

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

Learning Object Metadata Interchange Mechanism

A thesis presented in partial fulfillment of
the requirements for the degree of
Master of Information Science
at Massey University, Palmerston North, New Zealand.

Yuejun Zhang

2005

Acknowledgements

I would like to thank all my friends who provided various helps in completion of this thesis.

Special thanks are given to Associate Professor Kinshuk, my supervisor, for all his help throughout this project. Without his patient guidance and valuable advice, it would be impossible to complete this thesis.

Abstract

In spite of the current lack of conceptual clarity in the multiple definitions and uses, the term *learning objects* is still frequently used in content creation and aggregation in the online-learning field. In the mean time, considerable efforts have been initiated in the past few years for the standardization of metadata elements for consistent description of learning objects, so that learning objects can be identified, searched and retrieved effectively and efficiently across multiple contexts. However, there are currently a large number of standardization bodies and an even much larger number of ongoing standard initiatives in the learning field, and different learning objects repositories are likely to apply different metadata schemas to meet the specific needs of their intended communities. An interchange mechanism for the conversion between various metadata schemas, therefore, becomes necessary for intensive interoperability.

In this thesis, we first make a brief introduction to the concept *learning objects*, then the term *metadata*, followed by a description of the functional requirements of learning objects, the purposes of metadata, and the importance of metadata for learning objects. After that, this thesis investigates metadata schemas in various fields in general, focused on several mainstream metadata specifications developed for learning objects in particular. The differences among these metadata schemas for learning objects are analyzed and a mapping between their elements is identified. On the basis of literature review, a framework for interchange of metadata schemas is proposed and a prototype to demonstrate the functionalities of the framework is developed. For the high scalability and the high accuracy of the developed system, a so-called LOM-intermediated approach is suggested, and a so-called dynamic-database methodology is adopted. The LOM-intermediated approach significantly simplifies the metadata mapping issues by undertaking the schema-schema mapping in a way of schema-LOM-schema mapping, while the dynamic-database methodology effectively prevents any data-loss resulting as a by-product from the use of LOM-intermediated approach. The prototype currently generates and outputs XML metadata in IMS, EdNA, Dublin Core and LOM. It is a web-based three-tier architecture, using Java technologies for implementation, MySQL as the database server and JDBC for database access.

TABLE OF CONTENTS

CHAPTER 1 BACKGROUND AND PROJECT OUTLINE	1
1.1 Background	1
1.2 Project Outline	2
1.2.1 Scalable, General-purpose System	2
1.2.2 LOM-intermediated Approach	3
1.2.3 Dynamic-database Methodology	3
1.2.4 System Functionalities	4
1.2.5 System Architecture and Technologies	5
1.3 Outline of This Thesis	5
CHAPTER 2 LEARNING OBJECTS AND METADATA	7
2.1 Learning Objects	7
2.1.1 Definition of Learning Objects	7
2.1.2 Capabilities of Learning Objects	10
2.1.3 Pros and Cons of Learning Object Approach	11
2.1.4 Types of Learning Objects	12
2.1.5 Other Issues about Learning Objects	15
2.2 Learning Object Metadata	16
2.2.1 What Is Metadata?	18
2.2.2 Problems of Normal Search Technologies	21
2.2.3 Advantages and Disadvantages of Metadata	22
2.2.4 Categorization of Metadata	23
2.2.5 Who Creates Metadata?	25
2.3 Learning Objects Repositories	25
2.3.1 Introduction to Learning Objects Repository	25
2.3.2 Examples of Learning Object Repositories	27
2.4 Summary	27
CHAPTER 3 METADATA SCHEMAS	29
3.1 Metadata Standardization	29
3.2 Metadata Schemas Overview	31
3.2.1 Web Community and Dublin Core	32
3.2.2 Metadata Standardization in Government Sector	35
3.2.3 Libraries and MARC	36
3.2.4 Archives and ISAD	36
3.2.5 Publishing Industry and ONIX	37
3.2.6 Multimedia Metadata Standards	39

3.3 Metadata Schemas for Learning Objects	39
3.3.1 General Introduction	39
3.3.2 LOM	40
3.3.3 IMS and ADL SCORM	44
3.3.4 EdNA Metadata Standard	49
3.4 Summary	50
CHAPTER 4 METADATA INTERCHANGE AND PROTOTYPE DESIGN	52
4.1 The Issue of Metadata Interchange	52
4.2 Crosswalks	53
4.2.1 Definition and Issues	53
4.2.2 Examples of Crosswalks	54
4.2.3 Our Mapping Work	55
4.3 Current Work on Metadata Interchange	56
4.4 A Metadata Interchange Framework and Prototype Design	60
4.4.1 Purpose and General Requirements	60
4.4.2 The Proposed Framework	61
4.4.3 Prototype Functionalities	66
4.4.4 Prototype Architecture	68
4.4.5 Overall Processing Flow	69
4.4.6 The Database Design	71
4.4.7 Mappings between the Database and Metadata Schemas	75
4.4.8 XML Bindings and XML Schema Files	76
4.5 Summary	84
CHAPTER 5 PROTOTYPE IMPLEMENTATION AND EVALUATION	86
5.1 Implementation Technologies	86
5.1.1 Introduction	86
5.1.2 Java Technology	86
5.1.3 XML Technology	88
5.1.4 The JAXP API	91
5.1.5 The JDBC API	94
5.1.6 Java Servlets, JSPs, and JavaBeans	97
5.2 Main Java Components in the Application	102
5.3 Description of Java Components	107
5.3.1 The RecordBean and EdnaRecordBean Classes	107
5.3.2 The OpRecordBean and OpEdnaRecordBean Classes	109
5.3.3 The IMSCreator, EdNACreator, DCCreator, and LOMCreator Classes	109
5.3.4 The FileUploadBean Class	111
5.3.5 The FileRecordBean Class	112
5.3.6 The WebRecordBean Class	113

5.4 The Web Application Deployment	114
5.4.1 The Tomcat Server	114
5.4.2 Directory Structure of the Application	116
5.5 System Evaluation	117
5.5.1 Screen Interfaces	117
5.5.2 Examples of Interchange	121
5.6 Summary	126
CHAPTER 6 CONCLUSION AND FUTURE WORK	128
6.1 Conclusion	128
6.2 Future Work	129
REFERENCES	133
APPENDIX A METADATA ELEMENTS MAPPING	140
A-1 Mapping Between Dublin Core and IEEE LOM	140
A-2 Mapping between ARIADNE, LOM and Dublin Core	141
A-3 Mapping between LOM, GEM and EdNA	143
APPENDIX B XML SCHEMA FILES	146
B-1 Schema dc.xsd	146
B-2 Schema edna.xsd	147
APPENDIX C ACRONYMS	149

LIST OF FIGURES

CHAPTER 2 LEARNING OBJECTS AND METADATA

Figure 2.1 Terminology for learning objects	9
Figure 2.2 Anatomy of a learning object	17
Figure 2.3 The relationship between learning objects, metadata and LCMS	17
Figure 2.4 Metadata in learning object repositories	18
Figure 2.5 An example of document metadata	20
Figure 2.6 The example metadata in Dublin Core	20

CHAPTER 3 METADATA SCHEMAS

Figure 3.1 Specifications, application profiles and standards	31
Figure 3.2 LOM development process	41
Figure 3.3 LOM element set	44
Figure 3.4 Element by element comparison for metadata schemas	46
Figure 3.5 ADL SCORM	48

CHAPTER 4 METADATA INTERCHANGE AND PROTOTYPE DESIGN

Figure 4.1 Architecture of a metadata mapping system	57
Figure 4.2 A metadata schema translation service	58
Figure 4.3 The proposed framework for metadata interchange	61
Figure 4.4 The LOM-intermediated approach vs. the direct mapping	62
Figure 4.5 Functional compositions of the prototype	66
Figure 4.6 Architecture of the prototype	68
Figure 4.7 The overall flowchart of the processing procedures	70
Figure 4.8 Correlations between the data tables	75
Figure 4.9 The XML Schema file imsmd_rootv1p2p2.xsd	78
Figure 4.10 The XML Schema file lom.xsd	79
Figure 4.11 The XML Schema file dc.xsd	81
Figure 4.12 An example of EdNA record in XML	82
Figure 4.13 The XML Schema file edna.xsd	83

CHAPTER 5 PROTOTYPE IMPLEMENTATION AND EVALUATION

Figure 5.1 J2EE Multi-tier architecture	87
Figure 5.2 DOM Implementation via JAXP	93
Figure 5.3 Example Java code for parsing XML using DOM via JAXP API	93
Figure 5.4 Example Java code for creating XML using DOM via JAXP API	93
Figure 5.5 The JDBC library structure	94
Figure 5.6 Example Java code for implementing JDBC to access the database	96
Figure 5.7 Example Java code from the servlet class IMSCreator	99
Figure 5.8 Example code from the JSP page EditRecord.jsp	100
Figure 5.9 A JavaBean example – RecordIndexBean	102
Figure 5.10 Collaborations between the main Java components	106
Figure 5.11 Class diagram for the Java bean classes in the beans package	107
Figure 5.12 The UML view of the RecordBean class	108

Figure 5.13 The UML view of the OpRecordBean class	110
Figure 5.14 The UML view of the IMSCreator class	111
Figure 5.15 The UML view of the FileUploadBean class	112
Figure 5.16 The UML view of the FileRecordBean class	113
Figure 5.17 The UML view of the WebRecordBean class	115
Figure 5.18 Information flow between the client, the Web server, and the Tomcat server	116
Figure 5.19 Structure of the Web application for our project	116
Figure 5.20 User login, registration and the main menu	117
Figure 5.21 Record input form for various metadata schemas	118
Figure 5.22 The Main menu and the edit form when uploading XML	119
Figure 5.23 Create metadata from Web source	120
Figure 5.24 Edit an existing record	121
Figure 5.25 XML format of the metadata record in various schemas	124
Figure 5.26 Transform a record from one schema to another schema	126

CHAPTER 6 CONCLUSION AND FUTURE WORK

Figure 6.1 Response with different schemas automatically for different clients	129
Figure 6.2 Demo of a flexible DC editor	131

LIST OF TABLES

CHAPTER 2 LEARNING OBJECTS AND METADATA

Table 2.1 Learning objects functional requirements	11
Table 2.2 Pros and cons of learning object approach	12
Table 2.3 Types of learning objects	13
Table 2.4 Types of learning objects	14
Table 2.5 Examples of learning objects repositories with features and characteristics	28

CHAPTER 3 METADATA SCHEMAS

Table 3.1 The Dublin Core Metadata Element Set	33
Table 3.2 ONIX product groups	38
Table 3.3 EdNA Metadata Standard element set	50

CHAPTER 4 METADATA INTERCHANGE AND PROTOTYPE DESIGN

Table 4.1 EdNA-LOM Mapping	55
Table 4.2 Tables and data fields in the basic database	72
Table 4.3 Dynamic table and the data fields for EdNA	74
Table 4.4 Mapping between the database and the EdNA Metadata Standard	76

CHAPTER 5 PROTOTYPE IMPLEMENTATION AND EVALUATION

Table 5.1 Core J2EE packages	88
Table 5.2 Java packages for XML processing	92
Table 5.3 Core JDBC classes	95
Table 5.4 The main Java components implemented in the system	103
Table 5.5 Extracted HTML data and the corresponding database fields	114
Table 5.6 An example metadata record	122