NOTE TO USERS

Université d'Ottawa · University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Zheng YIN

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M. Sc. (System Science)

GRADE - DEGREE

System Science Program

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Study of Metadata for Learning Objects

A. El Saddik

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

CO-DIRECTEUR DE LA THÈSE - THESIS CO-SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

A. Nayak

J. Zhao

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

STUDY OF METADATA FOR LEARNING OBJECTS

Zheng Yin

(Thesis Director: Dr. Abdulmotaleb El Saddik)

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the
Master of Science degree in Systems Science

Department of Systems Science
School of Management
University of Ottawa

Your file Votre référence
ISBN: 0-494-01655-8
Our file Notre référence
ISBN: 0-494-01655-8

Acknowledgements

My thanks go first to my husband: Luhong Fan. I would not have come here and finished this thesis without his love, understanding and support.

I would like to give my special thanks to my supervisor: Dr. Abdulmotaleb El Saddik, for his encouragement and guidance all the time.

I also wish extend my appreciation to Dr.Jiying Zhao, Jianming ke, Dazhi Yang, Xiaofei Liu, Yimin Ma, Ronghui Tu, Zhengfang xu, Feiyuan Wang, Manli Zhong and other friends for their help to my work and their friendship. I have a great time studying and working in the MCRLab.

Abstract

Metadata is descriptive information for data. The purpose of metadata is to facilitate describing, managing and discovering resources in huge distributed repositories. Metadata experts worldwide create the Dublin Core (DC), which acts as a fundamental core metadata standard on which industrial metadata standards based.

For educational industry, the need is increasing for description and exploration of a learning object (LO) in distributed learning object repositories (LOR) worldwide. Several organizations aim to establish metadata standards for facilitating better identifying, exchanging and reusing learning objects according to their specific needs. The Institute of Electrical and Electronics Engineers (IEEE) published Learning Object Metadata (LOM), which is a credited standard on the global level since it best represents the characteristics of digital learning objects.

Conversion from one metadata standard to another is necessary in case that people want to exchange and reuse learning objects tagged using different kinds of metadata standards. Mapping between the DC and the LOM is an essential job in many e-learning systems.

In this thesis, we present a new web based metadata editor for the DC and LOM, and a Web Services oriented mapping tool between them. Other clients can use our editor to create DC or LOM metadata records when catalog their learning objects into our LOR, and integrate the mapping web services as a part of different systems regardless of different platforms, protocols, and displaying devices. Our objective is to promote the reusability and interoperability for both the DC and the LOM users, therefore benefit the learning object industry by lowering the cost of using metadata. The DC-LOM mapping tool is demonstrated in our e-learning system called UbiLearn developed at the MCRLab of University of Ottawa.

Table of Content

List of Figures

List of Tables

Chapter 1. Introduction

In this chapter, the motivation and objectiveness of this thesis is summarized at first. Then my contributions are given. At last, the outline of this thesis is described.

1.1 Motivation

We are experiencing an evolution of education with the latest development in computer and communication technology, especially with the rise of object-oriented design, the high speed Internet and the World Wide Web. Learning is extending beyond the confines of the traditional classroom and the usual school age limits. Everyone can learn anytime, anywhere, at his or her own pace in a manner adapted to individual backgrounds, preferences and pedagogical goals. Compared to tutor-centric traditional study, the learner-centric new method brings more convenience, efficiency and pleasure.

In an e-learning paradigm, an online course is presented through digital learning material. Learning object comes into being with the need of cutting big, monatomic learning material for a course into several independent small chunks. The benefits of doing this are obvious: if a high quality learning object is created once, then it could be reused in other courses many times.

For a learning object to be reused by other people through the web, it has to be found effectively. Just like the yellow pages can help one locate a specific telephone number quickly, Metadata for the learning object, rooted from catalogs used in library, facilitates identifying, managing, retrieving and reusing the learning object effectively. Owning to its accurate description and structural organization for learning objects, metadata becomes a crucial part in searching for learning objects worldwide.

Dublin Core (DC) is the core metadata standard, [1] on which every other metadata is based. In the educational community, the Institute of Electrical and Electronics Engineers

(IEEE) learning object metadata (LOM) [2] is a widely accepted standard. Since other industries also have their special concerns, there exist hundreds of metadata schemas and the number is still growing. Furthermore, since learning objects and their associated metadata have been developed at a very high rate, unique metadata cannot satisfy all needs for all kinds of applications in the educational business domain as special concerns vary from one community to another. The fact that several metadata schemes coexist in the educational domain is necessary and this situation will last for a long time.

Since different communities have good reason to build their own learning object repositories with different metadata schemes, all applications are facing a complex distributional environment that is made up of distributional learning object repositories with different metadata schemes. This situation forces applications to know as many metadata schemes as possible that are widely used in their business domain. The more metadata scheme can be supported by an application, the more learning objects can be reused.

Mapping between two different metadata schemes works like a bridge to different metadata users. Once a user knows one metadata scheme, it is easy for him or her to obtain metadata records of the other metadata counterparts by the conversion according to the mapping between them. In educational industry, the two mostly used metadata are the DC and the LOM, therefore, the mapping between the DC and the LOM is often implemented repeatedly in countless e-learning systems worldwide.

Although the reusability and interoperability between different systems gain much attention for a long time, it is a long way to achieve this goal because of the complicated heterogeneous environment: different programming languages, different protocols, different displaying devices, and so on. Fortunately, there has a big progress recently with the development of new technologies: Internet, World Wide Web and Web Services.

1.2 Objectiveness

First, the metadata creation processes are time-consuming and error-prone. Moreover, metadata creation is so subjective that two people give two descriptions for one same learning object. Therefore, different people create different metadata records for the same learning object.

In order to save creation time and lower the subjective differences during the creation, there is a need for metadata editors that provide a consistent way to support the indexing, storage, discovery, searching, and retrieval of learning objects.

Second, in order to lower the total cost of using metadata, reusability and interoperability among heterogeneous computer systems through the Internet should be advocated. Especially in the educational business domain, countless associations, research institutes, universities and commercial companies cooperate worldwide, and complex issues like distributed software, business integration, different platforms, protocols, and various displaying devices make the reusability and interoperability of every system crucial.

In this work, we propose a new metadata editor to facilitate the consistent creation of the DC or the LOM for learning objects, and a DC-LOM metadata mapping built on a Web Services oriented architecture in order to achieve high reusability and interoperability. Two ways were provided for other clients to take full use of the presenting system. We suggest that outside clients through the Internet interact with the mapping tool as a Web Services while clients through an intranet visit use the traditional web application to improve the efficiency of the whole system.

1.3 Contribution

After Analysis two mostly used metadata standards in educational community: DC and LOM, a Web Services oriented DC and LOM mapping tool was implemented, which can be consumed by any kind of applications in order to increase the interoperability and reusability of different applications. Furthermore, a metadata

editor was developed, which provides a template for easily collecting DC and LOM records. This editor works as an interface for the DC and LOM mapping tool by saving the input DC record and LOM record from screen to DC table and LOM table in the Ubilearn database respectively. Moreover, A three tier architecture: client interface, business logic and data service, has been applied to the web application in designing. This web application consumes the DC-LOM mapping component as Web Services. At last, this mapping tool is integrated into the Ubilearn e-learning system in MCRLab of university of Ottawa.

Published papers:

- Zheng Yin, Zhengfang Xu and Abdulmotaleb El Saddik. "Study of Metadata for Advanced Multimedia Learning Objects". In Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE 2003) Montreal, Quebec, Canada, May 4-7, 2003.
- Zhengfang Xu, Zheng Yin and Abdulmotaleb El Saddik. "A Web Services Oriented Framework for Wired and Wireless E-Learning Systems". In Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE 2003) Montreal, Quebec, Canada, May 4-7, 2003.

1.4 Outline of this thesis

The rest of this thesis is organized as following:

Chapter 2 introduces the main advantages and limits in every main stage of e-learning evolution. Then it presents an overview of SCORM architecture and its main components: learning objects and metadata. Emphasis is given to metadata in the educational domain and rationales for the increasing need for interoperability among multiple metadata schemes.

Chapter 3 summarizes several important related works under processing recently based on SCORM architecture. It includes their main purposes, innovations, achievements and future work.

Chapter 4 describes the Web Services oriented metadata mapping tool architecture. First, outlines the requirements and design goal and then visualizes them with UML use cases. Moreover, introduces service-oriented architecture and its benefits. At last, present design models built on technologies, such as the J2EE, Apache struts web application framework, and Web Services.

Chapter 5 gives the implementation details of our system using HTML/Javascript, XML, XSLT, JSP, Servlet, Tomcat, MySQL, and Web Services. It also gives insight information on how to use developed system effectively.

Chapter 6 concludes the thesis and points out possible future work.

Chapter 2. State of the art

Although "e-learning" has been a popular word for a long time, it has different meanings from time to time and also different for different groups of people. What we will discuss here is the next generation web based learning systems. The standards are so important as the ultimate goal only can be achieved by cooperation worldwide. In this chapter, the infrastructure of the new e-learning system will be introduced and its key parts: learning object, learning object repository and metadata. Web Services is important new technology toward interoperability and reusability for heterogeneous system integration in different organizations, universities or commercial companies.

2.1 Evolution of e-learning

Psychologists and educators have been trying to take the advantage of the power of Information System for education in the form of computer-based instruction (CBI) systems [3] at the early days: Initially CBI systems were developed and implemented on mainframe computer systems, then on mini-computers, on workstations and later on personal computers. The availability of new capabilities and features of computer systems makes it possible to further automate instructional design and hide the complexities of programming. These make CBI systems very attractive in many areas. Moreover, that the cost of owning CBI systems is getting lower makes them affordable to more users. However, an obstacle comes from CBI systems is that they were very closely tied to the highly procedural and structural nature of early programming languages and computer systems. As a result, huge amount of resources were invested in migrating the systems from one technology to another along with the newer and better technology came into being. These kinds of migrations significantly slow down the development of CBI systems.

At the same time, some other groups of researchers began to explore the potential of generating learning content dynamically according to user's instructions. This led to

Intelligent Tutoring Systems (ITS) [4]. "Intelligent" is one of the most important characters of the ITS, which means that an ITS can generate instructions on demand in real time according to user needs. At the same time, lack of computing power, immature of cognitive science, insufficient delivery method of communication in the past prevented ITS from being widely used. In addition to that, the cost of developing ITS system is very high. Norm Friesen reported: "the development of a high quality, stand-alone course for interactive distance learning has been estimated at up to $1,000,000 per unit!" [5].

It is widely accepted that the object oriented technology, the growth of the Internet and the World Wide Web bring a revolution to the development of CBI and ITS. First, the Internet has become the universal delivery platform across the world. Second, web content is platform independent, highly accessible to any browser over the Internet and exchangeable from one system to another freely no matter how different the systems that produces them. Common standards are widely accepted as they bring long-term benefits in such a distributed environment. Third, the asynchronous accessibility and reusability of learning content largely reduce the development cost of e-learning systems. Finally, inherited from the object oriented technology, the concept of cutting the traditional big, monolithic learning content into small sharable, reusable, researchable learning objects makes the new generation e-learning practical.

A long-term vision of new e-learning is illustrated in the following:

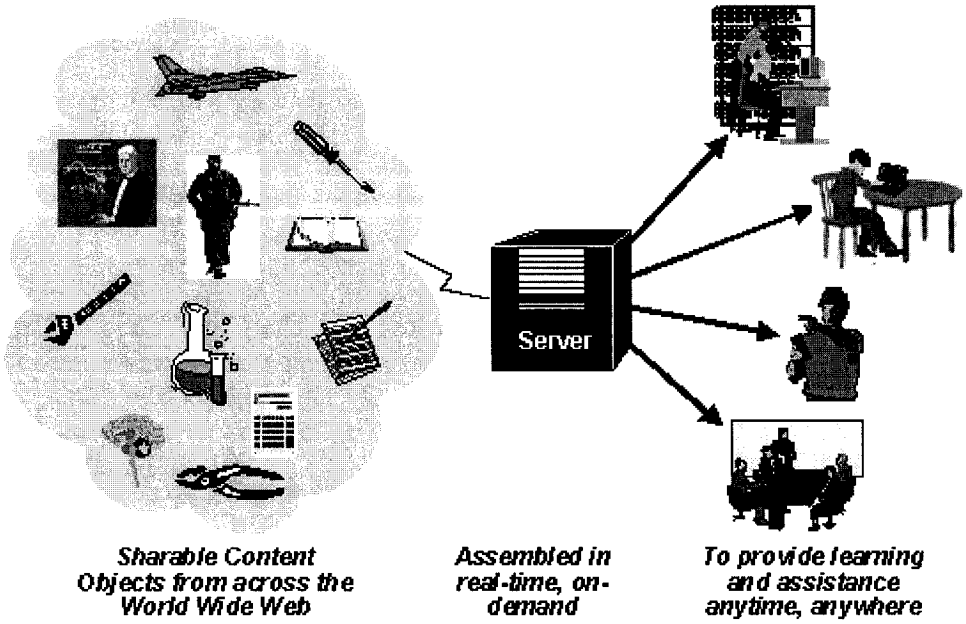


Figure 2-1: Long term vision of e-learning [6]

Figure 2-1 illustrates the ultimate goal of the e-learning industry. Large "knowledge" libraries or repositories should be created worldwide. Sharable digital learning objects are accumulated and well organized in a catalogued structure. These objects are required to be accessible, interoperable, durable and reusable. They can also be assembled in real time according to user's instruction. Therefore, user can study anytime anywhere through various device.

There is a raising demand for high quality, sharable learning objects because the creation of such learning objects is proven to be resource intensive and time consuming, not to mention that a good quality learning object takes countless academic peer review and revisions before it takes its shape. For developers, their learning objects need to be given more chance to be found and reused by others. On the other side, organizations, course developers and learners need learning objects that are ready to use since they want to reduce the cost and time of course development.

2.2 Learning object (LO)

Although definition of learning object varies from researchers to researchers and evolves from time to time, a learning object is essentially a digital resource with a demonstrated pedagogical value, which can be used, reused or referenced to support learning. Learning objects are described in CISCO reusable learning object strategy as "they can thus be a Java Applet, a Flash animation, an online quiz or a QuickTime movie, but it can also be a PowerPoint presentation, PDF file, an image, or a web page or site"[7]. No matter the number or the name is: Learning object, Reusable learning object, Reusable information object, Sharable content object, Modular building block, Chunk, Nugget, Lego, Whatever. The list still goes on. Lori Mortimer stated: "each term is simply a synonym for what we are calling a learning object"[8]. Obviously, learning object are building blocks of e-learning economy.

According IEEE Learning Technology Standards Committee (LTSC), "Learning Objects are defined here as any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning. Examples of technology supported learning include computer-based training systems, interactive learning environments, intelligent computer-aided instruction systems, distance learning systems, and collaborative learning environments. Examples of Learning Objects include multimedia content, instructional content, learning objectives, instructional software and software tools, and persons, organizations, or events referenced during technology supported learning" [9].

Since it has been proven that the creation of high quality learning objects is very costly, there is an increasing motivation for reusing them and sharing them. If they are constructed appropriately according to a given standard, identified, catalogued with appropriate metadata, and stored in well-constructed learning object repository, learning objects might find usage beyond their original audience and contractual context. The more they get reused, the lower the cost of ownership of these learning objects will be.

2.3 Learning object repository (LOR)

A LOR may be simply thought as place to host digital objects. The fundamental function of a LOR is managing its learning objects in a well-structured way to facilitate various searching operations. Two key technologies are involved for the repository strategy. The first is the architectures of building LOR (central repository architecture, distributed repository and peer-to-peer repository), as the learning objects can be vary in number, capacity, file type and pedagogical goal in different repositories. The second is the indexing mechanism that facilitates fast identification and efficient retrieving using metadata.

In general, a LOR stores both Learning Objects (LO) and their metadata. It is not necessary to store physically the actual object files in the same metadata repository; instead links can point to where the real learning object is and set a metadata repository will then act as portal to be visited first. Since the inherently nature of metadata is descriptive, it is better described in XML format. Although they are usually stored in traditional relational database management system (RDBMS), they will have higher performance if they are stored in XML database, using XML query to search them in the future when XML database technology get mature. At the same time, learning object should be stored in RDBMS or object orient databases or file systems.

In general, the metadata scheme chosen by a LOR depends on the purpose of the organization or company that implements it. For example, if a LOR only stores some special kinds of learning objects or have some special application purpose, it will adopt a metadata that can represent their special needs. For example, the Australian AVIRE contains only architectural object, and the metadata is specialized to the needs of the architectural community [10]. In the educational domain, LOM is widely used because it represents characteristics of learning objects. It is an IEEE standard, also recommended by many other international associations like ADL, AICC, ARIADNE, and so on for educational applications.

Special care has been given to the layout of LOR: whether it is centralized repository or hierarchical distributed repository. Some universities, associations build their LOR in centralized way especially when the number of LO available is small. It is good in rapid indexing and maintain. On the other hand, many others choose hierarchical distributed repositories as no single centralized repository can have sufficient size to hold all of the learning objects now and in the future.

It is unlikely that one organization can hold all learning objects worldwide in a single repository. Consequently, Griff Richards et al. [11] stated that the key to successful LOR strategy is its accessibility of their own learning objects as well as reusability to the learning objects in other LORs.

2.4 SCORM

Sharable Content Object Reference Model (SCORM), version 1.2 points out that "the department of Defense (DoD) and the White House Office of Science and Technology Police (OSTP) launched the Advanced Distributed Learning (ADL) initiative in November 1997. The purpose of the ADL initiative is to ensure access to high-quality education, training and decision aiding ("mentoring") materials that can be tailored to individual learner's needs and made available whenever and wherever they are required" [6].

ADL initiative envisions that learning material is represented by combination of several learning objects stored decentralized distributed learning object repositories. A learning object is created under some standards that make it sharable, interoperable and reusable. ADL assumes that learning objects exist all over the world and can be accessed and assembled in real time, on demand and then delivered to learners as needed anytime, anywhere under control of some learning management systems (LMS).

The assumption underlying is that any LMS can adapt its content, sequence, difficulty, etc. according to different users' special need, background, pedagogical goal, and so on. Moreover, every learning object is interoperable in different LMS.

ADL Initiative believes that a sharable learning object is the key technology that leads to their goal, and therefore, a widely accepted framework for web-based learning is needed in order to foster the creation of reusable learning object under some standards.

Now, ADL Initiative is a leading association among government, commercial companies and research organizations. Based on the work of Aviation Industry CBT (Computer-Based Training) Committee (AICC), IMS Global Learning Consortium Inc. (IMS), Institute of Electrical and Electronics Engineers (IEEE), Alliance of Remote Instructional Authoring & Distribution Networks for Europe (ARIADNE) a new architecture has been provided "SCORM" which is the first step towards learning revolution. SCORM provides the technical means for content objects to be easily shared across multiple learning delivery environments.

SCORM reference model defines a web-based learning "Content Aggregation Model" and "Run-Time Environment" for learning objects. It is made up of following components:

The SCORM Content Aggregation Model addresses the technical methods of utilizing learning objects; from identifying and retrieving them using metadata in learning object repository, to sequencing and aggregating them to form a course under the control of a given LMS.

SCORM Run-time Environment discusses the methods and means ensuring that learning contents developed by different tools can be interoperable in any other LMS, including launch, Application Program Interface (API) and data model.

Learning content need to move from large, inflexible courses to reusable, granular objects so that learners can personalize and assemble on demand in real time their needed courses; developers can build a small chunk of content in their best time, store it in

convenient location, reuse it in different formats with just a single mouse click. Distributed learning object repositories allow the developed learning content to be shared by countless people through the Internet, and incorporate learning objects can collaborate on and benefit immediately from new versions. This ultimate goal will be achieved through the interdependent components: learning object, metadata and LMS.

2.5 Metadata

The term "meta" derives from the Greek word "meta" denoting a nature of a higher order or more fundamental kind [12]. That means data about data.

For a learning object to be found and reused more easily, it should be described with metadata. According to Lrcan Dempsey et al. [13] a formal definition of metadata is as follows:

"Metadata is data associated with objects that relieves their potential users of having full advance knowledge of their existence or characteristics" [13].

Metadata is widely used in everyday life either for the digital resources or non-digital ones, such as: library catalogs, information on food packages and telephone yellow pages. Although there exist some other methods to locate a specific thing on the global level through the Internet, such as search engines (will compare them later this section), it is no doubt that metadata has played an important role for learning objects.

Metadata is descriptive information about resources for the purpose of finding, managing, using and reusing them more effectively. Metadata can be seen as a system of labels whose purpose is to describe a resource or object's characteristics and its objectives. A resource is anything that creators make available to others, such as: a book in a library, a document, a video or a music clip, multimedia content, instruction content, and software tools no matter if it is physically available on the Internet or not.

Since metadata can be used for different purpose, there are countless metadata standards and schemes. Anthony Roberts summarized that in educational domain, metadata are grouped into four tiers as illustrated below: "standards level, specifications level, application profile level and application level" [14].

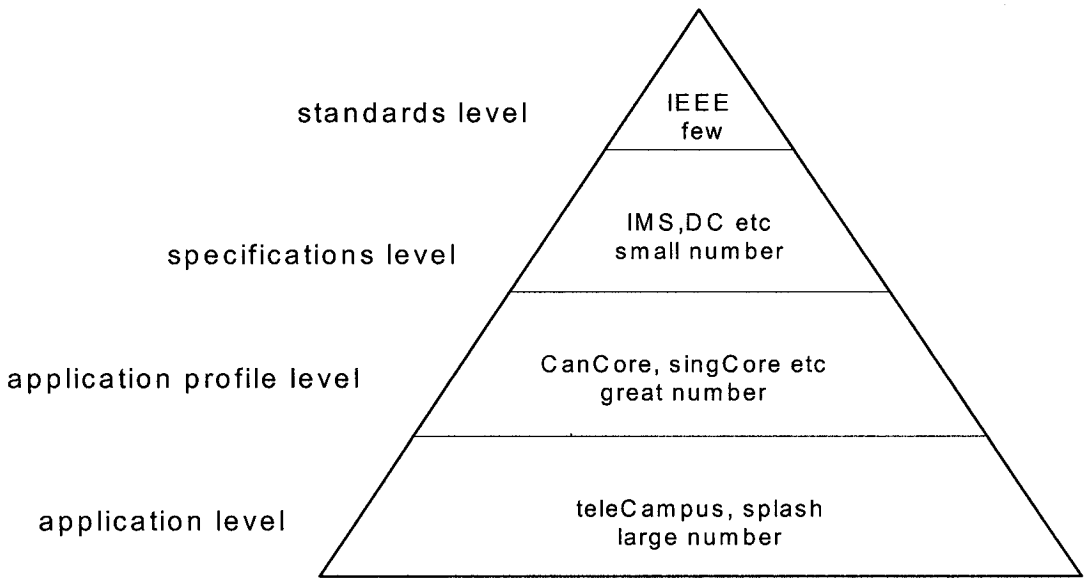


Figure 2-2: Metadata pyramid distribution [14]

The picture above shows that there are a great number of concepts and terms related to metadata in educational industry. The four tiers are divided by the scope of their usage. The one on the top represents the most general need. It aims to all the educational industry users worldwide. Based on this, several organizations add little more requirements according to their special concern of their smaller communities. Similarly, some associations define their own metadata scheme that is compatible to the fundamental ones in order for their convenience. At last, a great number of applications use personalized metadata for their specific interests.

Marek Hatala et al also pointed out: " there is a need for different levels of metadata exist – some generic perhaps following the global standard, while the other specific to the needs of the specific community of practice" [2].

2.5.1 The need for metadata

Although metadata was used in libraries during the 1960's, the idea of using metadata for digital resources came into being in the late 1990's. Before metadata, search engines were developed for facilitate retrieving information on the web.

Search engines, began around 1994 with the rise of the World Wide Web. Search engines achieved a big success that we can easily name quite a few: Google, Yahoo, Alta vista, Sina, and so on.

In general, search engines are built on crawling technologies that collect the web documents and keep them up to date. The crawling software is often called spider, crawler, harvester or gatherer. Moreover, search engines consist of storage technologies that store a large number of indices and optionally the web documents themselves. Equally important, search engines have sift mechanisms that can deal with users' queries quickly through millions of pages which are reordered in the database by index to find matches and rank them in the order of what it believes is most relevant.

First, the biggest shortcoming of search engine that has been realized is that every search engine will give users hundreds, thousands if not say millions of results most of the time: some items are good while others are not. This is quite frustrating situation, termed by Chris Taylor as "high recall" and "low precision" [12]. "The search results that come from search engines keep one busy with overloaded information. At the same time, it cannot provide users with the accurate documents users want to find because it lacks of means of distinguishing between important and incidental words in the document text" [12]. Chris Taylor also pointed out that the reason for this is that "the search engine companies market their products on the basis of their coverage of the Web, not in the precision of the search results" [12].

Second, Sergey Brin et al. mentioned another inherent disadvantage of search engines is that "current predominant business model for commercial search engines is advertising.

The goals of the advertising business model do not always correspond to providing quality search to users. The bias toward advertisers is inherent for advertising funded search engines"[15].

Third, since anyone can put anything on the web, the abuse of content is becoming a serious problem. "Misleading" can happen easily if the content creator wants to cheat, as search engines cannot find out what is right and what is wrong. This might be acceptable for web surfers, but definitely not for business or educational purposes.

Finally, we should take advantage of the huge number of non-text educational materials with the ongoing development of multimedia technologies and wide broadband network. For example, resources other than HTML (such as some learning contents), which take forms of image, video, audio, multimedia clips, etc. Considering that some important relevant information might be missed because search engines cannot identify them. A new efficient way is needed in order to efficiently search and reuse them.

The purpose of metadata is to facilitate fast identification and exploration of high quality resources on the global level. A great number of associations, research institutions, universities and commercial companies work together. They promise to make metadata a powerful tool. For example, Norm Friesen pointed out that "Metadata promises to solve this problem by describing data in structured and consistent way" [5]. Metadata works as a mechanism to ensure that high quality learning content can be easily found by others actively instead of being found by search engine crawlers passively. Because of the advantages metadata can bring, metadata has been enlisted as one of the most important parts in the next generation e-learning and e-business architecture.

2.5.2 Metadata for education

Dublin Core Metadata Initiative (DCMI)

Dublin Core Metadata Initiative (DCMI) began in 1995 with an international workshop in Dublin, Ohio that brought together librarians, digital library researchers, content providers, and text-markup experts to improve discovery standards for information.

The Dublin Core element set consists of 15 descriptors. It was made for wide purpose across all areas. It is not intended to be one unique standard; instead it is intended to co-exist with other metadata standards that provide more information for specific need. As a result, the Dublin Core is the fundamental metadata standard for other metadata standards.

According to the DC standard report, "The simplicity of Dublin Core can be both strength and weakness. Simplicity lowers the cost of creating metadata and promotes interoperability. On the other hand, simplicity does not accommodate the semantic and functional richness supported by complex metadata schemes. In effect, the Dublin Core element set trades richness for wide visibility. The design of Dublin Core mitigates this loss by encouraging the use of richer metadata schemes in combination with Dublin Core. Richer schemes can also be mapped to Dublin Core for export or for cross-system searching. Conversely, simple Dublin Core records can be used as a starting point for the creating of more complex descriptions"[16].

The following table gives a brief view of the 15 elements and their definition.

Table2-1: Dublin Core element set [16]

ID	Element Name	Definition
1	Title	A name given to the resource
2	Creator	An entity primarily responsible for making the content of the resource
3	Subject	The topic of the content of the resource

4	Description	An account of the content of the resource
5	Publisher	An entity responsible for making the resource available
6	Contributor	An entity responsible for making contribution to the content of the resource
7	Date	A date of an event in the lifecycle of the resource
8	Type	The nature or genre of the content of the resource
9	Format	The physical or digital manifestation of the content
10	Identifier	An ambiguous reference to the source within a given context
11	Source	A reference to a resource from which the present resource is derived
12	Language	A language of the intellectual content of the resource
13	Relation	A reference to a related resource
14	Coverage	The content or scope of the content of the resource
15	Rights	Information about rights held in and over the resource

IEEE Learning Technology Standards Committee (LTSC) Learning Object Metadata (LOM)

The Learning Technology Standards Committee (LTSC) created by the IEEE Computer Society Standards Activity Board to develop accredited technical standards, recommended practices and guides for learning technology. The LTSC coordinates formally and informally with other organizations that produce specifications and standards for similar purposes in educational area. Learning Object Metadata (LOM) is made from LTSC learning object metadata workgroup, named IEEE P1484.12. It is based on former work of several organizations: AICC, IMS, ARIADNE, and so on.

The LOM standard extends the Dublin Core in order to take into account the digital learning objects' special needs. According to LOM specification, elements describing a learning object are grouped into categories. The Base Scheme consists of nine categories [17]:

1) The *General* category groups the general information that describes the learning object as a whole.

2) The *Lifecycle* category groups the features related to the history and current state of this learning object and those who have affected this learning object during its evolution.

3) The *Meta-metadata* category group's information about this metadata record itself.

4) The *Technical* category groups the technical requirements and characteristics of the learning object.

5) The *Educational* category groups the educational and pedagogic characteristics of the learning object.

6) The *Rights* category groups the intellectual property rights and conditions of use for the learning object.

7) The *Relation* category group's features that define the relationship between this learning object and other targeted learning objects.

8) The *Annotation* category provides comments on the educational use of the learning object and information on when and by whom the comments were created.

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

2.5.3 Problems in using metadata

An ongoing controversy among metadata researchers is the question on how much information is just perfect for describing a learning object. On the one hand, more information within a metadata record results in a more accurate description of the learning object. Marek Hatala et al, (2002) pointed out "the more precise the metadata is,

the more precise the search result will be" [18]. Too little information in a metadata scheme causes lack of accuracy, and may result in too many false positive results. On the other hand, more time are needed if there are more tags (metadata descriptors) in a given metadata scheme because more work is involved to fill out the tags. Moreover, greater vocabulary used in a metadata leads to more subjective understanding by different people. More training should be given to people who take responsibility to create the metadata and to those who need to retrieve them. The main disadvantages of using metadata are its high cost and subjectivity. Another disadvantage is the number of ongoing standard initiatives, as quoted by Enric Peig et al (2001) "Another problem that we need to be aware of is the introduction of new metadata schemes that might appear in the future. Applications should adapt to these new emerging schemes." [19].

Since separating learning objects and their metadata is popular, an agent or third party instead of the learning object developers themselves creates some metadata records. The consistent using of the same metadata scheme semantics becomes an increasing problem. For example, do "author" and "creator" mean the same in different context? This problem causes confusion for reuse effectively.

Another fact we can see is there are literally at least tens of metadata schemas in education community, not say there are more in related fields like libraries. And the number is still growing rapidly as different researchers or companies have specific need for their research or commercial goals. Different metadata schemes are different in structure, listed vocabulary, and so on. These prevent learning object be discovered and reuse by other applications which use different metadata schemes.

2.6 Web Services

2.6.1 Web Services overview

Web services can be understood as simply as services provided through web. The purpose of web services is to enable a distributed environment in which applications can

interoperate seamlessly regardless of different platforms and program languages they build on.

A simple scenario for Web Services is currency exchange quote. The provider needs to implement the service: give the real-time rate conversion from one currency to another, and then publish the service to a repository including information about functions of the service, interface, and unique URI (Uniform Resource Identifier) for users to locate it, and so on. Other applications such as a bookstore which sells books from different countries then can search the repository and invoke the currency exchange quote as part of their system by just call it instead of building it from the start by scratch.

Low level networking control is necessary for most of application developers in the old days. For example, in the AT&T UNIX system source code, much work needs to do in order to implement cooperation through network: opening and closing sockets, listening on ports, formatting requests, decoding response. Remote Procedure Call (RPC) provides a easier way to do that. Middleware (Common Object Request Broker Architecture (CORBA), Component Object Model/Distributed COM (COM/DCOM), and Java Remote Method Invocation (RMI)) derived from the RPC offered developers better choices when design distributed application. Middleware takes care of all the networking and protocol details for application software developers and provides designate remote objects as remote. Although middleware is proven successful and efficient on private network environment, for example, RMI in pure java enterprise application, DCOM works in Microsoft environment, they are incompatible each other. Therefore, they are not suitable for inter-enterprise, large heterogeneous network environment- Internet. Not saying that they are hard to deploy and do not address special need of the World Wide Web.

The advantage comes with web services is interoperability. Provider and subscriber (user of the web services) can have independent operation system, UNIX or WINDOWS, no matter what programming tool they use: Java or Microsoft VB, and regardless of what component technology (CORBA, COM/DCOM, or RMI).

The Universal Description, Discovery and Integration (UDDI) repository broker is the mechanism matching providers and requestors by providing a public registry and interface to requestors through Internet or Intranet. There are variety of specification of XML registry (Electronic Business XML (ebXML) providers, UDDI providers or other providers), and more than one implementation for industry leading UDDI repositories (Microsoft UDDI repository, IBM UDDI repository and other UDDI repositories). Each broker collects and manages its entries of special interest, most probably relevant to a special industry, in a cataloged way. When a requestor makes a call on the repository, it will get an enumeration of entries that match its query. On completion of the matching process, the requests will interact with the provider directly.

In order to achieve interoperability, web service take advantage of the following important existing standards:

XML: The Extensible Markup Language (XML) is designed to describe data, focus on the syntax of the data rather than how to display it. Since XML is platform independent, it is ideal for exchanging data in web-based environment.

WSDL: The Web Services Description Language (WSDL) is an XML-based language used to describe the services a business offers and to provide a way for individuals and other businesses to find those services electronically. A WSDL entry describes what a service does, how to invoke its operations, and where to locate it.

UDDI: The Universal Description, Discovery and Integration (UDDI) is an XML-based registry for businesses worldwide, which enables businesses to list themselves and their services on the Internet. It is a collection of WSDL entries and act as publishing and locating information about Web Services just like yellow page for companies.

SOAP: Simple Object Access Protocol (SOAP) is a simple XML based protocol to let applications exchange information over different protocols. SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages.

Summed up by David A. Chappell et al. [20] Web Services have the following characteristics:

- **XML-based**: using XML as the data representation layer.
- **Loosely coupled**: the Web Services consumer and provider are not tightly tied with one another. They can be thought as different applications. This means any change in the interfaces on service side does not need the regeneration of the invocation part on the consumer side, or the service consumer can change to utilize any other Web Services which provide the same information.
- **Coarse-grained**: a service is made of several components, and a component is made of several objects. The bigger the building block, the powerful the reuse will be.
- **Ability to be synchronous and asynchronous**: service consumer blocks and waits for the response to complete its operation before continuing in synchronous invocation. Asynchronous service consumer can continue work and get their result later when the service is completed. Asynchronous is the key factor to loosely couples systems.

2.6.2 Web Services-based e-learning

In a Web Services-based e-learning system, major functionalities and features are designed and implemented as Web Services. In prototypical LearnServe system [21], a variety of aspects, features, and components are perceived and realized as Web Services, including content authoring, content configuration into classes or courses, LO management, content updating, learner registration and management, content adaptation, learner profiling and tracking, testing of acquired knowledge, tutoring, virtual classroom setups, organization of chat room and the search for the presentation of content itself.

By decomposing the functions and features of e-learning systems that are loosely coupled with Web Services, different e-learning systems can benefit from reusing the components directly beyond the compliant problems of heterogeneous platform they are built upon, therefore avoid developing them from scratch.

Chapter 3. Literature review

Since a great number of people have been researched in this area, there are some important projects that work as milestones for learning object and metadata.

3.1 LOM search among Learning object repositories

With the development of the international standardization for learning object and metadata (SCORM), the numbers of learning object repositories (LOR) are growing rapidly.

In the LOM-based repository survey, Filip Neven et al. compared architectures of repositories, number of learning objects available, metadata scheme based on the LOM standard, and each LOR's functional features.

In the following table, Filip Neven et al. summarized the different LOR's they studied, with a comparative analysis of their features and characteristics [22].

Table 3-1: A survey of LOM-based learning object repositories (abbreviated) [22]

	ARIADNE	SMETE	MERLOT	HEAL	CAREO	Learn-alberta
Organization	foundation	federation	cooperation	US Nat. Science Foundation	Universities	Alberta Learning
Metadata scheme	IEEE LOM profile	IEEE LOM profile	IEEE LOM profile	IEEE LOM profile (CanCore)	IEEE LOM profile (CanCore)	IEEE LOM profile (CanCore)
Subject domain	All	Science, math, engineering and technology	All	Health science	All	Kinder-garden to grade 12 education
#LO's	2498	1645	7408	N/A	1576	?
IPR mgmt	Free and restricted	Free	Free	Free	Free	Free
Peer review	Metadata validation	No	Review of content quality,	No	No	Yes

			effectiveness, ease of use			
Personal features	Metadata templates	Workspace, recommendations, communities	No	No	Workspace, download history	No
Distribution	Hierarch, knowledge pool system	Central server	Central server	Central server	Central server	Portal to repositories
Metadata store	Oracle	?	RDBMS, with export to XML	SQL server 2000	?	?
LO store	Document repository	Links	Links	Links	Document repository + Links	Document repository
Connection with other LOR's	Planned	API for federated search under development	Import of LOM records planned	Under investigation	Collects objects from other LOR's	Portal which links to different databases

As a conclusion to the above overview, an evaluation is given on the following topics to discuss the advantages and disadvantages of different design decisions.

- **Decide on fields of interest**. Some of them use the IEEE LOM standard while some of them custom it to different metadata schemes (application profiles): the CanCore that is used in HEAL, CAREO, and LEAN-ALBERTA. The costumed metadata scheme is useful when gather specialized collection of learning objects [22].
- **Whether to use Intellectual Property Rights (IPR).** This means some objects are freely available and others require payment or are only accessible to a certain group of users [22].
- **Multiple search modes or single search mode.** Simple search is useful for new users for them to get a quick impression of the repository's contents while advanced multiple search allows users to quickly find related resources through some criteria on metadata fields [22].
- **Whether provide Peer-review or not**. Peer-review facilitates the task of evaluating the quality of a resource when it appears in the result page of a query. It is a time-consuming activity, and need to be supported by the organization that serving the LOR [22].
- **Architectural design: client-server or peer-to-peer network.** Client-server is good in efficiency for querying and updating. Peer-to-peer is better for scalability [22].
- **How to obtain more learning objects outside**. For a repository to become useful it is crucial to obtain a 'critical mass' of learning material. In order to obtain this critical mass the ability for different systems to interconnect is becoming more and more important. Systems should allow other repositories to collect their free content, and tools should be developed that gather objects from other systems. Of course this introduces interoperability issues, especially because all systems use their own application profile [22].

- **Provide personal service to user or not**. Generating User profile as some system facilitates the generation of user profiles based on user's downloading behavior and point the user's attention to material he would be interested in [22].
- **Type of storage for metadata records:** RDBMS or XML databases.

3.2 POOL project and its metadata scheme (CanCore)

The Portal for Online Objects for Learning (POOL) project is a consortium to build learning object repository (LOR) that is scalable at the Canadian level. Funded in part by the "Canarie learning Program", POOL contributes to the development of two focal technologies: a distributed architecture for a peer-to-peer network of learning object repositories called "POOL, POND, and SPLASH" and Canadian Core Learning Resource Metadata Protocol (CanCore), a practical metadata protocol for cataloguing learning objects. CanCore is conformable with both ASL/SCORM and IMS Global consortium, [11], therefore compatible with IEEE LOM.

3.2.1 Architecture of POOL, POND and SPLASH

"POOL, POND and SPLASH" is a three layer distributed architecture, represents personal, community and national wide, illustrated as following:

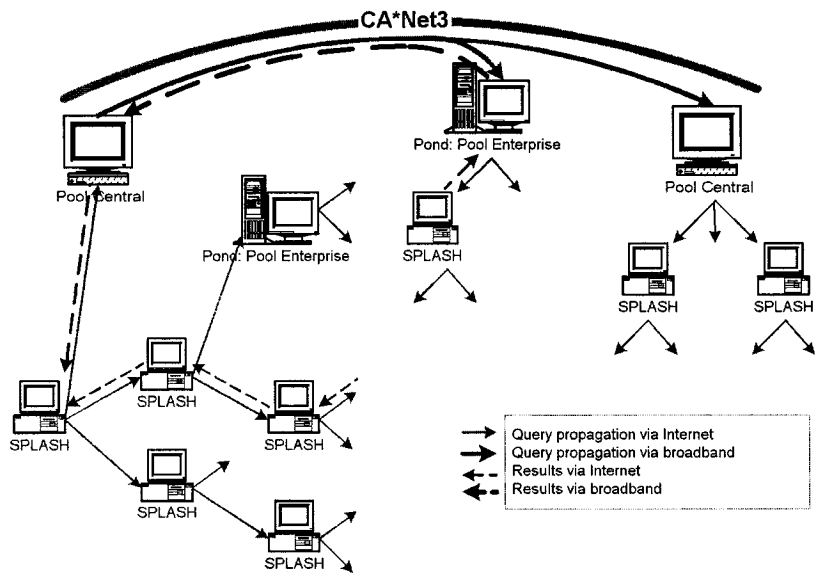


Figure 3-1: POOL, POND, and SPLASH architecture [11]

SPLASH is conceived of as a small single-user repository that would be made freely available for downloading via the Internet. SPLASH combines a database program and a peer-to-peer search engine with a CanCore metadata interface. Built on Sun Microsystems JXTA platform, each SPLASH site holds those objects of immediate importance to the owner, and has the ability to search other SPLASH peers and, subject to permissions granted, to exchange learning objects or learning object metadata with other members of the network [11].

PONDS represent collection of learning object repositories within a communities and organization. They are incorporating repositories holding learning object of common themes or purpose [10]. Communities and organizations are a reality of the world of education and training. Ministries of education, universities, school boards, schools and employers are typical of organizations that will have an interest in providing access of such a collection and govern the access, workflow and life-cycle management of the collection. For example, CAREO repository developed in the Province of Alberta. PONDS are accessible using POOL protocols and searchable using the CanCore metadata standard [11].

POOL central, a third level of aggregation, was devised to replicate search requests in topological regions of the repository network, and overcome the horizon effects that arise in decentralized peer-to-peer networks. The designation of a number of "super nodes" could facilitate a faster and more exhaustive search of all of the member repositories via a high-speed and high bandwidth connection to the Ca*Net3 optical highway that spans Canada. [11]

3.2.2 CanCore: Canadian protocol for learning object

CanCore has been developed as the metadata application profile for POOL and other CANARIE sponsored projects. CanCore provides a streamlined version of the complex IMS metadata standard, which forms the base for the SCORM metadata standard, also fully compliant with the international standard: IEEE LOM [11].

The POOL project assumes that neither the DC nor the IEEE LOM is ready-made metadata solution for the collection and sharing of learning object. The 15 elements provided by DC are too limited and is not capable in describing the special attributes of educational applications. On the other hand, 86 elements of IMS or 69 elements of IEEE LOM are too complex to be implemented. For example, many extensive set of attributes have wide definition and too much time and resource must be involved to implement due to their complexity.

CanCore has taken only the active "core" elements from IMS, which are considered essential for learning object implementations [13], provides 36 IMS fields, organized in nine categories based on the IMS structural approach.

Table 3-2: CanCore elements [10]

General	Groups information describing learning object as a whole. *Active elements:* Identifier, Title, Catalogentry.Catalog, Catalogentry.Entry, Language, Description, Coverage
Lifecycle	History and current state of resource.

	Active elements: Version, Contribute.Role, Contribute.Entity, Contribute.Date
Metametadata	Features of the description rather than the resource. *Active elements:* Identifier , Catalogentry.Catalog, Catalogentry.Entry, Contribute.Role, Contribute.Entity, Contribute.Date, Metadatascheme, Language
Technical	Technical features of the learning object. *Active elements:* Format, Size, Location, Otherplatformrequirements, Duration
Educational	Educational or pedagogic features of the learning object. *Active elements:* Learningresourcetype, Intendedenduserrole, Context, Typicalagerange, Language
Rights	Conditions of use of the resource. *Active elements:* Cost, Copyrightandotherrestrictions, Description
Relation	Features of the resource in relationship to other learning objects. *Active elements:* Kind, Resource.Identifier, Resource.Catalogentry
Classification	Description of a characteristic of the resource by entries in classifications. *Active elements:* Purpose, Taxonpath.Source, Taxonpath.Taxon.Entry, Keyword

In order to promote the use of CanCore, much work has been done involving training and documentation. CanCore has become a widely used metadata standard because of its large user base. On one hand, when creators of learning objects catalogue their learning objects, they put them into LORs using CanCore. On the other hand, when learners need to locate learning objects for a special topic, they turn to LORs that use CanCore. Obviously, as a result of the Cancore promotion, the reusability of the learning objects in the distributed repositories created by the POOL will be higher than others.

3.3 ARIADNE

3.3.1 Architecture of ARIADNE

ARIADNE is supported by the Commission of the European Union in the framework of the Education and Training sector of the Information Society Technologies (IST) Program. The primary goal of ARIADNE is to foster the share and reuse of electronic pedagogical material, both by universities and corporations. To this end, ARIADNE has built the Knowledge Pool System (KPS), a Europe-wide distributed repository for pedagogical documents, with associated indexation and query tools [23].

The ARIADNE system is based on the "core" tools that allow indexing, storage, and diffusion of the various teaching documents. Various authoring tools are also proposed to help the teaching engineers in the creation of these documents [24].

The diagram below illustrates the architecture of ARIADNE system.

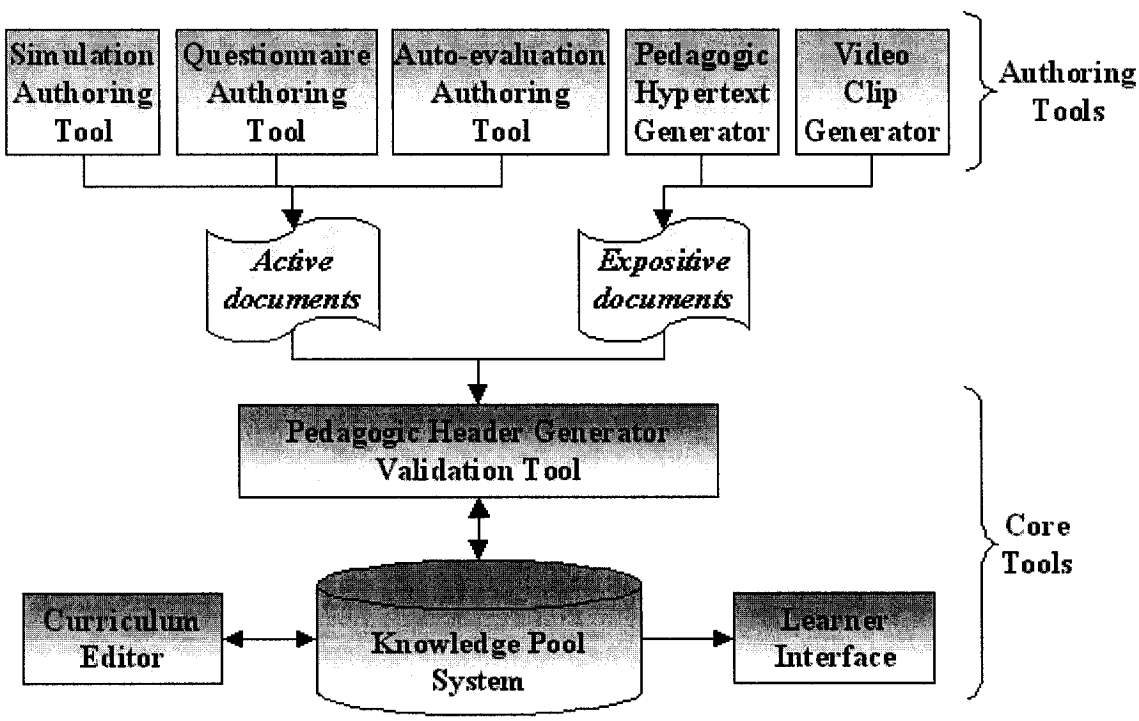


Figure 3-2: Architecture of ARIADNE system [24]

3.3.2 ARIADNE's work in educational metadata

One of the key features of the KPS is the underlying metadata specification, which is derived from work and experiments performed since 1995. ARIADNE submitted an early version of this specification in April'98 to the IEEE LTSC Learning Object Metadata (LOM). Together with a similar specification contributed by the IMS Project, that early ARIADNE version was the basis of the LOM standard [23].

According to the ARIADNE Educational Metadata Recommendation version3.2 (February 2002), the ARIADNE metadata is an application profile of the LOM specification. ARIADNE regrouped metadata into small number of categories because they assume that a smaller number of essential elements are easier to understand, indexation when creation and easy when exploitation. The six categories are as following [23]:

1. General information on the resource itself
2. Semantics of the resource
3. Pedagogical attributes
4. Technical characteristics
5. Conditions for use
6. Meta-metadata

3.4 EXPLORA

Explora-2 is the result of about ten year's research development effort by a group of research fellows in University of Montreal. At the beginning, they aimed to build a distance teaching and learning solution using Virtual Learning Center (VLC) model, architecture and prototypes. Then the group implemented a system (using web-base technology), called explora-1. Between 1999 to fall of 2002, explora-2 had been developed and two implementations had been made at a university of Quebec and an educational television station for French Canada [25].

Through an embed Explor@ window, the VLC make it possible for a set of courses to share a series of teaching resources, including tools, files, means of communication and

documents. It has a central bank of resources and assembles them dynamically to fit course requirement and different user role. The limit of Explor@-2 is that their courses are delivered to user in only HTML format over the Internet [26] .

Among the great number of functionalities Explor@-2 offers, learning object manager is the most interest part to us. In order to facilitate learning object identification-referencing-search-use in learning object repositories, a standardized metadata repository is build to publish and reference leaning objects.

A central component of the learning object manager is metadata editor that provides forms to describe IEEE/IMS/CanCore metadata and store the metadata file in their central relational/XML database. The other five components are specific user services of a metadata repository [25].

1. Metadata Repository Builder - help find the location of LO
2. Repository Structure Manage – select a folder or grouping where a LOM record will be found
3. Repository Search Agents – search metadata records according to constraints
4. Access Manager – defines the authority rules
5. Collaborative Annotator – manage a local user forum where users can exchange message about a LO.

Discussions also took place in Explor@-2 about how many attributes are sufficient for a metadata scheme. In their previous work, their classification scheme contained 187 sub-categories, which makes it a large superset of IEEE LOM because they thought that the vocabulary of LOM is not sufficient for LO description. While some researchers thought a possible solution is to distinguish metadata attributes in hierarchical structure based on the difference of the granularity of LO. Different level's LOs have their own metadata. The top level LOs are the most reusable ones. A good approach is using Resource Description Framework (RDF) to make semantic description for learning objects [25].

Several services are offered by Explor@-2:

- Adding a LO record
- Grouping and annotating Los
- Assigning access rights
- Searching the repository

3.5 Web Services based LOR - LORAX

The Le@rning Federation Initiative aimed to develop a high quality multimedia learning system for Australian and New Zealand schools in the past 5 years. The Le@rning Federation's Learning Object Repository Access and Exchange (LORAX) Specification defines Web Services for interacting with the Exchange repository of learning objects. The Web Services provide a simple programmatic interface allowing client systems to discover and download metadata and learning objects from a repository called "Exchange" [27].

The SOAP interface consists of four functions [27]:

- The Query Learning Object (QueryLO) operation takes a search request and returns short descriptions of published learning objects that match the search.
- Download Learning Object (DownloadLO) takes a list of learning object Identifiers and returns those learning objects. The learning objects are returned as zipped IMS Package Interchange Files (IMS Content Packaging Specification, 2001).
- The Retrieve Learning Object Metadata (RetrieveLOMetadata) operation takes a sequence of learning object identifiers and returns complete metadata descriptions for those learning objects.
- The Test Connection (TestConnection) operation can be used to test the Web Services connection without affecting any data.

In conclusion, there exist a great number of associations, organizations and commercial companies that are involved in building e-learning standards and implementing various

kinds of systems. They are cooperated with each other on the global level and compete as well. Therefore, a great number of new important concepts and buzzwords have been created and the trend will still go on. Furthermore, since each of them built its system separately for a special goal, its system has different features and characters. In another word, each system has its advantages and limits as well. Moreover, different systems use different programming languages, different platforms, different metadata application profiles and different learning object repositories. All existing systems cannot interoperate each other. Much precious recourse in developing them might be lost in the long run.

Concerning learning object metadata, there are a great number of buzzwords and confusing words used by different groups of researchers and experts. Therefore, on one hand, the users find it very difficult and expensive to accept more than one standard. On the other hand, if they just choose only one standard, they must face the risk and all limits related to that one standard. Mapping between often used standards are helpful.

Chapter 4. Design of the Web Services based metadata mapping tool for Ubilearn

Considering that the DC and LOM are the two most important metadata standards in education industry, in the following present new system, we try to make a bridge that connects both the DC and the LOM user groups together by implementing a mapping between them. Our first concern is the reusability, interoperability and durability of the presenting system to other systems that have already existed or will come in the future worldwide.

4.1 Requirements and design goal

According to the preceding discussions about the different metadata interoperability worldwide, a solution is designed and described in this chapter. Considering that the Dublin Core and the IEEE Learning Object Metadata (LOM) are the two mostly used metadata standards in the educational industry, our solution provides a mapping between them. Others as many as possible need reuse this mapping: either through a web application or web services. We will describe the design in detail, and then visualize them with Unified Modeling Language (UML) use cases.

A. Metadata Editor design

- Support the creation and editing of both Dublin Core and IEEE Learning Object Metadata
- Platform independency: it should be a web based editor
- Design the user interface for DC editor according to DC scheme structure
- Design the user interface for LOM editor according to LOM scheme structure
- Create Dublin Core XML data according to DCMI specification

- Create LOM XML Data according to IEEE LTSC LOM specification
- Save DC xml Data to a Java object
- Save LOM xml Data to a Java object

B. Web application design

- Store new DC record to DC database
- Store new LOM record to LOM database
- Search Learning Objects created by DC scheme by using DC editor
- Search Learning Objects created by LOM scheme by using LOM editor
- Integrate into Ubilearn e-learning system

C. DC-LOM mapping Web Services design

- Conversion from DC to LOM
- Conversion from LOM to DC

D. Display style design

- Design DC interface for displaying an enumeration of DC data
- Design LOM interface for displaying an enumeration of LOM data
- Transform LOM records to DC scheme interface
- Transform DC records to LOM scheme interface

From the above design description, the use cases of metadata mapping tool system are illustrated as follows:

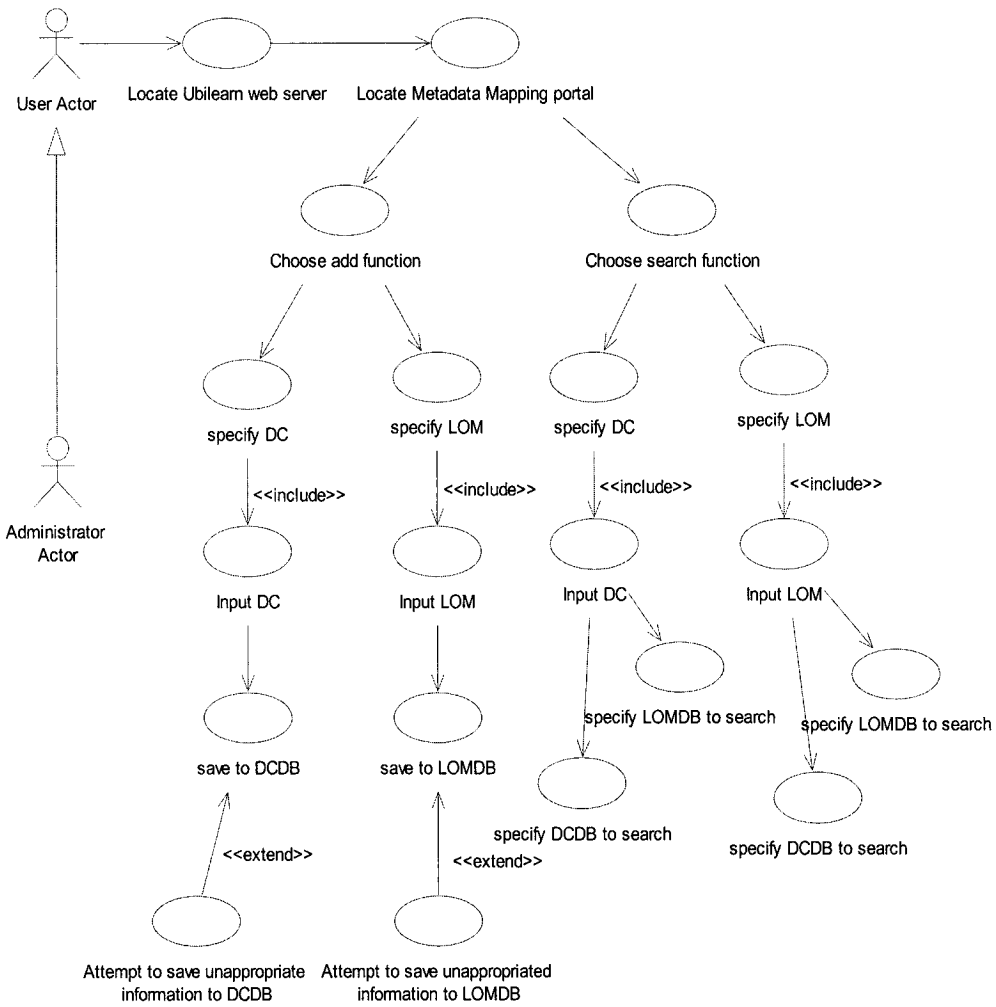


Figure 4-1: Use Case: Metadata mapping tool between DC and LOM

The "Input DC" use case in the Figure 4-1 can be further extended in the following Figure 4-2. It describes the process of collecting a DC record from user interfaces. The interfaces for the fifteen DC elements are presented in three consequent pages: the DC Content page, the DC Intellectual page and the DC Instantiation page. This process is a fundamental process that can be used in both saving a DC and searching DC records for DC scheme users.

One can work on any page and jump to another at any time. The information collected from any page should be saved temporally before moving on to another page or the information should be dumped in case of user choose to go back to the main page. Moreover, it is not necessary for a user to fill every field from scratch. The system provides default value for some fields and provides combo list for valid vocabularies according to DC standards. Finally, due to the stateless nature of HTTP protocol, state management mechanism should be provided to ensure that all these elements in different pages persist as a whole: When user navigate from one page to another, the information on the previous page should be saved in the user's session (not in database); when user choose to return to the main page or changing to the LOM standard, the session information should be refreshed.

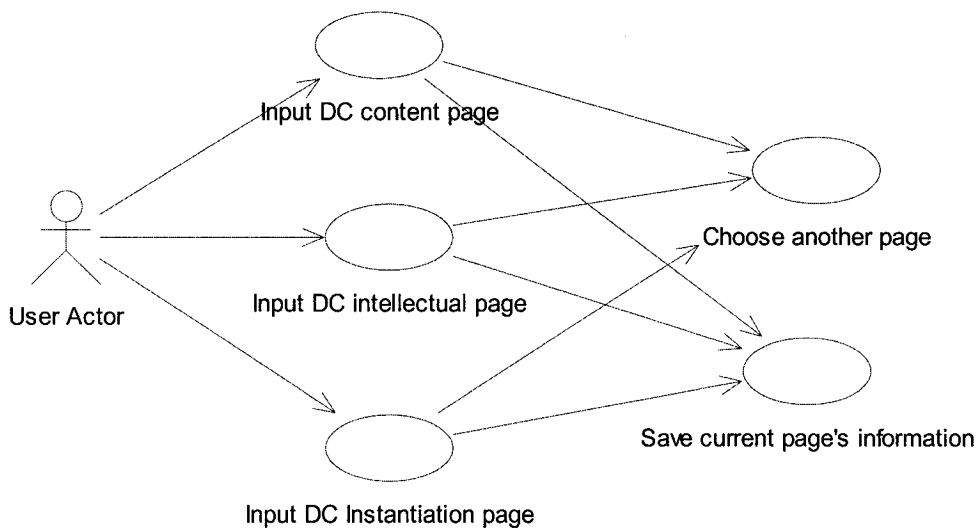


Figure 4-2: Use Case: Input DC

The similar mechanism goes for "Input LOM", which is a fundamental process for both saving and searching LOM records for LOM users. It is illustrated in Figure 4-3 as follows:

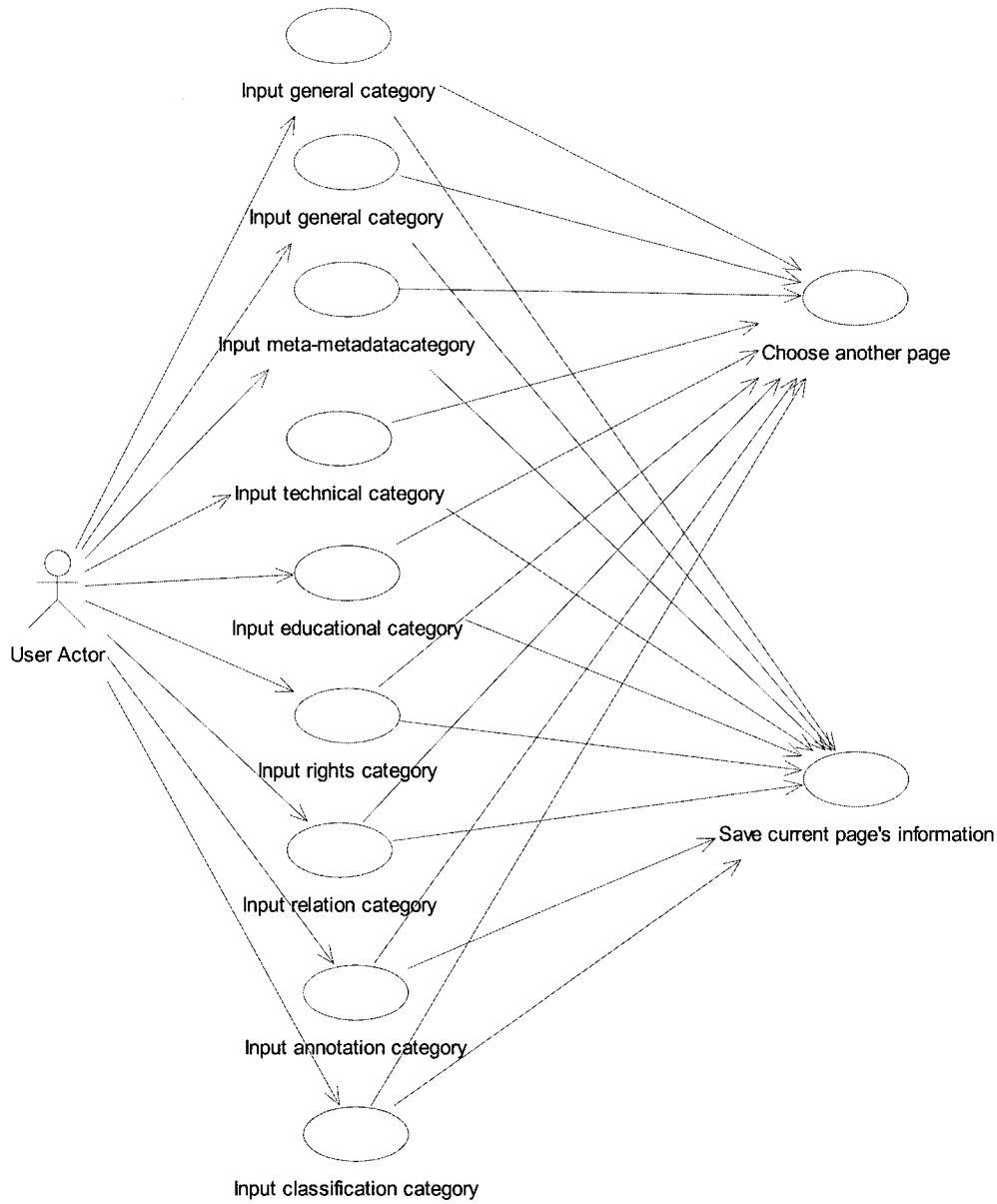


Figure 4-3: Use Case: Input LOM

The sequence diagrams used to describe these important processes will be discussed at later section in this chapter (section 4.3.2).

4.2 System architecture

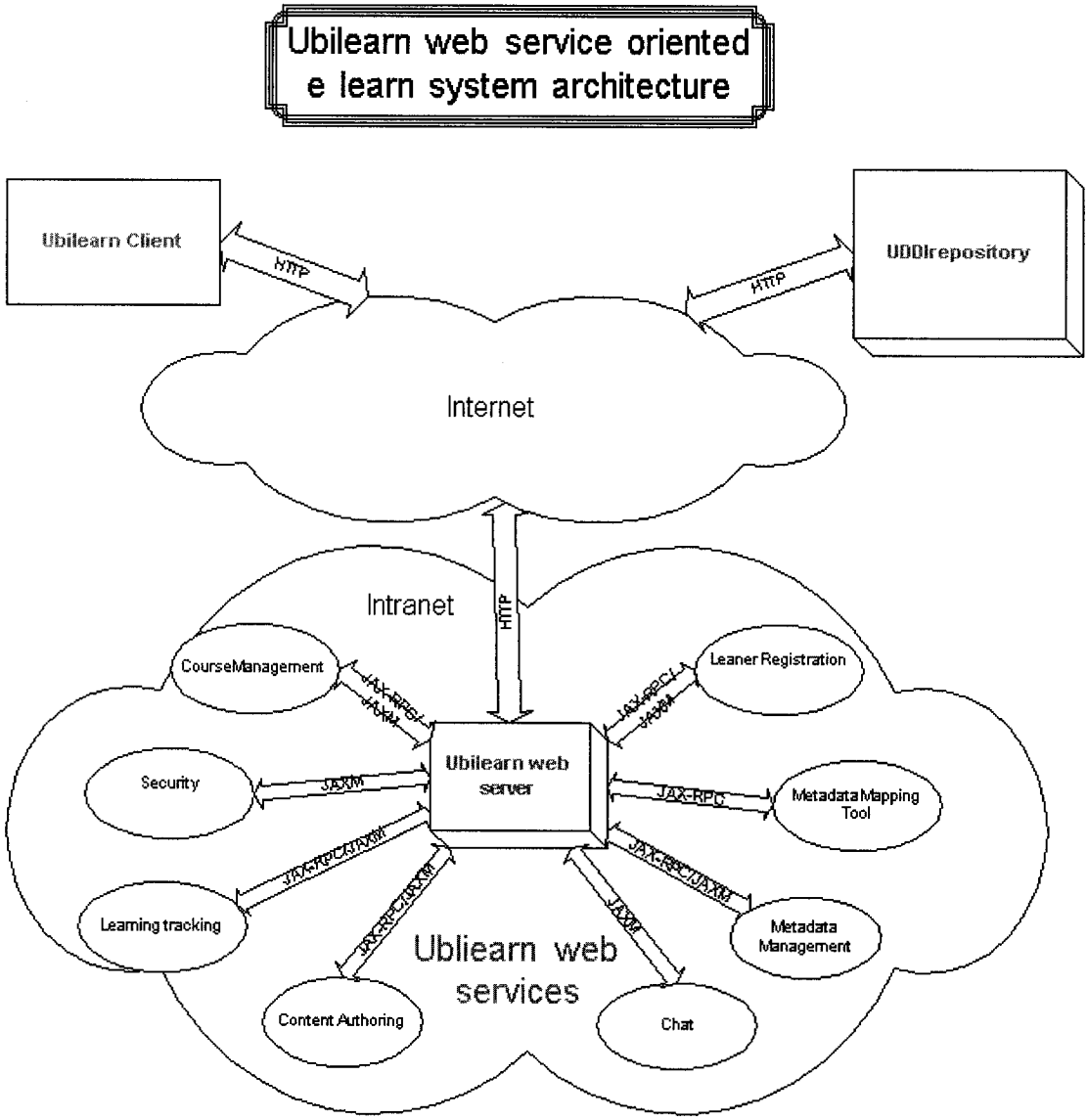


Figure 4-4: Ubilearn Web Services oriented e-learning system architecture

This section gives a description of the environment of the Ubilearn Metadata Mapping Tool. The system takes advantage of Service Oriented Architecture (SOA). And the services are implemented using Web Services technology.

Service Oriented Architecture views each software component as an atomic service. The integration of different software components is simplified by wrapping these self–contained software components under a document interface that acts as a contract between the service provider and the service subscriber. Another characteristic of SOA is that there is an information bus connecting every service and its client in order to facilitate data exchange between them.

Service Oriented Architecture exists for a long time. It is a logical extension of the object-oriented concept and the component oriented concept. Every service is made up of several components and every component is made up of several objects. As computer systems are getting more and more complex, software system become more and more complex as well. Various software venders manufacture countless software products, software developers have noticed that many same or similar software functionalities are written into different software products. It is economically sensible to develop new software product based on the ones already available in the market. In general, the larger the building blocks and the more complex it is internally, the more effort it requires in the architecture and design. However, the more sophisticated the building block is, the more powerful it will be in terms of reusability. Nevertheless, some experts argue that there are still some shortcomings in service-oriented architecture comparing to component-oriented technologies, such as performance, security and immaturity.

Service Oriented Architecture systems gain popularity with the advent of Web Services. Web Services make reuse a simple job as services expose an interface contract. This contract defines the behavior of the service and the messages it accepts and returns. Web Services provide high level of interoperability as any client accesses the service any time by subscribing to it without knowing its implementation details. Web Services technology reduces the risks, time and cost for software development and deployment as vertical knowledge is required for developing specific type of service. It would be better to leave that development work to those who focus on the specific industry.

E-learning is a big goal for various associations, research institutions, and universities, and so on. It is impossible for any one single organization to achieve success by itself.

We are facing cooperation in distributed environment, different platforms, protocols, application integrations, various devices and the Internet. Based on the concerns above, we have good reasons to adopt Web Services oriented architecture for our web application.

The system can be used as a traditional web application to users who want to create metadata for their learning objects using DC or LOM. They can use our DC or LOM editor and store their metadata data in our metadata repositories. The web application is also designed for users who want to search our metadata repository for learning objects. The details have been given in requirement A, B and D (Section 4.1). In this case, user visits our web site by our URL and application context directly.

The system also can be used as Web Services to any other applications, which only focus on one metadata scheme, either DC or LOM. The application should make a call to the UDDI repository first to retrieve the detail contract information about our services and integrate our services as parts of their systems. (Generate WSDL proxy class and so on), even if the two systems were built using different technologies, or deployed in different platform, they can interoperate seamlessly by using SOAP message over HTTP. In this case, user consume our service by invoking our interface and get response through messages. The details have been given in requirement C and D (Section 4.1).

4.3 Three-tier web application design

A web application of DC and LOM record collection and database retrieval, as well as Web Services based DC-LOM metadata mapping system prototype is designed and built and to demonstrate the process of metadata conversion. Figure 4-5 illustrates the high level design of the prototype architecture.

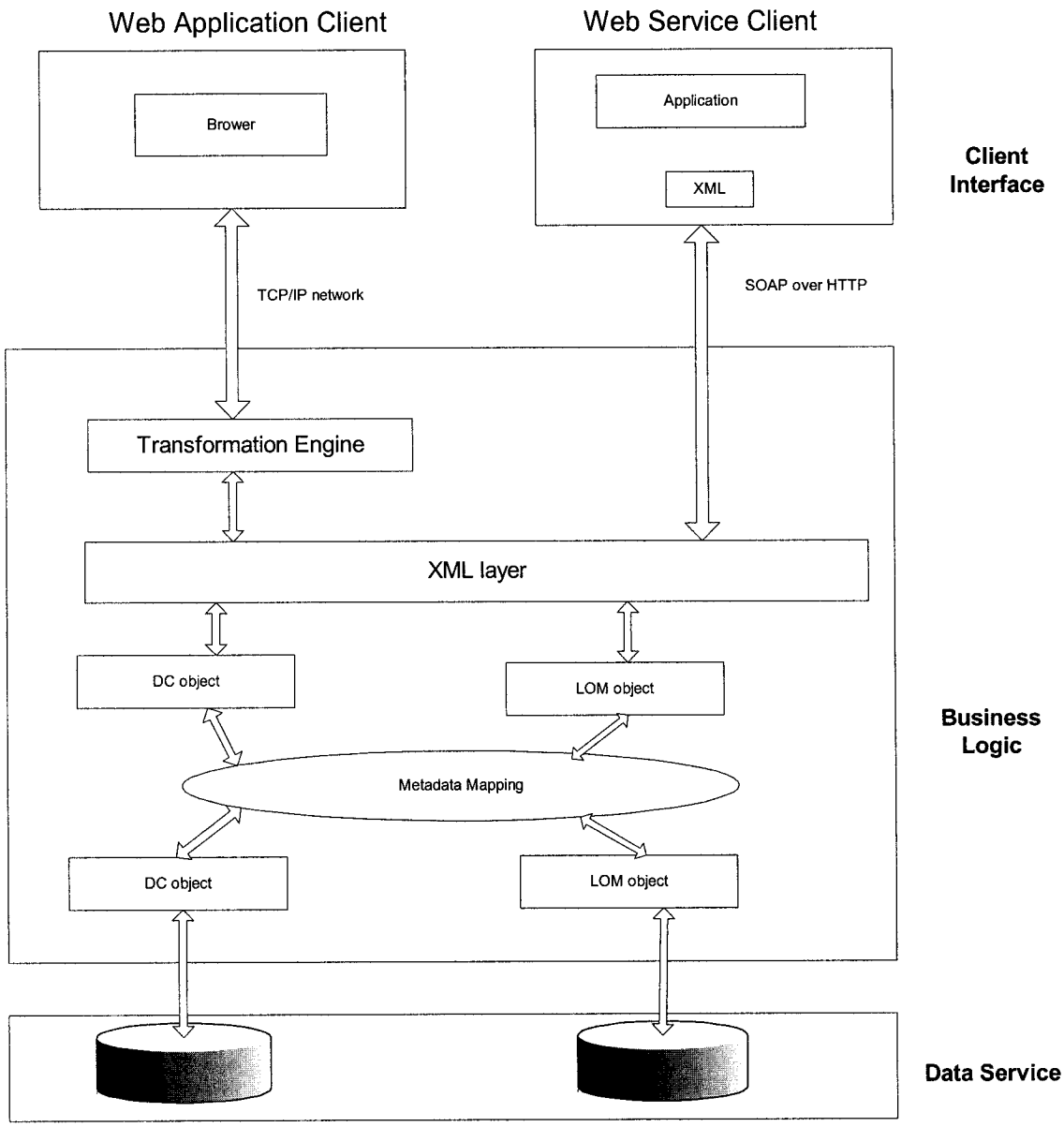


Figure 4-5: General architecture of Metadata Mapping system

This architecture is built up using three-tier model: client interface, business logic and data service. The client interface tier provides two choices to users: One is web application client, where client uses the browser to send HTTP requests and receive responses from the server using HTML or JSP pages. Another type of client could be any application that uses our Web Services by sending SOAP messages over HTTP to our Web Services. The business logic tier solves the metadata mapping between two

metadata standards using java Servlets and JavaBeans. The Data Service tier uses JDBC through JBDC_ODBC Bridge for database access.

4.3.1 User Interface

The web based application client requires a user-friendly interface within the browser. The initial page provides user with the choice of Metadata standards (DC or LOM). After a metadata standard been selected, a menu will be prompted for user to select a database corresponding to searching learning objects using the selected metadata standard. Then a different set of pages will be rendered for user data entry according to different metadata they choose and the database they prefer to search as well.

4.3.2 Business logic

Just as the platform and application framework works as different tiers, metadata mapping application business logic is also benefit from multi-tier architecture. The division between tiers in our business logic tier is between functions that are specific to particular client and those works for all application.

The business logic design is based on the Model-View-Controller (MVC) design pattern. MVC is recommended for interactive application by J2EE tutorial [28]. In MVC, three parts: the Model, the View and the Controller, have been used to decouple the functionality of the application.

The Model stores the rules such as how to map from one metadata object to another kind of metadata object. Both web application client and Web Services client share a common set of basic functions of semantic metadata mapping as illustrated in Table 4-1; and how to transform the search research result from database to user preferred presentation. It also stores the business data as well. When user requires manipulating the data, such as insertion or deletion, the model handles this action. It also deals with specific logic, such as generate dynamic content according to user query.

The View could be JSP pages, Servlet or static material such as HTML pages or graphics, etc. Since web browser is lightweight client, styles to render dynamic content should also be involved here. For Web Services request, SOAP messages or XML RPC will be used.

The Controller handles all HTTP requests from all kinds of clients and processes the screen flow in a centralized way. This single point of dispatch makes it an ideal place for facilitating security and logging, at the same time, simplifying the client request by using a single URL. All of the screen flow are settled by the controller instead of hard coded, therefore decoupled those Sevelets and JSPs work cooperated to display the view to user.

Table 4-1: DC-LOM semantic mapping

Dublin Core	IEEE LOM
Identifier	General.Identifier
Title	General.Title
Language	General.language
Description	General.Description
Subject	General.Keyword
Coverage	General.Coverage
Type	Educational.LearningResourceType
Date	LifeCycle.Contribute.Date when LifeCycle.Contribute.role='Publisher'
Creator	LifeCycle.Contribute.Entity when LifeCycle.Contribute.role='Author'
Contributor	LifeCycle.Contribute.Entity with LifeCycle.Contribute.role
Publisher	LifeCycle.Contribute.Entity when LifeCycle.Contribute.role='Publisher'
Format	Technical.Format
Rights	Rights.Descriptiion
Relation	Relation.Resource.Description
Source	Relation.Resource when Relation.Kind='IsBasedOn'

The following sequence diagram describes how to collect data from JSP pages and servlets to get a DC object.

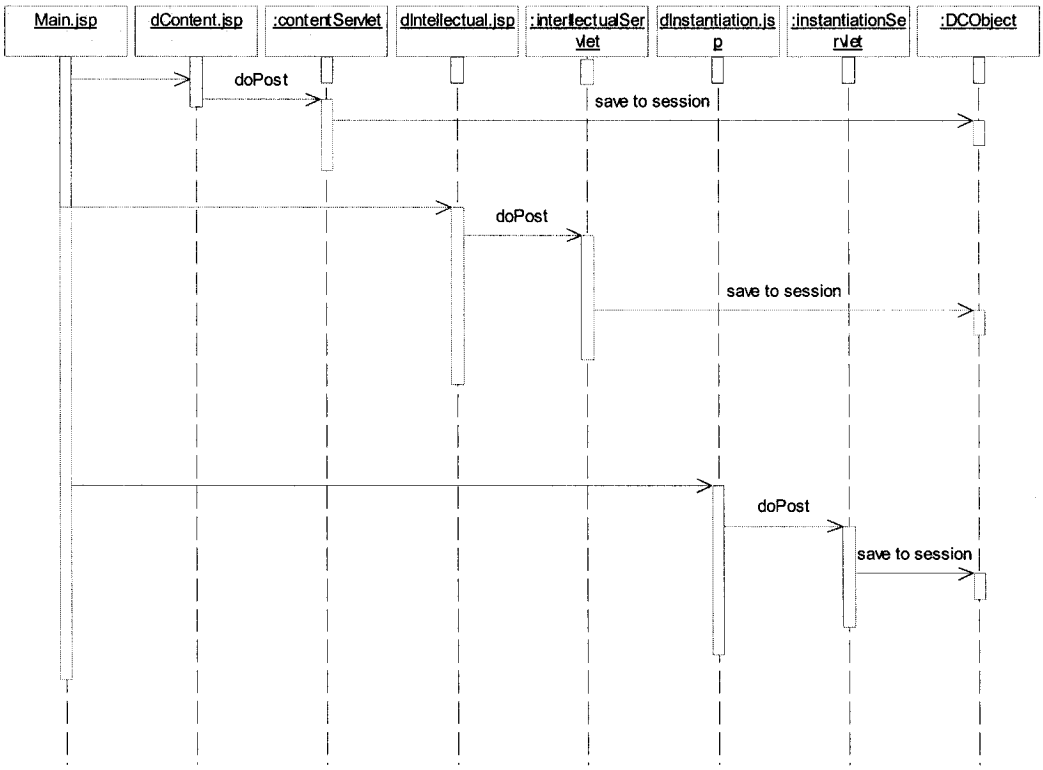


Figure 4-6: Sequence diagram: saving DC Object

Similarly, the following sequence diagram describes how to collect data from JSP pages and servlets to get a LOM object.

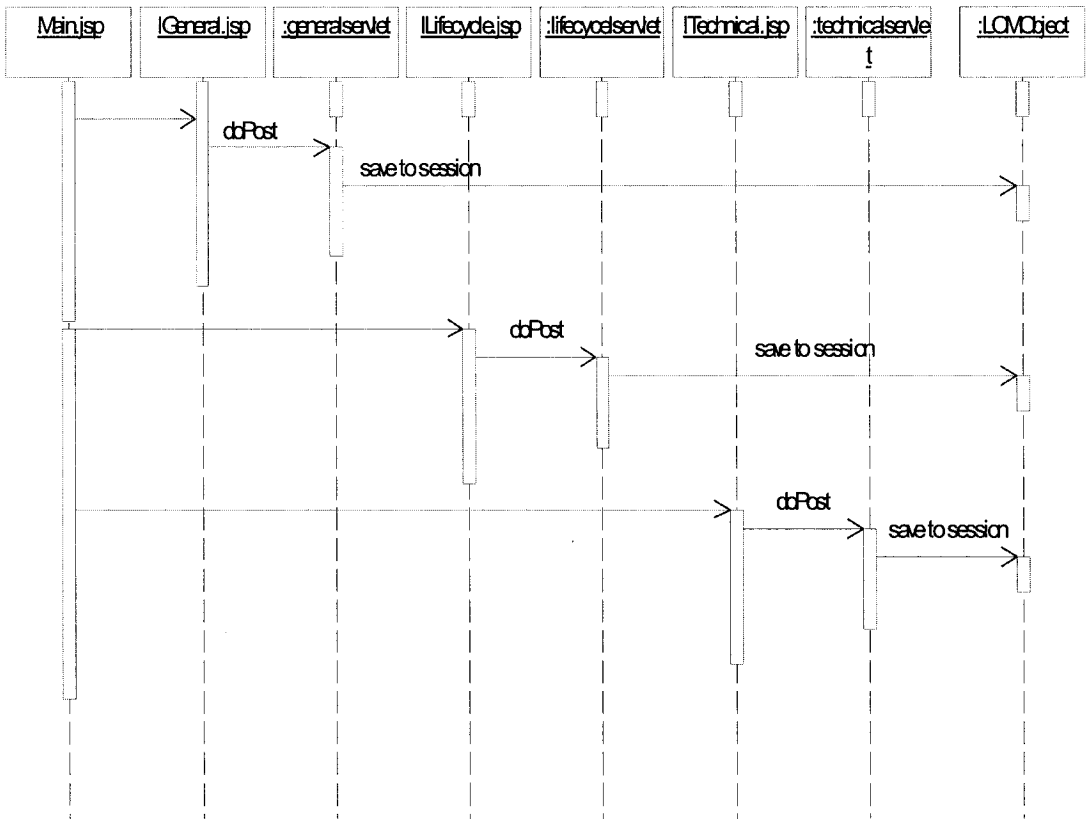


Figure 4-7: Sequence diagram: saving LOM Object

Figure 4-7 only shows the first three categories out of nine categories of LOM specification. The mechanism of saving the other categories is the same for every category. Nine Servlets are given to nine JSP Pages respectively. The information is saved to LOMObject within session scope.

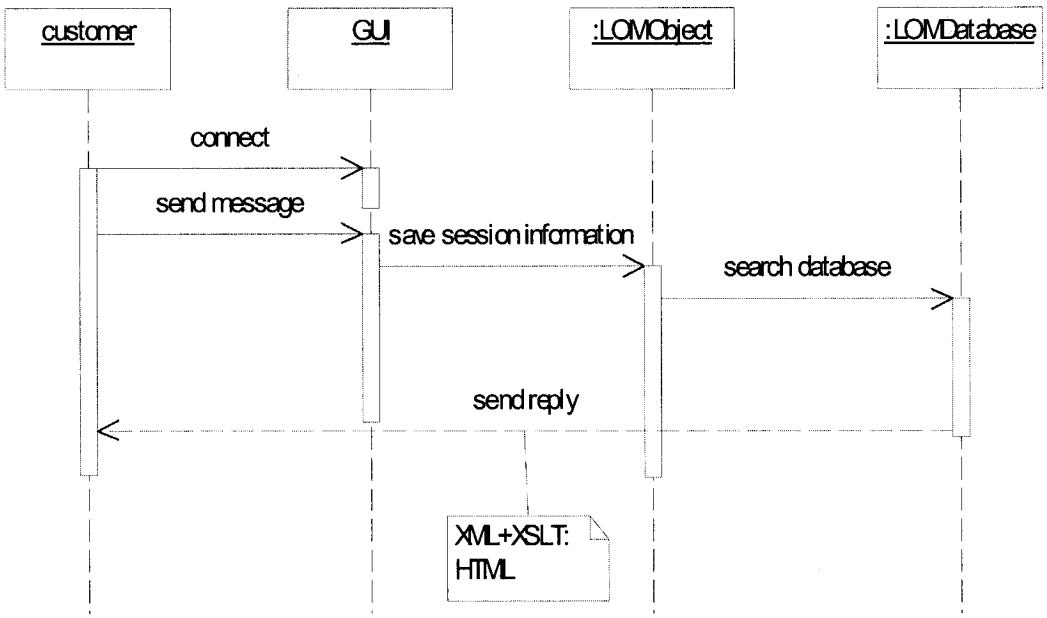


Figure 4-8: Sequence diagram: search LO using LOM by LOM scheme

4.3.3 Data repository

A relational database management system is used to store and manage persistent data for our mapping system. We can focus on the definition of every element in every metadata standard, their restriction and relationship between some of them. The RDBMS provides services to the data service: retrieving database records, inserting new record, predefine statement, stored procedure, view and so on through JDBC APIs for synchronous data service tier and the database engine.

4.4 Web Services design of the mapping tool

The design of metadata mapping Web Services involves cooperation between three business roles: the Metadata Mapping Web Services provider, the UDDI repository broker and the Ubilearn portal requestor. The relationship between them is described in Figure 4-9.

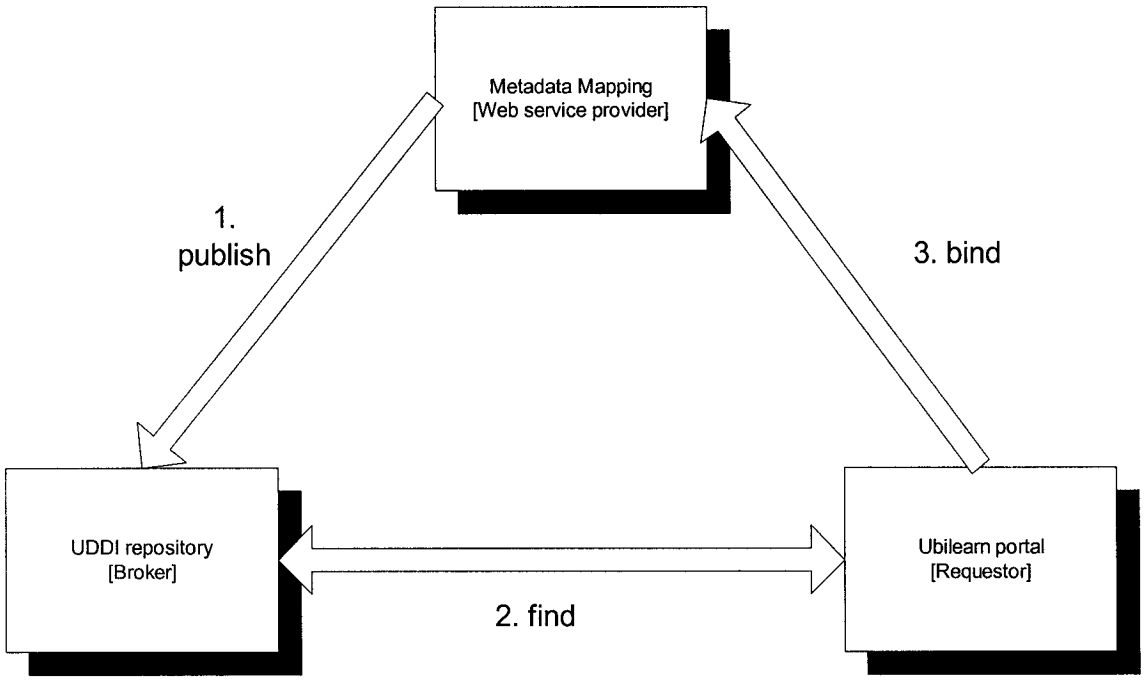


Figure 4-9: Metadata mapping business role

The detail interactions between them follow these steps:

1. "Publish" presents that the Metadata Mapping Web Services Provider implements mapping service between Dublin Core and IEEE LOM based on standard protocols: HTTP, XML, Simple Object Access protocol (SOAP). It also responsible for describing itself using WSDL and publishing on a public UDDI repository for other users to locate using UDDI. The WSDL document contains characteristics and metadata about the metadata mapping tool that useful for describe, discover and integration.

2. "Find" presents that Ubilearn web server (the Requestor) uses JAXR to communicate with the Metadata Mapping Web Services (the Web Service Provider), and after receiving the WSDL document for the metadata mapping web service, the Ubilearn web server make necessary arrangements for invocation the

web service, include choosing JAXM or JAX-RPC, synchronous or asynchronous, XML message format, and so on.

3. "Binding" means that first, the Ubilearn server encapsulates the metadata record (represent DC or LOM) into a SOAP message then sends the request and waits for the response. Second, the metadata mapping Web Services do the conversion and then return a XML document representing the counterpart metadata form for the requesting metadata sending from the SOAP message. Third, upon receiving the responses, the Ubilearn server processes the metadata set out of the SOAP message. Finally, when a preferred form of metadata record is received, databases exploration and XSL transformation can be made to the results to generate HTML using user preferred XSLT.

This scenario is illustrated in the Figure 4-10:

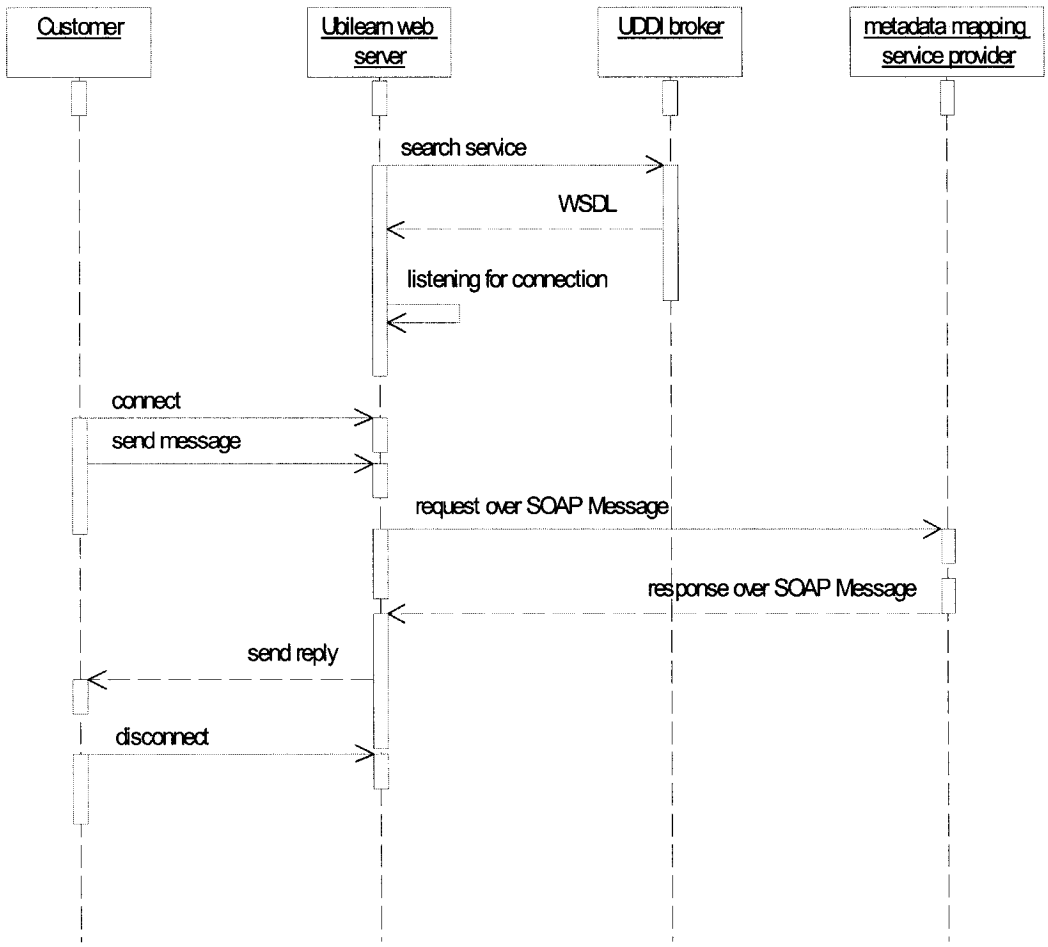


Figure 4-10: Sequence diagram: interaction among business roles

The metadata mapping Web Services provides the conversion between DC and LOM: either from DC to LOM or from LOM to DC. In the case of that users use DC as their choice, the DC editor is designed for their metadata creation and searching interface. Users send requests to the Ubilearn web server through browsers. From the information collected from the interactions between the users and the DC interface, a DC object has been generated on the Ubilearn web server side. It is assumed that metadata mapping web services has published its services on the public UDDI repository and the Ubilearn web server has found the service. Therefore the Ubilearn server can call the DC to LOM service using the DC object, which is encapsulated in a SOAP message in

DC2LOMRequest. The metadata mapping web services are waiting calls from its service requestor: Ubilearn web server. So it can build a connection after receive the requesting SOAP message. Then it calls the dcParser to parse the DC object out of the message, and converse from DC object to LOM object. At last, the metadata mapping web services send the LOM object back to Ubilearn web server through a SOAP message. The DC2LOMRequest take charge of extracting LOM object from the coming SOAP message.

The interaction between the Ubilearn web server and the metadata mapping Web Services is described as following Figure4-11:

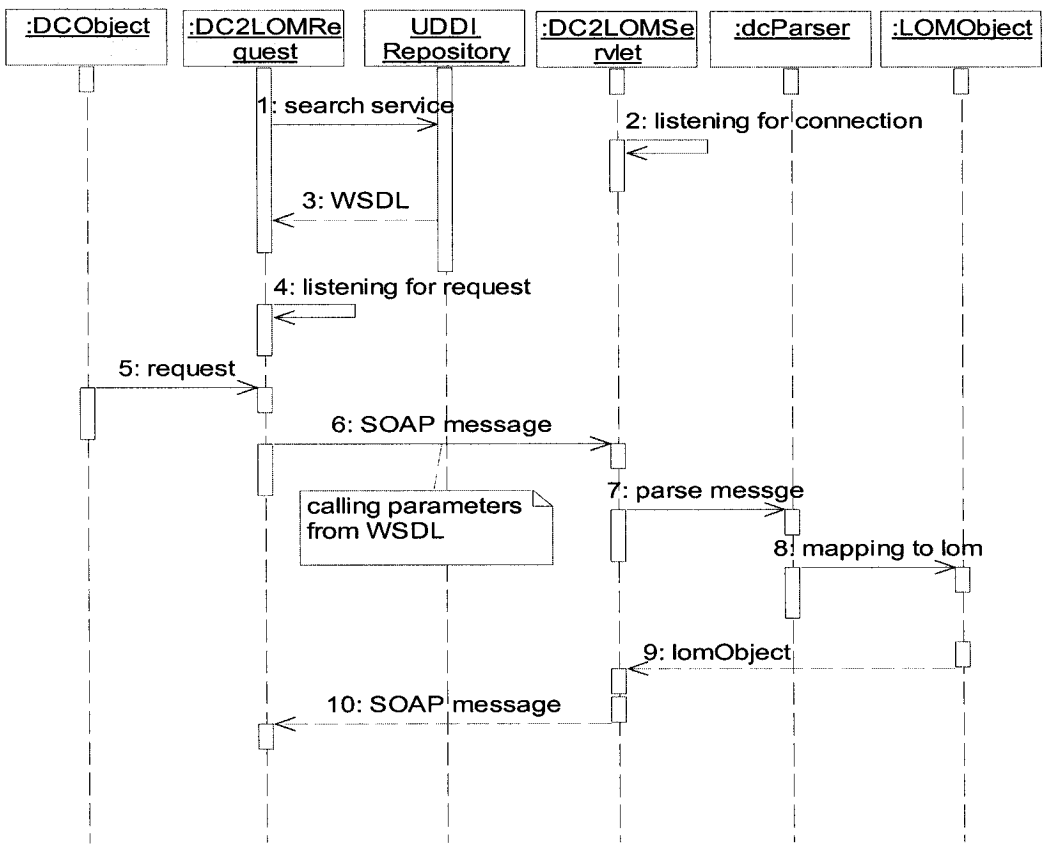


Figure4-11: Sequence diagram: metadata mapping Web Services

Chapter 5. System Implementation

The implementation details will be described according to three-tier architecture: user interface, business logic and data repository. Web Services technology is used to build business logic component.

5.1 User Interface

User interfaces are becoming more and more important to software design and implementation. It is reported that almost half of the work down is involved in user interface design for a web-based application [29] .

Web applications are thin client in nature, they require no or minimum client installation beside a browser. They present information to users and manage the interactions between users and the computer system. For applications focusing on the interactions, the pages shown on the browser are organized to help users do they want to accomplish with the specific application system.

Technologies like HyperText Makeup Language (HTML), Cascading Style Sheet (CSS), eXtensible Makeup Language (XML), Extensible Style sheet Language (XSL), Java Server Pages (JSP) and JavaScript are used to build our metadata mapping tool editor, search interfaces and search results pages.

HyperText Transfer Protocol (HTTP) is commonly used for communication between client and server over the Internet. HTTP defines request and response format. Each request consists of a request method, a request URL, header fields, and a body. Each response contains a result code, header fields, and a body. The most widely used request methods are GET and POST. GET is used for retrieving the resources identified by the requested URL while POST is used for sending data of unlimited length to the Web server.

JSP are text-base documents. A JSP page contains two parts: one is static part, which is created by markup language and handles static structure and content of a page. The other one is dynamic part, which is created by JSP tags and scriptlets and handles embedded application specific business logic. JSP pages often are seen as a template-based content generation method.

The following diagram describes web site information structure of metadata mapping tool.

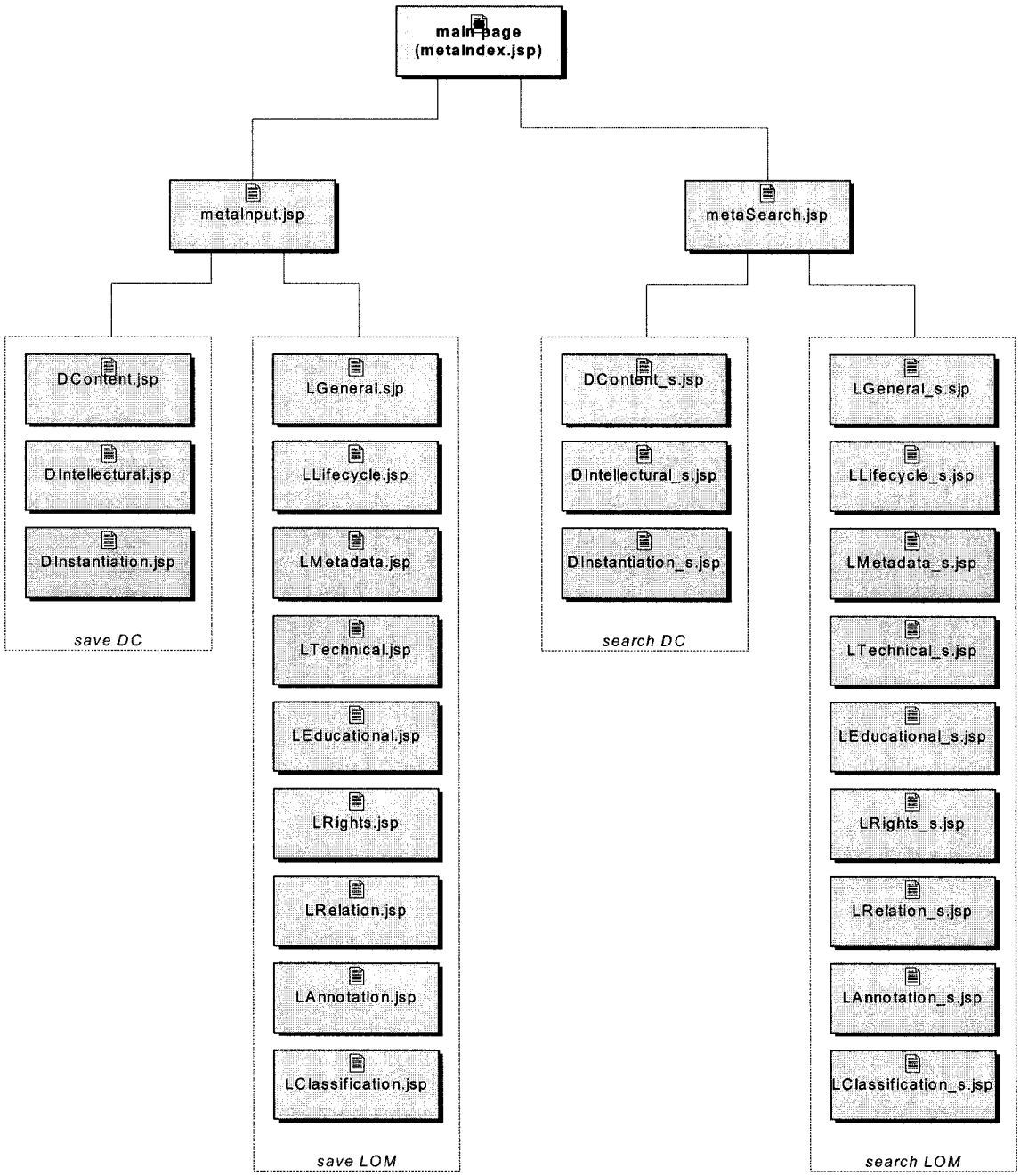


Figure 5-1: Interface structure of metadata mapping tool

The following picture shows the main page for metadata mapping tool interface.

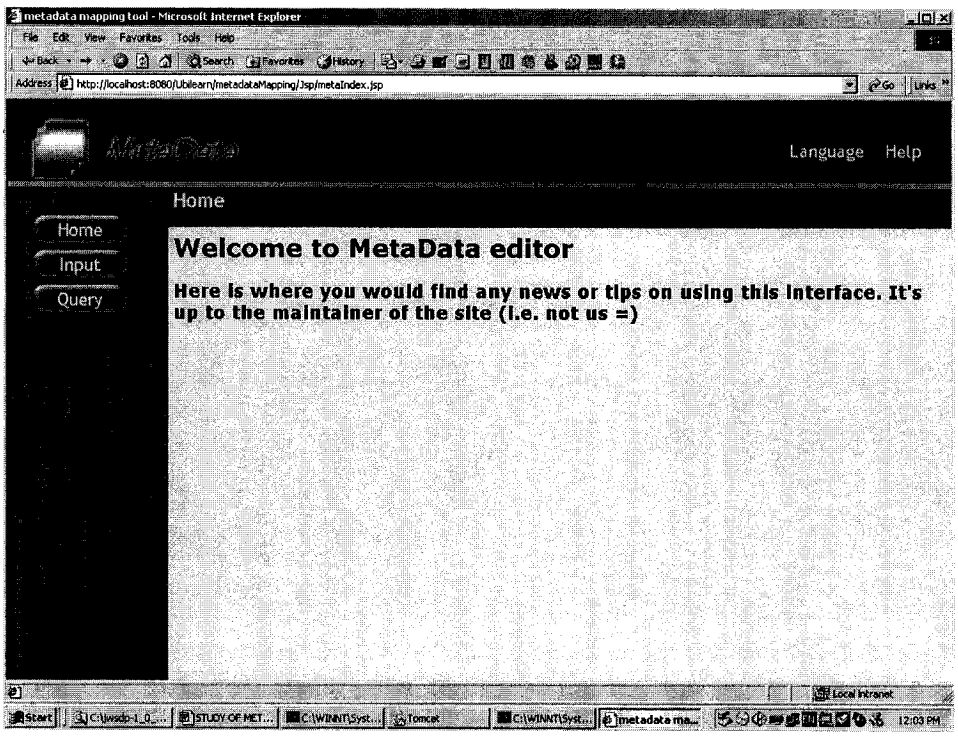


Figure 5-2: Homepage of metadata mapping tool

From the main page, we can choose either "input" or "query" from the menu. The input function will lead to a new DC or LOM record creation process using our metadata editor. The query function will lead to a search process according to the input information.

The follow figure shows the interface of DC editor:

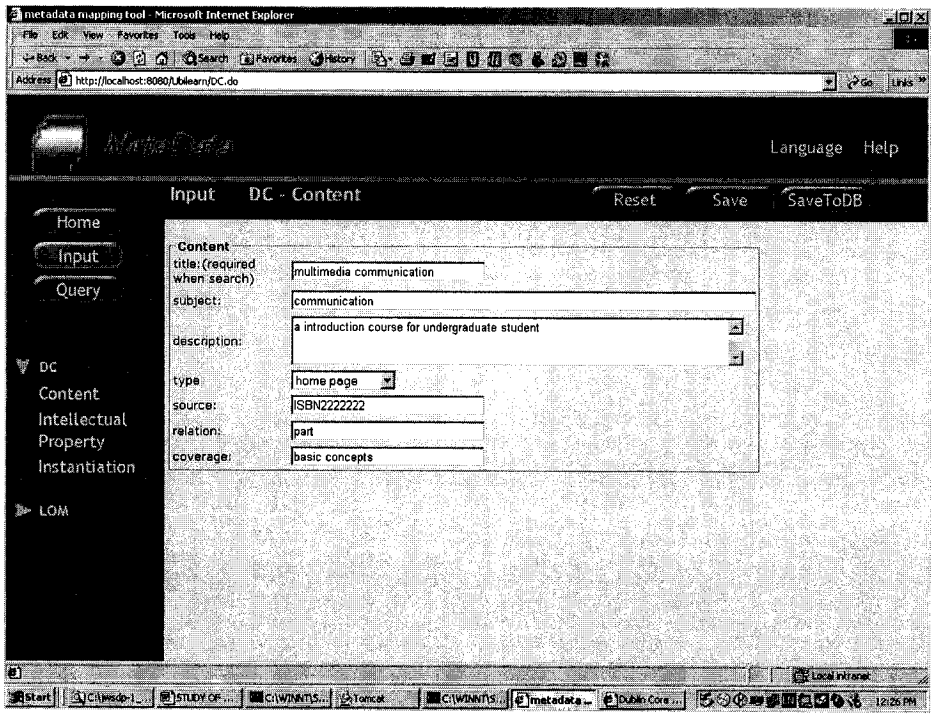


Figure 5-3: Dublin Core editor (the first of the three pages for DC)

Similarly, LOM editor is illustrated as follows:

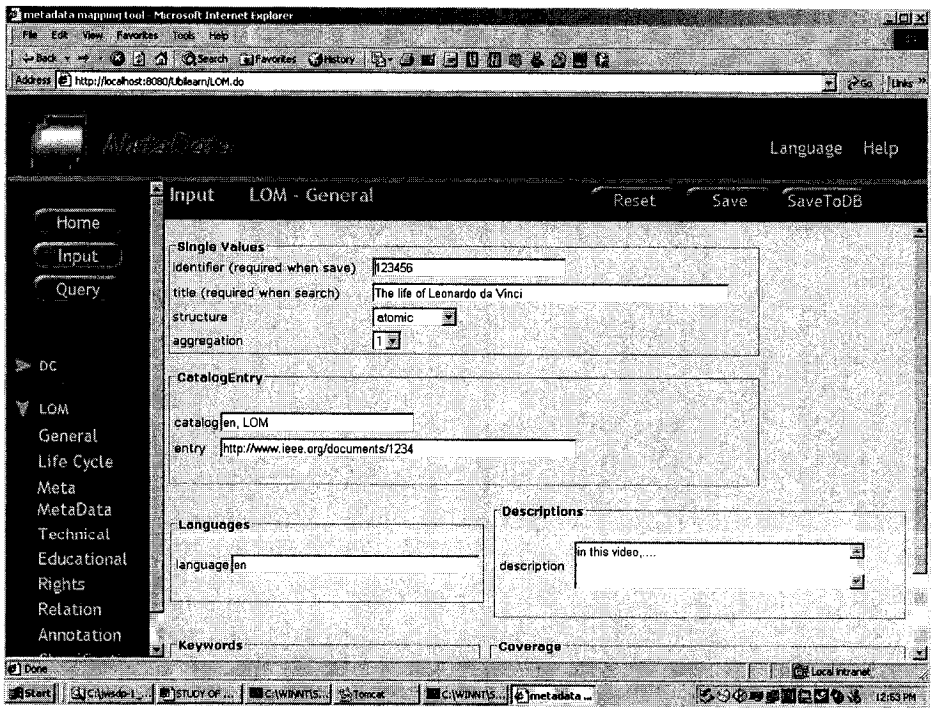


Figure5-4: LOM editor (the first of the nine pages for LOM)

Since both the DC and LOM are made of a number of elements that are too much to fit in one web page, they need a style mechanism to give these different pages a consistent layout. Cascading Style Sheet (CSS) is a simple style sheet mechanism that allows attaches style to HTML documents. Style Sheets act as a "template" that can control the layout and design of displaying pages. They save in separate files from the HTML pages, giving consistent display and easy maintainability across several pages in one application. Visual design issues, such as fonts, colors and page layout, can thus be addressed separately from the web page logical structure.

Apache struts are also used for decoupling the business logic and user interfaces, we will discuss this in detail next section.

Moreover, XML are also used to improve the interface design. XML is created for describing data. It is not replacement for HTML which purpose is display data. Since Ubliearn system have potential to deliver data in different formats, to different browsers (such as laptop, mobile phone and PDA) and to other Web servers, we need to transform XML data into different formats. Extensible Style sheet Language (XSL) is the new World Wide Web Consortium (W3C) standard to do this job. XSL is a language for expressing style sheets. XPath plays an important role in XSL to identify particular element or parts in XML documents. In our prototype, two different XSL transforms are used on the same XML file in order to convert it into two different HTML file send to DC or LOM user's browser respectively. For example, once a XML file that represents some DC records is generated from search results, it can be converted to "DC HTML" format to DC users using "DC XML to HTML" XSL transform or converted to "LOM HTML" format to LOM users using "LOM XML to HTML" transform. Similarly, XSL can transform an XML file into different format that is recognizable to a browser in a PDA or mobile phone in the future work.

The Figure 5-5 shows Dublin Core search results that represented by a HTML file, which is made by employing a DC to HTML XSL transform onto a DC XML file. The DC XML file includes several DC records. The XSL file presents DC schema in the following defined layout: a green bar highlights the header of every DC record; every alternate fields of DC data are having different background colors.

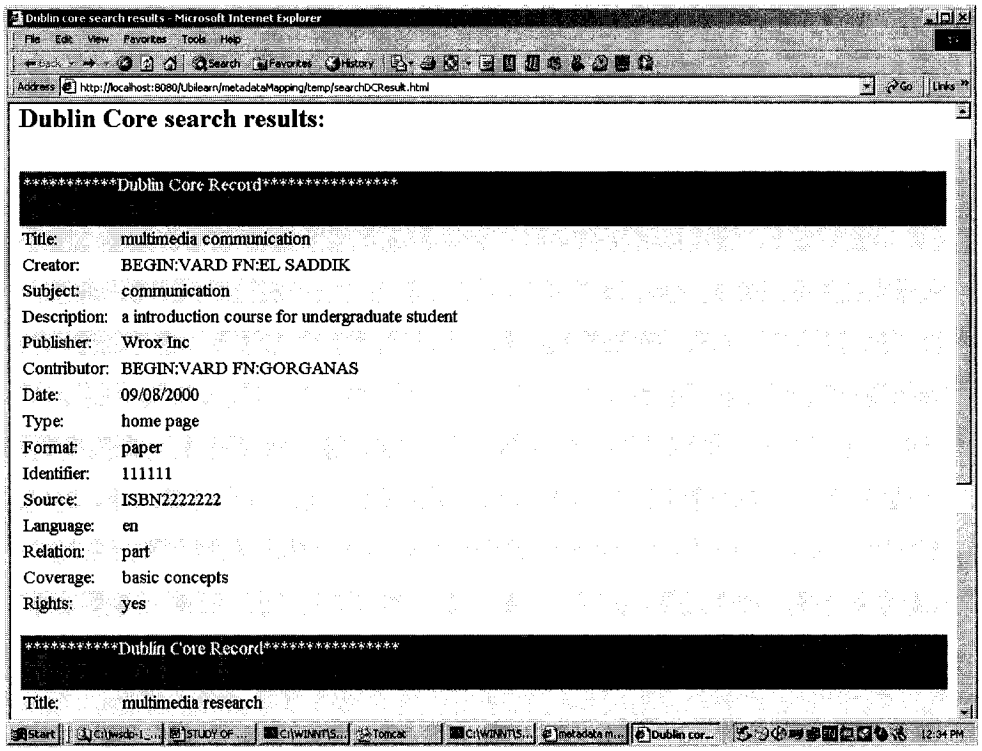


Figure 5-5: DC XML with DC2HTML XSL transform

Similarly, the Figure 5-6 shows LOM search results, which are represented by a HTML file, which is made by employing a LOM to HTML XSL transform onto a LOM XML file. The XSL file presents LOM schema in the following defined layout: the header of every LOM record are highlighted by a blue bar, every alternate groups of LOM data are having different background colors.

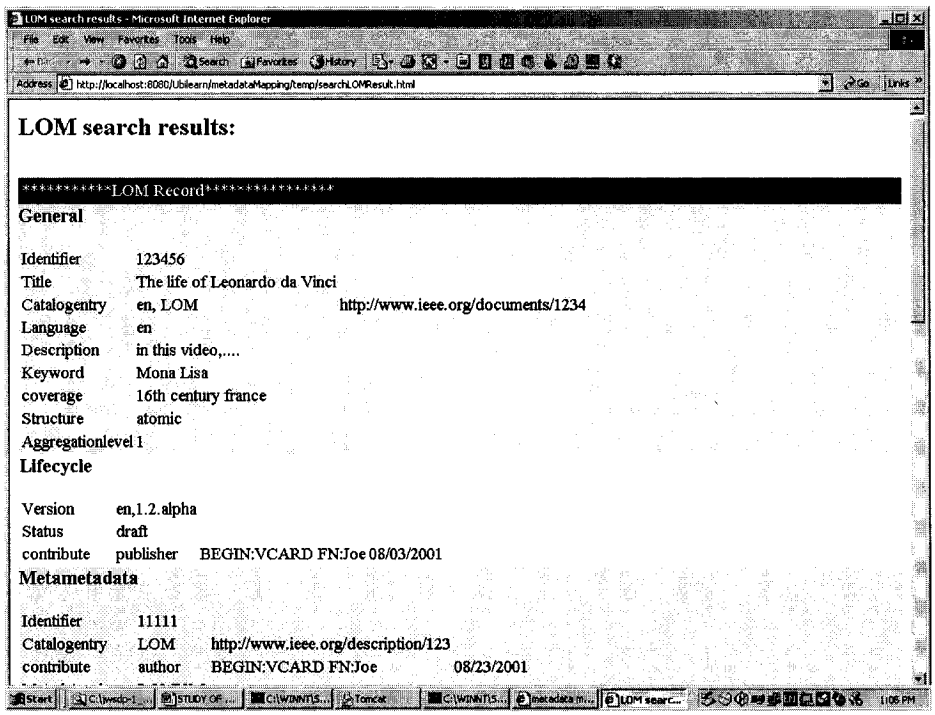


Figure 5-6: LOM XML with LOM XSL transform

5.2 Business logic tier

The Business logic tier is responsible for the implementation of the functions for web applications and extends some functions to the remote metadata mapping Web Services.

Java 1.4.1 is used as the programming language of choice in the presented work, and the Java Platform, Enterprise Edition1.3 (J2EE) provides a component-based environment to design, development and deployment of the application. Technologies like JSP, Servlet, Java Beans, Apache struts and Web Services are used to build the conversion between DC and LOM.

Both JSP and Servlet are used to handle communication between the Ubilearn web server using J2EE and users through web clients. The JSPs and Servlets take responsibility of processing HTTP requests and giving back the responses to clients, by invoking business logic and transmitting data.

Apache Tomcat v4.1 is used as the Servlet container that is used in the implementation for Java servlet and JSP. A web container is essentially a Java runtime that provides an implementation of Java Servlet API, and facilitates for JSP pages. The container is responsible for initializing, invoking, and managing the lifecycle of Java Servlets and JSP. Tomcat can either work as a standalone web server or works with other commercial web server. In this presented implementation, Tomcat is used as a standalone web server for development.

Servlets are some java code running on server side via request-response model. The Servlet API is specified in two Java extension packages: java.servlet and javax.servlet.http. They provide interfaces and classes for writing Sevlets. All Servlets must implement the Servlet interface, which define life-cycle methods.

Either Servlets or JSP pages can be used alone for both application logic and content generation. Also they can be combined. The work of displaying interface and handling business logic is quite different job using different skills, thus they should be done separately by different people in an enterprise application implementation. In metadata mapping tool for Ubilean, the two kinds of work are separated: the content generation is handled by JSP pages while application logic is managed by Servlets.

The Apache struts are used as our web application framework since it is idea to loose coupling the interfaces and the business logic parts. The whole business logic tier is further decoupled into several tiers that benefit from the struts Model-View-Controller design pattern. The struts cooperate with standard Java technologies like Servlet, JavaBean and Resource Bundle to provide the Model and cooperate with displaying technologies like HTML, JSP pages, XML and XSLT to provide the View. The struts provide a special Controller itself.

In the View, we have seen both the input and output interfaces in the previous part. There are two interfaces to end-user. If the metadata is in different schema, it is the Model's responsibility to transform the search result in form of XML using XSLT, and deliver the appropriate HTML to user.

In the Controller, a special configuration file is used for initializing resources like ActionForms in the struts just like that deployment descriptor is used for initializing resources like servlet and taglibs when servlet is executed the first time. The configuration file is presented as an XML file named "struts-config.xml" in the WEB-INF directory of our web application. It includes three kinds of tags: <form-bean>, <action-mapping>, and <global-forward>. <Form-bean> is used to collect information from its corresponding JSP pages that user interact with; <action-mapping> is used to direct request URI to server side Action; <action-forward> is used to select output pages. The controller delegates most of work to action classes.

In an ActionForm class, properties are designed to hold the state of the corresponding JSP form, and getters and setters are designed to access them. ActionForm Bean has no special function method other than properties, getters and setters, so they hold no business logic.

Action class is used to process request from client via its execute method, return an actionForward object to point out where control should be forwarded.

ActionMapping can be implemented as a Java class which instantiated ActionMapping in the struts. Or use struts component to parse XML-based description in the struts-config.xml and create appropriate object initialized to the appropriate value.

We implement struts application following these steps:

1. Create the Model – JavaBeans, XML and XSLT
2. Design the View – JSPs as input, and HTML as output
3. Implement Actions – process request from client and point out what to display to client. They connect the model and view together
4. Create the Controller – compose the configuration file for metadata mapping tool which is named struts-config.xml

In the Model, the metadata mapping tool includes 65 Java files which are divided into different groups according to their functionalities. The following figures show all seven packages for Metadata mapping tool for Ubilean.

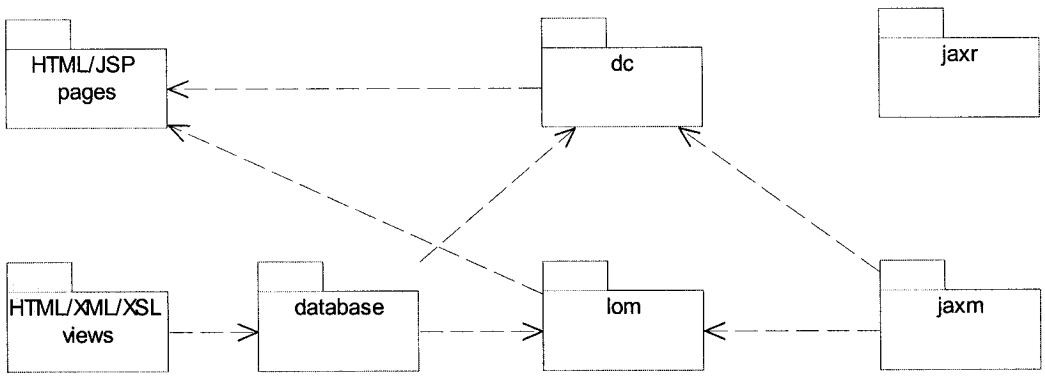


Figure 5-7: Package view of metadata mapping tool

All objects related to DC are included in the dc package. These objects are responsible for collecting information from user interfaces; saving a DC object to the DC database; searching a DC metadata from the DC database and searching a DC metadata from the LOM database either through Servlets or extended the DC-LOM conversion in the metadata mapping Web Services. The following diagram shows the DC component diagram:

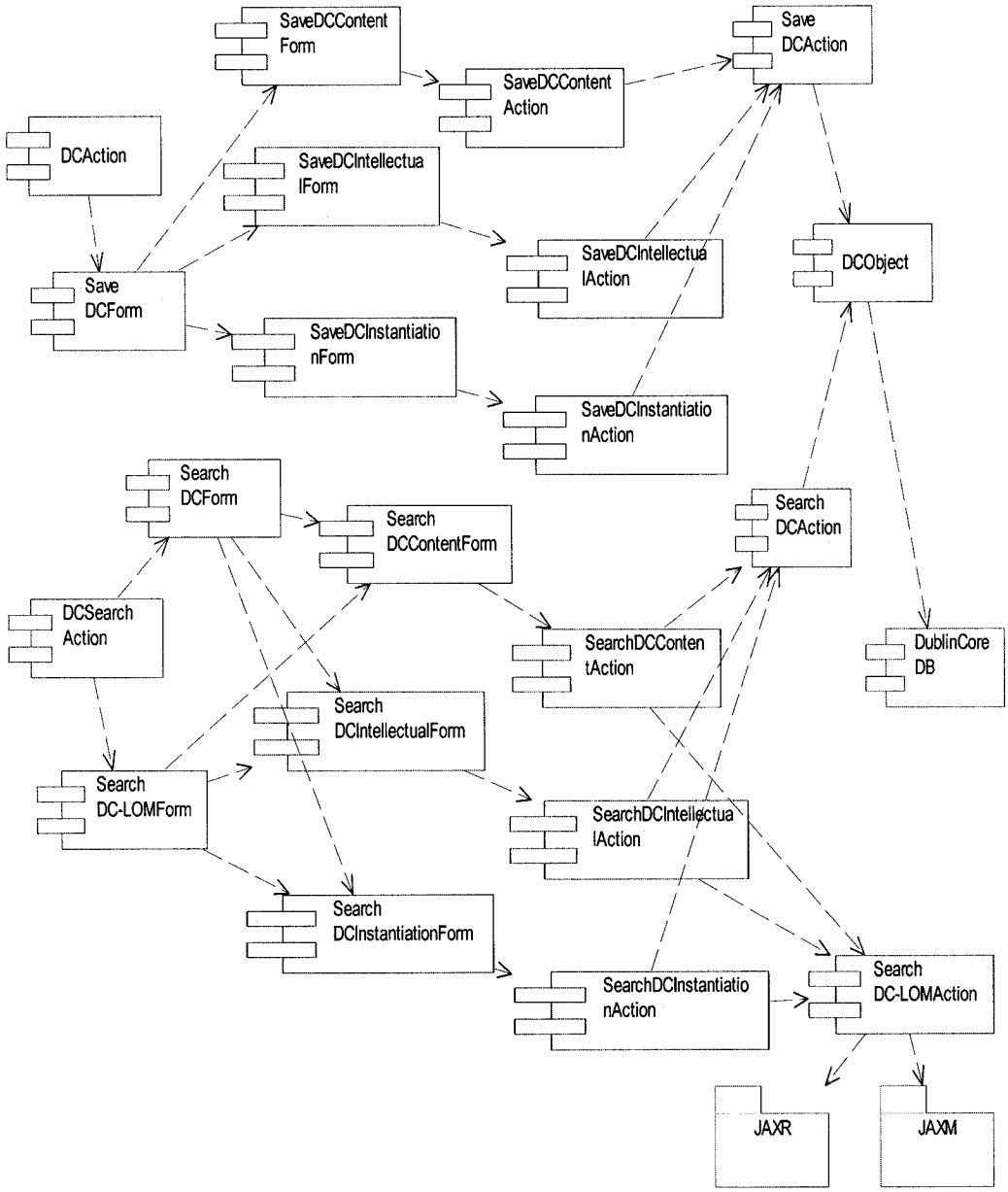
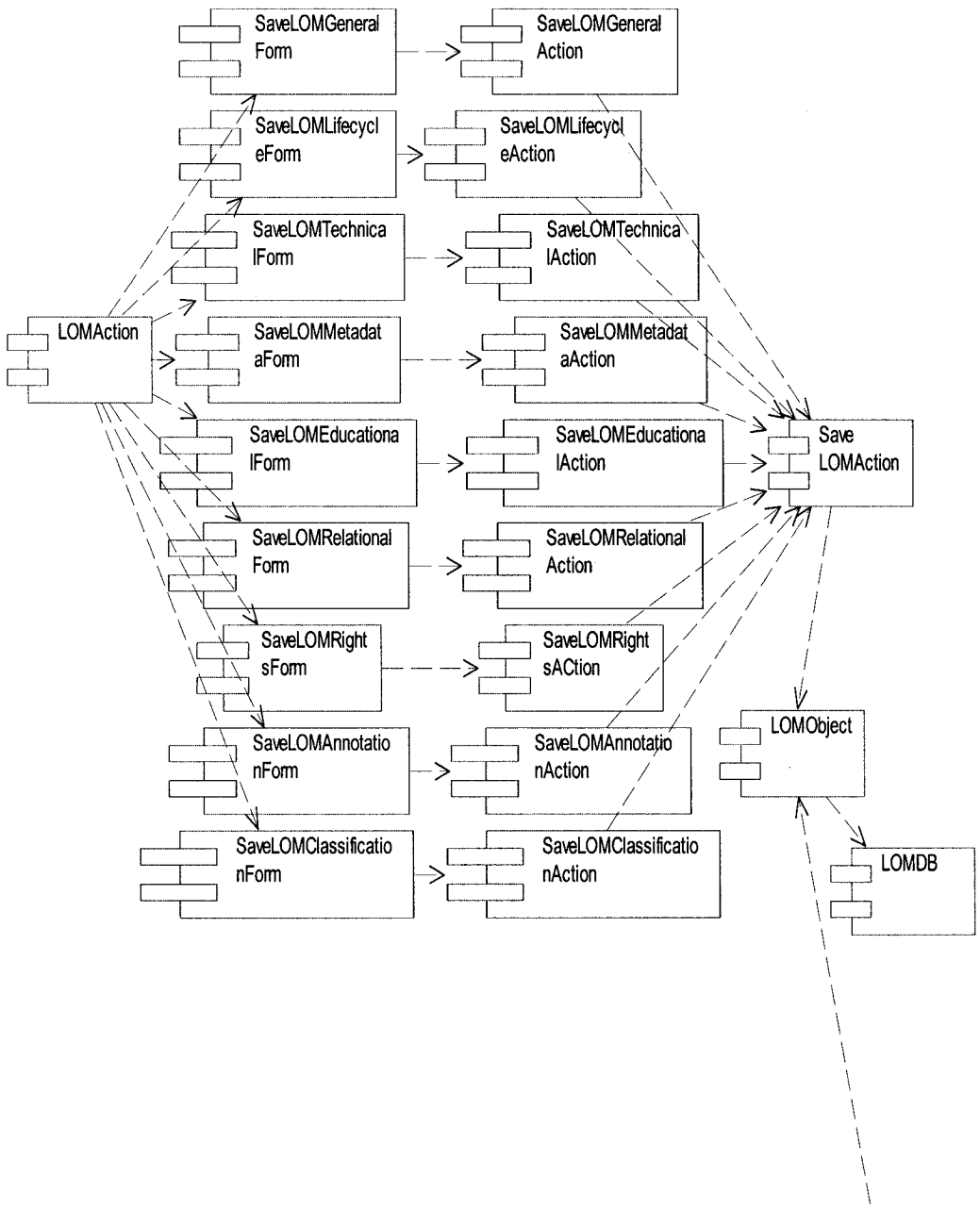


Figure 5-8: Component diagram for the DC package

Similarly, all objects related to LOM are included in the LOM package. These objects are responsible for collecting information from user interfaces; saving a LOM object to the LOM database; searching a LOM metadata from the LOM database and searching a LOM metadata from the DC database either through Servlets or extended the LOM-DC

conversion in metadata mapping Web Services. The following diagram shows the LOM component:



(This is more than half of the diagram; the rest of it is presented on next page)

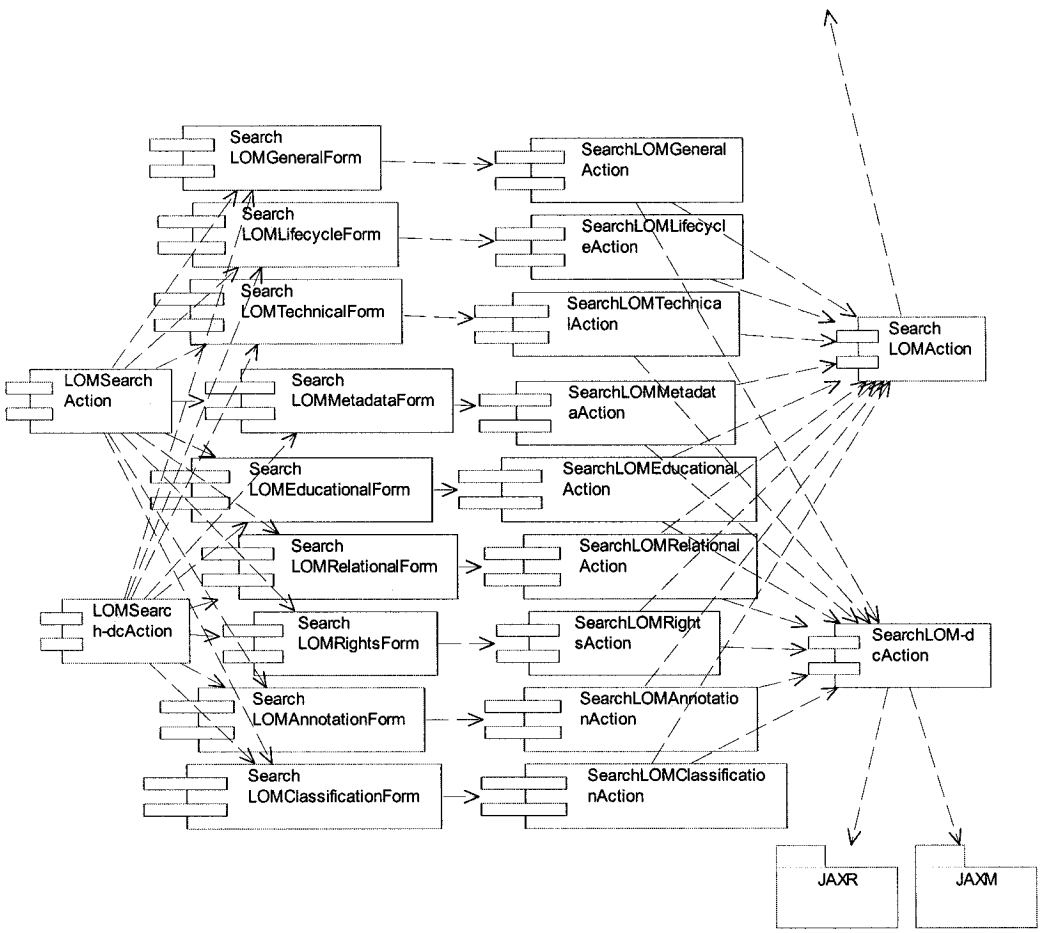


Figure 5-9: Component diagram for the LOM package

The JAXR package takes charge of publishing, modifying and removing the metadata mapping tool Web Services to the UDDI repository. The following diagram shows the JAXR component:

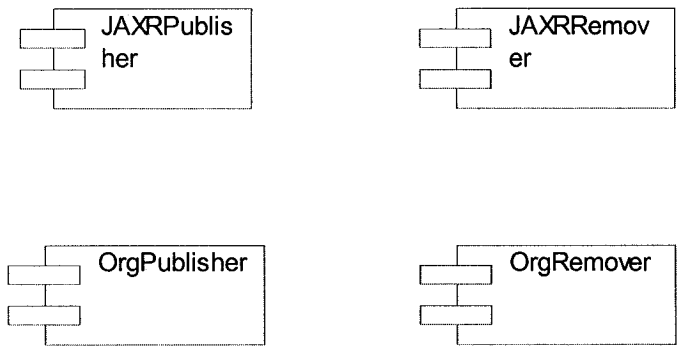


Figure 5-10: Component diagram for the JAXR package

The JAXM package includes objects for the metadata mapping Web Services, which includes two functions: conversion from DC to LOM and conversion from LOM to DC. Each conversion begins with parsing the incoming SOAP Message from the remote calling part, extracting metadata object from the message, and conversion to the counterpart metadata object. Moreover, there are two test objects designed for test those two functions locally.

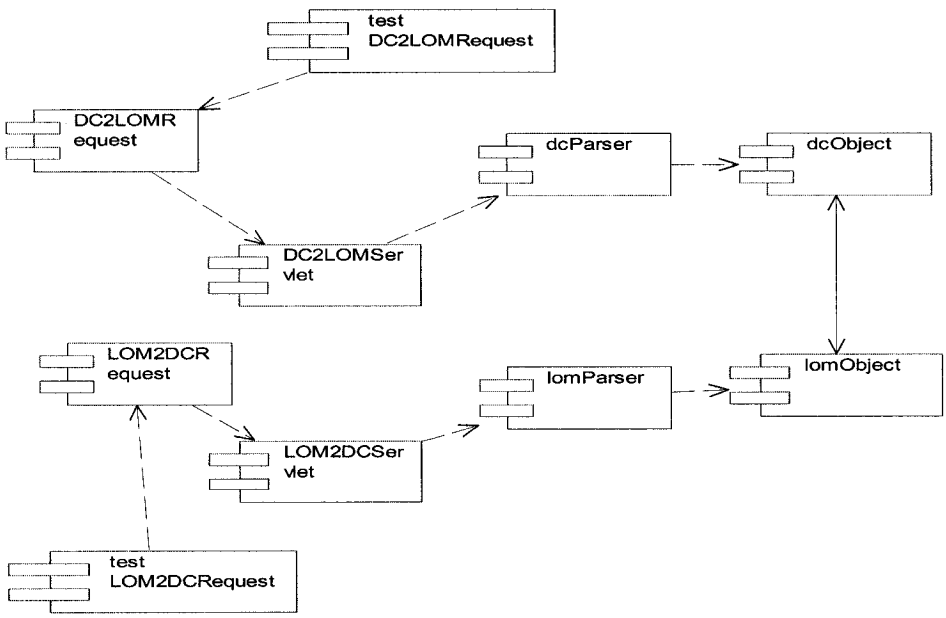


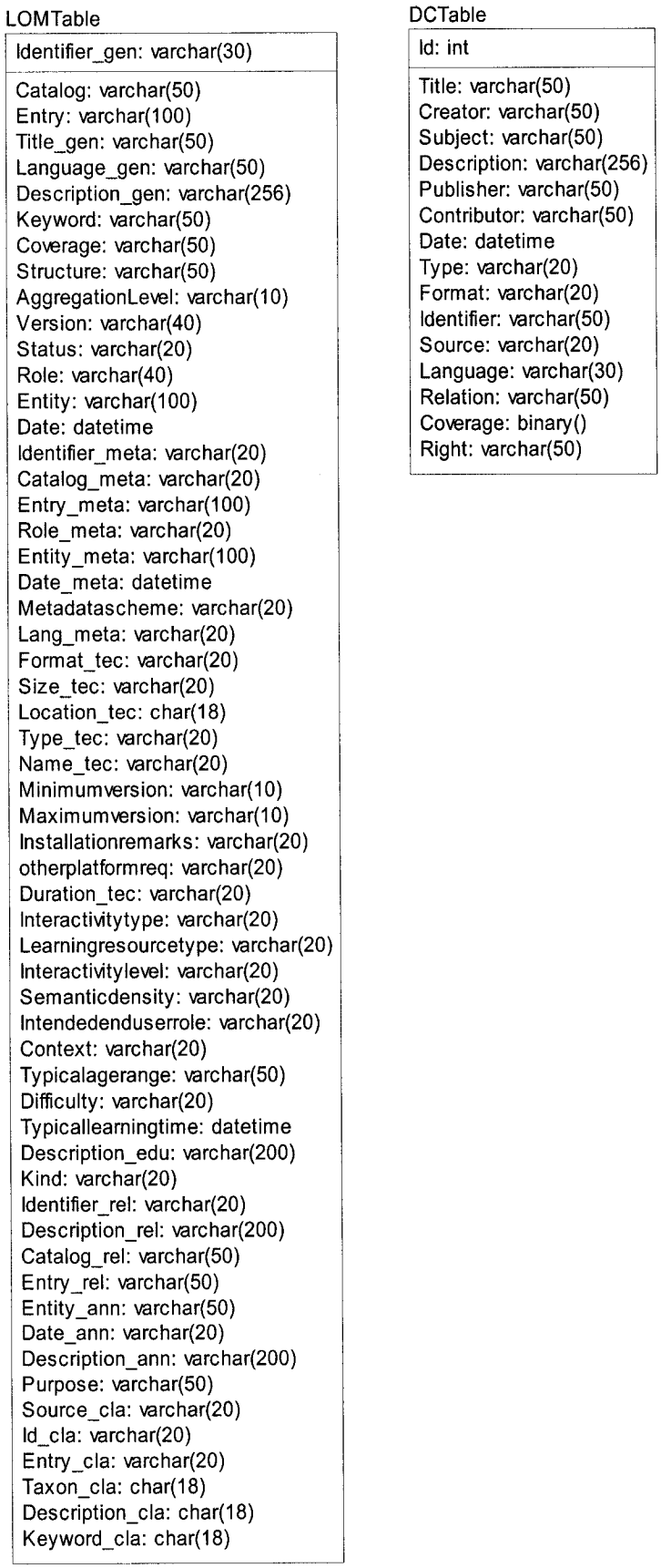
Figure 5-11: Component diagram for the JAXM package

5.3 Data repository

In our prototype, we use an open source relation database management system: MySQL 3.23. Two tables are designed to present Dublin Core and IEEE LOM respectively. They are integrated as part of the Ubilearn database.

The following table shows the DC and LOM elements:

Table 5-1: LOM and DC database design


LOMTable
Identifier_gen: varchar(30)
Catalog: varchar(50)
Entry: varchar(100)
Title_gen: varchar(50)
Language_gen: varchar(50)
Description_gen: varchar(256)
Keyword: varchar(50)
Coverage: varchar(50)
Structure: varchar(50)
AggregationLevel: varchar(10)
Version: varchar(40)
Status: varchar(20)
Role: varchar(40)
Entity: varchar(100)
Date: datetime
Identifier_meta: varchar(20)
Catalog_meta: varchar(20)
Entry_meta: varchar(100)
Role_meta: varchar(20)
Entity_meta: varchar(100)
Date_meta: datetime
Metadatascheme: varchar(20)
Lang_meta: varchar(20)
Format_tec: varchar(20)
Size_tec: varchar(20)
Location_tec: char(18)
Type_tec: varchar(20)
Name_tec: varchar(20)
Minimumversion: varchar(10)
Maximumversion: varchar(10)
Installationremarks: varchar(20)
otherplatformreq: varchar(20)
Duration_tec: varchar(20)
Interactivitytype: varchar(20)
Learningresourcetype: varchar(20)
Interactivitylevel: varchar(20)
Semanticdensity: varchar(20)
Intendedenduserrole: varchar(20)
Context: varchar(20)
Typicalagerange: varchar(50)
Difficulty: varchar(20)
Typicallearningtime: datetime
Description_edu: varchar(200)
Kind: varchar(20)
Identifier_rel: varchar(20)
Description_rel: varchar(200)
Catalog_rel: varchar(50)
Entry_rel: varchar(50)
Entity_ann: varchar(50)
Date_ann: varchar(20)
Description_ann: varchar(200)
Purpose: varchar(50)
Source_cla: varchar(20)
Id_cla: varchar(20)
Entry_cla: varchar(20)
Taxon_cla: char(18)
Description_cla: char(18)
Keyword_cla: char(18)
DCTable
Id: int
Title: varchar(50)
Creator: varchar(50)
Subject: varchar(50)
Description: varchar(256)
Publisher: varchar(50)
Contributor: varchar(50)
Date: datetime
Type: varchar(20)
Format: varchar(20)
Identifier: varchar(50)
Source: varchar(20)
Language: varchar(30)
Relation: varchar(50)
Coverage: binary()
Right: varchar(50)

5.4 Web Services-based metadata mapping

Java Web Services Developer Package (JWSDK) version1_0_01 is used for the Web Services-based metadata mapping implementation.

5.4.1 Java APIs for XML

The Java APIs for XML makes it easy to handle XML processing for applications write in Java. They fall into two broad categories: those that deal directly with processing XML documents and those that deal with procedures.

- Document-oriented
    - Java API for XML Processing (JAXP) -- processes XML documents using various parsers
- Procedure-oriented
    - Java API for XML-based RPC (JAX-RPC) -- sends SOAP method calls to remote parties over the Internet and receives the results
    - Java API for XML Messaging (JAXM) – creates and sends SOAP messages over the Internet
    - Java API for XML Registries (JAXR) -- provides a standard way to access business registries and share information

5.4.2 Parsing and manipulating XML

Since Web Services depend much on data exchange using XML format, parsing XML files becomes a crucial part in most Web Services applications. JAXP provides two mechanisms for parsing XML documents. The Simple API for XML (SAX) parser uses an event model for XML documents: a program registers a listener with the parser and the parser streams through the file, fires notifications when it encounters XML elements which in the listener. On the other hand, The Document Object Model (DOM) presents a

tree model for XML documents and provides an Application Programmers Interface (API) for handling the data in the tree.

Both SAX and DOM are useful for different kinds of applications and both have their advantages and disadvantages. In general, the SAX consumes a small memory and very fast. The DOM occupies a lot of memory and runs slow, but it gives an in-memory representation of the data, and provides an API to access and manipulate them, then give us a stronger control of searching some data under complex situations.

In our Web Services, the DOM is used as our XML parser. The SAX cannot meet the complex identifications for some XML elements, such as identification for some elements in the IEEE LOM: In the SAX, when we meet a beginning of a tag, in the startElement() method, we do not know the location of this tag in the whole document. This is a big trouble because some syntax of a tag in LOM is related to the location of the tag. For example, the same tag <Entry> in the following has different syntax in different location

```
<LOM>
<General>
        <Identifier>
                <Entry>a</Entry>
        </Identifier>
</General>
<Relation>
        <Resource>
                <Entry>b</Entry>
        </Resource>
</Relation>
```

</LOM>

5.4.3 XML Messaging

Both JAX-RPC and JAXM is java APIs built on top of the SOAP. They support different subsets of the SOAP specification. One difference between them is that the JAXM is used for message-oriented applications and the JAX-RPC otherwise offers RPC over the Internet. In general, data-oriented services are good for message-oriented service and procedure-oriented services are better suitable for RPC as far as the durability of the Web Services is a major concern.

Since metadata mapping is essentially conversion between two schemes that are presented in XML files, the metadata mapping Web Services should be better implemented as message exchange between service provider and service consumer.

There are two kinds of XML messages from the message perspective: messages with attachment and messages without attachment. If any content of a message is not in XML format, such as an image or PDF text file, attachments must be used to hold them. For a DC or LOM record file, it can be put either in the SOAPBody or the attachments. In our implementation, it is put in the SOAPBody. The anatomy for a SOAP message sending DC is illustrated in Figure 5-12:

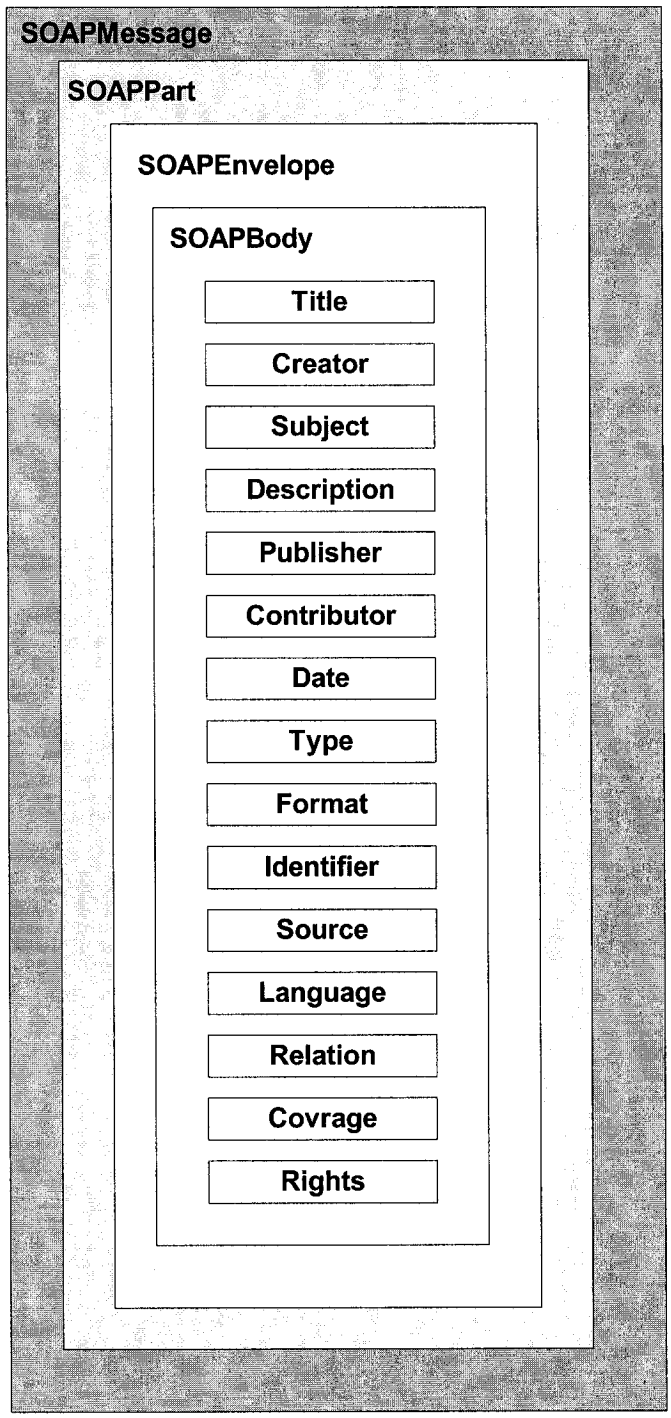


Figure 5-12: Structure of a SOAP message contain DC

All SOAP messages are sent and received over a connection. The connection can either go directly to a particular destination or to a messaging provider. The JAXM provides the following two classes and interfaces respectively.

- javax.xml.soap.SOAPConnection
- javax.xml.messaging.ProviderConnection

SOAPConnection object, which represents a point-to-point connection, is used in our prototype to send request-response message (synchronous). The following steps are needed for a SOAP message sender:

- Get a connection
- Create a message
- Add content to a message
- Send a message
- Retrieve the content from a response message
- Create and retrieve a SOAP fault element

The figure shows a request and its response SOAP messages for DC-LOM mapping. The request message represents a DC metadata scheme and the response represents a corresponding LOM scheme.

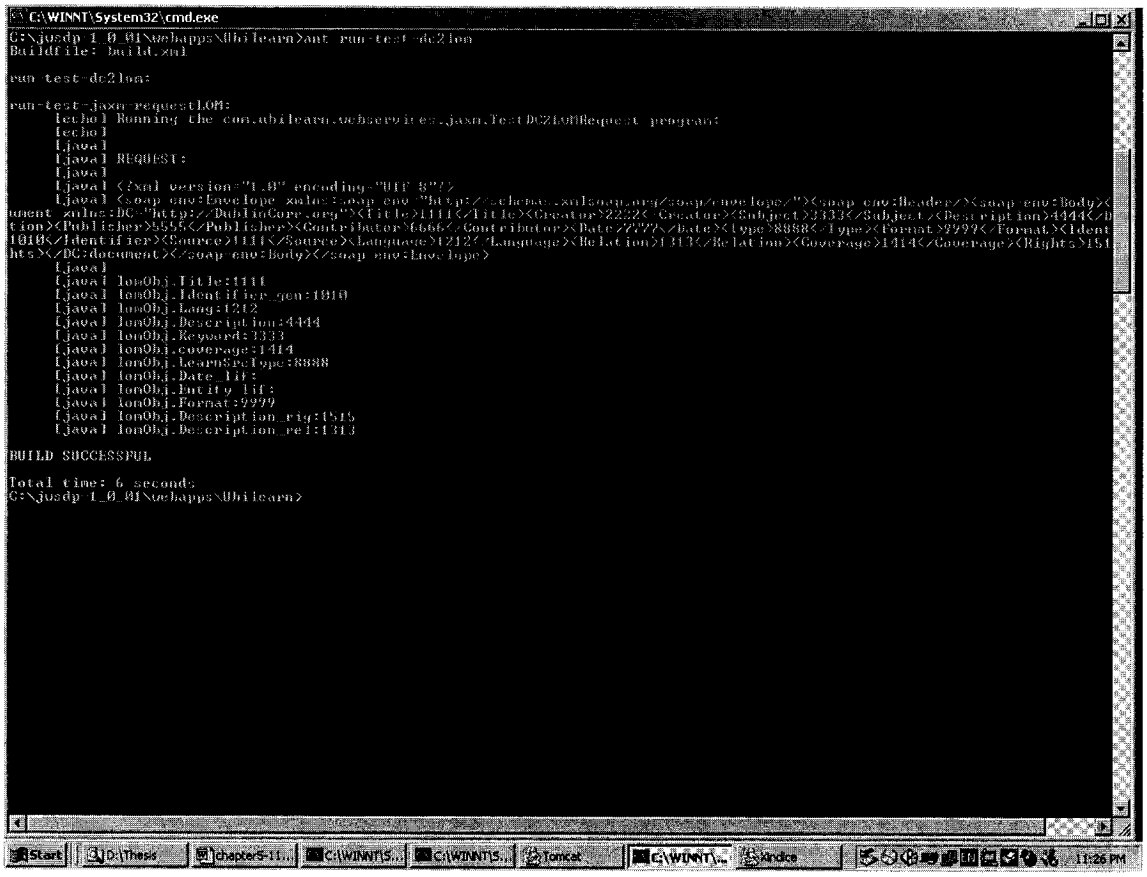


Figure 5-13: DC-LOM request and response SOAP messages

5.4.4 Describing, publishing, and finding Web Services

Currently several XML registries specifications and many XML registry providers exist. The most commonly used XML registry specifications are the ebXML and the UDDI. XML registry providers implement the XML registry specifications to users. Most vendors in the e-commerce and e-learning industry adopt the UDDI specification. The Java WSDP Registry Server is a XML registry provider that implements the UDDI specification (version 2). Since the Java WSDP Registry Server is bundled with JWSDP, it is used in our prototype as the registry provider to test our JAXR applications in our private environment.

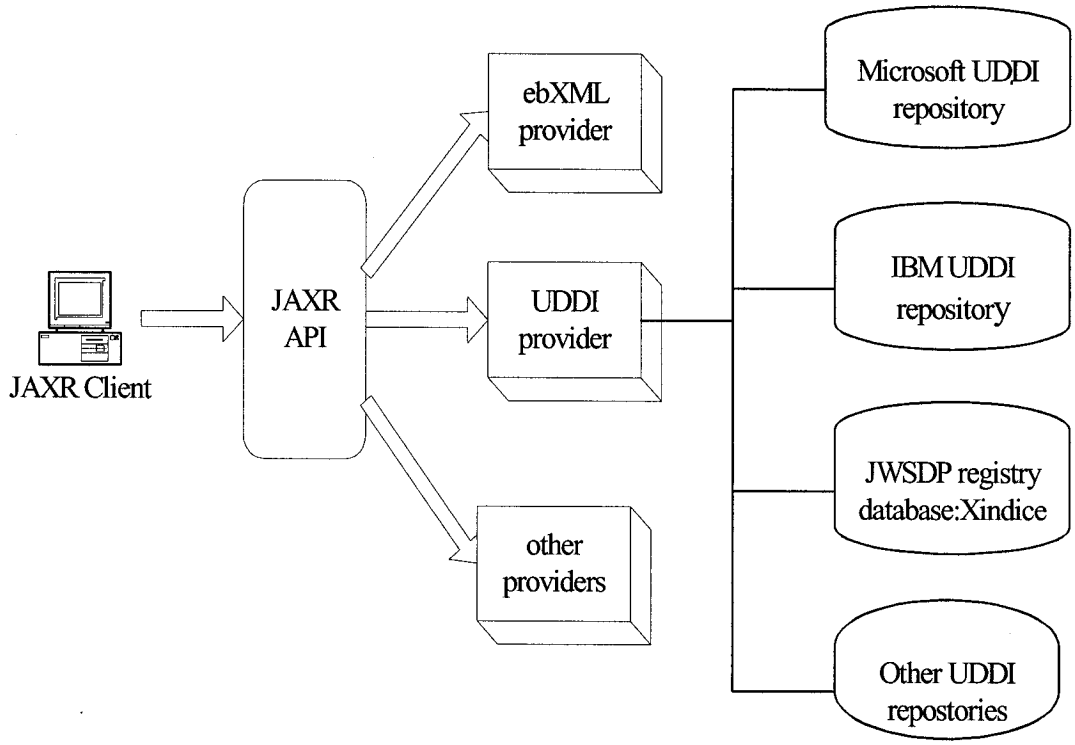


Figure 5-14: JAXR architecture

The JWSDP Registry Server includes a native XML database named "Xindice", which is part of the Apache XML project. This database provides the repository for registry data. And a tool named "Indri" that allows users to create and inspect database data using a graphical user interface.

There are two methods to implement JXAR client which is responsible to access the registry server. One is using JAXP APIs, the other is through a graphic interface called "JAXR Registry Browser" which provided with the Java WSDP to perform queries and updates on registry data. The common tasks they can handle are:

- Publish businesses
- Delete businesses
- Search businesses
- Obtain business details
- List the contents of the database

JAXRPublish.java, JAXRQuery.java, JAXRDelete.java are used to implement the services in the Registry provider. Include functions such as: get access to registry, create a connection factory to registry, create connection and set properties, and so on.

And the LearningRegistry.properties file in the "Ubilearn\webservices\jaxr" directory is used for specify the registry user wish to access. For both the `queryURL` and the `publishURL` assignments. We define that username as "testuser" and password as "testuser". They are used in manipulating the registry. The following figure shows publishing a Web Services to the registry server using registry browser.

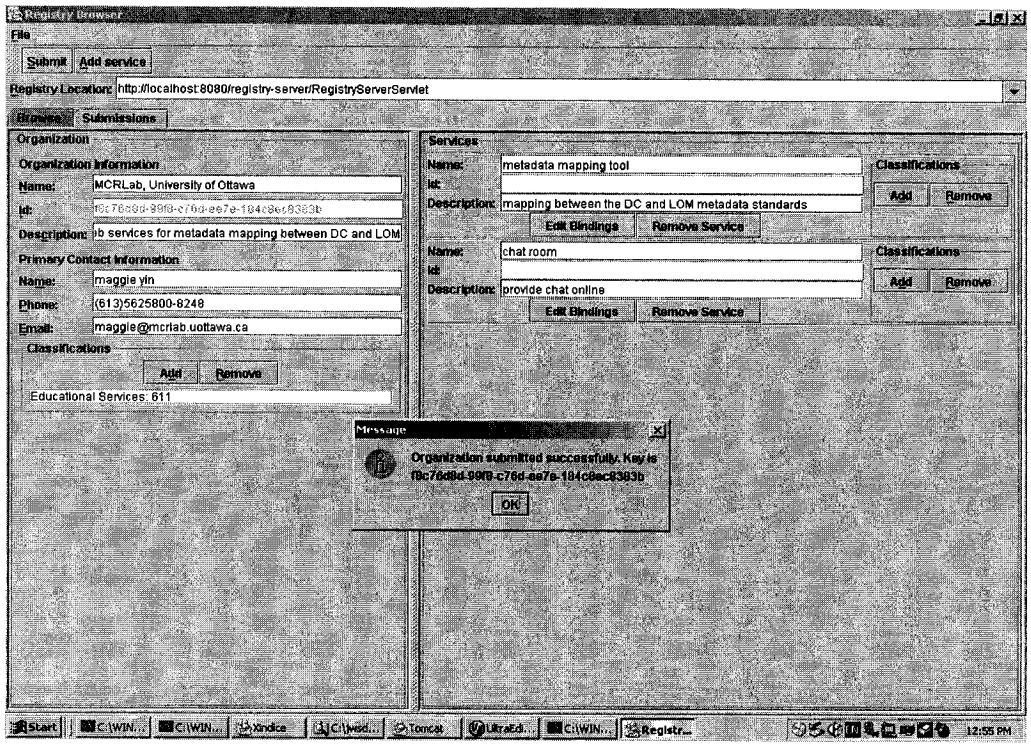


Figure 5-15: Publish metadata mapping Web Services using the Registry Browser

Chapter 6. Conclusion and future work

In this thesis, we have presented the importance of using metadata as a part of next generation e-learning systems. The advantages and disadvantages of using metadata have also been analyzed. Our proposal is described in detail: one method of lowering the cost and subjectivity of using metadata in order to promote the reusability of the learning objects.

The metadata mapping tool is a prototype towards more interoperability, reusability and durability among heterogeneous systems on the global level. Our prototype has these features by following the widely used standards: HTML, XML, SOAP and UDDI, building on Web Services oriented architecture and taking advantage of newest technologies such as: the Java Enterprise platform, Apache struts web application framework, XSL transform, MySQL and so on.

Metadata mapping tool provides transparent conversion between DC and LOM metadata users and other applications that need the conversion as part of their functions. It provides a user-friendly interface for DC and LOM creation. Any one can access it anytime anywhere by visiting our Ubilearn web server. And the metadata Web Services provide platform neutral and language independent way for integrating our components to other applications. It demonstrates the decoupling of different systems in distributed environment. Therefore, it promotes the reusability of the mapping between DC and LOM in different systems and helps to lower the cost of using metadata in educational community.

The future work may further the research in different metadata schema in different domains, such as mapping between LOM and MPEG7. There are a lot of metadata application profile level schemes in widely use for different educational community, such as CanCore and ARIADNE, integrating a mapping functionality in the Ubilearn based on the presented work would benefit both communities.

Reference

[1]. Dublin Core Working Group. 2003. Dublin Core Metadata Initiative. Homepage: http://dublineCore.org

[2]. IEEE Learning Technology Standards Committee. 2002. WG12: Learning Object Metadata. Homepage: http://ltsc.ieee.org/wg12/index.html

[3]. Advanced Distributed Learning Initiative (ADL). October, 2001. Sharable Content Object Reference Model (SCORM) version 1.2. The SCORM Overview-Early stages of computer-based Instruction. Page: 1-23. Http://www.adlnet.org

[4]. Advanced Distributed Learning Initiative (ADL). October, 2001. Sharable Content Object Reference Model (SCORM) version 1.2. The SCORM Overview-Emergence of Intellegent tutoring Systems. Page: 1-24. Http://www.adlnet.org

[5]. Norm Friesen. March 21, 2001. Building a vision for sharing educational objects in Alberta

[6]. Advanced Distributed Learning Initiative (ADL). October 2001. Sharable Content Object Reference Model (SCORM) version 1.2. The SCORM Overview- the ADL Initiative Page 1-13 Http://www.adlnet.org

[7]. Cisco Systems, Inc. November, 2001. Reusable learning object strategy.

[8]. Lori Mortimer. 2001. Learning Objects of Desire: Promise and Practicality. Learning circuits—ASTD's online magazine all about e-learning

[9]. IEEELTSC, 2003. WG12: Learning Object metadata. http://ltsc.ieee.org/wg12/index.html

[10]. Griff Richards and Marek Hatala. June 2002. POOL, POND, and SPLASH: A Peer to Peer Architecture for Learning Object Repositories. Technical University of BC

[11]. Griff Richards, Rory McGreal, Norm Friesen. June 2002. Learning Object repository Technologies for TeleLearning: The Evolution of POOL and CanCore

[12]. Chris Taylor. Jul 29, 2003. An introduction to metadata. University of Queensland Library

[13]. Lrcan Dempsey and Rachel Heery. March, 1997. [DESIRE]Specification for resource description methods Part1: A review of metadata: a survey of current resource description formats.

[14]. Anthony Roberts. 2003. Metaphysics of Metadata. , TeleEducation NB, New Brunswick, Canada

[15]. Sergey Brin and Lawrence Page. 2003. The Anatomy of a Large-scale Hyper textual Web Search Engine. Stanford University

[16]. Dublin Core Metadata Initiative. Sep.10, 2001. Dublin Core metadata element set. National information standards organization

[17]. IEEE WG12. July 5, 2002. IEEE Learning Object Metadata, final draft standard (IEEE 1484.21.1)

[18]. Marek Hatala and Griff Richards. 2002. Global vs. community Metadata Standards: Empowering Users for Knowledge Exchange. Technical University of British Columbia.

[19]. Enric peig, Jaime Delgado, Ismael Perez. Universitat Pompeu Fabra. 2001. Metadata Interoperability and Meta-search on the Web

[20]. David A. Chappell, Tyler Jewell, Java Web Services, O'REILLY. ISBN: 0-596-00269-6

[21]. Gottfried Vossen, Peter Westerkamp. July 2003. E-learning as a Web Service. University of Munster, Germany .IEEE

[22]. Filip Neven and Erik Duval. 2003. Reusable Learning Objects: a survey of LOM-based repositories

[23]. EPFL (Lausanne,CH), K.U.Leuven (Leuven,B) and the ARIADNE Foundation. 15 February 2002. ARIADNE Educational Metadata Recommendation, Version3.2

[24]. ARIADNE. Oct. 2003. ARIADNE systems and tools. http://www.ariadne-eu.org/en/system/index.html

[25]. Gilbert Paquette, Karin Lundgren-Cayrol, Alexis Miara and Louis Guerette. 2003. The Explor@-2 Learning Object Manager. http://ltsc.ieee.org/wg12/index.html

[26]. EXPLOR@ internet-based Virtual Learning Centre. 2003. http://www.licef.teluq.uquebec.ca/gp/docs/prod/explora_eng.doc

[27]. Curriculum Corporation and Education. Australia Limited. 24 January 2003. Learning Object Repository Access And Exchange Web Services Specification

[28]. Sun Microsystems Inc. April 24, 2002. The J2EE tutorial. A beginner's guide to developing enterprise applications on the JavaTM 2 Platform. Enterprise Edition SDK version 1.3.

[29]. Timothy C. Lethbridge & Robert Laganiere, Object-Oriented Software Engineering, Practical software development using UML and Java

Appendix A

List of abbreviations

Advanced Distributed Learning (ADL)

Aviation Industry CBT (Computer-Based Training) Committee (AICC)

Alliance of Remote Instructional Authoring & Distribution Networks for Europe (ARIADNE)

Canadian Core Learning Resource Metadata Protocol (CanCore)

Computer-Based instruction (CBI)

Component Object Model/Distributed COM (COM/DCOM)

Common Object Request Broker Architecture (CORBA)

Cascading Style Sheet (CSS)

Dublin Core (DC)

Dublin Core Metadata Initiative (DCMI)

Department of Defense (DoD)

Document Object Model (DOM)

Electronic Business XML (ebXML)

HyperText Makeup Language (HTML)

HyperText Transfer Protocol (HTTP)

Institute of Electrical and Electronics Engineers (IEEE)

IMS Global Learning Consortium, Inc (IMS)

Intellectual Property Rights (IPR)

Information Society Technologies (IST)

Intelligent Tutoring Systems (ITS)

Java Platform, Enterprise Edition1.3 (J2EE)

Java API for XML Messaging (JAXM)

Java API for XML Processing (JAXP)

Java API for XML-based RPC (JAX-RPC)

Java API for XML Registries (JAXR)

Java Server Pages (JSP)

Java Web Services Developer Package (JWSDK)

Knowledge Pool System (KPS)

Learning Management Systems (LMS)

Learning Object (LO)

Learning Object Metadata (LOM)

Learning object repositories (LOR)

Learning Object Repository Access and Exchange (LORAX)

Learning Technology Standards Committee (LTSC)

Multimedia Communication Research Lab (MCRLab)

Model-View-Controller (MVC)

North American Industry Classification System (NAICS)

Office of Science and Technology Police (OSTP)

Online Objects for Learning (POOL)

Relational Database Management System (RDBMS)

Simple API for XML (SAX)

Sharable Content Object Reference Model (SCORM)

Service Oriented Architecture (SOA)

Simple Object Access protocol (SOAP)

Universal Description, Discovery and Integration (UDDI)

Unified Modeling Language (UML)

Uniform Resource Identifier(URI)

World Wide Web Consortium (W3C)

Web Services Description Language (WSDL)

eXtensible Makeup Language (XML)

Extensible Style sheet Language (XSL)