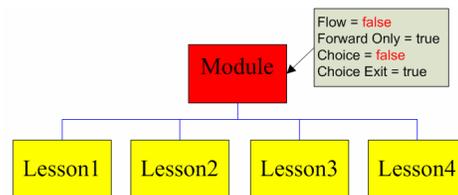


Figure 3.3 Blocking Example 2

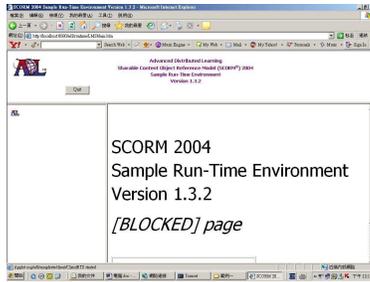


Figure 3.4 Blocking Example 2 tested on ADL runtime environment

The second example is shown in figure 3.3, a Parent Activity has Flow and Choice defined as false. LMS will disable the *Continue* and *Previous* navigation requests. All child activities will not be valid targets for learner to free choose either due to the parent activity has Sequencing Control Choice defined as false. As the result, a learner could be stuck in front of the Module node and Lesson1 to Lesson4 all become inaccessible nodes in this Activity Tree. This situation corresponds to blocking case includes case 1, case2 and case3 in Table 3.1. The practical content have to be tested on latest ADL runtime environment, the result is shown in figure 3.4.

### 3.2 Detecting Algorithm

In previous section, we introduce an assistant truth table constructed by Sequencing Control Elements definitions and behaviors. The following detecting algorithm extends the detecting mechanism to a general course consists of more than one single cluster. Furthermore the algorithm is based on recursion such that no matter the input course aggregation tree is a partial tree or a complete aggregation tree can be worked well.

```

Given:
SCT: Sequencing Control Truth Table
CAT: Course Aggregation Tree
CC: Course Cluster {
  Cluster Number;
  Sequencing Information
}
Steps:
j = 0;
SCA(CAT, CT, j){
  If(CAT can not be separated into any cluster anymore){
    Return;
  }
  Else{
    Separate a cluster unit CU from CAT
    according to Depth First Search.
    j = j + 1;
    Add CU to CT;
    CU.Cluster Number = j;
    SCA(CAT, CT, j);
  }
}

```

```

}
For(i = CT.length; i >= 0; i--) {
  Compare the CT[i].Sequencing Information with SCT;
  List the compared result and CT[i].Cluster Number.
}

```

An input content aggregation tree is going to be separated into several defined clusters according to the Depth First Search. The sequencing information defined in the individual cluster will also be saved. When the tracking status of an activity changes, the tracking status of its ancestors may be affected. Consequently, we use the bottom-up method to detect the collected cluster unit from the cluster tree. One of the children of a cluster is possible be a parent activity of another cluster. The basic idea is as long as we ensure the accuracy the cluster's all children clusters and then the cluster's accuracy can be ensured as well. Extend the recursive method from a single cluster can ensure the accuracy of a whole cluster tree.

## 4. Implementation and Experiment

In this section, we will use two different examples to demonstrate the detecting interface on application and web page respectively. In figure 4.1, Hard SCORM Authoring Tool[8] provides a click button to trigger the detecting mechanism for current authoring sequencing strategy anytime. Users can trace and understand the statuses of current authored sequencing information by visualizing results while course editing phase. It does not matter the editing course process is finished or not. The activity in red indicates that students might be stuck here and consequently its child activities or sibling activities will probably become inaccessible nodes expressed by yellow background color. At the same time, the program will list the authored sequencing information of red color activity and warning messages to describe the critical situation.

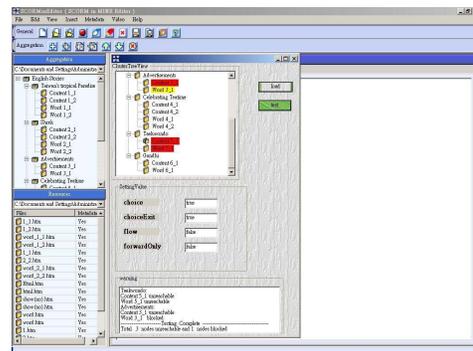


Figure 4.1 Detecting results on MINE SCORM authoring tool

We take another example course designed by a Junior High School teacher. Just like general teachers who dose not very familiar with the all sequencing definition but like to add the applicable sequencing strategy to her teaching materials. She used general authoring tool without sequencing error detecting function to edit learning content and define sequencing information. Although accuracy of her authored sequencing information can not be ensured at authoring client, the detecting function can be executed at our LMS as well. The course package uploaded to our developing LMS which provides sequencing error-detecting option that is similar to ADL runtime server has testing mechanism to ensure the uploading package containing complete SCORM Conformance. The detecting results are shown in Figure 4.2. In this case, Server found six inaccessible nodes and a blocking node, related detail information is shown at the bottom of this figure. LMS just indicates the possible improper sequencing information but does not constrain user to upload the course package.

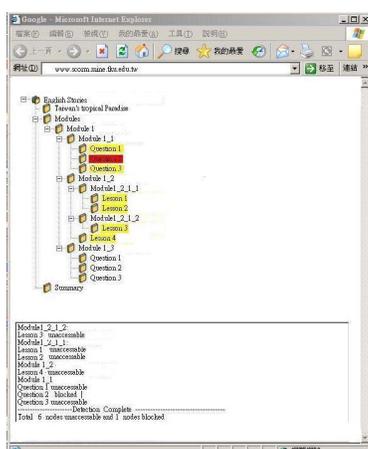


Figure 4.2 Detecting results on web page

## 5. Conclusion and Future Work

The proposed detecting mechanism for SCORM Sequencing can be used to help content developers or teachers to design a more robust sequencing strategy applied to learning activity. An authoring tool with the sequencing error detecting mechanism can ensure the accuracy of current defined sequencing strategy at editing learning materials phase. Our MINE SCORM LMS also provides a detecting option for other authored SCORM-compliant course package designed by general authoring tools without sequencing error detecting mechanism. Our main contributions are not only reducing the cost of revising uploaded courses with careless or improper sequencing information but also help general users have the inclination to try the

complicated sequencing definition. For the moment, our testing task focused on the generally used Sequencing Control Mode. Extending the sequencing errors detecting mechanism to other models and rules included in Sequencing Control Definition Model is our main future work.

## 6. References

- [1] Sequencing and Navigation (SN) 2004, Sharable Content Object Reference Model (SCORM) Sequencing and Navigation (SN) Version 1.3, Advanced Distributed Learning.<http://www.adlnet.org/index.cfm?fuseaction=DownFile&libid=648&bc=false>
- [2] Instructional Management System (IMS) 2004, IMS Global Learning Consortium.<http://www.improject.org/>
- [3] Sharable Content Object Reference Model (SCORM) 2004, Advanced Distributed Learning.<http://www.adlnet.org/>
- [4] Jun-Ming Su; Shian-Shyong Tseng; Jui-Feng Weng; Kuan-Ting Chen; Yi-Lin Liu; Yi-Ta Tsai (2005), "An Object Based Authoring Tool for Creating SCORM Compliant Course", Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)
- [5] Yang, J.-T.D.; Chiung-Hui Chiu; Chun-Yen Tsai; Tsung-Hsien Wu (2004) "Visualized online simple sequencing authoring tool for SCORM-compliant content package", Proceedings of the Advanced Learning Technologies (ICALT.2004)
- [6] Lin, H.W.; Shih, L.K.; Wen-Chih Chang; Chao-Hsun Yang; Chun-Chia Wang; (2004), "A Petri nets-based approach to modeling SCORM sequence", Proceedings of 2002 IEEE International Conference Multimedia and Expo, 2004. (ICME '04)
- [7] Academic Advanced Distributed Learning (ADL) Co-Lab, <http://www.academiccolab.org/initiatives/tools.html>
- [8] Shih, T.K.; Nigel H. Lin; Wen-Chih Chang; Te-Hua Wang; Hsiao Wen Lin; Hsuan-Pu Chang; Kuan-Hao Huang; Yun-Long Sie; Mon-Ting Tzou; Jin-Tan Yang(2005), "The Hard SCORM LMS: Reading SCORM Courseware on Hardcopy Textbooks", Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT 2005)