Building an Object-Oriented Learning Object in a Distributed Learning Environment

Yu-fang Yeh\textsuperscript{a}, Hsiu-yuan Wang\textsuperscript{a}, Chih-kai Chang\textsuperscript{b}

Department of Business Education
\textsuperscript{a}National Changhua University of Education

Department of Information Management
\textsuperscript{b}I-Shou University

ABSTRACT

The increase in the amount of learning content on the internet has made the reusability and compatibility of learning content, between different Learning Management Systems (LMSs), extremely important. Although the Sharable Content Object Reference Model (SCORM) was intended to solve these problems, the designing of learning content flexibly is still an open issue. As learning content is required to be adaptive to the learners’ knowledge level and learning ability, the design and management of learning content is more complex, and difficult to share among LMSs. This paper develops a design approach using adaptive learning content, which applies Object-Oriented concepts and distributed computing technologies. The learning object architecture presented in this paper was based on a SCORM compatible LMS, and provides distributed serving and distributed learning content. The Object-Oriented Learning Object (OOLO), also presented in this paper, assists the learning content designer in embedding objectives, assessments and learning strategies into learning objects; in addition, it facilitates the learning object management process by applying Web Service Technologies to develop and deliver adaptive SCORM compatible learning content.
Keywords: Object-Oriented; distance learning; learning object

1. INTRODUCTION

With the rapid development of network technologies, delivering learning content via the internet has become a popular approach of instruction in most educational organizations [1]. Distance learning not only could reduce training costs, but also could provide more adaptive learning content for the individual learner [2]. In most learning management systems (LMSs), the learning content has been designed pedagogically for a specific purpose, and does not lend itself to repeated use. To increase the reusability of learning content, learning objects (LOs), which contain learning content and learning strategy information, were introduced [3]. Learning principles should be considered in constructing these program. Bloom's taxonomy, for example, which was used to identify the cognitive levels of learning objects, including knowledge, comprehension, application, analysis, synthesis and evaluation [4][5]. Bloom's taxonomy is probably the most widely applied one in describing learning objectives of specific learning objects, which allow the learning system to discern a learner's knowledge level. A learning process, based on objective-centered learning objects, begins by assessing the learner’s knowledge background, progressing to the proper LO according to the gross deviation from expected learning goals. Although object based learning content design increases the potential for learning content reusability, the problem is that the smaller the LO, the more complex its management. The latest version ofSharable Content Object Reference Model (SCORM) uses a Sharable Content Object (SCO) to represent the LO, which composes of several Sharable Content Assets (SCAs) [6]. The SCORM defines a set of standard of learning content and system, namely a Content Aggregation Model (CAM) and Run-Time Environment (RTE), for supporting adaptive learning based on learner objectives [6]. In a general sense, the CAM describe how to build and organize learning content that could be recognized by SCORM compliant LMSs. The CAM describes four components used in a learning experience: (1) Content Model, (2) Content Packaging, (3) Meta-Data, and (4) Sequencing and Navigation. The first component, Content Model, is made up of Assets, SCOs, and Content Organization. Of these, An Asset is the most basic reusable element of SCORM.
An Asset can be a web page, a HyperText Markup Language (HTML) fragment, a flash object, an image file, audio, or an Extensible Markup Language (XML) document. A SCO is composed of several Assets, and forms the basic LO. An example of SCO is shown in Figure 1. The major difference between SCO and Asset is that the SCO is equipped with the functionality of triggering and terminating communication with an LMS. One or more SCOs construct an unit of instructional activities. The Content Organization is a map that represents the structure of instructional activities, indicating how activities relate to one another.

![Image of Sharable Content Object (SCO)](image)

**Figure 1. SCO [6]**

The second component of CAM is Content Packaging, which is a set of specific requirements and guidance to provide a standardized way to exchange learning content between SCORM complaint LMSs. The major component of the Content Packaging is a specific manifest file which describes the content structure and associated resources (e.g. image, media, etc.). On the other hand, the third component of CAM, Meta-Data, describes the physical file of Asset and SCO (see Figure 2).
Finally, Sequencing and Navigation, the fourth component of CAM, defines a set of rules that describe the intended sequence and ordering of instructional activities to be delivered to the learner. However, the Meta-Data and Sequencing and Navigation presented in SCORM are too general to fit certain specific learning situations, for example, the dynamic actions and interactions between learning objects and learners [7].

Cisco presented an RLO (Reusable Learning Object) / RIO (Reusable Information Object) hierarchy to identify different types of learning object [8]. An RLO (like a lesson) is composed of several RIOs (like topics), where each RIO can be categorized into five types (concept, fact, procedure, process, and principle) of learning content, according to the definition of the information [9]. The Cisco RLO can be adapted to different approaches, e.g. receptive, directive, guided discovery and exploratory. The RLO/RIO hierarchy can manage more LOs, but it reduces the interoperability of the learning content. Not all LMSs recognize the various learning attributes (learner's feedback) which are defined by certain content providers for adaptive learning, and simply move on to the next action [10].

This paper presents an Object-Oriented Learning Object approach to increase the interoperability and reusability of learning objects. To accomplish this goal, we applied object-oriented technologies and web service technologies to build the LO. The abstraction in Object-Orientation can be used to reduce the complexity of the LO [11]. Embedded attributes
and member functions in an LO can simplify the LO design process, and inherit all the attributes and member functions from a super-class. An overriding feature allows the LO designer to re-write the member functions no longer required in the subclass. The polymorphism specification allows different actions, with the same name, which can support different strategies. The design of an object-oriented LO using XML technologies can be achieved by using Schema for Object-Oriented XML (SOX), which was submitted to the W3C [12]. The SOX does not support all of the object-oriented specifications and tends to make the LO design more complex. An Object-Oriented XML (o:XML), supports the features of multiple inheritances, function overloading and most other object-oriented features [13]. When compared to Java, however, this object-oriented language is not as good when it comes to distributed computing. Java, known as a powerful object-oriented programming language, was used for the LO design in this paper. The reason for choosing Java was not only that it fully supports object-oriented functionality, but also because it was easy to implement using the Java Server Page (JSP); the Java LO can also be easily referenced in a distributed computing environment. Using web service technologies, we retrieved learning content from remote LMSs without worrying about the structure of the learning object. In a web service oriented LMS, the content consumer first retrieves the remote content, which is then launched to the learner; feedback is then received from the learner, and passed back to the content provider [14].

Web services are web applications consisting of XML, Universal Description Discovery and Integration (UDDI), Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL). UDDI is a specification for maintaining standardized directories of information about web services, and recording their capabilities, location and requirements in a universally recognized format. WSDL applies multiple XML schemas, defining the specific web service structure, such as describing partners' operating procedures. All interfaces that can be used by partners, to understand and to parse transmitted messages, are described in WSDL, including data types, messages, operations, port type, binding and port and service naming. This technology provides the ability to find and bind the content provider, and search contents during the design period. All information about the content provider service is written in WSDL format and published on the Server Register.
The essential components of a distance learning system are learning content and the learning management system. The content includes the learning object and its metadata, while the system must parse and launch the content. In the following section, we present an Object-Oriented Learning Object (OOLE) framework, based on adaptive learning management system architecture and describe both content and system. Next, we describe the application of Java and JSP technologies in designing the Object-Oriented Server Page, which was used to design the Learning Object. We then move on to describe how the OOLE-Based LMS was implemented. Finally, we present our conclusions and our vision for the future of this framework.

2. FRAMEWORK

2.1 Course Architecture

Most learning content has been designed based on the concept of objective. The objective architecture of a course helps associate content and learning experience with the expected learning outcomes, and leads possible learning paths [15]. In order to develop the objective hierarchy for a course, we used top-down objective analysis to refine the objective, described in Figure 1, and established the learning path of these objectives basically according to Reigeluth’s elaboration theory [16], which could be determined by the domain experts. In a case, an objective is too generic or complex, it would be refined into sub-objectives, until the domain experts looked upon one objective was simple enough to be an atomic one. Objectives in our framework were categorized into two types: aggregative objectives and atomic objectives, an atomic objective being one that could not be refined (Objectives 1.1, 1.2, 2.0, 3.1, and 3.2 in Figure 3). Each objective was developed into a LO which contains a set of related knowledge assets and learning strategy information. Those knowledge assets were categorized as concept, fact, procedure, process or principle, and are linked to physical learning assets in a learning content repository (the asset here is like the SCA in a SCORM, in which the SCA can be reused).
Building an Object-Oriented Learning Object in a Distributed Learning Environment

**Figure 3. Building objective refinement & learning**

Assessment assets, located in an assessment repository, were used to evaluate the learning outcomes of the LO for each learner; these outcomes were recorded as the learner’s competency. The default presentation sequencing of assessment assets and learning assets were determined by domain experts. While learning was in progress, the LMS first compared the objective hierarchy with the learner’s personal competency matrix and retrieved the personal deficiency matrix, composed by the LO. Each time an LO was accomplished, the outcome for that LO was placed into the personal knowledge matrix.

### 2.2 Personal Competency Matrix

In order to adaptively deliver learning content based on individual personal background knowledge, the personal competency matrix plays a key role on determining which LOs should be delivered to the learner. A personal competency matrix is regarded as a subset of learning records depicting the levels of testing outcomes of all individual LOs that have been delivered.
to the learner. For instance, in Figure 3, the Objective 3.1 points to a LO containing three Knowledge assets and two Assessment assets. At the first time of learning, the LMS may not have the personal competency matrix of the learner, and thus it delivers the learning content by default sequence. The default sequence for this LO starts form Knowledge asset 1.0, 1.3.1, 1.3.2 to Assessment asset 1.1, to Knowledge asset 1.3 and then to Assessment asset 1.2. The testing outcomes of Assessment asset 1.1 and 1.2 generate the competency level for Objective 3.1. An example of a personal competency matrix is shown in Table 1. The first column represents the capability, and the second column indicates the knowledge level for the capability. The checked symbols in the cell represent the current level of that competency.

Table 1. A sample personal competency matrix

<table>
<thead>
<tr>
<th>Capability</th>
<th>Knowledge Level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>3.0 Be able to use array to implement</td>
<td></td>
</tr>
<tr>
<td>linked list</td>
<td></td>
</tr>
<tr>
<td>3.1 Be able to use the array</td>
<td></td>
</tr>
<tr>
<td>3.2 Be able to understand the linked list</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Knowledge Level: A: Knowledge, B: Comprehension, C: Application, D: Analysis, E: Synthesis

Once the personal competency matrix is built, it can be used to construct the personal deficiency matrix for LO delivery. In Table 2, the Objective 3.1 is the same as Capability 3.1 in Table 1, and can be ignored according to its knowledge level if the required knowledge level of the objective is less than that of the capability. Different aggregate LOs may require different prerequisite knowledge level of its lower LOs to deliver high level LOs. For example, the Objective 3.0 may require the Objective 3.1 to be achieved D (Analysis) level, whilst other Objectives may only require A (Knowledge) level. The expert judgment can be used to assign proper knowledge level at learning content design phase. Thus, the personal deficiency matrix can be built by comparing personal competency matrix with the course objective hierarchy.
Building an Object-Oriented Learning Object in a Distributed Learning Environment

Table 2. A sample objective matrix

<table>
<thead>
<tr>
<th>Learning Objectives</th>
<th>Required Learning Level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>3.0 To be able to use array to implement stack</td>
<td></td>
</tr>
<tr>
<td>3.1 To be able to use the array</td>
<td></td>
</tr>
<tr>
<td>3.2 To be able to understand the stack</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

The big challenge for assessing learning outcomes is how to design assessment for specific learning objective. The assessment can be as simple as selection, or as complex as writing a program. For instance, to evaluate the Knowledge level of Learning Objective 3.2 in Table 2, a couple of selection questions is enough, while a complicated programming assessment (Figure 3) is required to evaluate the Application level of Learning Objective 3.1 in Table 2. Figure 4 shows the case that the leaner is asked to write a simple program to demonstrate his array application capability. The learner can use the [Run] button to test his program, and submit the final program if finished. However, to achieve an automatically judgment according to any answer is not an easy job. In a real case, LMS only can make a decision on totally right answer. When a partial answer is provided by the learner, the LMS will treat as incomplete and wrong answer.

Figure 4. A sample assessment asset of a LO
2.3 Adaptive Learning Content Delivery

The fundamental idea of adaptive learning content delivery is that LMS presents different learning assets to different learners according to their competency. The LOs will be delivered based on the personal deficiency matrix. The adaptive learning content delivery herein are divided into two levels, the course level and the LO level. Look at the example of Figure 2. At the course level, once the Objective 1.0 and 2.0 have been reached, the LMS change the default sequence to LO 3.1 → LO 3.2 → LO 3.0 → LO 0.0. In a case that the learner possesses the Capability 3.1, the LO delivery path can further be reduced to LO 3.2 → LO 3.0 → LO 0.0. Moreover, at the LO level of adaptive learning content delivery, knowledge and assessment assets are delivered based upon the level of learning outcome intended to reach by the learner. Assume that within the LO 3.1 the Assessment asset 1.1 is designed for assessing learning outcome at the Knowledge level, and the Assessment asset 1.2 is designed for assessing learning outcome at the Application level. In a case that the learning outcome set up for LO 3.1 is the Knowledge level, the learning asset presentation sequence will be Knowledge 1.0 → 1.3.1 → 1.3.2 → Assessment 1.1. On the contrast, once the learning outcome is set up to the Application level, the learning asset presentation sequence will turn out to Knowledge 1.3 → Assessment 1.2.

2.4 Learning Object & Class

While HyperText Markup Language (HTML) format is widely used and is compatible with almost all LMSs, its drawback is a lack of programs to control learning activity flow. With Java Server Page technologies, the design of the Java Learning Object Class is similar to the HTML design, but takes advantage of flow control. There are four types of learning content presented in this paper: course, lesson, learning object and learning asset. Figure 4 shows the structure of these four types of learning content. The basis of the learning content, which included the html file, images, video, audio and other types of media, was the Learning Asset (LA). In a SCORM, it can be treated as an SCA (Sharable Content Asset or Asset). This variable scorm, which contained the metadata of the asset or LA, presented the SCORM information model (not shown) of the LA. The second variable was media, which presented the outer resources included in the LA. The media was an html file or an applet, shown to the
learner for the learner's response. The response was then sent back to the LA for further processing. The second level of learning content in this paper is the learning object (LO), composed of several LAs, with a predefined LA sequencing rule. The following code segment illustrates how the LO used the LA as internal learning content.

```java
public class LO01 extends LO {
    SCA sca01 = new SCA("stack01.htm", 1);
    SCA sca02 = new SCA("stack01.htm", 2);
    ...
    Learning(Request req) {
        Switch(get_SCA_ID(current)) {
            Case -1: current = condition_null(req);
            ...
        }
    }
}
```

In this example, the LO01 included sca01 and sca02, which made up the html file. The sequencing of sca01 is coded in condition_01, which analyzed the learner's responses (req), and decided the next SCA. For each SCA, there was a unique condition checking function, which could vary, according to different conditions. During the condition checking, the condition function retrieved the learning record from the Learner class and saved the learning record after the learner had responded. This feature allowed the LO designer to design more than one learning sequence based on the learner's response and learning records.
Figure 5. The class diagram of a learning object

The lesson structure was almost the same as an LO, but was composed of several LOs. The course, comprising lessons, was the top level of the learning object structure. The sequencing of lessons within a course was defined by the condition of each lesson. The outer learning element of this framework was the course, which included many classes of lessons.

2.5 Design the Learning Object

With this design approach, the content designer could reuse the learning elements at different levels. If a learning element fulfilled his requirements, then he USED that element, directly. Otherwise, a class was inherited and any unnecessary operations overridden. The advantage of using an Object-Oriented Learning Object not only increased the reusability of the learning object, but also simplified its design process. Compared to the design process using HTML or XML format, the class approach seems more complex. However, with the Server Page technologies, the learning content designer could design the JSP learning object as easily as an HTML LO. When the LO was requested, the LMS parsed the JSP LO into Java class and it was invoked. To accomplish this goal, the Servlet Engine (The LMS in this
Building an Object-Oriented Learning Object in a Distributed Learning Environment

research was modified from Tomcat 5.1.) had to be modified; this is shown in Figure 5.

![Diagram of Developmental environment of LO]

**Figure 6. Developmental environment of LO**

The use of Java, as a learning object design language, is difficult for most learning content designers. To simplify the design process, a friendly user interface was required. This user interface was able to conceal the complexity of the learning object, and take advantage of its flexible and adaptable qualities. It was possible to design the Server Page LO in JSP format and compile the class while learning was in progress.

### 2.6 Learning Object Storage

After solving the complexity of designing and managing a learning object, using an Object-Oriented approach, another challenge was how to store the learning content. If a Relational Database was used to store the LO, two problems would arise while retrieving learning content: the first problem was that the LO would have to be transformed to a relational schema (this part must still to be done in most LMSs) and vice-versa while retrieving learning content; the second problem was that the learning status of each LO would have to be tracked, making the schema more complex. Using an Object-Oriented Database (OODB), we could store the LO directly to the Database with no need for transformation.

### 3. THE SYSTEM ARCHITECTURE

The importing / exporting of content is the main contribution in learning content exchange. The SCORM run-time environment differs from an online content provider in
distributed computing, where learning content comes from a remote site when required by the local LMS [14][17]. To solve the content exchange problem in a general LMS, we provided web-service oriented LMS architecture, shown in Figure 6. The content provider, who provides the learning content, is registered in the Content Supplier Service Registry. At the course design stage, a designer will search the Registry for content supplier services and place the LO reference in his course. While learning is taking place, the sequencing service of the local LMS will check to see if the content requested by the learner exists on the local LMS. If the content does not exist locally, the request is passed to the Remote SCO Finding Service to search the Content Supplier Service on the internet, and bind to the Supplier Server, to retrieve the remote contents [14]. This architecture was built to run on any SCORM compatible LMS without importing learning content to the local LMS before the learning process can begin.

![Diagram](image-url)

**Figure 7. SCORM Compatible Web Service Oriented LMS**

Another advantage of a distributed learning environment, such as this, is that the remote LMS conceals the complexity of the learning object, and delivers only the SCORM compatible content to the requester. With such a design, the sequencing within each LO can be done at the remote content provider and the local learning content repository. A local LMS can then share its LO with a remote LMS, through the local content supplier service, after registering with the UDDI.

4. CONCLUSIONS
Building an Object-Oriented Learning Object in a Distributed Learning Environment

Traditional learning methods are not easily translated via computer technologies. The major factors influencing distance learning are learning content and the system itself. As one LMS cannot possibly be expected to cover all learning content, the alternative is to REUSE learning content from more than one LMS. While a SCORM offers a way of overcoming the hurdles of content compatibility by facilitating the exchange of standard learning content among LMSs, it is too general to fit specific learning situations. In this paper, we have presented an object-oriented learning object approach, to increase the reusability of learning objects and take advantage of object-oriented programming language, in order to design learning content in a pedagogical way. The designer can use popular language to design complex learning object sequences without having to learn other script language. Object-Oriented Analysis has the advantage of building complex applications in a real world situation; this is similar to processing learning elements, such as objectives, learning objects and knowledge. To solve the problem of content exchange, we provided a web service oriented framework to build the learning management system. This Web Service Oriented Computing provided a way to support LMS content. To accomplish content distribution, many technological applications were required, such as XML, databases and servlets, which facilitated the design, management and supply of learning content. The advantages of the framework provided in this paper are summarized in Table 3.

<table>
<thead>
<tr>
<th>Features</th>
<th>OOLE</th>
<th>SCORM</th>
<th>Traditional Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Learning object can be delivered by SCORM complaint LMS ?</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>2. Support reusable learning content ?</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>3. Support competency based learning ?</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>4. Learning Object Extendibility ?</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>5. Flexibility for design LO ?</td>
<td>High</td>
<td>Median</td>
<td>High</td>
</tr>
<tr>
<td>6. Need extra module plug-in ?</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

The LO of OOLE is the same as the SCO of SCORM, and can be delivered by SCORM complaint LMS, while the learning contents of the traditional LMSs are difficult to export to other LMSs. The OOLE LMS framework provided in this paper can be designed as a
module (Tomcat module, for example), and can be plugged into LMS to support OOLO functionalities. The SCORM complaint LMS still can deliver the OOLO learning content in the default sequence, if lack of the OOLO module.

5. VISION OF THE FUTURE

While Object-Oriented Learning Objects have the advantage of reusability, a challenge is the storing of these learning objects in a Relational Database. To avoid the complex procedures involved in converting the class to a relational table schema, we used an Object-Oriented Database (OODB) to store learning objects; many other information records, however, such as learning records and status, are stored in a Relational Data Base Management System (RDBMS). The feature work of this research was to analyze a Learner Model in an object oriented way and deliver it to an OODB.

Further work is needed to investigate performance and learning content quality in a distributed computing environment. Designing the LO with object references and dynamic binding allows the designer to customize the learning object by deriving it from a base class, or invoking it at a remote site, regardless of its internal structure. To address the problem of internet performance, load balance and fault tolerance should be applied. To cache learning content at a local LMS, when there are large numbers of students, is another way to improve learning performance. Finally, another design issue is LO Quality. With the large increase in numbers of LOs, it is difficult to assess their quality. It is important that, in the future, we are able to easily locate quality LOs.

References:


Building an Object-Oriented Learning Object in a Distributed Learning Environment


