

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.

Web-course Search Engine

A thesis presented in partial fulfillment of

the requirements for the degree of

Master of Science in Computer Science

at Massey University, Palmerston North, New Zealand.

Xiaohua Gong

2002

Acknowledgements

Thanks to my supervisor Kinshuk for his patience, ideas, faith and enthusiasm. His support made the year a lot easier to bear.

Abstract

The World Wide Web is an amazing place that people's lives more and more rely on. Especially, for the young generation, they spend a significant amount of their play and study time using the Internet. Many tools have been developed to help the educational users in finding educational resources. These tools include various search engines, Web directories and educational domain gateways. Nevertheless, these systems have many weaknesses that made them unsuitable for the specific search needs of the learners.

The research presented in this thesis describes the development of the *Web-course search engine*, which is a friendly, efficient and accurate helper for the learners to get what they want in the vast Internet ocean. The most attractive feature of this system is that the system uses one universal language, which lets the searchers and the resources "communicate" with each other. Then the learner searchers can find the Web-based educational resources that are most fit to their needs and course providers can provide all necessary information about their courseware. This universal language is one widely acceptable Metadata standard. Following the Metadata standard, the system collects exact information about educational resources, provides adequate search parameters for search and returns evaluative results. By using the *Web-course search engine*, the learners and the other educational users are able to find useful, valuable and related educational resources more effectively and efficiently.

Some improvement suggestions of the search mechanism in the World Wide Web have been brought forward for the future research as a result of this project.

Table of Contents

Chapter 1 Outline.....	1
1.1 Problem Analysis.....	1
1.2 Solving Problems.....	2
1.3 Implementation.....	3
1.4 Summary.....	3
Chapter 2 Standard and Metadata.....	5
2.1 Introduction.....	5
2.2 Why standard?.....	5
2.2.1 Examples of standards.....	6
2.2.2 Benefits of using a standard in a Web search system.....	7
2.3 Metadata standards.....	8
2.4 Examples of Metadata standards.....	11
2.4.1 Learning Object Metadata (LOM).....	11
2.4.2 Dublin Core.....	15
2.4.3 EdNA Metadata.....	17
2.4.4 Gateway to Educational Materials (GEM) Metadata Standard.....	20
2.5 Comparison of various metadata standards.....	22
2.6 Conclusion.....	26
Chapter 3 Survey of various Web-based search systems.....	28
3.1 Introduction.....	28
3.2 Systems dedicated to educational area.....	28
3.2.1 Education World.....	28
3.2.2 GEM (The Gateway to Educational Materials).....	32
3.2.3 EdNA Online.....	38
3.3 Common search engines or Web directories used for both educational and other purposes.....	41
3.3.1 YAHOO.....	42
3.3.2 Northern Light.....	44
3.3.3 HotBot.....	45
3.3.4 Lii.org (Librarians' index to the Internet).....	47
3.4 Discussion on common search engines.....	50
3.5 Conclusion.....	50

Chapter 4 Using Metadata Standard to construct “Web-course Search engine”52

4.1 Introduction.....52

4.2 Functionality of the “Web-course Search engine”.....52

4.3 Metadata Standard of the “Web-course Search engine”.....53

 4.3.1 Summary of the “Metadata Standard”.....54

 4.3.2 Mapping LOM.....62

 4.3.3 Characteristic of the “Metadata Standard”.....64

4.4 Information Collection, Search criterion and search result.....67

 4.4.1 Information Collection.....67

 4.4.2 Search criterion.....73

 4.4.3 Search result.....76

4.5 Database structure.....79

 4.5.1 Database structure underlying search mechanism.....79

 4.5.2 Definition.....81

 4.5.3 Referential constraints.....84

 4.5.4 Relationships between tables.....85

4.6 Conclusion.....85

Chapter 5 Implementation of The Web-course search engine.....88

5.1 3-tiered Architectures.....88

5.2 Functionality Specification.....89

 5.2.1 Scenarios.....89

 5.2.2 Functional Specification.....93

5.3 Structure.....94

 5.3.1 Model-View-Controller design pattern (MVC).....94

 5.3.2 EJB-centric and Web-centric design.....95

5.4 The Model of Model-View-Controller.....96

 5.4.1 State in the Web-course search engine.....97

 5.4.2 Persistent Data.....99

5.5 The View of Model-View-Controller.....100

 5.5.1 Interaction Interface.....100

 5.5.2 Screen’s structure.....103

 5.5.3 Model-View Synchronization.....104

5.6 The Controller of Model-View-Controller.....106

 5.6.1 EJB-centric controller.....106

 5.6.2 Web-centric controller.....110

5.7 Deployment.....112

 5.7.1 Deployment Descriptors.....112

5.7.2 Packaging the Application.....	112
5.8 Transactions & Security.....	114
5.8.1 Transactions.....	114
5.8.2 Security.....	115
5.9 Summary.....	120
Chapter 6 Future of The Web-course search engine.....	121
6.1 Searcher Membership Module.....	121
6.2 Metadata Editor.....	121
6.3 Cooperation With Different Search Systems.....	122
6.4 Summary.....	124
References.....	125
Appendix A: Screenshots of the system.....	130
Appendix B: Outline of LOM Metadata Elements.....	135
Appendix C: Dublin Core Metadata Elements.....	138

List of Figures

Chapter 1 Outline

Figure 1.1 Architecture of the system.....	4
--	---

Chapter 2 Standard and Metadata

Figure 2.1 Nutrition information.....	6
Figure 2.2 Hierarchy/tree structure of LOM.....	13
Figure 2.3 Relationships of EdNA/DC and GEM/DC.....	22
Figure 2.4 Inheritance Relationship (e.g. the LOM inherits from Dublin Core).....	24
Figure 2.5 Relationships of Dublin Core, GEM, LOM and EdNA.....	26

Chapter 3 Survey of various Web-based search systems

Figure 3.1 "ADD A SITE" page.....	29
Figure 3.2 "Lesson Plan Submit" page.....	30
Figure 3.3 Search results.....	32
Figure 3.4 GemCat.....	35
Figure 3.5 GEM Search.....	35
Figure 3.6 GEM search result.....	36-37
Figure 3.7 EdNA search results.....	40
Figure 3.8 YAHOO Search Options.....	42
Figure 3.9 YAHOO search results.....	43
Figure 3.10 Northernlight.com Power Search.....	44
Figure 3.11 Northern Light search results.....	45
Figure 3.12 Hotbot advanced search.....	46
Figure 3.13 Hotbot search results.....	46-47
Figure 3.14 lii.org-advanced searches.....	49
Figure 3.15 lii.org search results.....	49-50

Chapter 4 Using Metadata Standard to construct "Web-course Search engine"

Figure 4.1 Relationships of Dublin Core, LOM, and "New Metadata Standard".....	67
Figure 4.2 Information-sheet.....	67-72
Figure 4.3 Search request.....	74-76
Figure 4.4 Search result.....	77-79
Figure 4.5 Relationships between tables.....	87

Chapter 5 Implementation of The Web-course search engine

Figure 5.1 A 3-Tier Architecture.....	89
Figure 5.2 Mixed EJB-centric and Web-centric design.....	96
Figure 5.3 Fine-grained Objects.....	99
Figure 5.4 A Template.....	104
Figure 5.5 Model-View Synchronization.....	105
Figure 5.6 CatalogWebImpl accesses Data.....	106
Figure 5.7 EJB-centric controlling process.....	110

Figure 5.8 Web-centric controlling process.....111
Figure 5.9 The enterprise beans protected by Web components.....118

Chapter 6 Future of The Web-course search engine

Figure 6.1 Cooperation between systems.....122
Figure 6.2 Insides a search system.....123

List of Tables

Chapter 2 Standard and Metadata

Table 2.1 EdNA Metadata Elements.....	18-19
Table 2.2 GEM Metadata Elements.....	20-21
Table 2.3 Dublin Core and LOM relation.....	22-23
Table 2.4 GEM mapping LOM.....	24-25
Table 2.5 EdNA mapping LOM.....	25

Chapter 3 Survey of various Web-based search systems

Table 3.1 Metadata elements compare.....	30
--	----

Chapter 4 Using Metadata Standard to construct “Web-course

Search engine”

Table 4.1 Definition of the “Metadata Standard”	55-61
Table 4.2 LOM Mapping.....	62-64
Table 4.3 Some Elements of “Metadata Standard”.....	65-66
Table 4.4 Definitions of tables.....	81-84

Chapter 1 Outline

“The World Wide Web facilitates co-operative teaching and learning, offering an exciting potential for sharing both the learning resources as well as the tasks of creating new learning resources through *international collaboration*.” (Kinshuk., Patel, A., & Russell, D. (1999)). The methods by which learners find information on the Web are tools like search engines, Web directories and metasearch engines. (Loren, Will & Brian. (1999)). Information banks (Perkins. (1991)), search engines/systems are the indispensable component of this *international collaboration*. The learners and the other educational users use these search systems to index, search, and find teaching and learning resources. This thesis describes the *Web-course search engine*, developed in this project, which is particularly focused on indexing, searching and finding educational resources. The system attempts to eliminate various deficiencies of commonly available search engines.

1.1 Problem Analysis

There are many search tools/systems for resources available on the World Wide Web. However, currently, these tools/systems cannot help users to find Web-Based educational resources effectively and efficiently. Eric, Steve, William & C.Lee. (1999) pointed out that metasearch engines, which rely on other search tools, can considerably increase coverage, but they are still limited by the engines they use with respect to the number and quality of results.

If a student wants to find some educational material for assignment or research on the Web, for example, for studying English, he/she will use some sort of keywords like “study English” in a search engine, such as Yahoo, Excite, or Google. The search engine will then display a number of the Web addresses about this topic. Not all of these links will be of his/her interest because the student may have certain requirements such as a high level course, some kind of multimedia inter-action course and Mac Platform only, cost free, using the Cambridge English, and so on. How can the student submit such a search request using currently available searching engines?

Literature suggests that it is hard to get accurate results from these search engines. There are various reasons for this phenomenon.

One reason is that these search engines just search the information inside the HTML document and HTML is not flexible enough to express and contain information about the content. It only has few meta tags for this purpose.

Another important reason is that there is no established standard or rule to describe the special material such as study English course. Because there is no established standard, courses providers are not able to provide uniform information to search engines, and search engines could not provide a standard request sheet for users to submit queries. In other words, there is not good communication between query user, search engine and courses' providers. Metadata is information about an object. As the number of educational objects grows exponentially and the needs for learning expand equally noticeably, the lack of information or metadata about educational objects places a serious and fundamental constraint on ability to discover, manage and use objects.

The chapter 2 will introduce the Metadata concept, explain the benefits of using Metadata on a Web search system, and then compare several Metadata Standards that are commonly accepted. The chapters 3 will analyze in-depth the problems existed in some commonly available search system in the WWW by discussing a survey of various Web-based search systems, and grouping them to systems that are dedicated to educational area, and common search engines or Web directories used for both educational and other purposes.

1.2 Solving Problems

For constructing an effective and efficient search system, the *Web-course search engine*, some problems have to be solved. First is to find a Metadata standard, which is suitable for describing the Web-Based educational course material. Second is to collect the information about Web-based courses following this Metadata standard. Final is to provide the request sheet for the searcher to submit query and search

results/response sheet, which can clearly and exactly express the information about Web-based course.

The chapter 4 will describe the functionality of the *Web-course Search engine*, the *Metadata standard* used in this system, and the details of the course information collection, search criterion, search results and database structure.

1.3 Implementation

The architecture of the *Web-course Search engine* is based on a Web-based three-tier model (Figure 1.1): web browsers as tier 1 (presentation tier), servers (for example, the HTTP server) as tier 2 (business logic tier) running on the Java 2 Platform Enterprise Edition (J2EE), and database system as tier 3. The web browser side includes the course information input sheet for the educational course providers to provide the information about the course according to the *Metadata standard* that is mentioned in chapter 4, request sheet for searcher to submit query/request and search result sheet. On the tier 2, the J2EE provides a suitable containment environment for the business logic components. The J2EE manages these components efficiently and provides a number of services to the components. It uses JDBC to connect the tier 3, database. The data tier uses Cloudscape database Server to control and manage databases access.

The chapter 5 will describe the architecture of the application, which follows the Model-View-Controller design pattern, and how the application uses the deployment, transaction, JNDI service, and security capabilities of the Java 2 Platform Enterprise Edition (J2EE) to simplify component development and provide richer functionality.

1.4 Summary

The *Web-course search engine* eliminates various deficiencies of other search engines as mentioned in chapter 3. The system collects suitable information about educational resources, provides adequate search parameters for searchers to find resources and returns evaluative search results. By using the *Web-course search engine*, the learners and the other educational users are able to index, search and find useful, valuable and

related Web-course resources more effectively and efficiently. This is what educational Web course search mechanisms try to do.

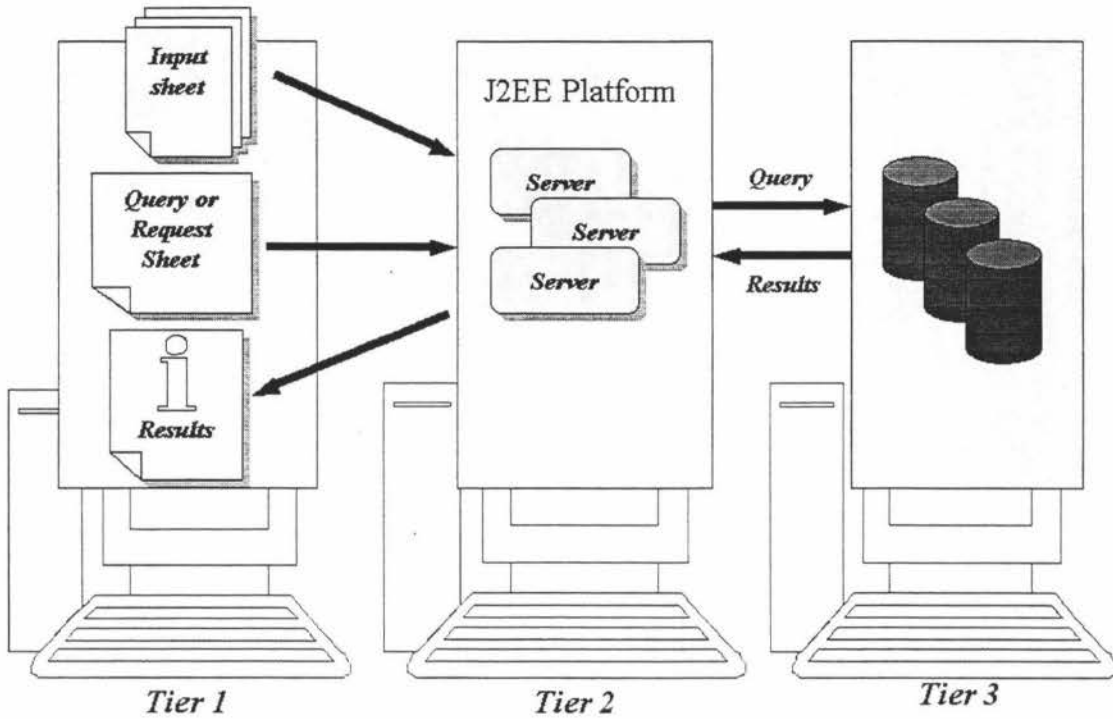


Figure 1.1 Architecture of the system

The chapter 6 provides some recommendations for future search engines/systems that are dedicated for finding educational teaching and learning resources.

Chapter 2 Standard and Metadata

2.1 Introduction

This chapter will introduce the Metadata concept, explain the benefits of using it on a Web search system, and then compare several Metadata Standards that are commonly accepted.

2.2 Why standard?

From Merriam-Webster Online: Collegiate Dictionary (Merriam-Webster Online: Collegiate Dictionary. (2002).), “Standard” means:

1. “Something set up and established by authority as a rule for the measure of quantity, weight, extent, value, or quality.”
2. “Something established by authority, custom, or general consent as a model or example.”

Standards are usually created to save time and money, and ensure quality, completeness and inheritance.

A technical/industrial standard is a specification of shared terms, interfaces, representations, practices, and so on. If a product is constructed according to an industrial standard, then that standard ensures that various manufacturers/users will be able to interact or interface with that product without requiring special help from the creator of the product. Therefore, a standard helps ensure cooperation and reuse.

2.2.1 Examples of standards

2.2.1.1 Food Nutrition Standard

Most food containers display information similar to Figure 2.1 on their labels.

Serving Size 4 pieces (30g)	
Amount Per Serving	
<hr/>	
Calories 150	Calories from fat 54
Total Fat 6g-----	9%
Cholesterol 0mg-----	0%
Sodium 0mg-----	0%
Total Carbohydrate 21g—	7%
Protein 2g-----	4%
Vitamin A	0%
Vitamin C	0%
Calcium	2%
Iron	4%
Potassium	3%

Figure 2.1 Nutrition information

Using this kind of nutrition label list, the consumer can make a knowledgeable decision about the product's suitability for certain purposes.

Why would the consumer trust this “*Nutrition information/ Nutrition facts*” label? Because it follows a food nutrition standard, every packaged food should have this nutrition label to show its nutritional content. This standard is used to ensure quality of food.

2.2.1.2 Library classification systems

Ip, Currie, Morrison and Mason (1999) have described a data structure to support resource discovery. The classification systems used in libraries are belonging to one kind of the data model advocated by Ip, Currie, Morrison and Mason (1999).

The Library of Congress Classification System (LC System) and the Dewey Decimal Classification (DDC) system are two most commonly used classification systems.

Library classification systems are example of **standards** that help users to search resources in the library more easily and efficiently.

2.2.2 Benefits of using a standard in a Web search system

Based on studies of existing information gateways (search engines), Ip, Morrison, Currie and Mason (1999) propose a data model and described mechanism to make educational metadata standards to enable educational resources. “Standards provide a common set of expressions. With standards, there is no confusion about what is being communicated by a particular expression. From one record to the next, the terminology remains the same.

Standards allow for quick location of a certain element. If a standard is used, finding a specific piece of information in a record is much easier than if no standard is used.

Standards enable automated searches. When standards are used, computers can be programmed to search and find useful data sets.” (Metadata education project: Metadata education suggestions and materials for: Overview of the Metadata content standard. (n.d.).).

Standards could “set the basis for software development by standardizing items and item descriptions. This has enabled the development of cataloging and indexing tools as well as search engines” (Tutorial- NC Metadata for the PRL and NCGDC- Why Bother?. (n.d.)) that can create queries against collections of vast number of records. Most records searched by Web search engines are based on metadata standards.

2.3 Metadata standards

Metadata is becoming a familiar term for information professionals. Until recently, only linguists were using the term "metadata." Today, it is hard to find a publication about electronic resources that ignores it.

"Metadata is data about data. It describes the attributes and contents of an original document or work, and can relieve potential data users of having to have full advance knowledge of a dataset's existence and characteristics...The term is generally applied to electronic resources and refers to "data" in the broadest sense--datasets, textual information, web pages, graphics, music, and anything else that is likely to appear electronically." (Milstead, J., & Feldman, S. (1999)).

“Metadata includes, but is not restricted to, characteristics such as the content, quality, currency, access, and availability of the data.” (ANZLIC Metadata Guidelines, Version 2, February 2001. Chapter Two: Metadata and the Metadata Standards. (2001)). The resources on the World Wide Web need to be described by some kind of metadata standard. The metadata, which is used for representing the Web-based educational resource, deals with the "what, when, who, where and how" of the source.

□ *What* – describes the title and description of the Web-based educational resource.

- *When* – shows when the Web-based educational resource was created and the update cycle.
- *Who* – mentions who are the Web-based educational resource originator or creator and supplier.
- *Where* – tells the user where to obtain this Web-based educational resource.
- *How* – presents how to obtain more information about the Web-based educational resource, available formats, access constraints, and so on.

In a standard format, metadata documents the characteristics of data so that the consumers can determine the data's fitness for their purpose.

Benefits of using Metadata

Besides the benefits of using a standard in a Web search system as mentioned in section 2.2.2, the document “Why is metadata important?” (Metadata education project: Metadata education suggestions and materials for: Why is metadata important?. (n.d.)) has mentioned more concrete benefits about metadata standard for a system. Smith (1996) enumerates the characteristics of metadata as it operates in traditional library contexts as:

- provides characterization of individual information objects in the collections of a library;
- is stored principally as the contents of library catalogs;
- is used principally in aiding users to access the information objects of interest.

An important use of metadata is to support selecting, understanding, utilizing, and

remembering sources and their contents (Rao et al. (1995)). One of encouraging facets of metadata is that they serve as an Interlingua, a common language by which a wide range of users can communicate. Though users' goals and jobs are various, metadata give data providers and users a way to describe aspects that are common across a variety of data themes and types.

The reasons that a corporation could protect its investment in data by using Metadata are that:

1. following Metadata, the old and new data is comparable, data is easy to reuse and update.
2. the descriptions of data sources are according to Metadata, and are fit for understanding and exchange.
3. according to a standard (Metadata), the other peers could easily follow the works.

There is not equivocal term under the same Metadata. The user will be not confused and can understand what is important according to the clear descriptions. It is easy for user to determine the data's fitness.

The searching and cataloging functions are more rely on and conform to Metadata standards. The multiple domains' data is becoming comparable and relative-able because of Metadata. The most common data catalog applications examples are the library catalog systems and the searching engines are very generally in the World Wide Web. Search engines can return the relevance of search results for pages that have correct metadata. In addition, at the meantime, many search engines did or will make use of metadata when indexing web pages sources.

With metadata, computer software scans the records, searching for matching

information in a few specific locations.

2.4 Examples of Metadata standards

From the description in the last two sections, it is evident that a suitable Metadata standard is needed to describe and manage all the information flow stages in our Web-course search engine. These information flow stages include the data collection, store, display, search queries and data exchange with other systems.

There are many standards, sub-sets and super-sets of standards available in the literature. The reason for so many standards is that metadata is used for different areas and aims. At its simplest, metadata is just a catalog record used for searching and locating data. Another use of metadata is as a management record. This requires more information about the dataset such as when it was last modified, what changes have been to it, and what are its limits. The third and most complex use of metadata is to design it to actually accompany a dataset, providing evaluable information on the dataset's development, specifications/structure, and content.

Next section describes four metadata standards, which are or could be used particularly for educational resources. They are widely accepted, and some of them have already been applied in educational systems. These Metadata standards are Learning Object Metadata (LOM), Dublin Core, EdNA (Education Network Australia) Metadata, and Gateway to Educational Materials (GEM) Metadata.

2.4.1 Learning Object Metadata (LOM)

The Draft Standard for Learning Object Metadata (LOM) is a proposed project of one of IEEE Learning Technology Standards Committee (LTSC)'s working groups "IEEE

P1484.12 Learning Object Metadata Working Group”. The mission of the LTSC working groups is “to develop technical Standards, Recommended Practices, and Guides for software components, tools, technologies and design methods that facilitate the development, deployment, maintenance and interoperation of computer implementations of education and training components and systems”(Learning technology standards committee (LTSC). (n.d.)).

The LOM Draft Standard defines a set of Metadata elements that can be used to describe learning resources. This includes the element names, definitions, data types, and so on.

The LOM Draft Standard is intended to support consistent definition of Metadata elements across multiple implementations, but does not (at the time of this writing) include information on how to represent Metadata in a machine-readable format, necessary for exchanging Metadata.

2.4.1.1 Definition and Purpose of LOM

Full Name of the Standard:

Standard for Information Technology --Education and Training Systems -- Learning Objects and Metadata.

The Draft Standard for Learning Object Metadata (LOM) “specifies a conceptual data schema that defines the structure of a metadata instance for a learning object. For this standard, a learning object is defined as any entity, digital or non-digital, that may be used for learning, education or training”(Draft Standard for Learning Object Metadata. (2001).).

The purpose of this standard is “to facilitate search, evaluation, acquisition, and use of

learning objects, for instance by learners or instructors”, and is also “to facilitate the sharing and exchange of learning objects, by enabling the development of catalogs and inventories while taking into account the diversity of cultural and lingual contexts in which the learning objects and their metadata will be exploited.” (Draft Standard for Learning Object Metadata. (2001).)

2.4.1.2 Overview

The LOM metadata structure is a hierarchy, or tree structure. At the top of the hierarchy/tree is the “root” element, LOM. The root element contains many sub-elements. If a sub-element itself contains additional sub-elements it is called a “branch”. Sub-elements that do not contain any sub-elements are called “leaves”. The relationship between the root, branches, and leaves is showed in Figure 2.2 using sample elements from the LOM Draft Standard.

Each element in the LOM Metadata hierarchy has a specific definition, data type, and allowable value space.

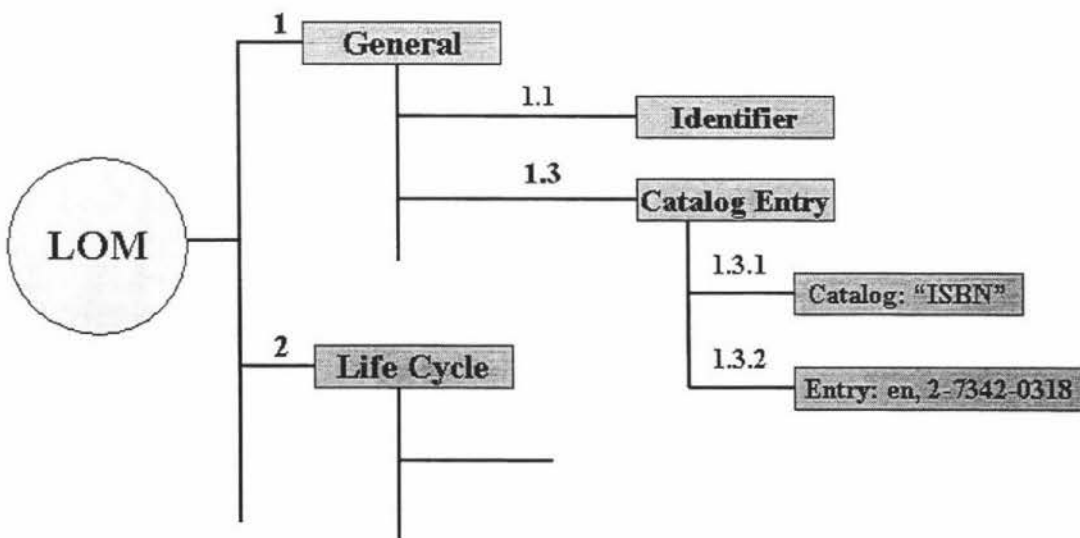


Figure 2.2 Hierarchy/tree structure of LOM

2.4.1.3 The LOM Metadata Elements and Structure

The LOM Draft Standard lists all of the meta-data elements in a table format. An outline of the LOM metadata elements as of draft 6.1 is described in Appendix B: Outline of LOM Metadata Elements.

Short descriptions of the main elements are as follows:

1: *General* provides the general information such as a globally unique label, title, language, and keywords.

2: *Life Cycle* describes the history and current state of the resource, for example version and status.

3: *Meta-Metadata* describes this metadata record itself, including information about those people or organizations that affected this metadata instance 's state (includes creator(s) and valuator(s)).

4: *Technical* provides information about format, size, requirements, some kinds of technical requirements and characteristics of the resource.

5: *Educational* lets you understand the major and basic educational or pedagogic characteristics of this resource, e.g., who are the audiences of this resource and the primary place/area where the resource will take place.

6: *Rights* shows all the information about copyright and about whether or not needs to pay for using this resource.

7: *Relation* tells you what is the relationship between this resource and other learning objects. If there are any those learning resources, this element should describe the details of those learning objects including identifier and description.

8: *Annotation* describes the comments from users who educationally used this learning object, and the details about the user(s).

9: *Classification* describes the particular classification system that the learning resource is belonging to. This category also provides a means of extending the LOM Metadata Elements to fit special aims and needs.

Further information on LOM is available at: <http://ltsc.ieee.org/wg12/index.html>.

2.4.2 Dublin Core

“Dublin Core metadata is used to supplement existing methods for searching and indexing Web-based metadata, regardless of whether the corresponding resource is an electronic document or a "real" physical object”(DCMI Frequently Asked Questions (FAQ). (n.d.).).

The Dublin Core Element Set comprises fifteen elements that together capture a representation of essential aspects related to the description of resources. Dublin Core metadata provides card catalog-like definitions for defining the properties of objects for Web-based resource discovery systems.

The majority of work on the Dublin Core has addressed the definition of semantics rather than syntax or structure, allowing rapid conceptual development free of the constraints imposed by specific implementation environments. Whilst beneficial in many ways, this has led to a certain lack of clarity at times, especially in relation to development of 'qualification' mechanisms that enrich descriptions in the Dublin Core. It has also made interoperable implementation difficult, as individual implementers have typically developed their own internal mechanisms for actually encoding Dublin

Core; mechanisms which are not always compatible with those of their potential collaborators elsewhere.

2.4.2.1 Elements and Structure

2.4.2.1.1 Structure

The Core contains just 15 metadata elements in three groups:

- ***Content***: Title, Subject, Description, Source, Language, Relation (to another resource), and Coverage (spatial or temporal characteristics of intellectual content).

- ***Intellectual Property***: Creator, Publisher, Contributor, and Rights.

- ***Instantiation***: Date, Type (such as home page, novel, working paper), Format (of data, to identify software and hardware required for use), and Identifier (such as URL or ISBN).

2.4.2.1.2 Elements

The Dublin Core Metadata Elements and their description are showed in Appendix C:
Dublin Core Metadata Elements

2.4.2.2 Benefit of using Dublin Core metadata

Dublin core metadata can be used to describe the resources of an information system. Although the Dublin Core is positioned as a simple information resource description, importantly it aims to provide a basis for semantic interoperability between other, probably more complicated, formats. One target use is to provide the basis for

resource-embedded description, initially with HTML documents. Web pages are one of the most common types of resources to employ the Dublin Core's descriptions, usually within HTML's meta tags.

The Dublin Core is easy to extend. The aim of Dublin Core is to define a simple set of data elements so that the authors and publishers of Internet documents could create their own metadata records with no professional training. It is recognized that the Dublin Core is a basic set, and that many 'publishers' or metadata producers may wish to extend this simple set with more specialized data.

Dublin Core metadata is being used as the basis for descriptive systems by several community interest groups such as: educational organizations, libraries, government institutions, scientific research sector, Web page authors, businesses requiring more searchable sites and corporations with vast knowledge management systems.

More information on Dublin Core is available at: <http://dublincore.org/>

2.4.3 EdNA Metadata

EdNA (Education Network Australia) is a framework for collaboration on the use of the Internet in education and training.

The EdNA Metadata Standard is based on the Dublin Core Metadata Element Set (Dublin Core Metadata Element Set, Version 1.1: Reference Description. (1999).).

The purpose of the EdNA Metadata Standard is "to support interoperability across all sectors of education and training in Australia in the area of online resource discovery and management"(EdNA Metadata Standard. (2001).). The principal application of

the standard at present is the EdNA Online.

2.4.3.1 Elements

The EdNA metadata include Dublin Core (DC) elements and elements specific on EdNA. Some of the EdNA elements are specifically for the administration of EdNA Online. The elements' details are showed at Table 2.1.

Dublin Core:

Identifier	Format
Title	Language
Description	Coverage
Subject	Rights
Publisher	Relation
Creator	Contributor
Date	Source
Type	

EDNA:

<i>Element</i>	<i>Definition</i>
EDNA. Audience	A category of user for whom the resource is intended
EDNA. CategoryCode	A numerical code derived from the database tables that support the EdNA Online Browse Categories.
EDNA. Entered	Date item was entered as an entry in the EdNA Online item database (used for management purposes).
EDNA. Indexing	A parameter which identifies to what extent the EdNA Online indexing ("spidering") software should follow hyperlinks from the described page.
EDNA. Review	A third-party commentary or formal review of the resource. This element is defined such that two forms of review are accommodated
EDNA. Reviewer	Name of person and/or organization or authority affiliated with the review.
EDNA. Version	Version of the EdNA Metadata Standard applied.

Table 2.1 EdNA Metadata Elements

More details on EdNA Metadata are available at:
<http://standards.edna.edu.au/metadata/>

2.4.4 Gateway to Educational Materials (GEM) Metadata Standard

The GEM Metadata Standard is used in the Gateway to Educational Materials (GEM) that is to solve educational materials resources, which are distributed on web sites across the Internet, discovery problem and by providing "The Gateway" to quality collections of educational resources.

Elements

The GEM Metadata Standard is also based on the Dublin Core Metadata Element Set (Dublin Core Metadata Element Set, Version 1.1: Reference Description. (1999).).

The details are described on Table 2.2.

Dublin Core:

Identifier	Format
Title	Language
Description	Coverage
Subject	Rights
Publisher	Relation
Creator	Contributor
Date	Source
Type	

GEM:

Element	Definition
GEM. Audience	The element contains information from a controlled vocabulary that most closely identifies the specific audience of the resource being described
GEM. Cataloging	The cataloging agency provides basic information about the agency that created the GEM catalog record.
GEM. Duration	The duration of the activity or lesson.
GEM. Grade	Grade, grade span, educational level, or age of the entity's audience.
GEM. Pedagogy	Denotes the student instructional groupings, teaching methods, assessment methods, and learning prerequisites of a resource.
GEM. Quality	The Quality Indicators element is a means for assessing the quality of instructional materials.
GEM. Standards	State and/or national academic standards mapped to the entity being described.

Table 2.2 *GEM Metadata Elements*

More details on GEM Metadata are available at:

<http://www.geminfo.org/Workbench/Metadata/index.html>

2.5 Comparison of various metadata standards

The Dublin Core Metadata Standard is the base of the EdNA Metadata and the GEM Metadata. In other words, the EdNA and GEM Metadata Standards are extensions of Dublin Core Metadata Standards. Their relationship is shown in Figure 2.3.

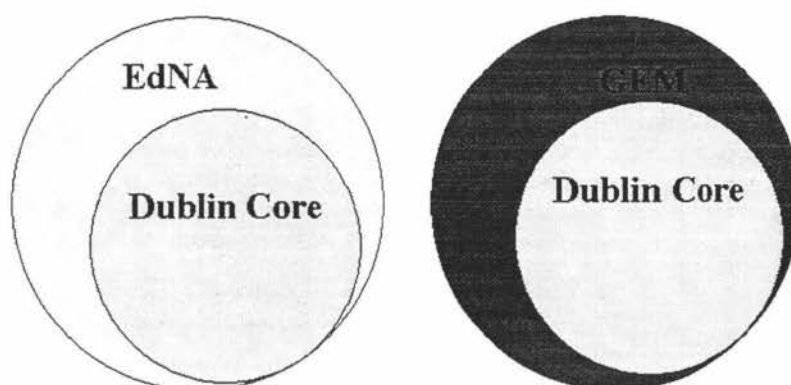


Figure 2.3 Relationships of EdNA/DC and GEM/DC

It is also possible to compare the LOM and Dublin Core metadata standards. One can easily map various Dublin Core elements to LOM elements as described in Table 2.3 (Draft Standard for Learning Object Metadata. (2001). Table B.1 - Dublin Core Mapping).

<i>DUBLIN CORE ELEMENT</i>	<i>LOM ELEMENT</i>
DC.Identifier	1.3:General.CatalogEntry. 1.1:General.Identifier
DC.Title	1.2:General.Title
DC.Language	1.4:General.Language
DC.Description	1.5:General.Description
DC.Subject	1.6:General.Keywords
DC.Coverage	1.7:General.Coverage

**DUBLIN CORE
ELEMENT**

LOM ELEMENT

DC.Type	5.2:Educational.LearningResourceType
DC.Date	2.3.3:LifeCycle.Contribute.Date when 2.3.1:LifeCycle.Contribute.Role has a value of "Publisher".
DC.Creator	2.3.2:LifeCycle.Contribute.Entity when 2.3.1:LifeCycle.Contribute.Role has a value of "Author".
DC.OtherContribut or	2.3.2:LifeCycle.Contribute.Entity with the type of contribution specified in 2.3.1:LifeCycle.Contribute.Role.
DC.Publisher	2.3.2:LifeCycle.Contribute.Entity when 2.3.1:LifeCycle.Contribute.Role has a value of "Publisher".
DC.Format	4.1:Technical.Format
DC.Rights	6:Rights
DC.Relation	7:Relation
DC.Source	7.2:Relation.Resource when the value of 7.1:Relation.Kind is "IsBasedOn".

Table 2.3 Dublin Core and LOM relation

It is therefore evident that the Learning Object Metadata (LOM) Metadata Standard includes the Dublin Core Metadata Standard.

It is also evident from the analysis that LOM, EdNA, and GEM are inherited from the Dublin Core Metadata Standard. The Figure 2.4 shows the inheritance relationship.

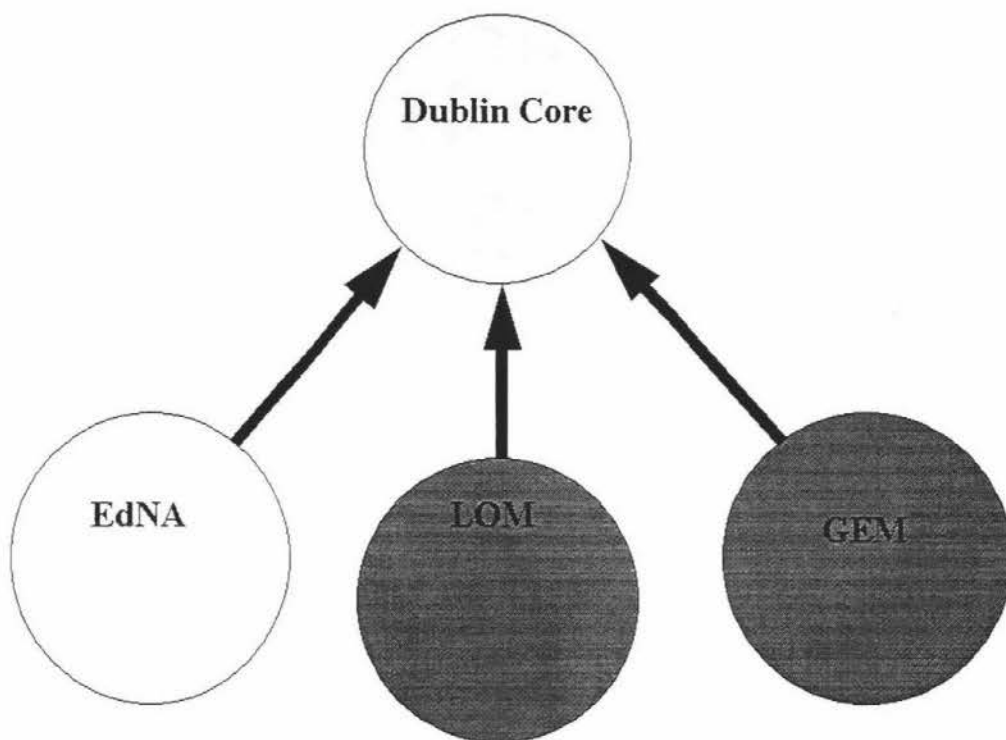


Figure 2.4 Inheritance Relationship (e.g. the LOM inherits from Dublin Core)

The analysis also reveals that most of the special Metadata Elements of GEM and EdNA also are the Elements of LOM, as shown in Table 2.4 and Table 2.5.

GEM METADATA ELEMENTS	LOM'S METADATA ELEMENTS
GEM. Audience	5.5 Intended end user role
GEM. Cataloging	9.2.1 Source
GEM. Duration	5.9 Educational. Typical Learning Time
GEM. Grade	5.6 Context & 5.7 Typical Age Range
GEM. Pedagogy	9.1 Purpose & 5.2 Learning Resources Type

GEM METADATA ELEMENTS	LOM'S METADATA ELEMENTS
GEM. Quality	4 Technical
GEM. Standards	2.1 Version

Table 2.4 GEM mapping LOM

EdNA Metadata Elements	LOM's Metadata Elements
EDNA. Audience	5.5 Intended end user role
EDNA. Approver	8.1 Annotation. Person
EDNA. CategoryCode	9.2.2.1 Id
EDNA. Version:	2.1 Version
EDNA. Review	8.3 Description, the content of this annotation
EDNA. Reviewer	8.1 Person, the person or organization who created this annotation.

Table 2.5 EdNA mapping LOM

The final results of the relationships of LOM, GEM, EdNA and Dublin Core are presented on Figure 2.5.

The other special Metadata Elements of GEM and EdNA are for their particular system and will not have any meaning for other systems. For example, the “*EDNA. Entered: Date*” item was entered as an entry in the EdNA Online item database (used for management purposes). Actually, after eliminating these little special elements from GEM and EdNA, these two Metadata standards are the real subsets of LOM Metadata standard.

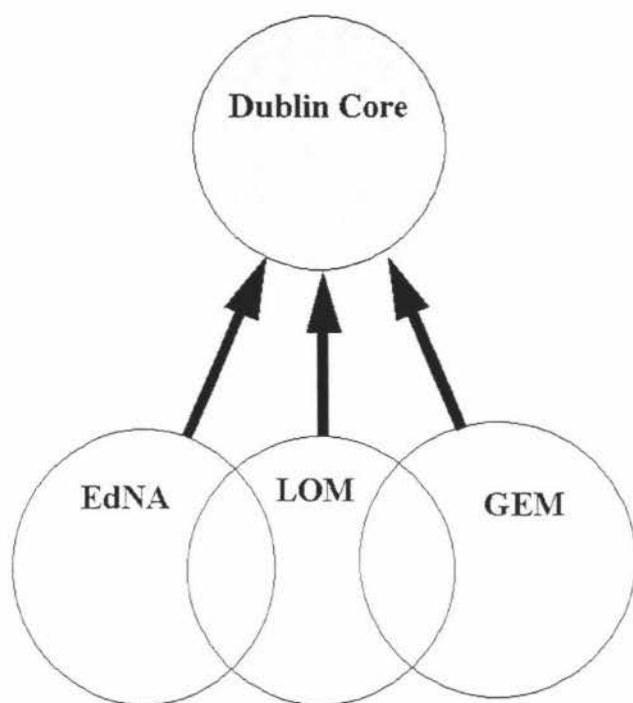


Figure 2.5 Relationships of Dublin Core, GEM, LOM and EdNA

2.6 Conclusion

This chapter introduced the Metadata concept, and explained the benefits of using it on a Web search system.

For a system that is based on data, Metadata can satisfy Data discovery, Data assessment, Data access, Data use, Data transfer, and Data management requirements. Metadata can provide a lot of benefits for the system, for examples, protect investment in data and help user understand data.

Different metadata standards are for different areas and aims. For the Web-course search purpose, there are several metadata standards including LOM, Dublin Core, EdNA and GEM Metadata. These Metadata standards are similar and comparable with each other.

The next chapter will discuss a survey of various Web-based search systems, and group them to systems that are dedicated to educational area, and common search engines or Web directories used for both educational and other purposes.

Chapter 3 Survey of various Web-based search systems

3.1 Introduction

The World Wide Web has presented researchers and learners all over the world with extraordinary opportunities to find and locate information. For example, in educational context, there are thousands of lesson plans, curriculum units, and other educational material distributed on web sites across the Internet. An increasing number of valuable resources are made available online every month, every day and even every minute. This provides a tremendous knowledge base for researchers and learners. However, it is quite often very difficult to find these useful resources in an efficient and effective manner, particularly by teachers and learners.

There are number of existing systems that attempt to help in the task of effective searching. These are systems that are dedicated to educational area such as “Education World”, “GEM (The Gateway to Educational Materials)” and “EdNA (The Educational Network of Australia) Online”. There are common search engines or Web directories that are used for both educational and other purposes such as “Lii”, “Yahoo”, “Northern Light”, “HotBot” and so on.

3.2 Systems dedicated to educational area

3.2.1 Education World

<http://www.education-world.com/>

The aim of “Education World” is to meet the needs for a Web site that would make the Internet easier for educators to use. It is a place on the Internet for teachers to gather and share ideas. It provides an online resource guide where educators could find the lesson plans and other research materials.

“Education World” is a search engine for educational Web sites only, a place where educators could find information without searching the entire Internet.

The “Education World” is quite similar to our system, the “*Web-course search engine*”. Nevertheless, the users of this system are educators, not the learners. Education World’s goal is to make it easy for educators to integrate the Internet into the classroom.

3.2.1.1 Information Collection

The system requests the course provider to provide following information: *Title*, *Author*, *Description*, *URL*, *Grade*, *Type*, *Subjects*, *Objectives/Goals*, *Keywords*, *Materials Needed*, *The Lesson*, *Assessment*, and *Lesson Plan Source* (Figure 3.1 and Figure 3.2).

The screenshot shows a web browser window with a form titled "ADD A SITE". The form has several sections:

- Title:** A text input field.
- URL:** A text input field containing "http://".
- Grade:** A dropdown menu with options: "All Grades", "K through 2nd", "3rd through 5th", and "6th through 8th".
- Type:** A dropdown menu with the option "Link".
- Top Category:** A dropdown menu with the option "CHOOSE ONE".
- Description:** A large text area.
- Comments:** A text area.
- FOR SCHOOLS & DISTRICTS ONLY:** A section with three dropdown menus: "Type of School" (Choose one or more), "School", and "Grades" (Choose one).
- YOUR INFORMATION:** A section with five text input fields: "Email", "Address", "City", "State" (with a dropdown menu showing "Outside United States and Canada"), and "Country".

On the left side of the browser window, there is a sidebar with a tree view of navigation links including Home, About Us, Search, and various subject categories like Art, Music, and Science.

Figure 3.1 “ADD A SITE” page

The Figure 3.1 is the “ADD A SITE” page: http://www.education-world.com/navigation/add_url_form.shtml and Figure 3.2 is the “Lesson Plan Submit” page: http://www.education-world.com/a_lesson/submit.shtml

3.2.1.2 Metadata standard

The *Education World* uses Metadata to describe the collected information, but there is no clear standard used in the system. The Metadata elements used in the system are sub-set of LOM (*Learning Object Metadata*) standard.



Figure 3.2 “Lesson Plan Submit” page

The comparison between the *Education World’s* Metadata standard and *LOM’s* Metadata standard is shown in the Table 3.1:

<i>Education World’s Metadata</i>	<i>LOM’s Metadata</i>
Teacher Name	2.3.1 LifeCycle Contributor Role has a value of “Author”
Lesson Title	1.2 General Title
Grade	5.6 Context or 5.7 Typical Age Range
Subject (s)	9.22 Taxon
Objectives/Goals	9.1 Classification Purpose
Keywords	1.6 keywords
Materials Needed	4.4 Technical Requirement & 4.3 Location
The Lesson	1.5 Description
Assessment	5.2 Learning Resource Type
Lesson Plan Source	6 Right & 7 Relation

Table 3.1 Metadata elements compare

Although Education World collects some primary information according to Metadata standard, it still lacks a lot of important information, for example, the course resource's language, format, and intended end user role.

3.2.1.3 Search criterion

The system's advanced search options (http://www.education-world.com/search/adv_search.jhtml) include *Keywords*, *Title*, *Description*, *URL*, and *Filter* (The principal environment within which the learning and use of this resource is intended to take place). Because of the limited search items, the search results lack precision. It will be, therefore, time-consuming for the searcher to determine the quality and authenticity of the resource. One reason for the limited search items is not enough Metadata elements in the collection stage.

3.2.1.4 Search results

The search results include *Title*, *Brief description*, and *URL* (Figure 3.3). These are not enough for the searcher to make informed decisions.

3.2.1.5 Users of the system

The *Education World* system is intended for educators.

3.2.1.6 Summary of the system

The *Education World* is a good search system for Web-based Educational courses, but because it uses a Metadata standard that is not enough to catch the suitable information from resources, the search function and results are not enough and precise for the users. However, because the area of *Education World* is targeted on education, its search results are targeted better than most of other common search engines.

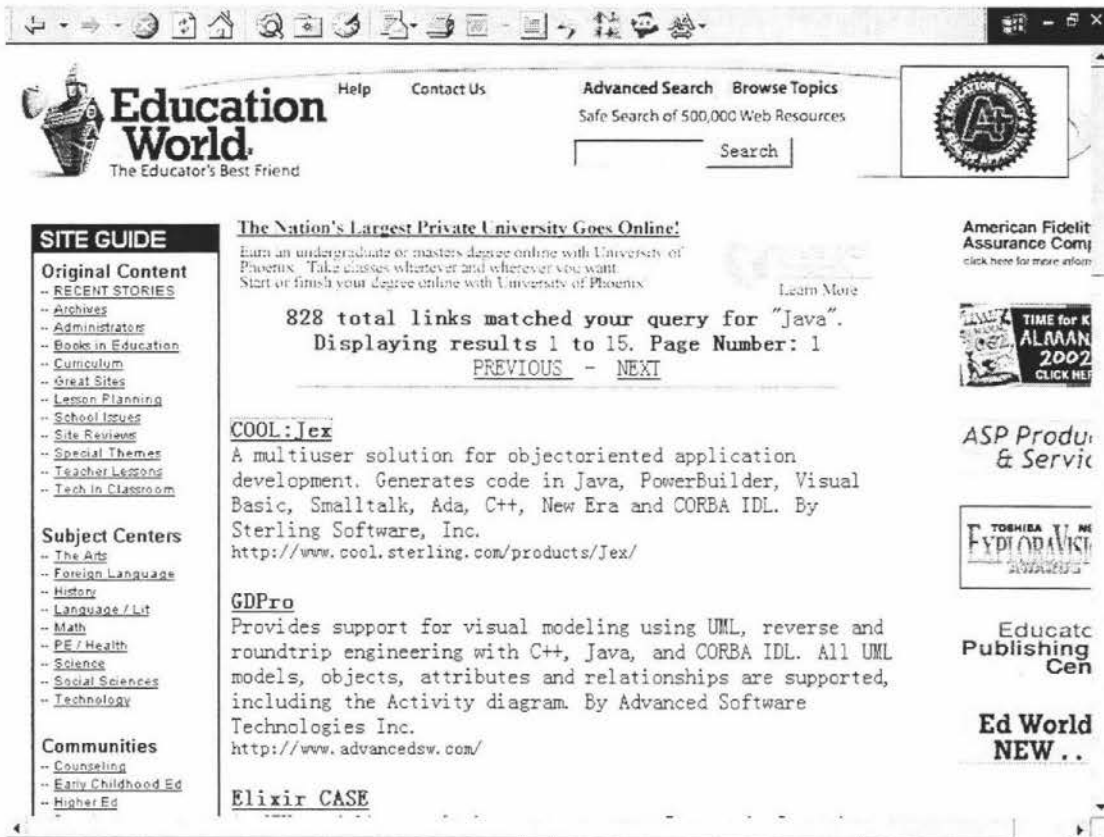


Figure 3.3 Search results

3.2.2 GEM (The Gateway to Educational Materials)

<http://www.thegateway.org>

GEM is particularly devoted to Web-based Educational Material including courses.

GEM is sponsored by the U.S. Department of Education (<http://www.ed.gov/>) and is a special project of the ERIC Clearinghouse on Information & Technology (<http://www.askeric.org/ithome/>).

The goal of the Gateway to Educational Materials (GEM) is to solve lesson plans, curriculum units and other educational materials discovery problem and to provide "The Gateway" to quality collections of educational resources.

The searchers can:

- Browse through lists organized by subject;

- Browse through lists organized by keywords;
- Search by subject, keyword, title, or grade level;
- Connect directly to the resource from The Gateway

The Gateway currently includes over 7800 records from nearly 200 collections.

3.2.2.1 Metadata standard

The *GEM* follows a set of Metadata standards and technical mechanisms to collect information. This set of Metadata standards and technical mechanisms are used to provide efficient, simple access to educational materials in *GEM*.

The Metadata standard used by *GEM* is the *Gateway to Educational Materials (GEM) Metadata* (section 2.4.4), which includes open standard Dublin Core and GEM Metadata Elements.

Dublin Core Elements:

Title	Coverage
Creator	Date
Description	Format
Identifier	Language
Other Contributor	Publisher
Relation	Resource Type
Rights	Source
Subject	

GEM Metadata Elements:

Audience	Duration
Essential Resources	Grade Level
Pedagogy	Quality Indicators
Academic Standards	Cataloging Agency

3.2.2.2 Information Collection

The system requests the course provider to provide information following the *Gateway to Educational Materials (GEM) Metadata*.

The *GEM* provides “*GEMCat Cataloging Module*” tool to resource providers to create and edit Metadata.

Through the “*GEMCat Cataloging Module*” software, resources are cataloged to Metadata. The cataloging module, *GEMCat* (Figure 3.4), is a stand-alone program for cataloging Internet resources using the GEM element set, profile, and controlled vocabularies.

3.2.2.3 Search criterion

The searchers can search by *Subject, Keywords, Title, Description, Grade Level, and Cost* (Figure 3.5).

They can also browse through lists organized by subject and by keywords.

3.2.2.4 Search results

The search results of *GEM* follow the same Metadata standards that the GEM uses to collect information from educational resources. Therefore, the results are quite useful and precise.

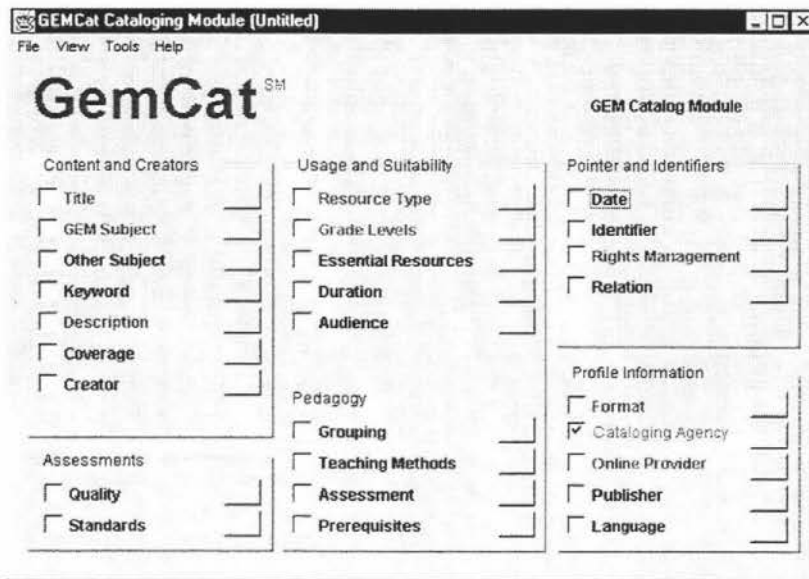


Figure 3.4 GemCat

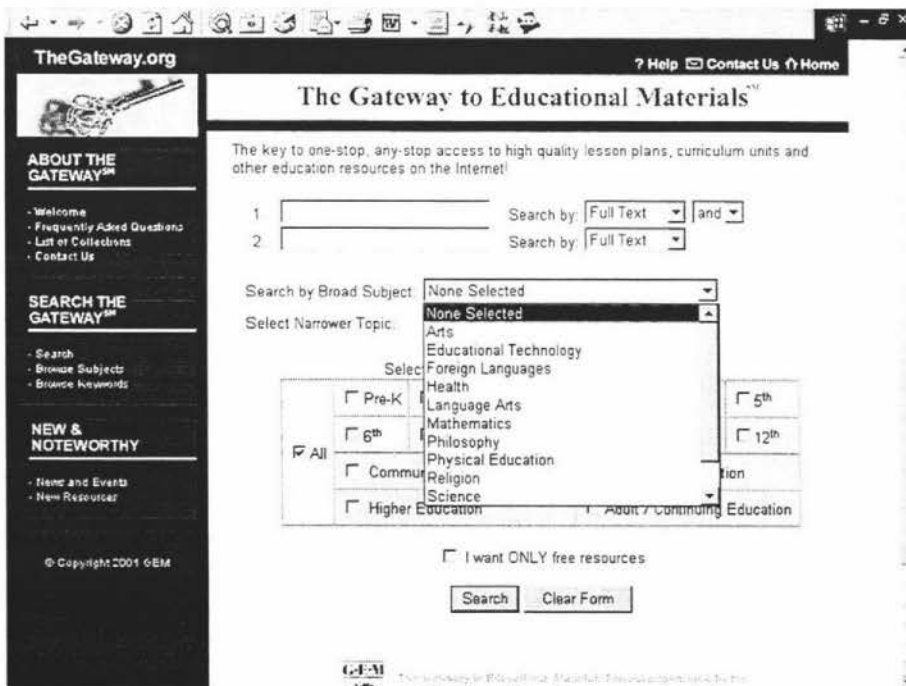


Figure 3.5 GEM Search

Here is an example of one of the search result shown in Figure 3.6:

Title: Mendelian Genetics

Note: Click above to connect directly to this resource.

Description:

Mendelian Genetics includes problems sets and tutorials on Monohybrid Cross, Dihybrid Cross and Sex-linked inheritance. The Biology Project, an interactive online resource for learning biology developed at The University of Arizona. The Biology Project is fun, richly illustrated, and tested on 1000s of students. It has been designed for biology students at the college and high school level, but is useful for medical students, physicians, science writers, and all types of interested people.

Grade Levels: 10 11 12
Adult/Continuing education,Higher education

GEM Subjects: Science--Biology
Science--Chemistry
Science--Embryology
Science--Pharmacology

Keywords:

Genetics, Chromosome, Blood type, Inheritance, Monohybrid Dihybrid, **Sex** linked, Drosophila, Fruit Fly,

GEM Audience:

Tool For: Teachers
Beneficiary: Students

Pedagogy:

Teaching: Computer assisted instruction
Method: Computer simulations
Learning modules
Tutorial programs
Grouping: Individualized instruction
Assessment: Self evaluation

Resource Type: Unit of instruction

Format: text/HTML

Relation:

The Biology Project of The University of Arizona

Essential Resources:

Internet Access and Browser,

Rights Management:

Cost: Free

Name: http://www.biology.arizona.edu/the_biology_project/copyright.html

Date:

Record Created: 2000-02-08T18:33:06-5:00

Language of
Resource:

English, Spanish,

Publisher:

Name: The Biology Project at The University of Arizona

Role: onlineProvider

Homepage: <http://www.biology.arizona.edu/contact/>

Publisher:

Name: The Biology Project of the University of Arizona

Role: publisher

Contact: Madeleine Lapointe

Email: <mailto:lapointe@u.arizona.edu>

Homepage: <http://www.biology.arizona.edu/ml/>

Dataloging Agency:

Name: Garry Forger

Version: 3.2

Homepage: <http://www.u.arizona.edu/~gforger>

Figure 3.6 *GEM search result*

3.2.2.5 Users of the system

Teachers, parents, administrators and students can search or browse the *Gateway* and find thousands of high quality educational materials, including lesson plans, activities, and projects from over 200 GEM Consortium member sites.

3.2.2.6 Summary of the system

The Gateway to Educational Materials (GEM) is an exciting, powerful existing search system for Educational Materials.

When teachers or learners connect to the *Gateway*, they are able to quickly and efficiently access the Internet-based educational resources of participating GEM Consortium members. When they use The *Gateway* database, rather than an Internet search engine, teachers, or other users are able to locate resources they need quickly and efficiently.

The system's Metadata Standard, the *Gateway to Educational Materials (GEM) Metadata*, is good enough to suit to the system needs. In addition, the *GEM* provides adequate search parameters to searcher to find resources. However, from the section "3.2.2.3 Search criterion", we know that the system does not provide adequate search parameters to the searcher to find resources. This is the main shortcoming of the *GEM*.

3.2.3 EdNA Online

<http://www.edna.edu.au/EdNA>

"EdNA Online is a service that aims to support and promote the benefits of the Internet for learning, education and training in Australia." (About EdNA online. (n.d.)). It contains 9000 evaluated sites and over 250000 linked sites. These resources have been collected from stakeholder collections and from international sources by Directory Officers and through local contributors (Currie, M., Moss, N., Morrison, I. Ip, A. (1999)). *EdNA* Online is a vast resource contributed to and used by the education community.

3.2.3.1 Metadata standard

The system uses *EdNA* Metadata Standard, which includes *Dublin Core* Metadata Standard, to collect information of resources. The *EdNA* Metadata Standard 's elements have been created specific to *EdNA*. The format is similar to *Dublin Core* elements, but these elements do not have any meaning outside the *EdNA* environment.

Dublin Core Elements:

- Identifier
 - Title
 - Description
 - Subject
 - Publisher
 - Creator
 - Date
 - Type
 - Format
 - Language
 - Coverage
 - Rights
 - Relation
 - Contributor
 - Source
-

EDNA Metadata Elements:

- Entered
 - Approver
 - Reassessment
 - Userlevel
 - Categories
 - Conditions
 - Indexing
 - Review
 - Version
-

The system requests the course provider to provide information following the *EdNA Metadata Standard* by using the *EdNA Metadata Toolset*.

The *EdNA Metadata Toolset* is to generate and save metadata conforming to *EdNA Metadata Standard*.

3.2.3.3 Search criterion

Users can use “Meta Search”, which facilitates complex searches only on EdNA evaluated items according to EdNA Metadata Standard.

3.2.3.4 Search results

The system displays the search results in a very simple way, showing the Title and a short description (Figure 3.7). These results lack the amount of information required by the searchers to make informed decisions.

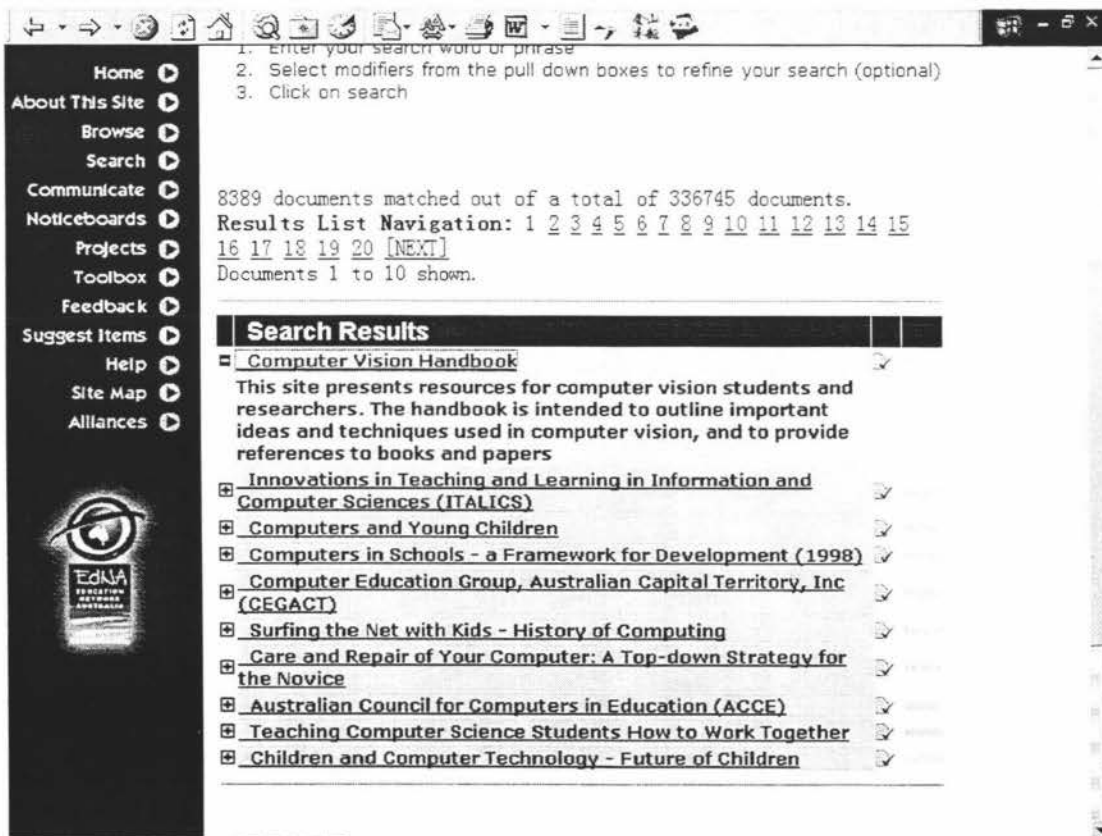


Figure 3.7 EdNA search results

3.2.2.5 Users of the system

The users of *EdNA Online* include a wide range of educators, librarians, support-staff, associations, trainers, and students. These people are both users and contributors of information on EdNA Online.

3.2.2.6 Summary of the system

EdNA Online has a good collection and search mechanism based on *EdNA Metadata Standard*, but is poor in representing the search results.

3.3 Common search engines or Web directories used for both educational and other purposes

Most search engines operate by looking through the Internet and indexing Web pages. The search engines find a lot of information. Most of the resources in the search tools are submitted by domain experts and are of high quality. Some entries are submitted by information agent, which used techniques found in (Neri & Saitta, 1997) and (Theilmann, Rothermel, 1998) to surf the web and recover resource related to those submitted by the domain experts. However, these search engines pay more attention to quantity than to quality. The search function and results are not enough and precise for the users.

Normally, commercial search engines use an overturned index for searching, and the results are thousand of pages covering widely different meanings to the keywords (Ip, A., & Naidu, S. (2001)). These search engines can be helpful if users understand their underlying mechanisms (Taylor and Clemson, 1996).

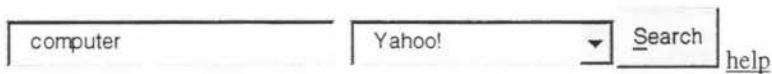
Web directories likes *Yahoo* are maintained by groups of professional catalogers. These Web directories also contain resources from user submissions, e.g. Yahoo's "Suggest a Site". Because these Web directories are created manually computer, they contain some helpful description for the searchers to help on their decision process. In addition, because the Web directories are also can come from user submissions, there should be some kind of untruth.

Following sub-sections discuss a number of common search engines: Yahoo, Northern Light, HotBot, Lii.org.

3.3.1 YAHOO

3.3.1.1 Search criterion

The Yahoo Advanced Search page offers additional options (Figure 3.8) for searching. Searches can also be limited to listings added in the last day, week, or six months. Boolean operators (and, or) and string search are also supported.



computer Yahoo! Search help

The options below apply only to Yahoo Directory searches:

Select a search method:

- Matches on all words
(AND)
- Matches on any word (OR)
- An exact phrase match

Select a search area:

- All
- Yahoo! Categories
- Web Sites

Find only new listings added during the past

After the first result page, display matches per page

Figure 3.8 YAHOO Search Options

3.3.1.2 Search results

Each item of the search results is a very simple description, which includes the title, short explanation and the link of “More sites about: this Title”. The search term(s) appear in bold on the results screen (Figure 3.9).

-
1. Apple **Computer**, Inc.
<http://www.apple.com/>
More sites about: **Computer** Manufacturers > Apple **Computer**
 2. Compaq **Computer** Corporation - designs, develops, manufactures and markets a range of computing products, including desktop and portable **computers** and tower PC servers.
<http://www.compaq.com/>
More sites about: **Computer** Manufacturers > Compaq **Computer** Corporation
 3. Dell **Computer** - designs, develops, manufactures, markets and services and supports a range of **computer** systems, including desktop, notebooks, and enterprise systems (includes servers and workstations). Dell **Computer** Corporation also markets software, peripherals and service and support programs.
<http://www.dell.com/>
More sites about: **Computer** Manufacturers > Dell **Computer**
 4. MIT Laboratory for **Computer** Science
<http://www.lcs.mit.edu/>
More sites about: Massachusetts > Cambridge > Massachusetts Institute of Technology (MIT) > Department of **Computer** Science
 5. **Computer** Words - defines common terms and concepts for non-technical people.
<http://www.computerwords.com/>
More sites about: **Computers** and Internet > Dictionaries
 6. CDW **Computer** Centers, Inc. - mall of PC and Macintosh products.
<http://www.cdw.com/>
More sites about: **Computer** Retailers > CDW **Computer** Centers
 7. **Computer** IEEE - the world-renowned monthly magazine received by all members of the IEEE **Computer** Society.
<http://www.computer.org/>
More sites about: Institute of Electrical and Electronics Engineers (IEEE) > IEEE **Computer** Society
-

Figure 3.9 YAHOO search results

3.3.2 Northern Light

<http://www.northernlight.com/>

3.3.2.1 Information collection

The system collects URL, Name, and E-mail Address of resource.

3.3.2.2 search criterion

Title, Publication and URL are the search items. Searches can also be limited to listings added in subject area, document sort, language, country and date range. Boolean operators (and, or) and string search are also supported. (Figure 3.10).

The screenshot shows a web browser window with the Northern Light search interface. The browser's address bar is empty. The search interface includes a search bar with a 'Search' button and a 'Tips' link. Below the search bar are four input fields: 'Words in title', 'Publication name', 'Words in URL', and 'See our list of publications'. A 'SELECT' section allows users to choose between 'All Sources', 'Special Collection', and 'World Wide Web'. There are two sections for limiting results: 'LIMIT SUBJECTS TO' and 'LIMIT DOCUMENTS TO'. The 'LIMIT SUBJECTS TO' section has 12 checkboxes for various subject areas. The 'LIMIT DOCUMENTS TO' section has 12 checkboxes for document types. Below these sections are two dropdown menus: 'Documents written in: Any language' and 'Web sites from: All countries'. At the bottom, there is a 'SELECT DATE RANGE' section with a note 'Fill in one date field or both to narrow your results by date.' and two input fields for 'Start date:' and 'End date:' with a date format 'mm/dd/yyyy'.

Search for _____ **Search** [Tips](#)

Words in title _____

Publication name _____ [See our list of publications](#)

Words in URL _____

SELECT All Sources Special Collection World Wide Web

LIMIT SUBJECTS TO

<input type="checkbox"/> Arts	<input type="checkbox"/> Gov't, Law & Politics	<input type="checkbox"/> Social sciences
<input type="checkbox"/> Business & Investing	<input type="checkbox"/> Health & Medicine	<input type="checkbox"/> Sports & Recreation
<input type="checkbox"/> Computing & Internet	<input type="checkbox"/> Humanities	<input type="checkbox"/> Technology
<input type="checkbox"/> Contemporary life	<input type="checkbox"/> Products & Services	<input type="checkbox"/> Travel
<input type="checkbox"/> Education	<input type="checkbox"/> Reference	
<input type="checkbox"/> Entertainment	<input type="checkbox"/> Science & Mathematics	

LIMIT DOCUMENTS TO

<input type="checkbox"/> Commercial Web sites	<input type="checkbox"/> Expert Bio	<input type="checkbox"/> Broadcast news transcripts
<input type="checkbox"/> Education Web sites	<input type="checkbox"/> Learning materials	<input type="checkbox"/> City & regional newspapers
<input type="checkbox"/> Government Web sites	<input type="checkbox"/> Questions and answers	<input type="checkbox"/> College newspapers
<input type="checkbox"/> Military Web sites	<input type="checkbox"/> For sale	<input type="checkbox"/> Newspapers, wires & transcripts
<input type="checkbox"/> Non-profit Web sites	<input type="checkbox"/> Job listings	<input type="checkbox"/> Press releases

Documents written in:

Web sites from:

SELECT DATE RANGE

Fill in one date field or both to narrow your results by date.

Start date: End date: *mm/dd/yyyy*

Figure 3.10 Northernlight.com Power Search

3.3.2.3 Search results

The results are arranged in order by relevance rank in percentile (100% is the best). Each search result details include URL, title, relevance rank in percentile, an extract from the file, Commercial sites and the "More results" link from this site (Figure 3.11).

1.  **COMPUTER PEOPLE**

54% - Directories & Lists: PEOPLE and PIONEERS. The Winter 2000 issue of "Invention and Technology" included an interesting picture (above) and description of the 20 "greatest" innovators of the ... 02/15/2001

Personal page: <http://ei.cs.vt.edu/~history/people.html>

[More results](#) from this site

2. [ugbook.....html](#)

53% - Articles & General info: INFORMATION UNDERGRADUATES Introduction The Nature of the Subject Is Computer Science you? Prospects Computer Science Graduates The Department Computing Provision Teaching Laboratories The College Computing ... Date Not Available

Personal page: <http://www.dcs.qmw.ac.uk/~suew/ugbook>

Figure 3.11 Northern Light search results

3.3.3 HotBot

<http://www.hotbot.com/>

3.3.3.1 Search criterion

The Advanced Search of the system supports the language limit, date limits, media type (VRML, Audio, JavaScript, and so on.) and location/domain. The search also supports the word filter, which provides the "must contain", "should contain" or

“must not contain” the words, the person or the phrase Boolean operators (Figure 3.12).

3.3.3.2 Search results

The search results are displayed with the document title, a brief description, the last update time, and the URL (Figure 3.13).

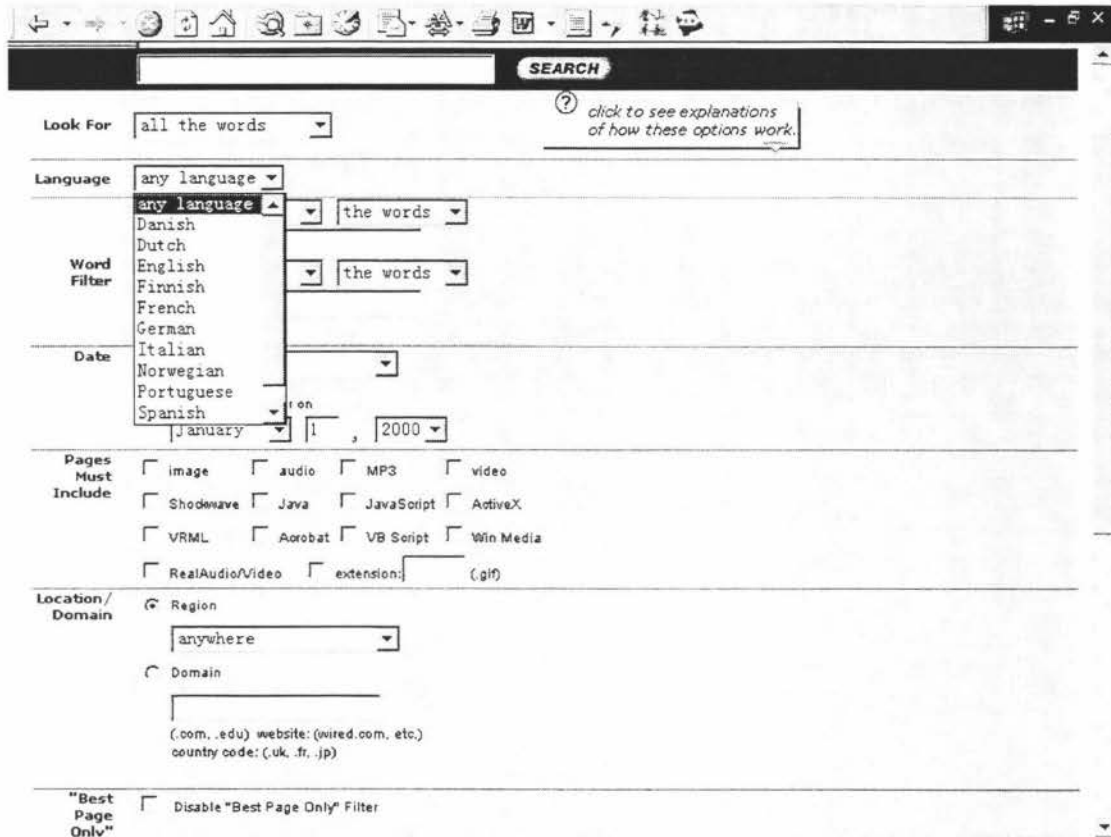


Figure 3.12 Hotbot advanced search

2.~/font> **JSG - Java Study Group**

JSG - Java Study Group news 001117: SC22 decided at its plenary meeting September 2000 to close down the JSG group. news 981214: Recommendations and documents from meeting 3 (Tokyo) available news 9

<http://www.dkuug.dk/JTC1/SC22/JSG>

See results from [this site only](#).

3.?

<http://www.woj.com/tech/javacert>

The Java Certification Online Study Group has moved to javacert.com.

Please update your bookmarks.

<http://www.woj.com/tech/javacert>

See results from [this site only](#).

4.?

[Click here to buy The Complete Java 2 Certification Study Guide at Amazon.com](#)

Shop at Amazon.com where you will find a huge selection of books at great prices. Shop here for The Complete Java 2 Certification Study Guide and find more books by Simon Roberts.

<http://www.amazon.com/exec/obidos/ASIN/0782127002/inktomi-bkasin-20>

See results from [this site only](#).

5.?

[C and Java SIG](#)

The C and Java SIG - A Special Interest Group of the NYPC User's Group

<http://www.cppsig.org/>

See results from [this site only](#).

Figure 3.13 Hotbot search results

3.3.4 Lii.org (Librarians' index to the Internet)

<http://lii.org/>

The **Librarians' Index to the Internet** (LII) is “a searchable, annotated subject directory of more than 8,500 Internet resources selected and evaluated by librarians

for their usefulness to users of public libraries. LII is meant to be used by both librarians and non-librarians as a reliable and efficient guide to described and evaluated Internet resources. “(About the LII. (2001)).

This subject directory is specifically aimed at the needs of California public library users, but in doing so, it often identifies the essential resources on a variety of topics.

3.3.4.1 Information Collection

The system collects *Author Name, Descriptions, Indexer, Keywords, Links, Publisher Name, Subjects, Titles, URL*, and so on.

There are around 8,500 links to Internet resources briefly annotated, with assigned subject headings

3.3.4.2 Search criterion

The advanced Search (Figure 3.14) offers multiple fields (Title, Subject, Description, Author, Publisher, URL, Indexer and Category). The Boolean relationships between these fields are AND, OR or NOT. The search limits to category of record to “Best of”, “Directories”, “Databases”, or “Specific Resources”, and supports the “Use Stemming in search” “No Stemming” options that turn stemming on/off.

3.3.4.3 Search results

The search results are arranged in alphabetic order by the resource titles (Figure 3.15). The information of each item includes the title, URL, description (bolded the search term(s)), subject(s), the created information (author(s), created date, modified by, modified date, and so on).

3.3.4.4 Users of the system

The main users of the system are the public library users.

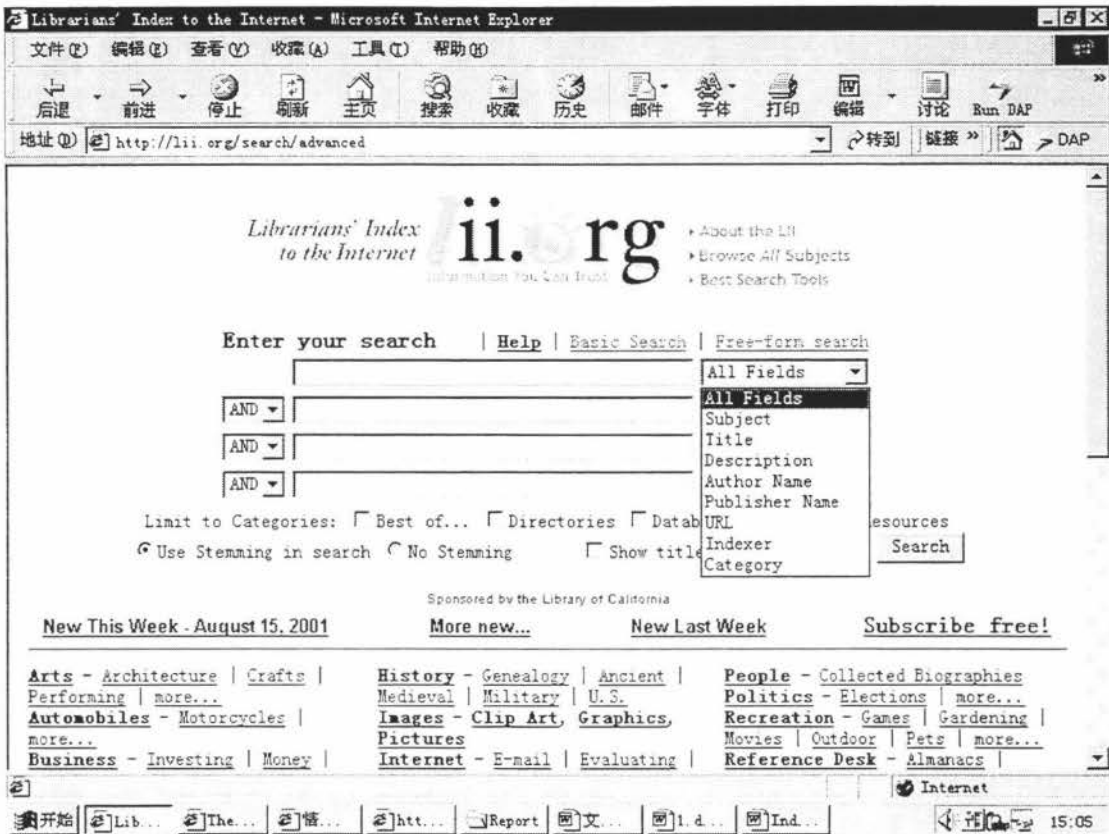


Figure 3.14 lii.org-advanced searches

Best of...

Andover News Network - <http://www.andovernews.com/>

Provides, from one Web page, daily Information Technology news in nine subject areas: Apple, Hardware, Internet, IT, Java, Microsoft, Netscape, Software, and Telco. Sources include *InfoWorld*, *Computer News Daily*, *Interactive Week*, *Wired*, *Techwire*, and *ZDNNews*, among others. A great current awareness source.

Subject: Information technology -- Newspapers

Created by: cl on Mar. 12, 1998 - last modified by pf on Dec. 10, 2000

CMPnet - <http://www.cmpnet.com/>

Excellent coverage of computer technology for end users and industry professionals. Some of their many resources include TechWeb with industry news; E-Business with details on computer-related investing; TechReviews with pc and software reviews for technology professionals. Also provides material concerning emerging technologies (including specialized coverage of ActiveX and Java), and a searchable encyclopedia of technology terminology.

Subjects: Computers | Computer industry

Created by: cl on Nov. 28, 1999 - last modified Aug. 2, 2001

Web Developer's Virtual Library - <http://wdvl.internet.com/>

WDVL contains an excellent collection of Web site development resources. They cover authoring, design, graphics, the Internet, and include beginner and intermediate guides and tutorials on HTML; DHTML; VRML; XML; ASP; Cascading Style Sheets (CSS); Flash; SSI; Web Programming Scripting (Java, JavaScript, CGI, Perl, PHP) Graphics (Adobe Photoshop, Paint Shop Pro); Databases (SQL) for the Web; and multimedia (Shockwave, Real Audio, Midi). Also check out their reviews of software related to Web site development.

Figure 3.15 lii.org search results

3.4 Discussion on common search engines

These common aims search engines or Web directories systems (e.g. Lii.org, Yahoo, Northern Light, HotBot) are not constructed according to some kind of publicly available standard, for example Metadata standards. Since these search engines do not collect and provide clear and sufficient information about the resources, it is difficult for the searchers (e.g. learners and researchers) to efficiently start an exact search according to their personal request.

There is lack of precision in web searching approach and therefore the results from search generally include thousands of hits, mostly irrelevant. These results then require sorting and evaluation by the searcher. If the searchers were the expert Web information hunters like a Yahoo catalogers, they would face fewer problems in such situation. However, the process of searching and determining quality, authority and serviceability of the thousands of search results is very time-consuming for the inexperienced searchers, specially the learners, who do not have much knowledge of the subject they intend to study.

3.5 Conclusion

This chapter provided a survey of various Web-based search systems, described their strong points and shortcomings, and showed the role of Metadata Standard in them. Some of these systems have educational focus; the others are for common purposes.

Various metadata standards for interactive communication areas, especially for the Internet, are an attempt to create a commonly accepted language between the professional catalogers and the common users, for example, the Internet searchers.

Following a broadly accepted Metadata standard in a particular area, such as education, it is easy to document and describe resources in their area. This is what educational Web course search mechanisms try to do.

Adoption of Metadata Standards makes it easier for these systems to collect the information about the resources and maintain the data on databases using XML and RDF, compared to other systems that do not use Metadata Standards. After getting information about resources, the searchers can also search and get the results in standard format based on Metadata Standard.

However, because data creation is time-consuming and complicated by itself, and metadata creation is often considered an added burden, Gelbman and Mathys (1999) discovered that a majority of data-producers continues to be unwillingness to assign time to creating metadata.

Further more, several questions remain: First, is a particular Metadata Standard good enough to suit the system needs or particular requests? Second, does the system provide adequate search parameters to the searcher to find resources? “Adequate search” is the “cause”; “adequate search results” is the “outcome”. Finally, does the system return evaluative search results? In other words, is it easy for the searchers to evaluate the suitability of results to their requirements?

The next chapter will discuss how the “*Web-course Search engine*” overcomes these problems. It will also introduce the Metadata standard that is used in the system.

Chapter 4 Using Metadata Standard to construct “Web-course Search engine”

4.1 Introduction

The “*Web-course Search engine*” is an online search engine, which is dedicated to educational area. This chapter describes the functional outline of the system, and explains the *Metadata Standard* that is used in the system. After introducing the standard, the chapter will show how the system uses this standard to construct its structure model by describing the “Information-sheet”, “Search Request”, “Search Result” and “Database Structure”.

4.2 Functionality of the “Web-course Search engine”

The “*Web-course Search engine*” is particularly devoted on searching Web-based Educational courses, and provides educators/teachers and learners with quick and easy access to various educational courses on the Internet.

The goal of the “*Web-course Search engine*” is to facilitate discovery of lesson plans, curriculum units, and other educational material and to index collections of educational courses by providing member providers the “Add Course” and “Modify Course” functions.

Education courses are the focus of “*Web-course Search engine*”. The learners can use this system to search Web courses to suit their needs. Information retrieval and resource discovery are important foundations on which “*Web-course Search engine*” is built.

The searchers can

- search by subject, keyword, language, author, or cost;
- jump directly to the resource from The “Web-course Search engine”.

In this system, the information of Web-course is provided directly by the Web-course providers, there is no middleman cataloger who involved in the process.

4.3 Metadata Standard of the “Web-course Search engine”

As per the discussion in previous chapters, it is evident that the use of an adoptable Metadata Standard is very important in the construction on a widely accepted system, especially for a Web search engine. The “*Web-course Search engine*” in this project is constructed under a Metadata standard.

Only those elements necessary for the discovery of the resource were considered. It was believed that resource discovery is the most pressing need that metadata can satisfy (Weibel (1995)). The “*Web-course Search engine*” uses a Metadata Standard that is subset of *LOM*. The reason why we don’t use the *LOM* Metadata Standard directly to the “*Web-course Search engine*” is that the system is particular focus on learners and friendliness. For example, the learners would probably not write down the Web course ID and it may be too much to ask them to provide some feedback or comments on the system after they have used the Web courses. Therefore, *LOM* Metadata Standard’s element “8 Annotation” is not an element of the “*Metadata Standard*” used in the system.

Like *LOM*, this Metadata Standard definition is also a hierarchy, or tree structure (Please refer section 2.4.1.2).

4.3.1 Summary of the “*Metadata Standard*”

4.3.1.1 *Purpose of the standard*

The purpose of this standard is to ease search, evaluation, acquisition, and use of Web courses, for instance by learners or educators/teachers.

4.3.1.2 *Basic metadata structure*

The data elements of the metadata describe a learning course and are grouped into *categories*. The Base Scheme consists of seven such categories:

- a) The General category groups the general information that describes the learning object as a whole.
- b) The Lifecycle category groups the features related to the history and current state of the learning object and those who have affected this learning object during its evolution.
- c) The Technical category groups the technical requirements and characteristics of the learning object.
- d) The Educational category groups the educational and pedagogic characteristics of the learning object.
- e) The Rights category groups the intellectual property rights and conditions of use for the learning object.
- f) The Relation category groups’ features that define the relationship between this learning object and other targeted learning objects.

- g) The Classification category describes where this learning object falls within a particular classification system.

4.3.1.3 Definition

The definition of the “*Metadata Standard*” used in this project is shown in Table 4.1. The “*value space*” in the fourth row means: the set of allowed values for the data element.

<i>Nr</i>	<i>Name</i>	<i>Explanation</i>	<i>Value space</i>
1	General	This category groups the general information that describes this resource as a whole.	
1.1	URL	A globally unique Web address that identifies this resource.	
1.2	Title	Name given to this resource.	
1.3	Language	The primary human language or languages used within this resource to communicate to the intended user.	Chinese, Danish, English, French, Germany, Hindi, Italian, Japanese, Korean, Latin, Norwegian, Spanish, Swedish, other
1.4	Description	A textual description of the content of this resource.	
1.5	Keywords	Keywords or phrases describing this resource.	

1.5.1 Primary Keyword

IRIS Keyword Thesaurus. (n.d.).

The list is:

Agriculture/ Area studies/ Arts/
 Behavioral sciences/ Biotechnology/ Business
 administration/
 Communication/ Computer Science/
 Earth Sciences/Education/
 Energy/Engineering/Environmental Sciences/Ethnic and
 Racial Group Studies/
 Foods and Food Science/
 Gender Studies/Geographic Locations/
 Home Economics/Humanities/
 Information Science/International Studies
 Law/
 Leisure Studies/Library Science/Life Sciences/
 Mathematics/Medical Sciences/Military
 Sciences/Musicology/
 Natural Sciences/Nuclear Sciences/
 Opportunity Restricted to Minorities - Minority
 Institutions/Opportunity Restricted to Women/Opportunity
 for Junior Faculty/
 Physical Sciences/ Plant Sciences /Population Studies/
 Rural Studies/
 Science/ Social Sciences/Space Sciences/
 Technology/ Transportation/
 Urbanism

1.5.2 Secondary Keyword

The same as 1.5.1 Primary Keyword

1.5.3 Additional Keyword

1.6 Coverage

The span or extent of such things as time, culture, geography or region that applies to this resource.

1.7 Aggregation Level

The functional granularity of this resource.

0 = the smallest level of aggregation
 e.g. raw media data or fragments.
 1 = a collection of atoms
 e.g. an HTML document with some embedded pictures
 e.g. a lesson.
 2 = a collection of level 1 resources
 e.g. a 'web' of HTML documents, with an index page that links the pages together
 e.g. a course.
 3 = the largest level of granularity
 e.g. a set of courses that lead to a certificate.

2 LifeCycle

This category describes the history and current state of this resource and those who have affected this resource during its evolution.

2.1	Version	The edition of this resource.	
2.2	Status	The state or condition of this resource.	Draft Final Revised
2.3	Contribute	This data element describes those people or organizations that have affected the state of this resource during its evolution (includes creation and publication).	
2.3.1	Author		
2.3.1.1	Entity	The identification of people or organizations, which is the Author(s) to this resource.	
2.3.1.2	Last Updated		
2.3.2	Publisher	The identification of people or organizations, which is the Publisher to this resource.	
3	Technical	This category describes the technical requirements and characteristics of this resource.	
3.1	Main Format	Technical data type(s) of (all the components of) this resource. This element shall be used to identify the software needed to access the resource. NOTE: -- An HTML page with a GIF and a JPEG image would belong to Mixed.	Html/Audio/ CSS/ Windows Executable/ Mac Executable/Image/Microsoft Word/PDF/Postscript/SGML Text/Video/XML Text/Mixed/Other

3.2	Requirements	<p>This sub-category describes the technical capabilities required in order to use this resource.</p> <p>If there are multiple requirements, then all are required, i.e. the logical connector is AND.</p>	
3.2.1	Operating System		<p>Any Multi-OS PC-DOS MS-Windows 95/98/NT/2000 MacOS Unix</p>
3.2.2	Browser		<p>Any Netscape Communicator 3.X and Below Netscape Communicator 4.X and Above Internet Explorer 3.X and Below Internet Explorer 4.X and Above Opera Amaya</p>
3.3	Installation Remarks	<p>Description of how to install this resource.</p>	
3.4	Other Platform Requirements	<p>Information about other software and hardware requirements.</p>	
4	Educational	<p>This category describes the key educational or pedagogic characteristics of this resource.</p> <p>This is the pedagogical information essential to those involved in achieving a quality learning experience. The audience for this metadata includes teachers, and learners.</p>	

4.1	Interactivity Type	The flow of interaction between this resource and the intended user. In an expositive resource, the information flows mainly from this resource to the learner. Expositive documents are typically used for learning- by- reading. In an active resource, information also flows from the learner to this resource. Active documents are typically used for learning- by- doing. Activating links to navigate in hypertext documents is not considered as an information flow. Thus, hypertext documents are expositive.	Active Expositive Mixed Undefined
4.2	Learning Resource Type	Specific kind of resource, most dominant kind first.	Exercise Simulation Questionnaire Diagram Figure Graph Index Slide Table Narrative Text Exam Experiment ProblemStatement SelfAssesment
4.3	Interactivity Level	The degree of interactivity between the end user and this resource.	very low low medium high very high
4.4	Intended end user role	Principal user(s) for which this resource was designed, most dominant first. A learner works with a resource in order to learn something.	Teacher Learner

4.5	Context	The principal environment within which the learning and use of this resource is intended to take place.	Preschool Education Primary Education Secondary Education Higher Education University First Cycle University Second Cycle University Postgrade Technical School First Cycle Technical School Second Cycle Professional Formation Continuous Formation Vocational Training
4.6	Complexity Level /Difficulty	This element defines how hard it is to work through this resource for the typical target audience.	Easy Medium Advanced Professional
4.7			
4.7.1	Typical Learning Time	Unit of the time data.	Hour Week Month
4.7.2	Typical Learning Time Data	Approximate or typical time it takes to work with this resource	
4.8	Description/URL	Comments or URL of the comments on how this resource is to be used.	
5	Rights	This category describes the intellectual property rights and conditions of use for this resource. NOTE:--The intent is to reuse results of ongoing work in the Intellectual Property Right. This category currently provides the absolute minimum level of detail only.	
5.1	Cost	Whether use of this resource requires payment.	yes no
5.2	URL	URL of the Comments on the conditions of use of this resource.	-

6	Relation	<p>This category defines the relationship between this resource and other resources, if any.</p> <p>To define multiple relationships there may be multiple instances of this category. If there is more than one target resource, then each target is covered by a new relationship instance.</p>	-
6.1	Kind	<p>Nature of the relationship between this resource and the target resource, identified by <u>7.2:Relation.Resource</u>.</p>	<p>IsPartOf HasPart IsVersionOf HasVersion IsFormatOf HasFormat References IsReferencedBy IsBasedOn IsBasisFor Requires IsRequiredBy</p>
6.2	Resource	The target resource that this relationship references.	
6.2.1	Identifier	<p>Unique Identifier of the target resource. This is and shall not be used.</p>	
6.2.2	Description	Description of the target resource.	-
7	Classification	<p>This category describes where this learning object falls within a particular classification system. To define multiple classifications, there may be multiple instances of this category.</p>	
7.1	Taxon	This sub-category describes a particular term within a taxonomy.	The same as 1.6.1 Primary Keyword

Table 4.1 Definition of the "Metadata Standard"

4.3.2 Mapping LOM

The mapping details between LOM and the “Metadata Standard” used in this project are shown in Table 4.2.

<i>Our Metadata</i>	<i>LOM</i>
General	1: General
URL	1.1: General.Identifier when the globally unique label is a globally unique Web address (URL)
Title	1.2: General. Title
Language	1.4: General.Language, 3.5: Meta-Metadata.Language, 5.11: Educational.Language The Value space is: Chinese, Danish, English, French, Germany, Hindi, Italian, Japanese, Korean, Latin, Norwegian, Spanish, Swedish, other
Description	1.5: General.Description
Keywords	1.6: General.Keywords Including Primary Keyword, Secondary Keyword and Additional Keyword
Primary Keyword	1.6: General.Keywords, the Value space is The IRIS Keyword Thesaurus. (n.d.).
Secondary Keyword	1.6: General.Keywords, the Value space is The IRIS Keyword Thesaurus. (n.d.).
Additional Keyword	1.6: General.Keywords
Coverage	1.7: General.Coverage
Aggregation Level	1.9: General.Aggregation Level
LifeCycle	2: LifeCycle
Version	2.1: LifeCycle.Version
Status	2.2: LifeCycle.Status The Value space is: Draft Final Revised
Contribute	2.3: LifeCycle.Contribute Just including two roles: Author and Publisher
Author	2.3.1: LifeCycle.Contribute.Role = Author
Entity	2.3.2 LifeCycle.Contribute.Entity when 2.3.1:LifeCycle.Contribute.Role equals Author.
Last Updated	2.3.3: LifeCycle.Contribute.Date when 2.3.1:LifeCycle.Contribute.Role equals Author, and is about the last update time.
Publisher	2.3.2: LifeCycle.Contribute.Entity when 2.3.1:LifeCycle.Contribute.Role equals Publisher.
Technical	4: Technical

Main Format	4.1: Technical.Format The Value space is simple as: Html/Audio/ CSS/ Windows Executable/ Mac Executable/Image/Microsoft Word/PDF/Postscript/SGML Text/Video/XML Text/Mixed/Other
Requirements	4.4: Technical.Requirements
Operating System	4.4.2: Technical.Requirements.Name when 4.4.1: Technical.Requirements.Type= 'Operating System' And Value space is changed to: Any Multi-OS PC-DOS MS-Windows 95/98/NT/2000 MacOS Unix
Browser	4.4.2: Technical.Requirements.Name when 4.4.1: Technical.Requirements.Type= 'Browser ' And Value space is changed to: Any Netscape Communicator 3.X and Below Netscape Communicator 4.X and Above Internet Explorer 3.X and Below Internet Explorer 4.X and Above Opera Amaya
Installation Remarks	4.5: Technical.Installation Remarks
Other Platform Requirements	4.6: Technical.Other Platform Requirements
Educational	5: Educational The audience for this metadata includes teachers/educators, and learners.
Interactivity Type	5.1: Educational.Interactivity Type
Learning Resource Type	5.2: Educational.Learning Resource Type
Interactivity Level	5.3: Educational.Interactivity Level
Intended end user role	5.5: Educational.Intended end user role When Value space is: Teacher/Educator, Learner.
Context	5.6: Educational.Context when Value space is including the "Preschool Education"
Complexity Level /Difficulty	5.8: Educational.Difficulty And Value space is changed to: Easy Medium Advanced Professional
Typical Learning Time	Unit of the time data. Value space is: Hour Week Month
Typical Learning Time Data	5.9: Educational.Typical Learning Time
Description/URL	5.10: Educational.Description Comments or URL of the comments on how this resource is to be used.
Rights	6: Rights

Cost	6.1: Rights.Cost
URL	6.3: Rights.Description used to list the URL of the Comments on the conditions of use of this resource
Relation	7: Relation
Kind	7.1: Relation.Kind
Resource	7.2: Relation.Resource
Identifier	7.2.1: Relation.Resource.Identifier
Description	7.2.2: Relation.Resource.Description
Classification	9: Classification
Taxon	9.2.2.2: Classification.Taxon Path.Taxon.Entry The Value space is the same as 1.6.1 Primary Keyword.

Table 4.2 LOM Mapping

Although the “*Metadata Standard*” used in this project is almost a sub-set of LOM, a lot of its elements’ *Value space* is different from LOM’s mapping elements’ *Value space*. The main reason why our “*Metadata Standard*” does not directly use the LOM’s *Value space* is the particular demand of the system. For example, the providers of Web courses are not professional Web-course catalogers, like the catalogers of YAHOO, they, for instance Professor of history or physics, would not have any ideas about what is “MIME” type (Draft Standard for Learning Object Metadata. (2001).). Our “*Metadata Standard*” uses easily understandable value to be the *Value space*, for example, “Html/Audio/ CSS/ Windows Executable/ Mac Executable/Image/Microsoft Word/PDF/Postscript/SGML Text/Video/XML Text” is the *Value space* of the “3.1 Main Format” element (Table 4.1).

Like LOM, our “*Metadata Standard*” inherits from *Dublin Core Metadata Standard*. Their relationship is shown in Figure 4.1.

4.3.3 Characteristic of the “*Metadata Standard*”

Comparing with the other Metadata standards as discussing in chapter 2, the “*Metadata Standard*” is more adequate than the Dublin Core, GEM and EdNA, and is simpler than the LOM.

It is simple because, firstly, it is a subset of the most complicated Metadata standard, LOM standard; secondly, it gets rid of some cramped elements/value space of LOM (see previous examples).

The “Metadata Standard” is adequate. According to Table 2.3 “*Dublin Core and LOM relation*” and Table 4.2 “*LOM Mapping*”, it is easy to get the result that the “Metadata standard” includes the Dublin Core.

From section 2.5 “Comparison of various metadata standards”, we know that GEM and EdNA Metadata standards include Dublin Core. From the comparison of Table 4.2 “*LOM Mapping*”, Table 2.4 “*GEM mapping LOM*” and Table 2.5 “*EdNA mapping LOM*”, these is one clear conclusion that the “Metadata Standard” is a reasonable extensions of Dublin Core, GEM and EdNA. Table 4.3 shows the elements that are not included in Dublin Core, GEM and EdNA. For example, the “Status” element explains the life cycle status of the source, and the “Complexity Level /Difficulty” gives a clear statement what the resource’s complexity level is, for example, it is easy, medium, advanced or professional.

“Metadata Standard”	LOM
Aggregation Level	1.9: General.Aggregation Level
Status	2.2: LifeCycle.Status The Value space is: Draft Final Revised
Requirements	4.4: Technical.Requirements
Operating System	4.4.2: Technical.Requirements.Name when 4.4.1: Technical.Requirements.Type= ‘Operating System’ And Value space is changed to: Any Multi-OS PC-DOS MS-Windows 95/98/NT/2000 MacOS Unix

Browser	4.4.2: Technical.Requirements.Name when 4.4.1: Technical.Requirements.Type= 'Browser ' And Value space is changed to: Any Netscape Communicator 3.X and Below Netscape Communicator 4.X and Above Internet Explorer 3.X and Below Internet Explorer 4.X and Above Opera Amaya
Installation Remarks	4.5: Technical.Installation Remarks
Other Platform Requirements	4.6: Technical.Other Platform Requirements
Interactivity Type	5.1: Educational.Interactivity Type
Interactivity Level	5.3: Educational.Interactivity Level
Complexity Level /Difficulty	5.8: Educational.Difficulty And Value space is changed to: Easy Medium Advanced Professional
Typical Learning Time	Unit of the time data. Value space is: Hour Week Month
Typical Learning Time Data	5.9: Educational.Typical Learning Time
Description/URL	5.10: Educational.Description Comments or URL of the comments on how this resource is to be used.
Cost	6.1: Rights.Cost
URL	6.3: Rights.Description used to list the URL of the Comments on the conditions of use of this resource
Taxon	9.2.2.2: Classification.Taxon Path.Taxon.Entry The Value space is the same as 1.6.1 Primary Keyword.

Table 4.3 Some Elements of "Metadata Standard"

The next section (4.4) will show how the simple and adequate "Metadata Standard" is used in the "Web-course Search engine" to collect information, provide search criterion and return search results details.

4.4 Information Collection, Search criterion and search result

4.4.1 Information Collection

The system follows previously discussed “*Metadata Standard*” to construct the information-sheet, which is used to collect Metadata elements’ information of the Web courses from the providers. The information sheet is described in Figure 4.2.

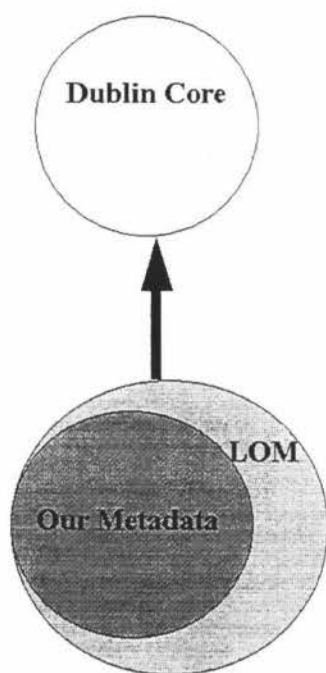


Figure 4.1 Relationships of Dublin Core, LOM, and “New Metadata Standard”

Create New Course Record:

*Title of the resource :(e.g. Online English Study)

*URL (Web address):

http://

*Language (Human language(s)used within this resource to communicate to the intended user)

Chinese

- Danish
- English
- French
- Germany
- Hindi
- Italian
- Japanese
- Korean
- Latin
- Norwegian
- Spanish
- Swedish

Other(s) :

Description (A short textual description of the content of this resource, 5-6 lines)

Keywords or Subject (where does this resource belong)

* Primary Keyword:

[Select one]

* Secondary Keyword:

[Select one]

Additional Keyword(s):

Coverage (The span or extent of such things as time, culture, geography or region that applies to this resource)(Ex. Circa, 16th century France):

Aggregation Level (The functional granularity of this resource)

0 = the smallest level of aggregation

(e.g. raw media data or fragments).

1 = a collection of atoms

(e.g. an HTML document with some embedded pictures; e.g. a lesson).

2 = a collection of level 1 resources

(e.g. a 'web' of HTML documents, with an index page that links the pages together; e.g. a course).

3 = the largest level of granularity

(e.g. a set of courses that lead to a certificate)

Life cycle of this resource:

*Version (The edition of this resource):

Last Updated: ,

*Status (The state or condition of this resource):

*Author(s):

*Author1: , Author 2:

Publisher:

Technical:

(This category describes the technical requirements and characteristics of this resource)

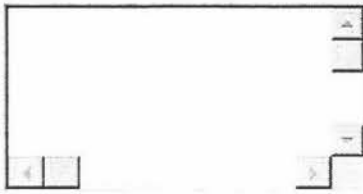
Main Format:

Requirement--Operating System:

Requirement--Browser:

Installation remarks:

Other software and hardware requirements:



Educational:

(Describes the key educational or pedagogic characteristics of this resource)

***Interactivity Type** (The flow of interaction between this resource and the intended user):

(Activating links to navigate in hypertext documents is not considered as an information flow. Thus, hypertext documents are expositive).

Active

(In an active resource, information also flows from the learner to this resource. Active documents are typically used for learning- by- doing.)

Expositive

(In an expositive resource, the information flows mainly from this resource to the learner. Expositive documents are typically used for learning- by- reading.)

Mixed

Undefined

Learning Resource Type: (Specific kind of resource)

Diagram

Exam

Exercise

Experiment

Figure

Graph

Index

NarrativeText

Problem Statement

Questionnaire

Self Assessment

Simulation

Slide

Table

Interactivity Level (The degree of interactivity between the end user and this resource) :

Intended end user role (Principal user(s) for which this resource was designed)

Teacher

Learner

Both

Context (The principal environment within which the learning and use of this resource is intended to take place):

Preschool Education

Primary Education

Secondary Education

Higher Education

University First Cycle

University Second Cycle

University postgraduate

Technical School First Cycle

Technical School Second Cycle

Professional Formation

Continuous Formation

Vocational Training

Complexity Level (Difficulty): [Select one] ▼

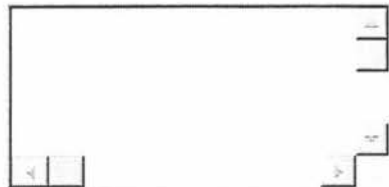
Typical Learning Time:

[] : Hours [] : Weeks [] : Months

Description or guide for using this resource :

URL: []

Description :(gives short description if you have provided URL)



Rights:

Cost: Yes No

Rights statement URL (if any): []

Relation(s):

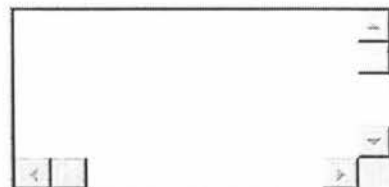
This category defines the relationship between this resource and other resources, if any.

This resource [Select one] ▼

Resource: []

URL: []

Description of the target resource:



Classification:

Taxon (The classification system is "The IRIS Keyword Thesaurus"):

[Select one] ▼

Figure 4.2 Information-sheet

4.4.2 Search criterion

In chapter 3, we discussed that the search criterion of some of the search systems is very simple and the search function is weak and less precise. For example, the “Education World” ‘s advanced search (http://www.education-world.com/search/adv_search.jhtml) options include just the Keywords, Title, Description, URL and Filter (The principal environment within which the learning and use of this resource is intended to take place).

The search function of the “*Web-course Search engine*” is more powerful and customized for learners’ requirements. The search criterion follows the “*Metadata Standard*”, which is a subset of *LOM*. The search parameters are subset of the “*Metadata Standard*”. They are:

1. Subject (or Taxon)
2. Keyword(s)
3. Language (Human language(s) used within this resource to communicate to the intended user)
4. Author
5. Publisher
6. Learning Resource Type
7. Intended end user role (Principal user(s) for which this resource was designed)
8. Context (The principal environment within which the learning and use of this resource is intended to take place)

9. Cost

The logical relationships between these search options are logical “AND”. Within each search option, for example among the “Keyword (s)”, the logical relationships are logical “OR”. The searchers can use all of these search options or just some of them. The search request sheet is shown in Figure 4.3.

Subject (or Taxon):

Keyword(s):

Or Keyword:

Or Keyword:

Language (Human language(s) used within this resource to communicate to the intended user)

Chinese (7 items)

Danish (1 items)

English (8 items)

French (5 items)

German (3 items)

Hindi (2 items)

Latin (1 items)

Spanish(1 items)

Swedish(1 items)

Author:

Publisher:

Learning Resource Type: (Specific kind of resource)

- Diagram
- Exam
- Exercise
- Experiment
- Figure
- Graph
- Index
- Narrative Text
- Problem Statement
- Questionnaire
- Self Assessment
- Simulation
- Slide
- Table

Intended end user role (Principal user(s) for which this resource was designed)

- Teacher
- Learner

Context (The principal environment within which the learning and use of this resource is intended to take place)

- Preschool Education
- Primary Education
- Secondary Education

- Higher Education
 - University First Cycle
 - University Second Cycle
 - University postgraduate
 - Technical School First Cycle
 - Technical School Second Cycle
 - Professional Formation
 - Continuous Formation
 - Vocational Training
- Cost: Yes No
-

Figure 4.3 Search request

4.4.3 Search result

It is very important that the system returns evaluative search results that are easy for the searchers to evaluate the suitability to their requirements.

However, most search systems fail to provide adequate search results. For example, the *EdNA Online* displays the search results in a very simple way, just showing the *Title* and *a short description* (section 3.2.3.4).

The search results of our “*Web-course Search engine*” provide all the information of

Web course following the “*Metadata Standard*”. The format and content on search results are same as the information-sheet. One search result example is shown in Figure 4.4.

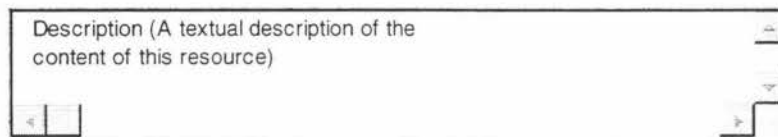
General:

Title of the resource: fifth

URL (Web address): <http://www.yahoo.com>

Language(s): Chinese, English

Description:



Description (A textual description of the content of this resource)

Coverage (The span or extent of such things as time, culture, geography or region that applies to this resource):

Coverage

Life cycle of this resource:

Version: 0

Last Updated: July/ 24, 2001

Status (The state or condition of this resource): Draft

Author(s): Xagong

Publisher: xagong

Technical:

(This category describes the technical requirements and characteristics of this resource)

Main Format: Html

Requirement--Operating System: MS-Windows 95/98/NT/2000

Requirement--Browser: Any

Installation remarks:



Installation remarks

Other software and hardware requirements:

Other software and hardware requirements

Educational:

(Describes the key educational or pedagogic characteristics of this resource)

Interactivity Type (The flow of interaction between this resource and the intended user):

Active

Learning Resource Type:

Diagram

Exercise

Figure

Questionnaire

Simulation

Interactivity Level (The degree of interactivity between the end user and this resource):

Very High

Intended end user role (Principal user(s) for which this resource was designed):

Teacher

Context (The principal environment within which the learning and use of this resource is intended to take place):

Preschool Education

Primary Education

Secondary Education Higher Education

University First Cycle

Complexity Level (Difficulty): Advanced

Typical Learning Time:

123 Hours

Description or guide for using this resource:

URL: [url](#)

Description :(gives short description if you have provided URL)

Description :

Rights:

Cost: Yes

Rights statement (URL): [Rights statement \(URL\)](#):

Relation(s):

This category defines the relationship between this resource and other resources, if any.

This resource is part of

Resource: first

URL: URL:

Description of the target resource:

Description (Description of the target resource)
--

Classification:

Taxon: Arts

Figure 4.4 Search result

After moderately checking the search results, it is very easy for the searchers to make informed decision: whether or not the search result is suitable to their requirements.

4.5 Database structure

4.5.1 Database structure underlying search mechanism

Database design needs a scheme to describe the conceptual or logical data structures of all the objects or entities with which the database is concerned (Strawman and Bretherton. (1994)), as well as the relationships between them (e.g., Sheth and Larson. (1990)). Strawman and Bretherton (1994) agree that within such a well defined and structured context, the difference between metadata and data disappears - metadata is simply data. The “*Web-course Search engine*” follows the “*Metadata Standard*” to construct the Database structure. There are altogether 14 tables.

These tables are:

1. *Account* table, which stores membership Web-course providers’ information. Its

Primary key “userid” is the Foreign key of Table *Course Details Part1* table.

2. *Course Details Part1* table, which stores all integrant information about the Web course and the single value information according to the search criterion. This arrangement could improve the system’s operation speed because the system just needs to search this table. Its Primary key “courseid” is the Foreign key of all the following Tables. The details that the table stores include the Metadata Elements: “1.1URL, 1.2Title, 1.5.1Primary Keyword, 1.5.2Secondary Keyword, 2.1Version, 2.2Status, 4.1Interactivity Type, 7.1Taxon, 2.3.2Publisher, 5.1Cost.”
3. *Course Details Part2* table, which stores that single short value information that is not integrant. The details that the table stores include the Metadata Elements: “1.6Coverage, 1.7Aggregation Level, 2.3.1.2Last Updated, 3.1Main Format, 3.2.1Operating System, 3.2.2Browser, 4.1Interactivity Type, 4.6Complexity Level /Difficulty, 4.7.1Typical Learning Time, 4.7.2Typical Learning Time Data, 4.8Description/URL, 5.2URL, 6.1Kind, 6.2.1Identifier, 6.2.2Description”.
4. *Author* table, which stores integrant information about author but would be many values. The details that the table stores include the Metadata Element “2.3.1.1Entity”.
5. *Context* table, which stores information about Context and there will be many values. The details that the table stores include the Metadata Element “4.5Context”.
6. *Intended End User Role* table, which stores information about "Intended End User Role “ but would be many values. The details that the table stores include the Metadata Element “4.4 Intended end user role”.
7. *Keyword* table, which stores Metadata Element “1.5.3 Additional Keyword ” and there will be many values.

8. *Language* table, which stores integrant information about Metadata Element “1.3Language” and there will be many values.
9. *Learning Resource Type* table, which stores Metadata Element “4.2Learning Resource Type” and there will be many values.
10. *Description Of This Resource* table, which stores Metadata Element “1.4 Description”.
11. *Installation remarks* table, which stores Metadata Element “3.3 Installation”.
12. *Other Requirements* table, which stores Metadata Element “3.4 Other Platform Requirements”.
13. *Related Resource Description* table, which stores Metadata Element “6.2.2 Description”.
14. *Using This Resource Description Text* table, which stores Metadata Element “4.8 Description/URL”.

4.5.2 Definition

The definitions of these tables are shown in Table 4.4.

<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>	<i>References Table</i>
Account	userid	varchar(80)	not null	primary key	
	email	varchar(80)	not null		
	firstname	varchar(80)	not null		
	lastname	varchar(80)	not null		
	homepageadd	varchar(80)	null		
	organization	varchar(80)	null		

<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>	<i>References Table</i>
Course Details Part1	courseid	varchar(80)	not null	primary key	
	userid	varchar(80)	not null	foreign key	Account
	Title	varchar(80)	not null		
	Identifier	varchar(80)	not null		
	KeywordsPrimary	varchar(80)	not null		
	KeywordsSecondary	varchar(80)	not null		
	Version	varchar(80)	not null		
	Status	varchar(80)	not null		
	InteractivityType	varchar(80)	not null		
	Taxon	varchar(80)	null		
	Publisher	varchar(80)	null		
	Cost	varchar(10)	null		
	<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>
Course Details Part2	courseid	varchar(80)	not null	primary key foreign key	Course Details Part1
	Coverage	varchar(100)	null		
	AggregationLevel	varchar(4)	null		
	LastUpdatedMonth	varchar(4)	null		
	LastUpdatedDay	varchar(4)	null		
	LastUpdatedYear	varchar(4)	null		
	Contributor1Name	varchar(80)	null		
	Contributor1Role	varchar(80)	null		
	Format	varchar(80)	null		
	OperatingSystem	varchar(80)	null		
	Browser	varchar(80)	null		
InteractivityLevel	varchar(80)	null			
<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>	<i>References Table</i>
Author	courseid	varchar(80)	not null	foreign key	Course Details Part1
	Author	varchar(80)	not null		

<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>	<i>References Table</i>
Context	courseid	varchar(80)	not null	foreign key	Course Details Part1
	Context	varchar(80)	not null		
<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>	<i>References Table</i>
Intended End User Role	courseid	varchar(80)	not null	foreign key	Course Details Part1
	IntendedEndUserRole	varchar(80)	not null		
<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>	<i>References Table</i>
Keyword	courseid	varchar(80)	not null	foreign key	Course Details Part1
	Keyword	varchar(80)	not null		
<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>	<i>References Table</i>
Language	courseid	varchar(80)	not null	foreign key	Course Details Part1
	WhichLanguage	varchar(80)	not null		
<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>	<i>References Table</i>
Learning Resource Type	courseid	varchar(80)	not null	foreign key	Course Details Part1
	LearningResourceType	varchar(80)	not null		
<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>	<i>References Table</i>
Description Of This Resource	courseid	varchar(80)	not null	foreign key	Course Details Part1
	DescriptionOfThisResource	varchar(256)	not null		
<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>	<i>References Table</i>
Installation remarks	courseid	varchar(80)	not null	foreign key	Course Details Part1
	Installationremarks	varchar(256)	not null		

<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>	<i>References Table</i>
Other Requirements	courseid	varchar(80)	not null	foreign key	Course Details Part1
	OtherRequirements	varchar(256)	not null		
<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>	<i>References Table</i>
Related Resource Description	courseid	varchar(80)	not null	foreign key	Course Details Part1
	RelatedResourceDescription	varchar(256)	not null		
<i>Table</i>	<i>Attribute</i>	<i>Data type</i>	<i>Null value</i>	<i>Constraint</i>	<i>References Table</i>
Using This Resource Description Text	courseid	varchar(80)	not null	foreign key	Course Details Part1
	UsingThisResourceDescriptionText	varchar(256)	not null,		

Table 4.4 Definitions of tables

4.5.3 Referential constraints

Foreign keys provide a way to enforce the referential integrity of a database. A foreign key is a column or group of columns within a table that references a key in some other table (or sometimes, though rarely, in the same table).

When inserting into or updating a table with an enabled foreign key constraint, the database server checks that the row does not violate the foreign key constraint by looking up the corresponding referenced key in the referenced table. If the constraint is not satisfied, the database server rejects the insert or update request.

When updating or deleting a row in a table with a referenced key (a primary or unique constraint referenced by a foreign key), the database server checks every foreign key

constraint that references the key to make sure that the removal or modification of the row does not cause a constraint violation. If removal or modification of the row would cause a constraint violation, the update or delete is not permitted.

4.5.4 Relationships between tables

A table with an enabled foreign key constraint depends on the referenced table with the corresponding referenced key (a primary or unique constraint referenced by a foreign key). The relationships between tables are shown in Figure 4.5.

4.6 Conclusion

This chapter discussed the Metadata Standard that is used in the “*Web-course Search engine*”, and described how this standard is applied in this system, for example, the database structure, and the expression of the Web-course information.

According to the section 3.5 of the last chapter (Chapter 3), to be a good Web-course search engine, the system should surely have the following features:

- First, the system’s Metadata Standard should be good enough to suit to the system needs or particular requests. Suitable Metadata Standard is the foundation of each such system.
- Second, the system should provide adequate search parameters for searchers to find resources.
- Finally, the system should return evaluative search results, which means that it should be easy for the searchers to evaluate the suitability of results to their requirements.

The discussion in this chapter makes it evident that the “*Web-course Search engine*” includes these features and therefore avoids the shortcomings of the other such systems.

Such system can be constructed in many ways. The next chapter will introduce the system’s implementation details, which include the architecture of the system and technologies used in the system.

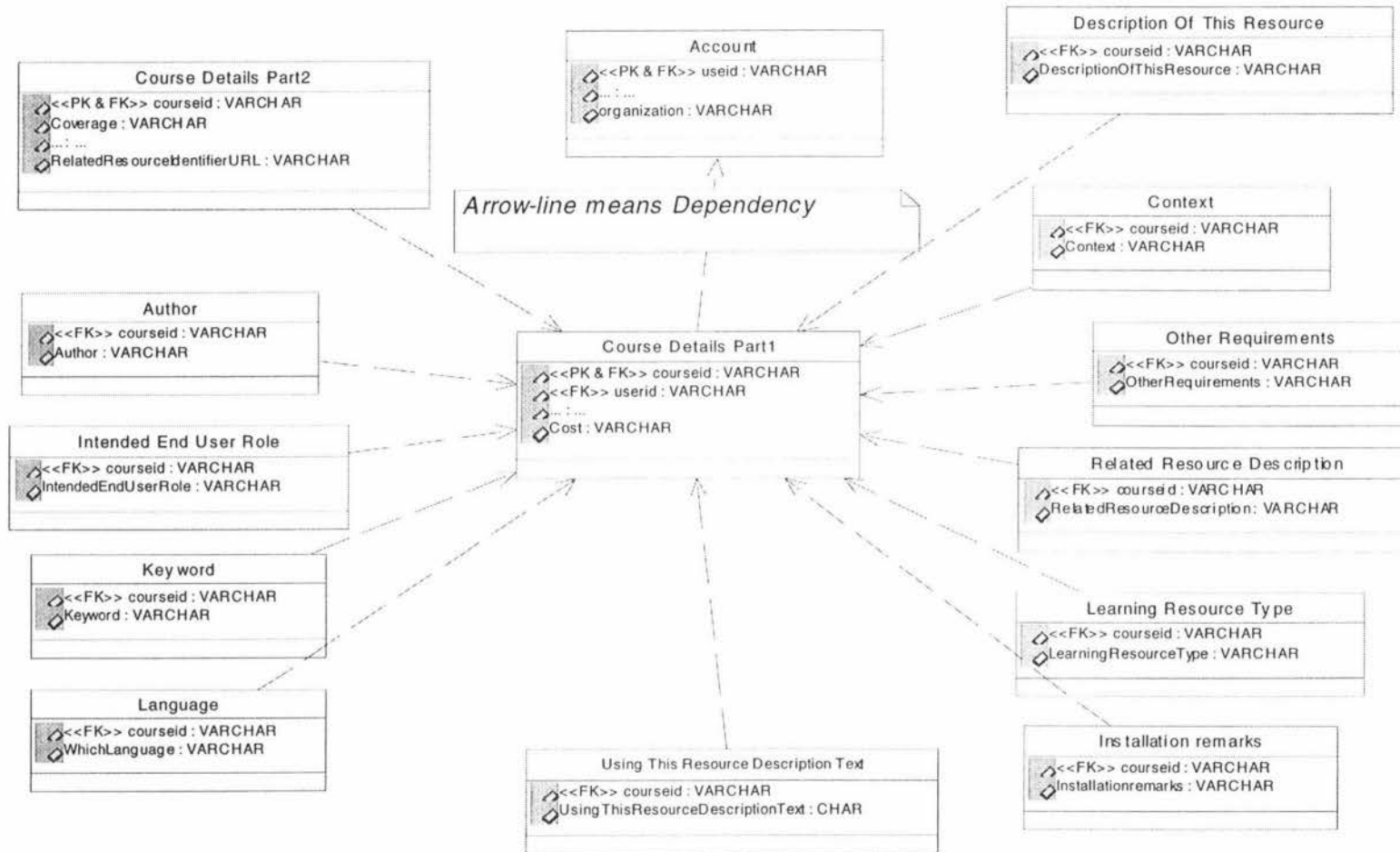


Figure 4.5 Relationships between tables

Chapter 5 Implementation of The *Web-course search engine*

This chapter provides a description of the *Web-course search engine*, a multi-tier Web application, and reviews the entire process of developing this application from specification to design to implementation.

This chapter will describe the architecture of the application, which conforms to the 3-tiered Web-based deployment and follows the Model-View-Controller design pattern (J2EE Design Patterns: Model-View-Controller (MVC) architecture. (2001).), and how the application uses the deployment, transaction, JNDI service, and security capabilities of the Java 2 Platform Enterprise Edition (J2EE) to simplify component development and provide richer functionality.

5.1 3-tiered Architectures

The “*Web-course Search engine*” is a Web-based application. Its architecture conforms the 3-tiered Web-based deployment (Figure 5.1). The presentation tier runs within the address space of one Web server (J2EE Web Container). It consists of Java Server Pages (JSPs), HTML scripts to customize look-and-feel, and workflow logic that ties things together. The business logic tier runs within the address space of one application server (J2EE EJB Container). The J2EE provides a suitable containment environment for the business logic components to run it. The J2EE manages these components efficiently and provides a number of services to the components. The application server provides a database access layer for the business components, allowing the business to persist data to and load from data tier. The J2EE is also responsible for making the business components available to be used, instantiating them as necessary. The data tier uses Cloudscape database Server to control and manage databases access.

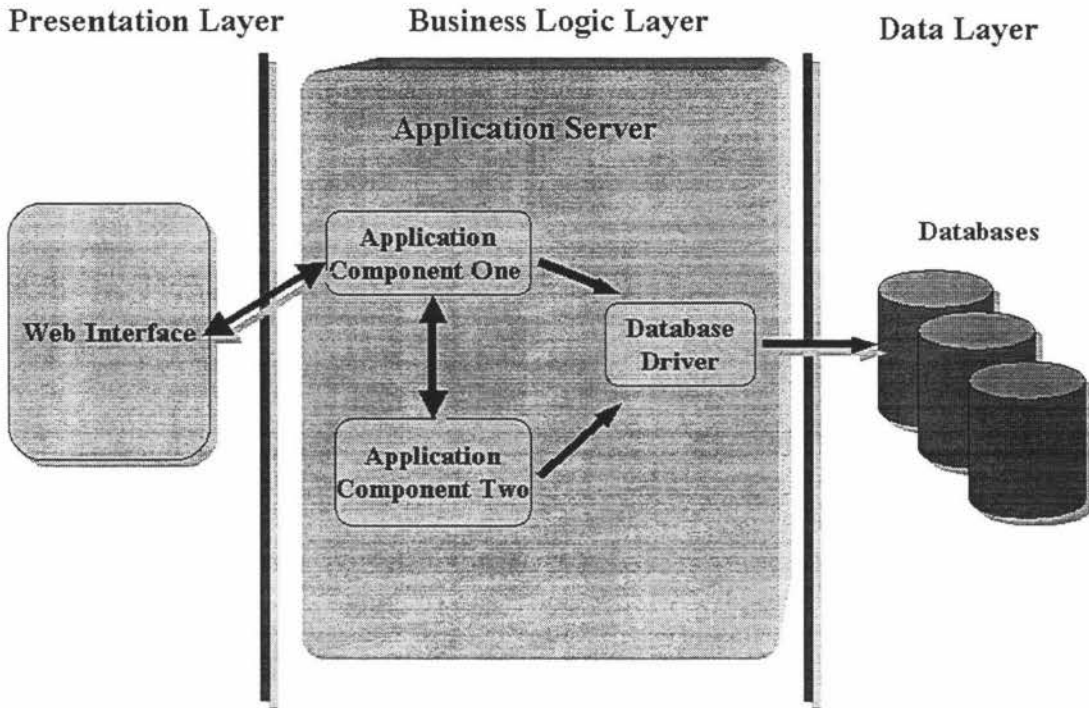


Figure 5.1 A 3-Tier Architecture

5.2 Functionality Specification

The application interface is presented to its users through a Web site and a user interacts with the application using a Web browser. The users of the application include Web-course providers responsible for maintaining and providing Web-course records, and Web-course searcher, the learner or teacher.

5.2.1 Scenarios

The application could support two different users' scenarios. First, there is the provider's scenarios that describes the Web-course providers to provide and maintain their Web-course records activities, also including the membership sign in or sign up. Second, there is a searcher's scenarios for carrying out searching Web-course activities. The scenarios in this section demonstrate all these types of interaction.

5.2.1.1 Scenarios of the Web-course provider

5.2.1.1.1 Applying Membership Scenario

Main flow 1:

Role: new provider

Connects to the application, by pointing the browser to the URL for the application's home page.

Selects the "Sign up" option.

Provides information to apply membership, including a user ID, user name, password and some other personal details.

Uses the new ID and the Password to login

Selects the options

Role: the system

Provides the home page which including for the provider member "Sign in", "Sign up" and for the searcher "Search" options

Provides the "Sign up" page.

Checks whether or not the user ID is already existed; if not, account is created with success; if yes, go to the *Sub-flow 1*.

Checks the ID and Password, if succeed, provides the "Create new Course Record" page.

Provides the user selected page

Sub-flow 1:

Role: new provider

Selects another user ID, and submit again.

Role: the system

Informs the user that the user ID is already existed.

Returns to the *Main flow 1* to check the user ID again.

5.2.1.1.2 Login Scenario

Main flow 2:

Role: membership provider

Enters the ID and Password to login

Selects the options

Role: the system

Recall information about the user.

Checks the ID and Password, if succeed, provides the “Create New Course Record”, “Modify Course” and “Delete Course” options’ page.

Provides the user selected page

5.2.1.1.3 Creating New Course Record Scenario

Main flow 3:

Role: member provider

Inputs course information and submit.

Role: the system

Provides the “Create New Course Record” page

Checks the information, if reasonable, stores the record to database and provides with success words; if not, go to the *Sub-flow 2*.

Sub-flow 2:

Role: member provider

Reviews and re-enter the data according to the comments, then submit again

Role: the system

Informs user the comments that mention unreasonable items.

Returns to the *Main flow 3* to check course data again.

5.2.1.1.4 Modifying Course Record Scenario

Main flow 4:

Role: member provider

Selects the course that needs to be modified.

Modifies the course data, then submits or abandon the action (*Sub-flow 3*).

Role: the system

Provides the “Modify Course Record” page, which lists all the courses’ records that the provider owns, for user to select.

Displays detailed information about the selected course to user.

Checks the information, if reasonable, stores the record to database and provides with success words; if not, go to the *Sub-flow 2*.

Sub-flow 3:

Role: member provider

Role: the system

Provides the “Create New Course Record”, “Modify Course” and “Delete Course” options’ page.

5.2.1.1.5 Deleting Course Record Scenario

Main flow 5:

Role: member provider

Selects the course that needs to be deleted.

Role: the system

Provides the “Delete Course Record” page, which lists all the courses’ records that the provider owns, for user to select.

Displays detailed information about the selected course and asks user to confirm

Confirms deletion or abandon deletion (<i>Sub-flow 3</i>).	or abandon deletion. Deletes the course record in the database and provides success words.
---	---

5.2.1.2 Scenario of the Web-course searcher

Search Scenario:

Role: searcher

Selects or enters course search request(s).

Selects one "Title" to view

Selects "BACK" or go to the Web-course.

Role: the system

Provides the "Courses search request sheet" page.

According to the searcher's request(s), finds the courses that match the request, then lists all the results with "Title"

Displays the selected course detailed information that has a link to the Web-course, and also provide the "BACK" link to the search results page.

If "BACK" link, returns to the search results page.

5.2.2 Functional Specification

Upon arriving at the pages of the Web-course search engine, a user (provider or searcher) would expect some of the following features:

- A *home page view* that the user can select the "Provider Sign in", "Provider Sign up", or "Search courses" options.
- A set of links or navigation bars on each page that provide quick access to common navigational tasks, for example, the "Help" link.
- An *organized view* of the site's contents through categorized options.

- A search mechanism to provide a way to locate courses based on search criteria, and then renders the search results to the searcher.
- A sub-function of search mechanism to provide a way to recall the searcher latest search criteria.
- A *master view* of courses that lists all courses of interest. This could be the result of the searcher searching through a search criteria or the aggregate of the provider owns courses by authenticating different provider.
- A *detail view* that describes the details of a particular course for the searchers. Searchers click on a course in the courses list to zoom in on details, including a description, a link to the Web courser's URL, and so on.
- In addition, the course *detail views* for course provider to add, modify, and delete course.
- A *modifying/deleting confirmation view* that lets provider review the contents of the course record before confirms "modify"/"delete". This allows the provider to fulfill "modify"/"delete" action or cancel "modify"/"delete" action.
- A *comment feedback view* that feeds back the unsuitable items that the provider inputs by verifying the rationality of the datum when the provider creates or modifies a course record, and applies the membership.

5.3 Structure

5.3.1 Model-View-Controller design pattern (MVC)

The 3-tiered Web-based deployment of the "*Web-course Search engine*" follows *Model-View-Controller* design pattern. These include objects that deal with *presentation* aspects of the application, objects that deal with the *business* rules and data, and objects that accept and interpret user requests and *control* the business objects to fulfill these request.

The *Model* represents the data information and its transaction methods and rules.

The *View*, of course, is the users' interface presentations that are different to different user, different stages.

The *Controller* is to control, maintain, and harmonize the *View* and *Model* work correctly and consistently.

The look and feel of the application interface changes often, its behavior changes less frequently, and business data and rules are relatively stable. Thus objects responsible for control are often more stable than presentation objects while business rules and data are generally the most stable of all.

The *MVC architecture* can be used in a multi-tier application straightforward. In the "*Web-course Search engine*", the Web tier or the client can handle the presentation logic of a user interface. In the Web tier, JSP pages are used to dynamically generate HTML for consumption by a browser. Control-related objects are present in each tier to enable coordination of actions across tiers. Objects that model business data and rules live in the EJB tier in an EJB-centric approach, and in the Web tier when using a Web-centric approach.

5.3.2 EJB-centric and Web-centric design

The J2EE platform is designed for multi-tier applications, and offers a lot of flexibility in choosing how to distribute application functionality across the tiers.

In the J2EE platform, an EJB-centric design, in which enterprise beans running on EJB servers encapsulate the enterprise information system resources and the core application logic, moves most of the core application functionality to the EJB tier, using the Web tier only as a front end for receiving client Web requests and for presenting HTML responses to the client. Enterprise beans have access to a broad set of enterprise-level services so that managing transaction and security aspects of the application is easier.

In a Web-centric design in the J2EE platform, the Web tier communicates directly with the enterprise information system resources that hold application data. The Web-centric approach is better for getting the application off to a quick start and run fast.

The solution of the Web-course search engine is an approach that benefits from the strengths of both approaches (Figure 5.2). In the Web-course search engine application, while all the functions, which need security protection, role authentication and data transaction, related with provider are implemented with an EJB-centric design, the search function that is needed to be implemented speedily uses the Web-centric model.

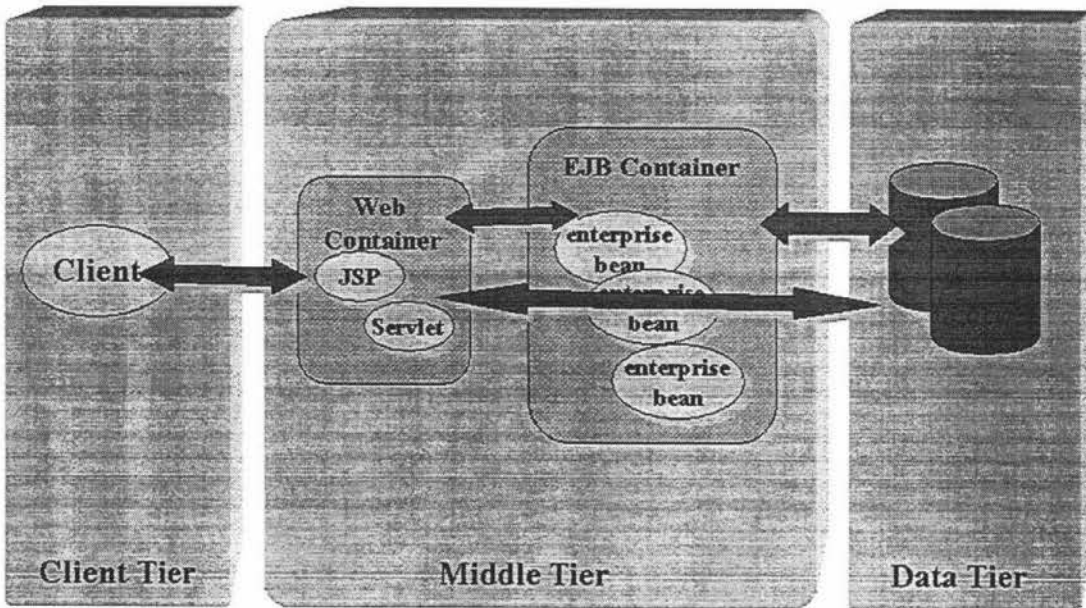


Figure 5.2 Mixed EJB-centric and Web-centric design

5.4 The Model of Model-View-Controller

This section explains how the We-course search engine application maintains state in the J2EE platform and persistent data in database tables.

5.4.1 State in the Web-course search engine

When the user visits the Web-course search engine, the application will maintain the user requests and status during the session. While the application does not need to store this information in a database, this information must be somehow tracked to maintain a meaningful dialog between the user and the application.

The J2EE platform provides several choices for storing the application state. An application can store state in the Web tier using the state maintenance capabilities of servlets, which include the *HTTPSession* and *ServletContext* objects as well as Java Beans components. The Java Beans components contain copies of the state maintained by corresponding model objects, which are maintained in the EJB tier. In the EJB tier, state can be maintained using enterprise beans. In addition, session state for an application can be divided between these tiers.

The application must maintain the following state:

- The user identity: Typically, the user *account model* maintains the user identify, which includes the user's login ID and certain security credentials.
- The search results: The *catalog model* maintains the results within the current search.
- The latest search request: The *HTTPSession* and *ServletContext* objects maintain the request.

5.4.1.1 Using Enterprise Beans to Maintain Session State

The application uses enterprise bean to provide an object view of individual rows in a database. The application uses "Account" entity bean to represent individual rows in the corresponding table, "account", and uses the "Catalog" stateful session bean to access all the tables related to courses except the "account" table. These tables are discussed in Section 4.5.

5.4.1.2 Data Access Objects and Fine-grained objects

Since enterprise beans are remote objects that consume a significant amount of system resources and network bandwidth, it is not appropriate to model all business objects as enterprise beans. Therefore, the application uses some objects that are subordinate to their respective enterprise beans for a number of purposes. The different types of these objects in the application are: data access objects, which encapsulate database access, and fine-grained objects that are dependent on enterprise beans.

Data Access Objects

The Intent of Data Access Objects is to “decouple business logic from data access logic and adapt the resource being accessed, so that the type of resource can change easily and independently.” (J2EE Design Patterns: Data Access Object (DAO). (2001).).

The details of any database access are encapsulated into the *data access objects* (DAO). In a J2EE application environment, this can make the EJB code more clear and simple. It is easy to update database or change different database vendors by using *data access objects*.

Data access objects can encapsulate access to more than one database and more than one table within one database. The application uses the abstract data access class *CatalogDAO* to access all the course related tables when a course is created, read, or updated, and uses the *AccountDAO* to access the “account” table when a account is create and read.

Fine-grained objects

A fine-grained object is a business object that can be passed by value as a *serializable* Java object. A business concept should be implemented as a value object when it is:

- Fine-grained, which means it only contains methods to get the values of fields.
- Dependent, which means its life cycle is completely controlled by another object.

- Immutable, which means that its fields are not independently modifiable.

A user's request for a fine-grained object can be fulfilled by the server more simply than for an enterprise bean; the object is serialized and sent over the network to the client where the object is *deserialized*. Such object allows a user to retrieve the value of a remote object in one remote call. The object can then be used as a local object. This conserves system resources by reducing the load on a remote object. It also reduces network traffic, as the method calls to get fields of the object are all local.

The application contains fine-grained objects representing details for each enterprise bean. An account entity bean's corresponding fine-grained object is *AccountModel*, which is a Java Bean and includes all the details of a provider's account (Figure 5.3). The Java Bean *CourseEvent* model represents all information of a course details.

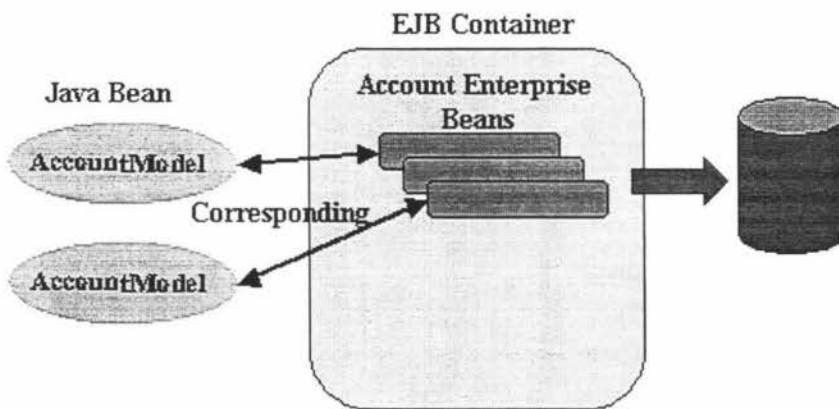


Figure 5.3 Fine-grained Objects

5.4.2 Persistent Data

The application maintains persistent data in database tables, organized according to the functional areas of the application. Figure 4.5 illustrates the database schema.

The application uses this database schema to maintain accounts and courses. There are two areas, for which data must be maintained: account and course information.

5.5 The View of Model-View-Controller

The view determines the presentation of the user interface of the application. In the application, the implementation of the view is contained completely in the Web tier. In the application, three kinds of components work together to implement the view: JSP pages, JSP custom tags, and Java Beans components.

JSP pages are used for dynamic generation of HTML responses. Custom tags make it easier for JSP pages to use Java Beans components when the underlying model is complex. Custom tags can also help encapsulate presentation logic and make it modular and more reusable.

Java Beans components represent the contract between JSP pages and the model. JSP pages rely on these beans to read model data to be rendered to HTML, while elsewhere in the system, the model and controller coordinate to keep the Java Beans components up to date.

5.5.1 Interaction Interface

The scenarios described in Section 5.2.1 provide a behavioral specification for the application interaction interface. This section translates this specification into the set of views that the users (provider and searcher) see when interacting with the Web-course search engine.

The interface for the application interaction is composed by a set of screens, which are saw by the users when they navigate to one of the application's URLs. A screen can be composed of several components each contributing a different part of its content.

These screens are:

- Name: PROVIDER_MAIN_SCREEN

This is the home page for the provider. If sign in success, this screen shows the “Add Course”, “Modify Course” and “Delete Course” options; otherwise shows the “Please sign in first” information.

□ Name: FORM_BASED_LOGIN_SCREEN

This screen displays a provider name and password, allowing the provider to sign into the application. The submit button initiates the sign in process.

□ Name: CREATE_NEW_ACCOUNT_SCREEN

This screen displays a form allowing new providers to sign up and register themselves with the application. Once registered, the provider can conveniently recall personal information each time.

□ Name: ACCOUNT_CREATION_SUCCESS_SCREEN

This screen is displayed after a new account is created.

□ Name: ADD_COURSE_SCREEN

This screen displays a form for the provider to input a new course's details.

□ Name: COURSE_MODIFY_SCREEN

This screen displays a master view of all courses that belong to a particular provider. The provider can click on the Title of any course on display to modify details of the course.

□ Name: COURSE_UPDATE_SCREEN

This screen displays the details of a course record to be modified. After modifying action, the provider could confirm this change or cancel this action.

□ Name: COURSE_DELETE_SCREEN

This screen displays a master view of all courses that belong to a particular provider. The provider can click on the Title of any course on display to delete the course.

□ Name: COURSE_SUBMIT_DELETE_SCREEN

This screen displays the main details of the course record to be deleted. The provider could confirm this deletion or cancel this action.

□ Name: SEARCH_MAIN_SCREEN

This screen displays a form where the searcher can fill in details necessary to place the search. A searcher places the search by clicking the submit button.

□ Name: COURSE_SEARCH_RESULT_SCREEN

This screen displays a master view of the results of a search. The searcher can click on the Title of any course on display to the details of the course.

□ Name: DISPLAY_COURSE_DETAILS_SCREEN

This screen displays the information about a particular search result course.

□ Name: LOGOUT_SCREEN

After provider logouts successful, the application displays this screen.

□ Name: HELP_SCREEN

The screen displays help information to user.

□ Name: MISSING_FORM_DATA_SCREEN

The screen displays the information about the missing data that the user should input.

□ Name: DUPLICATE_ACCOUNT_SCREEN

The screen tells the new provider that the account ID, which he selected, is already existed.

□ Name: DUPLICATE_COURSE_SCREEN

The screen tells the provider that the course, which he wanted to create, is already existed.

□ Name: NO_EXIST_COURSE_SCREEN

The screen informs user that the selected course is not exist.

5.5.2 Screen's structure

The application uses JSP pages to construct all the screens. The JSP pages provided by the application use a generic template mechanism and application-specific Java Beans components. The using template mechanism is the main benefit of JSP technology. This section describes the template mechanism and discusses several example JSP pages.

Template

The template mechanism (Figure 5.4) using in the application includes some of elements. Some of these are:

- Title of the screen.
- The application logo and tag line.
- A home page link to home page.
- A help button to get help information about the application.
- A signin/signout status button that changes state based on whether the provider is signed in. If they are signed in, they are presented with a sign out button. If they are not signed in, they see a sign in button.
- The main body.
- Some status information at the bottom of each page.

Among the elements that change on each page are the body and the title. A template mechanism provides a way to separate the common elements that are part of each screen from the elements that change with each screen. Putting all the common elements together into one file makes it easier to maintain and enforce a consistent look and feel in all the screens. It also makes development of individual screens easier since the designer can focus on portions of a screen that are specific to that screen while the template takes care of the rest. The template file is a JSP page, called *template.jsp*.

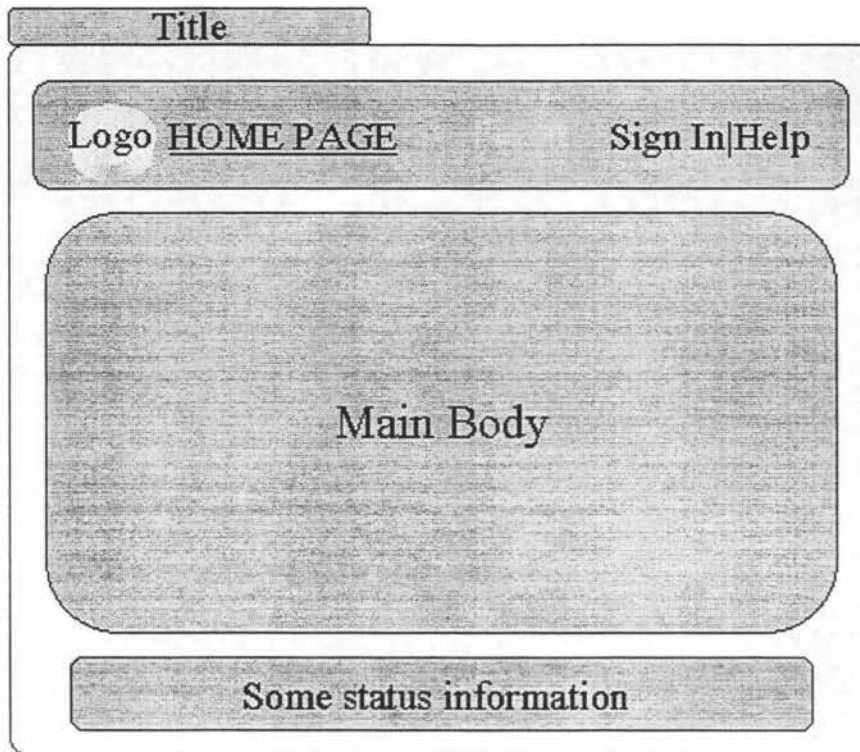


Figure 5.4 A Template

5.5.3 Model-View Synchronization

5.5.3.1 Account Model related Screens

In the implementation of the view, JSP pages rely on Java Bean components to mirror model data. Every view screens, which the course providers interact with, rely on a Java Bean *AccountWebImpl* to reflect the user login state. When a Java Bean *AccountWebImpl* is created (Figure 5.5), it adds itself to the list of listeners interested in updates to the account model. When an account model changes, the manager of the view objects invokes the *performUpdate* method on all views that have registered as listeners of the account model.

Enterprise beans named *Account* implement the model. Data access classes named *AccountDAO* and details classes named *AccountModel* support these beans. A user can retrieve the contents of an enterprise bean with one remote call that returns a details object.

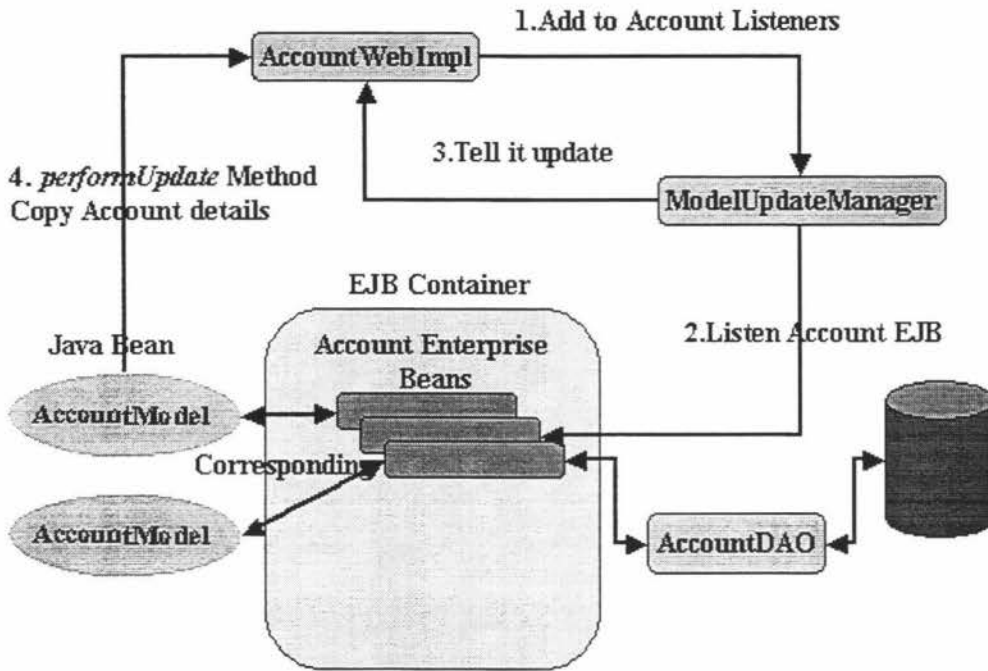


Figure 5.5 Model-View Synchronization

Java Bean component (*AccountWebImpl*) and details classes (*AccountModel*) share aspects of their implementation (that is, the *AccountModel* classes), because the *AccountModel* classes capture the essential information required to represent the application objects in any tier.

5.5.3.2 Catalog Model Related Screens

As mentions in Section 5.3.2 *EJB-centric and Web-centric design*, in the application, while all the functions, which need security protection, role authentication, and data transaction, related with provider are implemented with an EJB-centric design, the search function that is needed to implement speedily uses the Web-centric model. The implementation of the master views of course catalog does not follow the pattern just described because it implemented in both a Web-centric and EJB-centric fashion.

The Web-centric design is used for high performance since the course catalog for search function is read-only and the most frequently accessed object in the system. Thus, the Web-centric Java Bean component *CatalogModel* accesses the data access class *CatalogDAO* directly instead of calling an enterprise bean (Figure 5.6).

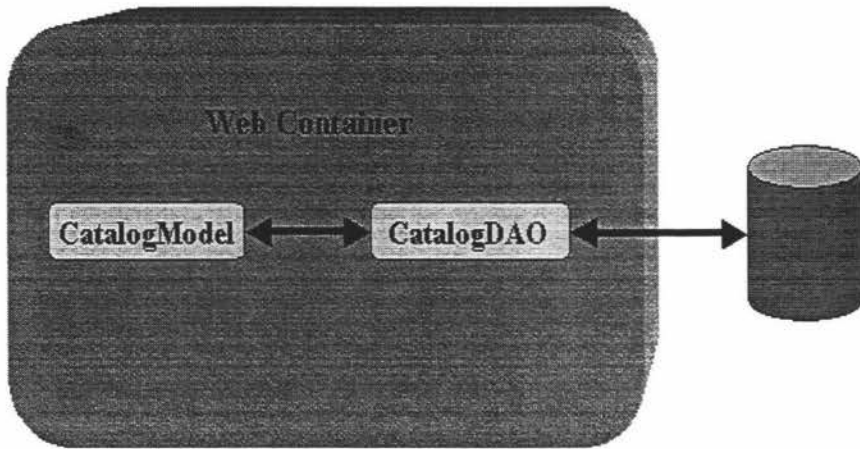


Figure 5.6 CatalogWebImpl accesses Data

The course catalog views of the functions, which need security protection, role authentication and data transaction, are implemented in EJB-centric fashion. Note that high performance is not as crucial in this case as compared to the earlier case. Enterprise beans named *CatalogEJB* implement the model. Data access classes named *CatalogDAO* support these beans.

5.6 The Controller of Model-View-Controller

From the Section 5.3.2 *EJB-centric and Web-centric design*, we know that the application uses both the EJB-centric and Web-centric design, so there are two kinds of control fashion in the system; they are EJB-centric controller and Web-centric controller.

5.6.1 EJB-centric controller

For fulfilling the provider's functions, which are constructed under EJB-centric fashion, the controller is made up of many components responsible for taking data posted in an HTTP request and converting it into an event to update the model data. The components that make up the controller include: front component, request processor, Web controller, and EJB controller.

Details of these components:

□ Front components

The front component, *main.jsp for provider functions*, is a component to which all requests for application URLs are delivered. The front component ensures that the Web components needed by the application are initialized at the correct time and that all HTTP requests are sent to the request processor.

□ Request processor

The request processor is the link between the Web application and an HTTP-based client. The request processor is responsible for converting HTTP requests to events, which will be used throughout the application. This component allows the application developer to centralize all HTTP-specific processing in one location. This component also allows the EJB portion of the application to be independent of any single client type.

□ Web controller

The Web controller is responsible for forwarding the event(s) generated by the request processor component to the EJB controller. The Web controller ensures that the resulting updated models returned from the EJB controller are propagated to the appropriate Web-tier view Java Beans components.

□ EJB controller

The EJB controller accepts events from the Web controller and makes the calls on the enterprise beans affected by the event. The EJB controller is also responsible for maintaining the state of the user session with the application. After each event is processed, the EJB controller is responsible for returning a set of updated models to the Web controller.

All requests from HTTP clients go to a front component. The requests are then sent to the request processor, which converts them to events and then sends the events to the Web controller. The Web controller acts as a proxy and sends the event to the EJB

controller, which processes the event and updates the model data maintained by the enterprise beans accordingly.

The EJB controller and enterprise beans handle all business logic. The EJB controller returns a set of changed models to the Web controller. The Web controller then sends the model update events to the respective views. The views then contact the enterprise beans that they mirror and update their data from the enterprise beans. The JavaBeans components do not change any data; they only read the model data contained by the enterprise beans when they receive the model update notification.

The following sections will discuss the implementation of each of these components in more detail.

5.6.1.1 *Main.jsp*

A front component is a component to which all requests for application URLs are delivered. The front component *Main.jsp* processes these requests and delegates the generation of the response to the template page. *Main.jsp* delegates all request processing tasks to *RequestProcessor*. The response is generated by forwarding to *template.jsp*. *Main.jsp* stores references to the request processor and other session-specific beans in the HTTP session object.

5.6.1.2 *RequestProcessor*

RequestProcessor contains logic that is executed for each request. For example, when a user tries to access a feature that requires sign in, *RequestProcessor* checks to detect whether the user is logged in.

The core responsibilities of *RequestProcessor* including:

- Initializing the client session. *RequestProcessor* instantiates an object that implements *EJB controller* and related application objects when a new session is initiated.

- Detecting when the user logs into the server using form-based authentication and generating a login event when this happens.
- Computing the event to generate based on the `HttpRequest` that came in.
- Raising an event by invoking the “*handle Event*” method on the EJB controller’s Web implementation.
- Gathering the outcome of the event processing, so the model change notifications can be processed by the view components.

5.6.1.3 Web controller

The *Web controller* is a proxy object that calls methods on the EJB tier controller, *EJB controller*. *Web controller* exposes a read-only interface to the model, so that the view can render the model as needed. Keeping this interface read-only minimizes dependencies between the view and the model, to prevent inadvertent modification of the model by the view outside the scope of the business rules encapsulated in the application.

All the methods of *EJB controller* are synchronized so that concurrent requests to *EJB controller* are serialized. This is done because an EJB container will throw an exception if a request is made to a session bean while it is servicing another request.

5.6.1.4 EJB Controller

The *EJB Controller* manages the life cycle of model objects such as the account enterprise beans and processes events. *EJB Controller* implements the core command processing business logic of the application. It is responsible for changing the state of the models in response to an event or command. *EJB Controller* consists of methods that handle each of the different events that the application can respond to. *EJB Controller* has both read and write access to all of the model objects so that it can coordinate event processing across multiple model objects.

5.6.1.5 Synchronization

In the MVC architecture of the “*Web-course Search engine*”, views implemented by JSP pages and Java Bean components present data owned by their associated models implemented as enterprise beans. Each Web-tier Java Bean component serves as the view, with corresponding EJB-tier classes representing the model. Whenever a model changes, it notifies interested views so that the views can update its presentation of the model (Figure 5.7). “*Model Update Manager*” and “*Model Manager*” manage the notification process. “*Model Update Manager*” is responsible for converting an event to a list of names of models that have changed due to this event. “*Model Manager*” uses this list to notify all views that have registered interest in the changed models to fetch the models' data.

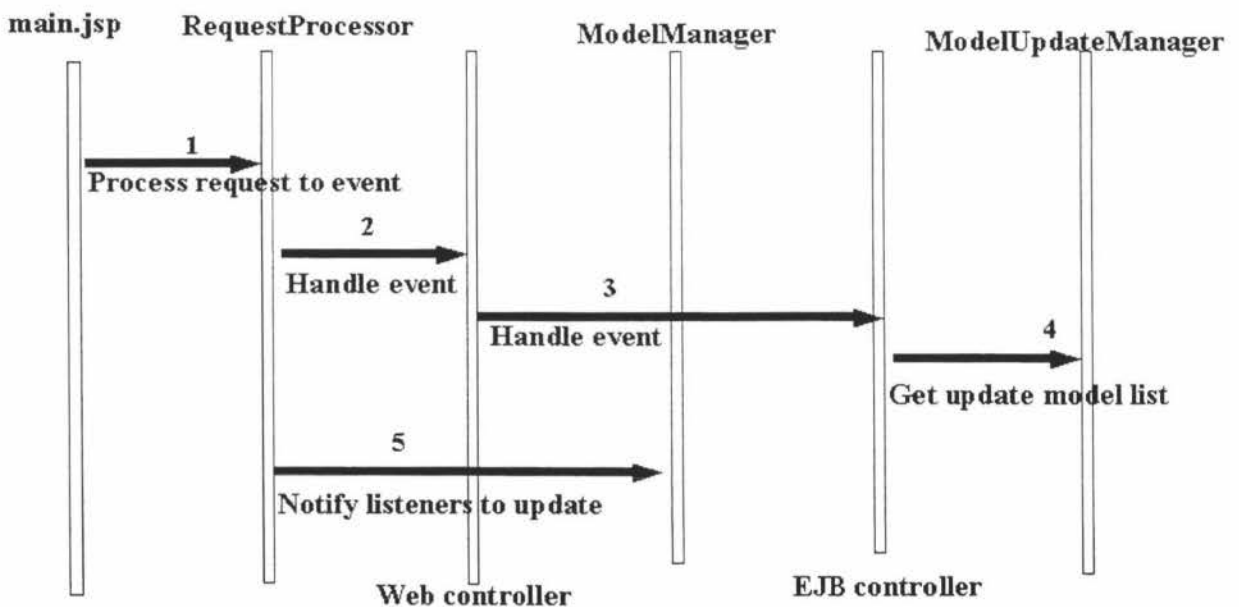


Figure 5.7 EJB-centric controlling process

5.6.2 Web-centric controller

For fulfilling the searcher's functions, which are constructed under Web-centric fashion, the components that make up the controller just include: front components and Web controller.

Details of these components:

- Front components

The *SearchMain.jsp* receives and forwards the initial request to a template, *SearchTample.jsp*. The template includes *SearchScreenDefinitions.jsp*, which uses *SearchScreenFlowManager*, which is responsible for selecting the next screen to be shown to the client after the completion of the current request, to map the screen to a JSP page.

- Web controller

The Web controller is responsible for finding the suitable search results according to the search request generated from the *SearchMain.jsp*, and returning the details of the course searcher selected.

Figure 5.8 shows how these components cooperate with each other.

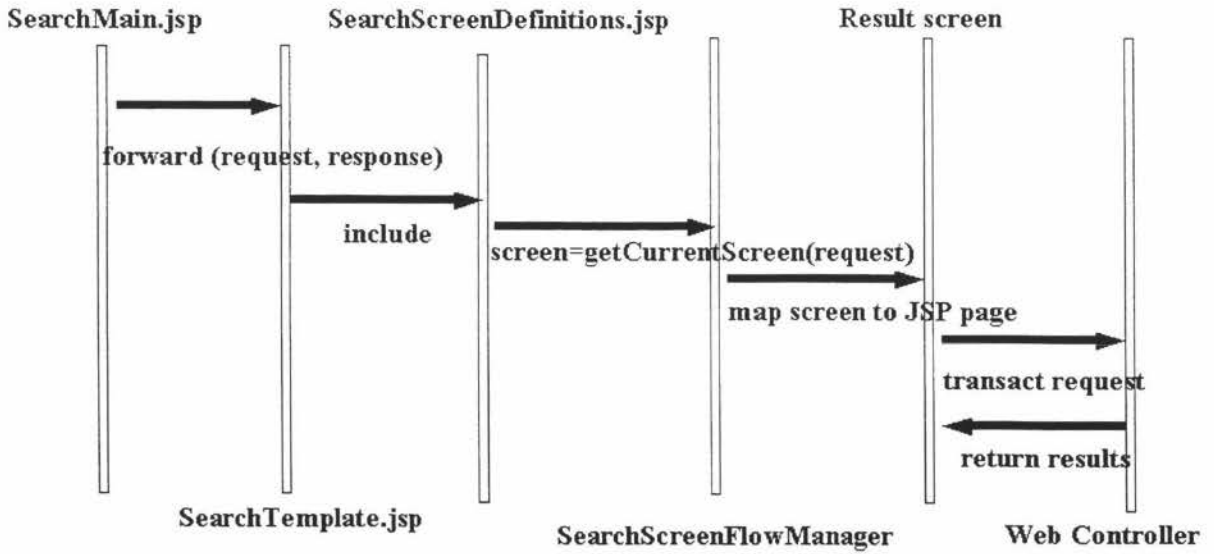


Figure 5.8 Web-centric controlling process

5.7 Deployment

The process of installing and customizing an application in an operational environment is called deployment. To enable customization, the components of an application need to be configurable. The deployment of a J2EE application involves three different types of components: enterprise beans, Web components, and application clients. The application includes two kinds of components: enterprise beans and Web components.

5.7.1 Deployment Descriptors

A *deployment descriptor* is an XML-based text file, which describes how to assemble and deploy the unit into a specific environment. Deployment descriptor elements contain behavioral information about components not included directly in code. Their purpose is to tell the *Deployer* how to deploy an application, not tell the server how to manage components at runtime.

There are different types of deployment descriptors: EJB deployment descriptor described in the Enterprise Java Beans specification, Web deployment descriptor described in the Servlet or JSP specification.

Deployment descriptors specify structural information describes the different components of the JAR or WAR file, their relationship with each other, and their external dependencies.

The Common Elements of deployment descriptor include environment entries, references to enterprise beans, references to connection factories, and security-related elements.

5.7.2 Packaging the Application

The J2EE application, Web-course search engine, is packaged as an Enterprise ARchive (EAR) file—“*WebCourseSearchEngine.ear*”, a standard Java JAR file with

an .ear extension. The goal of this file format is to provide an application deployment unit that is assured of being portable.

The application file contains three J2EE modules, which are independently deployable units, and a J2EE application deployment descriptor. Creation of the J2EE application is a two- step process. First is to create the J2EE modules: EJB, Web. Second is to package these modules together to create the J2EE application.

5.7.2.1 EJB Module

An EJB module is the smallest deployable and usable unit of enterprise beans. There is an EJB Module in the application. The EJB module is packaged and deployed as an EJB JAR file—“*EjbTier.jar*”, a JAR file with a .jar extension. It contains:

- Java class files for the enterprise beans, which include “*TheAccount*” Entity Bean, “*TheCatalog*” Stateful Session Bean, and “*TheController*” Stateful Session Bean, and their remote and home interfaces.
- Java class files for any classes and interfaces that the enterprise bean code depends on that are not included with the J2EE platform. This may include super classes and super interfaces and the classes and interfaces used as method parameters, results, and exceptions.
- An EJB deployment descriptor that is a XML file, which provides both the structural and application assembly information for the enterprise beans in the EJB module.

5.7.2.2 Web Modules

A Web module is the smallest deployable and usable unit of Web resources. A Web module is packaged and deployed as a Web Archive (WAR) file, a JAR file with a .war extension. Since a WAR file is typically deployed under its own context root, cross-linked Web pages must be packaged in a single Web module to avoid broken links. Moreover, cross-linked HTML Web pages are typically reusable as a bundle, so

it makes sense to package them together. There are two Web modules in the application. They are “*WebTier*” Web module and “*SearchCourse*” Web module.

A Web module normally contains:

- JSP pages and their helper Java classes (for example, the *SearchMain.jsp*, and *SearchScreenFlowManager.java* (Figure 5.8))
- Static documents (for example, HTML, images, and so on)
- A Web deployment descriptor

5.8 Transactions & Security

In the presence of J2EE platform support and using the capabilities of it, the application relies on the Web and EJB containers to handle transactions, security, and scalability related to enterprise information system access. The task of accessing enterprise information system resources from the application code is made even easier with enterprise application development tools.

5.8.1 Transactions

The most important concept when dealing with enterprise applications is the concept of a *transaction*. Transactions are a mechanism for simplifying the development of distributed multi-user enterprise applications. To a user, a transaction is a single change event that either happens or doesn't happen. To system implementers, a transaction is a programming style that allows them to code modules that can participate in distributed computations. Transactions provide a way to bundle a set of operations into an atomic execution unit.

Transactions are essential for distributed applications. Further, transactions provide modular execution, which complements a component technology's modular programming.

In J2EE environment, support for transactions is an essential element of the J2EE architecture. The J2EE platform supports both programmatic and declarative transaction demarcation. Declarative transaction demarcation is supported in enterprise beans, where transactions are started and completed automatically by the enterprise bean's container.

The J2EE server implements the necessary low-level transaction protocols, such as interactions between transaction manager and JDBC database systems, transaction context propagation, and optionally distributed two-phase commit. The J2EE server is responsible for coordinating and propagating transactions between the server and the enterprise information system.

The application's persistent data is stored in database: *DataSource*, which holds information about accounts and courses. The application uses J2EE SDK support for distributed transactions to update and add an entry to the course details tables or account table in an atomic operation.

The *EJB Controller* delegates all the implementation of updating and adding an entry to the tables. It is responsible for maintaining consistency among the database tables represented by the enterprise beans that it calls. These is not any methods of *EJB Controller* explicitly invoke transactions, because it uses container-managed transactions. As a result, all database operations invoked by *EJB Controller* are automatically wrapped in a transaction by the container. The transaction context is automatically propagated to any enterprise beans that *EJB Controller* invokes.

5.8.2 Security

The “*Web-course search engine*” application is a Web-based application. A new provider of the “*Web-course search engine*” can *sign up* using a form presented by the application. While searchers can freely use the “search course” URL to search course, only providers who have signed in are allowed to access all the URLs belonged to member provider.

J2EE security mechanisms combine the concepts of container hosting, plus the declarative specification of application security requirements, with the availability of application-embedded mechanisms. This provides a powerful model for secure, interoperable, distributed component computing.

5.8.2.1 Security Requirements

The “*Web-course Search engine*” requires following security requirements:

- User Authentication

Users of the application can be either authenticated or unauthenticated. The user must be authenticated to access protected URLs. The application should be able to identify, differentiate, and be able to make access control decisions based on this distinction.

- User Administration

The application has its own set of users. This set of users grows when new users add themselves using a Web-based interface. Note that other applications, such as those developed for in-house use within an enterprise assume and use the set of users defined in the operational environment.

5.8.2.2 User Authentication

In a typical J2EE application, a user would go through a client container to interact with enterprise resources in the Web or EJB tiers. Resources available to the user may be protected or unprotected. A J2EE application must be capable of authenticating users that access the application from a variety of clients.

Web Tier Authentication

When an anonymous user tries to access a protected Web resource, the Web container will prompt the user for a password to authenticate with the Web container. The request will not be accepted by the Web container until the user identity has been proven to the Web container and shown to be one of the identities granted permission to access the resource.

When a user tries to access a protected Web-tier resource, the Web container activates the authentication mechanism defined in the application's deployment descriptor. J2EE Web containers must support three authentication mechanisms: HTTP basic authentication, form-based authentication, and HTTPS mutual authentication, and are encouraged to support HTTP digest authentication.

Because all of the interactions with the application occur through the Web-based interface, Form-based authentication, which can customize the authentication user interface presented by an HTTP browser in the J2EE architecture, is used to authenticate these interactions.

The authentication mechanism is configured using the `login-config` element of the Web component deployment descriptor.

In the application, the Web container designates a specific page, *login.jsp* that contains an HTML form, for logging in. This page contains an HTML form that prompts for a user name and password and is displayed when the user tries to access a resource that has been designated as being protected.

EJB Tier Authentication

The J2EE 1.2 platform specification doesn't require interoperable caller authentication at the EJB container. In addition, network firewall technology may prevent direct Internet interaction (via RMI) between client containers and enterprise beans. One way that an EJB container can protect access to enterprise beans is to entrust the Web container to assure for the identity of users accessing the beans via protected Web components. The application uses the Web container to enforce protection domain boundaries for Web components and the enterprise beans that they call. The structure is illustrated in Figure 5.9.

A user who wants to visit a protected EJB resource must have visited a protected Web resource. The application ensures this by fronting every protected EJB resource with a protected Web resource, and linking to a protected Web resource on every Web resource that calls EJB resources.

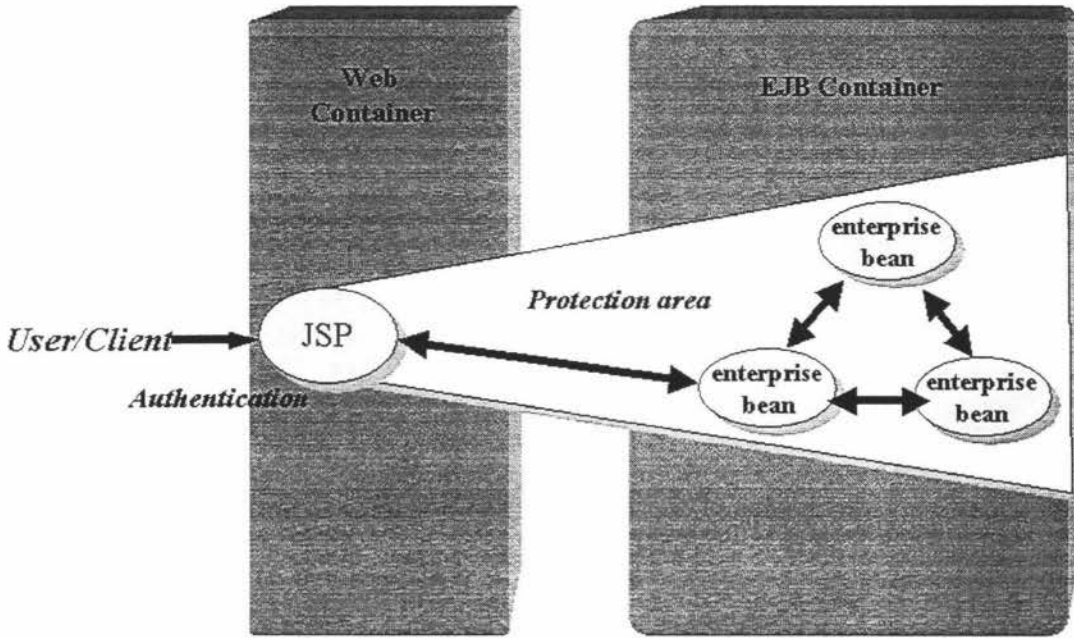


Figure 5.9 The enterprise beans protected by Web components

5.8.2.3 User Administration

The application needs to perform two tasks that aren't handled by the J2EE platform: managing user profile information and adding new users to the system dynamically.

Maintaining User Profiles

The application needs a separate relational table for storing user profile information. This table is called the *account* table, and is accessed through the Account enterprise bean. The user name is unique for each provider, and it is a key to the accounts database.

Adding New Users

Because the J2EE platform does not standardize a mechanism to add users dynamically to applications, the application does so in a non-portable, container-specific manner. The J2EE SDK provides a container-specific API for managing users based on the concept of realms. A realm is a collection of users under the same authentication policy. An application can provide its own realm and plug it into the

J2EE SDK for the container to use for authentication, or it can use realm API methods such *addUser*, on the existing default J2EE realm.

The application uses the default J2EE realm. It uses the *addUser* method of the realm to add new users while processing the *signup.jsp* form.

5.8.2.4 Programmatic Security

Although the J2EE platform can fulfill almost all security requests by using of declarative security, there are still some places where one needs to make access control decisions based on the current state of the system. In this application, before the system shows each membership provider interfaces/URLs to user, the user's login situation should be checked. Such decisions must be made by programmatically encoding their rules in the application.

The application uses the *AccountModel.isLoggedIn* method to check the user's login situation. This method is used to get the ID of the user that connects to the application. Code 5.1 shows how to use this method in one of *Screens* (Section 5.5.1 Interaction Interface)/URLs. The *AccountModel* is a Java Bean that contains the user ID information after the user success login the system. When the user logouts the system, this *AccountModel*, which belongs to the user, will be destroyed by the system.

```
<jsp:useBean
  id="account"
  type="account.model.AccountModel"
  scope="session" />
if (account.isLoggedIn()) {
    ...
} else {
  <p> <font size="5" color="red"> Please Sign in first! </font> <p>
}
```

Code 5.1 COURSE_MODIFY_SCREEN

5.9 Summary

This chapter illustrates a description of a multi-tier Web application: the “*Web-course search engine*”, implementation using the J2EE programming model.

The functionality of the application was determined using a scenario-driven approach. Walks through scenarios illustrated the requirements for the user interaction as well as the interactions that happen *within* the system. Analysis of the application identified two kinds of interactions: interfaces that allow providers to provide course records, and search interfaces for Web-course searchers carrying out searching activities.

The architecture of the application partitions its functionality into a 3-tiered Web-based deployment module, assigns functionality to tiers. The principles guiding the architecture include mixed the Web-centric and EJB-centric model, adapting the Model-View-Controller architecture to the domain of enterprise applications. The model represents the application data and the business rules that govern access and modification of this data. The view renders the contents of a model. It accesses data from the model and specifies how that data should be presented. The controller defines application behavior; it interprets user gestures and maps them into actions to be performed by the model.

The J2EE platform provides system services that simplify the work that application objects need to perform. The application uses the Java 2 SDK, Enterprise Edition support for distributed transactions across multiple JDBC databases. In addition, it uses deployment and security capabilities of the J2EE platform to support users with different profiles.

However, the functions of the “*Web-course search engine*” can be improved in many ways in the future, the next chapter will illuminate these possibility.

Chapter 6 Future of The *Web-course search engine*

This chapter discusses the possibility that the system described in this thesis can be improved in the future. These improvements of the “*Web-course search engine*” include the functions of the system and the cooperation with the other search systems.

6.1 *Searcher Membership Module*

The system can use searcher membership module to manage searcher’s profile, which includes the personal information of searcher and the searcher’s preferences regarding Web-course request or search criteria. Searcher can sign up as a member, the system will store all the information mentioned above. Then the system can use a search agent to return the new registered records to members according their search criteria in time. The searchers can use the main search function to get the search results immediately, or use search agent to wait for the new Web-course, which fit their search criteria stored in their profile. The system can research and analyze the searchers’ profiles and search history to get a lot of valuable information, which can be used in both academic and business areas.

6.2 *Metadata Editor*

The system can provides a Metadata Editor tool to help courseware providers edit their material records. This tool should have the abilities to evaluate the input metadata information and to provide other assistance functions.

6.3 Cooperation With Different Search Systems

Because of different reasons, there are many search systems around the Internet. How can we make use of these rich resources in the education area, or how can our system cooperate with other systems? Figure 6.1 and Figure 6.2 illuminate a possible solution scheme.

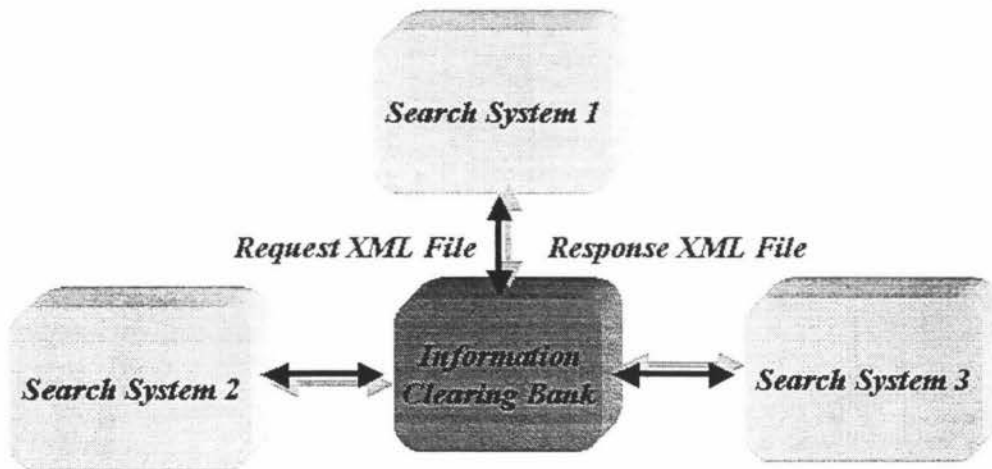


Figure 6.1 Cooperation between systems

The core part/organization of this scheme is the *Information clearing bank*, whose role is to manage and register the unique Web-course ID, check the rationality of request and response from each search system, and deliver the response to suitable systems. To fulfill these roles, there should be a common Metadata standard accepted by every search system. All communication files between the *Information clearing bank* and search systems should be XML style files following this Metadata standard.

When any of the search systems accepts a new Web-course record, it will send this course's details to the *Information clearing bank* using communication files. The *Information clearing bank* will then check other systems and return a unique Web-course ID back to the requesting system.

Every search system can send a search *Request* communication file to the *Information*

clearing bank. The *Information clearing bank* will deliver this *Request* to the other systems to get the search *Response*. The other systems, which will respond to this *Request*, will return search results to the *Information clearing bank*. Finally, the *Information clearing bank* will unify repetitious records by identifying the unique Web-course ID, compile all these search results together to a *Response* communication file, and deliver the file to the requesting system.

Of course, the *Information clearing bank* will check the rationality of all requests and response from each search system in all these processes.

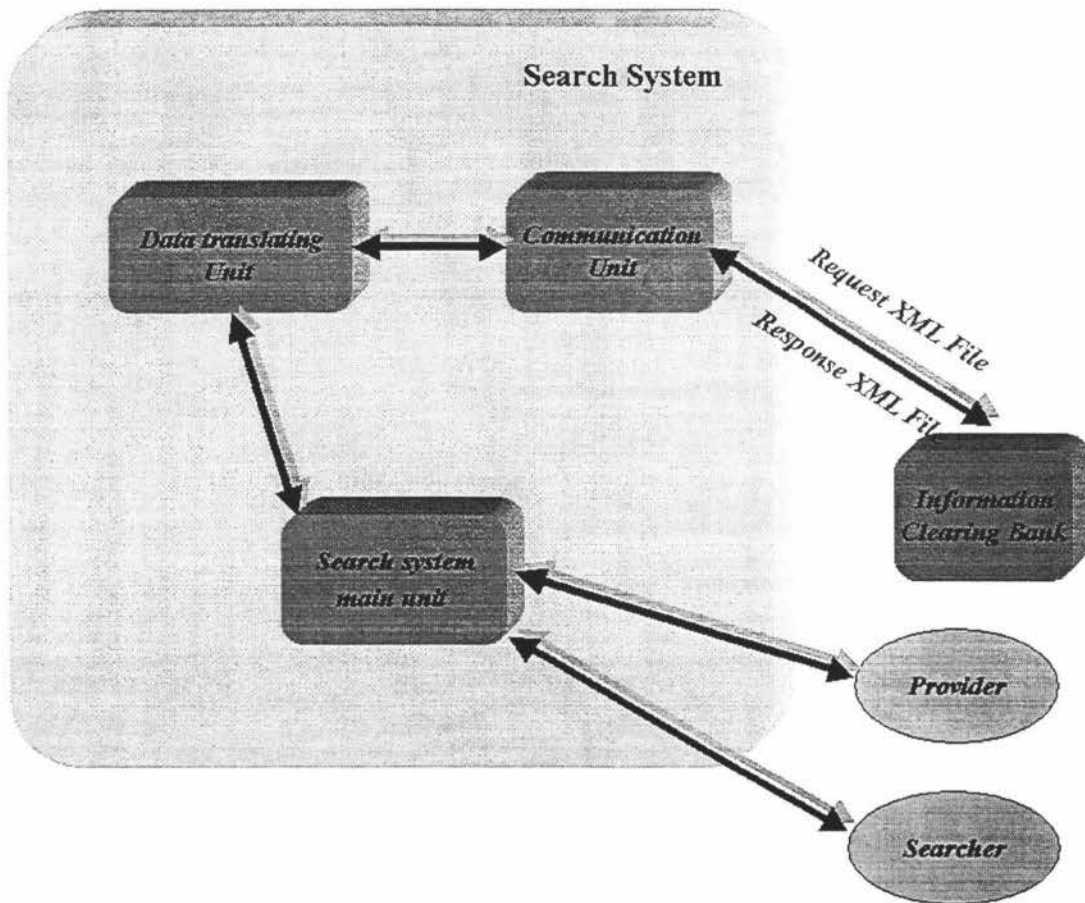


Figure 6.2 Inside a search system

There are three main parts (Communication unit, Data translating unit and main unit) inside a search system. The Metadata standard used in the search system should be based on or easily be transformed from/to the *Common Metadata Standard* using the

information clearing bank. The Communication unit's role is to send *Request* and receive *Response* from the *information clearing bank*. The Data translating unit is to translate data format of the communication files between the *Common Metadata Standard* of the *Information clearing bank* and the Metadata standard of this system. The functions of the Main unit of the search system are the same as our project, the *Web-course search engine*.

When the course provider submits new Web-course record, the search system requests a unique Web-course ID from the *Information clearing bank*. When searcher searches through this system, the search engine can just search the system, or can get help from the *Information clearing bank*. In this scheme, every search system has an outside communication relationship only with the *Information clearing bank*, not with other search systems.

6.4 Summary

This thesis has presented an overview of an application, the *Web-course search engine*, along with its design and development. It described the sequence from critiquing existing search systems, analyzing the requirements, and implementation of a new Educational Web-course search system. It explained the concepts of Metadata standards and technology used in designing the application. By using the *Web-course search engine*, the learners and the other educational users are able to index, search and find useful, valuable and related Web-course resources more effectively and efficiently.

References

About EdNA online. (n.d.). Retrieved July 31, 2002, from URL <http://www.edna.edu.au/aboutus/aboutsite.html>

About the LII. (2001). Retrieved July 31, 2002, from URL <http://lii.org/search/file/about>

ANZLIC Metadata Guidelines, Version 2, February 2001. Chapter Two: Metadata and the Metadata Standards. (2001). Retrieved July 31, 2002, from URL http://www.anzlic.org.au/asdi/metaelem_ch2.htm#ch2

Currie, M., Moss, N., Morrison, I. Ip, A. (1999). The EdNA Metadata Toolsets: a Case Study. Sixth Australian World Wide Web Conference (AusWeb2K), Cairns, June 2000. (Also online at <http://ausweb.scu.edu.au/aw2k/papers/currie/paper.html>)

DCMI Frequently Asked Questions (FAQ). (n.d.). Retrieved July 31, 2002, from URL <http://dublincore.org/resources/faq/>

Draft Standard for Learning Object Metadata. (2001). Retrieved July 31, 2002, from URL http://ltsc.ieee.org/doc/wg12/LOM_WD6-1_without_tracking.htm

Dublin Core Metadata Element Set, Version 1.1: Reference Description. (1999). Retrieved July 31, 2002, from URL <http://dublincore.org/documents/dces/>

EdNA Metadata Standard. (2001). Retrieved July 31, 2002, from URL <http://standards.edna.edu.au/metadata/>

Eric, J.G., Steve, L., William, P.B., and C.Lee, G. (1999). Architecture of a metasearch engine that supports user information needs. Proceedings of CKIM 1999.

Gelbman, R., and Mathys, T. (1999). Making Metadata Part of Your Daily Diet. Proceedings of the 1999 Urban and Regional Information Systems Association (URISA) annual conference.

Ip, A., Morrison, I., Currie, M. & Mason J. (1999). Managing Online Resources for Teaching and Learning. Sixth Australian World Wide Web Conference (AusWeb2K), Cairns, June 2000. (Also online at <http://ausweb.scu.edu.au/aw2k/papers/ip/paper.html>)

Ip, A., & Naidu, S. (2001). Reuse of Web-based resources in technology-enhanced student-centered learning environments. Campus Wide Information Systems, 18(4), 153-159. (Also online at <http://www.dls.au.com/metadata/Reusability.pdf>)

Ip, Albert., Currie, Mike., Morrison, Iain., Mason, Jon. (1999). Metasearching or Megasearching: Toward a Data Model for Distributed Resource Discovery. In Fred Castro, Robert Lai and Sr. Margaret Wong (eds) e-Education: Challenges and Opportunities. Proceedings of the Fifth Hong Kong Web Symposium 4-6 October 1999. 65-82. Social Sciences Research Centre, The University of Hong Kong, 1999. (Also online at <http://www.dls.au.com/metadata/DataModel.html>)

IRIS Keyword Thesaurus. (n.d.). Retrieved July 31, 2002, from URL <http://carousel.lis.uiuc.edu/~iris/thesaurus.html>

J2EE Design Patterns: Data Access Object (DAO). (2001). Retrieved July 31, 2002, from URL http://java.sun.com/blueprints/patterns/j2ee_patterns/data_access_object/index.html

J2EE Design Patterns: Model-View-Controller (MVC) architecture. (2001). Retrieved July 31, 2002, from URL http://java.sun.com/blueprints/patterns/j2ee_patterns/model_view_controller/index.html

Kinshuk., Patel, A., & Russell, D. (1999). Hyper-ITS: A Web-based architecture for evolving and configurable learning environment. Staff and educational development international, 3(3), 265-280. India: Aravali books international (P) Ltd.

Learning technology standards committee (LTSC). (n.d.). Retrieved July 31, 2002, from URL <http://ltsc.ieee.org/>

Loren, T., Will, H., & Brian, A. (1999). Constructing, organizing, and visualizing collections of topically related Web resources. ACM Transactions on Computer-Human Interaction, Vol.6 (1), March 1999.

Merriam-Webster Online: Collegiate Dictionary. (2002). Retrieved July 31, 2002, from URL <http://www.m-w.com/>

Metadata education project: Metadata education suggestions and materials for: Overview of the Metadata content standard. (n.d.). Retrieved July 31, 2002, from URL <http://www.sdvc.uwyo.edu/metadata/standard.html>

Metadata education project: Metadata education suggestions and materials for: Why is metadata important?. (n.d.). Retrieved July 31, 2002, from URL <http://www.sdvc.uwyo.edu/metadata/why.html>

Milstead, J., & Feldman, S. (1999). Metadata: Cataloging by any other name.... Retrieved July 31, 2002, from URL <http://www.onlinemag.net/OL1999/milstead1.html>

Neri, F. and Saitta, L. (1997). Machine learning for information extraction. Proceedings of the International Summer School on Information Extraction, SCIE '97, 171-191. Springer-Verlag Berlin Heidelberg, 1997.

Perkins, D.N. (1991). Technology meets constructivism: Do they make a marriage? Educational Technology, May, 1991, 18-23.

Rao, R., Pedersen, J. O., Hearst, M. A., Mackinlay, J. D., Card, S. K., Masinter, L., Haivorsen, P., & Roberson, G. G. (1995). Rich interaction in the digital library. Communication of the ACM, vol.38 (4), 29-39.

Sheth, A. P. and Larson, J. A. (1990). Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys, vol. 22, 183-236.

Smith, T. R. (1996). The meta-information environment of digital library. D-Lib Magazine, July/August 1996. (Also on <http://www.dlib.org/dlib/july96/new/07smith.html>)

Strawman, A. & Bretherton, F. (1994): A reference model for metadata. Retrieved July 31, 2002, from URL http://www.llnl.gov/liv_comp/metadata/papers/whitepaper-bretherton.html.

Taylor, A. G. & Clemson, P. (1996). Access to networked documents: Catalogs? Search Engines? Both? OCLC Internet Cataloging Project Colloquium, Retrieved July 31, 2002, from URL <http://www.oclc.org/oclc/man/colloq/taylor.htm>.

Theilmann, W. & Rothermel, K. (1998). Domain experts for information retrieval in the World Wide Web. Proceedings of Second International Workshop on Cooperative Information Agents, 216-227. Springer-Verlag Berlin Heidelberg, 1998.

Tutorial- NC Metadata for the PRL and NCGDC- Why Bother? (n.d.). Retrieved July 31, 2002, from URL <http://cgia.cgia.state.nc.us/tutorials/ncmeta/whybother.html>

Weibel, S. (1995). Metadata The Foundation of Resource Description. OCLC/NCSA Metadata Workshop. Retrieved July 31, 2002, from URL <http://www.oclc.org/research/publications/arr/1995/part2/weibel.htm>.

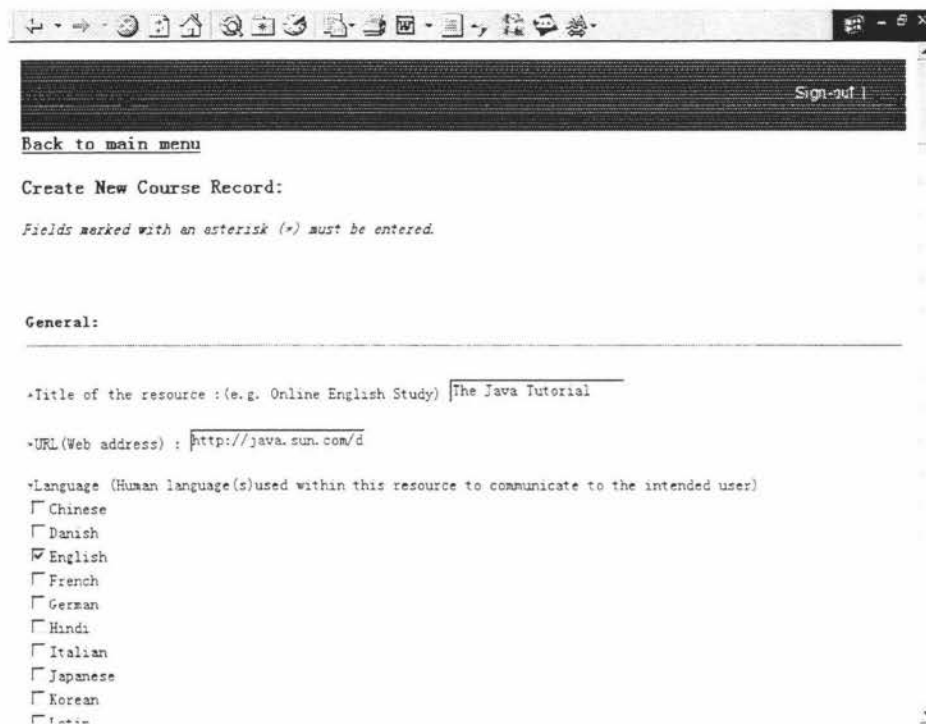
Appendix A: Screenshots of the system



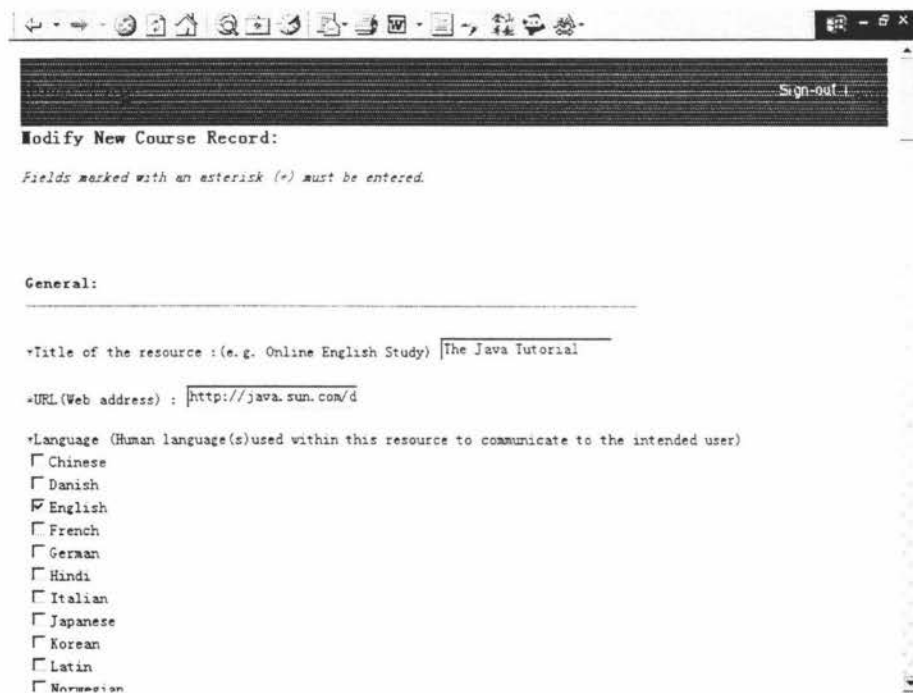
Course provider login.



Add, modify and delete courses options.



Add new course record.



Modify old course record.

Sign-out |

Delete this Course Record ? YES NO

Title of the resource is :

URL (Web address) is :

Language (Human language(s) used within this resource to communicate to the intended user) :
English ,

Description (A textual description of the content of this resource) :
A practical guide for programmers with hundreds of complete, working examples

Version (The edition of this resource) is : 1

Last Updated :
 / / (Month Day, Year)

Status (The state or condition of this resource) is: Final

Delete old course record.

Sign-in |

Account Information:

User ID:

Password:

E-Mail Address:

First Name:

Last Name:

Homepage Address:

Organization:

Running on Java 2 Enterprise Edition (J2EE)

New course provider register.

Courses search request

Please complete as much as possible before submitting to get better search results.

Subject (or Taxon) :

Keyword(s) :

Or Keyword :

[Select]

Or Keyword :

[Select]

Language (Human language(s) used within this resource to communicate to the intended user)

Chinese (7 items)

Danish (1 items)

English (9 items)

French (5 items)

German (3 items)

Hindi (2 items)

Latin (1 items)

Spanish (1 items)

Swedish (1 items)

Search criteria/options(1).

Or Keyword :

[Select]

[Select]

Agriculture

Area studies

Arts

Behavioral sciences

Biotechnology

Business administration

Communication

Computer Science

Earth Sciences

Education

Hindi (2 items)

Latin (1 items)

Spanish (1 items)

Swedish (1 items)

Author :

Publisher :

Learning Resource Type : (Specific kind of resource)

Diagram

Exam

Exercise

Experiment

Figure

...

Search criteria /options (2).

General:

Title of the resource : The Java Tutorial

URL (Web address) : <http://java.sun.com/docs/books/tutorial/index.html>

Language(s) :
English

Description :
A practical guide for java programmers with hundreds of complete, working examples

Coverage (The span or extent of such things as time, culture, geography or region that applies to this resource) :

Life cycle of this resource :

Version : 1
Last Updated : April / , 2002
Status (The state or condition of this resource) : Final
Author(s) :
sun
Publisher :

One searching result' details(1).

Learning Resource Type :

- Diagram
- Exercise
- Experiment
- Figure
- Graph
- Simulation

Interactivity Level (The degree of interactivity between the end user and this resource) : Low

Intended end user role (Principal user(s) for which this resource was designed) :
Teacher and Learner

Context (The principal environment within which the learning and use of this resource is intended to take place) :

- Higher Education
- University First Cycle
- University Second Cycle
- University postgraduate
- Technical School First Cycle
- Technical School Second Cycle
- Professional Formation
- Continuous Formation
- Vocational Training

Complexity Level (Difficulty) : Medium

Typical Learning Time :
200 : Hours

Description or guide for using this resource :
URL :

One searching result' details(2).

Appendix B: Outline of LOM Metadata Elements

1 General

- 1.1 *Identifier*
- 1.2 *Title*
- 1.3 *Catalog Entry*
 - 1.3.1 *Catalog*
 - 1.3.2 *Entry*
- 1.4 *Language*
- 1.5 *Description*
- 1.6 *Keywords*
- 1.7 *Coverage*
- 1.8 *Structure*
- 1.9 *Aggregation Level*

2 Life Cycle

- 2.1 *Version*
- 2.2 *Status*
- 2.3 *Contribute*
 - 2.3.1 *Role*
 - 2.3.2 *Entity*
 - 2.3.3 *Date*

3 Meta-Metadata

- 3.1 *Identifier*
- 3.2 *Catalog Entry*
 - 3.2.1 *Catalog*
 - 3.2.2 *Entry*
- 3.3 *Contribute*

- 3.3.1 *Role*
- 3.3.2 *Entity*
- 3.3.3 *Date*
- 3.4 *Metadata Scheme*
- 3.5 *Language*

4 Technical

- 4.1 *Format*
- 4.2 *Size*
- 4.3 *Location*
- 4.4 *Requirements*
 - 4.4.1 *Type*
 - 4.4.2 *Name*
 - 4.4.3 *Minimum Version*
 - 4.4.4 *Maximum Version*
- 4.5 *Installation Remarks*
- 4.6 *Other Platform Requirements*
- 4.7 *Duration*

5 Educational

- 5.1 *Interactivity Type*
- 5.2 *Learning Resource Type*
- 5.3 *Interactivity Level*
- 5.4 *Semantic Density*
- 5.5 *Intended End User Role*
- 5.6 *Context*
- 5.7 *Typical Age Range*
- 5.8 *Difficulty*
- 5.9 *Typical Learning Time*
- 5.10 *Description*
- 5.11 *Language*

6 Rights

- 6.1 *Cost*
- 6.2 *Copyright and Other Restrictions*
- 6.3 *Description*

7 Relation

- 7.1 *Kind*
- 7.2 *Resource*
 - 7.2.1 *Identifier*
 - 7.2.2 *Description*
 - 7.2.3 *Catalog Entry*

8 Annotation

- 8.1 *Person*
- 8.2 *Date*
- 8.3 *Description*

9 Classification

- 9.1 *Purpose*
- 9.2 *Taxon Path*
 - 9.2.1 *Source*
 - 9.2.2 *Taxon*
 - 9.2.2.1 *Id*
 - 9.2.2.2 *Entry*
- 9.3 *Description*
- 9.4 *Keywords*

Appendix C: Dublin Core Metadata Elements

<i>Element</i>	<i>Definition</i>
Title	The name given to the resource, usually by the Creator or Publisher.
Creator	The person or organization primarily responsible for creating the intellectual content of the resource. For example, authors in the case of written documents, artists, photographers, or illustrators in the case of visual resources.
Subject	The topic of the resource. Typically, subject will be expressed as keywords or phrases that describe the subject or content of the resource. The use of controlled vocabularies and formal classification schemas is encouraged.
Description	A textual description of the content of the resource, including abstracts in the case of document-like objects or content descriptions in the case of visual resources.
Publisher	The entity responsible for making the resource available in its present form, such as a publishing house, a university department, or a corporate entity.
Contributor	A person or organization not specified in a CREATOR element who has made significant intellectual contributions to the resource but whose contribution is secondary to any person or organization specified in a CREATOR element (for example, editor, transcriber, and illustrator).
Date	A date associated with an event in the life cycle of the resource.
Type	The category of the resource, such as home page, novel, poem, working paper, technical report, essay, and dictionary.
Format	The data format of the resource, used to identify the software and

possibly hardware that might be needed to display or operate the resource. For the sake of interoperability, FORMAT should be selected from an enumerated list that is under development in the workshop series at the time of publication of this document.

Identifier	String or number used to uniquely identify the resource. Examples for networked resources include URLs.
Source	A string or number used to uniquely identify the work from which this resource was derived, if applicable. For example, a PDF version of a novel might have a SOURCE element containing an ISBN number for the physical book from which the PDF version was derived.
Language	A language of the intellectual content of the resource.
Relation	The relationship of this resource to other resources. The intent of this element is to provide a means to express relationships among resources that have formal relationships to others, but exist as discrete resources themselves. For example, images in a document, chapters in a book, or items in a collection. Formal specification of RELATION is currently under development. Users and developers should understand that use of this element is currently considered to be experimental.
Coverage	The extent or scope of the content of the resource.
Rights	Information about rights held in and over the resource. A link to a copyright notice, to a rights-management statement, or to a service that would provide information about terms of access to the resource.

Dublin Core Metadata Elements