Applications of LOM and XML in Web-Based Distance Learning

Yang Dehua, Tongji University, Shanghai 200092, China
Li Hongying, Tongji University, Shanghai 200092, China
Lin Zhangxi, Texas Tech University, Lubbock, TX 79409, USA

Abstract

In this paper, a set of distance learning system models based on revised OMT and UML are presented. They are designed to describe the systems functionality, static data structures, and user interfaces for NCEC system, a EU sponsored distance learning project. The development technologies used in the distance learning systems are also investigated comprehensively in accordance with recent advances of LOM and the second generation web publishing technologies.

1. INTRODUCTION

Web-based distance learning has become one of the major areas in electronic commerce world. Recent technology developments in this field, including LOM (Learning Object Metadata) and XML (eXtensible Markup Language), facilitate the applications of learner-customizable or individualizable distance learning systems, which are highly adaptable to users' specific requirements [1]. NCEC (Network-training Collaboration in Europe and China) is a joint project between Chinese and European universities and institutions, aiming at producing an Internet-based interactive learning system, which is highly individualizable in order to satisfy diversified needs of users with reusable learning components namely learning objects (LO) paradigm [2]. NCEC system can provide broad online services to facilitate individual learning and group collaboration, and facilitate user's self-planning, self-assessment and self-regulating of learning materials and learning processes .It conforms to LOM, XML, CORBA and other international standards including DOM (Document Object Model) level 1 and 2 as well as SAX (Simple API for XML) when XML and HTML documents are processed. NCEC project applies mainly server side technologies like Java servlets, JSP and XSP (eXtensible Server Pages), and uses revised OMT and UML [3] methodologies in system analysis and design. In the systems implementation and programming, Java technologies including RMI, JDBC, EJB and JMS are used. It also has necessary components to support system deployment and exploitation.

This paper introduces some models including system static class model and functional model, and technologies used in NCEC project, with emphasis mainly on the applications of XML/XSL, XSP and LOM.

2. SYSTEM FUNCTIONAL MODEL

2.1. System Functional Structure

NCEC system consists of seven subsystems:

- LO authoring system
- User accessing system
- User management system
- Learning administration system
- LM composing system
- LO search engine system
- Repository accessing system

These subsystems function together to achieve the common objectives: LM (Learning Material) courseware production and exploitation. The system functional structure is showed in Fig.2.1 (simplified).

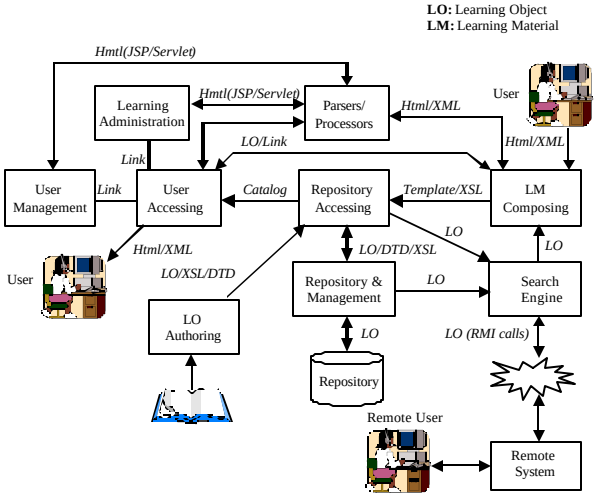

Fig 2.1 System functional model

2.2. Global Functional Descriptions

The following is a brief operational description of the system.

2.2.1. LM/Courseware Production

Online courseware authoring is a four-step production process including textbook parsing, presentation style

defining, XML validating, and courseware temple designing.

Textbook Parsing

Tokenizing textbooks includes the following tasks:

- Defining the tags for textbook parsing as well as the DTDs for the validation of these tags. The tags can be defined either on the basis of a discipline or in accordance with different kinds of courses, i.e. each course may have its own set of tags. It is preferable that each discipline is associated to a set of DTDs in a single file. So, all LOs for the courses within a discipline can be validated by the same DTD file.
- Dividing a textbook into small and reusable parts called Learning Objects.
- Generating the metadata and their values for each LO according to LOM standard.
- Setting taxonomies and dictionaries of LOM metadata, which describe attributes of the LOM elements, and can be understood by search engines specified to deal with LOs.

The semantic relations of LOs are defined by means of their structures, which are the basis upon which LO hierarchical structures can be established.

Presentation Style Defining

The presentation style for a LM is defined by the LO-related style sheets in XSL files. A LM is allowed to have multiple XSL files in order to provide different presentation styles on user browsers. For example, the LM may apply one presentation with multimedia support and the other with only textual display. Conversely, it is also applicable to use a special XSL for LMs from several books to maintain its original presentation style.

XML Validating

All LOs that are in the form of XML must be validated against the related DTDs. Errors must be corrected before the LOs are stored into system repository.

Courseware Template Designing

Courseware templates are XML files used to describe the structure of either the textbooks having been parsed or LMs to be composed by learners. Each textbook has at least one static template for restoring its original structure. In creating a customized book, a learner uses a courseware template to describe a desired LM from standard LM structures.

2.2.2. Courseware Exploitation

Courseware exploitation can be classified into two categories: learning materials delivery and user learning management.

Learning Materials Delivery

Learning materials delivery functions are supported by four subsystems: User Accessing, LM Composing , LO Search Engine , and Repository Accessing.

- User Accessing

User Accessing system provides user interfaces for accessing NCEC services, which are implemented mainly by server-side programming technologies such as JSP, and XML processors including XSP. The system interfaces are selection-driven, namely dialogs between users and system are conducted by selecting menus, lists, buttons and various links. The user interface components and their attributes are generated dynamically with names and values coming from repository catalog, which consists of course names, book titles, book/LM structures, and related LOMs.

The user requirements are extracted from user interface components. The requirements are in XML forms that mainly consist of LM headers, which are structured contents with various LO attributes expressed by LOM metadata. The requirements are then sent to LM Composing Subsystem for further processing. The Composing system will send the requested LM contents with necessary LO links in HTML forms to a user, after the LM has been composed and processed, incorporated with certain XSL files.

- LM Composing

When a request in the form of XML comes from User Accessing system, LM Composing system will search (Using LO Search Engine) the cache first for the LM. If the needed documents are found in the cache, the LM contents will be sent to the user browser. Otherwise, a courseware template in the repository will be used. In this case the LOs positioning in the template will be replaced by program links indicating LOs addresses in the repository, where LO Search Engine system searches necessary LOs and pass their addresses to LM Composing system. Based on the metadata contained in requirements, LM Composing system will generate a LM header that is the contents of the user requested LM with necessary hyperlinks. The generated LM header is then parsed into HTML documents and sent to the user browser.

- LO Search Engine

This subsystem is implemented using Java RMI protocol, a much simpler technology than IDL server which is a full-blown CORBA implementation. The primary idea behind the scheme is that using the same search engine system to search several sites can be done both locally and remotely with good efficiency.

The search engine is used specifically for the retrieval of LOs in courseware repositories as well as XML documents described by LOM. In retrieving LOs, local repositories and remote sites can be searched in order, by using Repository Accessing system and calling functions of remote instances in RMI servers.

- Repository Accessing

This system provides interfaces for accessing system database tables, caches and directories as well as files. It consists of modules used to maintain courseware repositories, especially the databases and files. The repository accessing interfaces encapsulate all database tables, whose inheritances are realized by using DTAL (Data Table Abstraction Layer) [4].

User Learning Management

User Learning Management consists of two subsystems: User Management and Learning Administration.

- User Management

This system provides functions such as user registration, user logon, user authentication, account services, etc. When a user wants to browse the protected pages, he/she must login to be authenticated by this system. Charges for the online browsing will be accounted duly, and relative reports can be generated upon requests, which the user can browse and print.

- Learning Administration

A registered user can get many services from the system, such as searching learning topics, taking examinations, and making a curriculum with the incorporation of teachers working with the system. It is very important for a user to have various kinds of online services like making notes on browsing pages, getting help from teachers, joining mailing lists and BBS, retrieving the interfaces used last time by clicking only a button. He can also get academic advices from experienced experts dedicated to the system.

3. DEVELOPMENT TECHNOLOGIES

3.1. XML

XML is well suited for the interchange of data, since XML documents are structured, self-describing, easily parsed and can represent complex data. In NCEC project, all learning objects and their structures in the systems are formulated by XML. A LO is a self-contained component that can be organized together to compose Learning Materials. The following is an example of Learning Material structure in XML form:

<course name=”java programming”>
<chapter name=”introduction”>
<section name=”section 1”>
<LO name=”Prerequisites”>
<author> Bruce Eckel</author>
This book assumes that you have some programming familiarity. You understand that a program is a collection of statements, the idea of a subroutine/function/macro, control statements such as “if” and looping constructs such as “while,” etc. However, you might have learned this in many places, such as programming with a macro language or working with a tool like Perl. As long as you’ve programmed to the point where you feel comfortable with the basic ideas of programming, you’ll be able to work through this book. Of course, the book will be easier for the C programmers and more so for the C++ programmers, but don’t count yourself out if you’re not experienced with those languages (but come willing to work hard; also, the multimedia CD that accompanies this book will bring you up to speed on the basic C syntax necessary to learn Java)…
</LO>…</section>…</chapter>…</course>

3.2. XSLT

A XSL Transformations (XSLT) specification defines an XML-based language for expressing transformation rules that map one XML document to other formats. XSLT has many structures and elements that can be found in traditional programming languages, including variables, functions, iteration, and conditional statements. In this system, we use XSLT to present LMs on users' browsers. The following is an example of LM presentation:

```
<?xml version="1.0" encoding="gb2312"?>
<!-- edited with XML Spy v3.0 NT
(http://www.xmlspy.com) by hongyingli (tongji) -->
<xsl:stylesheet  version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
        <xsl:template match="course">
                <html>
                        <head>
                                <title>
                  <xsl:value-of select="@name"/>
                  </title>
                        </head>
                        <body>
                <xsl:apply-templates
select="section"/>
                        </body>
                </html>
        </xsl:template>

        <xsl:template match="section">
    <xsl:value-of select="@name"/>
                <xsl:apply-templates/>
        </xsl:template>
        …
</xsl:stylesheet>
```

3.3. XSP

XSP is one of Apache Cocoon's technology for building web applications based on dynamic XML content, which can be generated by Java programs embedded in XML documents.

Beyond static content (i. e., hand-written documents produced by web authors), web applications demand dynamic content generation capabilities, where XML documents are programmatically produced at request time. The following XSP program show how to use XSP to get a LOContent whose LOCode equals 2 from database.

```
<?xml version="1.0" encoding="gb2312"?>
<!-- edited with XML Spy v3.0 NT
(http://www.xmlspy.com) by ncec (ncec)-->
<?cocoon-process type="xsp"?>
<?cocoon-process type="xslt"?>
<?xml-stylesheet version="1.0" type="text/xsl"
href="newlo.xsl"?>
<xsp:page language="java"
xmlns:xsp="http://www.apache.org/1999/XSP/Core">
        <xsp:structure>
                <xsp:include>
java.sql.*;
</xsp:include>
        </xsp:structure>
        <xsp:logic>
public String getLO(){
String userID="db2admin";
String userPass="12345678";
String url="jdbc:db2//202.120.182.102/LO";
Connection con=null;
ResultSet rs=null;

try{
Class.forName("COM.ibm.db2.jdbc.net.DB2Driver");
}catch(Exception e)
{
System.out.println("get Driver error");
System.out.println(e.getMessage());
}
try{
con=DriverManager.getConnection(url,userID,userPa
ss);
Statement stmt=con.createStatement();
rs=stmt.executeQuery("SELECT CONTENT from LO
where LOCode='2'");
}catch(Exception e)
{
System.out.println(e.getMessage());
}
try{
while(rs.next())
{
s=rs.getString(1);
return s;
}
}
catch(SQLException sqle1)
{
System.out.println(sqle1.getMessage());
}
return s;
}
</xsp:logic>
        <page>
                <LOContent>
                        <xsp:expr>
                        getLO()
                        </xsp:expr>
                </LOContent>
   …</page>
</xsp:page>
```

4. SYSTEM STATIC CLASS MODEL AND SYSTEM FRAMEWORK

The system static class model of NCEC project and its framework are shown in Fig 4.1 (simplified) and Fig.4.2. The detailed descriptions of these models are omitted due to the limited paper space.

The system class model depicts the system static classes and their relations, which use UML methodology. The instances instantiated from these classes a re thought to be persistent objects, which are actually stored in the repositories.

System framework depicts the components both hardware and software involved and their relations between them.

5. CONCLUSIONS AND REMARKS

Adopting the latest technologies for web-based applications including LOM, XML and related technologies has greatly enhanced the features of NCEC system in flexibility, interactivity and customizability, which are the future of the Internet based distance learning. The practice in NCEC system development allows us to explore the application of these technologies.

In the system analysis and design, UML and OMT are good methodologies for system model establishment. To reduce their complexity and overcome other shortcomings, one could tailor these models in accordance with specific system requirements.

REFERENCES

[1]. Yang, Dehua, *The Models and Technologies of Internet Based Distance Learning*, Computer Application Research, August, 2000.

[2] Lin, Zhangxi, et al, Network-Training Collaboration in Europe and China (NCEC) – Issues and Promises, Proceedings of AMCIS, August, 2000.

[3] *OMG Unified Modeling Language Specification, version 1.3,* June 1999.

[4]. Yang, Dehua, Establishment of Three-tier Architecture Internet Applications, Proceedings of the 3th NJU/OIT/TU Symposium on Management and Management Information System, September, 1999.

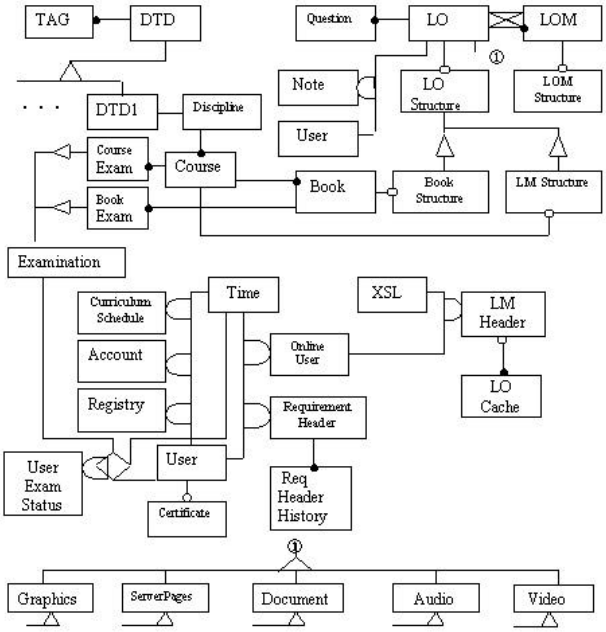

Fig 4.1 System Static Class Model

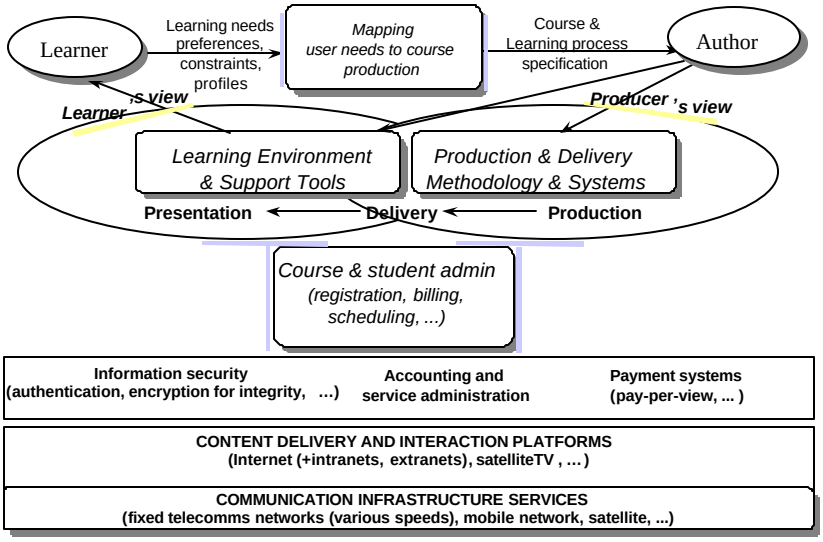

Fig 4.2 System Framework

Learning Object Metadata is a data model, usually encoded in XML, used to describe a learning object and similar digital resources used to support learning. The purpose of learning object metadata is to support the reusability of learning objects, to aid discoverability, and to facilitate their interoperability, usually in the context of online learning management systems (LMS).Â For UK Further and Higher Education, the most relevant family of application profiles are those based around the UK LOM Core.[2] The UK LOM Core is currently a draft schema researched by a community of practitioners to identify common UK practice in learning object content, by comparing 12 metadata schemas. Prospects of Implementing E-Learning Systems based on Learning Objects and XML in Nigeriaâ€™s Educational Sector. - Free download as PDF File (.pdf), Text File (.txt) or read online for free.Â Distance learning he says is one of these alternatives which became attractive where students and instructors are physically in different locations and time. By using distance learning tools, education can be more flexible with respect to place and time constraints.Â The Web has therefore been established as a major platform for applications in learning. According to Friesen, N. (2009), e-learning is an educational system for providing learning through electronic technologies especially the Internet. The machine learning techniques are not limited to offline applications and analyses, and they can be the predictive engine of your web services. For example, popular and useful applications of machine learning models in web applications include. Spam detection in submission forms, Search engines, Recommendation systems for media or shopping portals, and many more. Here is good examples that show you how to embed a machine learning model into a web application that can not only classify but also learn from data in real-time. Build and Deploy a Predictive Web App Using Python and Azure ML. 9. E