



uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES
ET POSTDOCTORALES



FACULTY OF GRADUATE AND
POSTDOCTORAL STUDIES

Kai Wang

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (Systems Science)

GRADE / DEGREE

Systems Science

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Framework Design of Web-services based E-Learning Portal System

TITRE DE LA THÈSE / TITLE OF THESIS

A. El Saddik

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Amiya Nayak

Shervin Shirmohammadi

Gary W. Slater

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Framework Design of Web-services based E-Learning Portal System

By

Kai Wang

A thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements for the degree of

Master of Science

In

System Science

School of Information Technology and Engineering

University of Ottawa

© Kai Wang, Ottawa, Canada, 2005



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-494-11447-9

Our file Notre référence

ISBN: 0-494-11447-9

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

Abstract

This research is to propose a Web services based framework for e-learning portal systems designed to present a flexible integration model and to make the e-learning environment intelligent and adaptive to each individual learner. It is very important that all the learning components and applications are clearly defined, effectively discovered and loosely coupled. Web services and Portals provide an essential deployment environment to realize collaborative e-learning environment by facilitating efficient communication of components and applications. Given the proposed framework, instructor and learner will be able to use their learning objects or services based on portal (collaboration environment) universally anywhere, any time through common communication protocols (HTTP, SOAP). The key values of interoperability and accessibility in the proposed architecture enhance the future collaborative e-learning portal systems to communicate more efficiently and to share data more easily. A prototype of the proposed system is designed and implemented in a J2EE (Java 2 Enterprise Edition) combined with multi-channel, using WSRP (Web Service for Remote Portlet) for building essential Web services and for parsing SOAP (Simple Object Access Protocol) messages on lightweight platforms. The implementation is a successful demonstration that learning services can be easily accessed by portlet through standard Web services interface. Another implementation about role base access control is presented. An internationalization solution (I18N, L10N) based web-service e-learning portal system is successfully accomplished in this implementation.

Acknowledgements

I would like to express my sincere gratitude to my thesis supervisor Dr. Abdulmotaleb El Saddik, for giving me an opportunity to pursue my graduate study, bringing me the interesting problem of e-learning and providing valuable support whenever I need. I greatly appreciate his trust, guidance, enthusiasm, and encouragement.

I would also like to thank my colleagues in the Multimedia Communication Research Laboratory, University of Ottawa, for providing me a pleasant working environment.

Finally, thank to all my family members, for giving me endless love, care and support.

Table of Contents

Abstract.....	ii
Acknowledgements	iii
Table of Contents	iv
List of Figures	viii
List of Abbreviations	xii
List of Abbreviations	xii
Chapter 1	1
Introduction	1
1.1 Background.....	1
1.2 Research Motivation.....	2
1.3 Research Contributions	3
1.4 Organization of the Thesis.....	4
Chapter 2	6
Overview of Portal and Web Service	6
2.1Portal Technology	6
2.1.1 Definitions	6
2.1.2 Portal functional architecture.....	7
2.1.3 The characteristics and benefits of portal solution	9
2.2 Web Service Technology	10

2.2.1 eXtensible Markup Language (XML).....	10
2.2.2 Exchanging Message with SOAP	11
2.2.3 Describing a Web Service with WSDL	12
2.2.4 Publishing and Finding Web Services Using UDDI.....	13
2.2.5 Web Services Architecture.....	14
2.2.6 Advantages of Web Service	15
Chapter 3	17
Infrastructure of E-Learning Portal System	17
3.1 Overview of E-Learning System.....	17
3.1.1 E-learning System	17
3.1.2 E-Learning Component	18
3.2 Related work	21
3.2.1 Learning Object Metadata	22
3.2.2 SCORM SYSTEM.....	24
Chapter 4	28
Analyzing and Designing the E-learning System.....	28
4.1 System Requirement.....	28
4.2 System Prototype Description.....	29
4.3 Proposed Framework.....	31
4.3.1 Logic Framework.....	31
4.3.2 System Framework	33
4.4 System Analysis	35
4.4.1 Use Case Model	35

4.4.2 Interaction Sequence Model	38
4.5 System Class Design	41
4.5.1 Package View.....	41
4.5.2 Class View	42
4.5.3 Component View	44
Chapter 5	46
Implementation.....	46
5.1 The Framework Implementation.....	46
5.2 The Role-Based Portal Access Control	48
5.3 Portlets Access to Web Services	50
5.4 System Internationalization	53
5.4.1 Internationalization and Localization.....	53
5.4.2 Making a Program I18n-Compliant	53
5.4.3 Making Ublearn internationalized with Weblogic	54
5.5 Web Service for Remote Portlet.....	56
5.5.1 How WSRP works	56
5.5.2 Develop Web Service for Remote Portlet with Weblogic	58
5.6 System Service Implementation.....	60
5.6.1 The services of Instructor	60
5.6.2 The Services of Student.....	62
5.6.3 The Relative Public Service	64
Chapter 6	69
Testing and Measurement	69

6.1 Testing Goal Analysis.....	69
6.2 Test Lab Environment	70
6.3 Test Methodology & Result.....	70
Chapter 7	75
Conclusions and Future Research	75
7.1 Conclusion	75
7.2 Future Research.....	76
References.....	77

List of Figures

Figure 2.1 A portal with different portlets represented through the internal windows ...	7
Figure 2.2 Portal functional architecture [14]	8
Figure2.3: Exchanging Message with SOAP [16]	11
Figure: 2.4 SOAP Message.....	12
Figure 2.5: web service Architecture [16]	14
Figure3.1 E-learning Components [28].....	19
Figure 3.2 Structure of LOM [30].....	23
Figure 3.3 What is SCORM [41]	25
Figure3.4: The SCORM Content Aggregation Model [38].....	26
Figure 3.5: The SCORM Run-Time Environment [40]	27
Figure 4.1 Virtual Learning Community	29
Figure 4.2 Portal Framework of Web Service based e-learning portal system	32
Figure 4.3: The web service Framework of Web service based e-learning portal system	34
Figure 4.6: Use Case Diagram for the administrator as actor	37
Figure 4.7: Interaction Sequence Model for Scheduling a Course	38
Figure 4.8: Interaction Sequence Model for Teaching a Course	39
Figure 4.9: Sequence Diagram for the Realization of Use Case for Searching Web Services	40
Figure 4.10 Sequence Diagram for the Realization of Use Case: Publish/Remove Web Services	40
Figure 4.11 Sequence Diagram for the Realization of Use Case: Get Shared Course	

Information	41
Figure 4.12: System Package View.....	42
Figure 4.13: System Class Diagram.....	43
Figure 4.14: System Component Diagram	45
Figure 5.1: Implementation Architecture of the System	47
Figure 5.2 Relationships between users, roles, permissions and services [47]	49
Figure 5.3: Role-based portal access control [50]	50
Figure 5.4: Portlet and service	51
Figure 5.5: Web Service Testing Console	52
Figure 5.6 the process of Internationalization [49].....	54
Figure5.7: I18n resource file	55
Figure5.8: JSP label sample.....	55
Figure 5.9: Web Service for Remote Portlet [52][54]	56
Figure 5.10: Interaction between consumer and producer [54].....	57
Figure 5.11: Portlet Type	58
Figure 5.12: Portlet Modes (A).....	59
Figure 5.13: Portlet Modes (B).....	59
Figure 5.14: Course information.....	60
Figure 5.15: Course information.....	61
Figure 5.16: Assignment Management (1).....	62
Figure 5.17 Assignment Management(2)	64
Figure 5.18 Calendar Summary View	65
Figure 5.19 Day view	65

Figure 5.20 Adding an Appointment.....	66
Figure 5.21 Contact Summary View.....	66
Figure 5.22 Finding Contact.....	67
Figure 5.23 Adding, Deleting Contacts.....	68
Figure 6.1: Network Configuration for Testing.....	69
Figure 6.2: Solution 1: each service receives and responds to the main portlet.....	72
Figure 6.3 Solution 2 services receive and send response following the chain	73

List of Tables

Table4.1: User role and action	36
Table5.1: I18n &L10n Jsp Tags[51].....	54
Table 6.1: Test Environment.....	70
Table 6.2 Test data from the intranet.....	73

List of Abbreviations

ADL	Advanced Distributed Learning
AICC	Aviation Industry CBT (Computer-Based Training) Committee
API	Application Program Interface
ARIADNE	Alliance of Remote Instructional Authoring & Distribution Networks for Europe
CBT	Computer-Based Training
CORBA	Common Object Request Broker Architecture
DCOM	Distributed Component Object Model
DoD	Department of Defences
DOS	Disk Operating System
DNS	Domain Name Server
DTD	Document Type Definition
EJB	Enterprise Java Bean
GUI	Graphic User Interface
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
I18N	Internationalization
IP	Internet Protocol
J2EE	Java 2 Platform, Enterprise Edition
JAX	Java APIs for XML
JAXM	Java API for XML Messaging
JDBC	Java Data Base Connection
JSP	Java Server Pages
JMS	Java Message Service
L10N	Localization
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LMA	Learning Management Applications
LMS	Learning Management Systems
LOM	Learning Object Metadata

LTSC	Learning Technology Standards Committee
MIME	Multi-purpose Internet Mail Extension Protocol
QoS	Quality of Service
RBAC	Role-Based Access Control
RPC	Remote Procedure Call
RTE	Run-Time Environment
SAX	Simple API for XML Parsing
SCO	Sharable Content Objects
SCORM	Sharable Content Object Reference Model
SMTP	Simple Message Transfer Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SoD	Secretary of Defence
SOL	Server Object Layer
TBT	Technology-Based Training
TCP	Transmission Control Protocol
UDDI	Universal Description Discovery and Integration
UML	Unified Modeling Language
URI	Uniform Resource Identifiers
VLC	Virtual Learning Community
W3C	World Wide Web Consortium
WSRP	Web Service for Remote Portlet
WSDL	Web Services Description Language
XML	eXtended Markup Language

Chapter 1

Introduction

1.1 Background

E-learning is a type of education that offers some interesting benefits over traditional learning in terms of independence. For example, learners work anywhere they prefer, such as, at home or in the office, to communicate with instructor and other learners via e-mail, electronic forums, chatting, videoconferencing and other forms of computer-based communication [1]. Although some unproved concepts of e-learning such as TBT (technology-based training), CBT (computer-based training) have appeared and other half-forgotten acronyms have been created over the years [2], it is obvious that e-learning is rapidly emerging as an effective and efficient way to deliver the certain classes of training in the specific environments [3]. E-enabled distance learning is becoming more than ever popular with learning providers that need to frequently re-train their learners, because it is less expensive than bringing all the learners together in a traditional classroom setting. Instructors can provide the learning resource anytime, anywhere and learners can work on their learning depending on when is available for themselves [1].

Most institutions have embraced the challenge of e-learning – the use of Internet technologies in teaching and learning. But too few are clear about the changing nature of the challenge. The challenge initially was framed as advancing the faculty willingness to adapt instruction to take advantage of the Internet revolution in human communication and resource sharing. And this accounts for the plethora of instructor support programs designed either 1) to enhance traditional classroom-based courses with web access to syllabi, materials, and discussions or 2) to put courses online in a distance education modality that eliminates or reduces "onground" classroom time. Today the challenge is to develop a prioritization and investment strategy that recognizes the evolving mission-critical role of e-learning. Students today expect much more than online access to course materials or even to courses. They expect online access to both academic and administrative services. So today, the challenge is to integrate academic and administrative

services on the web for presenting a single and personalized point of contact for students, instructors, and other stakeholders.

As the computer technology is developing rapidly, the internet has had a revolutionary effect on how learning provider conducts their education, opening up the new possibilities for communicating with instructors, learners and partners. E-learning has raised the stakes for organizations, greatly increasing the learning process speed at which they are expected to compete. One of the important aspects to bringing efficiencies to learning-to-learning interaction is the need for learning providers to integrate their e-learning applications with those of their partners in the e-learning marketplace.

1.2 Research Motivation

E-learning technology is seen by some as a viable solution for combating the problem of traditional education, such as growing enrollments and fewer instructors, which lead to over-crowded schools [4].

We are facing a great change in the education system, which is originated from not only the rapid development of the internet technologies, but also the new requirements of today's education [5]. Instructors and learners are sharing learning resources. For example, at the graduate level, joint programs emerge from the collaboration between the University of Ottawa and Carleton University. Graduate students can take courses from any of the two universities. This raises some issues about how to share learning resources from one system with other e-learning systems and how to access those systems seamlessly. Instructors and learners expect a secure, single and personalized point to access learning resources from the various learning providers.

Further, the key characteristics of an ideal e-learning system include:

- Personalization - the ability to serve dynamic response to the user based on personal profiles
- Collaboration - tools that allow e-mail, team rooms, shared places, and so forth to be exchanged
- Source sharing - facilitating not only the learning resources sharing but the management content as well

This thesis presents a new e-learning framework featured and modeled to support the flexibility and collaboration and to make the e-learning environment intelligent and adaptive to each individual user, whether s/he is an instructor, a learner, or even an

administrator and try to research this possibility with the emerging technologies.

A portal provides a single point of access to applications, application content, processes, and people in the network. To provide this unified access, the portal implements rich and valuable functions. The portal allows users to establish customized portlets for all kinds of user including instructor, learner and administrator [6]. The framework architecture implemented in the portal provides a unified access point to internal and external Web applications, as well as portals access to the other legacy applications. In this way, users sign on to the portal and receive personalized Web pages.

The personalized single point of access to all necessary resources reduces information overload, accelerates productivity, and increases Web site usage. In addition, portals do much more and provide additional valuable functions such as security, search, collaboration, document management, document viewing, and workflow [7].

Web service, as a standard, proposes a service-oriented paradigm for describing, publishing, discovering and binding application interfaces. The set of standards include a standard specification for public registries known as Universal Description Discovery and Integration (UDDI), a description language namely Web Services Description Language (WSDL), a distributed object communication protocol called Simple Object Access Protocol (SOAP) and a dynamic, self-defining information specification Language with semantic support known as eXtended Markup Language (XML) [9].

1.3 Research Contributions

In this thesis, a new framework is proposed, which is not only Web service oriented but also portal based. It provides a flexible integration model to achieve the emerging needs of the novel collaborative e-learning system. The framework facilitates collaborative e-learning systems by providing a comprehensive platform in which all the collaborative learning components are published, described, located and invoked in a standardized way. The major contributions related to the proposed framework are briefly summarized as follows:

- The thesis describes the details of how a web service oriented e-learning portal system based on this framework is analyzed and designed with Unified Modeling

Language (UML).

- Based on the analysis and design, a Web services based on e-learning portal system prototype called UbiLearn has been developed at the Multimedia Communication Research Laboratory at the University of Ottawa, Canada.
- The essential implementation technologies for portlets to access Web services are depicted.
- The essential technologies of Web Service for Remote Portlets on UbiLearn are also implemented.
- The essential technologies of internationalization on UbiLearn are also implemented.

Publication:

- Kai Wang, Jianming Ke and Abdulmotaleb El Saddik. "Architecture for Personalized Collaborative E-learning Environment". In Proceedings of World Conference on Educational multimedia, Hypermedia & Telecommunications (ED-MEDIA 2005) Montreal, Quebec, Canada, June 27- July 2, 2005.
- UbiLearn system as a demonstration of the new technologies was exhibited in CeBIT (2005) focussing on the commercialization of the research and on the innovative technologies.

1.4 Organization of the Thesis

We have stated our research motivation, problem, objective and contributions. The remaining part of the thesis is organized as follows:

Chapter 2 presents the general architecture and standard-based protocols of the new emergence of Web services and the overview of portal and portlet technology.

Chapter 3 describes the system components of distributed e-learning. The two related works of distributed e-learning: LOM and SCORM are introduced.

Chapter 4 presents the details of the Web services oriented framework for distributed e-learning portal system. The system architecture, the analysis and design models for the Web services-based e-learning portal systems are discussed.

Chapter 5 describes the implementation details of the e-learning system enhanced by Web services: UbiLearn. The internationalization technology of E-learning system is given, and the essential technologies for portlet accessing web service are presented. This chapter is ended with new technology “web service for remote portlet”.

Chapter 6 presents the methodology for the testing measurements and analyzes the test results of a specific Web service to show which solution is better to handle Web service for Remote Portlet under some loading conditions.

Chapter 7 gives the conclusions of the thesis and suggests the directions for the future research.

Chapter 2

Overview of Portal and Web Service

2.1 Portal Technology

2.1.1 Definitions

"Portals provide a secure, single point of interaction with diverse information, business processes, and people, personalized to a user's needs and responsibilities." There is a good analogy between what portals add to Web applications and what window managers (like Microsoft Windows) add to operating systems (like DOS). Both provide a consistent and uniform way to interact with applications [10]. A portal-based Web application typically provides a variety of services such as content search, news providing, personalization of content displayed in portlets, collaboration, e-commerce and links to other sites. A good portal application is similar to a "Mobile Workshop" which provides a suite of internal linked Web-based portal components (called portlets), including e-mail, conferencing, instant messaging, message boards and more to help users communicate wherever they are around the globe [11]. Portals are becoming more and more important to companies, who have an ever-increasing requirement to provide employees, partners, and customers with an integrated view of applications, information, and business processes. Portal meets these requirements, allowing organizations to build portals that combine functionality and resources into a single interface while enforcing business policies, processes, and security requirements, and providing personalized views of information to the end users [12].

Portlets are visible active components that users can see within their portal pages and enable the presentation behaviour of a subset of an application to be managed as a single

unit. A portlet exists as a set of associated files, mostly XML and JSPs [12]. Given the simplest terms based on J2EE, a portlet is a Java servlet that operates inside a portal based on J2EE [10].

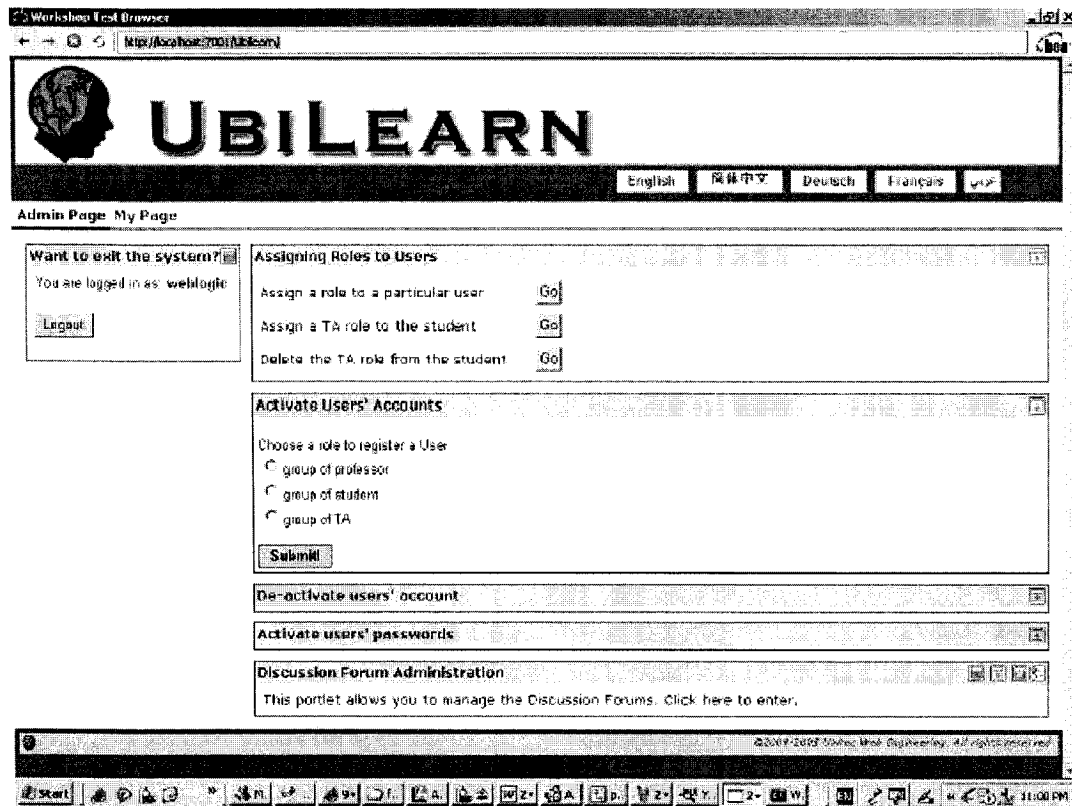


Figure 2.1 A portal with different portlets represented through the internal windows

2.1.2 Portal functional architecture

As discussed in [14] a Portal functional architecture consists of three layers and will be described in the following.

Unified Portal Framework

The Portal Framework lowers the total cost of ownership by providing a flexible and industrial strength foundation for enterprise portals [14].

- Enterprise Portal Architecture - Portal works whether our operations require a department portal, a distributed network of portals, or one massively large portal.
- Enterprise Integration - Portal extends the reach of portals, reduces costs with standards-based integration, and lets user reuse existing Web content and application functionality in new ways.

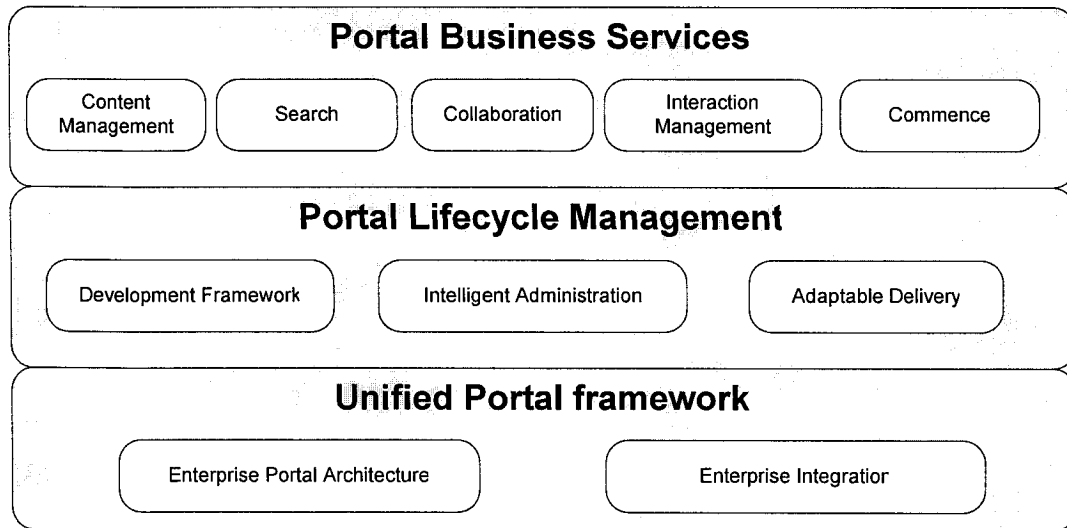


Figure 2.2 Portal functional architecture [14]

Portal Lifecycle Management

Portal Lifecycle Management accelerates the delivery of business solutions by reducing the time to build, deploy, and manage portals. Tools for both developers and business level portal managers enable a custom-fit of Portal to enterprise and best practices [14].

- **Development Framework** - Using Workshop unified development environment with its portal extensions, developers can quickly combine Web services, Web applications, and business processes to create new portal resources. By combining a development environment with an application runtime, Portal lets developers focus on portal applications instead of infrastructure.
- **Intelligent Administration** - Browser-based tools let business-level portal managers assemble, configure, and manage enterprise portals. Portal administration tools provide granular control and administration delegation for custom role-based management.
- **Adaptable Delivery** - Accommodate custom portal interface requirements for customers, partners, and employees with an exible portal delivery framework. Deliver custom portals to cross-functional teams or enable portals for mobile users with multi-channel delivery.

Portal Business Services

Portal Business Services allows for incremental evolving of custom portal functionalities. Pre-integrated modular business services also minimize project risk and provide the flexibility to meet unique business requirements [14].

- (Content Management) - Add and centrally manage content and provide dynamic access to content from multiple repositories.
- Search - Help visitors find the information they need by delivering accurate and meaningful search results.
- Collaboration - Use a rich collaborative environment to build communities and increase productivity and portal adoption.
- Interaction Management - Improve the visitor experience, increase adoption and loyalty, and achieve business goals with visitor interactions.
- Commerce - Increase portal value and return on investment by adding commerce functionality.

2.1.3 The characteristics and benefits of portal solution

There are numerous characteristics of portal solutions.

- Presentation - a Web user interface plus pervasive device support.
- Personalization - the ability to serve dynamic response to the user based on personal profiles.
- Collaboration - tools that allow e-mail, team rooms, shared places, and so forth to be exchanged.
- Portlets - a framework for easily attaching software modules (portlets) and services
- Applications and workflow - integration of legacy and new applications.
- Search and categorize - categorizing repositories of content and searching them for the relevant information.
- Publish and subscribe - the ability to author new content and publish it to subscribers.
- Administration and security – the basic web site services such as page designers, performance monitors, cluster services, and metadata management.
- Integration - metadata sharing, XML, connectors, standards, EAI.

It is worth that a portal solution has some advantages and benefits when compared to a standard web application. Among these advantages are:

- A single point of access to all resources associated with the portal domain.
- Personalized interaction with the portal services.
- Federated access to hundreds of data types and repositories aggregated and categorized.
- Collaboration technologies that bring people together.
- Integration with applications, documents, and workflow system.

2.2 Web Service Technology

2.2.1 eXtensible Markup Language (XML)

XML (eXtensible Markup Language) is a markup language similar to HTML and designed to describe data. XML tags are not predefined. Programmer can define their own tags depending on what they require. XML uses a Document Type Definition (DTD) or an XML Schema to describe the data. XML with a DTD or XML Schema is designed to be self-descriptive [15]. XML, in combination with other essential protocols such as SOAP (Simple Object Access Protocol), provides a comprehensive solution to the problems of data portability and integration standards. XML provides the channel protocols for sending message across several protocols such as HTTP [16].

The component of XML:

- **XML document:** A file that obeys the rules of XML. It contains data and can be thought of as a data store or a mini-database.
- **XML parser:** A computer program that takes XML as its input and produces a program-readable representation of its contents.
- **Document Type Definition (DTD):** A description of the tags that are allowed to exist in a document and their relationships to each others.
- **XML Schema:** A description of the tags that are allowed to exist in a document and their relationships to each others.
- **Namespaces:** A unique name can be used to avoid conflicts between tag names.

2.2.2 Exchanging Message with SOAP

SOAP (Simple Object Access Protocol) is a simple lightweight XML based protocol to let applications exchange information over HTTP in a decentralized and distributed environment [17]. Or more simply: SOAP is a protocol for accessing a Web Service.

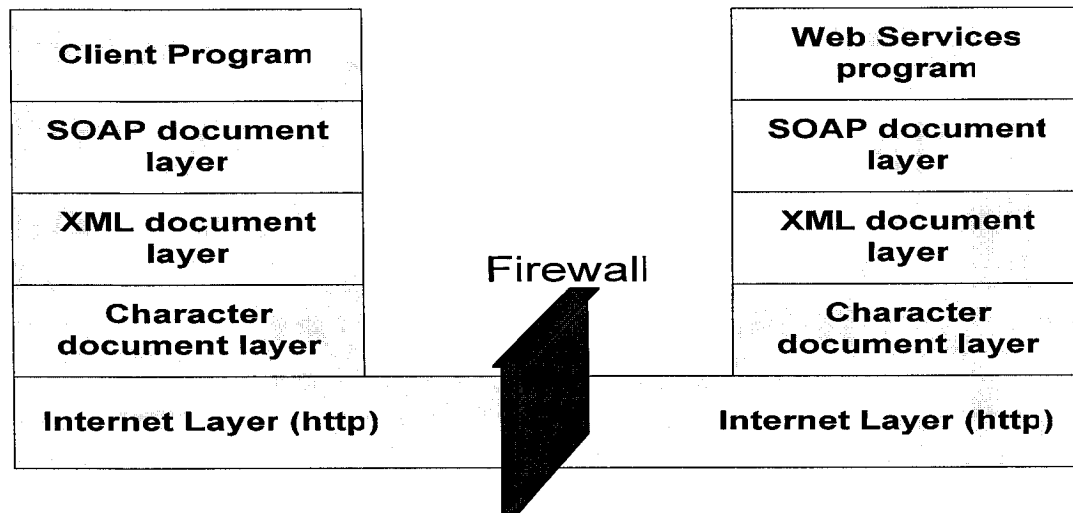


Figure2.3: Exchanging Message with SOAP [16]

A soap message is an XML document. Using SOAP can be thought of as a set of layers as shown in Figure 2.3

From the view point of the Internet layer computer A is sending an XML document to computer B via HTTP. Computer B's firewall policy states that XML document are allowed to pass through via HTTP. The Web server on Computer B receives the file and hands it to the SOAP processor, which uses an XML parser to read the document.

From the XML parser's point of view, the document is simply a well formed and valid XML document. The SOAP processing engine evaluates the file against the rules of the SOAP grammar and an XML schema. It examines the SOAP vocabulary to determine if it is valid. If it is valid, the SOAP processor make a call to the Web services described in the SOAP document might contain.

When the Web service finishes its processing, it creates a response that is formatted in an application-specific way. It wraps this response in a SOAP message format, which is a

valid XML document also. It stores this document in a file and handles it to the Web server for delivery back to the client computer, computer A.

On computer A, the HTTP client program receives the response file. It calls the SOAP processor to parse and to validate it. If it is valid document, the SOAP processor passes the response back to the Web services client program that sent the original request.

A couple of details can change in the preceding scenario, but the basic thrust remains the same. Other transport protocols such as JMS (Java Message Service) or SMTP (Simple Message Transfer Protocol), can be used to actually move the message from computer A to computer B. In addition, the SOAP message might not be a method call and response; it could simply be a single call or even just a document being moved.

A SOAP message contains two SOAP-specific sub elements within the overall Envelope element, namely a Header element and a Body element [18]. Here is an example (Figure 2.6):

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope xmlns:soap-
env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Header/>
  <soap-env:Body>
    <CourseList>
      <course>
        <courseID>CSI5111</courseID>
        <name>Web Services</name>
      </course>
    </CourseList>
  </soap-env:Body>
</soap-env:Envelope>
```

Figure: 2.4 SOAP Message

2.2.3 Describing a Web Service with WSDL

WSDL (Web Services Description Language) is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented

or procedure-oriented information [19]. WSDL has three parts: Definitions, Operations and Service bindings [20].

Definitions are normally expressed in XML and include both data type definitions and message definitions that use the data type definitions. These definitions are usually based upon some agreed upon XML language.

Operations describe events for the messages supported by a Web service. There are four types of operations:

- One-way: Messages sent without a reply required
- Request/response: The sender sends a message and the receiver sends a reply.
- Solicit response: A request for a response. (The specific definition for this action is pending.)
- Notification: Messages sent to multiple receivers. (The specific definition for this action is pending.)

Service bindings connect port types to a port. A port is defined by associating a network address with a port type. A collection of ports defines a service.

The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate, however, the only bindings described in this document describe how to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME [20].

2.2.4 Publishing and Finding Web Services Using UDDI

The UDDI (Universal Description Discovery Integration) is an industry initiative that is working to enable businesses to quickly, easily, and dynamically find and transact with one another. UDDI will benefit businesses of all sizes by creating a global, platform independent, open architecture for describing businesses and services, discovering those businesses and services, and integrating businesses using the Internet. Any kind of service can be registered in the UDDI Business Registry, such as manual services and electronic

services, but the primary intent behind UDDI is to provide a global registry for Web Services[21][74].

UDDI can give a business visibility on a global scale by providing means for an organization to advertise its business and services in a global registry. The UDDI Business Registry provides a place for a company to programmatically describe their services and business processes and their preferred methods for conducting business. UDDI makes it possible for organizations to quickly discover the right business out of the millions that are currently online. Integrate with these other businesses: Once an organization finds a potential business partner, there's no standard mechanism to figure out how to conduct electronic business with this partner. UDDI can simplify the effort of integrating disparate business processes, so that partners can quickly and easily begin trading [22].

2.2.5 Web Services Architecture

Service oriented architecture (SOA) forms the basis of the web-services model. SOA has evolved from the object-oriented model, in which all entities are treated as objects. Similarly, in SOA all entities are treated as services. Services are created by using objects, which are merely reusable software components. SOA provides a programming model that enables services residing on any network to be published, located and invoked by other services [16].

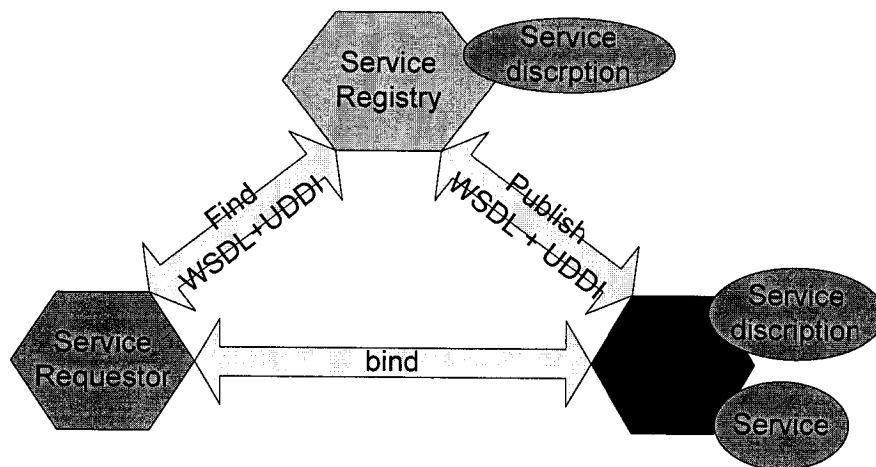


Figure 2.5: web service Architecture [16]

The service provider defines an abstract service description using the Web Services Description Language (WSDL). A concrete Service is then created from the abstract service description yielding a concrete service description in WSDL. The concrete service description can then be published to a registry such as Universal Description, Discovery and Integration (UDDI). A service requestor can use a registry to locate a service description and from that service description select and use a concrete implementation of the service [23].

The abstract service description is defined in a WSDL document as a Port Type. A concrete Service instance is defined by the combination of a Port Type, transport and encoding binding and an address as a WSDL port. Sets of ports are aggregated into a WSDL service [24].

2.2.6 Advantages of Web Service

The use of Web services for building distributed applications has a number of advantages over preceding distributed computing technologies. One of the most important is the promise of true interoperability. Web services technology is the industry's first uniting interoperability framework. Previous approaches to distributed computing have always suffered from vendor partitioning, generally along the lines of Microsoft versus the rest of the industry, e.g., DCOM versus CORBA. Web services are broadly supported by all of the relevant players in the computing software field including Microsoft, IBM, Sun, Oracle, HP, etc. This agreement means that an application implemented in C# running on the .Net runtime environment on a Windows machine will be able to communicate natively with services implemented in, say, Java, running on an IBM application server running on a Sun Solaris operating system[26]. Web services technology provides interoperability across programming language, runtime environment, operating system, and hardware.

Another key advantage is that Web services is an 'internet-native' technology, that is, the basic underpinnings leverage proven technologies from the World-wide Web, e.g., TCP/IP, DNS and URLs. Furthermore, one of the reasons for the rapid adoption of Web services technology is due to its 'firewall friendliness'[27]. That is, all Web services technologies are based on XML, a text-based language that is relatively easy to validate at an organizational

boundary. Moreover, although Web services traffic (SOAP messages) can be transported over arbitrary protocols, HTTP is by far the most popular transport today. Enterprise security administrators are generally much happier to allow an easily validable text message through the existing HTTP channel than they are to allow arbitrary binary data through specialist ports [66].

Chapter 3

Infrastructure of E-Learning Portal System

3.1 Overview of E-Learning System

3.1.1 E-learning System

E-Learning refers to learning that is delivered or enabled via electronic technology. It encompasses learning delivered via a range of technologies such as the internet, television, videotape, intelligent tutoring systems, and computer-based training [28].

E-Learning is a subset of the larger worlds of both “information technology” and education and training. It can be valuable when used as a part of a well-planned and properly supported education and training environment, but e-learning is not a magic bullet that replaces or renders the obsolete existing pedagogical theories and approaches [29]. Many learning and technology professionals believe that e-learning will have “been here” when we stop referring to it by a divided name and begin taking into consideration it as an integral part of a whole learning situation [55].

E-Learning does not do away with the existing educational methods and technologies. Rather, it complements them when used properly, because e-learning is rapidly rising as a useful and competent way to deliver the certain classes of training in the specific environments [56]. Recent advances in the availability and speed of Internet access and in the power and accessibility of personal computing platforms have considerably amplified the opportunities for the use of collaborative environments and the other distributed learning technologies [57].

New categories of products keep on emerging, some providing new capabilities and others combining the current functionality into the new product configurations. It can be a challenge to decide how these systems relate to each other and how they fit into a whole e-learning environment [58]. The appearance of e-learning does not mean that the existing software applications are out of date. Systems such as Student Administration, Human Resources, and Library Management provide the significant components of e-learning environments [59]. The challenge is to integrate these systems successfully with e-learning application services.

In discussing e-learning systems it is necessary to come to grips with another buzz word: learning objects. Learning objects are defined in dozens of white papers, standards, reports, and research papers, and all in wildly varying ways [60]. From an operational perspective, learning objects are chunks of data that are used by e-learning system- they are authored, stored, catalogued, assembled, delivered, and reported on [61]. A more down-to-earth approach is to think of a learning object as a digital part of a course ranging in size and complexity from a single graphic to an entire course.

As e-learning implementations grow in size and complexity, the demands on underlying technology become more rigorous. The technology infrastructure must have the capacity to support the users and network load, it must be scalable to support growth, it must be stable to ensure a high level of availability for learners, it must provide an open environment to support interoperability between components, and it must provide security to protect distributed users and content [62].

3.1.2 E-Learning Component

To understand how different systems might work together, it is useful to have a simple functional model of an e-learning application environment. Figure 3.1 below provides a visual representation of the components that make up an e-learning environment and the objects that must be moved among these components. This is not an architecture reference model for use but rather a conceptual model that can be used to position e-learning products and their functionality in an e-learning environment [28].

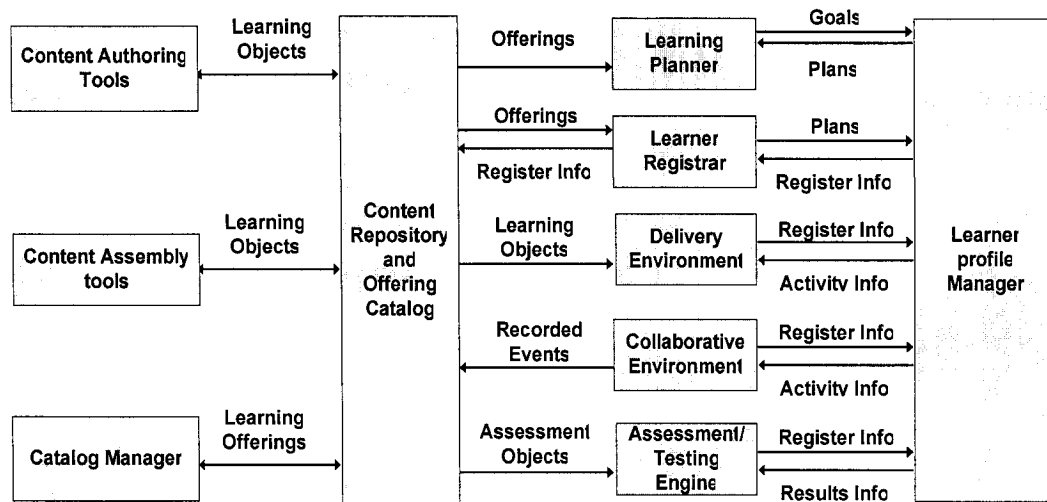


Figure3.1 E-learning Components [28]

Content Repositories and Offering Catalog

Content Repositories are learning object storehouses that can be accessed on one hand by people and systems creating content and on the other hand by people and systems using the content.

Content Authoring Tools

Content (and assessment) authoring tools and services allow subject matter experts and instructional developers to create and modify learning content objects.

Content Assembly Tools

Content assembly tools may support the creation and application of content templates that act as the basis for packaging content consistently and efficiently into learning modules.

Catalog Manager

Catalog management is the process of defining the learning that will be offered to different audiences, establishing learning plans (degree paths, certification paths, skill development curricula), scheduling the resources needed to support learning delivery, establishing the business processes for registering learners in offerings, and making the offering catalog accessible to the target audiences.

Learner Profile Manager

The learner profile manager makes learner information available to other components and retrieves and updates learner information on the basis of data reported by other components.

Learning Planner

Depending on the organizational context, learning can be planned by learners themselves, by teachers, by advisors, by curriculum administrators, by HR (Human Resource) managers, and by line managers.

Learner Registrar

The learner registrar component provides learners with access to learning offerings and administers the business processes related to that access.

Delivery Environment

The delivery environment provides the learner with access to learning content and other components of a learning environment such as chat, email, quizzes, multimedia players, collaboration tools, application sharing, shared whiteboards, equation editors, etc.

Offline Learning, Nomadic Learning, and Mobile Learning

Until recently, online learning meant learning through a Web browser connected to a network. Increasingly this is being challenged. Learners are using portable devices and even traditional computers may not maintain a constant connection to a server.

Accessibility

Student profiles can also provide accessibility requirement information that can be used to vary the delivery methods used for the learner. For example, hearing impaired learners may be provided with text transcripts of video presentations.

Collaborative Environment

Some e-learning delivery systems are built almost exclusively around synchronous delivery and collaboration. They are called virtual classrooms because they try to extend the physical environment and interactions of a classroom to an online setting.

Informal Learning

A significant amount of learning that occurs, particularly in the corporate space, is informal.

Assessment and Testing Engines

Assessment and testing may be integrated with learning content and delivered with it, or it may be managed as a separate process.

3.2 Related work

E-learning has increased in importance as people realize that the use of technology can improve the learning process. During the last few years, new learning environments have been developed. The research in the area of E-learning brought new ideas in setting up a new form of Web content that is meaningful to computers [79]. However, in general they are oriented to address a specific e-learning functionality. Therefore, in most of the cases, they are not developed to interoperate with other e-learning tools, which make the creation of a fully functional e-learning environment more difficult.

Aware of these problems, some organizations such as IMS Global Learning Consortium, Advanced Distributed Learning and IEEE Learning Technology Standards Committee are working to develop technical standards such as LOM (Learning Objects Metadata [31]) and SCORM [68]. These are recommended best practices and guides for E-learning technology. The main focus of their work has been on enabling content reuse through the description of the content.

In the E-learning domain, these standards are becoming increasingly important and becoming more advanced in terms of content interoperability.

LOM is gradually becoming the reference standard for educational systems managing learning objects of many kinds. LOM makes it possible to select and integrate relevant learning experiences from a relatively small library of learning objects. Reusable learning objects permit lessons to be generated and customized for specific groups or even for individuals [31].

SCORM has a very sophisticated way of packaging content via the manifest that makes it easy to reorganise the content or even to have several alternative organisations within one package for different purposes. This can greatly reduce the cost of creating learning resources for specific purposes. SCORM-compliant content can be used on any SCORM compliant learning management system. This means that content from many different sources can be used on any SCORM compliant LMS.

3.2.1 Learning Object Metadata

LOM is a multi-part standard that specifies Learning Object Metadata. Learning Object Metadata standard specifies a conceptual data schema that defines the structure of a metadata instance for a learning object. For this Standard, a learning object is defined as any entity -digital or non-digital- that may be used for learning, education or training [31]. For this Standard, a metadata instance for a learning object describes relevant characteristics of the learning object to which it applies. Such characteristics may be grouped in general, life cycle, meta-metadata, educational, technical, educational, rights, relation, annotation, and classification categories. The conceptual data schema specified in this part permits linguistic diversity of both learning objects and the metadata instances that describe them. This conceptual data schema specifies the data elements which compose a metadata instance for a learning object. This Part is intended to be referenced by other standards that define the implementation descriptions of the data schema so that a metadata instance for a learning object can be used by a learning technology system to manage, locate, evaluate or exchange learning objects. This Part of this Standard does not define how a learning technology system represents or uses a metadata instance for a learning object [30].

The purpose of this multi-part Standard is to facilitate search, evaluation, acquisition, and use of learning objects, for instance by learners or instructors or automated software processes. This multi-part Standard also facilitates the sharing and exchange of learning objects, by enabling the development of catalogs and inventories while taking into account the diversity of cultural and lingual contexts in which the learning objects and their metadata are reused. By specifying a common conceptual data schema, this Part of this Standard ensures that bindings of Learning Object Metadata have a high degree of semantic interoperability. As a result, transformations between bindings will be straightforward. This Part of this Standard specifies a base schema, which may be extended as practice develops, e.g., facilitating automatic, adaptive scheduling of learning objects by software agents.

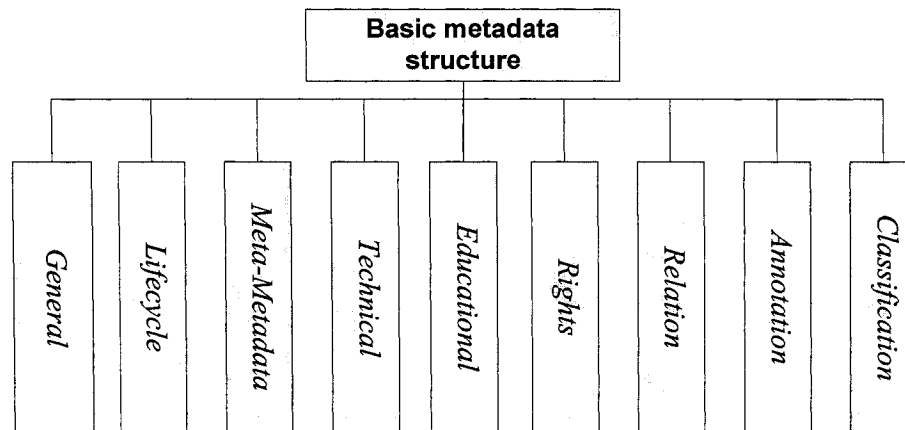


Figure 3.2 Structure of LOM [30]

Data elements describe a learning object and are grouped into categories. The LOMv1.0 Base Schema consists of nine such categories:

- The General category groups the general information that describes the learning object as a whole.
- 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.
- The Meta-Metadata category groups information about the metadata instance itself (rather than the learning object that the metadata instance describes).
- The Technical category groups the technical requirements and technical characteristics of the learning object.
- The Educational category groups the educational and pedagogic characteristics of the learning object.
- The Rights category groups the intellectual property rights and conditions of use for the learning object.
- The Relation category groups features that define the relationship between the learning object and other related learning objects.
- The Annotation category provides comments on the educational use of the learning object and provides information on when and by whom the comments were created.

- The Classification category describes this learning object in relation to a particular classification system. Collectively, these categories form the LOMv1.0 Base schema.
- The Classification category may be used to provide certain types of extensions to
- The LOMv1.0 Base Schema, as any classification system can be referenced.

3.2.2 SCORM SYSTEM

The Sharable Content Object Reference Model (SCORM) is a suite of technical standards that enable web-based learning systems to find, import, share, reuse, and export learning content in a standardized way [68]. SCORM is often described to be like a bookshelf housing volumes (specifications) that originated in other organizations. However, these specifications have been extended and additional detail and implementation guidance has been added (along with test software). SCORM is, therefore, more than just a collection of others work, though it directly relies on the source specifications [41].

SCORM is defined by the Advanced Distributed Learning (ADL) Initiative [32], sponsored by the Office of the Secretary of Defence (OSD) in United States. SCORM is a model that references a set of consistent technical specifications and guiding principle designed to meet Department of Defences (DoD) high-level requests for Web-based learning content [33]. SCORM is intended to be such a model that deals with the reusability, accessibility and interoperability of sharable learning objects. A key ADL requirement for learning content is the ability to reuse instructional components in multiple applications of the tools used to create them [33]. SCORM applies several technology standards: IMS Global Learning Consortium, Inc. [34], the Aviation Industry CBT (Computer-Based Training) Committee (AICC) [35], the Alliance of Remote Instructional Authoring & Distribution Networks for Europe (ARIADNE) [36], and the IEEE Learning Technology Standards Committee (LTSC). Building upon the Internet standards and the Shared Place infrastructures system developers have switched their focuses more on how to define reusable learning objects and develop new content models [37]. SCORM is divided into two main parts: the Content Aggregation Model and the Run-Time Environment.

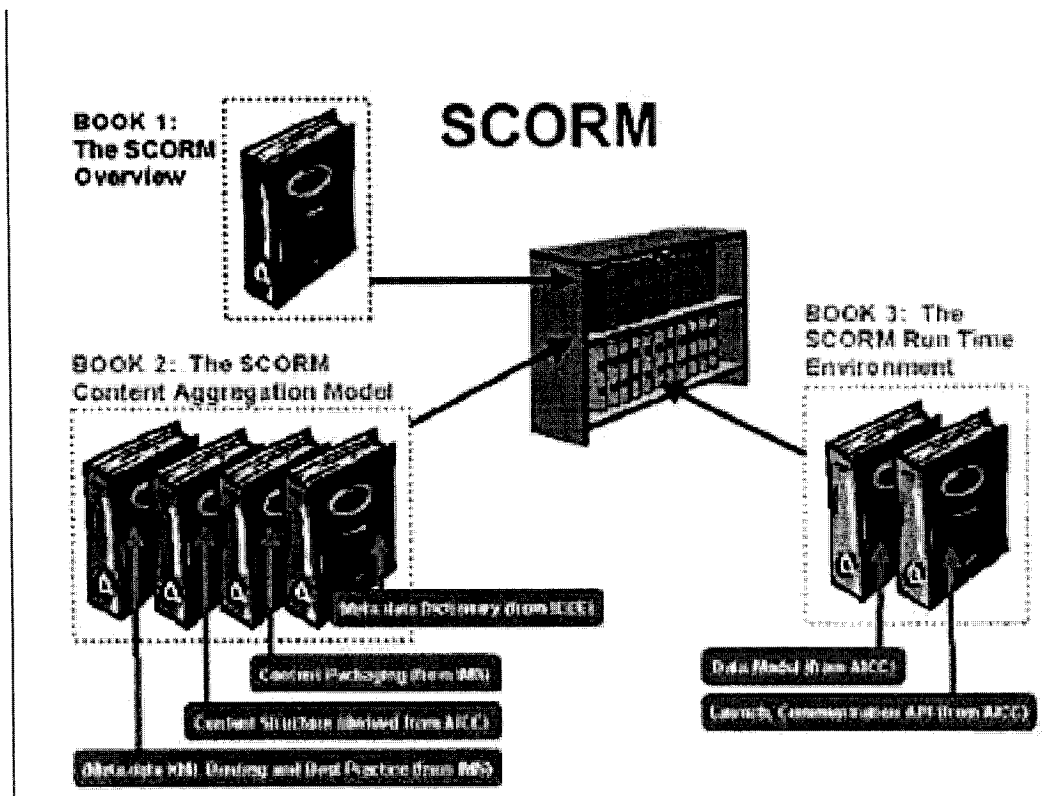


Figure 3.3 What is SCORM [41]

The SCORM Content Aggregation Model involves the creation, discovery and gathering together, or aggregation, of simple assets into more complex learning resources and then organizing the resources into a predefined sequence of delivery. It represents a pedagogically neutral means for designers and implementers of instruction to aggregate learning resources for the purpose of delivering a desired learning experience [38]. The SCORM Content Model is made up of the following components: Assets, Sharable Content Objects (SCO) and Content Aggregations. Asset is the most basic form of a learning object.

It is an electronic representation of media, text, image, sound or web page. A SCO represents a collection of one or more Assets that include a specific launchable asset that utilizes the Run-Time Environment to communicate other learning systems. A Content Aggregation is a map (content structure) that can be used to aggregate learning resources into a cohesive unit of instruction (e.g. course, chapter, module, etc.). The IEEE LTSC Learning Object Metadata (LOM) [39] is applied to Assets, SCOs and Content

standard set of data elements used to define the information being communicated, such as, the status of the learning resource.

Figure 3.5 below shows these three aspects of the Run-Time Environment [40].

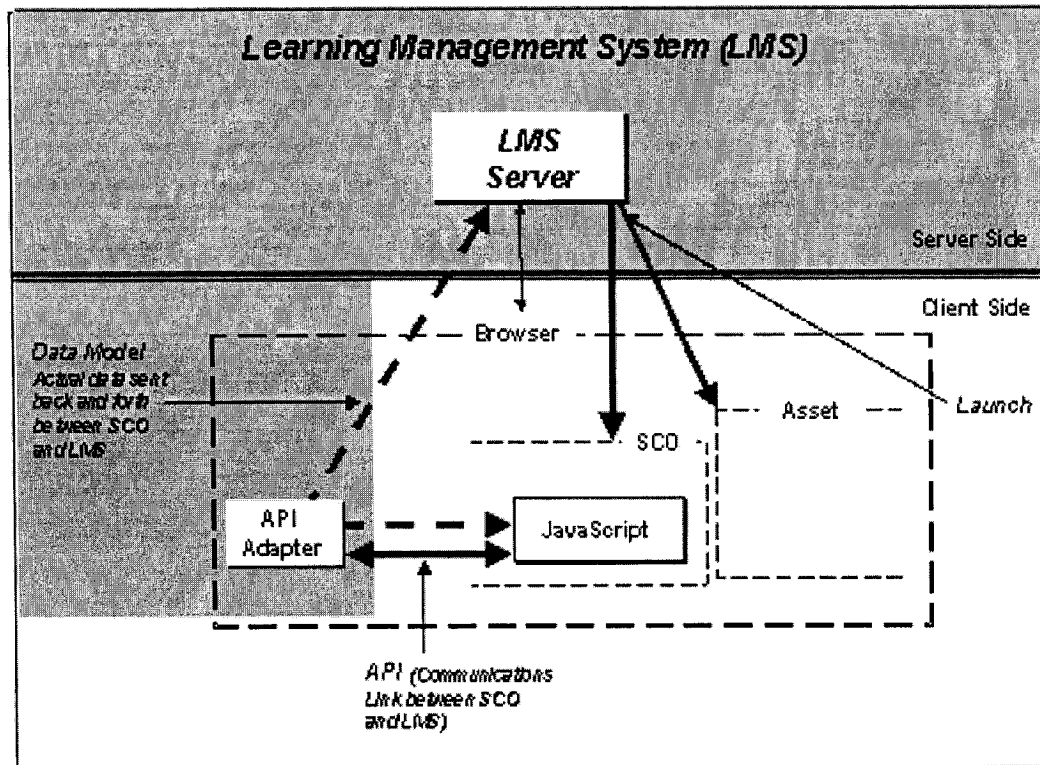


Figure 3.5: The SCORM Run-Time Environment [40]

Chapter 4

Analyzing and Designing the E-learning System

In this section we present our system design following Unified Modeling Language (UML) version 1.4 standards. The diagrams in the system design are drawn by using UML tool Rational Rose Enterprise Edition.

4.1 System Requirement

In order to raise the traditional Web-based distributed learning to a new level achieving a true dynamic model the following requirements must be given special consideration:

- Open architecture and interfaces: the framework should have an open architecture and open application interfaces to enable interaction and integration seamlessly between educational institutions learning service providers and other entities that enable distributed learning. The architecture is able to take advantage of the open, dynamic nature of the web by supporting just -in-time application integration.
- Durability and reusability: the system should be designed in an object-oriented model in which learning materials or application are treated as encapsulated components with well defined interfaces as long as the interfaces remain consistent these components can withstand technology changes without redesign, reconfiguration or recoding and can be reused in multiple applications and contexts.
- High interoperability for information exchange: the framework should have the ability to take learning components or application developed in one location with a different set of tools or platform in other words these components can be retrieved from anywhere, anytime with any device.
- Flexibility: All the sharable learning objects in this model are loosely coupled, which means the system should offer a dynamic mechanism that allows developers to easily add or remove components at any time.
- Accessibility: The learning objects can be published with well-defined description

in a universal repository for search, discovery and retrieval by other remote applications that need them.

- Compatibility with other systems: the system based on this framework should provide an open standard interface to communicate with the third party learning applications

4.2 System Prototype Description

Based on key requirements of a distributed e-learning system mentioned in Section 4.1, we can prototype a virtual learning community. In this community, there are three roles of users namely learners, instructors and administrators that take different actions. Figure 4.1 presents a learning administration, a learning collaboration system, and two learning systems, Institute-1 and Institute-2, in which provide different learning services and contents.

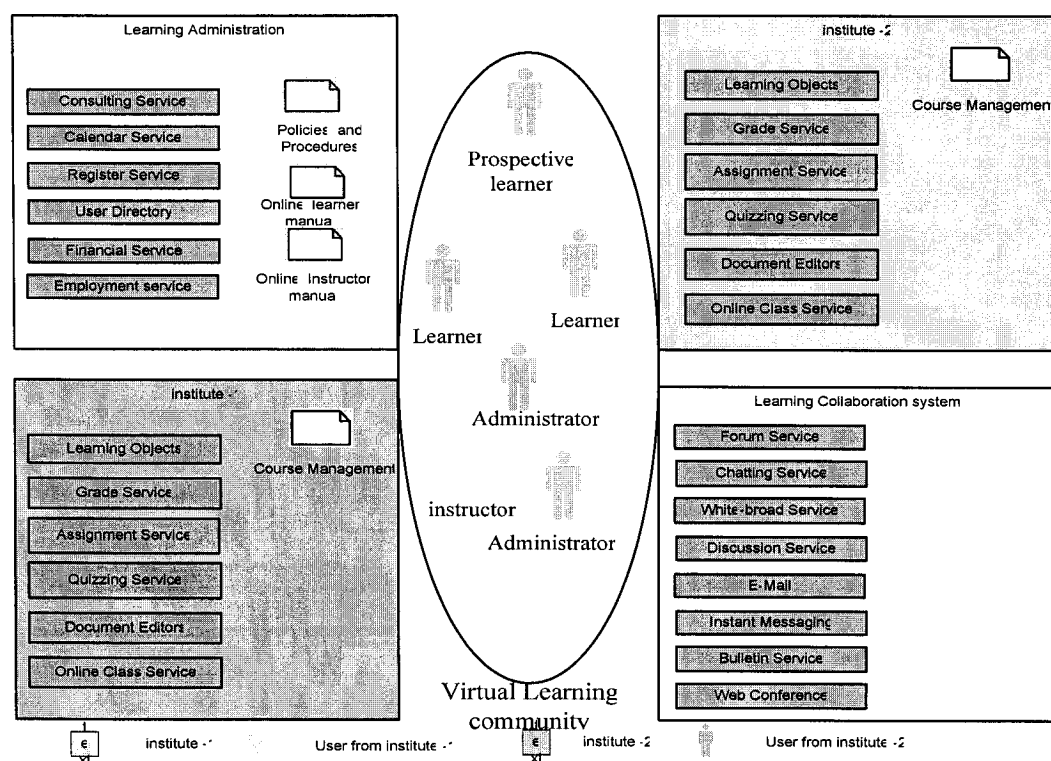


Figure 4.1 Virtual Learning Community

There are some services shared between institute-1 and institute-2 such as learning object, grade service, assignment service, quizzing service, document editor and so on. Learning

Administration has some functional services that include consulting service, calendar service, register service, user directory, financial service and employment service. The Learning Collaboration System provides user some service such as forum service, chatting service, white board service, discussion service, e-mail service, instant message, bulletin service and web conference. Learners from different institutes can login to The Virtual Learning Community through single sign on service to take their favourite courses from any institute which is part of the virtual learning community and communicate with their instructors from different institutes through The Learning Collaboration System such as chatting, forum or email.

The following scenarios can be achieved in the Virtual Learning Community (VLC):

1. Search for and display content: The VLC provides the ability to search content provided by various (internal and/or external) sources. The search is carried out on an index that has been pre-built, rather than directly on the content itself. The search can be free-text based, keyword based, or category based. Search results can be filtered based on the user's identified type.
2. Update personal information: A user of the VLC can update some of her personal information, such as address, phone number and e-mail address (note that this is personal information, rather than personal preferences).
3. Update VLC preferences and personalize VLC home page: The user updates personal preferences that control what is shown on his VLC home page, such as language or color scheme, in order to make the VLC more effective to use.
4. Submit an inquiry: This use case makes use of asynchronous collaboration allowing a user to submit a request/question and get a response later without the need to interrupt the flow of their activities. This allows the user to continue with other activities while the inquiry is being handled.
5. Hold an impromptu e-meeting: A VLC can support the ability to collaborate with other users through a chat or e-meeting feature. This type of collaboration is synchronous, with users interacting in real time, and requires that all participating users be online and available
6. Visit team-room: VLC can contain collaborative team-room(s) that are available to users who have the proper level of authorization. These users can access team-rooms to communicate with team members, share documents, or participate in processes that the team is involved in.

7. Create or edit a document: Advanced portal features include the ability to create and edit documents, spreadsheets, and presentations, eliminating the need to install these applications on individual desktop client machines. In this use case a user makes use of this portal feature to create and save a new document, which can then be selectively made available to other users to read and/or edit. A user makes use of the portal's document management feature to find and edit a previously saved document.
8. Create and manage content: A VLC user who has been identified as a content contributor can participate in the process of creating and approving content. Display of informational content (text, images, rich media) is one of the major features of VLC.
9. Publish content: Publication of content that has been submitted and approved through the content creation and management process makes that content available to VLC users. This use case represents the actual process of making that content available to the VLC. The content can be published through a manual invocation or through an automated, scheduled process.
10. Manage user accounts, policy and privileges: This use case represents the ability to manage user accounts and account-related policy. It encompasses the ability to add, delete, and modify accounts and the groups and policies that are used to designate user account privileges.

4.3 Proposed Framework

In this section, we propose the design of a framework that is able to comply with most of the system features and scenarios outlined in Section 4.2. Figure 4.2 and Figure 4.3 illustrate two block diagrams for a multiple-tier e-learning Web services-oriented portal framework running in distributed environments. This framework includes two parts: logic framework and system framework.

4.3.1 Logic Framework

The logic framework is comprised of four layers: access control layer, portlet layer, web service agencies layer and service layer.

The access control layer: A role based access control directs different users to different portals that include different portlets based on the user role. Once a user has been authenticated, authorization determines what resources a user is allowed access.

Administration refers to the ability to add, delete, and modify user accounts and user account privileges.

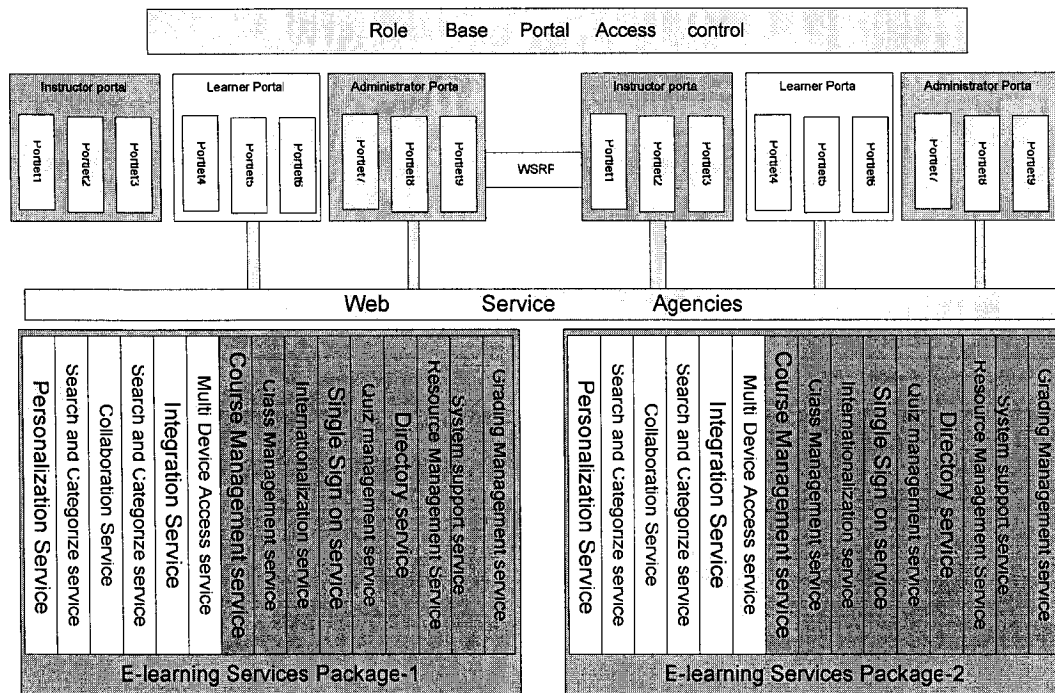


Figure 4.2 Portal Framework of Web Service based e-learning portal system

The portlet layer: this layer can be treated as a container of web services that consists of many local portlets called service requesters and some remote portlets. A learner portal can call remote portlet from another e-learning system as its portlet through web services for remote portlets to share system functionality.

Portlets can call web service inside its system or the other e-learning system. WSRP (web service for remote portlet) allows portals to display remotely-running portlets inside their pages without requiring any additional programming by the portal developers.

The web services agencies layer is a service engine that provides a pure Java engine whose main responsibility is to aggregate content from different sources and serves the aggregated content to multiple devices. It also provides a framework that allows the portlet layer of the portal to be decoupled from the service implementation details.

The service layer provides a set of service that includes collaboration service and administration service.

When a user logs in with a user ID and password, access control will verify the user ID and his/her user role and direct the user to the relative portal. Users can personalize his/her portlet through WSRP, if he/she finds some portlet interfaces that he/she prefers. The web services agencies then receives the request from portlet and helps the portlets to search the proper web service. Then web services agencies send the response back to portlet, if they find the relative web service.

4.3.2 System Framework

A system framework usually includes portal presentation service layer, portal engine layer, service provider layer (portal container), service broker layer and service requester layer. Portal presentation service layer provides customized and personalized pages for users through aggregation model. Page content is aggregated from a variety of sources via content and applications. The portal presentation framework simplifies the development and maintenance of the portal by defining the page structure independent the portlet definition. Portlets can be changed without an impact on the overall portal page structure. Aggregation Modules for the requesting device renders the overall portal page by collecting information from all the portlets on the page and adding standard decorations such as title bars, edit buttons, etc.

Portal provides a portal engine layer (a pure Java engine) whose main responsibility is to aggregate content from different sources and to serve the aggregated content to multiple devices. The Portal Engine also provides a framework that allows the presentation layer of the portal to be de-coupled from the portlet implementation details. This allows the portlets to be maintained as discrete components.

In our learning context, a Service provider provides any learning information, material or process as a self contained, self-describing modular service across different platform (J2EE, dot net and others). It could contain one or more of the following:

- Learning Objects and Learning Contents are reusable learning resources as defined by SCORM Content Model [43] and can be any content stored in a database repository or a file system.
- Learning Management Applications can be any program or function considered as shareable services by businesses or educational institution such as register management, course management and so on.

portlet or a remote portlet in a portal e-learning system, which needs some specific services to integrate learning contents, remote course registration, and shared whiteboard together. Our framework enables a true flexible integration among educational institutions or businesses, e.g. the role-base portal access control for a student to register a course from one institution and discuss course issue with instructors from other institutions. Based on this framework, a distributed e-learning system called UbiLearn has been analyzed and designed in a new and original way.

4.4 System Analysis

We used the industry-standard language: Unified Modeling Language (UML) to specify, visualize and construct the artefacts of the system. The goal of the analysis phase is to truly understand the requirements for the new system and develop a system concept that addresses them [44]. The following sections present some UML models about typical scenarios such as to register a course and to schedule a course. The requirements based on the previous scenarios in the section 4.2 are first visualized into a use case model, then a structural model is derived to realize those use cases, and finally a behavioural model is constructed to specify the internal processes.

4.4.1 Use Case Model

In the Course Management Component, there are four actors: Learner, Instructor, Administrator and Learning Service Provider.

The following table gives the detailed use cases related to these actors.

User role	Action
Learner	• Browse course information • Register courses • Take online lessons, tutorials or assessments • Take an online test • Collaborate with other learners or instructors using discussion forum, on line chat or whiteboard

Instructor	• Construct online learning resources like online courses, tutorials and assessments • Create learning objects • Publish a test • Search learning objects • Evaluate learner performance • Collaborate with other instructors or learners using discussion forum, on line chat or whiteboard
Administrator	Maintain course related information like course time schedule, course registrations, etc
Other Learning Systems	• Search learning services from UDDI • Get shared course information • Register shared courses remotely • Get or invoke shared learning objects remotely

Table4.1: User role and action

The follows are the use case diagrams, which illustrate the relationships between different actors.



Figure4.4: Use Case Diagram for Actors: Learner and other Learning Systems

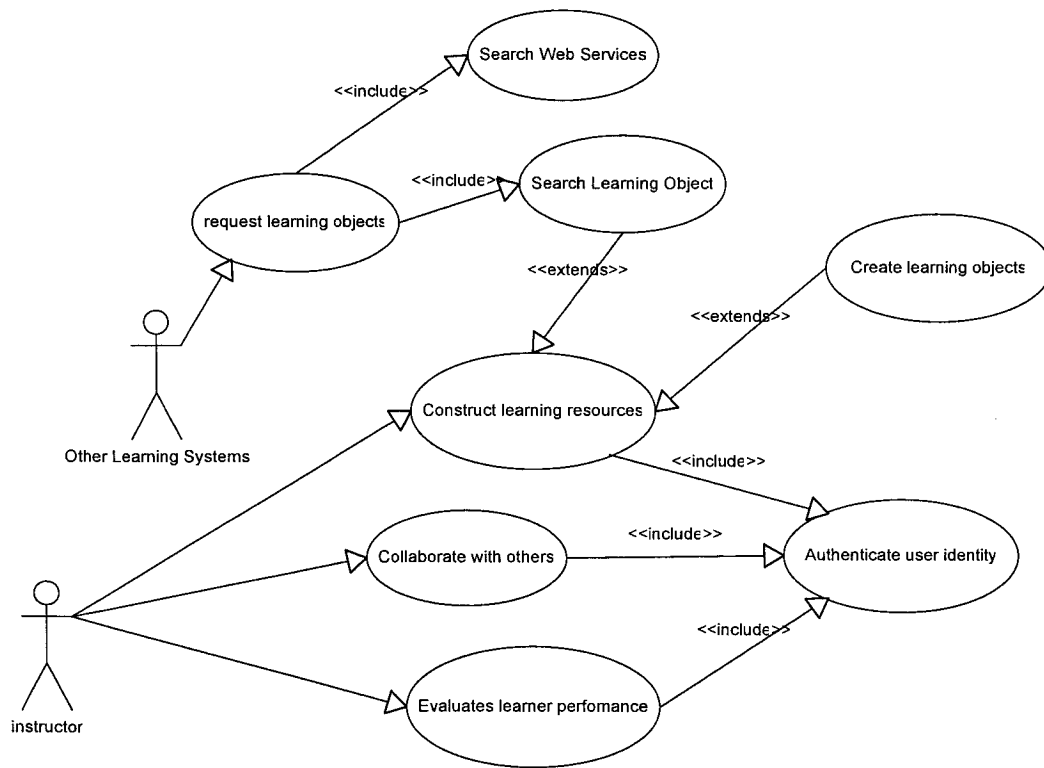


Figure 4.5: Use Case Diagram for Actors: Instructor and Other Learning Systems

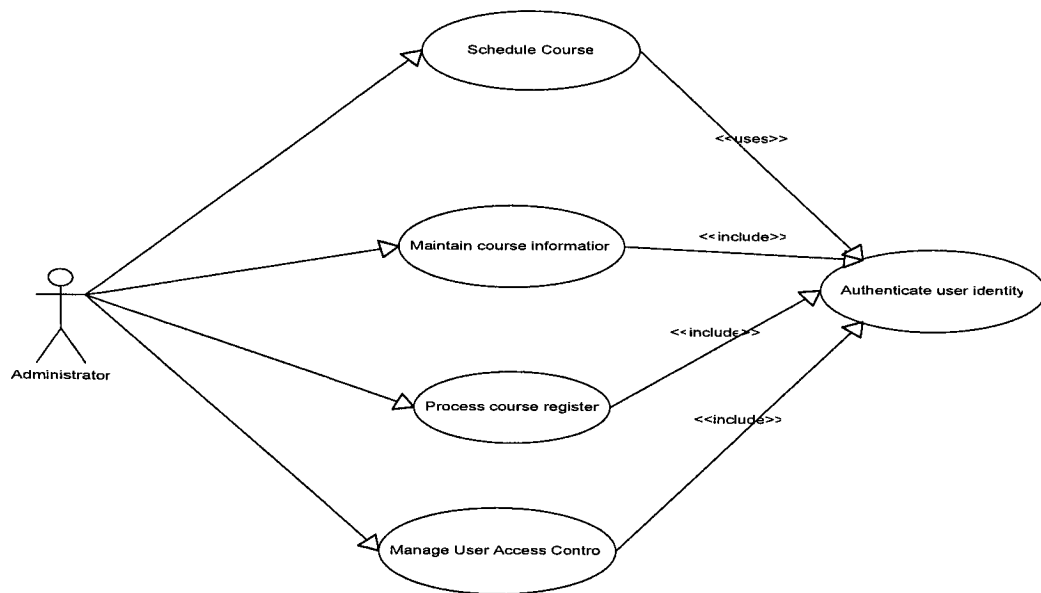


Figure 4.6: Use Case Diagram for the administrator as actor

Some association relationships like “extends” and “includes” are used in our model. An “extends” relationship between use cases means one use case optionally adds the

functionality of the other use case when certain conditions are met. An “includes” relationship means one use case completely encompasses all the functionality of another use case [45][75].

4.4.2 Interaction Sequence Model

Sequence diagrams are used to model the interaction between object instances in the context of collaboration [67].

The same use case realizations are illustrated by sequence diagrams shown below.

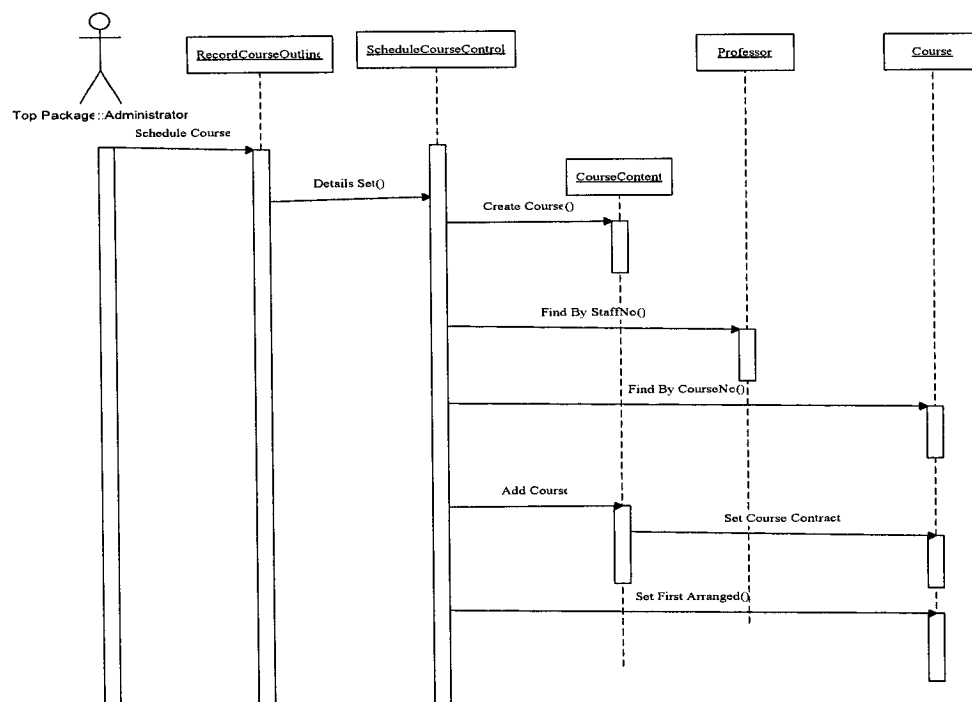


Figure 4.7: Interaction Sequence Model for Scheduling a Course

Figure 4.7 presents how an administrator schedules a course with course control. The Administrator creates a request for scheduling a course. Course control creates a course schedule through finding by Staffno(), CourseNo(), and then set a course contract.

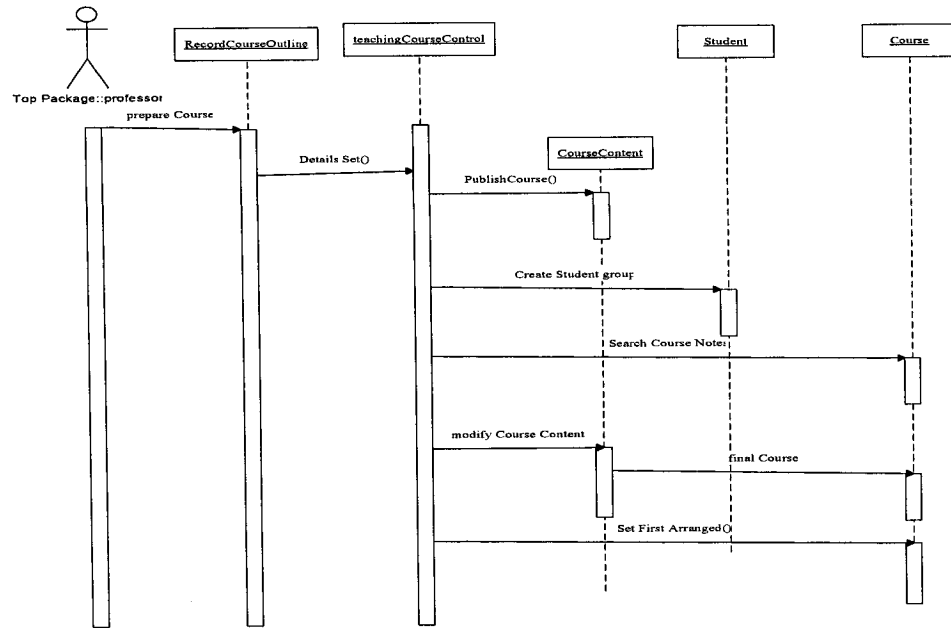


Figure 4.8: Interaction Sequence Model for Teaching a Course

Figure 4.8 shows how a professor teaches a course with course control. The Professor builds a request for preparing a course. Course control creates a course model through creating a student group, searching course notes and modifying course content, and then sets a final course.

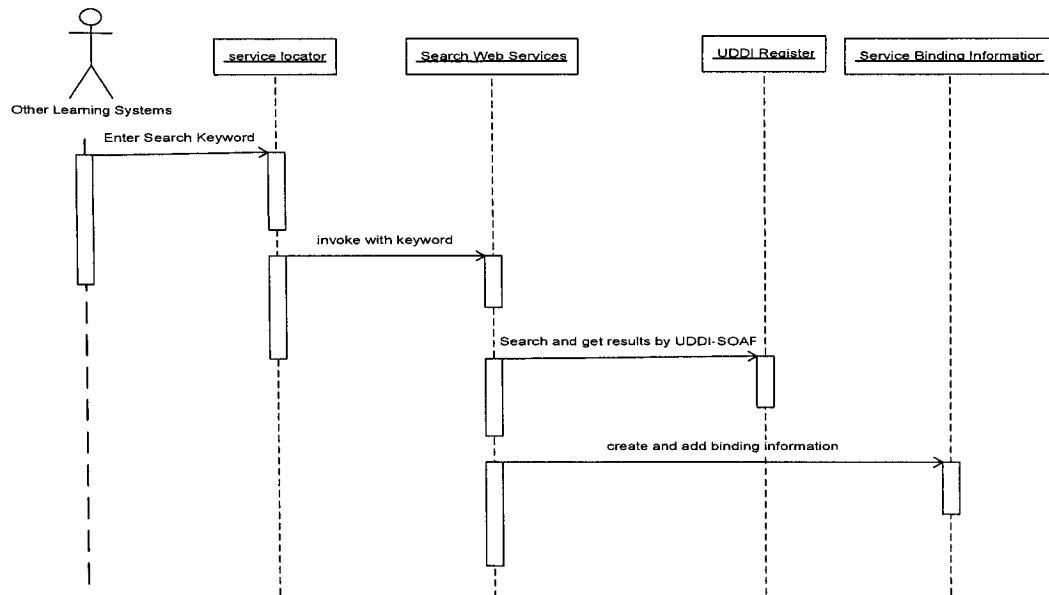


Figure 4.9: Sequence Diagram for the Realization of Use Case for Searching Web Services

Figure 4.9 shows how an external learning system uses the service locator interface to search its desired learning services and get their binding information through UDDI-SOAP messages.

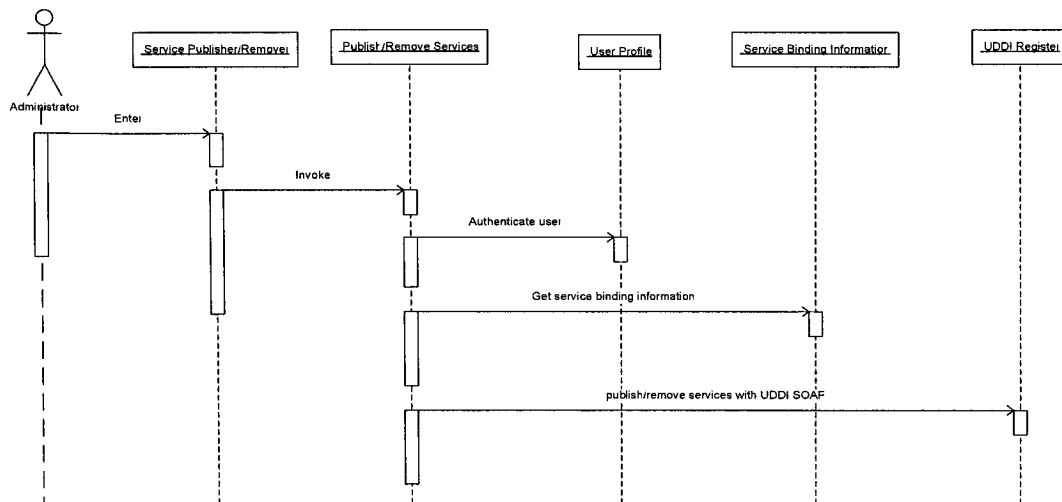


Figure 4.10 Sequence Diagram for the Realization of Use Case: Publish/Remove Web Services

In figure 4.10, a system administrator also uses UDDI SOAP to publish and remove Web services. He needs his user identity to be verified before he can manipulate UDDI registry.

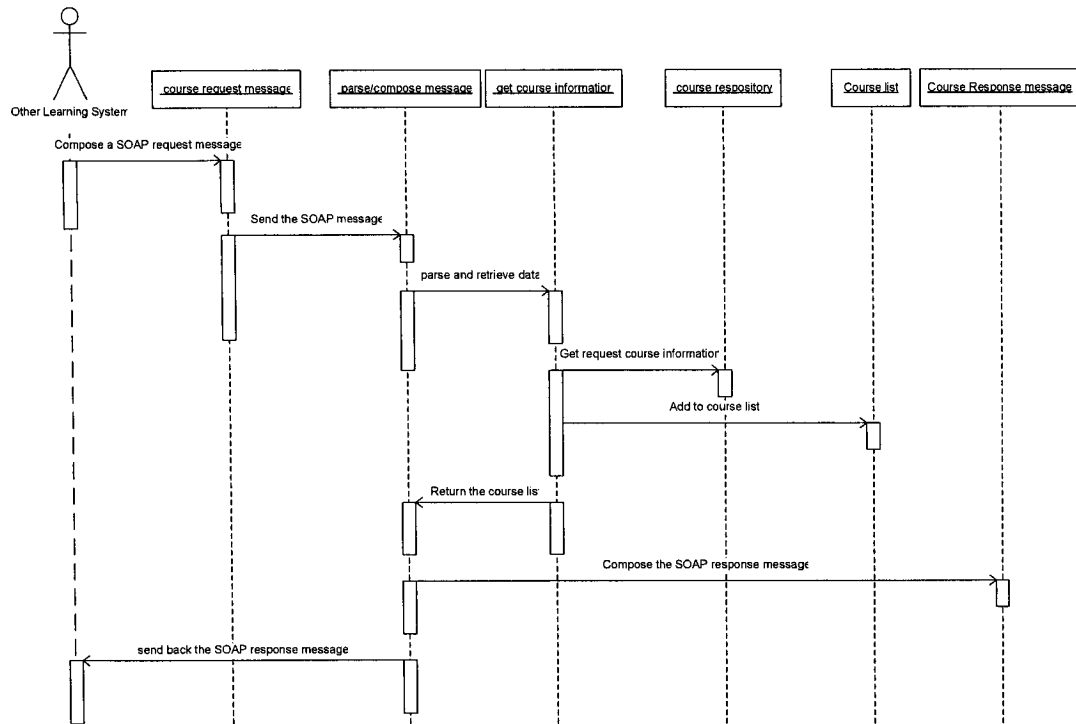


Figure 4.11 Sequence Diagram for the Realization of Use Case: Get Shared Course Information

Figure 4-11 shows the sequential details of how other learning system get shared course information. Both the course request and course information response are transmitted by SOAP.

4.5 System Class Design

Through the analysis phase we have seen that there are four major requirements to be considered: verify identity, get shared courses, register shared courses for external learning systems and publish/remove web services for administrators. The final class design is determined in this section by means of three approaches: Package View, Class View and Component View.

4.5.1 Package View

Packages are used to represent groups of classes and serve to partition the logical model of an application [46]. The following figure illustrates the packages and their relations to the system:

There are six main packages:

- Portlet Package: as a portlet container, it contains the portlet interface classes to personalize portlets.
- Web Services Agent Package: it contains the remote interface classes for external learning systems to access Web services and also contains a UDDI query class.
- SOAP Messaging Package: it contains the Web service request client, the request message handler and the Web services using SOAP messaging mechanism.
- SOAP-RPC Package: it contains the RPC service client, the RPC service interface and the RPC service using SOAP RPC mechanism.
- Common Learning Services Package: it contains the classes of implementing the actual learning services that the service requesters are ultimately interested in accessing.
- Service Registration Package: it is an independent package that contains the classes for publishing and removing Web services.

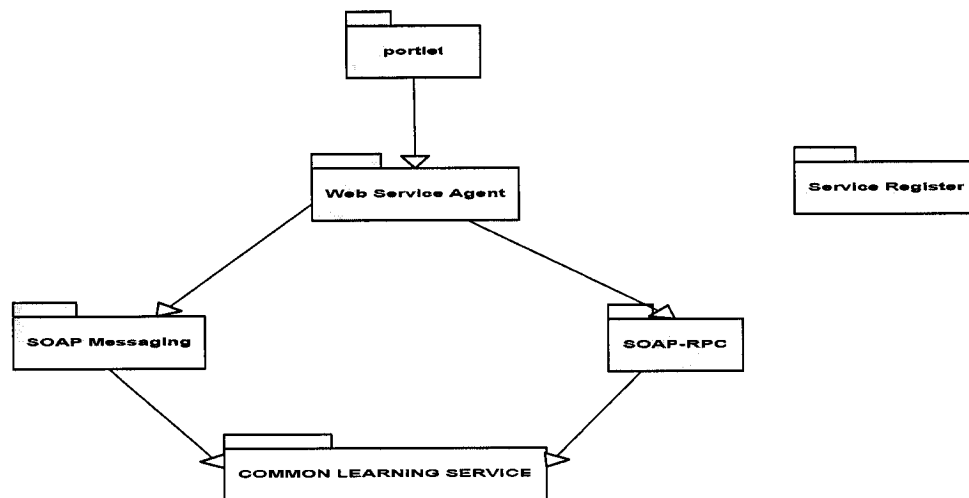


Figure 4.12: System Package View

4.5.2 Class View

Figure 4.13 shows how the main functions are addressed properly:

- **JAXRQueryByName**: the class for searching Web services from a UDDI registry using Java API for XML Registries (JAXR).
- **RemoteCourses**: the portal class for getting the shared course information through SOAP messaging.
- **MyRemoteCourses**: it is similar to RemoteCourses class but for a specific registered course only.

Figure 4.13: System Class Diagram

- UserVerifier: the RPC client class for user identity verification.
- EnrollCourses: the RPC client class for registering remote courses.
- OppositeIF: the interface providing RPC based Web services using Java API for XML-based RPC (JAX-RPC).
- OppositeImpl: the implementation of the interface OppositeIF.
- CourseListRequest: the client class for sending course information request in a SOAP format.
- Page Flow: a HTTP servlet class, in which there is the SOAP messaging based Web service for providing shared course information using Java API for XML Messaging (JAXM).
- PortletAdapter: manage portlet html controller for portlet modes that include Edit, Help, Title, View and so on.
- RequestSAXParser: a parsing class based on the standard SAX (Simple API for XML Parsing) for extracting user request data from the SOAP request message.
- OrgPublisher, OrgRemover: the classes for publishing Web services to UDDI registry and removing from UDDI registry respectively.
- Course: the course class with its main properties.
- Services: the interface of the ultimate common learning services for the Web services tier to invoke.

4.5.3 Component View

Figure 4.14 clearly demonstrates how those Web services components are related to construct the Web services tier of our system architecture. Once a service request from Client component is sent in the portlet, the corresponding action is activated to either call the common learning service locally or dispatch the request to the Web services agent component to invoke the remote service of other LMS. This services invoked could be user verification or course registration by instantiating one of the Web service client components. On the other hand, the system's learning services to external learning systems are ready to be invoked with their remote portlet interfaces or HTTP servlet portals. The

Web services publish/remove component is responsible to publish/remove those services binding information such as its URI and its WSDL document into/from a UDDI registry.

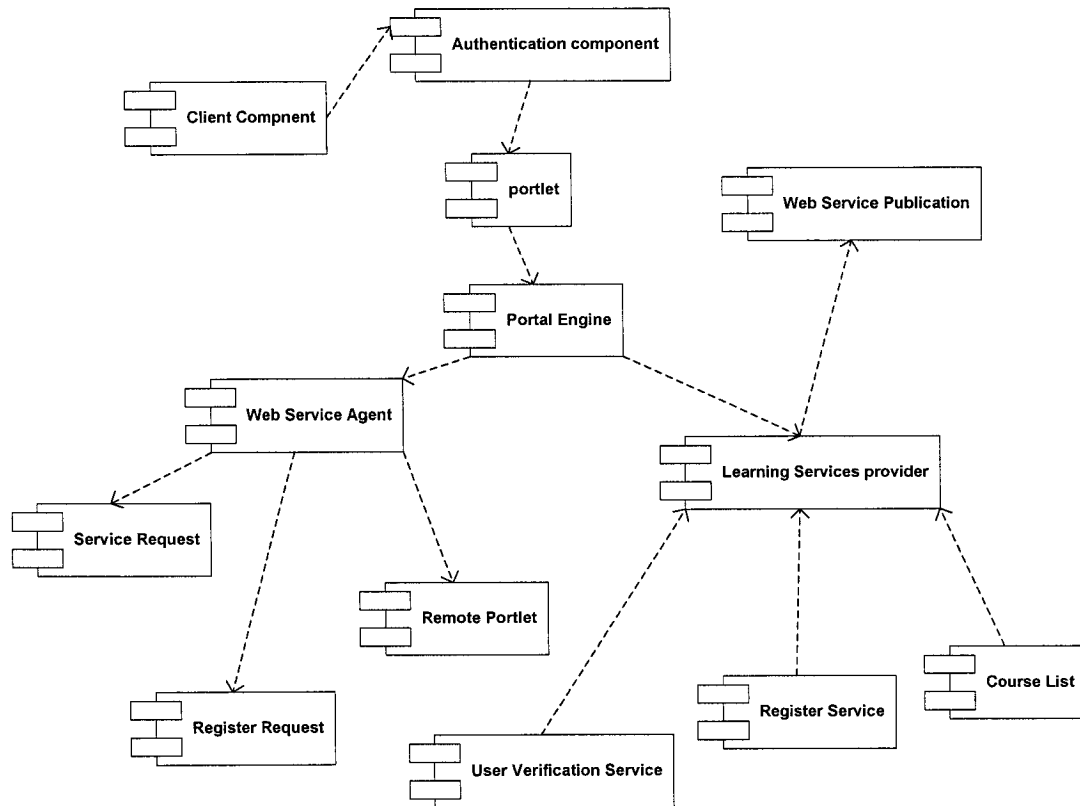


Figure 4.14: System Component Diagram

Chapter 5

Implementation

Ubilearn is a distributed learning portal system based on the Web services framework, which is designed following the design presented in Chapter 4 and implemented by the Java compatible technologies at the Multimedia Communication Research Laboratory, University of Ottawa. This chapter describes how the Web services in Ubilearn are implemented, the essential technologies for building Web services, the technologies for sharing portlet through web service for remote portlet is also introduced.

5.1 The Framework Implementation

The overall system architecture is described in section 4.3. The architecture in this section is more focusing on the implementing flows between different layers. Flows 1, 2, and 3 are the normal ways on how a Web-based learning system works.

1. From the portlet container (e.g. an Internet browser or a Web enabled wireless device), in which the client contents are presented by the forms of HTML, JSP (Java Server Pages) or XML, learner or instructor can submit their information through HTTP requests to the Server Object Layer and receives HTTP responses.
2. Form Beans and Action Beans, which systematically represent the user request data and user actions, dispatch the requested data to a specific Server Object such as a Java Servlet, a Java Message Service (JMS) or an Enterprise Java Bean (EJB).
3. The Server Objects use a Database Connection Layer such as JDBC (Java Data Base Connection) API to get information from the database.

Flows from 4 to 8 indicate the interactions among service request layer, service agent layer and service layer of a Web services-based learning system.

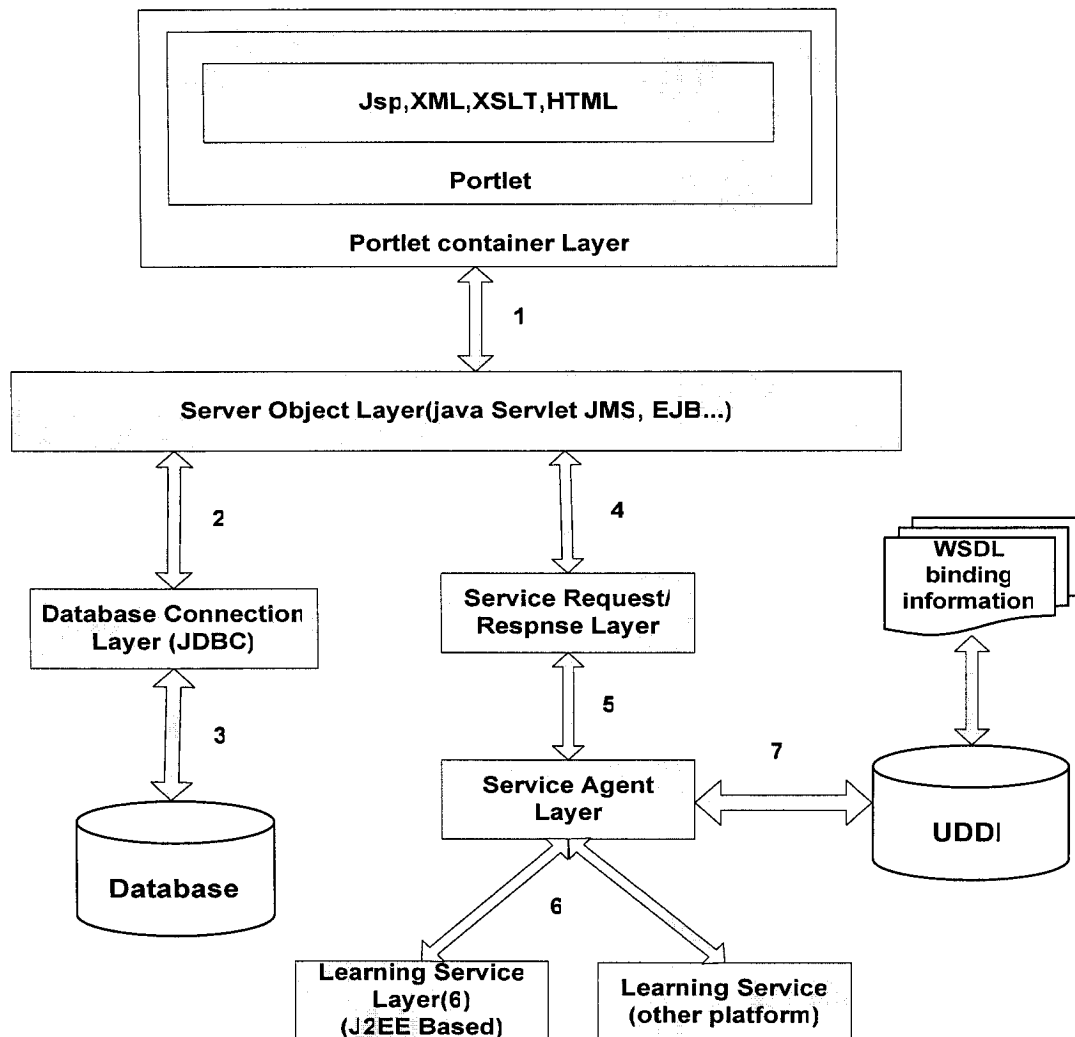


Figure 5.1: Implementation Architecture of the System

4. A service requester in Service request layer sends SOAP requests and gets SOAP responses over HTTP.
5. The Service Agent Layer parses the request streams and transmits the data to the corresponding Web services components. Once the data is extracted from the request stream, the same Server Objects will be invoked by the Web service components.

6. If a Server Object needs a remote learning object or service, it can locate and invoke the object or service through the Web Services Proxy Layer where the Web service clients reside.
7. By doing a look up in a UDDI registry, the Web Service Agent Layer can search and locate the desired learning service. The location of WSDL binding information will be sent back as a SOAP message. The binding information of our own sharable learning services is also published here. After the service Agent Layer gets the binding information, it can directly invoke the learning service by passing the essential data indicated by the WSDL file in a SOAP message over the Internet or Intranet. And the learning service could be J2EE-based or other platform-based residing on any platform.

5.2 The Role-Based Portal Access Control

For managing complexity of security administration in distributed systems such as e-learning, e-business and e-government, access control has become one of the most challenging issues. Access control is the process to identify users and grant with privileges of accessing services and resources [76]. A variety of models and mechanisms on access control have been presented on protecting private and confidential resources and controlling access to services and from unauthorized users. Access control systems usually provide authentication, authorization, and administration. Authentication is a process in which users are challenged for identity credentials so that it is possible to verify that they are who they say they are [77]. Once a user has been authenticated, authorization determines what resources a user is allowed to access. Administration refers to the ability to add, delete, and modify user accounts and user account privileges.

Being increasingly integrated into the security mechanism, role-based access control (RBAC) is a proven technology for managing and enforcing security in organization wide systems [78]. RBAC is more scalable in that it allows users to access systems and information based on their role within the organization rather than user-based security specifications. We implemented our Ubilearn by utilizing Web services with portal technologies. Extending Java Servlets, portlets can be used as a user interface connected to

all services located on local or remote servers. Being a container for portlets, portal provides a single entrance for any users to access multiple, heterogeneous services through this point.

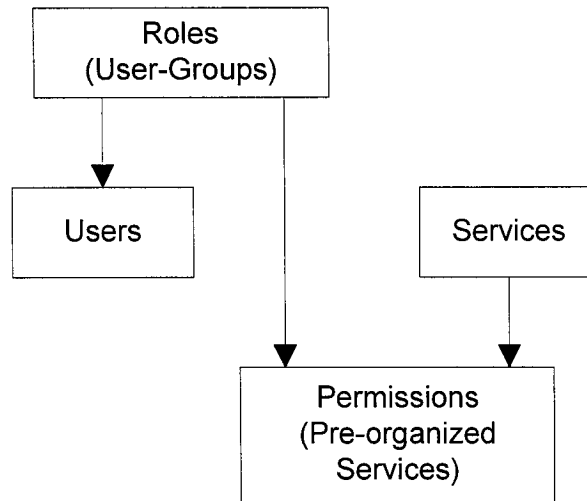


Figure 5.2 Relationships between users, roles, permissions and services [47]

By analysing role characteristics of an online institute, we classify Ubilearn users into three groups (roles), administrators, instructors, and learners, which are used to name three portals; and then we categorize services by integrating them into those pre-defined portals according to the relationships between service functionalities and roles pre-assigned to users. In the Figure 5.2, permissions are represented by corresponding pre-organized services that can be accessed by only the users corresponding to their roles.

Figure 5.3 presents a mechanism of a user accessing the Ubilearn system. The advantages for the mechanism are as follows: once a user is authenticated and the user's role data is retrieved, the portal container directs the user's browser to the corresponding portal the user is entitled to; any change made to a group's permissions will automatically be inherited by all members of that group; changes to a group's permissions to access services can be made very easily, e.g., by adding or removing portlets within that portal; if an individual user is switched to another group, we can only change the role of user by removing the role and assigning a new role to her/him. When a user, let's say, Jill, browses our Ubilearn welcome page, she can use either register if she is a new user or login if her identity is activated in the "login/Registration module".

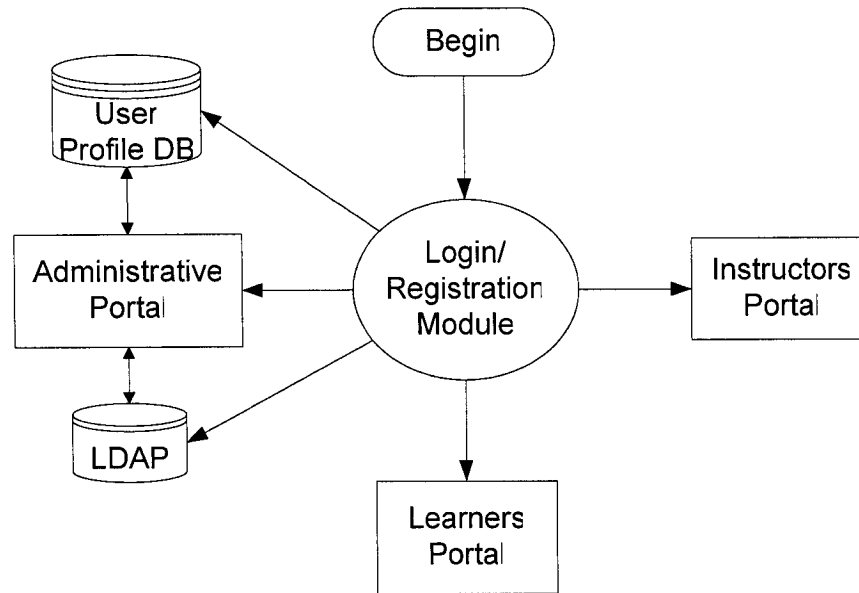


Figure 5.3: Role-based portal access control [50]

To specify a role to the user, we have two methods: one is to link the user's profile to a static role manually; the other method is to link the user's profile to role automatically based on her profile attributes. We apply first method to our role-based portal access control model and explain it as follows. Suppose Jill is a new user and she wants to access services and learning resource as a learner, she has to register and let an administrative user, lets' say, John, links her profile to a role before she can access those services through a portal. Upon Jill registering her profile which is stored in the user profile database, John will link the user profile to a role she (Jill) is entitled to and he will update her identity into LDAP server so that Jill can access those entitled services via a portal. when Jill logs into the system next time, the login module will check with the LDAP directory to see whether Jill's credential is matched and if yes, the login module will direct Jill's browser to learner portal, where a group of portlets that connected with those corresponding services and through those services, she can reach those expected learning resources.

5.3 Portlets Access to Web Services

Unlike servlets, JSPs, or Java Page Flows, portlets are units of code and UI that we can reuse in a given portal. Portlets also are containers of servlets, JSPs, or Java Page Flows. Each portlet has a UI to present data/content, and code to process user interactions [70]. Once a portlet is created, portal administrators and/or end users can add the same portlet several times to pages in various portals and desktops. Portal administrators and/or end users can then customize various properties of the portlet (e.g., the title of a portlet). In doing so, portal administrators and/or end users are dealing with specific usages of a portlet. That is, in this process, portlet instances are being created and each instance is being customized.

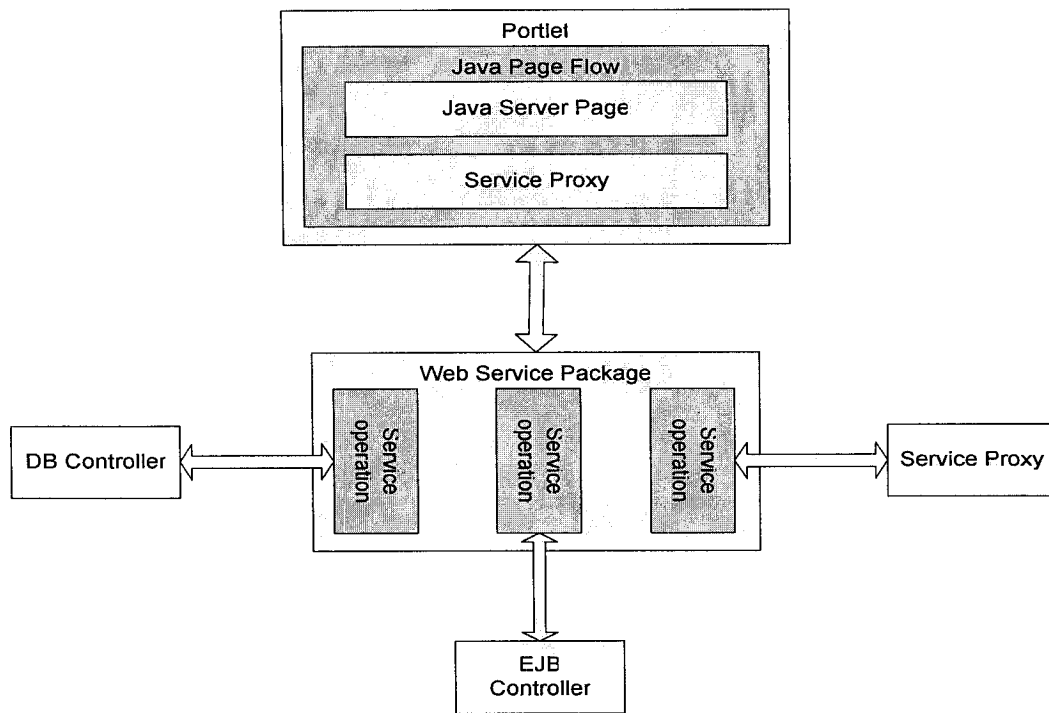


Figure 5.4: Portlet and service

Figure 5.4 shows how to call a web service from a portlet. Portlet as Service Requestor is a component in the application that wants to receive a service from another application. It does not know where this other application is or how to locate it, so it turns to the Service Registry and requests a find operation. The Service Provider publishes this service as a

Service Proxy in the Service Registry. Service Proxy is a java proxy that helps the portlet to communicate with web service.

Figure 5.5 shows how to create a web service client though loading a java proxy to a portlet. Firstly, the button “Java Proxy” needs to be clicked. When prompted, we save the file to the hard disk. Secondly, we save the file to the WEB-INF/lib directory of the portlet from where we wish to use the proxy. The default name of the file is <web service name>.jar; we accept the default name unless it conflicts with an existing JAR file in WEB-INF/lib.

If we want to use the web service proxy JAR from more than one project in our application, we can add it to the Libraries folder at the root of our application. Saving the JAR directly to the APP-INF/lib directory at the root of our application will automatically add the JAR file to the Libraries folder.

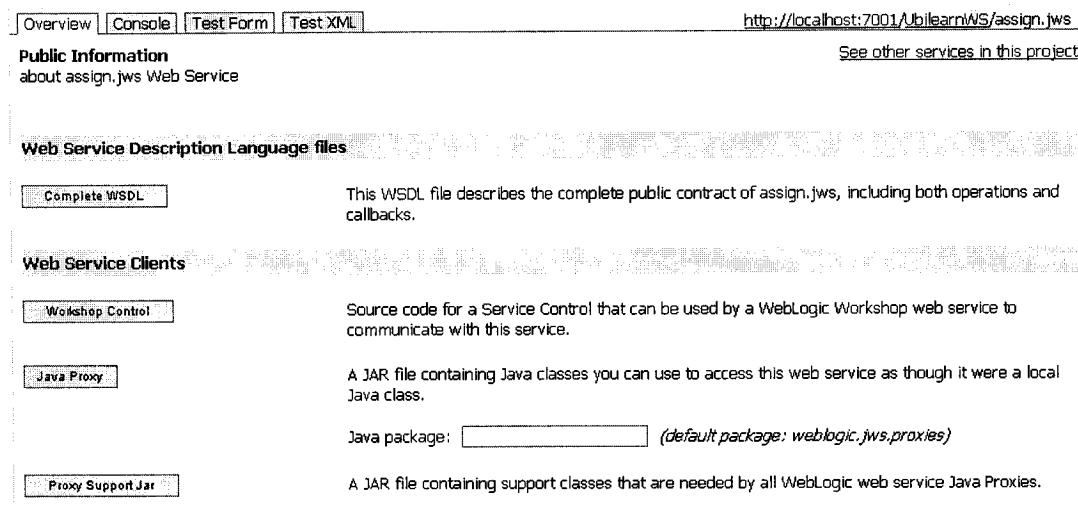


Figure 5.5: Web Service Testing Console

Thirdly, add an import of the web service proxy package as shown here:

```
<%@ page import="weblogic.jws.proxies.*" %>
```

Fourth, create an instance of the proxy class as shown below. The generic proxy class is the name of the web service with "_Impl" appended to the end:

```
Register_Impl proxy=new Register_Impl();
```

Fifth, the generic proxy returns protocol-specific proxies that in turn contained the actual interface of the web service. The following example assumes a communication with the web service using SOAP. We use registerXXX methods to get proxies for other protocols:

```
RegisterSoap soapProxy = proxy.getregisterSoap();
```

Sixth, call the appropriate methods on the protocol-specific proxy, as in the following example:

```
soapProxy.register(Userid, Password);
```

5.4 System Internationalization

5.4.1 Internationalization and Localization

The idea of using the same program but changing its properties according to the cultural traditions of different peoples is called internationalization or shortly i18n [48].

Once we have written an i18n program, we may want to add a new language. This is not an i18n problem. In general, we need a person who will translate messages from the program for a specific language. This problem is called localization, or i10n. I10n refers to the implementation of a specific language for internationalized software, or in other words, the creation of localized objects according to the specific region's rules [65].

5.4.2 Making a Program I18n-Compliant

Figure 5.6 represents all necessary steps for producing an i18n program:

To create an i18n version, we have to edit a non-i18n program. If we use a special editor mode we will create an additional file at the same time, called a properties file. (That is also called POT file, where POT stands for portable object, and the letter T is for template.)

- Copy the template properties file into XXXX_cn.properties files, where “cn” refers to a certain language. Translate messages into the language cn.
- Compile our source; put the binary program and all kinds of properties files into the right place. This and the previous steps are better accomplished with a Makefile.

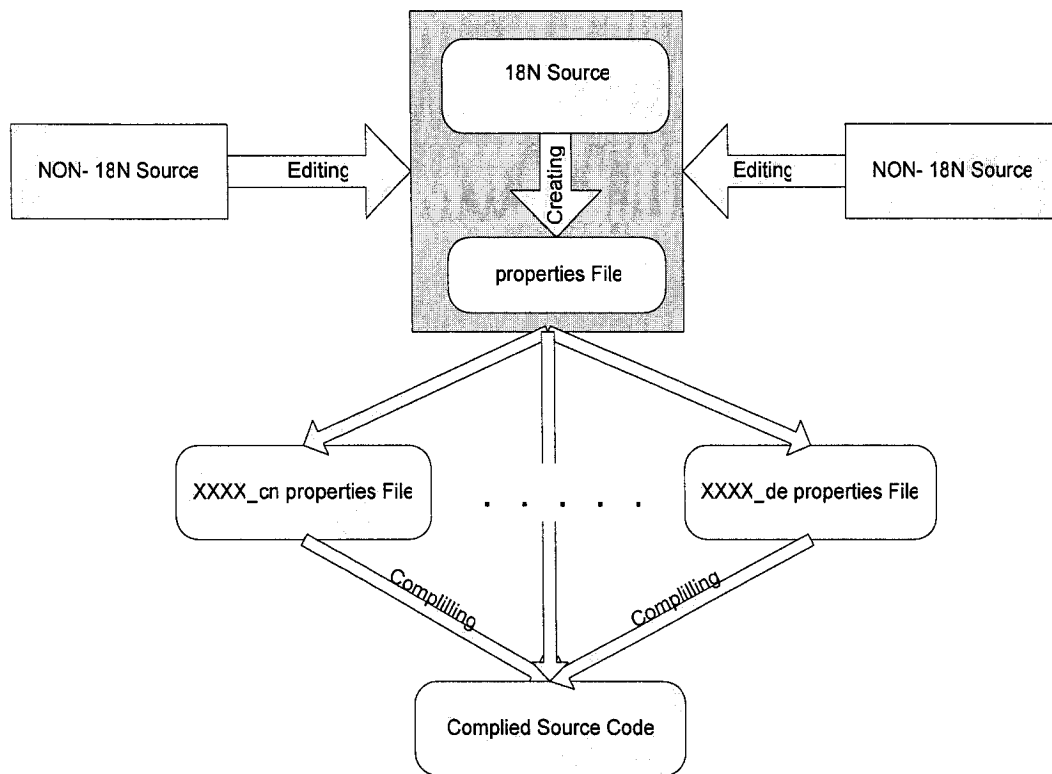


Figure 5.6 the process of Internationalization [49]

5.4.3 Making Ublearn internationalized with Weblogic

In Bea Weblogic Development environment, the Internationalization and Localization JSP Tags are presented in Table 5.1. They are contained in i18n_taglib.jar and l10n_taglib.jar.

I18n:getmessage	Retrieves and displays localized static text or messages from a JspMessageBundle.
I18n:localize	Defines the language, country, variant and base bundle name to be used throughout a page when accessing resource bundles with the i18n:getmessage tag.
L10n:include	Includes a localized version of a page.
L10n:forward	Forwards to a localized version of a page.
L10n:resolve	Resolves to a localized version of a page.

Table5.1: I18n &L10n Jsp Tags [51]

```

<%@ taglib uri="il8n.tld" prefix="il8n" %>
<%@ page import="java.util.*"%>
<%@ taglib uri="render.tld" prefix="render" %>

<% String locale=(String)session.getAttribute("locale"); %>
<il8n:localize language="<%=locale%>" bundleName="register"/>

```

Figure5.7: I18n resource file

Figure 5.7 represents public I18n resource file that is included in the every i18n jsp file. These i18n jsp files listen variable “local” to make a decision which local file will be used. At the same time, the related local file will be bind to i18n jsp file.

Non-i18n jsp label

```
<td>First Name:</td>
```

i18n jsp label

```
<td><il8n:getMessage messageName="register.label.firstname" /></td>
```

English properties file

```
register.label.firstname = <span class="bea-portal-body-ubilearn-content"> First
Name: </span>
```

Arabic properties file

```
register.label.firstname = <span class="bea-portal-body-ubilearn-content"> :
\ۛ0627\ۛ0644\ۛ0627\ۛ0633\ۛ0645 \ۛ0627\ۛ0644\ۛ0627\ۛ0648\ۛ0644 </span>
```

Chinese properties file

```
register.label.firstname = <span class="bea-portal-body-ubilearn-
content">\ۛ540D:</span>
```

German properties file

```
register.label.firstname = <span class="bea-portal-body-ubilearn-content">
Vorname: </span>
```

French properties file

```
register.label.firstname = <span class="bea-portal-body-ubilearn-content">
Prénom: </span>
```

Spanish properties file

```
register.label.firstname = <span class="bea-portal-body-ubilearn-content">
nombre: </span>
```

Figure5.8: JSP label sample

Figure 5.8 shows how to internationalize a simple jsp label. Firstly, we should replace Non-i18n jsp label with i18n jsp label. Secondly, translate label information to different language and put them to different properties files.

5.5 Web Service for Remote Portlet

5.5.1 How WSRP works

Web Services for Remote Portlets (WSRP) is a Web services standard created by OASIS that allows for the plug-n-play of visual, user-facing web services with portals or other intermediary web applications [52]. By defining a set of common interfaces, WSRP allows portals to display remotely-running portlets inside their pages without requiring any additional programming by the portal developers. To the end-user, it appears that the portlet is running locally within their portal, but in reality the portlet resides in a remotely-running portlet container, and interaction occurs through the exchange of SOAP messages. WSRP services can be published into public or corporate service directories (UDDI) where they can easily be found by intermediary applications that want to display their content [69][71]. Web application deployment vendors can wrap and adapt their middleware for use in WSRP-compliant services. Vendors of intermediary applications can enable their products for consuming Web Services for Remote Portals. Using WSRP, portals can easily integrate content and applications from many internal and external content providers. [53]

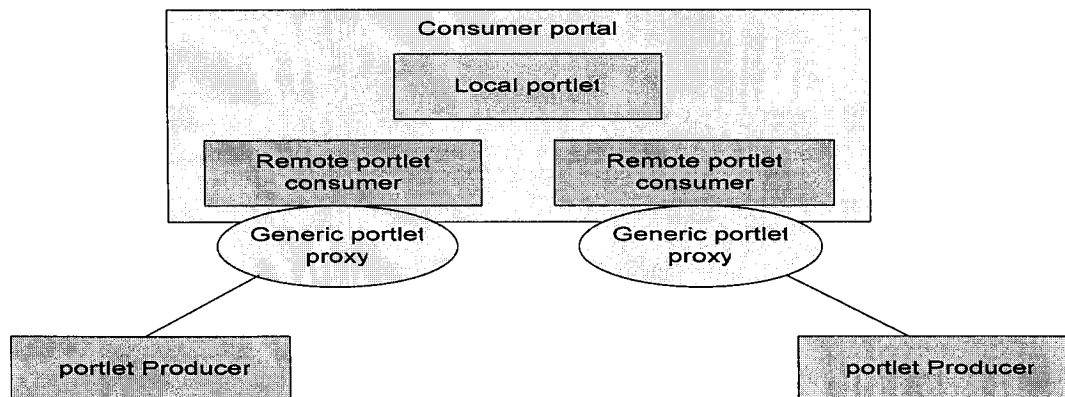


Figure 5.9: Web Service for Remote Poetlet [52][54]

To accomplish these goals, the WSRP standard defines a Web services interface description using WSDL. Firstly, we send a requirement of creating a remote portlet.

Secondly, Portlet Wizard starts looking for portlet producer with WSDL, for instance, <http://www.ubilearn.com/sampleportal/producer?WSDL>. Thirdly, the portlet producer will be registered in the consumer portal. Any available portlets will appear in the consumer portal. Lastly, a generic proxy is required to call portlet producer as local portlet.

WSRP introduces the notion of portlet producers and portlet consumers.

A portal is taken as WSRP Producers and/or WSRP consumers. By using WSRP, customers will be able to expose portlet applications in a portal as producers and aggregate application functionalities by integrating WSRP compliant portlets into the portal as a consumer.

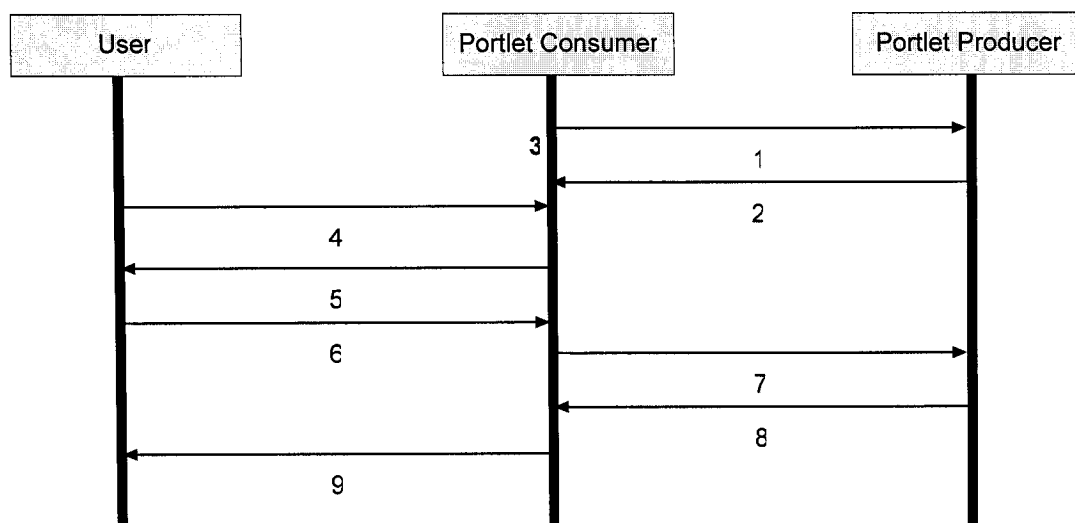


Figure 5.10: Interaction between consumer and producer [54]

End users will be able to interface with consumers to view the integrated applications.

Figure 5.10 shows the typical interactions between WSRP consumer and producer and here we give a short description:

- (1) Consumer discovers a Producer.
- (2) Consumer and Producer relationship is established.
- (3) Consumer learns all the capabilities of a Producer.
- (4) End user establishes a relationship with a Consumer.
- (5) Consumer aggregates pages, often with portlets, for users.
- (6) End user sends a page request to a Consumer.
- (7) Consumer requests information from Producer.

- (8) Producer provides Consumers with logic and data.
- (9) End user sees aggregated page.

5.5.2 Develop Web Service for Remote Portlet with Weblogic

Remote, or "proxy," portlets present content collected from WSRP-compliant producers. Remote portlets route requests from users to the appropriate Producers which, in turn process the request and send results back to the consumer. Consumers then aggregate the results coming from various producers and send the final result back to the user, who can view and use those results in remote portlets. Consumers have the ability to keep traffic separated and maintain all interactions private to that specific user during the interaction.

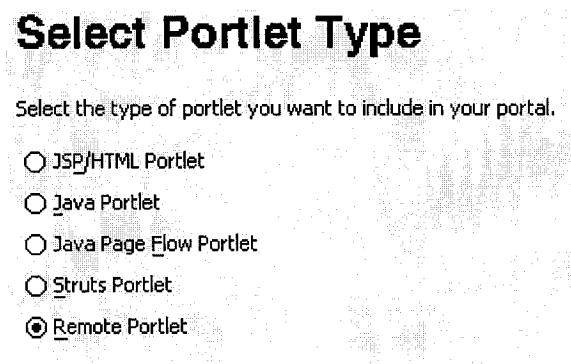


Figure 5.11: Portlet Type

Firstly, create a new portlet in the portal project which we have designed previously. Secondly, we select remote portlet as portlet type (as shown in figure) and a wizard window will pop up. Then give a name to this remote portlet and search producers for this remote portlet through WSDL. If we can find a producer that matches, we register this producer to our portal project. Otherwise, we can go back to searching another producer. Since we have found a producer, we can select a portlet as our remote portlet in the portlet lists from producers.

As with all portlets built with WeLogic Workshop, remote portlets can exist in one of three states: minimizable, maximizable, and deletable. We can select which of these states we want to include with the portlet by doing the following:

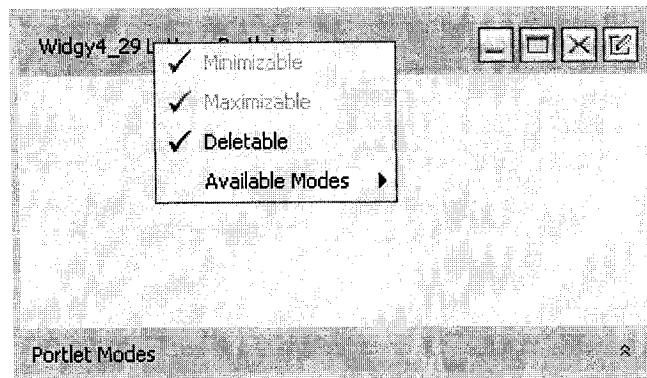


Figure 5.12: Portlet Modes (A)

Modifying Portlet Modes

When we want to customize a remote portlet, the following modes are provided (as shown in Figure 5.12 and Figure 5.13):

- Edit - Lets us specify a custom file that lets users modify the portlet's content when they click the Edit button.
- Help - Lets us specify a custom file that shows users help content for the portlet when they click the Help button.
- Float - Lets us display the portlet in a popup window when users click the Float button.

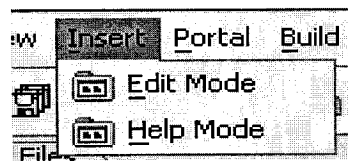


Figure 5.13: Portlet Modes (B)

5.6 System Service Implementation

5.6.1 The services of Instructor

The major portion of course management falls on teaching assistants and administrative staff. Ordinarily these tasks can be time consuming and error-prone, but Ubilearn helps simplify and automate these processes.

Managing course information. Administrator can allow instructor to create a new course based on the course requirement. The basic course information should be defined in the course content that includes course-type, course-level, course-name, course ID, credit, instructor ID and academic unit and so on. Prerequisite courses are available to allow to students to follow course-enroll rules to register a course based on what major they want to pursue. The other additional course information based on LOM is available to share the public information with the other e-learning system, for example language, document format, contribution date and so on. The detail course information is shown in Figure 5.14.

The screenshot shows a 'New Course' form with the following fields and values:

Field	Value
Course Type:	SYS
Course Level:	level-1
Course ID:	* (001 - 999)
Course Name:	*
Description:	
Academic Unit:	Computer Science
Credit:	*
Professor:	JSmit247
TA:	deyang
Prerequisite Course 1:	no course
Prerequisite Course 2:	no course
Prerequisite Course 3:	no course
Prerequisite Course 4:	no course
Language:	english
Keyword:	
Contribution Date:	
Document Format:	application/pdf
Learning Resource Type:	exercise
Intended End User Role:	teacher
Educational Context:	school
Rights Cost:	
Copyright Restrictions:	yes
Rights Description:	

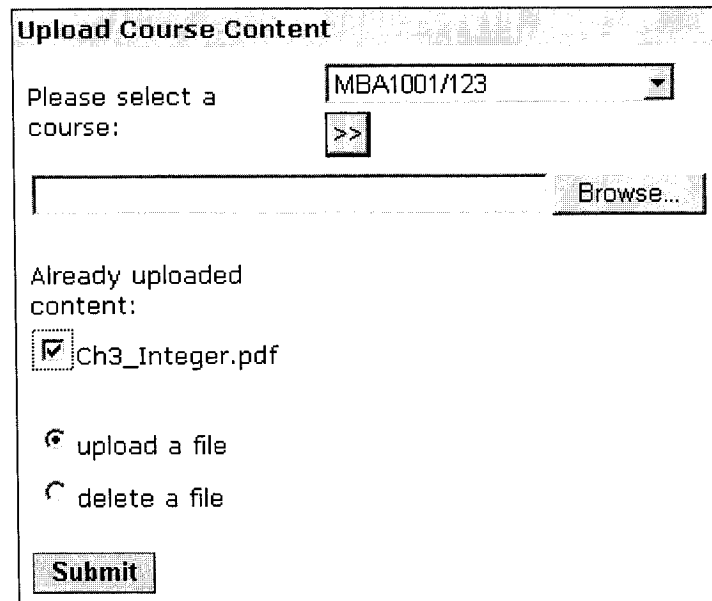
Submit

Figure 5.14: Course information

Course Content Upload

Instructors can upload a course content that include images, doc files or pdf files, after they have been created in the local computer. From course content upload portlet(shown in Figure 5.15), select a c course name in the course lists. The portlet will present what

content has been uploaded. Instructors can select an uploaded file and delete it. At the same time, they can click the button “brows” to choose a file to upload. After upload, a new file will be appeared in the course content list. Student can review and download course content in the student course view portlet.



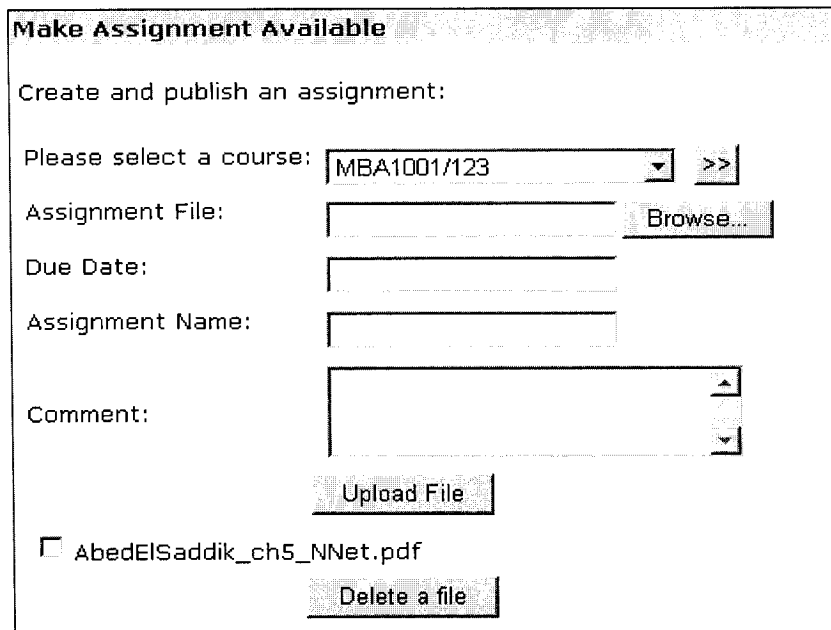
The screenshot shows a web form titled "Upload Course Content". At the top, there is a dropdown menu with "MBA1001/123" selected. Below it, the text "Please select a course:" is followed by a button with ">>". A text input field is present with a "Browse..." button to its right. Under the heading "Already uploaded content:", there is a list item "Ch3_Integer.pdf" with a checked checkbox. Below this list, there are two radio buttons: "upload a file" (which is selected) and "delete a file". At the bottom of the form is a "Submit" button.

Figure 5.15: Course information

Managing assignments. To create a new assignment, a web form is used to enter the relevant information(shown in figure), including (1) the assignment description and associated files, (2) the deadline for submission, (3) a short grace period during which submissions will not be considered late, (4) the assignment and source files, (5) the names of the files that the student should submit, and (6) whether re-grades are allowed, whether they are tracked by Ubilearn, and the deadline for a re-grade. This information can be changed at any time. Figure 5.16 present the detail assignment information.

To track the workflow involved in managing an assignment, assignments can be in one of two states: hidden from students (used during assignment creation) and shown to students and available for submission. Even though an assignment is closed to submission, a staff member can override this setting to allow an individual student or group of students to submit, and can even submit files on their behalf. This flexibility helps when a student has

a good reason to submit their assignment after the deadline or has submitted the assignment outside Ublearn.



The screenshot shows a web form titled "Make Assignment Available". The form contains the following elements:

- A heading "Make Assignment Available" with a small icon to its right.
- A sub-heading "Create and publish an assignment:".
- A "Please select a course:" label followed by a dropdown menu showing "MBA1001/123" and a ">>" button.
- An "Assignment File:" label followed by a text input field and a "Browse..." button.
- A "Due Date:" label followed by a text input field.
- An "Assignment Name:" label followed by a text input field.
- A "Comment:" label followed by a text area with up and down arrow buttons.
- An "Upload File" button.
- A checkbox labeled "AbedElSaddik_ch5_NNet.pdf".
- A "Delete a file" button.

Figure 5.16: Assignment Management (1)

Grading and re-grading. For each assignment, an instructor staff member assigns a set of students (or a set of groups, in case of group assignments) to each grader. Student or student groups are selected using check boxes and assigned to a grader selected using a pull-down menu. This functionality also provides an overview of the files submitted by each group and flags late submissions.

Each grader downloads the files for the students/groups assigned to that grader, grades the assignments, produces a feedback file for each one, and finally uploads the assignment grades and the feedback files (individually or all together). Online grading seems to promote more extensive, detailed feedback. Although a grader is assigned to each student/group, any staff person with the proper

5.6.2 The Services of Student

A student enrolled in more than one course using Ubilearn has a single portal—a single web page—that gives access to all the courses. Students can even access courses that they have taken in previous semesters, if the department allows it.

Managing course information. For each student, Ubilearn provides an overview page for each course in which the student is enrolled. Different instructors have different opinions on what information should be accessible to students, and should have some control over this. In Ubilearn, the staff can control when the grades on an assignment or exam become viewable, whether the students see the statistics for an assignment or exam, and when (if ever) the students can see their weighted total score or final grade.

Ubilearn also has many ways to notify students about important events. For example, the overview page shows upcoming assignments, due dates, and the time left on the next assignment. Students also receive notification from discussion forum about events such as availability of grades, feedback on an assignment, and actions on their re-grade requests. Thus, Ubilearn is a management tool for students as well as staff.

Students can be added to a course either from the university's files for the course, or individually. In both cases, only the university ID of the student is given, and all other information is retrieved from university system using an LDAP lookup. A student may decide to drop a course — in that case, the student can be removed. However, no information about the student is lost, and if the student decides to re-enroll, a click of the mouse button in UBilearn will do the trick.

Managing assignments. Ubilearn provides various ways for a student to manage course assignments (shown in Figure 5.17). First, Ubilearn provides the ability for students to obtain assignment write up and other needed files such as source code from the course overview page. Second, Ubilearn allows students to submit files as part of completing an assignment. Since students may make mistakes when submitting files, Ubilearn allows students to submit files any number of times before the assignment is due. Ubilearn keeps a history of all submitted files. When grades and feedback for an assignment become available, the student can view the grades and feedback for the assignment, along with

statistics about the overall class performance. If desired, re-grade requests can be submitted and will be automatically routed to the appropriate grader.

Please select a course:
SYS3007/system engineering

SYS3007/system engineering

Assignment Name: JSmit247_assignment_1

Due Date: 02-03-2005

Comment: pls submit by the due date

Assignment file:  JSmit247_README.pdf

Solution File:

Solution Status:  kevinwang_README.pdf

Figure 5.17 Assignment Management(2)

Managing the Quizzes. Instructor can build up some quiz templates based on course requirement. Some typical quiz templates include single-choice, multi-choice and Q&A. Quizzes are a special case of assignments for which students need to submit their online-quiz at the same time. Ublearn will mark students' quizzes automatically and send a grade to students. Content upload feature for grades is particularly useful for exams: a text file of student IDs and grades, compatible with Microsoft Excel, can be uploaded into an assignment or exam. Ublearn performs consistency checks before allowing bulk uploads: for example, validating student IDs.

5.6.3 The Relative Public Service

1. Calendar Service:

The My Calendar Service provides a full-featured calendar system to let users manage and configure appointments and reminders. The service also provides an edit mode to let users set calendar preferences and options.

Calendar Summary View. In a single portlet instance (shown in figure 5.18), clicking on the small calendar icon will bring up the calendar summary view. This view shows the

current month. Any day containing an appointment will appear in bold. Today's date is highlighted for easy viewing. From this view you can quickly add a new appointment by clicking Add appointment. Clicking on any date inside the portlet will maximize it and take you to the appropriate day or month.

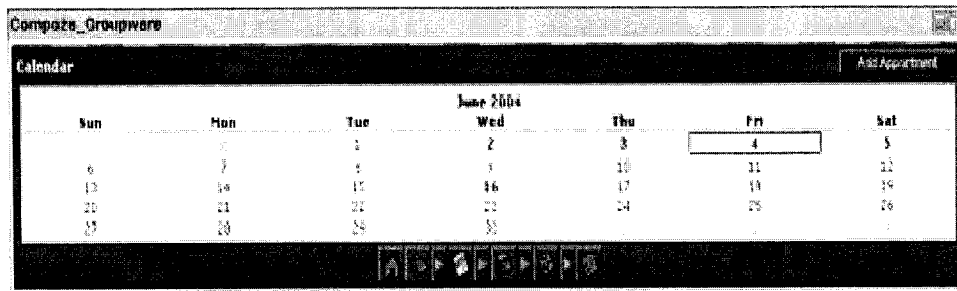


Figure 5.18 Calendar Summary View

Day View. The day view (shown in figure 5.19) lists all of your appointments for a given day. Pertinent details about these events include the type (meeting, indicated by a clock icon, or appointment, indicated by a two person icon), subject, time, location, importance, attachments and recurrence. All day appointments are shown at the top of the calendar page and indicated with a yellow bullet point to clearly distinguish them from timed appointments

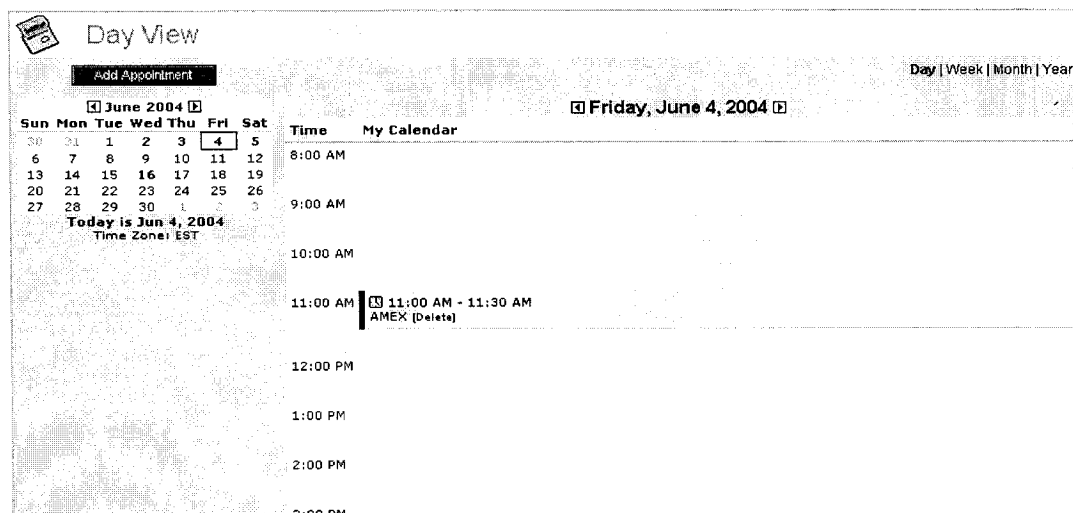


Figure 5.19 Day view

Adding an Appointment. To create a new appointment, click on a time inside the calendar

or click Add Appointment (shown in figure 5.20). Type in or select the appropriate information for the appointment including date, time and location. You can also show the time in your calendar as busy, as indicated by a dark blue bar, out of office, indicated by a purple bar, and tentative, indicated by a light blue bar.

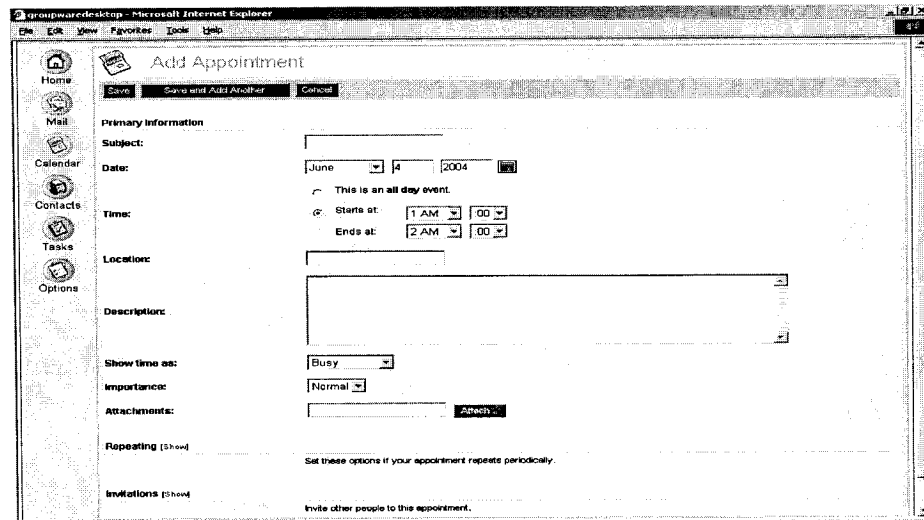


Figure 5.20 Adding an Appointment

2. Notebooks Service:

The Notebooks Service stores names, addresses, phone numbers, email addresses, and notes about personal contacts

Contacts Summary View. The Notebooks Service(shown in figure 5.21) allows users to manage names, addresses, phone numbers, email addresses, and other information in a personal address book. In the single portlet instance, clicking on the small Rolodex icon will bring up the contacts summary view. This page allows users to add new contacts and keyword search for contacts in users' address book by first name.

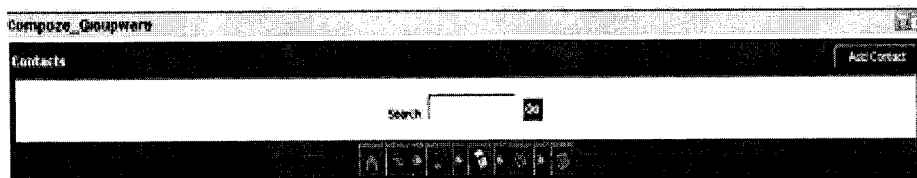


Figure 5.21 Contact Summary View

Finding Contacts. A limited number of contacts are displayed on each page(shown in figure 5.22). The users can modify the number of contacts to be shown per page in the

Options Portlet. To scroll through pages of contacts, click on the First, Previous, Next, or Last links. The total number of contacts in their address book is noted to the left of these links. To jump to a specific contact, type the number in the input box and click Go. For example, if users have 60 contacts in their address book, and type in the number 35, then users will jump to the thirty-fifth (35th) contact in the list.

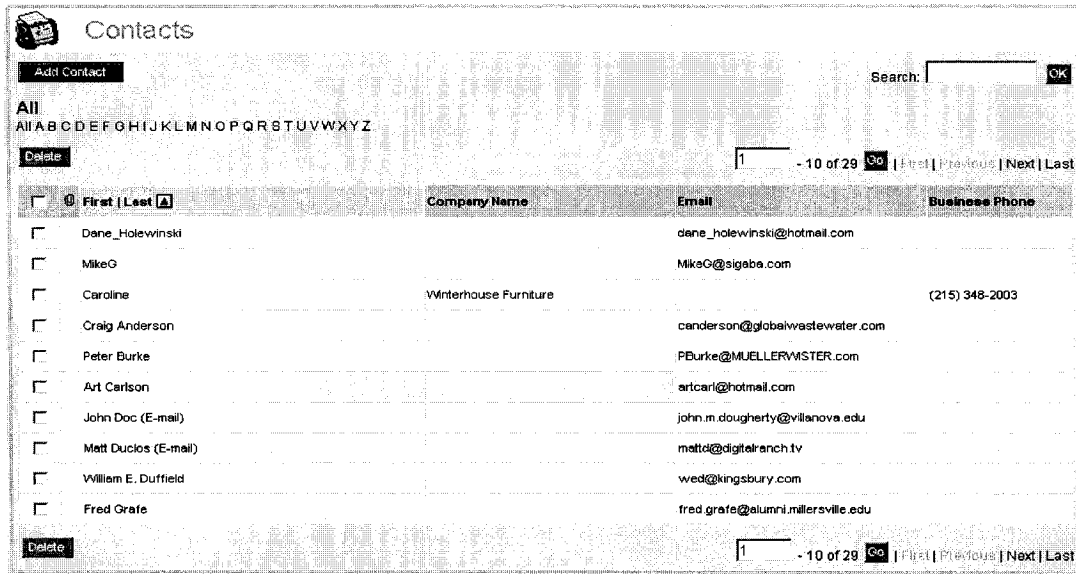


Figure 5.22 Finding Contact

Adding, Editing, and Deleting Contacts. Add a new contact to users' address book by clicking Add Contact (shown in figure 5.23). On this page users can enter detailed information about a person including name, home and business addresses, multiple phone numbers, and email addresses and even personal information such as spouse's name, anniversary, important dates, or other comments. Users can also attach files to a contact, such as a photograph or logo.

The screenshot shows a web browser window with the title 'Add Contact'. The browser's address bar shows a local file path. The page layout includes a sidebar on the left with icons for Home, Mail, Calendar, Contacts, Tasks, and Options. The main content area is titled 'Add Contact' and contains several sections for entering contact information. The 'Primary Information' section includes fields for Name (First, Middle, Last), Email, Display Name, Address, Nickname, and Phone Numbers (Business, Home, Mobile, Other). The 'Business Information' section includes fields for Company Name, Job Title, and Business Address (Street, City, State, Postal Code).

Figure 5.23 Adding, Deleting Contacts

3. News Service:

Administrators create some news with news service. In the news portlet, administrators input the news name, news link and who will read this news. Click the button “submit”, the news will be created. Administrator can setup the expired date of the news. After the expired date, the news will be disappeared automatically. Users with the different roles will access the different news in the news portlet.

Chapter 6

Testing and Measurement

Web services are very different to test compare to desktop software. Since Web Services have been applied to E-learning widely, the efficiency of service response is the most important characteristic for evaluating a Web service application. In this chapter, we discuss the methodology for testing the efficiency of service response of a specific Web service and analyze the results of the testing.

6.1 Testing Goal Analysis

The efficiency of a service response gives us an indication of a Web service's capability to serve clients under varying levels of load [73]. To measure the efficiency of service response, we run a Web service test program sending SOAP requests and measure its response time. Then we take this service as a client to call the second service for sending SOAP requests and measuring its response time as well. The rest may be deduced by analogy, the second service calls the third service and the third service calls the fourth service.

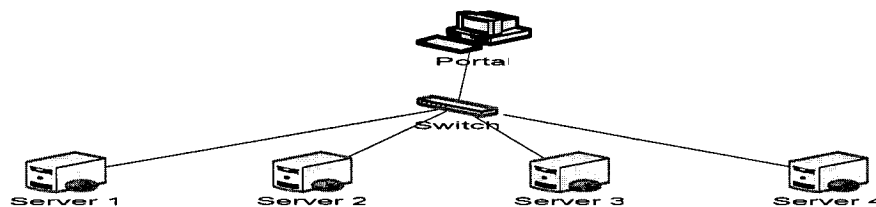


Figure 6.1: Network Configuration for Testing

Summarizing the measurements enables a software tester to predict the Web service's capability to serve users under load conditions.

We already implemented some Web services in UbiLearn project such as the “get course information” service that provides shared course information upon client requests.

To test this service, test agents that can send SOAP requests over and over again as needed.

For testing the “Register a course” functionality we used five machines hosting the following configurations.

6.2 Test Lab Environment

Machine	Server/device types	Hardware specification	Software specification
Client	Web Application Server	• Intel Pentium 4 CPU 2.20 GHz • 1G of RAM, 80GB HD • Intel PRO/100 VE Network Card	• Windows XP Server • Weblogic Application Server 8.1
Server1	Web Application Server	• Intel Pentium 4 CPU 2.20 GHz • 1G of RAM, 80GB HD • Intel PRO/100 VE Network Card	• Windows XP Server • Weblogic Application Server8.1
Server2	Web Application Server	• Intel Pentium 4 CPU 2.20 GHz • 1G of RAM, 80GB HD • Intel PRO/100 VE Network Card	• Windows XP Server • Weblogic Application Server8.
Server3	Web Application Server	• Intel Pentium 4 CPU 2.20 GHz • 1G of RAM, 80GB HD • Intel PRO/100 VE Network Card	• Windows XP Server • Weblogic Application Server8.1
Server4	Web Application Server & Database server	• Intel Pentium 4 CPU 2.20 GHz • 1G of RAM, 80GB HD • Intel PRO/100 VE Network Card	• Windows XP Server • Weblogic Application Server8.1 • Pointbase Database Server

Table 6.1: Test Environment

6.3 Test Methodology & Result

As a complex Web service based e-learning system, it is combined with some other e-learning system to provide the efficient and abundant e-learning services. So a web service, as a service provider, not only provides the response from the service client’ request, but as a service consumer sends other SOAP request to another web service. How to measure the response time from the Web Service Client’ request is an important issue to complex web service based e-learning system.

The following test case revolves around registering to a course. This functionality includes the four web services:

- Verify user information (Web service 1)
This service will check the learner's ID and make sure that he/she has the privileges to register to the course.
- Get course information(Web service 2)
This service will retrieve the course information that the learner wants to review. The course information includes course instructor, course credit, prerequisite courses and academic unit.
- Verify the prerequisite course information(Web service 3)
This service will check study history of the learner and verify whether the learner has finished the prerequisites of the course he/she wish to register to.
- Submit course registration information(Web service 4)
This service will submit course-registration information that includes registration day and learner ID.

Solution 1(as shown in Figure 6.2):

- The Registration portlet sends a request to verify user information to Web Service 1
- Web Service 1 receives this request, handles it and sends back a response to the Registration portlet.
- The Registration portlet receives this response from Web Service 1, handles this response and sends another request about getting course information to Web Services2.
- Web Service 2 receives this request, handles it and sends back a response to the registration portlet.
- The Registration portlet receives this response from Web Service 2, handles it and sends another request about verifying the prerequisite course information to Web Services3.
- Web Service 3 receives this request, handles it and sends back a response to the Registration portlet.

- The Registration portlet receives this response from Web Service 3, handles it and sends another request about submitting course registration information to Web Services4.
- Web Service 4 receives this request, handles it and sends back a response to the Registration portlet.
- The Registration portlet receives this response from Web service 4 and presents this response

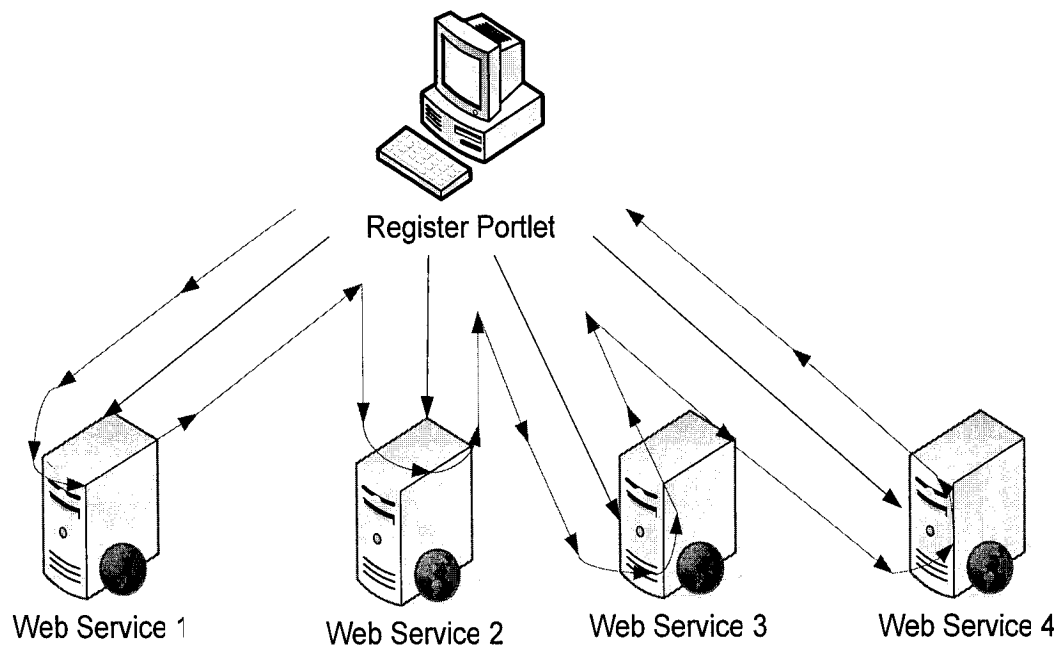


Figure 6.2: Solution 1: each service receives and responds to the main portlet.

Solution 2(as shown in Figure 6.3):

- The registration portlet sends a request about submitting course registration information to Web Service 4
- Web Service 4 receives this request and sends another request about verifying the prerequisite course information to Web Services3 according to registration portlet's requirements.
- Web Service 3 receives this request and sends another request about getting course information to Web Services2 according to Web Service 4's requirements.

- Web Service 2 receives this request and sends another request about verifying user information to Web Services1 according to Web Service 3's requirements.
- Web Service 1 receives this request, handles it and sends back a response to Web service 2
- Web Service 2 receives this response from Web service 1, handles it from receive Web service 3 and sends back a response to Web service 3
- Web Service 3 receives this response from Web service 2, handles it from receive Web service 4 and sends back a response to Web service 4
- Web Service 4 receives this response from Web service 3, handles it from receive register portlet and sends back a response to the registration portlet
- The registration portlet receives this response from Web service 4 and presents it

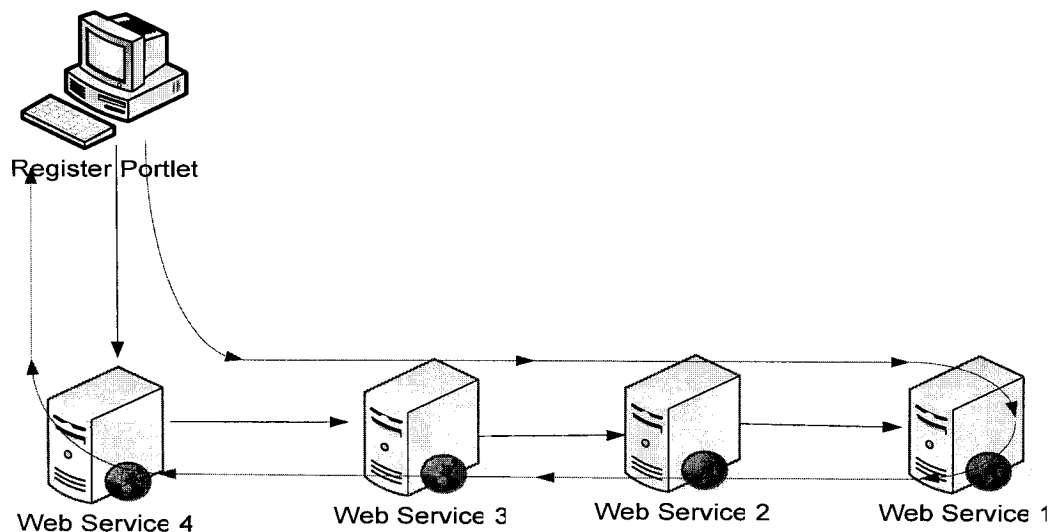


Figure 6.3 Solution 2 services receive and send response following the chain

The test data is shown in Table 6.3. The result is shown in milliseconds.

Register portlet	Web service 1	Web service 2	Web service 3	Web service 4	Total
Solution 1	272.64	299.62	329.14	375.07	1276.47
Solution 2	271.78	375.32	412.25	483.16	1542.51
Delay	N/A	75.7	83.11	108.09	266.04

Table 6.2 Test data from the intranet

Looking at the measurements and the calculated results, we can draw the following conclusions:

1. The solution2 provides practical scalability at the expense of delay.
2. It is very interesting to note that most of the delay is caused primarily by processing time at the intermediate Web services, rather than network propagation time in terms of message transmission.
3. The control overhead introduced by the solution 2 is 70-100 msec more per extra node on the intranet on the average. This is quite acceptable in the context of a man-machine conversation e-learning system, since receiving messages with a certain consistent delay does not detriment users' experience significantly (think of reviewing course information , with a couple of seconds delay: the user does not notice the delay due to lack of a reference point).
4. The control overhead is introduced by the refresh message exchanges. It is influenced by the network load as well as the deployment configuration.
5. The Solution 2 also has an impact on reliability. When extra web service on the intranet is added, that service will load a service proxy for calling the other services. Most of the delay is caused primarily by processing web service proxy.
6. In solution 1, Register portlet just loads web service proxy one time before it starts calling web service. It saves process time to load Web service proxy for other web services.
7. In Solution 2, Register portlet has light source code loaded. Web services help it handle business logic. It is very useful to share portlet user interface through Web service for Remote Portlet.

Chapter 7

Conclusions and Future Research

7.1 Conclusion

In this thesis we have presented a Web services based framework that facilitates distributed e-learning portal systems by providing a comprehensive platform in which all the shareable Learning objects and contents are published, described, located and invoked in a standardized way. Based on this framework a distributed e-learning system prototype called UbiLearn was analyzed and designed in a new and original way.

The prototype was implemented by some powerful Java technologies provided by the J2EE infrastructure like JAXP, JAXM, JAX-RPC and JAXR in a extendible and compatible developing and deploying environment. Moreover, the scalability and performance test measures how well the Web service performs under different loading levels. The next generation of distributed e-learning systems requires a sharable place that enables them to launch learning services from different vendors to a common accessible repository and to exchange data between learning applications regardless on what platforms. The Web services and portal technologies fulfill these requirements perfectly.

The Web services based portal framework UbiLearn enables systematic application-to-application interactions over the Web, also, a smooth integration of existing infrastructure into the model. The definition of interoperable interfaces and standard communication protocols between application and application are the keys of Web services infrastructure. That enhances the cross platform development in a lightweight data exchange level. Using the tools described in the thesis, the working prototype has been developed to prove that this kind of Web services based e-learning portal systems can be

implemented and different learning services can be shared in a feasible mechanism. Many learning services (namely User register, register course, assignment management, quiz management, grade management) are developed in the system prototype, more learning Web services can be designed and implemented by adapting the idea of those services. And the testing methodology used in this thesis can be applied to any Web service even with more complicated network configurations. Also the insights resulting from this thesis can be extended to other types of distributed systems like e-business and e-commerce. Lots of distributed applications have been powered by the Web services and portal technology, but the Web services standards are still undergoing and continuing development pushed by several organizations like W3C, OASIS, IBM, Microsoft, Sun, etc.

7.2 Future Research

The proposed framework has been formulated in the thesis for future research or implementation to be based on. The following research items need further discussion:

- Development of specifications for launching shareable learning object, aggregating and delivering learning resource by using the Web services infrastructure.
- WSRP security and access control (authorization & authentication) for learning services publication and invocation.
- Quality of Services (QoS) is another issue should be focused on related with Web security. Scalability and performance improvement for Web services based learning applications using SOAP or XML marshalling in the heavy loading conditions.
- Different preferences of accessing learning Web services with different end devices like wired and wireless devices.
- Synchronous and asynchronous collaborative tools for learners and instructors communicate over networks.

References

- [1] F. H. Sondgeroth, thesis of Doctor of Philosophy, "Designing a systemic e-learning delivery system", Saybrook Graduate school and Research Center, San Francisco, California, November, 2003.
- [2] G. M. Piskurich, "what is e-learning", The AMA handbook of E-learning, 2003 AMACOM, a division of American Management Association, Page 15-23.
- [3] B. Ellet, A. Naiman, "Is E-learning better than...?", The AMA handbook of E-learning, 2003 AMACOM, a division of American Management Association, Page 35-42.
- [4] P. Brusilovsky, "Adaptive e-learning systems: KnowledgeTree: a distributed architecture for adaptive e-learning", International World Wide Web Conference archive, Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, New York, NY, USA SESSION: Adaptive e-learning systems, Pages: 104 - 113
- [5] K. Kim, C.J. Bonk and T. Zeng, "Surveying the future of workplace e-learning: the rise of blending, interactivity, and authentic learning", eLearn, Volume 2005 , Issue 6 (June 2005),SECTION: Research paper, Page: 2
- [6] "Sun Java System Portal Server 6 2004Q2 Deployment Planning Guide" available at <http://docs.sun.com/source/817-5321/1-portal.html>, last visited on Jun.16, 2005.
- [7] M.A. Smith, " Portals: toward an application framework for interoperability " Communications of the ACM Volume 47 , Issue 10 (October 2004), Voting systems, Pages: 93 - 97
- [8] "Web Services Architecture Working Group of World Wide Web Consortium (W3C)", available at <http://www.w3.org/2002/ws/arch/>, last visited on Jun.16, 2005.

- [9] "web services concepts — a technical overview", available at http://www.hpmiddleware.com/downloads/pdf/web_services_tech_overview.pdf, last visited on Jun.16, 2005.
- [10] M. Fred and S. Lindesmith Sta Software Engineer, IBM "How to decide if portlets are your best option", available at <http://www-128.ibm.com/developerworks/ibm/library/i-portletintro/>, last visited on Jun.16, 2005.
- [11] "IBM to sell its internal portal technologies to customers", available at <http://www.computerworld.com/news/special/pages/story/0,5364,1865-69690,00.html>, last visited on Jun.16, 2005.
- [12] "BEA WebLogic Workshop Online Help" available at <http://edocs.bea.com/workshop/docs81/doc/en/core/index.html>, last visited on Jun.16, 2005.
- [13] W. Luo and M. Najdawi "Trust-building measures: a review of consumer health portals", Communications of the ACM, Volume 47 , Issue 1 (January 2004) Multimodal interfaces that flex, adapt, and persist, Pages: 108 - 113
- [14] "BEA WebLogic Portal Product Overview" available at <http://www.bea.com/framework.jsp?CNT=overview.htm&FP=/content/products/weblogic/portal>, last visited on Jun.16, 2005.
- [15] "Introduction to XML", available at http://www.w3schools.com/xml/xml_what.asp, last visited on Jun.16, 2005.
- [16] R. Mogha,V.V. Preetham "Java Web Services Programming" Publish by Wiley Publishing,Inc., August 26, 2002
- [17] "the introduction of SOAP", available at <http://www.perfectxml.com/soap.asp>, last visited on Jun.16, 2005.
- [18] SOAP Version 1.2 Part 1: Messaging Framework by W3C Candidate Recommendation available at <http://www.w3.org/TR/soap12-part1/> , last visited on Jun.16, 2005.

- [19] E. Christensen, F. Curbera, Meredith and S. Weerawarana "Web Services Description Language (WSDL) 1.1" available at <http://www.w3.org/TR/wsdl>, last visited on Jun.16, 2005.
- [20] "Web Services Description Language (WSDL)", available at http://www.servicearchitecture.com/web-services/articles/web_services_description_language_wsdl.html, last visited on Jun.16, 2005.
- [21] "the introduction of UDDI", available at <http://uddi.microsoft.com/about/FAQbasics.aspx>, last visited on Jun.16, 2005.
- [22] "Universal Description Discovery Integration (UDDI) standard", available at <http://www.webservicesmall.com/profile.asp?pid=105>, last visited on Jun.16, 2005.
- [23] "Technology Report on WSDL", available at <http://xml.coverpages.org/wsdl.html>, last visited on Jun.16, 2005.
- [24] "Web Services Tool", available at http://search.pramati.com/docstore/1230004/jh/docs/files/web_service.html, last visited on Jun.16, 2005.
- [25] "Advantages of Web Services", available at <http://www.codeworks.net/content/80000/webservices/guide/AdvantagesofWebServices.asp>, last visited on Jun.16, 2005.
- [26] W. Hoschek, "The Web Service Discovery Architecture", Conference on High Performance Networking and Computing Proceedings of the 2002 ACM/IEEE conference on Supercomputing, Baltimore, Maryland, Pages: 1 - 15
- [27] L. Chen, M. Li, "Using Web Services in TMN environment", ACM International Conference Proceeding Series; Vol. 87 Proceedings of the IEEE EEE05 international workshop on Business services networks, Hong Kong, Pages: 6 - 6

- [28] "E-learning Application Infrastructure" available at <http://newsletter.paragon-systems.com/articles/53/5/opt-edu/7291>, last visited on Jun.16, 2005.
- [29] Hejazi and M. Dilmaghani, "A Preliminary Study on the Implementation of Campus Portals in Iranian Higher Education Communities", IEEE Computer Society, Technical Committee on Learning Technology (LTTC), Volume 7 Issue 1
- [30] "Draft Standard for Learning Object Metadata" Learning Technology Standards Committee of the IEEE IEEE 1484.12.1-2002
- [31] "LOM Overview", available at <http://www.cen-ltso.net/Users/main.aspx?put=211>, last visited on Jun.16, 2005.
- [32] Advanced Distributed Learning (ADL), available at <http://www.adlnet.org>, last visited on Jun.16, 2005.
- [33] Sharable Content Object Reference Model (SCORM) 1.2 Overview by Advanced Distributed Learning (ADL), available at http://www.adlnet.org/ADLDOCS/Documents/SCORM_1.2_Overview.pdf, last visited on Jun.16, 2005.
- [34] IMS Global Learning Consortium, Inc., available at <http://www.imsglobal.org/>, last visited on Jun.16, 2005.
- [35] Aviation Industry CBT (Computer-Based Training) Committee (AICC), available at <http://www.aicc.org>, last visited on Jun.16, 2005.
- [36] Alliance of Remote Instructional Authoring & Distribution Networks for Europe (ARIADNE) available at <http://www.ariadne-eu.org/>, last visited on Jun.16, 2005.
- [37] Institute of Electrical and Electronics Engineers (IEEE) Learning Technology Standards Committee (LTSC) available at <http://ltsc.ieee.org/>, last visited on Jun.16, 2005.
- [38] Sharable Content Object Reference Model (SCORM) 1.2 Content Aggregation Model by Advanced Distributed Learning (ADL), available at

- http://www.adlnet.org/ADLDOCS/Documents/SCORM_1.2_CAM.pdf , last visited on Jun.16, 2005.
- [39] IEEE LTSC Learning Object Metadata (LOM) available at <http://ltsc.ieee.org/wg12/>, last visited on Jun.16, 2005.
- [40] Sharable Content Object Reference Model (SCORM) 1.2 Run-Time Environment by Advanced Distributed Learning (ADL) available at http://www.adlnet.org/ADLDOCS/Documents/SCORM_1.2_RunTimeEnv.pdf, last visited on Jun.16, 2005.
- [41] "Randall House Associates" available at <http://www.rhassociates.com/scorm.htm>, last visited on Jun.16, 2005.
- [42] "IBM WebSphere Portal V4.1 Handbook Volume 1", SG24-6883-00, International Technical Support Organization, January 2003
- [43] Advanced Distributed Learning (ADL), available at <http://www.adlnet.org>, last visited on Jun.16, 2005
- [44] Alan Dennis, Barbara Haley Wixom, David Tegarden. "Systems Analysis and Design", ISBN: 0-471-41387-9, John Wiley & Sons, Inc. 2002.
- [45] Unified Modeling Language (UML) resource page, available at <http://www.omg.org/uml/>, last visited on Jun.16, 2005
- [46] Leszek A. Maciaszek "Requirements Analysis and System Design – Developing Information Systems with UML", ISBN: 0-201-70944-9, Addison – Wesley, 2001.
- [47] T. Jaeger and A. Prakash, "Requirements of role-based access control for collaborative systems", Symposium on Access Control Models and Technologies Proceedings of the first ACM Workshop on Role-based access control, Gaithersburg, Maryland, United States, Article No. 16
- [48] "Introduction to i18n" available at <http://www.debian.org/doc/manuals/intro-i18n/index.en.html#contents>, last visited on Jun.16, 2005

- [49] Y. Tykhomyrov, "Introduction to Internationalization Programming", available at <http://www.suodenjoki.dk/us/productions/articles/localization.htm>, last visited on Jun.16, 2005

- [50] Kai Wang, Jianming Ke and Abdulmotaleb El Saddik. "Architecture for Personalized Collaborative E-learning Environment". In Proceedings of World Conference on Educational multimedia, Hypermedia & Telecommunications (ED-MEDIA 2005) Montreal, Quebec, Canada, June 27- July 2, 2005.

- [51] "Internationalization and Localization JSP Tags" available at <http://e-docs.bea.com/workshop/docs81/doc/en/core/index.html>, last visited on Jun.16, 2005.

- [52] "Web Services for Remote Portlets Specification", Approved as an OASIS Standard August 2003, available at <http://www.oasis-open.org/committees/wsrp>, last visited on Jun.16, 2005.

- [53] Bryan Castle, "Introduction to Web Services for Remote Portlets -Use WSRP in a Service-Oriented Architecture", available at <http://www-128.ibm.com/developerworks/webservices/library/ws-wsrp/index.html>, last visited on Jun.16, 2005.

- [54] Alex Toussaint, "Using Web Services for Remote Portlets (WSRP) with WebLogic Portal 8.1 SP3", available at <http://dev2dev.bea.com/pub/a/2004/11/wsrp.html> last visited on Jun.16, 2005.

- [55] F.I.Tantono and C. E.Siong, "A Web Based Framework for E-Learning: A Model for Online Presentation Authoring", ED-MEDIA 2004 - World Conference on Educational Multimedia, Hypermedia and Telecommunications, 6/21/04 - 6/26/04

- [56] T. Nodenot, C. Marquesuzaa, P.Laforcade and C. Sallaberry, "Model based Engineering of Learning Situationsfor Adaptive Web Based Educational Systems", International World Wide Web Conference, Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, New York, NY, USA,SESSION: Adaptive e-learning systems, Pages: 94 - 103

- [57] P. Dolog, N. Henze, W. Nejdl and M. Sintek, "Personalization in Distributed eLearning Environments", International World Wide Web Conference, Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, New York, NY, USA, SESSION: Student tracking and personalization, Pages: 170 - 179
- [58] D. Zhang, J. L. Zhao, L. Zhou, J. F. Nunamaker, "Can e-learning replace classroom learning?", Communications of the ACM, Volume 47, Issue 5 (May 2004), New architectures for financial services, Pages: 75 - 79
- [59] S. Marshall and G. Mitchell, "Applying SPICE to e-learning: an e-learning maturity model?", ACM International Conference Proceeding Series; Vol. 57 Proceedings of the sixth conference on Australian computing education - Volume 30, Dunedin, New Zealand, Pages: 185 - 191
- [60] K. M. Kapp, "E-learning basics: essay: The e-Learning Market: It's About the Learner, Not the Instructor!", eLearn, Volume 2003, Issue 6 (June 2003), COLUMN: E-learning basics: essay, Page: 1
- [61] L. Minton, R. Boyle, V. Dimitrova, "If diversity is a problem could e-learning be part of the solution?: a case study", Annual Joint Conference Integrating Technology into Computer Science Education, Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education, Leeds, United Kingdom, SESSION: Online and electronic learning, Pages: 42 - 46
- [62] W. H. Graves, "The new challenges of E-learning", Ubiquity archive Volume 1, Issue 43 (January 2001), Article No. 2
- [63] G. Singh, S. Bharathi, A. Chervenak, E. Deelman, C. Kesselman, M. Manohar, S. Patil, L. Pearlman, "A Metadata Catalog Service for Data Intensive Applications", Conference on High Performance Networking and Computing archive Proceedings of the 2003 ACM/IEEE conference on Supercomputing, Page: 33
- [64] H. Schuster, J. Neeb and R. Schamburger, "A configuration management approach for large workflow management systems", International Conference on Work

- activities Coordination and Collaboration,
 Proceedings of the international joint conference on Work activities coordination and
 collaboration, San Francisco, California, United States, Pages: 177 - 186
- [65] N. Hoft "reparing for the inevitable: localizing computer documentation", ACM
 Special Interest Group for Design of Communications
 Proceedings of the 9th annual international conference on Systems documentation,
 Chicago, Illinois, United States, Pages: 37 - 43
- [66] A. Kern, M. Kuhlmann, A. Schaad, J. Moffett, "Role Engineering: Observations on
 the role life-cycle in the context of enterprise security management", Symposium on
 Access Control Models and Technologies archive
 Proceedings of the seventh ACM symposium on Access control models and
 technologies Monterey, California, USA, SESSION: Role Engineering, Pages: 43 -
 51
- [67] S. R. Schach. "An introduction to object-oriented systems analysis and design with
 UML and the unified process", Publisher Boston, Mass., McGraw-Hill/Irwin, c2004.
- [68] D. Simões, R. Luís, N. Horta "Posters: Enhancing the SCORM metadata model",
 International World Wide Web Conference,
 Proceedings of the 13th international World Wide Web conference on Alternate
 track papers & posters, New York, NY, USA, POSTER SESSION: Posters, Pages:
 238 - 239
- [69] "Administering the Web Services for Remote Portlets (WSRP) Service", Sun Java
 System Portal Server 6 2005Q1 Administration Guide.
- [70] O. Díaz, J. Iturrioz, A. Irastorza, "Web engineering with semantic annotation:
 Improving portlet interoperability through deep annotation", International World
 Wide Web Conference, Proceedings of the 14th international conference on World
 Wide Web, in Chiba, Japan SESSION: Web engineering with semantic annotation,
 Pages: 372 - 381
- [71] F. Bellas, D. Fernández, A. Muiño, "Web site engineering: A flexible framework for
 engineering 'my' portals", International World Wide Web Conference,

Proceedings of the 13th international conference on World Wide Web New York, NY, USA, SESSION: Web site engineering Pages: 234 - 243

- [72] Douglass, Bruce Powel, "Real time UML: advances in the UML for real-time systems", Publisher Boston: Addison-Wesley, 2004. ISBN 0-321-16076-2
- [73] S. Joines, R. Willenborg, K. Hygh, "Performance analysis for Java WEB SITES", Publisher Boston: Addison-Wesley, 2003. ISBN 0-201-84454-0
- [74] A.T.Manes, "Web Services: A Manager's Guide", Publisher Boston: Addison-Wesley, 2003. ISBN 0-321-18577-3
- [75] S.Bennett, J.Skelton, K.Lunn, "Schaum's outlines, UML", Design and patents Act, McGraw-Hill International (UK), 1988. ISBN 0-07-709673-8
- [76] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, R. Chandramouli "Proposed NIST standard for role-based access control" ACM Transactions on Information and System Security (TISSEC), Volume 4 , Issue 3, pages: 224 - 274
- [77] J. Bacon, K. Moody, W. Yao, "A model of OASIS role-based access control and its support for active security", ACM Transactions on Information and System Security (TISSEC), Volume 5, Issue4, Pages 492 - 540
- [78] J. S. Park and R. Sandhu, "Role-based access control on the web", ACM Press Transactions on Information and System Security (TISSEC) New York, NY, USA, Volume 4, Issue1, Pages 37-71
- [79] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. Scientific American, May 2001
- [80] IMS Global Learning Consortium Inc. IMS Question and Test Interoperability Specification, V1.2. Available at: <http://www.imsglobal.org/question/index.cfm>.