Towards the Standard-oriented E-Learning: An RDF-based E-Learning Content Management P2P Infrastructure and Some Enabling Technologies

Vom Fachbereich Informatik
der Universität Hannover
zur Erlangung des akademischen Grades
Doktor-Ingenieurin /Doktor-Ingenieur

genehmigte Dissertation
von

M.Sc. CHANGTAO QU
geboren am 24. Mai 1968 in Jilin, V.R. China

2004

Referent: Prof. Dr. techn. Dipl.-Ing. Wolfgang Nejdl
Korreferent: Prof. Dr.-Ing. Klaus Jobmann

Tag der Promotion: 20. August 2004

Abstract

The E-Learning standardization is a key technology to liberate learning content from local implementations to enable some high-level "abilities" of E-Learning such as accessibility, interoperability, durability and reusability. In this thesis we focus on standardization of the E-Learning content management, the most important component of a generic E-Learning architecture, and propose the standard-oriented E-Learning content management, which can implement two major functionalities of LCMSs: the learning content discovery and learning content delivery based on sets of E-Learning standards and specifications. With regard to the standard-oriented learning content discovery, we propose Edutella, an RDF-based E-Learning content management P2P infrastructure, to connect heterogeneous educational repositories and further enable efficient learning content discovery across various back-end systems and learning resource metadata sets. Whereas P2P provides Edutella with essential support to cross heterogeneity of educational repositories, RDF provides Edutella with basic abilities to describe distributed learning content, to represent and mediate distributed P2P functionalities, as well as to discover learning content across various learning resource metadata sets. In order to address diversity of the current E-Learning regarding infrastructures and content formats applied, we further propose two extensions to the main framework of Edutella.

- First, we investigate an approach for interacting Edutella/JXTA with Web services with the purpose of exchanging distributed functionalities between the two platforms and further constructing a unified distributed computing architecture converging P2P and Web services for E-Learning.
- Second, we explore an approach for integrating XML metadata repositories into RDF-based Edutella with the purpose of managing XML and RDF metadata in a consistent manner in Edutella.

By means of these two extensions, Edutella can ensure a relatively widespread searching scope of the standard-oriented learning content discovery, being able to cover several major distributed computing paradigms such as the client-server architecture, P2P, Web services, OGSA/Grid, etc., and two principal metadata formats: XML and RDF.

With regard to the standard-oriented learning content delivery, we propose a reference model being able to standardize two major operations of a learning content delivery process: the learning content organization and run-time environment implementation based on sets of E-Learning standards and specifications. We further develop sets of enabling technologies to implement the reference model based on standard Web technologies and metadata technologies. Going a step further, we apply the reference model to conventional AEHSs and investigate an approach for converging the E-Learning standardization and conventional AEHS research to realize the standard-oriented AEHS that can ensure reusability and portability of learning content while achieving adaptability of the learning content delivery.

Taking an engineering approach, the research in this thesis is focused on constructing a testbed for the standard-oriented E-Learning, starting from its most critical component: the standard-oriented E-Learning content management. This testbed is not only purposed to accommodate most of "best practice" of the current E-Learning standardization but also provide the guidance for its future development. We believe that such a testbed can promote the development and distribution of standardized learning content, based on which the standard-oriented E-Learning can start on a journey leading to its final success.

Zusammenfassung

Die Standardisierung im E-Learning ist der Schlüssel für die verteilte Nutzung lokal vorhandener, elektronischer Lerninhalte und für die Erzielung der häufig zitierten Potentiale von E-Learning wie Verfügbarkeit, Interoperabilität, Nachhaltigkeit und Wiederverwendbarkeit. Diese Dissertation fokussiert auf Standardisierung für das E-Learning-Content-Management, das die wichtigste Komponente einer generischen E-Learning-Architektur dar stellt. Es wird ein Standard-orientiertes Content-Management beschrieben, das die beiden Kernfunktionalitäten von LCMS implementiert: das Auffinden und die Bereitstellung (oder Zugriff) von elektronischen Lerninhalten unter Berücksichtigung existierender E-Learning-Standards und Spezifikationen. Im Hinblick auf das Standard-orientierte Auffinden von elektronischen Lehrinhalte wird Edutella vorgestellt, eine RDF-basierte P2P-Infrastruktur für das E-Learning-Content-Management. Edutella erlaubt die Verknüpfung heterogener Inhaltssammlungen (Repositories) und weiterhin das effiziente Auffinden elektronischer Lehrinhalte über verschiedene Plattformen (Back-End-Systeme) und verschiedene Metadaten-Sätze. Während eine P2P-Architektur in Edutella die Verknüpfung heterogener Repositories erlaubt, ermöglicht RDF grundlegende Fähigkeiten zur Beschreibung verteilter, elektronischer Lerninhalte, zur Darstellung und Vermittlung von P2P-Funktionalitäten sowie zur Auffindung elektronischer Lerninhalte über unterschiedliche Metadaten-Sätze. Um die Vielfältigkeit von E-Learning-Infrastrukturen und Inhaltsformaten zu adressieren, werden zwei Erweiterungen für die Edutella Rahmenstruktur vorgeschlagen.

- Zum einen wird die Verbindung von Edutella/JXTA mit Web-Diensten untersucht mit den Zielen, verteilte Funktionalitäten zwischen unterschiedlichen Plattformen auszutauschen, sowie eine einheitliche, verteilte Rechnerarchitektur, die P2P und Web-Dienste miteinander verknüpft, zu konstruieren.
- Zum anderen wird der Ansatz erforscht, XML-Metadaten-Repositories in die RDF-basierte Edutella-Architektur zu integrieren. Dabei sollen XML- und RDF-Metadaten konsistent verwaltet werden.

Mit diesen beiden Erweiterungen kann mit Edutella eine relativ weit ausgedehnte Suche zum Standard-orientierten Auffinden von Lerninhalten erzielt werden. Verschiedene, verteilte Rechner-Architekturen wie Client-Server-Architekturen, P2P, Web services, OGSA/Grid, usw., sowie die beiden grundlegenden Metadaten-Formate XML und RDF werden unterstützt.

Im Hinblick auf den Standard-orientierten Zugriff auf elektronische Lerninhalte wird ein Referenzmodell vorgestellt, an dem zwei wesentliche Vorgänge im Bereitstellungsprozess standardisiert werden können: die Lerninhalts-Strukturierung und die Implementierung der Laufzeitumgebung unter Berücksichtigung vorhandener E-Learning-Standards und Spezifikationen. Verschiedene Basis-Technologien werden weiter entwickelt, um das auf Web- und Metadaten-Technologien basierende Referenzmodell zu implementieren. Davon ausgehend wird zunächst das Referenzmodell auf konventionelle AEHS angewendet. Anschließend wird versucht, E-Learning-Standardisierung und die Forschung für konventionelle AEHS zu konvergieren, um eine Standard-orientierte AEHS zu realisieren, die in der Lage ist, Wiederverwendbarkeit und Übertragbarkeit von Lerninhalten bei gleichzeitiger Adaptierbarkeit des Zugriffs auf die Inhalte zu erreichen.

Durch den verfolgten ingenieurwissenschaftlichen Ansatz ist die Forschung in dieser Dissertation an der Konstruktion eines Testumfeldes für Standard-orientiertes E-Learning orientiert, beginnend an der kritischsten Komponente: das Standard-orientierte Content-Management. Zielvorstellungen dieser Testumgebung waren die Berücksichtigung vieler „Best Practice“ - Beispiele aus dem Bereich E-Learning-Standardisierung sowie eine Orientierung für weitere mögliche Entwicklungen zu geben. Eine derartige Testumgebung kann die Entwicklung und Verteilung elektronischer Lerninhalte fördern, mit Hilfe derer sich Standard-orientiertes E-Learning erfolgreich entwickeln kann.

Keywords

English: E-Learning Standardization, Standard-oriented E-Learning Content Management, Peer-to-Peer
Deutsch: Standardisierung im E-Learning, Standard-orientiertes E-Learning-Content-Management, Peer-to-Peer

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACM	Association for Computing Machinery
ACM CCS	ACM Computing Classification System
ADL	Advanced Distributed Learning
AEHS	Adaptive Educational Hypermedia System
AGR	AICC Guidelines & Recommendations
AH	Adaptive Hypermedia
AICC	Aviation Industry CBT Committee
ALIC	Advanced Learning Infrastructure Consortium
API	Application Program Interface
ARIADNE	Alliance of Remote Instructional Authoring & Distribution Networks for Europe
ARIADNE KPS	ARIADNE Knowledge Pool System
CanCore	Canadian Core Learning Resource Metadata Application Profile
CBT	Computing Based Training
CEN/ISSS	European Committee for Standardization/Information Society Standardization System
CEN/ISSS LTWS	CEN/ISSS Learning Technology Workshop
CLDC	Connected Limited Device Configuration
CMI	Computer Managed Instruction
CORBA	Common Object Request Broker Architecture
CWA	CEN Workshop Agreement
DARPA	Defense Advanced Research Projects Agency
DAML	DARPA Agent Markup Language
DAML + OIL	DAML + Ontology Inference Layer
DAML-S	DAML Services
DCMI	Dublin Core Metadata Initiative
DCMI-Ed	DCMI Education Working Group
DCMES	Dublin Core Metadata Element Set
DNS	Domain Name System
DCOM	Distributed Component Object Model
DOM	Document Object Model
DTD	Document Type Definition
ECDM	Edutella Common Data Model
EJB	Enterprise JavaBeans
EU/IST	The European Union/Information Society Technologies
FLWOR	For-Let-Where-Order by-Return
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IEEE API	IEEE P1484.11.2 ECMAScript API for Content to Runtime Services Communication
IEEE Data Model	IEEE P1484.11.1 Data Model for Content Object Communication
IEEE LOM	IEEE Learning Object Metadata Standard
IEEE LTSA	IEEE Standard for Learning Technology Systems Architecture
IEEE LTSC	IEEE Learning Technologies Standards Committee
IEEE-SA	IEEE - Standards Association
IMS	Instructional Management System

IMS CP	IMS Content Packaging Specification
IMS DRI	IMS Digital Repositories Interoperability Specification
IMS LIP	IMS Learner Information Package Specification
IMS MD	IMS Learning Resource Metadata Specification
IMS QTI	IMS Question & Test Interoperability Specification
IMS QTI ASI	IMS QTI Assessment, Section and Item
IMS RDCEO	IMS Reusable Definition of Competency or Educational Objective Specification
IMS SS	IMS Simple Sequencing Specification
IMS SS ASM	IMS SS Activity State Model
IMS SS SDM	IMS SS Sequencing Definition Model
IMS SS TM	IMS SS Tracking Model
IP	Internet Protocol
IRTL	Information Resource Transaction Layer
ISO/IEC	International Standards Organization/International Electrotechnical Commission
ISO/IEC JTC1 SC36	ISO/IEC Joint Technical Committee 1 Subcommittee 36
IT	Information Technology
ITS	Intelligent Tutoring Systems
J2EE	Java 2 Platform Enterprise Edition
J2ME	Java 2 Platform Micro Edition
JDK	Java Development Kit
JMS	Java Message Service
JSP	Java Server Pages
LCMS	E-Learning Content Management System
LMS	Learning Management System
MIDP	Mobile Information Device Profile
MIME	Multi-purpose Internet Mail Extensions
NAT	Network Address Translation
OAI	Open Archives Initiative
OAI-PMH	OAI Protocol for Metadata Harvesting
OGSA	Open Grid Services Architecture
OKI	Open Knowledge Initiative
OKI OSID	OKI Open Service Interface Definition
OODB	Object-Oriented DataBase
OWL	Web Ontology Language
P2P	Peer to Peer
PADLR	Personalized Access to Distributed Learning Repositories
PC	Personal Computer
PDA	Personal Data Assistant
POOL	Portal for Online Objects in Learning
QBE	Query by Example
RDB	Relational DataBase
RDF	Resource Description Framework
RDF-QEL	RDF Query Exchange Language
RDFS	RDF Schema
RMI	Remote Method Invocation
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOAP RPC	SOAP Remote Procedure Call
SCA	Sharable Content Assets
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model

SCORM CAM	SCORM Content Aggregation Model
SCORM RTE	SCORM Run-Time Environment
SCORM SDM	SCORM Sequencing Definition Model
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
TCP	Transport Control Protocol
TLS	Transport Layer Security
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
URI	Universal Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WebDAV	Web-based Distributed Authoring and Versioning
WG	Working Group
WSDL	Web Service Definition Language
WSFL	Web Services Flow Language
WWW	World Wide Web
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language Transformations

1. Introduction

In a broad sense, E-Learning can be defined as the effective learning process created by combining digitally delivered content with learning support and services [49]. E-Learning pursues to eventually enable some high-level requirements provided by Web-based education and training, including [2]:

- *Accessibility*: The ability to locate and access instructional components from one remote location and deliver them to many other locations.
- *Interoperability*: The ability to take instructional components developed in one location with one set of tools or platform and use them in another location with a different set of tools or platform.
- *Durability*: The ability to withstand technology changes without redesign, reconfiguration or recoding.
- *Reusability*: The flexibility to incorporate instructional components in various applications and contexts.

From the functional perspective, E-Learning encompasses four major components: the E-Learning content creation, E-Learning content management, E-Learning content distribution and E-Learning community collaboration. Among them, the E-Learning content management is the most important component of a generic E-Learning architecture. It constitutes the basis for all other E-Learning components, enabling authoring, locating, reusing, organizing, personalizing, exchanging, delivering and marketing learning content based on network technologies.

Generally, the E-Learning content management has to accomplish two major functionalities. The first functionality is the learning content discovery, which enables people to find learning content and understand context between them through appropriate learning content description. The second functionality is the learning content delivery, which enables people to organize, re-purpose and deliver learning content for achieving new learning objectives. Although all these two functionalities are of importance for the E-Learning content management, the learning content discovery plays a more vital role. Actually, successfully discovering learning content and understanding their built-in context is the prerequisite to the accomplishment of the effective E-Learning content management.

In conventional LCMSs, the two functionalities of the E-Learning content management may be accomplished through various technical approaches. Individual LCMSs may use arbitrary formats to describe learning content and their context, adopt arbitrary strategy to discover learning content across educational repositories, use arbitrary methods to describe and organize learning content aggregations, and build arbitrary run-rime environments to deliver learning content. As most of these technical approaches are based on highly proprietary designs, partly or entirely, it is rather hard to achieve interoperability, reusability and portability of learning content across various educational repositories and LCMSs.

For E-Learning, one key to accomplishing its “abilities” is the E-Learning standardization, which is purposed to standardize every components of a generic E-Learning architecture by means of sets of E-Learning standards and specifications. Just like standardization in other application areas, the E-Learning standardization can essentially enable E-Learning technologies to scale, to work together independent of who built them, and eventually reduce E-Learning costs as well as create common E-Learning markets. Starting from the most important component of E-Learning, in this thesis we focus on standardization of the E-Learning content management and propose the standard-oriented E-Learning content management as the first step towards the standard-oriented E-Learning. The standard-oriented E-Learning content management is aimed to essentially enable interoperability, reusability and portability of learning content by means of standardizing the whole process of the learning content discovery and delivery.

1.1. Motivation

The research in this thesis is aimed to address several critical issues regarding the design and implementation of two major functionalities of the standard-oriented E-Learning content management.

Standard-oriented Learning Content Discovery. For the standard-oriented learning content discovery, the most critical design and implementation issue to address is interoperability between various learning resource metadata sets as well as between their different bindings. As the E-Learning standardization is still in the early stage, E-Learning standards and specifications specified for the learning content discovery are far from unified. Until the end of 2003, there have been over 10 E-Learning standards and specifications commonly applied in this area, possessing varying size of user communities, different semantic data models as well as different binding formats. In consequence, educational repositories constructed based on these standards and specifications are rather heterogeneous in the sense that they adopt various learning resource metadata sets as well as their different bindings to describe learning content, and apply various back-end systems to store learning resource metadata. In order to discovery learning content across these heterogeneous educational repositories, the standard-oriented learning content discovery must first address some critical interoperability issues such as how to cross incompatibility between different semantic data models of learning resource metadata sets, how to handle different bindings of these standards and specifications in a consistent manner, as well as how to determine the minimal interoperable basis for various learning resource metadata sets to enable a consistent discovery strategy, etc. In addition, the standard-oriented learning content discovery also has to adopt an effective infrastructure to connect heterogeneous educational repositories in a way that it is able to facilitate the Web-based E-Learning content management, fit into a generic E-Learning architecture, and provide technical support for achieving interoperability between various back-end systems, between various learning resource metadata sets as well as between different bindings. In this thesis, we propose Edutella[1], an RDF-based E-Learning content management P2P infrastructure, to address all these issues.

Standard-oriented Learning Content Delivery. For the standard-oriented learning content delivery, the most critical design and implementation issue to address is reusability and portability of learning content. A typical learning content delivery process is composed of two major operations: the learning content organization and run-time environment implementation. On the one hand, the standard-oriented learning content delivery must support standardization of the learning content organization, being able to address some critical issues such as how to define content aggregations, how to represent the content packaging and content sequencing, etc. On the other hand, it also has to implement standard run-time environments, being able to address some critical issues such as how to track learning content and manage course status in a standard way, how to standardize the communication between learning content and run-time environments, etc. Moreover, as the current E-Learning content management puts more and more emphasis on personalization and adaptation abilities of the learning content delivery [17], the standard-oriented learning content delivery also has to explore approaches for achieving these advanced abilities while ensuring reusability and portability of learning content. In this thesis, we propose a reference model and sets of enabling technologies to address all these issues.

As an engineering approach, the research in thesis is basically distinct from other related research (see also Section 2.3) in that it provides both the reference model design and technical implementation support for the standard-oriented E-Learning content management. With regard to the standard-oriented learning content discovery, Edutella can connect heterogeneous educational repositories by means of a P2P infrastructure, and further achieve interoperability between various learning resource metadata sets and their different bindings by means of RDF. With regard to the standard-oriented learning content delivery, the proposed reference model covers most of applicable E-Learning standards and specifications, whose implementation is supported by sets of enabling technologies based on standard Web technologies and metadata technologies. Going a step further, we apply the reference model to conventional AEHSs and investigate the approach for converging the E-Learning standardization and conventional AEHS research to realize the standard-oriented AEHS, which can ensure reusability and portability of learning content while achieving adaptability of the learning content delivery.

1.2. Research Context

The research in this thesis is conducted in the context of two internationally cooperated academic projects.

PADLR Project. The driving vision of PADLR[2] is a distributed "E-Learning Web infrastructure", which makes it possible to exchange/author/annotate/organize/market and personalize/navigate/use/reuse modular learning content. PADLR is comprised of a set of sub-projects that can be grouped into three intertwined modules as illustrated in Figure 1.1.

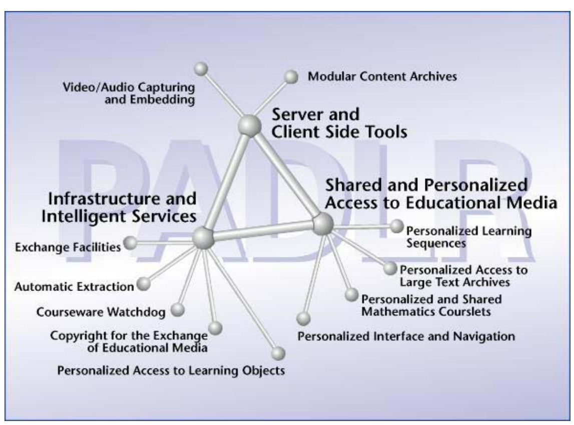


Figure 1.1 PADLR project

[1] http://edutella.jxta.org
[2] http://www.learninglab.de/padlr/index.html

- *Infrastructure and intelligent services*: The module of the "infrastructure and intelligent services" includes work on exchange facilities and basic infrastructure, personalized queries and views over distributed learning materials, automatic extraction of metadata and ontological information, as well as courseware discovery and annotation.
- *Server and client side tools:* The module of the "server and client side tools" includes work on modular content archives and video/audio capturing and metadata annotation tools.
- *Shared and personalized access to educational media:* The module of the "shared and personalized access to educational media" includes work on personalized learning sequences, interfaces and guidance, personalized access to large text archives, personalized and shared mathematics courselets, as well as Web services oriented courseware, marketing efforts and copyright issues.

The research in this thesis mainly contributes to the module of the "infrastructure and intelligent services", focusing on providing the basic infrastructure and exchange facilities for the whole PADLR project. Some research output also directly contributes to other PADLR sub-projects, particularly the sub-project of the "modular content archives" and "personalized learning sequencing".

ELENA Project. The objective of the EU/IST project ELENA[3] is to create a smart space for E-Learning building upon the P2P infrastructure. A smart space for E-Learning is defined in ELENA as the educational service mediator, which is comprised of educational nodes, delivering learning services, and can support the "smart" mediation of learning services based on the learner profiling, service evaluation and reputation ratings [102]. In Figure 1.2 we illustrate the architecture of the ELENA project [102].

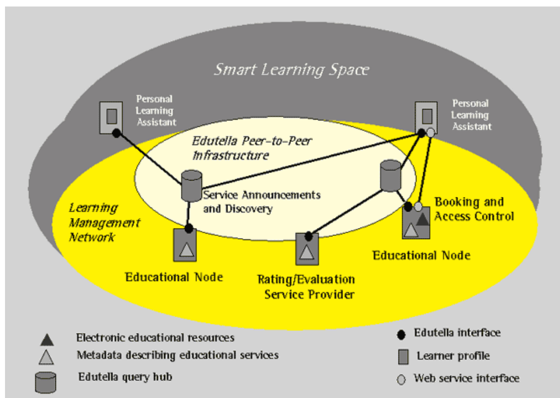


Figure 1.2 ELENA project

At the architecture level, ELENA shares with PADLR the same basic infrastructure. However, as a generic educational service mediator, it also plans some extensions to the basic P2P infrastructure, particularly including Web services into the essential exchange facilities [102]. The research in this thesis meets this new requirement of ELENA by proposing an approach for interacting Edutella with Web services. In addition, some enabling technologies developed in this thesis for the standard-oriented learning content discovery and delivery can also directly be applied in ELENA.

1.3. Contribution of the Study

We make following principal contribution in this thesis:

- Propose Edutella, an RDF-based E-Learning content management P2P infrastructure, to implement the standard-oriented learning content discovery [74].
- Propose an approach for interacting Edutella/JXTA with Web services as the first exploration towards a unified distributed computing architecture converging P2P and Web services for E-Learning [94].
- Propose sets of enabling technologies to achieve interoperability between various back-end systems, between various learning resource metadata sets as well as between their XML and RDF bindings for the standard-oriented learning content discovery in Edutella [91][92][95].
- Propose a reference model for the standard-oriented learning content delivery. Propose sets of enabling technologies for the implementation of the reference model [89][90][93].
- Propose an approach for converging the E-Learning standardization and conventional AEHS research to realize the standard-oriented AEHS.

[3] http://www.elena-project.org/

Most of contribution of this thesis has been published as a number of scientific papers. In addition, all implementation source code can be downloaded from the Edutella website as part of the Edutella open source output.

1.4. Thesis Overview

In chapter 2 we provide the background of the E-Learning standardization and E-Learning content management. We first overview the current E-Learning standardization and then introduce two major types of distributed infrastructures applicable to the Web-based E-Learning content management: the centralized infrastructure and decentralized infrastructure. At last we present several decentralized LCMSs and some relevant research as the related work of this thesis

In chapter 3 we first introduce the general design of Edutella and then present its two cornerstone technologies: the P2P platform JXTA and metadata language RDF. We further detail several critical design and implementation issues in Edutella, such as the super peer based network topology, RDF-QEL, ECDM, etc.

In chapter 4 we propose an approach for interacting Edutella/JXTA with Web services with the purpose of exchanging distributed functionalities between the two platforms. We first discuss several challenges we have to tackle while achieving the interaction and then detail the technical implementation of the service layer interaction between Edutella/JXTA and Web services in terms of two typical interaction scenarios: exposing existing Edutella/JXTA P2P services as Web services, and integrating Web services enabled content providers into Edutella/JXTA.

In chapter 5 we propose two different approaches for integrating XML metadata repositories into Edutella in terms of XML query languages they support. We first discuss several challenges we have to tackle while achieving the integration and then detail two integration approaches respectively applicable to XPath-enabled and XQuery-enabled XML metadata repositories. At last we propose an approach for querying complex XML data schemas in Edutella through QBE.

In chapter 6 we propose a reference model for the standard-oriented learning content delivery, and further present sets of enabling technologies for the implementation of the reference model. We then propose an approach for converging the E-Learning standardization and conventional AEHS research to realize the standard-oriented AEHS. At last we provide a design comparison between the standard-oriented AEHS and conventional AEHS.

In chapter 7 we conclude the research in this thesis.

2. Background and Related Work

This chapter provides the background of the E-Learning standardization and E-Learning content management. In section 2.1 we first overview the current E-Learning standardization. In section 2.2 we introduce two major types of distributed infrastructures applicable to the Web-based E-Learning content management: the centralized infrastructure and decentralized infrastructure. In section 2.3 we present several decentralized LCMSs and some relevant research as the related work of this thesis.

2.1. Overview of the E-Learning Standardization

Standards can be defined as "documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure that materials, products, processes and services are fit for their purpose" [58]. The E-Learning standardization is purposed to provide sets of standards for E-Learning. At present, the E-Learning standardization is principally promoted by three types of organizations [36]:

- *Accredited standardization bodies*: The accredited standardization bodies are responsible for developing and approving standards. Currently such a type of organizations mainly include the IEEE LTSC[4], ISO/IEC JTC1 SC36 [5] and CEN/ISSS LTWS[6], etc.
- *Consortia*: The consortia produce specifications that the standards will be based on. Currently such a type of organizations mainly include IMS[7], AICC[8] and ARIADNE[9], etc.
- *Communities*: The communities adopt application profiles that adapt the standards and specifications to their own needs and requirements. Currently such a type of organizations mainly include the ADL initiative[10], CanCore initiative [11], DCMI-Ed[12] and ALIC[13], etc.

Apart from these principal contributors, there are also many other organizations that have contributed a lot to the E-Learning standardization. Two notable ones of them are W3C[14] and DCMI[15]. W3C develops sets of interoperable technologies such as specifications, guidelines, software and tools, etc., which form the foundation upon which E-Learning standards and specifications as well as E-Learning architectures can be built. DCMI produces several important metadata specifications that lay the basis for today's Web-based E-Learning content management. As these organizations address a very broad application area, taking the E-Learning standardization only as a small part, we do not categorize them as principal contributors to the E-Learning standardization. However, their output will be referenced throughout this thesis.

In Figure 2.1 we illustrate principal contributors to the E-Learning standardization and their corresponding output [2]. This figure also perfectly schematises the development process of E-Learning standards and specifications.

With regard to the output of the E-Learning standardization, we often distinguish between specifications, standards and application profiles [36].

- *Specifications*: The specifications represent standards early in their development, prior to receiving approval from standards bodies. They capture a rough consensus, and are usually experimental, incomplete and rapidly evolving.
- *Standards*: The standards are more conclusive, complete, and evolve more slowly. They should capture general acceptance, and can serve regulatory purposes, as well as be used to manage long term risk.
- *Application profiles:* The application profiles are implementations and reference models that refer to ways that specifications or standards are applied in communities. They include systems and tool development, as well as "application profiling" work that integrates multiple specifications or standards, or interprets and applies a single standard.

---

[4] http://ltsc.ieee.org/
[5] http://jtc1sc36.org/index.html
[6] http://www.cenorm.be/isss/workshop/lt/
[7] http://www.imsproject.org/
[8] http://www.aicc.org/
[9] http://www.ariadne-eu.org/
[10] http://www.adlnet.org/
[11] http://www.cancore.ca/indexen.html
[12] http://dublincore.org/groups/education/
[13] http://www.alic.gr.jp/
[14] http://www.w3.org/
[15] http://dublincore.org/

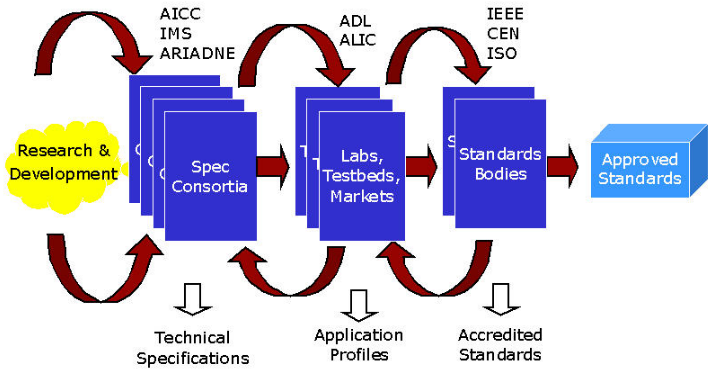


Figure 2.1 Principal contributors to the E-Learning standardization and their output

Until the end of 2003, the E-Learning standardization is still in the early stage. In consequence, even E-Learning standards are sometimes under change over the time according to newly emerging requirements, let alone E-Learning specifications and application profiles. However, it is worth noting that the design goals and design perspectives of the current E-Learning standards and specifications have mostly been determined. As these design goals and perspectives envisaged by different organizations have considerable overlap between each other, it is relatively difficult to categorize E-Learning standards and specification in an unambiguous way. In order to overview the E-Learning standardization, we roughly categorize E-Learning standards and specifications into three categories with the intension to identify three focuses of the current E-Learning standardization. These three categories are the E-Learning learner management, E-Learning content management and E-Learning architecture design, covering almost all of E-Learning standards and specifications contributed by six major E-Learning accredited standardization bodies and consortia: the IEEE LTSC, ISO/IEC JTC1 SC36, IMS, CEN/ISSS LTWS, AICC and ARIADNE.

These three categories of E-Learning standards and specifications address E-Learning at different levels. Among them, the category of the E-Learning architecture design sits on the top level, providing guidance and recommendations for the design of E-Learning standards and specifications in the other two categories. The category of the E-Learning learner management and E-Learning content management sit at the lower level, addressing two specific aspects of E-Learning. Relatively, the category of the E-Learning content management plays a more important role. It acts as the key to achieving interoperability, reusability and portability of learning content, and also usually cooperates with standards and specifications in the category of the E-Learning learner management, which provide the basis for personalizing or cooperating the E-Learning content management process. Until the end of 2003, most of E-Learning standards and specifications from the ISO/IEC JTC1 SC36, CEN/ISSS LTWS, ARIADNE and AICC are far from finalized. They only represent the directions envisioned by these organizations for the E-Learning standardization. In contrast, some standards and specifications from the IEEE LTSC and IMS are relatively stable, although some future changes cannot yet be excluded. Currently, the only stable E-Learning standard is the IEEE LOM. It was approved by IEEE-SA in June 2002.

2.1.1. Standardization of the E-Learning Architecture Design

In Table 2.1 we list standards and specifications designed for the E-Learning architecture design.

Table 2.1 E-Learning standards and specifications for the E-Learning architecture design

IMS	IEEE LTSC	ISO/IEC JTC1 SC36	CEN/ISSS LTWS	AICC
IMS Enterprise Spec.	IEEE P1484.3 Standard for Learning Technology Glossary	WG5: Quality Assurance and Descriptive Frameworks	CWA14644 Quality Assurance Standards	AGR002- Courseware Delivery Stations: Hardware
IMS DRI	IEEE P1484.1 LTSA		CWA 14040 A Standardization Work Programme for “Learning and Training Technologies & Educational Multimedia Software”	AGR004-Courseware Delivery Stations: Software
IMS Learning Design Spec.	IEEE P1484.15 Standard for Data Interchange Protocols			AGR005-CBT Peripheral Device
IMS RDCEO	IEEE P1484.14 Standard for Semantics and Exchange Bindings			
	IEEE P1484.20 Standard for Competency Definitions			

E-Learning standards and specifications contained in the category of the E-Learning architecture design specify a high-level architecture for E-Learning. They describe the high-level system design and the components of E-Learning systems covering the E-Learning technology glossary, E-Learning framework design, quality assurance, architectural level interoperability and data exchange, E-Learning system modelling, competency definition and delivery devices, etc. These standards and specifications can promote interoperability and portability of E-Learning systems by identifying critical system interfaces, thus promote the design and implementation of E-Learning components and sub-systems that are reusable, cost-effective and adaptable to new technologies. For the design and implementation of the standard-oriented E-Learning content management, these standards and specifications can provide us with some high-level guidance.

2.1.2. Standardization of the E-Learning Learner Management

In Table 2.2 we list standards and specifications designed for the E-Learning learner management.

Table 2.2 E-Learning standards and specifications for the E-Learning learner management

IMS	IEEE LTSC	ISO/IEC JTC1 SC36
IMS LIP	IEEE P1484.2 Standard for Learner Model	WG2: Learner to Learner Interaction Schema; Agent/Agent communication; Collaborative workplace
IMS Learner Information Package Accessibility for LIP		WG3: Participant Information

E-Learning standards and specifications contained in the category of the E-Learning learner management describe the characteristics of a learner needed for the general purposes of recording and managing learning related history, goals and accomplishments, engaging a learner in a learning experience, and discovering learning opportunities for learners. They provide standardized learner model, which constitutes the basis for the realization of adaptive and personalized E-Learning systems to enable content developers to provide more personalized and effective instruction. For the design and implementation of the standard-oriented E-Learning content management, these standards and specifications can provide the basis for implementing personalization and cooperation features.

2.1.3. Standardization of the E-Learning Content Management

As the focus of the E-Learning standardization, the category of the E-Leaning content management includes most of standards and specifications that are currently under active development and are also relatively mature. For the sake of clarification, we divide this category into four sub-categories: the E-Learning content vocabulary, E-Learning content formats, E-Learning content metadata and E-Learning content delivery. In Table 2.3 we list standards and specifications designed for the E-Learning content management.

Table 2.3 E-Learning standards and specifications for the E-Learning content management

IMS	IEEE LTSC	ISO/IEC JTC1 SC36	CEN/ISSS LTWS	AICC	ACM/DCMI/ ARIADNE
Sub-category of the E-Learning Content Vocabulary					
		WG1: Vocabulary			ACM CCS
Sub-category of the E-Learning Content Formats					
	IEEE 1484.18 Standard for Platform and Media Profiles		CWA 14590 Description of Language Capabilities	AGR003-Digital Audio	
				AGR008- Digital Video	
				AGR009-Icon Guidelines	
Sub-category of the E-Learning Content Metadata					
IMS MD	IEEE LOM	WG4: Management and Delivery for Learning, Education and Training	CWA14645 Availability of alternative language versions of a learning resource in IEEE LOM		DCMI DCMES
	IEEE P1484.12.2 ISO/IEC 11404 Binding for IEEE 1484.12.1 LOM		CWA 14643 Internationalisation of the IEEE LOM		DCMI Metadata Terms

	Data Model				
	IEEE P1484.12.3 XML binding for IEEE 1484.12.1 LOM Data Model				ARIADNE Metadata Recommendation
	IEEE P1484.12.4 RDF binding for IEEE 1484.12.1 LOM Data Model				DCMI Encoding Guidelines: Expressing Simple Dublin Core in RDF/XML
					DCMI Encoding Guidelines: Expressing Qualified Dublin Core in RDF/XML
Sub-category of the E-Learning Content Delivery					
IMS CP	IEEE Data Model	WG4: Management and Delivery for Learning, Education and Training		CMI001-AICC/CMI Guidelines for Interoperability	
IMS SS	IEEE API			AGR006-CMI	
IMS QTI				AGR007-Courseware Interchange	
				AGR010- Web-based CMI	

Sub-category of the E-Learning Content Vocabulary. E-Learning standards and specifications contained in this sub-category are intended to facilitate international communication in information technology for learning, education and training. They present core terms and definitions of selected concepts, and identify relationships among the entries for describing and classifying learning content. The principal contributor to this sub-category is the ISO/IEC JTC1 SC36 WG1. Besides, ACM also contributes to this sub-category through the ACM CCS [1].

Sub-category of the E-Learning Content Formats. E-Learning standards and specifications contained in this sub-category are intended to identify existing standards and specifications of learning technology platforms and their content in order to provide compatibility for representing content formats and functionality. Some standards and specifications in this sub-category are developed for several operating scenarios such as “browser platform”, “workstation platform” or “Web media types”, etc., which are generally based on functionality thus allow consumers and vendors a wider range of implementations that are conforming. The principal contributor to this sub-category is the IEEE LTSC 1484.18 Standard for Platform and Media Profiles WG, which also references the output from AICC. This sub-category, together with the sub-category of the E-Learning content vocabulary, constitutes the basis for other standards and specifications contained in the category of the E-Learning content management, providing a uniform and standard approach for describing E-Learning content.

Sub-category of the E-Learning Content Metadata. E-Learning standards and specifications contained in this sub-category specify the syntax and semantics for the standard description of instructional components for education and training. In E-Learning, these instructional components are usually referred to as learning objects. Despite that there is a little bit of variation in perception about the size, the scope and the assessment of learning objects, a learning object is usually defined as “any digital or non-digital entity, which can be used, reused or referenced during technology supported learning”[115]. As this sub-category provides a standard approach for the characterisation of learning objects, it can be viewed as the first area where the E-Learning standardization has to address in order to achieve interoperability, reusability and portability of learning content.

Until the end of 2003, the sub-category of the E-Learning content metadata is the most advanced area of the E-Learning standardization. The leading role in this sub-category is the IEEE LOM [51], which is proposed based on the IMS MD [54] and ARIADNE Metadata Recommendation [6], and also makes heavy use of the DCMES [33] and DCMI Metadata Terms [34]. The IEEE LOM also has two accompanying specifications provided by CEN/ISSS LTWS: the CWA 14643 Internationalisation of the IEEE Learning Object Metadata [22] and CWA14645 Availability of Alternative Language Versions of a Learning Resource in IEEE LOM [23]. Besides, it has three binding specifications: the IEEE P1484.12.2 ISO/IEC 11404 Binding for IEEE 1484.12.1 LOM Data Model, IEEE P1484.12.3 XML binding for 1484.12.1 Data Model and IEEE P1484.12.4 RDF binding for 1484.12.1 Data Model. Since these three binding specifications cannot be expected to be finalized before 2006, in practice the IMS MD XML Binding Specification [54] and IMS MD RDF Binding Specification [54] are usually used as the XML binding and RDF binding specification of the IEEE LOM. The future IEEE

P1484.12.3 XML binding specification and IEEE P1484.12.4 RDF binding specification are expected to directly be derived from these two IMS binding specifications.

Apart from the IEEE LOM, other two important specifications in this sub-category are the DCMES and DCMI Metadata Terms, which, as the general-purpose metadata specifications for describing Web resources, are also frequently used to annotate learning objects independent of the IEEE LOM. Similar to the IEEE LOM, the DCMES and DCMI Metadata Terms also have corresponding encoding guidelines for the XML and RDF binding, respectively the "Expressing Simple Dublin Core in RDF/XML" [7] and "Expressing Qualified Dublin Core in RDF/XML" [62]. In Figure 2.2 we illustrate the relationships between all E-Learning content metadata standards and specifications. As currently various E-Learning standards and specifications co-exist in this area, a critical issue to address for the standard-oriented learning content discovery is to achieve interoperability between various learning resource metadata sets as well as between their different bindings.

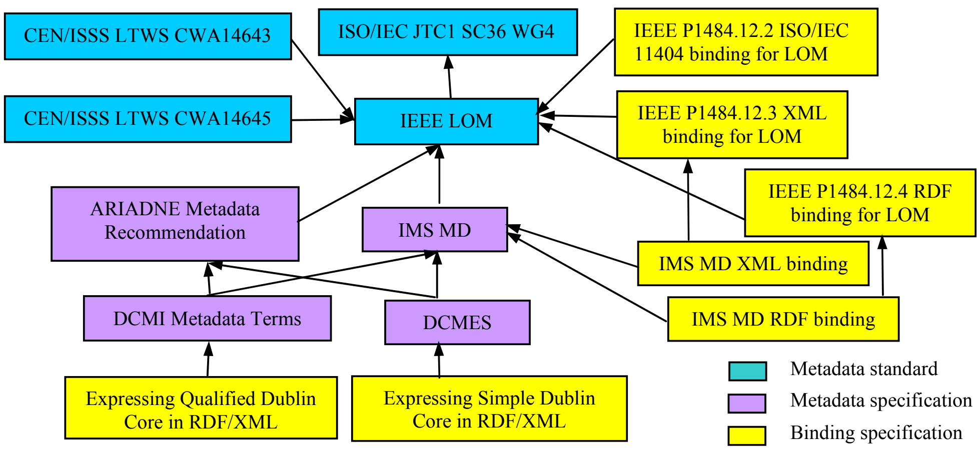


Figure 2.2 Relationships between E-Learning content metadata standards and specifications

Sub-category of the E-Learning Content Delivery. E-Learning standards and specifications contained in this sub-category are intended to standardize two major operations of a learning content delivery process: the learning content organization and run-time environment implementation. The learning content organization is concerned with the creation, composition and aggregation of sets of simple learning objects into more complex learning content, as well as the organization of these learning content into a defined sequence for delivery. The run-time environment implementation accomplishes the practical delivery of learning content, enabling the communication between learning content and run-time environments and further launching, tracking learning content as well as controlling their delivery sequences. In Figure 2.3 we illustrate a complete E-Learning content delivery framework [55] envisioned by the IMS Global Learning Consortium, Inc.

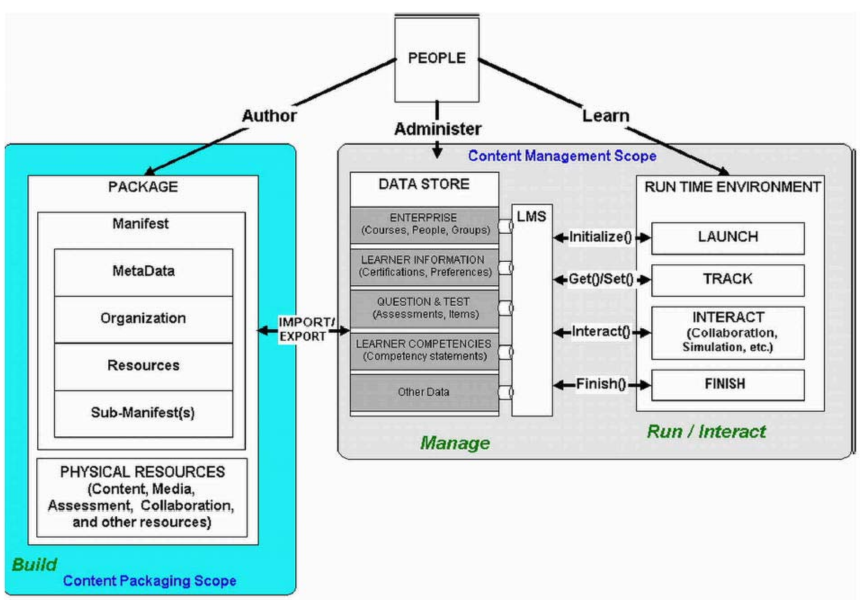


Figure 2.3 A complete E-Learning content delivery framework

With regard to the run-time environment implementation in a complete E-Learning content delivery framework, the most important E-Learning standard is the IEEE Data Model [52], which specifies the information that may be communicated between learning content and LMSs thus enables LMSs to gather information on the learner's performance on each learning objects. Besides, another important standard is the IEEE API [53], which specifies standard mechanisms to enable the communication between learning content and LMSs based on the IEEE Data Model. As part of the overall CMI design, both IEEE Data Model and IEEE API are derived from the AICC specification: CMI 001- AICC/CMI Guidelines for Interoperability [4].

With regard to the learning content organization in a complete E-Learning content delivery framework, the most important E-Learning specification is the IMS CP [55], which is focused on achieving interoperability between E-Learning systems that wish to import, export, aggregate and disaggregate packages of learning objects by means of defining a standardized set of structures that can be used to exchange learning content. Besides, another important specification is the IMS SS [56], which defines a standard-based method for representing the intended behaviour of an authored learning experience such that any LMS can sequence discrete learning activities in a consistent way according to the outcomes of the learner's interactions with content. In practice, the IMS SS is also closely related to the IMS QTI [57], which defines a means to associate competency, skill, mastery or learning objectives with a question or set of questions with the purpose of supporting the exchange of question and test data between different LMSs in a standard way. During the design of the content sequencing, the IMS QTI is usually used to provide the mastery test results for associated rules predefined by the IMS SS. Actually, in a complete E-Learning content delivery framework, the IMS CP, IMS SS, IMS QTI, IEEE Data Model and IEEE API are interrelated to each other. The IMS SS and IMS QTI directly leverage learning content structures defined by the IMS CP to describe content behaviours. The IEEE Data Model and IEEE API determine how information flows between the IMS QTI and IMS SS. For the standard-oriented learning content delivery, a critical issue to address is to define a reference model to cover all these E-Learning standards and specifications. Moreover, we also need to develop sets of enabling technologies to implement the reference model based on standard Web technologies and metadata technologies.

2.1.4. E-Learning Application Profiles

The E-Learning application profile can be viewed as a compound specification that combines multiple E-Learning standards and specifications and further adapts these standards and specifications to satisfy specific application purposes and requirements [36]. As no single E-Learning standard or specification can accommodate all functional requirements of E-Learning systems, E-Learning application profiles pursue to mix and match sets of interrelated E-Learning standards and specifications in order to cross boundaries between different categories of the E-Learning standardization to address E-Learning at a higher level. Until the end of 2003, most of existing E-Learning application profiles are designed with the focus on the E-Learning content management. In Table 2.4 we list four major E-Learning application profiles: the ADL SCORM, CanCore, DCMI-Ed application profile and ALIC application profile. Whereas the CanCore and DCMI-Ed application profile can be categorized into the sub-category of the E-Learning content metadata, the ADL SCORM and ALIC application profile cover a much broader application area, referencing most of E-Learning standards and specifications in the category of the E-Learning content management.

Table 2.4 E-Learning application profiles

Application Profiles	E-Learning Standards and Specifications Referenced
ADL SCORM	IEEE LOM/IMS MD IMS CP IMS SS IEEE P1484.12.3 XML binding for IEEE 1484.12.1 LOM Data Model/IMS MD XML Binding IEEE P1484.12.4 RDF binding for IEEE 1484.12.1 LOM Data Model/IMS MD RDF Binding IEEE Data Model /CMI 001- AICC/CMI Guidelines for Interoperability IEEE API /CMI 001- AICC/CMI Guidelines for Interoperability
CanCore	IEEE LOM/IMS MD
DCMI-Ed	IEEE LOM/IMS MD DCMES DCMI Metadata Terms
ALIC	ADL SCORM IEEE LOM/IMS MD IMS QTI IMS LIP

In a sense, some E-Learning application profiles can be viewed as the elementary design of the standard-oriented E-Learning content management. Taking the ADL SCORM, a typical E-Learning application profile, as an example, it references almost all major standards and specifications in the category of the E-Learning content management. In the SCORM CAM, the SCORM not only defines the components used to build a learning experience from reusable learning resources, but also describes how to represent the intended behaviour of a learning experience (content structure and content sequencing) and how to package learning resources for the movement between different LMSs (content packaging) [2]. In the SCORM RTE, the SCORM addresses the standard-based run-time environment implementation by defining a common mechanism to manage the learning content delivery based on the IEEE Data Model and IEEE API. For the design of the standard-oriented E-Learning content management, these application profiles can be taken as the primary framework, which has to be complemented by sets of enabling technologies supporting the technical implementation of the standard-oriented content management, e.g., technologies for networking heterogeneous educational repositories to enable the standard-oriented learning content discovery, technologies for addressing interoperability between various learning resource metadata sets as well as between their different bindings, and technologies for implementing the standard-oriented learning content delivery, etc. Actually, the research in this thesis can be viewed as such sort of complementary work. In a sense, it can be compared as a "technical" E-Learning application profile, principally addressing the standard-oriented E-Learning content management using an engineering approach.

2.2. E-Learning Content Management and its Infrastructure Design

The goal of the E-Learning content management is to accumulate, assemble and distribute personalized E-Learning through providing the more effective access to learning content by means of sets of tools and services for searching, locating, organizing and delivering learning content based on the learner's learning style and needs. Going a step further, the standard-oriented E-Learning content management pursues the same goal by means of the E-Learning standardization rather than other proprietary technologies. It is focussed on implementing all E-Learning content management functionalities based on E-Learning standards and specifications as well as sets of standard technologies to ensure interoperability, reusability and portability of learning content.

At present, almost all of LCMSs are Web-based, adopting various distributed computing paradigms as the basis of their infrastructure design. In terms of distributed technologies applied, E-Learning content management infrastructures can be divided into two types: the centralized infrastructure and decentralized infrastructure.

2.2.1. Centralized E-Learning Content Management Infrastructure

The centralized E-Learning content management infrastructure adopts the client-server design pattern using a single server to centralize all distributed functionalities, services and information. This single server serves as the unique access point of the whole system, enabling clients to submit and retrieve information as well as to accomplish usual content management activities such as creating index, cataloguing resources, etc. As a typical working scenario in the centralized E-Learning content management infrastructure, content providers submit learning objects or learning object metadata onto the central repository, centralized control mechanisms accomplish indexing and cataloguing, and then content consumers search and retrieve information from the central repository taking advantage of indices or catalogues. Since the pure client-server architecture usually has to face scalability and decentralization problems, the concept of middleware is often introduced into the purely centralized design pattern to provide interoperability and transparent location of servers. Enabled by some distributed technologies such as RMI, CORBA, DCOM and EJB, middleware can provide the centralized E-Learning content management infrastructure with transparency layers that deal with distributed system complexities such as location of objects, heterogeneity of software platforms and back-end systems [83], etc. Apart from the client-server architecture, some less decentralized infrastructures adopted by LCMSs can also be categorized into the centralized E-Learning content management infrastructure, e.g., E-Learning content management infrastructures similar to SETI@Home[16] and Napster[17], etc. Although these infrastructures adopt the decentralized design pattern to connect information sources, they still need a rather centralized component to accomplish all system functionalities.

In the E-Learning content management, a typical application of the centralized E-Learning content management infrastructure is the E-Learning portal, which consolidates disparate learning resources and technologies into a centralized point of access and management. In the past a few years, E-Learning portals top the list of effective E-Learning content management strategies, being able to offer features such as 24/7 accessibility, convenience and flexibility, cost-effectiveness, user-centric learning and easy customisation, etc. In the early stage, most of E-Learning portals are applied within business or academic communities with a limited number of users. They are usually supported by good indexing systems for learning content classification, identification and retrieval. With the expansion of user communities, E-Learning portals continuously have to

[16] http://setiathome.ssl.berkeley.edu/
[17] http://www.napster.com

evolve in order to deal with rapidly increasing learning content. Some efforts include, e.g., enriching the intelligence of tools and services, distributing systems functionalities [35], etc.

So far most of E-Learning portals are commercially marketed, e.g., Hyperwave eKnowledge[18], WebCT[19], etc., which increasingly depend on some middleware technologies and are also usually designed following some open industry standards as well as E-Learning standards and specifications, e.g., J2EE, OKI OSID [80], ADL SCORM, etc. Apart from these commercial products, there are also several academic E-Learning portals and some LCMSs adopting the design pattern similar to the E-Learning portal, e.g., the ARIADNE KPS[20], eCMS[21], Teachware on Demand[22], UNIVERSAL[23], etc. In order to present the general design of the centralized E-Learning content management infrastructure, we take a typical E-Learning portal: the ARIADNE KPS [35] as an example. In Figure 2.4 we illustrate the network topology of the ARIADNE KPS [35].

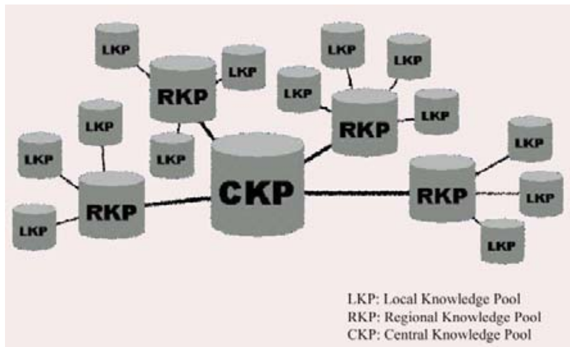


Figure 2.4 Network topology of the ARIADNE KPS

The ARIADNE KPS uses a star-shape network topology to connect and manage distributed educational repositories. This topology can be viewed as a hierarchical client-server architecture. The lowest level repositories are LKPs, which, acting as clients, replicate learning objects and learning object metadata onto RKPs. All RKPs are connected to a single central server CKP, which is responsible for managing the whole ARIADNE KPS, e.g., indexing learning content for the ARIADNE KPS, controlling the data replication between LKPs and RKPs, etc. For the ARIADNE KPS, the CKP serves as the unique access point and central control point. Supported by sets of auxiliary tools, learning content producers, instructors and learners can accomplish various E-Learning content management activities through corresponding ARIADNE interfaces operating via the CKP. In Figure 2.5 we illustrate the overall architecture of the ARIADNE KPS [35]. It clearly shows the centralized operation and control mode in the ARIADNE KPS.

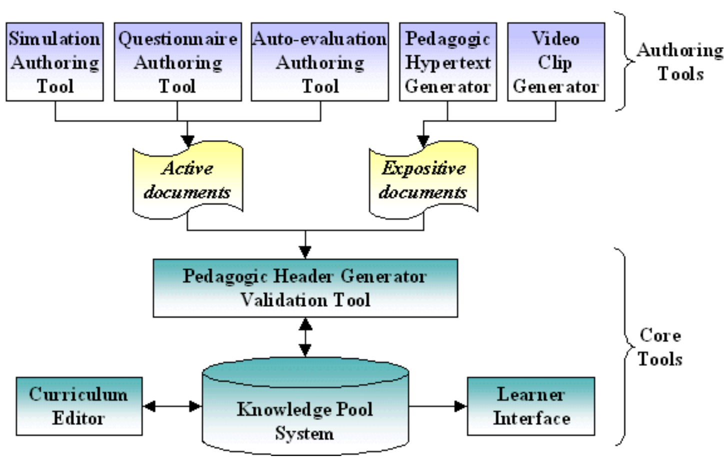


Figure 2.5 Overall architecture of the ARIADNE KPS

18 http://www.hyperwave.com/e/products/eki.html
19 http://www.webct.com/
20 http://www.ariadne-eu.org/en/system/index.html
21 http://elearning.noc.uth.gr/
22 http://www.teachware-on-demand.de
23 http://www.ist-universal.org

The centralized E-Learning content management infrastructure has several notable advantages, e.g., rapid content indexing and retrieval, easy content validation and cataloguing, controllable protection on rights of ownership and usage, easy user identification and access control, workload balancing, etc. However, with the expansion of user communities, the centralized E-Learning content management infrastructure also exposes some notable drawbacks.

First, the centralized E-Learning content management infrastructure separates learning content from their point of origin and their point of use [45]. Learning content producers often lose their control over learning content after uploading them onto the central server thus cannot easily and conveniently update learning content to support ad-hoc content management activities. For learners, they always need to come back to the unique access point to search and retrieve learning content, which unavoidably leads to the increase of network traffics. Moreover, nowadays there is a new trend in the E-Learning content management that more and more learning content are produced and managed by some small institutions or even ordinary instructors and learners. The centralized E-Learning content management infrastructure cannot easily and securely bring this part of learning content into the scope of management.

Second, the centralized E-Learning content management infrastructure cannot efficiently manage large amounts of learning content across heterogeneous educational repositories as well as across various metadata schemas. Although many centralized E-Learning content management infrastructures have adopted some decentralized design styles in order to enhance flexibility and scalability of systems, the increasing decentralization and heterogeneity of educational repositories continuously complicate the system design and maintenance. In addition, since most of centralized E-Learning content management infrastructures can only accommodate specific, and mostly unique metadata schemas as well as their specific bindings, they are usually beyond capabilities to address heterogeneity of the current E-Learning content management by means of the centralized control mode.

Third, the centralized E-Learning content management infrastructure improves the collapsing risk of systems [107]. Taking the ARIADNE KPS as an example, if the CKP collapses, the whole system goes down. This implies that the centralized E-Learning content management infrastructure needs more maintenance in order to provide stable and ad-hoc services. Actually, with the increase of the decentralization and heterogeneity of educational repositories and learning content, the centralized E-Learning content management infrastructure usually needs to adopt more and more decentralized design styles. Such design styles gradually lead to the emergence of the fully decentralized E-Learning content management infrastructure.

2.2.2. Decentralized E-Learning Content Management Infrastructure

The decentralized E-Learning content management infrastructure connects and manages heterogeneous educational repositories without a central control point. It typically adopts loosely coupled, highly distributed network topologies and leverages a SOA architecture to implement distributed functionalities. Nowadays there are several promising decentralized infrastructures that can be used to build fully decentralized LCMSs, e.g., P2P, Web services, OGSA/Grid [39], etc. Since P2P is the most typical decentralized computing architecture, we here take P2P as an example to present the design of the decentralized E-Learning content management infrastructure.

In the E-Learning content management P2P infrastructure, all peers communicate with each other symmetrically and have no clear differentiation between the roles of the client or server. As a typical working scenario, content providers manage educational repositories by themselves. These repositories are loosely networked following certain P2P protocols. Content consumers can connect to P2P networks from any points through mostly very simple bootstrapping process and then retrieve information utilizing distributed P2P services. Although the peer bootstrapping process sometimes involves a little bit of centralized control, e.g., utilizing DNS or IP addresses of some well-known starting points, the whole P2P infrastructure does not need a centralized access point or control point to accomplish system functionalities.

The P2P infrastructure mainly has three application areas: the distributed file sharing including applications such as Gnutella[24], FreeNet[25] and Morpheus[26], etc., the collaboration and messaging including applications such as Jabber[27] and Groove[28], etc., and the distributed computing including applications such as SETI@home and Metacomputing[29], etc. Although some work in the area of the distributed file sharing have some similarities to the decentralized LCMSs, until now there exist no well-known P2P-based decentralized LCMSs. In order to

[24] http://www.gnutella.com
[25] http://www.freenetproject.org
[26] http://www.morpheus.com/
[27] http://www.jabber.org
[28] http://www.groove.net
[29] http://www.cnds.jhu.edu/research/metacomputing

present the general design of the decentralized E-Learning content management P2P infrastructure, we take Gnutella, a typical P2P application for the distributed file sharing, as an example.

Gnutella itself is a protocol for distributed search. Although the Gnutella protocol supports a traditional client-server search paradigm, Gnutella's distinction is its decentralized P2P model, in which every client is a server, and vice versa. These so-called Gnutella servents perform tasks normally associated with both clients and servers. They provide client-side interfaces through which users can issue queries and view search results, while at the same time they also accept queries from other servents, check for matches against their local data set, and respond with applicable results [29]. The data transfer in Gnutella is end-to-end and does not use any central server. Due to its distributed nature, a network of servents is highly fault-tolerant, as operation of the network will not be interrupted if a subset of servents go offline.

As illustrated in Figure 2.6, Gnutella can implement four P2P operations based on five P2P protocols: Ping, Pong, Query, QueryHit and Push [29]. As these operations are typical for every P2P systems, the E-Learning content management P2P infrastructure also has to define similar protocols and implement similar operations.

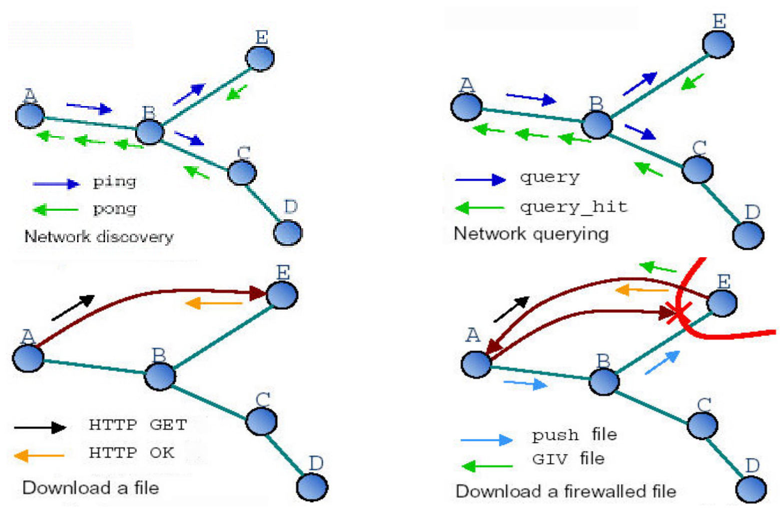


Figure 2.6 P2P operations supported by Gnutella protocols

The P2P infrastructure promises robustness, extensibility and infinite scalability for the decentralized E-Learning content management. However, such a decentralized infrastructure also means that P2P systems tend to be difficult to manage and that data in the systems are never fully authoritative. These systems also tend to be insecure in the sense that it is easy for a node to join the network and start putting unauthorized data into the system. In addition, scalability of P2P systems is hard to evaluate. In theory, the more hosts are added, the more capable a P2P network becomes. In practice, however, the algorithms required to keep a P2P system coherent often carry a lot of overhead. If that overhead grows with the size of the system, the system may not scale well. As the Web-based E-Learning content management involves a large community of content providers and content consumers as well as various learning content, for the design of the decentralized E-Learning content management infrastructure, these disadvantages of P2P systems have to be addressed carefully. Generally, we cannot directly use some existing P2P protocols or infrastructures, which are mostly concentrated on domain specific data formats, to build decentralized LCMSs, since the E-Learning content management involves more complicated resources than usual P2P applications thus requires more complex functionality implementations. The decentralized E-Learning content management infrastructure should possess some special abilities to meet high-level requirements of the E-Learning content management, e.g., the ability to support scalable and ubiquitous computing, the ability to cross heterogeneous educational repositories and various learning resource metadata sets for the learning content discovery, and the ability to facilitate the Web-based learning content delivery, etc. In order to address complexity and diversity of the E-Learning content management, we need some new designs of the decentralized computing architecture.

2.3. Related Work

The research in this thesis is mainly related with some emerging academic projects in the area of the decentralized E-Learning content management.

2.3.1. POOL Project

POOL is a project funded in part by the Canarie Learning Program to promote a national repository strategy for learning objects [45]. It was first conceptualised and implemented as a centralized E-Learning portal but shortly has migrated from the centralized client-server architecture to a decentralized P2P infrastructure with the purpose to encourage local management of learning content. In Figure 2.7 we illustrate the network architecture of POOL [45].

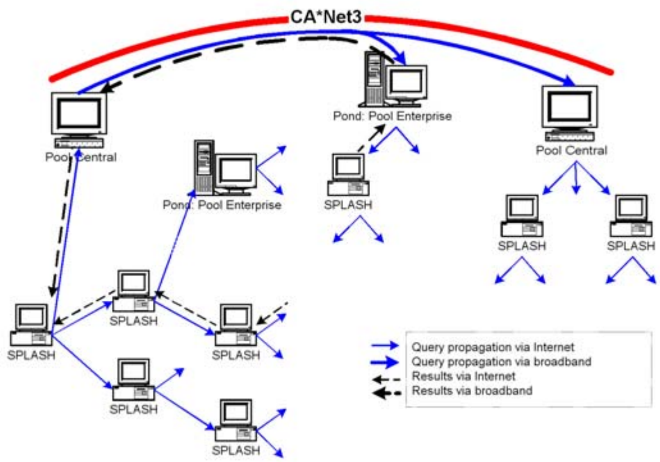


Figure 2.7 Network architecture of POOL

POOL is aimed to support the decentralized E-Learning content management based on the unique learning resource metadata set: the CanCore metadata standard [40]. The P2P infrastructure of POOL is built on three types of peers: SPLASH, POND and the POOL Central, which are named in the "water" analogy according to their relative size, purpose and persistence level [45]. SPLASH is a desktop client communicating with other peers via the POOL P2P protocols. It provides metadata creation tools, a limited storage capacity for metadata records and the searching capability for the POOL network, through which to allow individuals to create metadata and maintain their collection of learning content. PONDs are bigger learning content repositories that incorporate several community repositories and are accessible using the POOL protocols as well as searchable based on the CanCore metadata standard. POND comes typically with a robust database support and a suite of tools for managing the learning content workflow. The POOL Central is a specialized peer connected to the network and a high speed Internet. The purpose of the application of the POOL Central is to replicate queries through the other POOL Central peers over the broadband connection thus enhance the reach of the network. The POOL Central does not necessarily have a storage capacity, although caching of records might be possible.

The POOL project is rather similar to Edutella in that both leverage the JXTA[30] P2P platform as the basis to build P2P network components. However, whereas Edutella implements a more advanced super peer based network topology using RDF to define the content part of the messages being exchanged, POOL implements its SPALSH-POND-POOL network topology using XML to specify message types, message formats and message routing rules. As RDF possesses more semantic richness than XML, the RDF-based Edutella is able to handle distributed searches across heterogeneous educational repositories by means of more advanced strategies, possessing more generality, flexibility and interoperability than the XML-based POOL. This implies that POOL can easily be integrated into Edutella through a specific protocol binding, but the reverse integration is generally impossible. Moreover, whereas RDF provides Edutella with the capability to naturally handle various learning resource metadata sets and their different bindings, currently POOL can only manage the CanCore metadata set and its XML binding. Although POOL promises to be able to work with different metadata sets by means of the XML queryspaces as wells as sets of tools supporting the transformation between different metadata schemas [45], such transformations cannot cover all popular learning resource metadata sets and may also lead to the loss of metadata information when a direct mapping between different schema elements does not exist. Actually, these drawbacks cannot easily be overcome in the XML-based POOL.

2.3.2. ARIADNE LOMster Project.

LOMster is the further development of the ARIADNE KPS. It is purposed to develop a P2P infrastructure to improve the ARIADNE KPS for sharing and reusing learning content [107]. The P2P design idea of LOMster is

---

[30] http://www.jxta.org

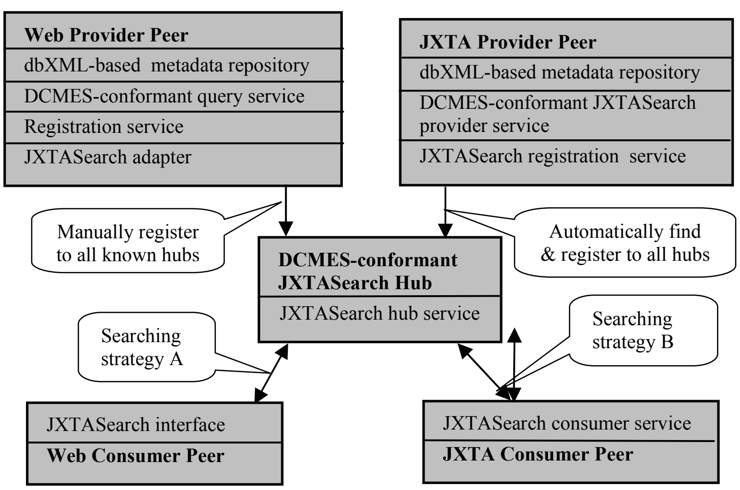
directly inspired by the lessons learned during the practical use of the ARIADNE KPS. It is intended to run on some lightweight user stations, thus directly migrates the design of some distributed file sharing P2P applications into the context of the E-Learning content management.

LOMster adopts a technical approach that is quite similar to the one adopted in the POOL project. It is built upon the JXTA platform and also adopts XML to define messages exchanged within the P2P network. In consequence, LOMster basically shares all features of POOL except that it is uniquely based on the IEEE LOM instead of the CanCore metadata standard for the learning content discovery. As a lightweight P2P application, LOMster is much weaker than POOL and Edutella. It does not yet address design and implementation issues regarding the distributed search across heterogeneous educational repositories and various learning resource metadata sets. As a P2P network based on the JXTA platform, LOMster and all its distributed functionalities can easily be integrated into POOL or Edutella.

2.3.3. JXTA Search Project

The JXTA Search is a P2P network designed for realizing both "wide" (the ability to access various devices on the Web) and "deep" (the ability to access various backend repositories) searches based on certain metadata schemas [112]. Like POOL and LOMster, it is an XML-based P2P infrastructure built upon the JXTA platform. The JXTA Search proposes a novel concept for searching XML metadata: the queryspace [60], which is a unique identifier for an abstract space over which a query travels. By means of the queryspace, together also with the open design of the searching algorithm, the JXTA Search can easily be extended to search any XML metadata sets including the XML binding of various learning resource metadata sets.

As the exploration to the design and implementation of Edutella, we extend the JXTA Search to support the search of DCMES RDF/XML binding metadata [88]. In Figure 2.8 we illustrate the P2P infrastructure of the JXTA Search prototype. As the DCMES, especially its RDF/XML binding [7], can be taken as the minimal interoperable basis of other learning resource metadata sets such as the LOM/IMS/SCORM, the JXTA Search prototype can also be used to search LOM/IMS/SCORM XML binding metadata taking advantage of a simple mapping approach [91].



Searching strategy A: For a Web consumer peer, hubs' addresses must be known before searching. The searching must employ hubs as mediators. After discovering, the results returned from both Web provider peers and JXTA provider peers must use hubs as mediators in order to reach the Web consumer peer.

Searching strategy B: For a JXTA consumer peer, the JXTASearch consumer service can find all available hubs and available JXTA provider peers. The searching can employ hubs as mediators. In this case the working scenario is similar to the searching strategy A. The searching can also directly be pointed to any available JXTA provider peers. After discovering, the results returned from JXTA provider peers can directly reach the JXTA consumer peer without the need of using hubs as mediators.

Figure 2.8 P2P infrastructure of the JXTA Search prototype used for searching DCEMS metadata

The kernel of the P2P infrastructure of the JXTA Search is the XML-based query routing protocol based on which four P2P services are implemented: the JXTA Search provider service (accept queries from a JXTA Search hub or JXTA Search consumer peer directly, and respond to the requestor), JXTA Search consumer service (send queries to a hub or JXTA Search provider peer directly, and await responses), JXTA Search registration service (send requests for registration to a JXTA Search hub and maintain the registration file for the provider), and JXTA Search hub service (perform routing of queries from consumers to providers). As the JXTA Search query routing protocol has two transport bindings respectively for the JXTA platform and Web, the JXTA Search can accommodate two types of providers, either pure JXTA peers or Web servers with the JXTA Search adapter. It can also support two types of consumers, either pure JXTA peers or Web browsers with HTTP client interface to the JXTA Search. Whereas pure JXTA peers can naturally interact with each other in the JXTA Search, Web-based provider peers and consumer peers must leverage the JXTA Search adapter and JXTA Search HTTP client interface in order to participate in the JXTA Search.

The network topology of the JXTA Search is quite similar to Edutella's super peer based topology. However, since the JXTA Search uses the XML syntax to manage P2P messaging, it is not comparable to the

RDF-based Edutella. Although the JXTA Search can theoretically accommodate any XML metadata sets, for some complex XML data schemas, the JXTA Search query routing protocol is too limited to represent complex queries, thus would have to be extended along the lines of current XML query languages. However, such sort of extension would have to touch the core implementation of the JXTA Search such as the query resolving and routing algorithms, etc. In addition, the current JXTA Search protocol does not yet define methods for multiple hub chaining, which would lead to the decrease of the whole searching efficiency with the increase of the number of the JXTA Search hubs. Generally, these critical issues cannot easily be resolved by simple extensions to the current JXTA Search.

Like POOL and LOMster, the XML-based JXTA Search can also be integrated into the RDF-based Edutella. Through a so-called Edutella/JXTA Search gateway, the JXTA Search "sub-network" could be integrated into Edutella as a whole.

2.3.4. Other Relevant Research

Apart from above projects closely related to Edutella, there exist also some other relevant research including:

OAI-PMH. OAI-PMH provides an application-independent interoperability framework for connecting DCMES metadata repositories based on metadata harvesting [65]. Although currently most of OAI-PMH implementations such as ARC [68] and DSpace[31] adopt the centralized client-server architecture, OAI-PMH itself does not preclude the P2P networking of its data providers (metadata repositories supporting OAI-PMH as a means of exposing metadata) and service providers (content providers using metadata harvested via OAI-PMH as a basis for building value-added services). In terms of the wrapper-like content provider integration architecture, Edutella is able to integrate any OAI-PMH conformant data providers and service providers into the P2P searching scope [3]. Such an integration approach is also applicable to other OAI-PMH like server-retrieval protocols such as Z39.50.

OKI. OKI is an open and extensible architecture that specifies how the components of an educational software environment communicate with each other and with other enterprise systems [80]. By focusing on the modularity of services and particular dimensions of interoperability, the OKI provides a modular development platform for building both traditional and innovative applications leveraging existing and future infrastructure technologies. As an architecture level protocol, OKI can provide beneficial guidance to the Edutella design, especially to the design of standard Edutella P2P services such as the authentication and authorization services, etc. It is expected that Edutella can achieve limited interactions with OKI-conformant systems, e.g., utilizing OKI-conformant services, integrating OKI-conformant educational repositories such as Stellar[32] and CARET[33] into Edutella through the OKI SQL OSID or OKI DBC OSID, etc.

OGSA. OGSA is built upon two other distributed computing paradigms: Grid and Web services, and further augments the two paradigms. It defines a uniform exposed service semantics (the Grid service), defines standard mechanisms for creating, naming and discovering transient Grid service instances, provides location transparency and multiple protocol bindings for service instances, and supports integration with underlying native platform facilities [39]. Although OGSA does not yet have practical implementations in the area of the E-Learning content management[34], it is considered as a promising architecture. As OGSA makes heavy use of Web services, it represents another important design style of the decentralized E-Learning content management infrastructure besides P2P. As both OGSA and Edutella leverages a SOA architecture, it is expected that the service layer interaction between two platforms is implementable. As a typical interaction scenario, Edutella P2P services could be integrated into an OGSA implementation through WSDL interfaces, and vice versa, OGSA services described by WSDL could be interpreted and utilized by Edutella. For other Web services based decentralized computing paradigms such as .Net[35], the similar interaction with Edutella can also be expected.

InfoQuilt. InfoQuilt is a framework for formulating complex information requests, which can capture the semantics of user's request involving multiple ontologies and support a form of knowledge discovery [108]. It is an agent-based system, having a more generic design purpose than Edutella but sharing a quite similar design approach. Currently InfoQuilt's semantic capabilities are being integrated with a P2P infrastructure towards realizing the so-called P2P Semantic Web, which is focused on correlating data from different data sources across heterogeneous data types or representations by means of several major Semantic Web technologies such as RDF, DAML+OIL [30] and DAML-S [31], etc. As InfoQuilt peers are more heavyweight than usual Edutella peers, it might be feasible to directly integrate an InfoQuilt peer into Edutella in terms of the wrapper-like

[31] http://dspace.org/
[32] http://stellar.mit.edu/
[33] http://www.caret.cam.ac.uk/
[34] It is being adopted by some ongoing projects such as Elegi (http://www.elegi.org)
[35] http://www.microsoft.com/net/

Edutella content provider integration architecture. However, the reverse integration might be rather difficult, especially when heterogeneity of Edutella peers is taken into account. Besides such sort of integration, some InfoQuilt tools designed for querying and managing ontologies are also expected to directly be used by usual Edutella peers.

On-To-Knowledge. Similar to InfoQuilt, On-To-Knowledge is focused on applying ontologies to improve the quality of knowledge management in large and distributed organisations[36]. Although On-To-Knowledge employs the client-server architecture and possesses a more generic design purpose than Edutella, its intensive exploration to sets of technologies regarding the connection of RDF metadata repositories makes it closely related to the Edutella design. It is expected that some output of On-To-Knowledge can directly be used in Edutella. In addition, the centralized RDF metadata repository of On-To-Knowledge could also be integrated into Edutella.

IRTL. IRTL is a middleware model design for addressing some of technical challenges associated with heterogeneous resource transactions in a P2P computing environment [50]. It provides an adaptive service that is able to integrate heterogeneous peer-based resources through a service interface independent of any particular systems. As a middleware platform, IRTL has a very broad design purpose, focussing on facilitating the discovery, valuation, negotiation, coordination, charging and exchange of resources across multiple distributed computing paradigms. It is expected that the research output of IRTL could provide guidance to the extension of Edutella, e.g., interacting Edutella with other distributed computing paradigms based on a unified middleware platform design.

PIAZZA. PIAZZA is a Semantic Web application that uses a P2P infrastructure to connect provider peers accommodating either XML or RDF metadata [44]. It proposes a language for mediating between peers that allows mapping simple forms of domain structure and rich document structure, as well as an algorithm for answering queries in PIAZZA that chains semantic mappings specific in the language. PIAZZA possesses a more generic design purpose than Edutella, addressing both XML and RDF metadata through defining a new language and using OWL ontologies [72]. However, regarding PIAZZA languages and searching algorithms, it seems a little bit proprietary than Edutella, in which we uniquely take RDF as the design basis and employ extensions to accommodate XML metadata thus greatly reduce the system complexity. Through the syntax-level transformation between RDF-QEL and PIAZZA language, the two frameworks are expected to be able to achieve somewhat interactions.

As part of an internationally cooperated academic project, Edutella is also closely related to lots of research conducted by our project partners in the context of the PADLR and ELENA project, including the KAON Ontology and Semantic Web Framework[37], Courseware Watchdog [101], UNIVERSAL brokerage platform, SCAM[38], Conzilla [73] and AMOS-II [97], etc. These research and their relationship with Edutella will partly be discussed in the thesis.

[36] http://www.ontoknowledge.org/index.shtml
[37] http://kaon.semanticweb.org/
[38] http://sourceforge.net/projects/scam/

3. Edutella: Design of an RDF-based E-Learning Content Management P2P Infrastructure

Edutella is an E-Learning content management infrastructure proposed based on two cornerstone technologies: the P2P platform JXTA and metadata language RDF. In section 3.1 we first describe several typical P2P usage scenarios in Edutella. In section 3.2 we present the P2P infrastructure design of Edutella based on the JXTA platform. In section 3.3 we introduce the implementation of the learning content discovery in Edutella based on RDF.

3.1. P2P Usage Scenarios in Edutella

There are three types of peers in Edutella: Edutella provider peers, Edutella consumer peers and Edutella super peers. In addition, as a P2P network based on the JXTA platform, Edutella also contains some peers that are commonly applied in JXTA-based applications, e.g., JXTA J2ME relay peers, JXTA rendezvous/relay super peers, etc. Within Edutella, all these peers can discover each other, self-organize into peer groups, discover peer resources, and communicate with each other across firewall/NATs. In Figure 3.1 we generalize several typical P2P usage scenarios in Edutella.

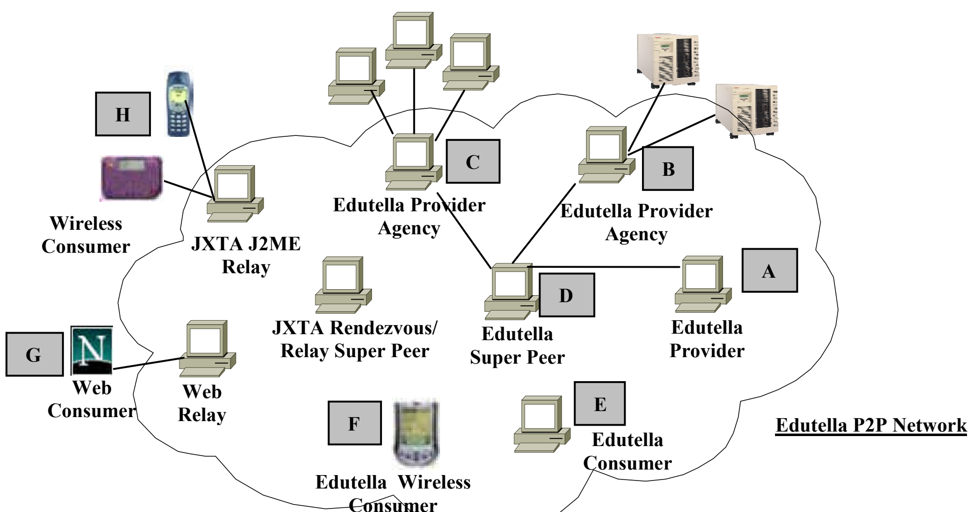


Figure 3.1 Typical P2P usage scenarios in Edutella

3.1.1. Usage Scenarios of Edutella Provider Peers

Edutella provider peers are content providers that accept queries and generate query results. Edutella mainly contains two types of provider peers. The first type is the simple provider peer (A), which typically possesses a learning content metadata repository and can naturally interact with other Edutella peers and P2P services. The second type is the provider agency peer, which acts either as the representative of a number of provider peers existing in a sub-network outside of Edutella (C), or as the representative of sets of search services typically provided by some popular search engines (B). Generally, the subordinating provider peers and search services of the Edutella provider agency are invisible to Edutella thus cannot directly interact with other Edutella peers or P2P services. They use the Edutella provider agency peer as the mediator in order to be integrated into Edutella and contribute their resources. The provider peers leveraging the first type of Edutella provider agencies (C) to achieve the integration include some centralized E-Learning portals such as Teachware on Demand, UNIVERSAL, etc., which can be integrated into Edutella in this way without the need of altering the existing architectures, and some P2P search applications such as the JXTA Search. The provider peers leveraging the second type of Edutella provider agencies (B) to achieve the integration include several "open" search engines such as Google[39]. As these search engines usually support a SOA-based computing paradigm, they can exchange distributed functionalities with Edutella (see also Chapter 4). In these usage scenarios, Edutella provider agencies can be viewed as Edutella provider peers on behalf of sets of unreachable edge-resource possessors outside of Edutella.

[39] http://www.google.com/

3.1.2. Usage Scenarios of Edutella Super Peers

Edutella super peers (D) serve as the entry points of usual Edutella provider peers. The backbone of Edutella is actually constructed by a limited number of Edutella super peers, which connect usual Edutella provider peers and take care of query distribution and query routing between Edutella consumer peers and provider peers based on routing indices [75]. For an Edutella super peer, it has to maintain two kinds of routing indices. The first kind of indices are super-peer/peer routing indices. Usual Edutella provider peers first need to register their discovery capabilities to Edutella super peers, e.g., what sort of resources they possess, what sort of queries they wish to response for this type of resource, etc., based on which super-peer/peer routing indices can be built. The second kind of indices are super-peer/super-peer indices, which store information about discovery capabilities of other Edutella super peers. When queries are sent to super peers, super peers either rout the queries to sets of usual Edutella provider peers that are expected to be the most appropriate candidates to answer these queries according to the super-peer/peer routing indices, or forward the queries to other suitable super peers according to the super-peer/super-peer routing indices. In Edutella, usual provider peers are not randomly assigned to a super peer, instead they are clustered to a super peer in terms of the similarity of their resources in order to reduce the amount of messages transferred in Edutella [75]. Besides the query distribution, Edutella super peers are also responsible for merging query results returned from usual Edutella provider peers or other super peers before sending them back to Edutella consumer peers.

3.1.3. Usage Scenarios of Edutella Consumer Peers

Edutella consumer peers utilize the Edutella query service to consume Edutella resources. In a broad sense, Edutella consumer peers also include other Edutella peers functioning based on the Edutella query service, e.g., Edutella annotation peers [76].

Edutella mainly contains three types of consumer peers. The first type is the simple consumer peer (E), which is the usual JXTA peer running the Edutella consumer service and can consume all Edutella resources through direct interactions with other Edutella peers. The second type is the Web consumer (G), which is typically any of Web-based applications outside of Edutella, e.g., Web browsers, Web services clients, etc. As this type of consumers cannot naturally be incorporated into Edutella, they can only consume Edutella resources through the Web relay peers, which can be viewed as Edutella consumer peers on behalf of sets of Web consumers outside of Edutella. The third type is the wireless consumer peer (F, H). In Edutella, these wireless consumers can further be divided into two types in terms of their footprints: the low-capable wireless device such as cell phones, two-way pagers, etc., and high-capable wireless device such as PDAs. Whereas all these wireless devices generally utilize CLDC and MIDP to participate in the Edutella network, their usage scenarios are somewhat different. For the high-capable wireless devices (F), they can achieve limited but direct interactions with other Edutella peers thus can directly consume Edutella resources. For the low-capable wireless devices (H), they have to use JXTA J2ME relay peers as the mediator to consume Edutella resources in order to overcome the nature drawbacks caused by their too small footprints. It is worth noting that wireless consumers are not expected to be able to realize the same functionalities as usual PC-based Edutella consumer peers. In Edutella, they can only utilize some Edutella P2P services specifically designed for them.

3.2. P2P Infrastructure Design of Edutella Based on JXTA

Edutella is aimed to provide access to distributed collection of digital resources through a P2P network. Regarding the design purpose, it is quite similar to some P2P applications for the distributed file sharing, which connect heterogeneous data sources using loosely coupled P2P topologies and support the efficient discovery of distributed resources based on certain P2P protocols. The applicable P2P infrastructures used to build distributed file sharing applications can be classified into two major types in terms of searching, discovering and routing strategies they adopt: the content-agnostic P2P infrastructure and content-based P2P infrastructure [13]. The content-agnostic P2P infrastructure adopted by some typical P2P systems such as Napster, Gnutella/LimeWire and FreeNet, etc., organizes peers not directly depending on resources they index, whereas the content-based P2P infrastructure adopted by some typical P2P applications such as Pastry [99], PAST [32], OceanStore/Tapestry [64], CAN [96] and Chord/CFS [103], etc., organizes peers directly based on the content they index. According to the P2P usage scenarios in Edutella, neither the content-agnostic P2P infrastructure nor content-based P2P infrastructure can directly be adopted to build Edutella.

In comparison to resources accommodated in the distributed file sharing P2P applications, learning content in Edutella are much more complicated and diverse. Instead of using simple metadata fields for their description, these learning content mostly describe themselves using RDF/RDFS metadata that may cross multiple learning resource metadata sets. For the content-based P2P infrastructures, as they often use searching and routing strategies based on the Hash-table, they work well while addressing resources that can be described by simple attributes, but are lack of efficiency or simply beyond their capabilities while addressing resources described by multiple attributes like in Edutella. Furthermore, as data are hashed in such networks, it becomes quite hard to apply Semantic Web technologies to manage learning content. For the content-agnostic P2P infrastructures, their network topologies, not only the symmetric ones like in Gnutella and FreeNet but also asymmetric ones like in Napster, cannot fully meet some design requirements provided by Edutella, e.g., the provider aggregation, firewall/NAT traversing and ubiquitous computing, etc. As the P2P infrastructure of Edutella has to accomplish more complicated functionalities than usual distributed file sharing P2P applications, it needs a new design to implement more advanced P2P network topologies and message routing mechanisms that can fit with RDF/RDFS metadata and meet some high-level design requirements of Edutella.

In this thesis, we propose to design the P2P infrastructure of Edutella based on the open source project JXTA, the P2P platform originally conceived by Sun Microsystems Inc. In comparison to usual P2P infrastructures for the distributed file sharing, the JXTA platform is located at a much lower level [43]. On the one hand, JXTA can ensure a widespread interoperability between various P2P applications. On the other hand, it can also easily be extended to build more advanced P2P systems with high performance network topologies and message routing mechanisms. Moreover, as JXTA has carefully addressed several critical design issues of P2P systems, e.g., platform/network/programming language independence, ubiquitous computing [43], etc., it can essentially meet some high-level design requirements of Edutella.

3.2.1. JXTA Platform and its Layering Architecture

At the highest abstraction level, JXTA is nothing more than six XML-based protocols that can easily be implemented on uni-directional links and asymmetric transports, enabling a ubiquitous connectivity in a P2P network. The design purpose of JXTA protocols is to be as pervasive as possible, and easy to implement on any transport. The six JXTA protocols have very low overhead, make few assumptions about the underlying network transport, and impose few requirements on the peer environment. They can be used to deploy a wide variety of P2P applications and services in a highly unreliable and changing network environment [111].

According to compliance requirements on implementations, the JXTA protocols can be divided into two categories: the JXTA core specification protocols and JXTA standard services protocols, as illustrated in Figure 3.2 [111].

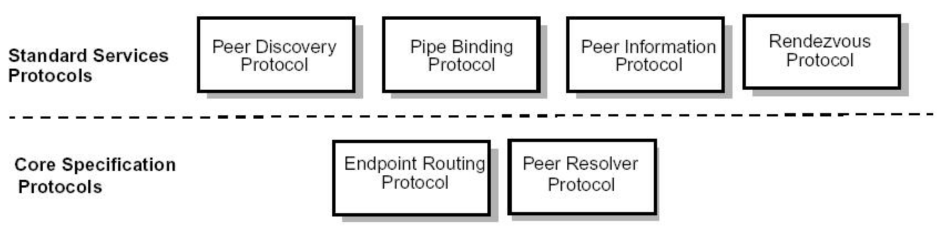


Figure 3.2 JXTA protocols

The JXTA core specification protocols define the functionality required by all P2P implementations. Any JXTA compliant P2P applications must implement all of the JXTA core specification protocols, although implementation of the JXTA core specification does not guarantee or even necessarily provide interoperability with other JXTA implementations. The JXTA core specification defines two protocols [104]:

- *Endpoint Routing Protocol (ERP):* ERP is used by a peer to discover a route to send a message to another peer.
- *Peer Resolver Protocol (PRP):* PRP is used by a peer to send a generic query to one or more peers, and receive a response or multiple responses to the query. It permits the dissemination of generic queries to one or more handlers within the peer group and to match them with responses.

Whereas the JXTA core specification defines the required components and behaviors for all JXTA implementations, the JXTA standard services protocols define some additional components that all implementation should provide in order to create a complete JXTA implementation. As the optional JXTA protocols and behaviors, implementations are not required to implement the JXTA standard services protocols. However, implementing these protocols will provide greater interoperability with other implementations.

The JXTA standard services protocols specification defines four protocols [104]:

- *Peer Discovery Protocol (PDP):* PDP is used by a peer to advertise its own resources and discover the resources from other peers. It uses PRP for sending and propagating discovery requests.
- *Pipe Binding Protocol (PBP):* PBP is used by a peer to establish a virtual communication channel (pipe) between one or more peers. PBP binds the two or more ends of the connection (pipe endpoints) through using PRP for sending and propagating pipe binding requests.

- *Peer Information Protocol (PIP):* PIP is used by a peer to obtain status information about other peers, such as state, uptime, traffic load, etc. PIP uses PRP for sending and propagating peer information requests.
- *Rendezvous Protocol (RVP):* RVP is used by a peer to subscribe or to be a subscriber to a propagation service. RVP allows messages to be sent to all of the listeners of the service and is used by PRP in order to propagate messages.

On the whole, the JXTA platform adopts a layering architecture as depicted in Figure 3.3 [43].

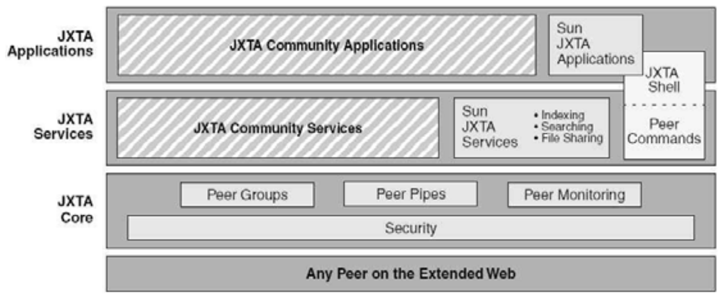


Figure 3.3 JXTA platform architecture

The JXTA platform is composed of three layers: the JXTA core layer, JXTA services layer and JXTA applications layer [43]. The JXTA core layer deals with peer establishment, communication management such as routing, and other low-level "plumbing". It consists of several building blocks that can be used by almost all P2P applications regardless of their intended users, platforms, devices and specific implementations. The JXTA services layer deals with some higher-level concepts such as indexing, searching and file sharing. These services, which make heavy use of the plumbing features provided by the JXTA core layer, are useful by themselves but also are commonly included as components in an overall P2P system. At the top of the JXTA platform is the JXTA applications layer, which accommodates various JXTA applications. Note should be paid that some JXTA features such as security manifest in all three layers and throughout a P2P system.

Edutella takes JXTA as the basis of its P2P infrastructure and strictly follows the layering architecture of the JXTA platform to design its functionalities. At the JXTA services layer, several major Edutella P2P services are implemented leveraging the JXTA core layer primitives, including [74]:

- *Edutella query service*: The service designed to support standardized query and retrieval of RDF metadata.
- *Edutella replication service*: The service designed to provide data persistence/availability and workload balancing while maintaining data integrity and consistency.
- *Edutella mapping service*: The service designed to translate between different metadata vocabularies to enable interoperability between different peers.
- *Edutella mediation service*: The service designed to define views that join data from different metadata sources and reconcile conflicting and overlapping information.
- *Edutella annotation service*: The service designed to annotate materials stored anywhere within the Edutella network.
- *Edutella clustering service*: The service designed to use semantic information to set up semantic routing and semantic clusters.

At the JXTA applications layer, sets of Edutella-specific tools and applications are developed based on Edutella P2P services, e.g., the Edutella annotation peer implementation [76], various interaction tools such as Courseware Watchdog and Conzilla, etc. Besides the use of Edutella P2P services, these applications also make use of other JXTA services and applications implemented by other JXTA sub-projects to implement some common P2P activities, e.g., the JXTA J2ME relay peers are used to connect Edutella wireless consumer peers, the JXTA relay peers are used to enable peers to traverse firewall/NAT, etc.

3.2.2. JXTA-conformant P2P Design of Edutella

As sets of low-level protocols, JXTA allows for different implementations. It provides several building-blocks for all JXTA-conformant P2P implementations at the JXTA core layer, including the peer, peer group, advertisement, service, pipe and message, etc. The JXTA-conformant P2P design of Edutella has to apply these JXTA core components to implement Edutella P2P functionalities.

Peer Group Based Resource Partition. The JXTA peer group represents a dynamic set of peers that have a common set of interests and share a common set of policies. In general, the JXTA peer group has three functionalities [43][111]:

- *To create secure domains for exchanging secure contents*. Peer groups form logical regions whose boundaries limit access to non-members. A peer group does not necessarily reflect the underlying physical network boundaries such as those imposed by routers and firewalls. Peer groups virtualize the notion of routers and firewalls, subdividing the network in secure regions without respect to actual physical network boundaries.
- *To create a scoping environment*. Peer groups are typically formed and self-organized based upon the mutual interest of peers. They serve to subdivide the network into abstract regions, providing an implicit scoping mechanism for restricting the propagation of discovery and search requests.
- *To create a monitoring environment*. Peer groups allow monitoring (traffic inspection, accounting, tracing) of peers for any purpose.

In Edutella, peer groups are mainly used to partition Edutella resources, especially to differentiate between various peer roles and Edutella P2P services. Each Edutella peer group contains P2P services specifically designed for implementing particular functionalities of peer roles. As peer groups can dynamically be created for scoping interactions between peers and matching their applications demands, the super peer based network topology of Edutella also makes intensive use of peer groups to implement the dynamic peer clustering.

Advertisement-based Resource Description. The JXTA core protocols rely on five types of advertisements to describe JXTA resources, including the peer advertisement, peer group advertisement, module class advertisement, module specification advertisement and module implementation advertisement. Besides, JXTA also standardizes advertisements for several other JXTA core resources such as the pipe, metering, route, content, rendezvous, peer endpoint and transport, etc. All these JXTA core advertisements can be sub-typed to add unlimited amounts of additional and richer metadata information to each resource description thus create new type of advertisements [104][111].

In Edutella, each P2P service is described through the module class advertisement, module specification advertisement and module implementation advertisement. As such a description strictly follows JXTA service definition style, all Edutella P2P services can directly interoperate with other JXTA-conformant P2P services or applications. Moreover, all proposed Edutella P2P services are designed as peer group services associated with particular Edutella peer groups, which are composed of a collection of cooperating instances of the service running on multiple peers. The advertisements of these peer groups are mostly extended to include metadata entries that describe sets of group-specific services to inform the existence and further provide invocation details of corresponding Edutella P2P services.

Rendezvous-based Discovery. The JXTA platform uses a universal resource binding mechanism called the resolver to perform all resolution operations, such as resolving a peer name into an IP address, binding a socket to a port, or searching and locating a service [111]. In JXTA, all resolution operations are unified under the simple discovery of one or more advertisements. The default JXTA resolver policy is proposed based on rendezvous peers, which are peers that have agreed to cache advertisement indices.

As the default JXTA rendezvous policy can only support a low-level discovery mechanism, in Edutella we extend this minimum discovery infrastructure to implement high-level discovery services. As the super peer based network topology of Edutella provides us with better knowledge of the content distribution, these high-level discovery services go beyond a simple advertisement-based searching. They can utilize RDF/RDFS metadata information to improve the searching performance.

Pipe-based Communication. JXTA pipes are virtual communication channels used for sending and receiving messages between JXTA services and applications [111]. They provide a virtual abstraction over the peer endpoints to provide the illusion of virtual in and out mailboxes that are not physically bound to a specific peer location. Pipes can connect one or more peer endpoints. The receiving end and sending end of a pipe is respectively referred to as the input pipe and the output pipe.

JXTA pipes offers two modes of communication [104]:

- *Point-to-point pipe*: A point-to-point pipe connects exactly two pipe ends with a unidirectional and asynchronous channel. No reply or acknowledgment operation is supported in point-to-point pipes. Additional information in the message payload like a unique ID may be required to thread message sequences. The message payload may also contain a pipe advertisement that can be used to open a new pipe to reply to the sender
- *Propagate pipe*: A propagate pipe connects one output pipe to multiple input pipes. Messages flow into the input pipe ends from the output pipe end. The propagate message is sent to all listening input pipe ends in the current peer group context. Propagate pipes can be implemented using point-to-point communication on transports that do not provide multicast, e.g., HTTP.

In Edutella, the communication between peers mainly leverage several advanced pipe services implemented on top of the JXTA core pipe services, including the bi-directional pipe service and secure pipe service. The pipe advertisements of these pipe services are extended to describe the valid set of messages to be sent or received through a pipe. These improvement to the JXTA core pipe services are realized using the JXTA 2.0 socket API.

Message-based Data Exchange. JXTA messages are data bundles that are passed between peers through JXTA pipes. JXTA uses a binary format to allow efficient transfer of both binary and XML payloads in messages [104]. Messages are formed as an ordered sequence of elements, each of which has a unique name, length and MIME-type. As JXTA messages allow to add or remove message elements, in Edutella we extend JXTA messages in terms of Edutella P2P services, which enables each service to manipulate message elements to isolate the different parts of a message for multi-purpose communication between Edutella peers.

3.2.3. Super Peer Based Network Topology of Edutella

Edutella adopts a super peer based network topology as depicted in Figure 3.4 [75].

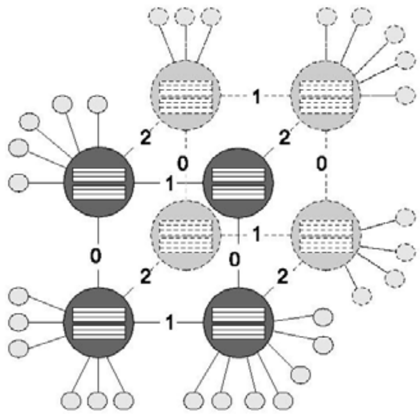


Figure 3.4 Super peer based network topology of Edutella

In the super peer based network topology, usual Edutella provider peers are connected to Edutella super peers in a star-like fashion. Super peers themselves are organized into a hypercube network topology in terms of the HyperCup protocol, which enables efficient query broadcast and guarantees non-redundant broadcast [100]. The hypercube network topology allows for *log2 N* path length and *log2 N* number of neighbors, where *N* is the total number of super peers in Edutella. As the hypercube topology is vertex-symmetric, it features inherent load balancing among all Edutella super peers. Between any two Edutella super peers, there exists a path length of *log2 N.* Any two distinct schemas can be reached within a short number of hops from each other.

Among Edutella super peers, the broadcast algorithm works as follows [75]. First, each connection is labeled with its dimension in the hypercube. A node invoking a broadcast sends the broadcast message to all its neighbors, tagging it with the edge label on which the message is sent. The nodes receiving the message restrict the forwarding of the message to those links tagged with higher edge labels. In this way the algorithm guarantees that exactly *N-1* messages are required to reach all nodes in the hypercube topology. The last nodes are reached after *log2 N* forwarding steps.

For a new Edutella super peer, it can join the hypercube network topology by asking any other already integrated super peers, which then carry out the peer integration protocol. In order to integrate a new super peer and maintain a hypercube topology, *O(log(N))* messages have to be sent. For the super peer based network topology, any number of super peers can be accommodated. Suppose some peers are "missing" for constructing a complete hypercube topology consisting of 2^d nodes in a d-dimensional binary hypercube, some super peers would occupy more than one position on the hypercube. When new super peers join the network, these new super peers will fill the gaps in the hypercube topology and possibly extend the dimensionality of the hypercube.

In order to ensure the routing efficiency in Edutella, each Edutella super peer maintains two types of routing indices constructed based on RDF/RDFS: super-peer/peer routing indices and super-peer/super-peer routing indices [75][77]. Super-peer/peer routing indices are used by Edutella super peers to characterize associated Edutella provider peers to ensure that queries are only forwarded to provider peers being able to answer them. These indices are updated when a peer connects to a super peer, and contain all necessary information about connected peers. Entries are valid only for a certain time. They will be deleted when the peers do not renew/update them regularly. Edutella provider peers notify super peers when their content changes in the way that they trigger an update of the index. Unlike other approaches based on the content indexing, such indices do not refer to individual content elements but to peers. Indices contain information about peers at four granularities: schema identifiers, schema properties, property value ranges, and individual property values

Similar to super-peer/peer routing indices, super-peer/super-peer routing indices are used by Edutella super peers to avoid broadcasting queries to all super peers. These super-peer/super-peer routing indices are essentially extracts and summaries from the local super-peer/peer indices, containing the same kind of information as super-peer/peer indices. However, unlike super-peer/peer indices, super-peer/super-peer indices only refer to the direct neighbors of a super-peer. Queries are forwarded to super peer neighbors based on super-peer/super-peer indices.

The update of super-peer/super-peer indices is based on the registration or update messages from connected provider peers. Whenever a super-peer/peer index changes, the change is propagated to all Edutella super peers using a reversed HyperCup broadcast. Whenever a super-peer/super-peer index stays unchanged after the update, the propagation stops. Note should be paid that super-peer/super-peer indices are not replicated versions of a central index, but rather parts of a distributed index similar to routing indices in TCP/IP networks.

On each Edutella super peer, both super-peer/peer routing indices and super-peer/super-peer routing indices can dynamically be constructed and updated based on various aggregation strategies of content indices of Edutella provider peers, as well as based on availability of super peers. If a super peer fails, its formerly connected peers must register with another super peer chosen at random. The respective super-peer/super-peer indices entries at other super peers are sequentially removed based on the dynamic optimizations [77]. In this way, the super peer based network topology can always ensure that super peers distribute queries only to the appropriate subset of provider peers thus can significantly reduce the workload of each Edutella provider peers. Such a P2P network can essentially provide better scalability than usual broadcast-based P2P networks.

The super peer based network topology in Edutella is implemented leveraging basic facilities of the JXTA platform. First of all, the HyperCup protocol is implemented in Edutella based on the JXTA core components. Additionally, several JXTA-conformant services are particularly designed for implementing Edutella super peers, e.g., services for peer registration and query routing table management, etc. These services are specified by the super peer service configuration, as depicted in Figure 3.5 [75].

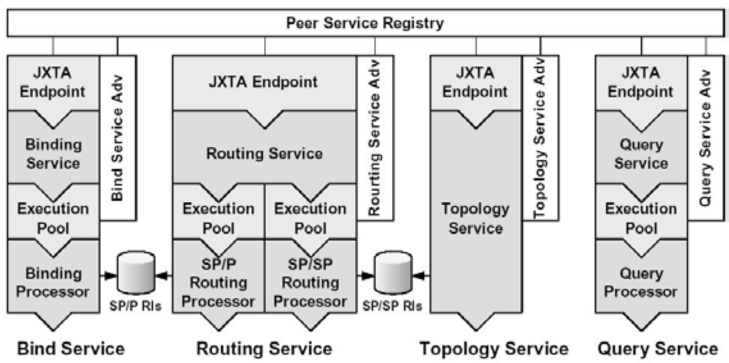


Figure 3.5 Super peer service configuration

The minimal super peer service configuration consists of at least four services [75]:

- *Super peer binding service*: The super peer binding service handles peer registration. Usual Edutella provider peers call this service with their self-description to establish the connection to a super peer. The binding service takes care of the hand-shaking process and is also responsible for updating super-peer/peer routing indices.
- *Super peer routing service*: The super peer routing service routes queries it receives to the appropriate provider peers and super peers according to the indices created by the binding and topology service.
- *Super peer topology service:* The super peer topology service takes care of the maintenance of the super peer network topology. It is responsible for keeping super-peer/super-peer routing indices up-to-date. If a new super peer bootstraps into Edutella, its topology service will connect to the topology service of another super peer, and the location of the new super peer in the network is negotiated. Afterwards, the new neighbors exchange super-peer/super-peer routing information.
- *Super peer query service*: The super peer query service provides a defined interface to issue new query requests within Edutella. These requests are then distributed via the routing service.

Each Edutella super peer service is composed of standard modules such as a JXTA endpoint handler, which manages service requests arriving via the JXTA infrastructure and some service-specific modules. Attached to each service is a JXTA-conformant service advertisement, which is published in Edutella on peer startup. The discovery of published services is achieved by the rendezvous-based discovery mechanism of JXTA.

3.3. Learning Content Discovery in Edutella Based on RDF

RDF is a framework for metadata standards with the primary target of providing a standardized way for creating and using self-developed metadata schemas to describe arbitrary resources on the Web [66]. Edutella adopts RDF as anther cornerstone technology of the standard-oriented E-Learning content management principally due to following capabilities of RDF:

- RDF uses the URI-based identifying mechanism, which can potentially describe any resources and state any relationships between these resources.
- RDFS offers a mechanism to define specific RDF vocabularies based on classes, properties and property constraints, allowing for the development of arbitrary metadata schemas to describe resources [15]. RDFS and its possible extensions such as DAML+OIL can enable the joint annotation about resources based on various metadata schemas, leading to RDF metadata repositories possibly containing more than one schemas. As RDFS is rather flexible and extendable, these metadata schemas as well as RDF metadata repositories can easily evolve over time.
- RDF can use distributed annotations for one and the same resource, which makes it very suitable for the construction of distributed metadata repositories. For RDF metadata repositories, it is not necessary to store all annotations of a resource on one server. The annotations about a resource can be allocated across multiple repositories possibly using different metadata standards/schemas.
- RDF statements are quite similar to a number of other formats for recording information, e.g., rows in RDBs, simple assertions in the formal logic, etc. This feature allows RDF to be used as a unified model for integrating data from various data sources.

In order to accomplish the RDF-based learning content discovery in a P2P setting, Edutella first needs to realize its query capability being able to query data across heterogeneous metadata repositories, across multiple metadata schemas, as well as across multiple schema languages. The accomplishment of such a capability depends greatly on an expressive query formalism building upon rule-like query languages. Although RDF metadata are basically graphs, the query languages based on the simple graph matching and sub-graph extraction are not sufficient for querying RDF, since these query languages cannot reason about the underlying semantics of RDF metadata, which are usually defined through several schema languages such as RDFS or OWL. Even if we have a query language that takes into account RDFS, such built-in support for exactly one fixed schema language is still insufficient, as it does not allow to query and combine RDF metadata expressed in multiple schema languages, which is one of the common query scenarios in the Edutella network [77]. In order to query RDF metadata in Edutella, we need a rule-like query language being able to support the definition of the semantics across multiple schema languages.

Besides, Edutella also needs to realize its data integration capability being able to integrate data from different educational repositories, which are described by various metadata schemas. The accomplishment of such a capability depends on query routing, processing and mediation strategies adopted in Edutella. As there co-exist multiple metadata schemas in Edutella and Edutella can only organize resources based on the local knowledge of each repository, it is generally impossible to realize the data integration functionality based on a single global metadata schema as in some traditional database systems [25][77]. In terms of the super peer based network topology, we need some new designs to implement the data integration functionality in Edutella.

3.3.1. RDF-QEL: the RDF Query Exchange Language

RDF-QEL is an RDF query language used to express and exchange queries and results in the RDF syntax. As a rule-like query language, RDF-QEL can internally be expressed in Datalog [41], although it externally uses RDF encoding to be transferred and exchanged within Edutella.

The design of RDF-QEL has taken into account the following criteria [78]:

- *Completeness*: RDF-QEL should not only be limited to simple conjunctive queries. It should also be able to represent other types of queries commonly applied in modern database systems.
- *Simplicity*: RDF-QEL itself should not contain any "syntactic sugar". To simplify implementation, it should be a bare-bone language for complex RDF queries.
- *Portability*: RDF-QEL should allow many kinds of implementations, either using traditional databases for RDF storage, or using RDF-specific storage systems. It should be relatively simple to translate RDF-QEL into other query languages.
- *Modularity*: Many features of RDF-QEL should be optional in order to allow for different kinds of implementations.

According to the query expressivity, RDF-QEL is defined to support five basic query complexity levels [78]:

- *Ruleless query*: A ruleless query is a query that does not contain rules.
- *Conjunctive query*: A conjunctive query is a query that contains a maximum of one rule per predicate. It does not contain any disjunctions.
- *Disjunctive query*: A disjunctive query is a query that may contain several rules for each predicate, but does not allow for queries to be recursive in any sense.
- *Linear recursive query*: A linear recursive query is a query that contains recursive predicates, but the recursion is linear. Such a query can theoretically be translated into SQL99.

- *General recursive query*: A general recursive query is a recursive query that is not linear recursive. Such a query requires the equivalent of a Datalog or Prolog processor to be executed.

Corresponding to above query complexity levels, the simplest RDF-QEL queries can be expressed as un-reified RDF graph, whereas the more complex queries are more expressive than RDF itself thus have to be expressed using reified RDF statements. Moreover, unlike usual query languages like Datalog, RDF-QEL also defines how query results are returned depending on the originating query. It specifies an ordered list of variables in a query, which are used to create the query result set. When a query is executed, the variables occurring in the query body are bound to values in the way that the query evaluates to be true if the bindings are substituted for the corresponding variables in the query body. A query result is then constructed by an ordered list of values taken from a complete variable binding.

The RDF-QEL query and query result can uniformly be represented in the RDF syntax regardless of different query complexity levels. In Figure 3.6 we illustrate the UML model diagram of RDF-QEL [78]. This model forms the basis for constructing the RDF syntax for RDF-QEL. The detailed description about the semantics of RDF-QEL queries and results can be found in the RDF schema of RDF-QEL [78].

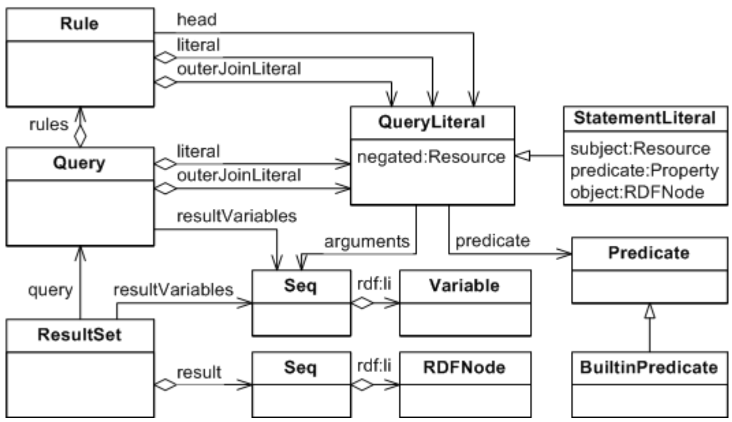


Figure 3.6 UML model diagram of RDF-QEL

In order to describe the RDF syntax of RDF-QEL queries and results, in the following we take a simple RDF knowledge base as an example to construct RDF-QEL queries against it. In Figure 3.7 we illustrate the RDF graph of the example knowledge base [74].

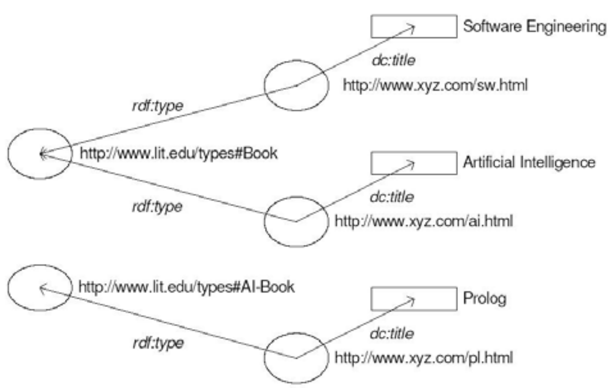


Figure 3.7 RDF graph of the example knowledge base

This RDF graph can be serialized in the XML syntax as:

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<rdf:RDF  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:lib="http://www.lit.edu/types#"  xmlns:dc="http://purl.org/dc/elements/1.1/">

<lib:Book about="http://www.xyz.com/sw.html">
<dc:title>Software Engineering</dc:title>
</lib:Book>

<lib:Book about="http://www.xyz.com/ai.html">
<dc:title>Artificial Intelligence</dc:title>
```

```
</lib:Book>

<lib:AI-Book about="http://www.xyz.com/pl.html">
<dc:title>Prolog</dc:title>
</lib:AI-Book>

</rdf:RDF>
```

In order to query this example knowledge base, we propose an example RDF-QEL query, which can be read in plain English as: *return all resources that are a book having the title of "Artificial Intelligence", or that are an AI book*. In terms of the RDF-QEL syntax, this example query can be represented as:

```
<?xml version='1.0' encoding='ISO-8859-1'?>

<!DOCTYPE rdf:RDF [
  <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
  <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
  <!ENTITY dc 'http://purl.org/dc/elements/1.1/'>
  <!ENTITY qel 'http://www.edutella.org/qel#'>
  <!ENTITY lit 'http://www.lit.edu/types#'>
]>

<rdf:RDF xmlns:rdf="&rdf;" xmlns:rdfs="&rdfs;" xmlns:dc="&dc;"
  xmlns:qel="&qel;" xmlns:lit="&lit;">

<qel:Query rdf:ID='AI_Book_Query'>
  <qel:rule rdf:resource='#r1'/>
  <qel:rule rdf:resource='#r2'/>
  <qel:literal rdf:resource='#l1'/>
  <qel:resultVariables>
        <rdf:Seq>
              <rdf:_1 rdf:resource="#X"/>
        </rdf:Seq>
  </qel:resultVariables>
</qel:Query>

<qel:Variable rdf:ID="X"/>

<qel:Rule rdf:about='#r1'>
  <qel:head>
    <qel:QueryLiteral rdf:about='#h1'>
      <qel:predicate rdf:resource='#aibook'/>
      <qel:arguments>
        <rdf:Seq>
          <rdf:_1 rdf:resource='#X'/>
        </rdf:Seq>
      </qel:arguments>
    </qel:QueryLiteral>
  </qel:head>
  <qel:literal>
    <qel:StatementLiteral rdf:ID='st2'>
      <rdf:subject rdf:resource='#X'/>
      <rdf:predicate rdf:resource='&rdf;type'/>
      <rdf:object rdf:resource='http://www.lit.edu/types#Book'/>
    </qel:StatementLiteral>
  </qel:literal>
  <qel:literal>
    <qel:StatementLiteral rdf:ID='st3'>
      <rdf:subject rdf:resource='#X'/>
      <rdf:predicate rdf:resource='&dc;title'/>
      <rdf:object>Artificial Intelligence</rdf:object>
    </qel:StatementLiteral>
  </qel:literal>
</qel:Rule>

<qel:Rule rdf:about='#r2'>
  <qel:head>
    <qel:QueryLiteral rdf:about='#h2'>
      <qel:predicate rdf:resource='#aibook'/>
      <qel:arguments>
        <rdf:Seq>
          <rdf:_1 rdf:resource='#X'/>
        </rdf:Seq>
      </qel:arguments>
    </qel:QueryLiteral>
```

```
  </qel:head>
  <qel:literal>
   <qel:StatementLiteral rdf:ID='st5'>
    <rdf:subject rdf:resource='#X'/>
    <rdf:predicate rdf:resource='&rdf;type'/>
    <rdf:object rdf:resource='http://www.lit.edu/types#AI-Book'/>
   </qel:StatementLiteral>
  </qel:literal>
</qel:Rule>

<qel:QueryLiteral rdf:ID='l1'>
 <qel:predicate rdf:resource='#aibook'/>
 <qel:arguments>
  <rdf:Seq>
   <rdf:_1 rdf:resource='#X'/>
  </rdf:Seq>
 </qel:arguments>
</qel:QueryLiteral>

</rdf:RDF>
```

Corresponding to the example knowledge base, the query results of this example RDF-QEL query would seem as:

```
<rdf:RDF
  xmlns:qel='http://www.edutella.org/qel#'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>

   <qel:ResultSet>
     < qel:query rdf:resource='AI_Book_Query'/>
        < qel:result>
            <rdf:Seq>
               <rdf:li rdf:resource='http://www.xyz.com/ai.html'/>
             </rdf:Seq>
         </ qel:result>

        < qel:result>
            <rdf:Seq>
               <rdf:li rdf:resource='http://www.xyz.com/pl.html'/>
             </rdf:Seq>
         </ qel:result>
    </ qel:ResultSet>

</rdf:RDF>
```

3.3.2. ECDM: Integrating Heterogeneous Data Sources into Edutella

Edutella contains different data sources that are highly heterogeneous in terms of the resources, functionalities and services they offer. Whereas RDF-QEL provides us with a common query interchange format within Edutella, we still need a common data model in Edutella to support the internal representation of RDF-QEL queries and query results. Based on this common data model, heterogeneous data sources can be integrated into Edutella in terms of a unique interface.

Since RDF-QEL uses Datalog expressions to define the semantics of the query and query result, the Edutella common data model ECDM is proposed based on Datalog. As illustrated in Figure 3.8, Edutella uses a wrapper-like architecture [81] to achieve the data model translation based on ECDM.

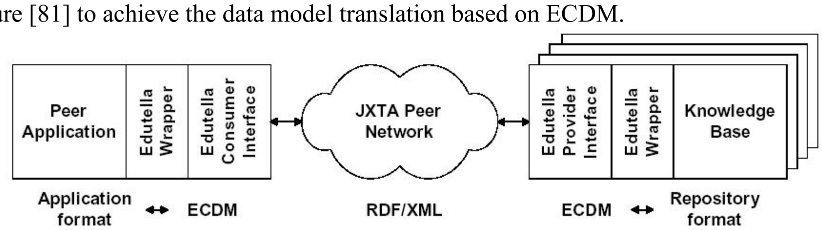


Figure 3.8 Wrapper-like architecture for data model translation based on ECDM

Datalog is a non-procedural query language based on Horn clauses without function symbols [41]. It shares with RDBs and also with RDF the central feature that data are conceptually grouped around properties. In consequence, Datalog queries can easily be mapped to relations and relational query languages like relational algebra or SQL. In terms of relational algebra, Datalog is capable of expressing selection, union, join and projection thus is a relationally complete query language. Additional features of Datalog also include transitive closure and other recursive definitions.

The Datalog data model is fully compliant with RDF's binary relational data model thus can represent any RDF knowledge bases and queries. In RDF, any statement is considered to be an assertion. Therefore, an RDF knowledge base can be viewed as a set of ground assertions either using binary predicates, or using ternary statements *s(S,P,O)*, if we include the predicate as an additional argument. Taking the RDF knowledge base described in section 3.3.1 as an example, it can be represented in Datalog as:

title(http://www.xyz.com/ai.html,'Artificial Intelligence).
type(http://www.xyz.com/ai.html,Book).
title(http://www.xyz.com/sw.html,'Software Engineering').
type(http://www.xyz.com/sw.html,Book).
title(http://www.xyz.com/pl.html,'Prolog').
type(http://www.xyz.com/pl.html,AI-Book)

As RDF-QEL is proposed based on the Datalog constructs, the example RDF-QEL query can also be translated into Datalog using the binary surface representation.

aibook(X) :- title(X, 'Artificial Intelligence'), type(X, Book).
aibook(X) :- type(X, AI-Book).
?- aibook(X).

Since the example query is a disjunction of two purely conjunctive sub-queries, its Datalog representation is composed of two rules with identical heads. The literals in the rules' bodies directly reflect RDF statements with their subjects being the variable *X* and their objects being bound to constant values such as "Artificial Intelligence". Literals used in the head of rules denote derived predicates. The example query expression *aibook(X)* asks for all bindings of *X*, which conform to the given Datalog rules and the knowledgebase to be queried. In terms of the example RDF knowledge base, the query results of the example query can be represented in Datalog as:

aibook(http://www.xyz.com/ai.html)
aibook(http://www.xyz.com/pl.html)

In Edutella, the Datalog-based ECDM is shared by all data sources and provides the common data view of underlying heterogeneous repositories. As illustrated in Figure 3.9, Edutella adopts a wrapper-like architecture to integrate heterogeneous data sources.

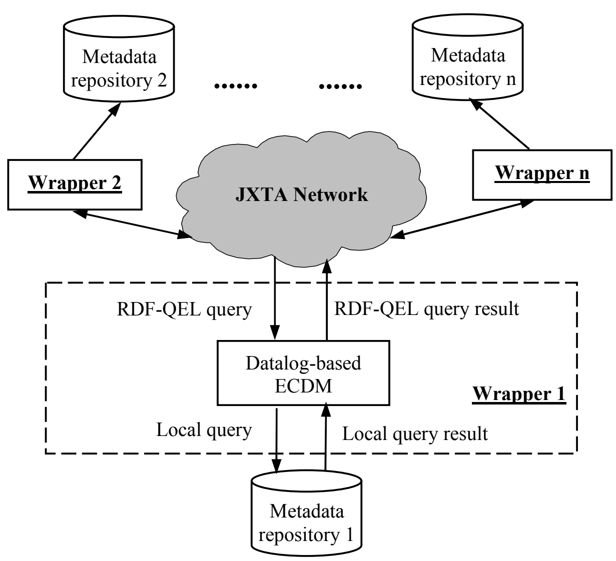


Figure 3.9 Edutella content provider integration architecture

For each wrapper program, it is on the one hand responsible for generating the ECDM-based common data view of heterogeneous data sources, on the other hand, it is also responsible for translating RDF-QEL queries into the local query language, and vice versa, transforming local query results into RDF-QEL query results. As ECDM uses Datalog as the internal query language, the RDF-QEL queries coming from the Edutella network are first translated into Datalog representations, which are sequentially translated into different local query languages practically operating the knowledge base. Similarly, after query results are available, local query results are first transformed into Datalog representations, which are sequentially transformed into RDF-QEL results. These RDF-QEL results are then transferred and exchanged in Edutella in the pure RDF syntax.

For different types of Edutella provider peers, we have to develop different kinds of wrapper programs in order to achieve the integration. As the RDF data model is compliant with the ECDM data model, RDF metadata repositories are the most suitable type of repositories for the integration. Until the end of 2003, Edutella has successfully integrated several RDF metadata repositories built upon various back-end systems, including the Oracle RDB using SQL as the local query language [116], the OODB ConceptBase using O-Telos as the local query language [117], flat file knowledge base using Jena RDQL as the local query language, and MINERVA using Prolog as the local query language [42], etc. As an extension to Edutella, in chapter 5 we will also discuss some more complex approaches for integrating XML metadata repositories into Edutella.

3.3.3. Distributed Query Routing, Processing and Mediation Based on RDF

RDF-based Distributed Query Routing. As we have mentioned, Edutella super peers leverage super-peer/peer and super-peer/super-peer routing indices to manage distributed query routing. On each super-peer, elements used in a query are matched against both routing indices in order to determine Edutella provider peers or other super peers to which the query should be sent. As an RDF-based P2P network, Edutella uses the RDF syntax to build the routing indices on each super peer. These indices contain the information about other peers or super peers at four granularity levels: schema identifiers, schema properties, property value ranges and individual property values [75][77].

- *Schema index*: Edutella provider peers may support different schemas, which can uniquely be identified through the namespace URI. The schema routing index contains schema identifier as well as peers supporting this schema. Queries are forwarded only to peers that support the schemas used in the query.
- *Property/sets of properties index*: Edutella provider peers may choose to use only parts of (one or more) schemas, e.g., certain properties, to describe their content. This is more often applied for data stores using semi-structured data. In this kind of index, super peers use the properties uniquely identified by the schema ID plus property name, or sets of properties to describe peers.
- *Property value range index*: For properties that contain values from a predefined hierarchical vocabulary, Edutella super peers may use an index that specifies taxonomies or part of a taxonomy for properties. In the context of Edutella, it is a common case that quite a few applications use standard vocabularies or ontologies. Therefore, Edutella peers could be characterized by their value ranges.
- *Property value index*: For some properties it may be advantageous to create value indices to further characterize Edutella peers. Such an index contains only properties that are used very often compared to the rest of the data stored on the peers.

In general, each RDF-QEL query can be decomposed into above four granularity levels thus can be routed to appropriate provider peers or other super peers in terms of the schema index, property/set of properties index, property value range index, and property value index. However, such a routing still has the problem that most queries must be broadcast if provider peers are arbitrarily distributed within the super peer backbone network of Edutella. In order to avoid broadcast as much as possible, in Edutella we employ three clustering algorithms: the ontology-based clustering, rule-based clustering and query-based clustering, to ensure the efficiency of the distributed query routing [69][77]. In the ontology-based clustering, clusters are created according to the ontology-level similarity of provider peers, which themselves are represented by Edutella super peers and organized into the hypercube network topology. In the rule-based clustering, clusters represented by Edutella super peers group usual provider peers with equal properties, e.g., some static properties such as specific query and result schemas, specific domain/IP address ranges, or some more dynamic properties such as a minimum number of resources at a peer, average answer time or average number of results, etc. In the query-based clustering, super-peer/peer and super-peer/super-peer routing indices on each Edutella super peers are further extended with additional frequency information about queries, which allows to adapt the network topology and peer clustering based on the query frequency information.

RDF-based Distributed Query Processing. In Edutella, Edutella super peers are responsible for providing query optimisation and query processing capabilities based on super-peer/super-peer and super-peer/peer routing indices they maintain. As most of RDF-QEL queries in Edutella have to be resolved by multiple provider peers, the Edutella super peers dynamically generate a query evaluation plan using the allocation schema provided by the index structures, based on which part of the query are locally executed by connecting provider peers whereas the remainders of the query are pushed to the neighbour super peers thus executed by other Edutella provider peers [16]. During the generation of the query plan, the remainders of the query that cannot locally be executed are further optimised according to statistics of the input data, the network topology and provider peer clusters.

This leads to a dynamic distribution and expansion of query plans, which enables to place query operators close to the suitable Edutella provider peers thus utilize distributed data sources more effectively [16][77].

RDF-based Distributed Query Mediation. In Edutella, the distributed query mediation between different schemas is supported at the super peer level. Each Edutella mediator provides coherent views of the data in the data sources by performing semantic reconciliation of data representations defined by different local data schemas. As there exists no unique global schema for describing data sources in Edutella, the distributed query mediation between heterogeneous local data schemas has to depend on local transformation mechanisms and rules. So far Edutella has two approaches for achieving the distributed query mediation: either using explicit mediation peers such as AMOS-II [61], or leveraging rule-based mediation facilities implemented on each Edutella super peer. Both approaches evaluate and mediate RDF-QEL queries based on the schema-level information defined by RDFS or its extensions [77].

4. Interacting Edutella/JXTA with Web Services

The initial use cases of Edutella/JXTA are defined and implemented in the context of a pure P2P setting where all Edutella participants are JXTA peers interacting with each other via various Edutella/JXTA P2P services. In this chapter we propose an approach for interacting Edutella/JXTA with Web services with the purpose of exchanging distributed functionalities between the two platforms. In section 4.1 we first provide an overview of the interaction between P2P and Web services. In section 4.2 we discuss several challenges we have to tackle while achieving the interaction between both. In section 4.3 we detail the technical implementation of the service layer interaction between the two platforms in terms of two typical interaction scenarios.

4.1. Interaction between P2P and Web Services: Overview

P2P and Web services are considered as two promising technologies for distributed computing. However, with regard to high-level design goals, the two technologies are originally proposed to address different problem domains. Whereas P2P usually has a broader definition, which is generally applied to a wide range of technologies that can greatly increase the utilization of information, bandwidth and computing resources at the edge of the Internet with the purpose of extending/overcoming the current client-server based architectures on the Web [38][43], Web services are more intended to promote interoperability and extensibility among various applications, platforms and frameworks by means of externalising and modularising application functionalities as sets of interoperable services on the Internet [12][63]. At present the whole P2P realm is not yet solidified enough to support a set of common-agreed standards. In contrast, the Web services world is relatively unified, possessing a clearly defined conceptual model and sets of supporting industrial standards.

4.1.1. Web Services Conceptual Model and the Layering Architecture

A Web service is an interface that describes a collection of operations that are network accessible through standardized XML messaging [12]. It is described by a service description, which covers all the details necessary to interact with the Web service including message formats, transport protocols and location. The Web services interface hides the implementation details of the Web service, allowing it to be used independently of the hardware or software platform on which it is implemented, and also independently of the programming language in which it is written. This leads to several notable features of Web services applications, e.g., loosely coupled, component-oriented, cross-technology [63], etc.

The Web services conceptual model is based upon the interactions between three types of entities: the service provider, service requestor and service registry, as illustrated in Figure 4.1 [63].

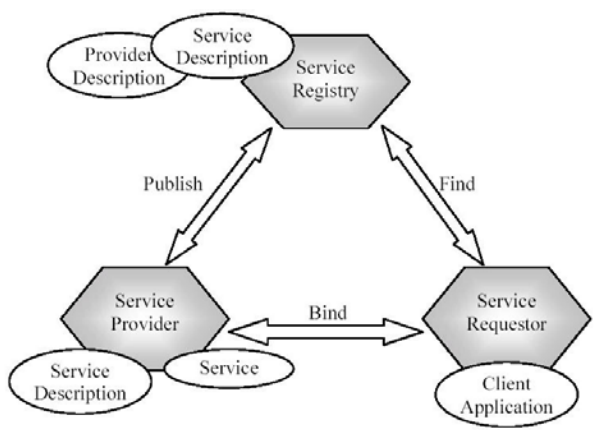


Figure 4.1 Web services conceptual model

The service provider is the platform that hosts access to the Web service. The service requestor is the application that is looking for and invoking or initiating an interaction with a Web service. The service registry is a searchable registry of service descriptions where service providers can publish service descriptions and provider descriptions. Via the service registry, service requestors can find Web services and obtain information for binding services during development (static binding) or during execution (dynamic binding) [63]. For statically bound service requestors, the service registry is an optional role in the Web services architecture, since a service provider can also send the description directly to service requestors. Besides, service requestors can obtain a service description directly from other sources instead of a service registry. In contrast, for dynamically bound service requestors, the service registry is indispensable and has to be accessed on each execution.

The Web services entities interact with each other through three types of operations [63].

- *Publish*: The service provider has to publish the service description to the service registry to make it possible for a service requestor to find a Web service and access it.
- *Find*: The service requestor retrieves a service description by inquiring the service registry. The find operation can occur in two different lifecycle phases for the service requestor: either at design time to retrieve the service interface description for the static binding, or at run-time to get the service's binding and location description for the dynamic binding.
- *Bind*: The service requestor invokes or initiates an interaction with the service at run-time using the binding details in the service description to locate, contact and invoke the service.

In the Web services conceptual model, these entities and operations act upon three major Web services objects: the Web services software module, Web service description and client application. In a typical working scenario, a service provider hosts a Web services software module implementing service functionalities. The service provider defines a service description for the Web service and publishes it to a service requestor or service registry. The service requestor uses a "find" operation to retrieve the service description locally or from the service registry and then uses the service description to bind with the service provider and invoke or interact with the Web service software module through the client application.

As envisioned by the W3C Web services architecture WG, Web services can be investigated in terms of a layering architecture. In Figure 4.2 we illustrate the Web services architecture [12].

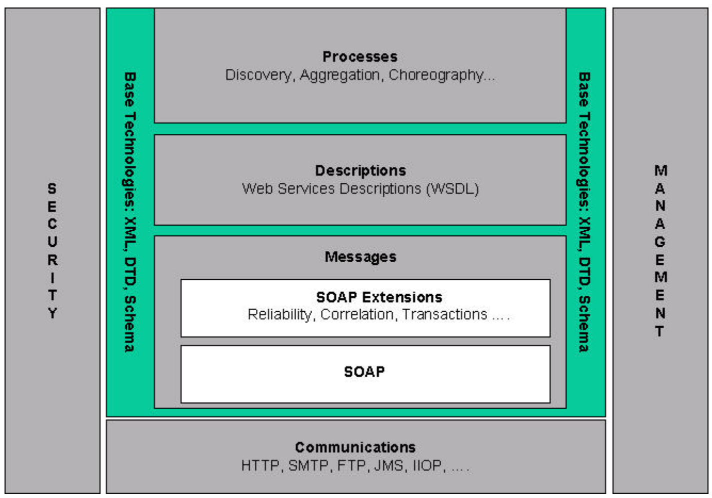


Figure 4.2 Web services architecture

First and foremost, the Web services architecture adopts XML as the base technology. It depends on sets of XML standards such as the XML Schema [10][110], XML Base [70], etc., to manage the message exchanging in the platform. These XML-based messages are exchanged over the communication layer, which encompasses a wide variety of communication mechanisms such as HTTP, SMTP, FTP, JMS, etc. On the top of the communication layer is the message layer, which employs SOAP [14] as the key messaging technology to provide a standard way of packaging the messaging information in a protocol neutral way. Since interoperability across heterogeneous systems requires a mechanism to allow the precise structure and data types of the messages to be commonly understood by Web services providers and requesters, the Web services architecture adopts WSDL [26] as the key at the description layer to support the precise description of Web services as well as messages to be exchanged between them. Finally, beyond the description of Web services, the Web services architecture envisions a variety process descriptions at the process layer, including the process of discovering service descriptions that meet specified criteria, the process of describing multipart and stateful sequences of messages, the aggregation of processes into higher-level processes [12], etc. Besides these essential layers, the Web services architecture also defines functionalities for security and management. These functionalities manifest across all essential layers of the Web services architecture.

According to the SOA, the function layers of the Web services architecture can further be generalized into a wire-service-application layer model. Since a Web service is usually defined as being network-accessible via SOAP and represented by a service description [12], the bottom three layers of the Web services architecture: the communication layer, message layer and description layer are essentially required to provide or use any Web services thus can be viewed as the interoperable basis of the whole Web services world. Among them, the communication layer and message layer can jointly be viewed as the wire layer of the Web services architecture, taking SOAP as the layer cornerstone to manage the XML-based message exchanging in the platform. The description layer can be viewed as the service layer, using WSDL as the layer cornerstone to describe and expose Web services functionalities. The process layer of the Web services architecture can be viewed as the application layer, depending on UDDI [8] and some other advanced Web services standards, e.g., WSFL [67], Web Services Coordination [21], Web Services Invocation Framework [5], etc., to build practical Web services applications. It is worth noting that unlike some other Web services architectures proposed by some leading Web services vendors [63], W3C intentionally categorizes some UDDI-centred Web service process layer functionalities such as the service publication, service discovery, etc., into the application layer instead of the service layer of the Web services architecture. This is because that according to the Web services definition from W3C, these process layer functionalities are not mandatory for every Web services applications, e.g., the Web services discovery can also be realized through other approaches instead of the UDDI-based centralized discovery mechanism.

4.1.2. Interaction between P2P and Web Services

From the basis, P2P and Web services are complementary technologies. While Web services provide a universal information architecture to solve the functional integration problem, P2P provides a distributed network architecture that takes us directly and securely to data sources at the edge of the Internet. The interaction between the two platforms may potentially combine advantages of both, leading to a promising distributed computing architecture converging P2P and Web services. Since currently P2P technologies are not yet unified enough, in the following we take Edutella/JXTA as an example P2P platform to investigate the interaction between P2P and Web services.

As both Edutella/JXTA and Web services employ a layering architecture to implement a SOA, the interaction between the two platforms may occur at different layers. At the wire layer, Edutella/JXTA and Web services may directly exchange low-level messages. At the service layer and application layer, the two architectures may directly exchange distributed functionalities. Due to considerable wire layer incompatibility, the direct wire layer interaction between Edutella/JXTA and Web services is rather difficult to achieve before JXTA becomes SOAP-aware. In contrast, the service layer and application layer interaction are expected to be able to overcome the wire layer incompatibility thus enable the exchange of distributed functionalities between the two platforms without shaking their individual basis. As the service layer interaction is the prerequisite to the accomplishment of the application layer interaction, in this chapter we focus on achieving the service layer interaction between Edutella/JXTA and Web services in terms of two typical interaction scenarios. The first interaction scenario is to expose existing Edutella/JXTA P2P services as Web services in order to enable Edutella/JXTA functionalities to be integrated into other distributed computing paradigms built on Web services. The second interaction scenario is to integrate Web services enabled content providers such as a .Net driven E-Learning “portal” into Edutella/JXTA in order to extend the reach of Edutella/JXTA. Actually, these two scenarios represent the principal needs for the interaction between the two platforms.

4.2. Interacting Edutella/JXTA with Web Services: Challenges

As far as the current application status of Edutella/JXTA and Web services is concerned, they belong to two isolated realms, adopting different wire protocols, using incompatible functionality descriptions, and applying different mechanisms to manage distributed functionalities on each platform. In order to achieve the interaction between Edutella/JXTA and Web services, we have to face four sides of challenges.

First, Edutella/JXTA and Web services adopt different entity identification systems and also use different mechanisms for searching and discovering entities and entity functionalities. This makes it rather difficult to mutually locate entities and entity functionalities in order to achieve the direct interaction.

The Web services architecture is composed of three types of entities: the service provider, service registrar and service requestor, which are clearly distinguished and interact with each other through the publish, find and bind operations. These entities and operations work in concert to provide a loosely coupled computing paradigm, whose manifestation is described in a set of standards, most notably SOAP, UDDI and WSDL. Within the Web services world, the service provider, service registrar and service requestor leverage the UDDI-based centralized discovery mechanism to locate each other and further achieve various functionality interactions.

Unlike in Web services, in Edutella/JXTA there is no clear identification between the service provider, service requestor and service registrar. An Edutella/JXTA peer typically implements one or more of JXTA protocols, and may play multiple roles simultaneously. Since all Edutella/JXTA peers share the same JXTA core implementation module, they can direct interact with each other based on JXTA platform facilities. This implies that there exists no direct functional entity mapping between a Web services node and an Edutella/JXTA peer, and also implies that it is impossible for a Web services node to achieve a direct interaction with Edutella/JXTA without conducting the JXTA bootstrapping process in advance.

With regard to the entity and entity functionality discovery, Edutella/JXTA mainly depends on the decentralized discovery mechanism for searching and locating peers and P2P services, although it does not exclude other discovery mechanisms such as centralized ones. However, as the current decentralized discovery mechanism of Edutella/JXTA is implemented based on the JXTA Peer Discovery Protocol [104], which is not compatible with UDDI, Edutella/JXTA P2P services cannot be directly advertised or discovered in the Web services world. Likewise, the UDDI-based centralized Web services registry cannot directly be integrated into Edutella/JXTA and used by Edutella/JXTA peers for discovering and binding Web services.

Second, Edutella/JXTA and Web services adopt different transport and message protocols for the communication within each realm. This wire layer incompatibility constitutes a big obstruction to the functionality interaction between the two platforms.

With the focus on decentralization and ubiquitous computing, the JXTA platform neither mandates the transport protocol nor mandates on how the messages are specified or propagated [43]. In contrast, although theoretically Web services also do not mandate the transport protocol, since they principally depend on SOAP for the XML-based messaging, they practically use HTTP for transport. This implies that there might exist the transport layer incompatibility between a Web services node and an Edutella/JXTA peer, which could lead to the communication blockage during the interaction. In addition, with regard to the message protocol, although the current JXTA specification adopts XML format to define messages, the definition is not compliant with SOAP. This makes it rather difficult to achieve the direct SOAP RPC between Edutella/JXTA and Web services.

Third, Edutella/JXTA and Web services adopt different formats to describe distributed services. This makes it impossible to directly exchange service descriptions between the two platforms and further interact these services based on the service descriptions.

Web services use WSDL to describe services and their invocation details, whereas Edutella/JXTA uses XML structured service advertisements to describe P2P services and their invocation details. In Edutella/JXTA, a P2P service is generally described by three types of JXTA advertisements: the module class advertisement, module specification advertisement and module implementation advertisement, which jointly describe the same Edutella/JXTA P2P service at different abstraction levels [104]. Despite that all these abstraction levels can find their conceptual mappings in WSDL, the description formats of both are not compatible. Moreover, as Edutella/JXTA and Web services adopt different entity identification systems and also use different protocols for the service invocation, the Edutella/JXTA service advertisements cannot provide a Web services node with necessary information (e.g., IP address) for locating the access point of P2P services and further invoking these services. Likewise, the SOAP RPC information provided by WSDL also makes no sense for Edutella/JXTA peers that wish to invoke Web services, as the SOAP RPC is not yet supported in Edutella/JXTA.

Fourth, although both Web services and JXTA addresses some common security issues such as the encryption, hashing and authentication [79][105], their security implementations are based on different security models. This security model incompatibility further obstructs the direct interaction between the two platforms.

The security of the Edutella/JXTA platform is implemented based on the so-called “Web of trust” security model [105]. An Edutella/JXTA peer can loan out its credentials to another peer and a community is established by linking trusted peers. As a Web services node does not conduct any JXTA configurations, it has no chance to participate in an Edutella/JXTA community and sequentially interact with Edutella/JXTA peers and P2P services.

In overall, all four sides of challenges stem from the wire layer incompatibility between Edutella/JXTA and Web services. By defining and implementing sets of wire layer protocols dealing with the peer identification, peer establishment, communication management and some other low-level “plumbing”, Edutella/JXTA constructs a relatively independent realm, which is not yet SOAP-aware thus excludes any direct interaction with Web services nodes that are not “JXTAlized” at the wire layer. Although JXTA has promised to become SOAP-aware at the wire layer through the JXTA SOAP binding [106], the ongoing efforts show that the JXTA SOAP binding might likely impact the primary JXTA specification thus still has a long way to go when the issues regarding its compatibility with other JXTA bindings are considered. Before JXTA becomes SOAP-aware at the wire layer, we need a technical approach being able to overcome the wire layer incompatibility between the two platforms in order to achieve the interaction between Edutella/JXTA and Web services.

4.3. Service Layer Interaction between Edutella/JXTA and Web Services: Technical Implementation

The service layer interaction between Edutella/JXTA and Web services is referred to the functionality exchange between the Web services’ service layer and JXTA service layer (see also Section 3.2.1). The key to achieving the service layer interaction between the two platforms is the use of the Web-services/Edutella proxies. A Web-services/Edutella proxy is an Edutella/JXTA peer, which also possesses a SOAP implementation module responsible for mediating the SOAP RPC between Edutella/JXTA and Web services. In order to be identified outside of Edutella/JXTA, unlike usual Edutella/JXTA peers, the Web-services/Edutella proxy has to expose its IP address besides its usual JXTA ID. It actually plays a double role, being able to directly address a few of aforementioned challenges such as the functional entity mapping and the security management. In the following we take two typical scenarios of the service layer interaction between the two platforms as an example to elaborate how the design and implementation of the Web-services/Edutella proxies can also address other challenges left through resolving several critical issues to the realization of the service layer interaction, e.g., how to overcome the service description incompatibility between P2P service advertisements and WSDL, in which way to expose existing Edutella/JXTA P2P services as Web services, as well as how to mediate the service invocation between Edutella/JXTA and Web services, etc.

4.3.1. Exposing Existing Edutella/JXTA P2P Services as Web Services

The first Edutella/JXTA prototype implements sets of P2P services including the Edutella query service, annotation service, replication service, etc. In this section we take the Edutella query service as an example to present the interaction scenario of exposing existing Edutella/JXTA P2P services as Web services. The technical implementation of this interaction scenario is illustrated in Figure 4.3.

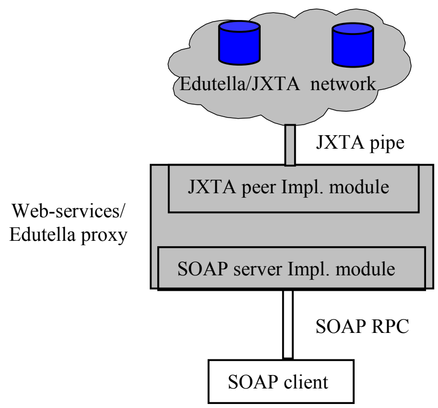


Figure 4.3 Exposing existing Edutella/JXTA P2P services as Web services

As a standard Edutella/JXTA P2P service, the Edutella query service can be discovered by any Edutella/JXTA peers within Edutella/JXTA by means of JXTA's decentralized discovery mechanism. Any Edutella/JXTA peers wishing to consume Edutella resources first need to bootstrap into Edutella/JXTA, search and discover the Edutella query service advertisements, and then follow the advertisements to open a JXTA output pipe to send the query to the Edutella query service, as well as open a JXTA input pipe to wait for the query result. In practice, an Edutella/JXTA consumer peer can accomplish all these operations leveraging the Edutella consumer service initiated during its JXTA bootstrapping process. The Edutella query service advertisements can provide the Edutella consumer service with all necessary information for the service invocation.

For a Web services client outside of Edutella/JXTA, the Edutella query service is neither discoverable nor directly invokable. The key to exposing the Edutella query service to the Web services world is the use of the Web services/Edutella proxy, which, as a usual JXTA peer, initiates the Edutella consumer service that is responsible for searching, locating and interacting the Edutella query service during its JXTA bootstrapping process. Besides this JXTA implementation module, the Web services/Edutella proxy also runs a SOAP server implementation module responsible for exposing the functionality of the Edutella query service.

In general, we have two approaches for exposing the functionality of the Edutella query service to the Web services world. The first is the so-called direct wrapping approach, in which the Edutella query service has to be re-implemented as a pure Web service hosted by the SOAP server implementation module. This SOAP version of the Edutella query service would have to directly communicate with Web services clients through the SOAP RPC, manage the message exchanging between the SOAP RPC and JXTA pipes, as well as complete its original duty assigned in Edutella/JXTA, namely, interacting with all Edutella/JXTA content providers to distribute the query and gather the query result using JXTA pipes. Due to the considerable wire layer incompatibility between Edutella/JXTA and Web services, implementing the Web services/Edutella proxy according to this direct wrapping approach would have to deal with many technical complexities.

The second is the so-called broker wrapping approach, in which the Web service to be exposed is not the re-implementation of the Edutella query service, but instead a broker Web service that invokes the Edutella query service. This broker Web service does not need to directly interact with JXTA pipes, instead it is merely responsible for managing the message exchanging with Web services clients via the SOAP RPC, and directly leverages the Edutella consumer service to handle the interaction with the Edutella query service. In our implementation, we employ the broker wrapping approach to realize the broker Web service for exposing the functionality of the Edutella query service. This broker Web service is described by a WSDL version of the Edutella query service advertisements, as illustrated in Figure 4.4. Hosted by the SOAP server implementation module on the Web services/Edutella proxy, this broker Web service can exist in the Web services world and undergo the publish-find-bind lifecycle just like other usual Web services. From the perspective of Web services clients, the Web services/Edutella proxy acts as a usual Web services provider identified by its IP address. In Edutella/JXTA, however, this proxy acts as an Edutella/JXTA "consumer" peer identified by its JXTA ID.


```xml
  <?xml version="1.0" encoding="UTF-8" ?>
  <!-- generated by GLUE standard 4.0b2 on Fri Apr 11 10:58:33 CEST 2003  -->
- <wsdl:definitions name="EdutellaQueryWS" targetNamespace="http://www.learninglab.de/wsdl/"
    xmlns:tns="http://www.learninglab.de/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tme="http://www.themindelectric.com/">
  - <wsdl:message name="getEdutellaQueryResult0In">
    - <wsdl:part name="Query" type="xsd:string">
        <wsdl:documentation>The RDF-QEL query sent to Edutella</wsdl:documentation>
      </wsdl:part>
    </wsdl:message>
  - <wsdl:message name="getEdutellaQueryResult0Out">
    - <wsdl:part name="Result" type="xsd:string">
        <wsdl:documentation>The query results returned from Edutella</wsdl:documentation>
      </wsdl:part>
    </wsdl:message>
  - <wsdl:portType name="EdutellaQueryWS">
    - <wsdl:operation name="getEdutellaQueryResult" parameterOrder="Query">
        <wsdl:input name="getEdutellaQueryResult0In" message="tns:getEdutellaQueryResult0In" />
        <wsdl:output name="getEdutellaQueryResult0Out" message="tns:getEdutellaQueryResult0Out" />
      </wsdl:operation>
    </wsdl:portType>
  - <wsdl:binding name="EdutellaQueryWS" type="tns:EdutellaQueryWS">
      <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
    - <wsdl:operation name="getEdutellaQueryResult">
        <soap:operation soapAction="getEdutellaQueryResult" style="rpc" />
      - <wsdl:input name="getEdutellaQueryResult0In">
          <soap:body use="encoded" namespace="http://www.learninglab.de/"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:input>
      - <wsdl:output name="getEdutellaQueryResult0Out">
          <soap:body use="encoded" namespace="http://www.learninglab.de/"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  - <wsdl:service name="EdutellaQueryWS">
      <wsdl:documentation>The Edutella Query Web service accepts Edutella RDF-QEL query in the format of
        xsd:string and returns query results as flat RDF document</wsdl:documentation>
    - <wsdl:port name="EdutellaQueryWS" binding="tns:EdutellaQueryWS">
        <soap:address location="http://130.75.152.216:8004/glue/EdutellaQueryWS" />
      </wsdl:port>
    </wsdl:service>
  </wsdl:definitions>
```


Figure 4.4 WSDL description of the broker Web service

Besides the Edutella query service, all other Edutella/JXTA P2P services can also be exposed to the Web services world through the same broker Web service. In the WSDL description, each Edutella/JXTA P2P service corresponds to a WSDL "operation" element, having its own input and output message types and possessing individual SOAP interaction styles. In terms of these "operation" requirements, Web services clients can easily generate code to invoke the broker Web service, which sequentially conducts corresponding operations to expose Edutella/JXTA functionalities to the Web services world.

4.3.2. Integrating Web Services Enabled Content Providers into Edutella/JXTA

Just like usual Edutella/JXTA content provider peers, in order to be integrated into Edutella/JXTA, Web services enabled content providers must strictly follow Edutella/JXTA content provider integration protocols (see also Section 3.3.2) in the way that they develop the Web services to accomplish usual Edutella wrapper program functionalities. In a sense, these Web services can be viewed as the SOAP-based wrapper program implementations. They must take the RDF-QEL query as the SOAP RPC message input and return the RDF-QEL query result as the SOAP RPC message output. In addition, they also have to translate the RDF-QEL query into the local query of the underlying content repository, and then transform the local query result into the RDF-QEL query result.

Corresponding to different types of content provider repositories, although the functional implementations of each "wrapper" Web services may be somewhat distinct, they use rather similar WSDL descriptions sharing almost all major description elements except the element of the SOAP server address. This feature can be seen from an example WSDL description of a Web services enabled content provider as illustrated in Figure 4.5. During the implementation of the Web services/Edutella proxy, this feature enables us to use a unique SOAP client implementation module to bind to different Web services enabled content providers.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!-- generated by GLUE Professional 4.1.2 on Sat Jul 05 14:11:05 CEST 2003 -->
- <wsdl:definitions name="EdutellaProviderWS" targetNamespace="http://www.learninglab.de/wsdl/"
    xmlns:tns="http://www.learninglab.de/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:tme="http://www.themindelectric.com/">
  - <wsdl:message name="getEduResultSet0In">
      <wsdl:part name="qel" type="xsd:base64Binary" />
    </wsdl:message>
  - <wsdl:message name="getEduResultSet0Out">
      <wsdl:part name="Result" type="xsd:base64Binary" />
    </wsdl:message>
  - <wsdl:portType name="EdutellaProviderWS">
    - <wsdl:operation name="getEduResultSet" parameterOrder="qel">
        <wsdl:input name="getEduResultSet0In" message="tns:getEduResultSet0In" />
        <wsdl:output name="getEduResultSet0Out" message="tns:getEduResultSet0Out" />
      </wsdl:operation>
    </wsdl:portType>
  - <wsdl:binding name="EdutellaProviderWS" type="tns:EdutellaProviderWS">
      <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
      <tme:optimizations tag="1" href="1" env="1" />
    - <wsdl:operation name="getEduResultSet">
        <soap:operation soapAction="getEduResultSet" style="rpc" />
      - <wsdl:input name="getEduResultSet0In">
          <soap:body use="encoded" namespace="http://www.learninglab.de/"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:input>
      - <wsdl:output name="getEduResultSet0Out">
          <soap:body use="encoded" namespace="http://www.learninglab.de/"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  - <wsdl:service name="EdutellaProviderWS">
    - <wsdl:port name="EdutellaProviderWS" binding="tns:EdutellaProviderWS">
        <soap:address location="http://130.75.152.216:8004/glue/EdutellaProviderWS" />
      </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```


Figure 4.5 Example WSDL description of a Web services enabled content provider

For Web services enabled content providers, developing the “wrapper” Web service according to Edutella/JXTA content provider integration protocols is only the prerequisite to the integration. Although the “wrapper” Web service is described in WSDL and can also be discovered within Edutella/JXTA through our P2P-based Web services registry network [109], it cannot directly be invoked in Edutella/JXTA, as the current Edutella/JXTA is not yet SOAP-aware thus cannot deliver the RDF-QEL query to Web services enabled content providers and sequentially gather the query result using the SOAP RPC. The key to invoking this “wrapper” Web service is the use of the Web-services/Edutella proxy, which, as a usual JXTA peer, also runs a SOAP client implementation module responsible for finding and dynamically binding the “wrapper” Web services. In Figure 4.6 we illustrate the technical implementation of this interaction scenario.

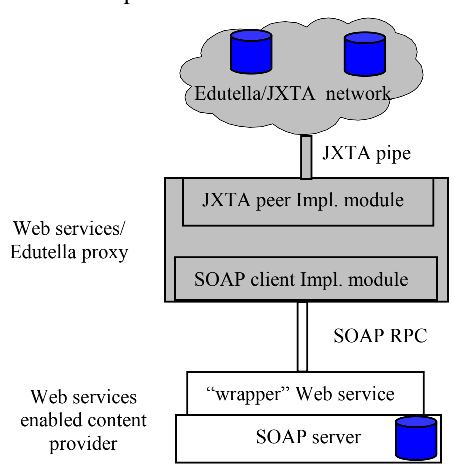


Figure 4.6 Integrating Web services enabled content providers into Edutella/JXTA

The Web services/Edutella proxy actually acts as the representative of Web services enabled content provider in the Edutella/JXTA P2P network. Like usual Edutella/JXTA content provider peers, it initiates the Edutella provider service and publicizes corresponding service advertisements during its JXTA bootstrapping process. These service advertisements declare the existence of the Web services/Edutella proxy (actually the existence of Web services enabled content provider), and also describe how the proxy can be accessed by the Edutella query service through JXTA pipes.

The Edutella provider service is responsible for opening a JXTA input pipe for accepting the RDF-QEL query from Edutella/JXTA, as well as opening a JXTA output pipe for sending back the query result. In contrast to the usual Edutella provider service, which directly forwards the RDF-QEL query coming from the JXTA input pipe to underlying content repositories and directly returns the RDF-QEL query result to the JXTA output pipe, the Edutella provider service running on the Web services/Edutella proxy needs a little bit of improvement to deal with the SOAP RPC based message exchanging between Edutella/JXTA and Web services enabled content providers. In the new implementation of the Edutella provider service, we interpose a SOAP client implementation module into the usual message transferring process. The RDF-QEL query coming from the JXTA input pipe is first transferred to the SOAP client implementation module, which then dynamically binds the wrapper Web service and forwards the RDF-QEL query to Web services enabled content providers through the SOAP RPC. After the query result is returned from Web services enabled content providers, this SOAP client implementation module is responsible for unmarshalling the query result from the SOAP RPC and then transferring the result back to the JXTA output pipe. From the perspective of Web services enabled content providers, they are not aware that their resources are being utilized in the context of a P2P setting. They exist in the Web services world and do not need to conduct any P2P configurations.

Via the Web services/Edutella proxy, Web services enabled content providers are "virtually" included into Edutella/JXTA's searching scope. Their resources can then be accessed by any Edutella/JXTA peers. In addition, these resources can also be accessed by usual Web services clients mediated by the Edutella/JXTA P2P network (see also Section 4.3.1). Note should be paid that in such accessing scenario, the resources provided by Web services enabled content providers are delivered as part of the Edutella/JXTA output following the common delivery protocols defined by Edutella/JXTA. In contrast, if these resources are directly accessed in the Web services world, they are individually delivered, not being embedded in the Edutella/JXTA resource network.

5. Integrating XML Metadata Repositories into Edutella

Due to considerable incompatibility between the XML data model and RDF data model, integrating XML metadata repositories into RDF-based Edutella introduces several critical challenges in comparison to integrating RDF metadata repositories. In this chapter we propose two approaches for integrating XML metadata repositories into Edutella in terms of XML query languages they support. In section 5.1 we first discuss several challenges we have to face while integrating XML metadata repositories into Edutella. In section 5.2 and 5.3 we propose two integration approaches respectively applicable to XPath-enabled XML metadata repositories and XQuery-enabled XML metadata repositories. In section 5.4 we propose an approach for querying complex XML data schemas in Edutella through QBE.

5.1. Integrating XML Metadata Repositories into Edutella: Challenges

Edutella is an RDF-based E-Learning content management P2P infrastructure, which obviously leads to RDF being the most naturally applicable metadata representation in Edutella and RDF metadata repositories being the most natural form of content repositories. However, nowadays a number of learning resource metadata still exist on the Web in the format of XML. In comparison to RDF metadata repositories, at least at present XML metadata repositories still occupy a quite dominant place in E-Learning. Apart from the reason that simple XML has a flatter learning curve and a more straightforward binding strategy to popular learning resource metadata sets than RDF, another important reason is that XML has a longer history to be applied for binding E-Learning standards and specifications. Taking the IMS MD as an example, it has provided the XML binding since the version 1.0 released in August 1999, whereas its RDF binding has been introduced since the version 1.2 released in June 2001. As a mater of fact, currently most of existing educational repositories are XML-based. They contain a large number of learning content to be addressed by Edutella.

Besides above reasons, the popularity of XML metadata repositories in E-Learning can also be attributed to a new type of XML back-end system: the native XML database [24]. Unlike some other XML back-end systems such as RDBs and OODBs, in which XML metadata usually need to be pre-processed and stored in some transformed representations, e.g., decomposed relational tables in RDBs or decomposed objects in OODBs, native XML databases provide a more straightforward way for constructing XML metadata repositories in that all XML metadata profiles can directly be stored and managed in their original hierarchical forms without the need of any pre-processing. In a native XML database, the database schema used to define how XML metadata are stored is virtually identical to the XML data schema defined by XML DTD or XML Schema. Therefore, based on a specific XML data schema, multiple XML metadata profiles can be contained in a single collection thus be queried as a whole using XPath [9][28] or XQuery [11]. Also XML metadata profiles can easily be updated through direct manipulation on XML fragments instead of on the whole metadata profiles. Actually, all these features of native XML databases satisfactorily fit into typical usage and management scenarios of learning resource metadata thus greatly promote the application of XML metadata repositories in E-Learning. Taking into account the current application status of XML metadata repositories, in this chapter we take native XML metadata repositories as the focus of our investigation.

Although XML and RDF are very close to each other, they are based on two different paradigms [98]. Whereas XML is concerned with syntax, focussing on providing a standard document format for writing and exchanging information on the Web, RDF is concerned with semantics, focussing on providing a standard model for describing the semantics and reasoning about information. As an RDF-based E-Learning content management P2P infrastructure, Edutella is proposed based on the essential assumption that all Edutella resources can be described in RDF and further all Edutella functionalities can be mediated through RDF statements and queries against these statements. However, despite that RDF provides essential possibilities for the integration of XML metadata, we still have to face several critical challenges while integrating XML metadata repositories into RDF-based Edutella.

First, there exists a considerable incompatibility between XML's tree-like hierarchical data model and ECDM/RDF's binary relational data model. This makes it rather difficult to generate a common data view of XML metadata, whose underlying data model is compatible with the ECDM/RDF data model.

According to the wrapper-like Edutella content provider integration architecture, in order to be integrated into Edutella, an XML metadata repository first has to provide a common data view whose underlying data model is compatible with the ECDM/RDF data model thus can be queried via ECDM's internal query language Datalog (see also Section 3.3.2). For a native XML metadata repository, as its database schema is virtually identical to the XML data schema stored, such a common data view has to be generated through manipulating original XML data models using local XML query languages, either XPath or XQuery. Since arbitrary XML data models may be very complex, for XPath-enabled XML metadata repositories, there exists no generic approach for manipulating arbitrary XML data models to generate such a common data view. This determines that Edutella can only integrate several schema-specific XPath-enabled XML metadata repositories whose XML data models are simple enough to be manipulated through XPath. For XQuery-enabled XML metadata repositories, as XQuery is more powerful than XPath especially with regard to the transformation capability, it is possible to find a generic integration approach. However, a new challenge here is how to reasonably design the XML syntax representation of the common data view. Such an XML syntax representation of the common data view should have an underlying XML data model that is compatible with the ECDM/RDF data model. It should also be able to be easily transformed from any XML data schemas by means of XQuery.

Second, with regard to query expressivity, the two XML query languages, particularly XPath, are not comparable to ECDM's internal query language Datalog. This makes it rather difficult to translate Datalog queries into XML queries as well as to transform local XML query results into RDF-QEL results.

According to the wrapper-like Edutella content provider integration architecture, the wrapper programs of XML metadata repositories first have to translate RDF-QEL queries into Datalog queries, which are sequentially translated into local XML query languages. After the query results are available, the wrapper programs also have to transform local XML query results into Datalog representations, which are sequentially transformed into RDF-QEL results transferring back to the Edutella network (see also Section 3.3.2). For XPath-enabled XML metadata repositories, whereas Datalog is a relationally complete query language being able to express relational algebra, e.g., "selection", "union", "join", "projection", etc., XPath is relationally incomplete thus cannot represent all Datalog queries. In terms of XML's tree-like data model, XPath can only express part of relational algebra such as "union", "negation", and limited "selection", but lacks the ability to express "join" and "projection". In consequence, XPath-enabled XML metadata repositories can only handle a limited number of RDF-QEL queries including the conjunctive query, disjunctive query and query negation, but cannot handle any type of recursive queries. In addition, due to XPath's limited capability of expressing "selection", as well as its incapability of expressing "join" and "projection", XPath-enabled XML metadata repositories can only return local XML query results as sets of XML fragments selected by XPath expressions rather than sets of tuples that can naturally be brought into the RDF model. In order to transform these local query results into RDF-QEL results, some additional processing are still needed.

For XQuery-enabled XML metadata repositories, as XQuery itself is relationally complete, it can represent all Datalog queries. However, in most cases these Datalog queries cannot simply be represented through simple XQuery expressions, instead they usually have to be translated into sets of XQuery function calls. A new challenge here is how to develop sets of XQuery functions in terms of the XML representation of the common data view in order to implement the relational algebra in XQuery. Moreover, despite that local XML query results returned by XQuery-enabled XML metadata repositories already possess an underlying XML data model compatible with the ECDM/RDF data model, they still need some additional processing in order to be transformed into RDF-QEL results.

Third, for schema-agnostic XML metadata repositories, since XML data schemas might complex enough, it is quite difficult to construct RDF-QEL queries against these complex XML data schemas.

Whereas for RDF metadata repositories the queries are always against a binary relational data model, for XML metadata repositories, the queries are against an XML data model that possesses a tree-like hierarchy. As the user's query interest are unforeseeable, for complex XML data schemas with multiple metadata entries, we cannot foresee which metadata entry would be queried and how Boolean logics between these queries would seem. Including all metadata entries and all possible query Boolean logics in a form-like query GUI is a straightforward first idea, but can usually lead to some cumbersome and inefficient query experience, especially when the XML data schema is complex enough. Moreover, as the queries should uniquely be in the RDF-QEL format in Edutella, the query construction process becomes more complex for inexperienced users. In order to query schema-agnostic XML metadata repositories in Edutella, we need some methods to simplify the RDF-QEL query construction process.

Due to considerable differences between XPath and XQuery regarding the query capability, the integration approaches for XPath-enabled and XQuery-enabled XML metadata repositories diverse quite a lot. In the following we propose two integration approaches respectively applicable to two types of XML metadata repositories.

5.2. Integrating XPath-enabled XML Metadata Repositories into Edutella

As the first tool designed for querying and manipulating XML metadata, XPath is currently supported by almost all XML metadata repositories. XPath itself is not a fully-fledged XML query language, rather it is designed to be embedded in some host languages such as XSLT [27] or XQuery. XPath operates on the data model of an XML document through using a path notation for navigating through the hierarchical structure of an XML document. It is a functional language that allows various kinds of expressions to be nested with full generality [9]. However, unlike the pure functional language, it does not allow variable substitutability if the variable definition contains construction of new nodes. With regard to the transformation and expression capability, XPath is too weak to manipulate arbitrary XML data models to generate the ECDM/RDF based common data view.

With the focus on the E-Learning content management, in this section we take the XML binding of two popular learning resource metadata sets: the DCMES and LOM/IMS/SCORM as an example to present an approach for integrating XPath-enabled XML metadata repositories into Edutella. The integration approach is principally discussed based on the DCMES. The LOM/IMS/SCORM can be mapped into the DCMES and then share the same integration approach.

5.2.1. Generating the ECDM/RDF Based Common Data View

The DCMES XML binding is the guideline proposed by DCMI for the XML encoding of the DCMES [7]. The primary goal of the guideline is to provide a simple DCMES encoding, where there are no extra elements, qualifiers, operational or varying parts allowed. The secondary goal is to make the encoding also be valid RDF, allowing the XML binding to be manipulated using the RDF model. The DCMES XML binding data model contains 15 elements. It can be described in the format of XML DTD as [33]:

```
<!ENTITY rdfns 'http://www.w3.org/1999/02/22-rdf-syntax-ns#' >
<!ENTITY dcns 'http://purl.org/dc/elements/1.1/' >
<!ENTITY % rdfnsdecl 'xmlns:rdf CDATA #FIXED "&rdfns;"' >
<!ENTITY % dcnsdecl 'xmlns:dc CDATA #FIXED "&dcns;"' >

<!ELEMENT rdf:RDF (rdf:Description)* >
<!ATTLIST rdf:RDF %rdfnsdecl; %dcnsdecl; >
 <!ENTITY % dcmes "dc:title | dc:creator | dc:subject | dc:description | dc:publisher |
 dc:contributor | dc:date | dc:type | dc:format | dc:identifier | dc:source | dc:language |
 dc:relation | dc:coverage | dc:rights" >

<!ELEMENT rdf:Description (%dcmes;)* >
<!ATTLIST rdf:Description rdf:about CDATA #IMPLIED>

<!ELEMENT dc:title (#PCDATA)>
<!ATTLIST dc:title xml:lang CDATA #IMPLIED>
<!ATTLIST dc:title rdf:resource CDATA #IMPLIED>

<!ELEMENT dc:creator (#PCDATA)>
<!ATTLIST dc:creator xml:lang CDATA #IMPLIED>
<!ATTLIST dc:creator rdf:resource CDATA #IMPLIED>

<!ELEMENT dc:subject (#PCDATA)>
<!ATTLIST dc:subject xml:lang CDATA #IMPLIED>
<!ATTLIST dc:subject rdf:resource CDATA #IMPLIED>

<!ELEMENT dc:description (#PCDATA)>
<!ATTLIST dc:description xml:lang CDATA #IMPLIED>
<!ATTLIST dc:description rdf:resource CDATA #IMPLIED>

<!ELEMENT dc:publisher (#PCDATA)>
<!ATTLIST dc:publisher xml:lang CDATA #IMPLIED>
<!ATTLIST dc:publisher rdf:resource CDATA #IMPLIED>

<!ELEMENT dc:contributor (#PCDATA)>
<!ATTLIST dc:contributor xml:lang CDATA #IMPLIED>
<!ATTLIST dc:contributor rdf:resource CDATA #IMPLIED>

<!ELEMENT dc:date (#PCDATA)>
<!ATTLIST dc:date xml:lang CDATA #IMPLIED>
<!ATTLIST dc:date rdf:resource CDATA #IMPLIED>

<!ELEMENT dc:type (#PCDATA)>
<!ATTLIST dc:type xml:lang CDATA #IMPLIED>
<!ATTLIST dc:type rdf:resource CDATA #IMPLIED>

<!ELEMENT dc:format (#PCDATA)>
<!ATTLIST dc:format xml:lang CDATA #IMPLIED>
<!ATTLIST dc:format rdf:resource CDATA #IMPLIED>

<!ELEMENT dc:identifier (#PCDATA)>
<!ATTLIST dc:identifier xml:lang CDATA #IMPLIED>
<!ATTLIST dc:identifier rdf:resource CDATA #IMPLIED>

<!ELEMENT dc:source (#PCDATA)>
<!ATTLIST dc:source xml:lang CDATA #IMPLIED>
<!ATTLIST dc:source rdf:resource CDATA #IMPLIED>
```

```
<!ELEMENT dc:language (#PCDATA)>
<!ATTLIST dc:language xml:lang CDATA #IMPLIED>
<!ATTLIST dc:language rdf:resource CDATA #IMPLIED>

<!ELEMENT dc:relation (#PCDATA)>
<!ATTLIST dc:relation xml:lang CDATA #IMPLIED>
<!ATTLIST dc:relation rdf:resource CDATA #IMPLIED>

<!ELEMENT dc:coverage (#PCDATA)>
<!ATTLIST dc:coverage xml:lang CDATA #IMPLIED>
<!ATTLIST dc:coverage rdf:resource CDATA #IMPLIED>

<!ELEMENT dc:rights (#PCDATA)>
<!ATTLIST dc:rights xml:lang CDATA #IMPLIED>
<!ATTLIST dc:rights rdf:resource CDATA #IMPLIED>
```

When the DCMES XML binding data model is viewed in a schematic way, it can also be represented in the hedgehog model as depicted in Figure 5.1 [62].

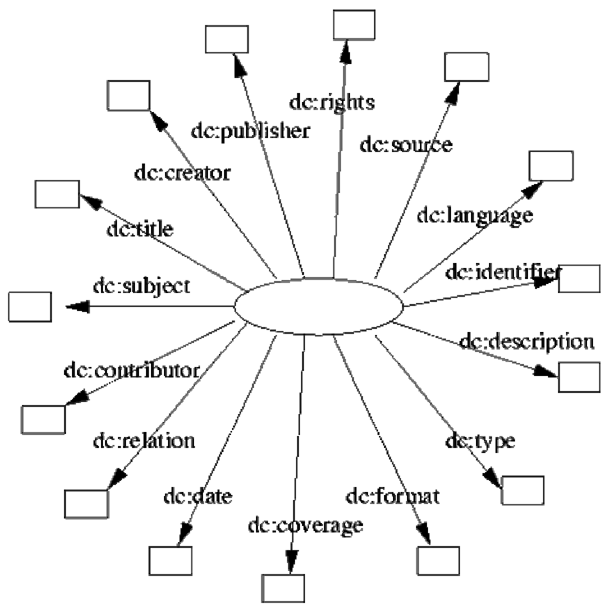


Figure 5.1 Hedgehog model of the DCMES XML binding

From the hedgehog model of the DCMES XML binding, in which all assertions are made about a fixed resource, we can see that there exists an obvious mapping from the DCMES XML binding data model to ECDM/RDF's binary relational data model. As the DCMES XML binding only uses limited sets of RDF constructs (e.g., *rdf:Bag*, *rdf:Seq* and *rdf:Alt* are excluded), the mapping becomes quite straightforward. In terms of XPath, we draw out three rules to map the DCMES XML binding data model into the binary relational data model. The XML data model is expressed through XPath location paths using XPath's abbreviated syntax.

Rule 1: *//*[@rdf:about] as u1* ⟶ *Subject*
Rule 2: *u1/* as u2* ⟶ *Predicate*
Rule 3: *u2[@rdf:resource] or u2[text()]* ⟶ *Object*

In the DCMES XML binding data model, the value of an element can be either plain text or another resource with a URI. This definition complies with the RDF data model in which an *object* can be either a literal or another resource. According to above three mapping rules, the wrapper program of XPath-enabled XML metadata repositories can generate the ECDM/RDF based common data view of any DCMES XML binding instances.

5.2.2. Developing the Wrapper Program

Translating RDF-QEL Queries into XPath. According to the wrapper-like Edutella content provider integration architecture (see also Section 3.3.2), translating RDF-QEL queries into XPath can be divided into two tasks. The first task is to translate RDF-QEL into Datalog, which, as the common task of all types of wrapper programs, is completed by a common parser program in Edutella. The second task is to translate Datalog into XPath, precisely speaking, to translate non-recursive Datalog queries into XPath queries.

One basic construct of Datalog is the *Literal*, which describes ground assertion and can be represented in a simplified form corresponding to the binary relational data model as: *P(arg1, arg2)*, where *P* is *Predicate* that might be a relation name or arithmetic predicates (e.g., “<”, “>”, etc.), and *arg1, arg2* are *Literals* that might be variables or constants. In Datalog, an *Literal* can be negated and represented as: *NOT P(arg1, arg2)*. In general, a Datalog query can be expressed as a set of Datalog rules. Each Datalog rule has a general representation as *head :- literal1, literal2,..., literaln*, where head is a single positive *Literal*, and *literal1 to literaln* are a set of *Literals* conjunctively called the body of the Datalog rule. The disjunction in Datalog is expressed as a set of rules with the identical head. In order to present the translation process from Datalog into XPath, we construct an example non-recursive Datalog query, which covers the conjunctive query, disjunctive query and query negation.

```
H(X) :- P1(X,U), NOT P2(X, V)
H(X) :- P3 (X,W)

H is head; P1, P2, P3 are predicates; X is variable; U, V, W are constants.
```

Corresponding to the DCMES XML binding data model, the example Datalog query can be translated into XPath as:

```
//*[@rdf:about
    and
    (P1 [@rdf:resource=U] or P1 [text()=U])
   and
   not (P2 [@rdf:resource=V] or P2 [text()=V])
  ]
  |
//*[@rdf:about
  and
    (P3[@rdf:resource=W] or P3 [text()=W])
  ]
```

XPath can be viewed as a general purpose query notation for addressing and filtering elements and text of XML documents. A notation indicates the hierarchical relationship between the nodes and is used by a pattern to describe the types of nodes to match. All XPath queries occur within a particular context, which is the single node against which the pattern matching operates. The collections of all elements selected from the current context by XPath queries preserve document order, hierarchy and identity, to the extent that these are defined. In addition, constraints and branching can be applied to any collection by adding a filter clause to the collection. The filter in XPath is analogous to the SQL WHERE clause, expressed in the form of *[filter pattern]*. The filter pattern evaluates to a Boolean value and is tested for each element in the collection. Any elements in the collection failing the filter pattern test are omitted from the result collection.

In general, each Datalog rule can be mapped into an XPath pattern based on which a set of elements is selected under a certain context. The conjunctive queries, represented in Datalog by a number of Datalog *Literals* contained in a single rule, are translated into sets of filter patterns that are combined together using the XPath Boolean operator *“and”* and are applied to the collection selected by the XPath pattern. The negation of a Datalog *Literal* can be represented using the XPath Boolean operator *“not”*.

The disjunctive queries, represented in Datalog by a number of Datalog rules with the identical head, are expressed in XPath by a number of patterns combined together using the XPath union operator *“|”*. Multiple union operators can union together sets of collections selected by multiple XPath patterns, and can also exclude duplicates. In the example XPath query, we also use several XPath operators for grouping operation, filtering operation, Boolean operation and path operation. These operators are used in terms of certain precedence orders. In Table 5.1 we list these operators according to their precedence orders, from the highest to the lowest [28].

Table 5.1 XPath operators and their precedence orders

1	( )	Grouping
2	[ ]	Filter
3	/ //	Path operations
4	|	Union
5	not ( )	Boolean not
6	and	Boolean and
7	or	Boolean or

Note should be paid that the example Datalog query against a binary relational data model can be viewed as a query for *Subjects*. In Datalog it is also easy to express the queries for *Predicates* and *Objects*. Referencing the example XPath expressions, we can easily translate these Datalog queries into XPath through the similar approach.

Transforming Local XML Query Results into RDF-QEL Results. With regard to the DCMES XML binding, XPath queries can only return a set of whole XML metadata profiles as local query results, since any XPath query must take an XML metadata profile as a whole in order to get a virtual binary relational data model against which the XPath query can be operated. Although most of XPath-enabled XML metadata repositories also provide means for further identifying underlying elements/attributes of any XML fragments, we determine to use the whole XML metadata profile as the direct output and leave the further processing work on local query results to an RDF parser: Jena [71]. An important consideration for this choice is that DCMES XML binding metadata profiles themselves are in valid RDF syntax thus can directly be handled by RDF parsers. Through using RDF parsers, local XML query results can easily be transformed into the RDF model and then adapted into the RDF-QEL result format. Nevertheless, it is worth noting that in comparison to local query results returned from RDF metadata repositories, the local query results returned from XPath-enabled XML metadata repositories are always somewhat redundant.

5.2.3. Integrating LOM/IMS/SCORM XML Binding Metadata Repositories into Edutella

In comparison to the DCMES XML binding data model, the LOM/IMS/SCORM XML binding data schema is much more complex, consisting of nine categories, over 60 metadata entries and possibly recursive elements, e.g., in the category of "Classification". For such a complex XML data schema, it is rather difficult to generate the ECDM/RDF based common data view using XPath and further apply the same integration approach that is applicable to the DCMES XML binding. In order to achieve the integration, we employ a transformation approach, which depends on the DCMES XML binding as a lingua franca and scale-down maps the LOM/IMS/SCORM XML binding into the DCMES XML binding through using XSLT. After the transformation, LOM/IMS/SCORM XML binding based XML metadata repositories can directly leverage the same integration that is applicable to DCMES based XML metadata repositories.

As one can imagine, such a transformation from LOM/IMS/SCORM to DCMES will unavoidably lose some original LOM/IMS/SCORM metadata information. However, we argue that most of lost metadata information are only useful for detailed description of learning resources rather than for the simple discovery of these resources. Thus, the integration approach proposed for LOM/IMS/SCORM based XML metadata repositories can still ensure the essential discoverability of learning content. Moreover, the validity of this integration approach is also guaranteed by the common efforts from the IEEE LTSC and DCMI (especially DCMI-Ed), which have continuously focused on providing enough interoperability between the LOM/IMS/SCORM and DCMES, as outlined in the Memorandum of Understanding between the IEEE LTSC and DCMI.

In Table 5.2 we list 15 rules used to map the LOM/IMS/SCORM to DCMES [51]. According to these rules, the transformation from the LOM/IMS/SCORM to DCMES can easily be accomplished through an XSLT program.

Table 5.2 Rules used to map the LOM/IMS/SCORM to DCMES

LOM/IMS/SCORM	DCMES
1.1.2:General.Identifier.Entry	DC.Identifier
1.2:General.Title	DC.Title
1.3:General.Language	DC.Language
1.4:General.Description	DC.Description
1.5:General.Keyword or 9:Classification with 9.1: Classification.Purpose equals "Discipline" or "Idea".	DC.Subject
1.6:General.Coverage	DC.Coverage
5.2:Educational.LearningResourceType	DC.Type
2.3.3:LifeCycle.Contribute.Date when 2.3.1: LifeCycle.Contribute.Role has a value of "Publisher".	DC.Date
2.3.2:LifeCycle.Contribute.Entity when 2.3.1: LifeCycle.Contribute.Role has a value of "Author".	DC.Creator
2.3.2:LifeCycle.Contribute.Entity with the type of contribution specified in 2.3.1: LifeCycle.Contribute.Role.	DC.Other-Contributor
2.3.2:LifeCycle.Contribute.Entity when 2.3.1: LifeCycle.Contribute.Role has a value of "Publisher".	DC.Publisher
4.1:Technical.Format	DC.Format
6.3:Rights.Description	DC.Rights
7.2.2:Relation.Resource.Description	DC.Relation
7.2:Relation.Resource when the value of 7.1:Relation.Kind is "IsBasedOn".	DC.Source

In XML metadata repositories, all XML metadata profiles are stored in separate XML collections in terms of specific XML data schemas. Utilizing an XSLT program, we can create a specific collection to store the LOM/IMS/SCORM metadata profiles transformed from the original XML metadata instances. Since each XML metadata profile stored in XML metadata repositories usually possesses a unique key to identify itself, we can also retrieve the original LOM/IMS/SCORM metadata profile and sequentially get all metadata information.

5.3. Integrating XQuery-enabled XML Metadata Repositories into Edutella

XQuery is a fully-fledged XML query language containing XPath 2.0 as a subset [37]. It is a functional language in which a query is represented as an expression. Unlike XPath, which only consists of path expressions used to specify a sequence of nodes or primitive values, XQuery adopts a main engine of expressions: the FLWOR expression, which generalizes *select-from-having-where* expression from SQL thus makes XQuery relationally complete. The FLWOR expression supports iteration and binding of variables to intermediate results. Such kind of expression is often useful for computing joins between two or more documents and for re-structuring data [11]. Additionally, XQuery also employs several other expressions such as the sequence expressions, arithmetic expressions, comparison expressions, logical expressions, constructors, unordered expressions, conditional expressions, quantified expressions, and expressions on SequenceTypes. In Table 5.3 we give an overview of all XQuery expressions [11].

Table 5.3 Overview of XQuery expressions

XQuery Expressions	Functionality
Primary Expressions	include literals, variables, function calls, constructors, and the use of parentheses to control precedence of operators.
Path Expressions	used to locate nodes within a tree-like XML data model.
Sequence Expressions	support operators to construct and combine sequences. In XQuery, a sequence is an ordered collection of zero or more items. An item may be an atomic value or a node.
Arithmetic Expressions	provide arithmetic operators for addition, subtraction, multiplication, division, and modulus, in their usual binary and unary forms.
Comparison Expressions	allow two values to be compared. XQuery provides four kinds of comparison expressions, including value comparisons, general comparisons, node comparisons, and order comparisons.
Logical Expressions	either an “and-expression” or an “or-expression”.
Constructors	be able to create XML structures within a query. Constructors are provided for every kind of node in the XML data model except namespace nodes. A special form of constructor called a computed constructor can be used to create an element or attribute with a computed name or to create a document node or a text node.
FLWOR Expressions	support iteration and binding of variables to intermediate results. This kind of expression is often useful for computing joins between two or more documents and for restructuring data.
Unordered Expressions	take any sequence of items as its argument, and return the same sequence of items in a non-deterministic order.
Conditional Expressions	Operate based on the keywords “if”, “then”, and “else”.
Quantified Expressions	support existential and universal quantification.
Expressions on SequenceTypes	In addition to the use in function parameters and results, SequenceTypes are used in “instance of”, “typeswitch”, “cast”, “castable”, and “treat expressions”.

Since XQuery expressions greatly improve the transformation and expression capability of XQuery, it becomes possible for XQuery-enabled XML metadata repositories to find a generic approach for integrating arbitrarily complex XML data schemas into Edutella. With the focus on the E-Learning content management, in this section we use the LOM/IMS/SCORM XML binding data schema as an example to present an approach for integrating schema-agnostic XML metadata repositories into Edutella. In contrast to the XPath-enabled integration approach, the XQuery-enabled approach can achieve the integration without the loss of original metadata information.

5.3.1. Generating the ECDM/RDF Based Common Data View

ECDM/RDF’s binary relational data model can be represented in the XML syntax in various ways. As the syntactic form of the XML data may strongly affect complexity and run-time cost of XQuery, we propose a triple-like XML data schema to represent the ECDM/RDF based common data view. The XML DTD of the triple-like common data view can be described as:

```
<!ELEMENT statements (statement+)>
<!ELEMENT statement (subject, predicate, object)>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT predicate (#PCDATA)>
<!ELEMENT object (#PCDATA)>
```

The triple-like XML representation of the ECDM/RDF based common data view has two features. First, its underlying data model is fully compatible with the ECDM/RDF data model thus can directly be queried via ECDM’s internal query language Datalog. Second, it adopts a very simple XML syntax, which can easily be manipulated through XQuery thus can ensure the query's run-time performance. By means of corresponding XQuery functions, any arbitrarily complex XML data schemas can be transformed into the common data view.

In order to present the generating process of the triple-like common data view, we take a LOM/IMS/SCORM XML binding metadata entry: *lom.general.catalogentry* as an example. In Figure 5.2 we show the graphical data model of this metadata entry in the form of the XML Schema.

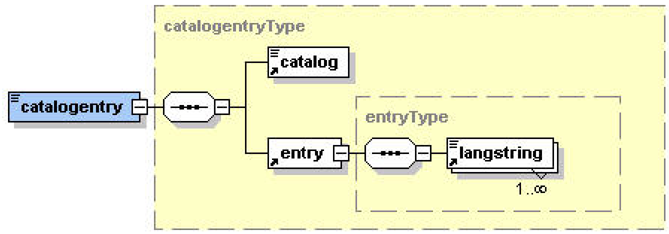


Figure 5.2 Graphical data model of an example LOM/IMS/SCORM XML binding metadata entry

An example metadata instance of this metadata entry might seem as:

```
<?xml version="1.0" encoding="UTF-8"?>
<lom xmlns="http://www.imsglobal.org/xsd/imsmd_rootv1p2p1"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.imsglobal.org/xsd/imsmd_rootv1p2p1
                        http://www.imsglobal.org/xsd/imsmd_rootv1p2p1.xsd">
  <general>
    <catalogentry>
      <catalog>Blah</catalog>
        <entry>
          <langstring xml:lang="en"> Blah EN</langstring>
          <langstring xml:lang="de"> Blah DE</langstring>
        </entry>
    </catalogentry>
  </general>
</lom>
```

When the elements of the XML data model are represented using RDF properties, the example metadata instance can be described through an RDF graph as illustrated in Figure 5.3.

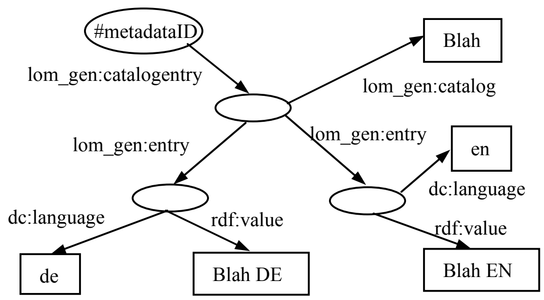


Figure 5.3 RDF graph representing an example LOM/IMS/SCORM XML binding metadata instance

While representing XML data schemas as RDF graphs, we try the best to follow the RDF Model and Syntax specification [66]. On the one hand, we make heavy use of anonymous resources as well as some RDF built-in properties such as *rdf:type* and *rdf:value*. On the other hand, we also try to remain compatibility with some popular metadata sets such as the DCMES, DCMI Metadata Terms and vCard, etc., in the representation. With regard to the LOM/IMS/SCORM XML binding, we also take into account the potential LOM/IMS/SCORM RDF binding [54]. As a matter of fact, the design of the RDF graph covers most of principal design criteria proposed for the potential LOM/IMS/SCORM RDF binding.

According to the XML DTD defined for the triple-like common data view, the RDF graph representing the example LOM/IMS/SCORM XML binding metadata instance can be serialized into the triple-like common data view by means of XQuery functions. In terms of such kind of triple-like common data view, the XML metadata repositories can directly be queried via RDF-QEL in Edutella.

```
<?xml version='1.0' encoding='ISO-8859-1'?>
```

```
<statements xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:lom_gen="http://www.imsproject.org/rdf/imsmd_generalv1p2#">

<statement>
 <subject>#metadataID</subject>
 <predicate>lom_gen:catalogentry</predicate>
 <object>#anonymous_0</object>
</statement>
<statement>

 <subject># anonymous_0</subject>
 <predicate>lom_gen:catalog</predicate>
 <object>Blah</object>
</statement>

<statement>
 <subject># anonymous_0</subject>
 <predicate>lom_gen:entry</predicate>
 <object>#anonymous_1</object>
</statement>

<statement>
 <subject>#anonymous_0</subject>
 <predicate>lom_gen:entry</predicate>
 <object>#anonymous_2</object>
</statement>

<statement>
 <subject># anonymous_1</subject>
 <predicate>dc:language</predicate>
 <object>en</object>
</statement>

<statement>
 <subject># anonymous_1</subject>
 <predicate>rdf:value</predicate>
 <object>Blah EN</object>
</statement>

<statement>
 <subject># anonymous_2</subject>
 <predicate>dc:language</predicate>
 <object>de</object>
</statement>

<statement>
 <subject># anonymous_2</subject>
 <predicate>rdf:value</predicate>
 <object>Blah DE</object>
</statement>

</statements>
```


5.3.2. Developing the Wrapper Program

Similar to the wrapper program of XPath-enabled XML metadata repositories, the wrapper program of XQuery-enabled XML metadata repositories also has to accomplish two tasks: translating RDF-QEL queries into XQuery, and transforming local XML query results into RDF-QEL results. Since we adopt a triple-like common data view whose underlying data model is quite close to ECDM/RDF's binary relational data model, and also because XQuery is capable of returning query results in any desirable XML formats, the accomplishment of the second task is relatively straightforward. In this section, our discussion will be focused on the first task.

Translating RDF-QEL queries into XQuery consists of two steps. First, RDF-QEL queries have to be translated into Datalog queries. This is a common task for all types of Edutella wrapper programs and is completed by a common parser program in Edutella. Second, Datalog queries have to be translated into XQuery. As XQuery is relationally complete and can perform all operations of relational algebra on XML's tree-like data model including also some complicated query operations such as the recursive query, all types of Datalog queries corresponding to the full set of RDF-QEL can be translated into XQuery.

In order to present the translation process from Datalog into XQuery, we still take the LOM/IMS/SCORM XML binding data model as an example. We first propose an example query that can be read in plain English as:

find a LOM/IMS/SCORM metadata record, whose lom.general.title entry contains English value "computer" and lom.general.keyword entry contains English value "TCP", or a LOM/IMS/SCORM metadata record, whose lom.general.description entry contains English value "network". This example query can be represented in Datalog as:

```
scorm(X) :- lom_gen:title(X,U), dc:language(U,"en"), rdf:value(U,"computer"),
            lom_gen:keyword(X,V), dc:language(V,"en"), rdf:value(V,"TCP")
scorm(X) :- lom_gen:description (X,W), dc:language(W,"en"), rdf:value(W, "Network")
? - scorm(X)

Here X,U,V,W are Datalog variables
```

For the LOM/IMS/SCORM XML binding data model, we have developed an XQuery function library containing sets of functions used to query different metadata entries. The example Datalog query can be translated into sets of calls to the XQuery function library.

```
let $p := query_on_element_with_langstring("lom_gen_title","","en","computer")
let $q := query_on_element_with_langstring("lom_gen_keyword ","","en","TCP")
let $r := query_on_element_with_langstring("lom_gen_description","","en", "Network")
return handle_Boolean_OR( handle_Boolean_AND($p union $q) union $r)
```

The XQuery query is generally represented through FLOWR expressions. The returned query results are further handled by two specific XQuery functions: "*handle_Boolean_OR*" and "*handle_Boolean_AND*", which are responsible for managing Boolean logics between the queries against multiple metadata entries. These two functions can also eliminate duplicate local XML query result sets.

5.4. Querying Complex XML Data Schemas through QBE

In contrast to queries against RDF metadata, which can always be proposed based on a definite binary relational data model, queries against XML metadata have to be proposed based on a hierarchical tree-like XML data model thus have no generic construction rules. Corresponding to different XML data schemas, the complexity of XML queries may vary a lot. When an XML data schema becomes complex enough, the query construction may become rather difficult, since the complexity of XPath or XQuery queries may drastically increase with the increasing complexity of XML data schemas. For XML metadata, the most straightforward way to construct a query is to use the form-like query GUI, which can graphically show the XML data model as sets of forms and enable users to build queries through filling in forms. However, for complex XML data schemas with multiple metadata entries, including all metadata entries and all possible query Boolean logics in a form-like query GUI can usually lead to some cumbersome and inefficient query experience. Moreover, in Edutella, queries against XML metadata have to be expressed in the RDF-QEL format, which further complicates the query construction process for inexperienced users. In order to facilitate the query construction process, we propose to use QBE to query complex XML data schemas.

QBE is a graphical language originally designed for querying RDBs. The idea behind QBE is that the user provides an example of outputs that he expects from the query and constructs the query by filling example tables [118]. Whereas QBE fits well with RDBs in that QBE's tabular query interface is quite analogous to the internal tabular structure of RDBs, it cannot directly be used to query XML metadata repositories, which, as the document databases by nature, adopt hierarchical tree-like data model to store XML metadata. In order to query complex XML data schemas, we propose an improved QBE, which uses a visual template to represent the query against individual XML metadata entry, and further adopts a single table to represent Boolean logics between multiple visual templates. While the visual template provides a quite analogous representation of the internal structure of individual XML metadata entry, the single tabular structure inherits QBE's original advantage for representing Boolean logics between queries.

In order to present the approach for querying complex XML data schemas through QBE, we still take the LOM/IMS/SCORM XML binding data model as an example. In Figure 5.4 we illustrate the QBE-based query GUI used to construct the example query mentioned in section 5.3.2.

The QBE-based GUI for querying the LOM/IMS/SCORM XML binding data model has four features:

- Arbitrary LOM/IMS/SCORM XML binding metadata entries could be taken as the "query example".
- User-friendly drag & drop manipulation based on the LOM/IMS/SCORM XML binding DOM tree.
- Automatic RDF-QEL query output.
- Integration of the graphical RDF-QEL result presentation.

In the QBE-based query GUI, the user can first choose metadata entries to be queried from the LOM/IMS/SCORM XML binding DOM tree through drag & drop manipulation. The user can then compose sets of visual templates corresponding to different metadata entries in a table in terms of Boolean logics between the queries. The output of the QBE-based query GUI is RDF-QEL queries. These RDF-QEL queries can directly be sent to the Edutella P2P network and are expected to get query results from all XML metadata repositories accommodating LOM/IMS/SCORM XML binding metadata.

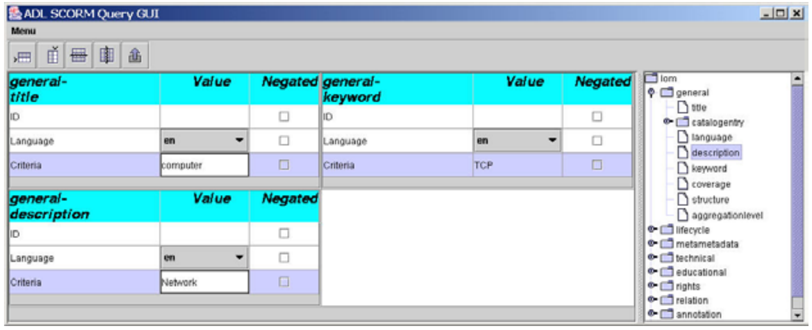


Figure 5.4 QBE-based GUI for querying LOM/IMS/SCORM XML binding metadata

Besides XML metadata, QBE can also be used to query RDF metadata repositories if only the data model of these RDF metadata repositories can graphically be represented so that the user can figure out a reasonable "query example". In this case the generic form of visual templates is based on the RDF binary relational data model. The Boolean logics between various RDF triples are represented through the QBE's tabular structure. In Figure 5.5 we demonstrate a QBE-based example GUI for querying an RDF metadata repository in which RDF metadata are stored according to the RDF graph depicted on the right side of Figure 5.5. The example QBE query can be read in plain English as: *find the author names of all AI books or author names of all books with the title AI.*

Subject: X Predicate: rdf:type Object: #AI-Book	Subject: X Predicate: dc:creator Object: Y	Subject: Y Predicate: vcard:fn Object: Z	
Subject: X Predicate: dc:title Object: AI	Subject: X Predicate: rdf:type Object: #Book	Subject: X Predicate: dc:creator Object: Y	Subject: Y Predicate: vcard:fn Object: Z

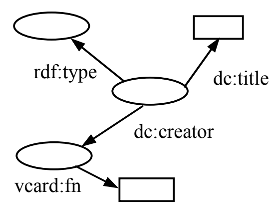


Figure 5.5 QBE-based example GUI for querying RDF metadata

6. Standard-oriented Learning Content Delivery

Besides the learning content discovery, the learning content delivery is another critical component of the E-Learning content management. In this chapter we propose an approach for realizing the standard-oriented learning content delivery, which can standardize the entire process of the learning content delivery by means of E-Learning standards and specifications. In section 6.1 we first propose a reference model for the standard-oriented learning content delivery, which is basically based on the ADL SCORM. In section 6.2 we take the design and implementation of a SCORM-conformant courseware as an example to present several enabling technologies for the implementation of the reference model. In section 6.3 we apply the reference model to the conventional AEHS and propose the standard-oriented AEHS, which can converge the E-Learning standardization and conventional AEHS research to ensure reusability and portability of learning content while achieving adaptability of the learning content delivery.

6.1. Reference Model for the Standard-oriented Learning Content Delivery

The reference model for the standard-oriented learning content delivery is purposed to knit together E-Learning standards and specification to standardize the entire process of the learning content delivery. A typical learning content delivery process is generally composed of two major operations: the learning content organization and run-time environment implementation. The learning content organization is mainly concerned with several design issues at authoring time, e.g., the definition of the content model, the design of the content packaging and content sequencing, etc. The run-time environment implementation is mainly concerned with several design issues at run-time, e.g., the definition of a data model to store content interaction status and track the learner's interactions, the design of a common API to enable the communication between learning content and LMSs, etc. To address these design issues, we propose a reference model covering following aspects.

- *Content model*: The content model defines how learning content can be identified and described, aggregated into a course or portion of course, and moved between different LMSs or content repositories. In the reference model, we propose to use the SCORM content model defined in the SCORM CAM [2] as the content model specification. So far the SCORM is the only E-Learning specification that defines a content model.
- *Content packaging*: The content packaging defines how learning content can be organized to represent the intended behaviour of a learning experience and how learning content can be packaged for movement between different LMSs. In the reference model, we propose to use the IMS CP [55] as the content packaging specification. The IMS CP is also adopted by the SCORM.
- *Content sequencing*: The content sequencing defines how learning content can be delivered as sets of interrelated learning activities that can be sequenced in a consistent way according to the outcomes of the learner's interactions. In the reference model, we propose to use the IMS SS [56] as the content sequencing specification. The IMS SS is also adopted by the SCORM.
- *Data model*: The data model supports the interchange of agreed upon data elements and their values between learning content and run-time services. In the reference model, we propose to use the IEEE Data Model [52] as the data model standard. The IEEE Data Model is directly derived from the AICC CMI 001- AICC/CMI Guidelines for Interoperability [4]. It is currently under development and is expected to be adopted by the future SCORM.
- *API*: The API provides a common mechanism to enable the communication between learning content and LMSs based on the data model. In the reference model, we propose to use the IEEE API [53] as the API standard. The IEEE API is directly derived from the AICC CMI 001- AICC/CMI Guidelines for Interoperability. It is currently under development and is expected to be adopted by the future SCORM.

The goal of the reference model is to improve reusability and portability of learning content. It provides an overall solution to standardize every aspects of a learning content delivery process thus constitutes the basis for the standard-oriented learning content delivery. As the proposed reference model is basically identical to the SCORM, in the following we take the design and implementation of a SCORM-conformant courseware as an example to present sets of enabling technologies for the implementation of the reference model.

6.2. Standard-oriented Learning Content Delivery in a SCORM-conformant Courseware

The kernel of the SCORM is the SCORM CAM, which represents a pedagogically neutral means for instructors to aggregate learning resources for the purpose of delivering a desired learning experience. The SCORM CAM consists of four components [2]:

- *SCORM content model*: A nomenclature defining the components of a content aggregation and content sequencing.
- *SCORM metadata*: A mechanism for describing specific instances of the components of the content aggregation.
- *SCORM content packaging*: A mechanism defining how to represent the content structure and how to package learning content.
- *SCORM SDM*: A rule-based model defining how to sequence and deliver learning activities.

The design and implementation of the SCORM-conformant courseware is centred upon these four components of the SCORM CAM. In this section we take a Web-based courseware: "Introduction to Java Programming" (Info1 for short) [84][85][86] as an example to present the standard-oriented learning content delivery in a SCORM-conformant courseware.

6.2.1. SCORM-conformant Content Model

The SCORM content model introduces four types of components: the Assets, SCA, SCO and Content Aggregations [2].

- *Assets*: An Asset is the most basic of learning resources. Assets are an electronic representation of media such as text, images, sound, assessment objects or any other piece of data that can be rendered by a Web client. Individual Assets are not launchable in a SCORM-conformant LMS.
- *SCA*: A SCA is a collection of one or more Assets packaged as a single launchable learning resource. The SCA is different from the SCO in the sense that the SCA does not communicate with a SCORM-conformant LMS.
- *SCO*: A SCO is a collection of one or more Assets that represents a single launchable learning resource that utilizes the SCORM RTE to communicate with LMSs. A SCO represents the lowest level of granularity of learning resources that is tracked by a LMS using the SCORM RTE data model.
- *Content Aggregations*: A Content Aggregation is a map (content structure) that describes cohesive units of instruction (activities), relates activities to one another, and may associate learning taxonomies to the activities (e.g., course, chapter, module, etc.). A Content Aggregation consists of activities, which may consist of other activities, nested arbitrarily deep. Activities that consist of other activities are called clusters. Clusters group activities into cohesive units of instruction, but do not have associated learning resources. Activities that do not consist of other activities will have an associated learning resource, either a SCA or a SCO.

For the design of the SCORM-conformant content model, a critical issue is to distinguish the SCO and Content Aggregations from the SCA and Assets. As the latter two components cannot be tracked by a LMS whereas the former two can be, the distinction between them may influence the overall strategy of the learning content delivery. Taking Info1 as an example, we take into account following rules for the distinction.

The original Info1 is a typical Web-based courseware that delivers learning content leveraging some standard Web technologies such as HTML, JavaScript, Java Applets, etc. Info1 adopts a new pedagogical method: project-based teaching/learning to organize its learning content. In Info1, learning resources are not sequentially structured according to some commonly used Java books or Java topics, rather they are organized into a comprehensive example project. Info1 includes about 30 self-developed course units or "project slices", each of which consists of corresponding Java concepts, appropriate learning materials about elementary computer science concepts, further references and source code, etc. Apart from these "internal" resources, Info1 also integrates a lot of "external" learning resources such as Web pages from the Sun Java Tutorial and Sun JDK API documents. During the SCORM-conformant re-design of Info1, both "internal" and "external" resources have to be adapted into the SCORM content model reasonably.

In the SCORM-conformant content model of Info1, whereas each course unit that has to be tracked during delivery can naturally be re-designed as a SCO, and all raw materials can naturally be re-designed as Assets or SCAs according to their original subordinating relationships, the re-design of "external" learning resources has to receive special attention. Despite of our desire to track these "external" learning resources using the SCORM RTE, all "external" learning resources are finally re-designed as SCAs. This is due to the strict definition of the SCO that a SCO by itself must be independent of learning context and there is no context relations among different SCOs, e.g., a SCO cannot be launched by another SCO and at a time only one SCO is allowed to be active.

As a matter of fact, all "external" resources in Info1 have certain learning context. These context exists not only within each course unit but also in the scope of the whole course. As an example, an "external" Java API document introduced in a specific course unit has the internal relationship with the Java concepts contained in the same course unit, and it also usually serves as the prerequisite to the study of following course units. Such sort of learning context has proven beneficial or even indispensable to help students understand each course unit and the whole Info1. However, if we would re-design these “external” resources as SCOs, they would have to lose all these valuable learning context because we cannot retain these context between SCOs due to the reason mentioned above. In contrast, if they are re-designed as SCAs, all learning context can be retained within a “parent” SCO.

Another critical issue to the design of the SCORM-conformant content model is to associate SCOs or SCAs into higher-level Content Aggregations. The purpose of applying Content Aggregations is to further modularise learning content in order to facilitate the reuse and exchange of learning resources between different LMSs at different aggregation levels. Taking Info1 as an example, about 30 course units (SCOs and SCAs) are aggregated into eight Content Aggregations. Besides considerations on the modularisation, aggregating SCOs and SCAs into Content Aggregations also has to take into account the design of the content sequencing. Although we cannot retain learning context between each SCOs, the use of Content Aggregations can to a certain degree facilitate the control over the launching sequence of each SCOs within a higher-level module through the SCORM RTE thus provide SCOs with some “virtual” context. In Info1, all SCOs and SCAs are aggregated into Content Aggregations in terms of the “virtual” learning context between each SCOs, e.g., the Java Applet Content Aggregation consists of three SCOs about the same topic, and the server-side programming Content Aggregation includes four SCOs. In terms of the SCORM content model, all Assets, SCAs, SCOs and Content Aggregations can be annotated with corresponding SCORM metadata and further be discovered within Edutella (see also Chapter 5).

6.2.2. SCORM-conformant Content Packaging

The SCORM content packaging is used to present the course content structure and provide the description for a course content package. It is based on the IMS CP but further extends the IMS CP with several additional SCORM-specific elements particularly in the “organization” section where the SCORM content structure is located. In Figure 6.1 we illustrate the SCORM content packaging information model [2][55].

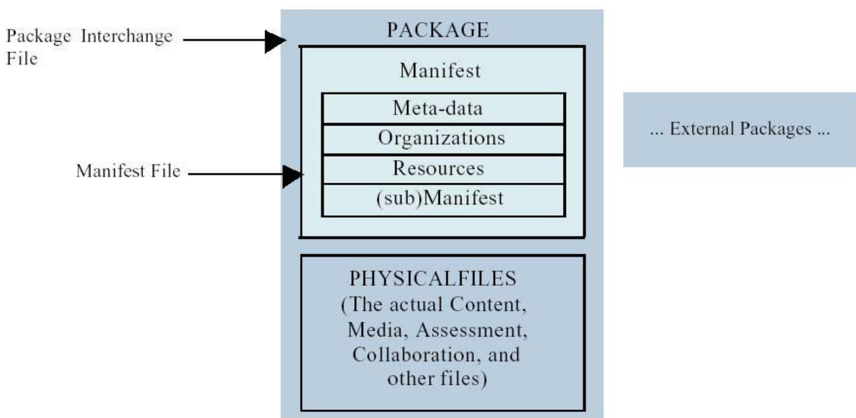


Figure 6.1 SCORM content packaging information model

The SCORM content packaging information model consists of three principal elements [55]:

- *Package*: A package represents a unit of reusable learning content. This may be part of a course that has instructional relevance outside of a course organization and can be delivered independently, as an entire course or as a collection of courses. Once a package arrives at its destination to a run time service, the package must allow itself to be aggregated or disaggregated into other packages. A package must be able to stand-alone. It must contain all the information needed to use the contents for learning when it has been unpacked.
- *Manifest*: A manifest is a description of the resources comprising meaningful instruction. It may also contain zero or more static ways of organizing the instructional resources for presentation. A manifest can describe part of a course that can stand by itself outside of the context of an instructional object of a course, an entire course, or a collection of courses. A package always contains a single top-level manifest that may contain one or more sub-manifests. The top-level manifest always describes the package. Any nested sub-manifests describe the content at the level to which the sub-manifest is scoped, such as a course, instructional object or other.
- *Resource*: The resources described in the manifest are physical assets such as web pages, media files, text files, assessment objects or other pieces of data in file form. Resources may also include assets that are outside the package but available through a URL, or collections of resources described by sub-

manifests. The combination of resources is generally categorized as content. Each resource may be described within a manifest, including a list of all the assets required to use the resource.

For the design of the SCORM-conformant content packaging, a critical issue is to determine the content structure and then represent the content structure through the SCORM content packaging application profile. With regard to Info1, it is structured with four aggregation levels: the course (Content Aggregation), lesson (Content Aggregation), course unit (SCO or SCA) and raw materials (Assets). In Figure 6.2 we illustrate part of the SCORM content packaging application profile that describes the course structure of Info1. Based on the application profile, Info1 can be presented by any SCORM-conformant LMSs, as depicted in Figure 6.3.

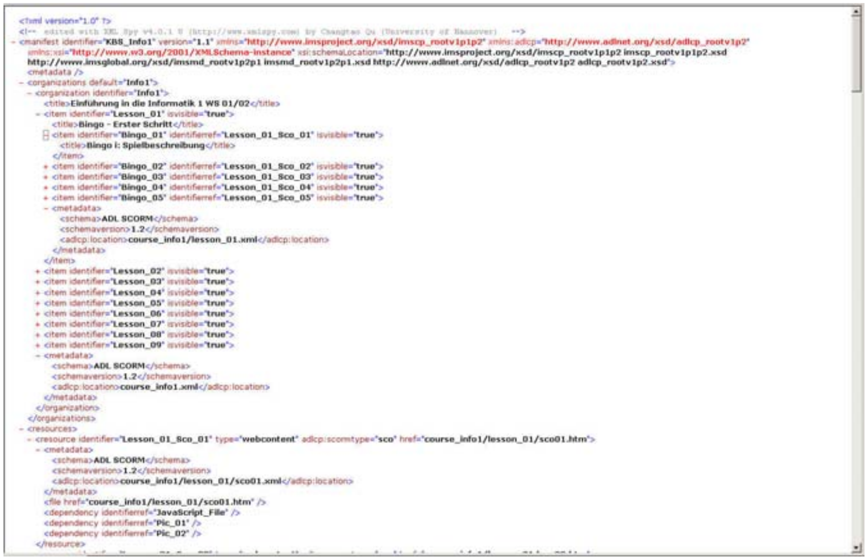


Figure 6.2 SCORM content packaging application profile of Info1

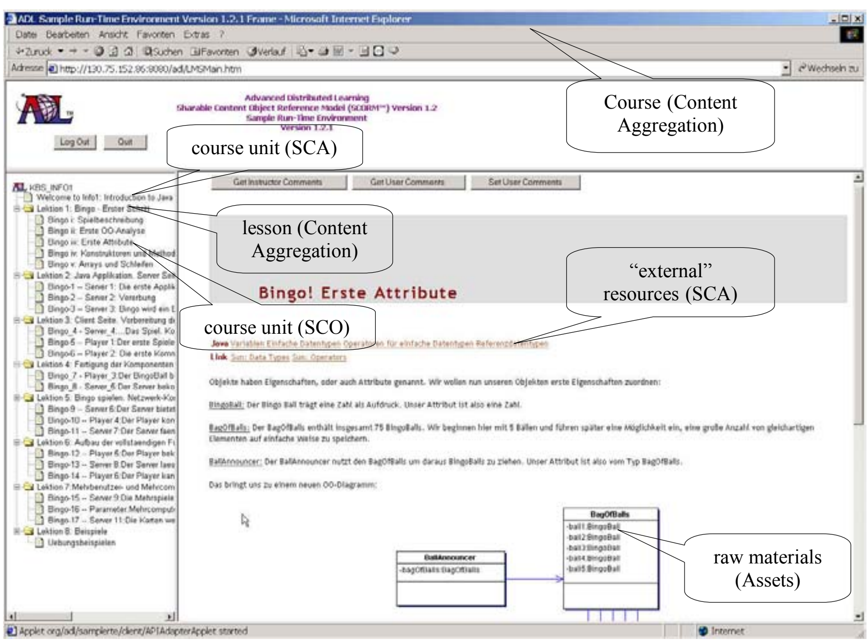


Figure 6.3 SCORM-conformant content structure of Info1

As the SCORM content packaging application profile defines the SCORM content structure in a self-contained way, including all descriptions of dependency and relationships existing between learning resources, not only those "internal" resources physically existing in a package and described by URI but also those "external" resources existing on the Web and described by URL, the SCORM content structure can easily be imported or exported, either partially or entirely, by any SCORM-conformant LMSs. Such sort of exchange, namely, importing, exporting, aggregating or disaggregating packages of learning content, can greatly improve reusability and portability of learning content.

6.2.3. SCORM-conformant Content Sequencing

The SCORM content sequencing is based on the same content organization and tree structure defined by the SCORM content packaging specification. The kernel of the SCORM content sequencing is the SCORM SDM, which is directly derived from the IMS SS SDM. The SCORM SDM defines a set of elements that can be used by content developers to define intended sequencing behaviour of learning content at authoring time. The SCORM-conformant LMSs must be able to support the behaviours that result from the definition of all of these elements, and deliver associated learning activities in a sequence determined at run-time according to the progress made by the learner in previous learning activities, learner intention and the authored sequencing directions.

The SCORM SDM defines following categories [56]:

- *Sequencing control modes*: The sequencing control modes are defined to allow content developers to control the sequencing behaviour for a cluster. Multiple control modes can be enabled simultaneously for a cluster to create combinations of sequencing control behaviours. In the SCORM SDM, four sequencing control modes are defined: “choice”, “choice exist”, “flow” and “forward only”.
- *Sequencing rules*: The sequencing rules consist of a set of conditions and a corresponding action or behaviour that is performed if the set of conditions evaluates to be true. The sequencing rule conditions are based on the tracking model of an activity including objective progress information, activity/attempt progress information, or on the status of an activity with regard to defined limit conditions.
- *Limit conditions*: The limit conditions are defined to control certain circumstances regarding if an activity is allowed to be delivered. A limit condition is a pair value: a control variable used to indicate whether or not a particular limit condition has been defined by the content developer, and the limit containing the specified limit condition. In the SCORM SDM, seven control variables are proposed for the limit conditions: “attempts”, “attempt absolute duration”, “attempt experienced duration”, “activity absolute duration”, “activity experienced duration”, “begin time limit” and “end time limit”.
- *Auxiliary resources*: The auxiliary resources are associated with an activity to provide the learner with additional services or resources. The SCORM SDM uses resource ID and purpose to enable the auxiliary to be associated with a given activity. The resource ID is the identifier of the auxiliary resource. The purpose indicates the purpose of the identified auxiliary resource.
- *Rollup rules*: The rollup is defined as the process of evaluating the objective progress information and attempt progress information data for a set of child activities to determine the objective progress information and attempt progress information data for the parent. The rollup rules can be defined for an activity to control its rollup behaviours. The evaluation of rollup rules may alter the default sequencing path or may affect the availability of the activity and/or its sub-activities for delivery.
- *Rollup controls*: The rollup controls enable to conditionally restrict what information an activity provides for rollup. The rollup controls include descriptions of the types of rollup behaviours specified for an activity. In the SCORM SDM, three types of tracking model information are permitted to be involved in the rollup process: “objective satisfaction”, “objective measure” and “activity completion status”.
- *Objectives*: The objectives provide a mechanism to associate learning objectives with an activity. Each learning activity may have the unlimited number of learning objectives defined. In the SCORM SDM, each objective is described by four elements: “objective ID”, “objective satisfied by measure”, “objective minimum satisfied normalized measure” and “objective contributes to rollup”.
- *Objective map*: The objective map defines a mapping of an activity’s local objective information to and from a shared global shared objectives. In the SCORM SDM, the objective mapping is the key enabler for sharing objective information between activities.
- *Selection controls*: The selection controls include descriptions of how the children of an activity should be selected during the sequencing process. They enable to define sequencing rules that indicate when to select certain activities and limit the number of activities to be chosen.
- *Randomisation controls*: The randomisation controls include descriptions of whether and how child activities in a cluster of activities should be randomly sequenced when a new attempt is initiated for the parent activity of the cluster. They enable to define sequencing rules that indicate whether or not a sequencing engine implementation shall randomly select activities for delivery.
- *Delivery controls*: The delivery controls describe actions and controls used when an activity is delivered. These controls indicate if objective, activity and attempt progress data are recorded when the activity is delivered. In the SCORM SDM, there types of controls are defined: “tracked”, “completion set by content” and “objective set by content”.

For the design of the SCORM-conformant content sequencing, a critical issue is to design sequencing behaviours of learning activities and further associate each learning activity with the sequencing information through the SCORM SDM. The process of defining a specific sequence of learning activities begins with the creation of an aggregation of content to be interchanged using the SCORM content packaging application profile. In the SCORM content packaging application profile, the sequencing information can be associated with either an item element, or a collection of item elements within an organization element, or an organization element itself, as illustrated in Figure 6.4 [2][56].

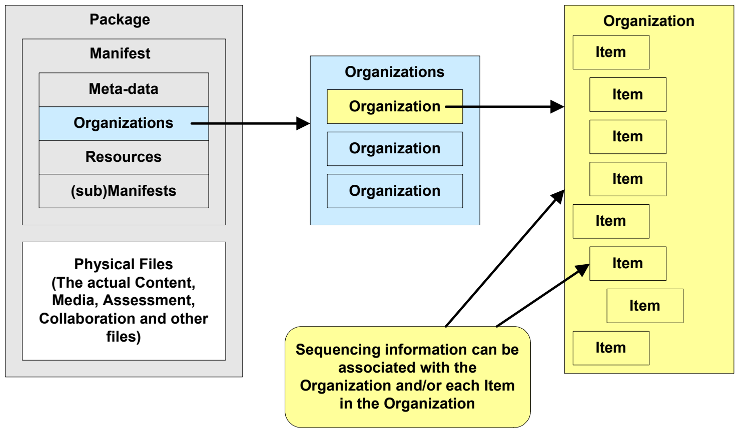


Figure 6.4 Sequencing information associated with the SCORM content packaging application profile

Taking Info1 as an example, in order to present the design of its SCORM-conformant content sequencing, we first illustrate the graphical course structure of Info1 according to its SCORM content packaging application profile (see also section 6.2.2), as depicted in Figure 6.5.

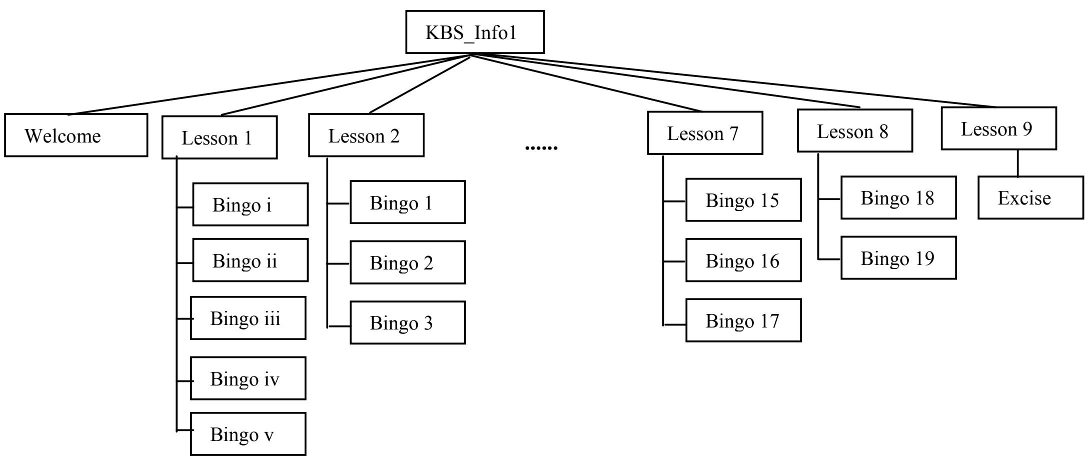


Figure 6.5 Graphical course structure of Info1

The design of the SCORM-conformant content sequencing in Info1 is mainly focused on improving adaptability of the learning content delivery. Adaptability here means the dynamic, predictable and consistent ordering and delivery of learning activities in an instructionally meaningful manner based on the learner's interactions. For simplicity reasons, we propose a simple adaptive sequencing scenario: the performance-based adaptive sequencing, to demonstrate the design of the SCORM-conformant content sequencing in Info1. The implementation of the performance-based adaptive sequencing includes following procedures:

- Associate "Lesson 9: Excise" with Lesson 1-8 through the SCORM SDM objective map definition.
- The learner is first presented with "Lesson 9: Excise", which can evaluate the learner's mastery of Info1.
- Based on the evaluation, the SCORM RTE will dynamically deliver part of Lesson 1-8 whose objectives have not yet been satisfied.

In Figure 6.6 we illustrate the new SCORM-conformant course structure of Info1, in which the content of "Lesson 9: Excise" is somewhat enriched and its delivering location is also moved forward in order to act as a pre-test for the whole Info1. In order to realize the performance-based adaptive sequencing, we define sets of sequencing rules for Content Aggregations and SCOs based on the SCORM SDM.

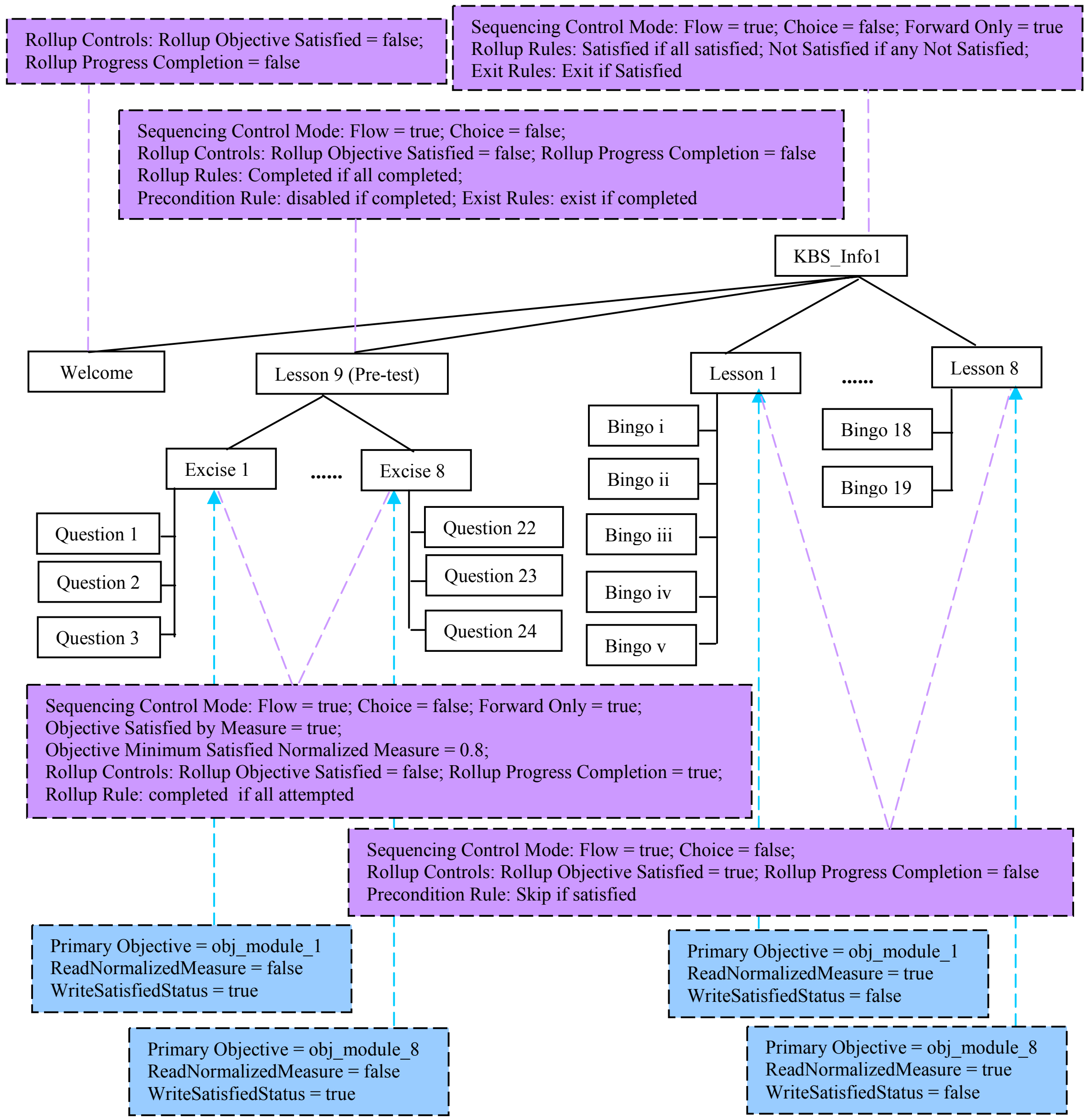


Figure 6.6 SCORM-conformant content sequencing in Info1

Info1 itself is a organization that employs "forward only" sequencing control mode to control the delivery of "Welcome"(SCA) and nine lessons (Content Aggregations). Its rollup rules are defined as: "*if all child activities' objectives are satisfied, its objective is satisfied, and if any child activity's objective is not satisfied, its objective is not satisfied*". Note should be paid that according to the rollup controls defined for "Welcome" and Lesson 1-9, only Lesson 1-8 affect the rollup behaviours of Info1.

When the learner logs to Info1, he is first presented with "Welcome" and then Lesson 9, which serves as the pre-test of Info1. In Lesson 9, we associate the learning objectives of Excise 1-8 to the objectives of Lesson 1-8 through the SCORM SDM objective map definition. As Excise 1-8 share the same objectives with Lesson 1-8, if the learner can pass any of Excise 1-8, it is supposed that he has understood the knowledge contained in corresponding lessons. It is worth noting that according to the definition of "*WriteSatisfiedStatus*", only Excise 1-8 can set up the global objectives. Lesson 1-8 can only read the global objectives to determine whether they should be delivered or not. They cannot set up the satisfaction status of the global objectives.

For each of Excise 1-8, it employs "forward only" sequencing control mode to control its underlying Questions. These Questions are IMS QTI compliant SCOs responsible for evaluating the learner's performance. Each Excise defines "*objective minimum satisfied normalized measure equals to 0.8*", which means that if the learner can achieve a score higher than 0.8 from three Questions in the Excise, the objective of the Excise can be judged to be satisfied and the corresponding global objective can be set as satisfied. After the learner has gone through all of Excise 1-8, the SCORM RTE will dynamically deliver Lesson 1-8 according to the learner's performance. Here the key to the dynamic delivery of Lesson 1-8 is the pre-condition rule defined for each of lessons. In terms of the pre-condition rule, the SCORM RTE will determine whether the lesson should be delivered or simply skipped at time of delivery based on the satisfaction status of the global objectives.

6.2.4. SCORM-conformant Run-time Environment

The SCORM RTE provides a common way to start learning content, a common mechanism for learning content to communicate with a LMS, and a pre-defined language or vocabulary forming the basis of the communication. These three functionalities of the SCORM RTE are accomplished by three SCORM RTE components: the launch, API and data model, as illustrated in Figure 6.7 [2].

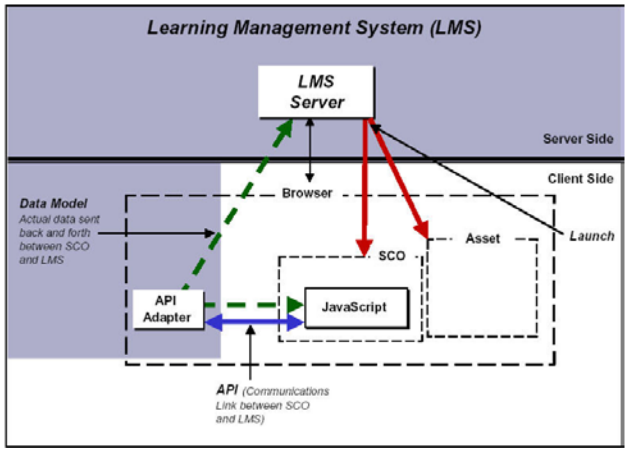


Figure 6.7 Launch, API and data model in the SCORM RTE

SCORM RTE Data Model. The SCORM RTE data model specifies the information that may be communicated between learning content and a run-time service to enable a LMS to maintain a communication link with the client. In Figure 6.8 we illustrate the conceptual diagram of the SCORM RTE data model [2][52].

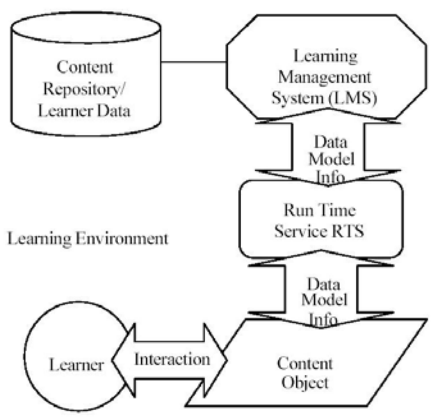


Figure 6.8 Conceptual diagram of the SCORM RTE data model

The SCORM RTE data model is the kernel of the SCORM RTE. It is purposed to make sure that a defined set of information about SCOs can be tracked by different LMS implementations. There are some key relationships between certain SCORM RTE data model elements to the tracking model and activity state model for the corresponding activities, which can enable a LMS to control the sequencing between these activities in a consistent way. As the SCORM makes all SCORM RTE data model elements mandatory, from the perspective of LMSs, there exist no optional data model elements any more. For content developers, the principal design issue regarding the SCORM RTE data model is to choose adequate data model elements for a SCO to implement functionalities expected. Taking Info1 as an example, as it is a lecture-centred courseware, we mainly make use of two data model elements in a SCO to improve the interaction between learners and instructors:

- *cmi.comments*: The ability for a SCO to get/set comments. This data model element is used in Info1 to enable learners to add comments to SCOs.
- *cmi.comments_from_lms*: The ability to provide read-only comments to a SCO. This data model element is used in Info1 to enable instructors to add guidance to SCOs.

SCORM RTE API. The SCORM RTE API provides the communication mechanism for informing a LMS of the state of learning content (e.g., initialised, finished or in an error condition), and for getting and setting data between the LMS and the SCO through run-time services in terms of the SCORM RTE data model. In Figure 6.9 we illustrate the conceptual model of the SCORM RTE API [2][53].

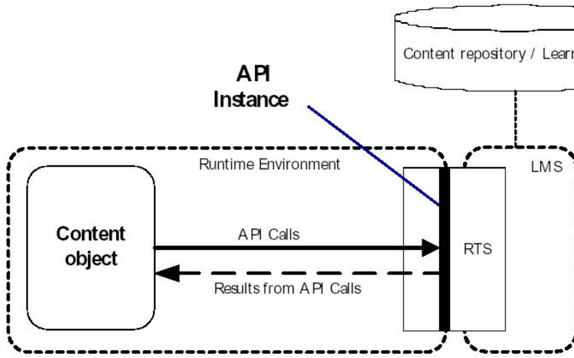


Figure 6.9 Conceptual model of the SCORM RTE API

The SCORM RTE hosts a run-time service that delivers a content object to the learner and starts it. This run-time service is then responsible for handling and behaving in response to calls from a content object. The behaviour of the API implementation in the run-time service is described in terms of the state transitions for each of the events. During the communication, both SCO and the run-time service leverage three kinds of methods to communicate the SCORM RTE data model information [53]. The calling conventions to these methods are specified by the ISO/IEC 16262:1998 [59].

- *Session methods*: Used to mark the begin and end of a communication session between a SCO and the run-time service through the API instance.
- *Data transfer methods*: Used to transfer data model values between a SCO and the run-time service through the API instance.
- *Support methods*: Used for auxiliary communications (e.g., error handing) between a SCO and the run-time service through the API instance.

For content developers, the principal design issue regarding the SCORM RTE API is to incorporate in the content object the capability to discover and communicate with an API instance. This API instance is usually instantiated as an object within the DOM environment [48] of the content object. In Info1 we develop sets of JavaScript functions through which each SCO can initiate and bind the API instance when it is delivered by the run-time service.

SCORM RTE Implementation. Any conforming SCORM RTE implementation has to support all data model elements. It has to implement a function module to support the interpretation of the SCORM content packaging application profile, and a function module to support the SCORM content sequencing in conjunction with the data model implementation. As the SCORM requests the fully Web-based learning content delivery, the implementation of the SCORM RTE also has to be based on sets of standard Web technologies and metadata technologies. In Info1 we propose a SCORM RTE implementation based on the JSP & Servlet enabled Web server: Apache Tomcat 3.2.3, as illustrated in Figure 6.10.

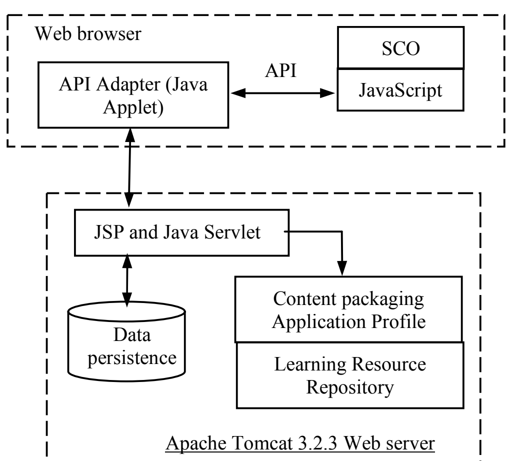


Figure 6.10 SCORM RTE implementation

On the server side, a JSP component is used to dynamically render the SCORM content packaging application profile into series of hyperlinks whose targets contain the corresponding launching locations of SCOs. Besides, there are also several Java Servlet components that are responsible for controlling sequencing of SCOs, handling the communication between the SCORM RTE and SCOs, and managing persistence of the SCORM RTE data model. While implementing the SCORM RTE data model, we directly employ the CMI data model Java binding API provided by AICC.

On the client side, a non-face Java Applet is implemented as the SCORM RTE API adapter, which provides the communication to the server-side Servlet components for the persistence management of the data model. On the client side, SCOs cannot make direct communication with the SCORM RTE server to call API functions. All calls from SCOs must take the API adapter as a broker and use client-side JavaScript.

As the SCORM-conformant learning content communicate with the SCORM RTE in a standard manner, they can be delivered in any SCORM conformant LMSs regardless of different technical implementations. As an example, in Figure 6.11 we show the SCORM-conformant Info1 delivered in Microsoft LRN 3.0[40].

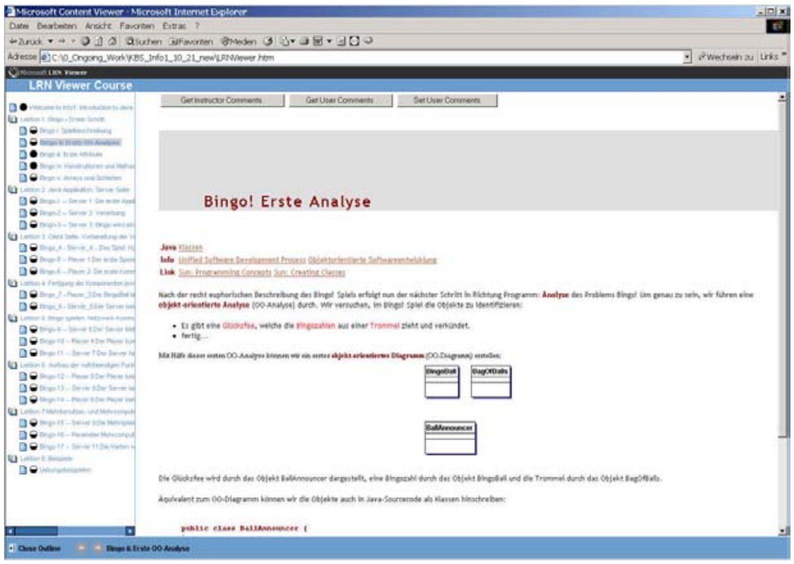


Figure 6.11 SCORM-conformant Info1 delivered in Microsoft LRN 3.0

6.3. Towards the Standard-oriented AEHS: Converging the E-Learning Standardization and Conventional AEHS Research

With regard to the learning content delivery, the E-Learning standardization is more focused on improving reusability and portability of learning content, aiming to provide learning content that are deliverable independent of different delivery environments. In particular, with the introducing of the content sequencing specifications such as the IMS SS, the E-Learning standardization can further support somewhat adaptability of the learning content delivery thus possesses a considerable overlap with the AEHS research.

The AEHS is another large research and development stream concerning the learning content delivery. It emerges as an alternative to the traditional "one-size-fits-all" approach in the development of educational courseware. AEHSs build a model of the goals, preferences and knowledge of each individual learner, and use this model throughout the interaction with the learner in order to adapt to the needs of that learner [17]. Unlike the E-Learning standardization, the AEHS research is more focused on adaptability of the learning content delivery. It depends on various AH [18] and ITS [82] technologies to implement the adaptive learning content delivery, which can select the most relevant learning content and present them to the learner at the right time and in the right way. As AEHSs usually adopt proprietary technologies to realize the learning content organization and run-time environment implementation, it is quite hard or even impossible to ensure reusability and portability of learning content in conventional AEHSs.

One key to the change of such a situation is to apply the E-Learning standardization to the conventional AEHS. As we can see from the reference model proposed for the standard-oriented learning content delivery, the E-Learning standardization can address almost all aspects of a learning content delivery process, which provides the possibility to build the standard-oriented AEHS that is able to converge the E-Learning standardization and conventional AEHS research to ensure reusability and portability of learning content while achieving

[40] http://www.microsoft.com/elearn/

adaptability of the learning content delivery. In contrast to conventional AEHSs, which only focus on the complex adaptability implementation in a stand-alone system using the proprietary design to fulfil specific, often domain-dependent requirements [19], the standard-oriented AEHS puts reusability and portability of learning content at the first order, providing a fully standardized approach for implementing AEHSs.

6.3.1. Reference Model for the Standard-oriented AEHS

A typical AEHS can logically be decomposed into four basic components [47]:

- *Document space*: The document space associates information about all hypermedia documents contained in an AEHS as well as relationships between these documents. The associated information might be annotations such as metadata attributes, usage attributes, etc., or domain graphs that model the document structure, or knowledge graphs that describe the knowledge contained in the document collections.
- *User model*: The user model stores, describes and infers information, knowledge, preferences etc., about an individual learner. The user model might be updated by observations during the learner's interactions with an AEHS.
- *Observations*: The observations provide information about the learner's interactions with an AEHS. Here everything about the run-time behaviours of the system concerning the learner's interactions is included. Examples are observations whether a learner has visited a document, or visited document for some amount of time, etc.
- *Adaptation component*: The adaptation component defines rules for adaptive functionality, e.g., whether to suggest a document for learning, or for generating reasonable learning paths, etc., as well as rules for adaptive treatment, e.g., sorting the links leading to further documents according to their usefulness for a particular learner, etc.

In terms of the reference model proposed for the standard-oriented learning content delivery, we can standardize all of four AEHS components to build the standard-oriented AEHS. In Table 6.1 we list E-Learning standards and specifications applicable to the standard-oriented AEHS.

Table 6.1 E-Learning standards and specifications applicable to the standard-oriented AEHS

AEHS Components	Applicable E-Learning Standards and Specifications
Document Space	IMS CP, IMS SS, IMS QTI ADL SCORM
User Model & Observations	IMS SS IEEE Data Model /CMI001- AICC/CMI Guidelines for Interoperability IEEE API /CMI001- AICC/CMI Guidelines for Interoperability ADL SCORM
Adaptation Component	IMS SS ADL SCORM

Standardization of the AEHS Document Space. Learning content contained in the AEHS document space are usually beyond simple learning objects such as a text file, an image, etc. In most cases these learning content exist in the form of learning activities, which may aggregate several lower-level learning objects and also associate with sequencing information to realize adaptability of the learning content delivery [19]. In conventional AEHSs, these learning activities are mostly bound to proprietary descriptions and can only be delivered through the dedicated run-time environments. It is rather difficult to reuse these learning activities in another delivery context.

The standardization of the AEHS document space has to address three critical issues. First, the learning activities have to be characterized in a standard way. Here we propose to use the IMS SS to address the issue, which characterizes the learning activity as "an instructional event or events embedded in a content resource, or as aggregation of activities that eventually resolve to discrete content resources with their content instructional events"[56]. Actually, such a characterization of learning activities can further be complemented by the IMS CP, which defines a standard structure to describe learning activities and their hierarchical organizations (activity tree) to enable the reuse and exchange of learning activities at different aggregation levels. For the discovery of the learning activities and activity trees, we propose to use the ADL SCORM, which provides a standard way to describe learning activity metadata based on the IEEE LOM.

Second, the learning activities have to be associated with the sequencing information in a standard way. In order to ensure reusability and portability of learning content, the sequencing information should also be external from learning activities themselves. Here we propose to use the IMS SS to address the issue, which provides a standard way to associate sequencing information with learning activities through the IMS SS SDM. As the IMS SS can also handle the sequencing information independent of learning activities by means of the IMS SS TM and IMS SS ASM, it can essentially ensure reusability and portability of learning activities.

Third, as the AEHS document space usually has to include test assignments for evaluating the learner's performance [47], it is necessary for the AEHS document space to build and organize the test assignments in a standard way. Here we propose to use the IMS QTI to address the issue, which describes a basic structure for the representation of question and test data and their corresponding results reports to enable the exchange of the question, test and results data between different LMSs and educational repositories [57].

Standardization of the AEHS User Model and Observations. The standardization of the AEHS user model and observations is closely related to the design of run-time environments in the standard-oriented AEHS. In contrast to conventional AEHSs, in which arbitrarily complex user model can be built and observed by the proprietary run-time environment implementations, in the standard-oriented AEHS the user model must be built in the way that it is independent of run-time environment implementations thus can be observed and managed by any conforming run-time environments. This implies that the design of the AEHS user model and observations must comply with the design of run-time environments in the standard-oriented AEHS.

In terms of the reference model for the standard-oriented learning content delivery, we propose to use the IEEE Data Model and IEEE API to construct standard run-time environments in the standard-oriented AEHS. Among them, the IEEE Data Model is taken as the basis for the standardization of the AEHS user model and observations. On the one hand, the IEEE Data Model defines sets of data elements to describe learner information and learner performance, based on which a simple user model can be built. On the other hand, it also provides the mapping to the IMS SS TM and IMS SS ASM, through which the major data elements of the IEEE Data Model can be observed by the standard run-time environments. Moreover, the IEEE Data Model based communication between learning content and run-time environments is further supported by the IEEE API. As we can imagine, the standardized communication is also the key to the standardization of the AEHS user model and observations.

Standardization of the AEHS Adaptation Component. As currently the IMS SS is the only E-Learning specification that is able to accomplish somewhat adaptability of the learning content delivery, we propose to use the IMS SS to implement the AEHS adaptation component. In comparison to the adaptation components in conventional AEHSs, which are mostly implemented based on proprietary run-time environments supporting various adaptation algorithms, the adaptation components implemented based on the IMS SS are somewhat backward. This is because that the IMS SS is quite simple in the sense that it can only cover a limited number of widely used sequencing behaviors and does not address some advanced sequencing behaviours, e.g., artificial intelligence based sequencing, schedule-based sequencing, etc., and also because that the AEHS user model and observations in the standard-oriented AEHS, which serve as the design and implementation basis of the adaptation components, are relatively simple thus cannot provide comprehensive information about the learner's interactions. However, despite of the retrogress of adaptability, the standardization of the AEHS adaptation component leads to a notable advantage. That is, in the standard-oriented AEHS, the adaptation components are no longer bound to the run-time environments. They can now be reused and exchanged between any conforming run-time environment implementations.

6.3.2. Design Comparison between the Standard-oriented AEHS and Conventional AEHS

From the viewpoint of the learning content management, reusability and portability of learning content can usually be viewed as the opposite of adaptability of the learning content delivery. Because the accomplishment of advanced adaptability has to depend on more specific designs of run-time environments and more complex control over the content sequencing, the more advanced adaptability the learning content delivery may achieve, the less reusability and portability the learning content may have. Putting reusability and portability of learning content at the first order, the standard-oriented AEHS cannot be expected to be able to achieve comparable adaptability as conventional AEHSs. However, we may expect that the standard-oriented AEHS can achieve a balance between reusability, portability and adaptability, being able to ensure reusability and portability of learning content while achieving satisfactory adaptability of the learning content delivery.

In order to present the design of the standard-oriented AEHS, in this section we take a typical conventional AEHS: NetCoach as a reference system to conduct a design comparison between the standard-oriented AEHS and conventional AEHS. NetCoach is an authoring system that allows to create adaptive course modules [113]. It is directly derived from ELM-ART [114], one of the first and by now most comprehensive AEHSs. The design and implementation of NetCoach cover most of AH and ITS techniques commonly used in conventional AEHSs [47]. In the comparable design of the standard-oriented AEHS, we do not expect to implement all adaptation features of NetCoach. However, we can show that some of functionalities of NetCoach regarding the document space, user model, observations and adaptation component can be realized in the standard-oriented AEHS.

Standard-oriented AEHS Document Space Vs. Conventional AEHS Document Space. The document space is the key to the design of both standard-oriented AEHS and conventional AEHS. It provides the basis for the implementation of the AEHS user model, observations and adaptation component, and ultimately determines adaptability of AEHSs. In comparison to the conventional AEHS document space, the standard-oriented AEHS document space has two features that put more limitations to the standard-oriented AEHS design.

First, the standard-oriented AEHS document space can only be organized into a learning activity tree. This implies that the learning activities in the standard-oriented AEHS can only be delivered according to the pre-defined hierarchical structure, and it is impossible to dynamically generate delivery sequence of learning activities at run-time. During delivery of the learning activities, a node of the activity tree can be disabled or skipped to get out of the pre-defined delivery sequence but cannot be delivered in a sequence against the pre-defined hierarchical structure of the activity tree. This feature determines that the standard-oriented AEHS document space is not comparable to an important type of conventional AEHS document spaces, which depends on a conceptual model to dynamically generate the document delivery sequence thus does not possess a pre-defined document hierarchy, e.g., the document space of the KBS Hyperbook [46][87]. For another important type of conventional AEHS document spaces, which organizes documents in terms of the hierarchical structure, e.g., the document space of NetCoach, ELM-ART and Interbook [20], etc., the standard-oriented AEHS document space can partially achieve the comparable design.

Second, the standard-oriented AEHS document space can only define a limited number of adaptation behaviours for learning activities by means of the IMS SS SDM, which is the only applicable static data model for describing authored sequencing intentions for a given learning activity. In the IMS SS SDM, the sequencing rules can only be associated with the cluster of learning activities. As the scope of a particular rule never extends beyond the cluster, it is quite hard to associate two activities in different clusters to describe some complex sequencing or navigation behaviours. In the IMS SS SDM, the only mechanism for associating different learning activities is the objective map. As each learning activity may have an unlimited number of learning objectives and objective maps, we can define a little bit complex sequencing behaviours in the standard-oriented AEHS document space through objective maps. However, as objective maps usually become rather complicated with the increase of the complexity of sequencing behaviours, they cannot be used to define all types of sequencing behaviors that are implementable in conventional AEHS document spaces.

As a typical conventional AEHS document space adopting the hierarchical document structure, the NetCoach document space consists of three types of basic constructs: documents, test-groups and test-items [47][113]. The documents in the NetCoach document space are structured hierarchically in a section, sub-section, sub-sub-section manner. This hierarchical structure delivers information for adaptation by giving for each document a predecessor and successor in the document space. There are four kinds of relations between documents: the prerequisite-relation, inference-relation, successor-relation and part-of-relation. Additionally, there is also a flag "terminal-page" attached to each document indicating whether this document is a terminal page, a "criterion" that defines the number of tests necessary to learn a document, and a "test assignment" that relates some test items or test groups to a document [113]. In Figure 6.12 we illustrate the hierarchy and relations of documents in the NetCoach document space [47][113].

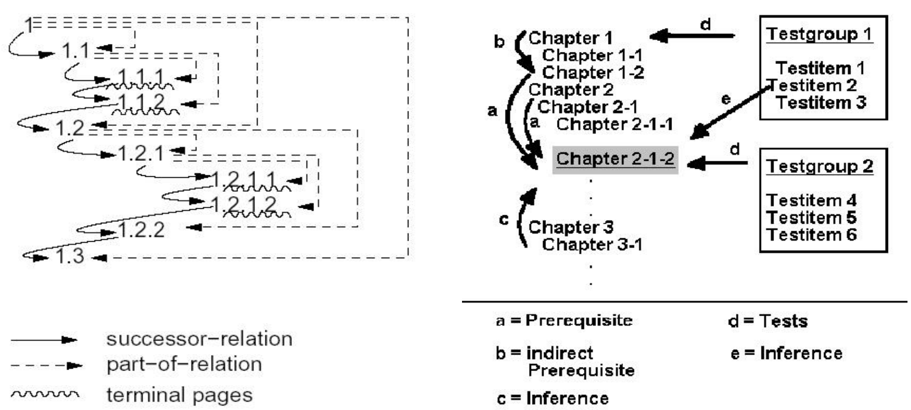


Figure 6.12 Hierarchy and relations of documents in the NetCoach document space.

In the comparable design of the standard-oriented AEHS document space, the hierarchical structure of documents can be represented through the IMS CP. From the IMS CP application profile, several structure-level relations including the successor-relation, part-of-relation and the "terminal-page" flag can directly be inferred.

The inference-relation can be defined through the IMS SS SDM objective maps. Suppose a document *Di* can be inferred to be learned whenever a document *Dj* has been learned, *inference(Di, Dj) for certain Di ≠ Dj,* the document *Di* can first define sets of local objectives *OIij (WriteSatisfiedStatus: false, ReadSatisfiedStatus:*

true), which are respectively mapped to the global objective *OIGj*. The document *Dj* can also define a local objective *OIj (WriteSatisfiedStatus: true, ReadSatisfiedStatus: false)*, which is mapped to the global objective *OIGj*. As *OIij* and *OIj* share the same global objective *OIGj,* the inference-relations between *Di* and *Dj* can be built. In *Di*'s pre-condition rules, *OIij* will first be judged according to the global objective *OIGj,*

If OIij satisfied, then pre-condition actions

Note should be paid that the local objectives *OIij* and *OIj* as well as the global objective *OIGj* are defined in the IMS SS SDM with the unique purpose to represent the inference-relation between *Di* and *Dj*. They are related with each other through the objective map definitions. In the IMS SS TM, these objective map data are evaluated whenever local objective information is changed.

The prerequisite-relation is used by NetCoach to define which other documents are required to be learned to understand the current document. Although the prerequisite-relation and inference-relation are somewhat interrelated, the definition of the prerequisite-relation in the IMS SS SDM is more complicated, since the actions corresponding to the inference-relation are conducted based on the judgement on a single document objective, whereas the actions corresponding to the prerequisite-relation are conducted based on the judgement on sets of document objectives. Suppose the document *Dj* needs to be learned before a learner can learn *Di*, *prerequisite (Di, Dj) for certain* $Di \neq Dj$, the document *Di* can first define sets of local objectives *OPij (WriteSatisfiedStatus: false, ReadSatisfiedStatus: true)*, which are respectively mapped to sets of global objectives *OPGij*. Each *Dj* can define a local objective *OPj (WriteSatisfiedStatus: true, ReadSatisfiedStatus: false)*, which is mapped to one global objective *OPGij*. In *Di*'s pre-condition rules, sets of local objectives *OPij* will first be judged as a whole according to the global objectives *OPGij*,

if all OPij satisfied, then pre-condition actions

if any OPij unsatisfied, then pre-condition actions

It is worth noting that here the pre-condition actions are defined and implemented by the standard-oriented AEHS adaptation components. Based on the same IMS SS SDM, it is possible to build different adaptation components in order to accomplish different kinds of adaptation behaviours.

In the NetCoach document space, the indirect prerequisite-relation can directly be inferred from the prerequisite-relations. However, in the standard-oriented AEHS document space, such a direct inference is not supported by the IMS SS SDM objective maps. Moreover, due to the rapidly increasing complexity, it is also quite hard to explicitly define the indirect prerequisite-relations in the standard-oriented AEHS document space through the IMS SS SDM objective maps. As a result, the comparable design of the indirect prerequisite-relation cannot be achieved in the standard-oriented AEHS document space.

Besides documents, the NetCoach document space also contains test-groups and test-items used to evaluate the learner's performance. In order to improve reusability and portability of these tests, the original test-groups and test-items can be re-structured in the standard-oriented AEHS document space as the "assessment", "section" and "item" according to the IMS QTI ASI information model. Sets of test-items associated with a learning activity can explicitly be organized into a test-group, which shares the same global objective with the learning activity through the IMS SS SDM objective maps. The evaluation rules of the learner's performance on the test-group can be defined through the "*Objective Minimum Satisfied Normalized Measure*". According to the IMS QTI Result Reporting Information Model, each test-item can report the learner's score in a standard way. These reports are then evaluated by the test-group according to the pre-defined sequencing rules to judge whether the learner has understood the document or not. In the NetCoach document space, test-items may not only test one document but also assess aspects of other documents to enable the quantification of the inference of test-items to other documents [113]. Such kind of an inference-relation cannot be defined in the standard-oriented AEHS document space, since a test-item can only explicitly be associated with a learning activity. It is impossible to evaluate the learner's performance on a learning activity based on the test-item that is associated with another learning activity.

In comparison to the conventional AEHS document space, the learning activities contained in the standard-oriented AEHS document space can be reused at different aggregation levels. Since the IMS CP provides a very flexible way to define content aggregations, the IMS SS conformant content sequencing definition can be applied to different levels of content aggregations to support sets of learning activities to accomplish a particular learning task that may contain internal logic such as branching within a content aggregation depending on the learner's interactions. If the internal branching does not reference external learning resources that may or may not be present in other content aggregations, these branching can be self-contained defined thus reusable as a whole. In addition, the standard-oriented AEHS document space also provides a major improvement to the conventional courseware development. In the past, in order to build a conventional AEHS document space, all authoring tools typically have to embed all of the proprietary and sometimes unique course sequencing and navigation information that governs what part of the course the learner will view next in proprietary data formats [56]. Now in the standard-oriented AEHS document space, the content sequencing information is uniquely based on the IMS SS and external from learning content. This enables the share of learning content between different authoring environments to reuse learning content in other contexts that involve different sequencing requirements.

Standard-oriented AEHS User Model and Observations Vs. Conventional AEHS User Model and Observations. The basis of the standardization of the AEHS user model and observations is the IEEE Data Model. In the standard-oriented AEHS, the IEEE Data Model elements concerning the user model and observations design are principally managed by two dynamic run-time data models defined in the IMS SS: the IMS SS TM and IMS SS ASM. The IMS SS TM is used to capture information gathered from the learner's interactions with learning resources associated with activities. The IMS SS ASM is used to manage sequencing state of each activity in the activity tree and the global state of the activity tree [56]. Since the IMS SS ASM is mainly used by sequencing engines of LMSs to manage the state of the activity tree during a sequencing session, the IMS SS principally depends on the IMS SS TM to implement the user model and observations functionalities in the standard-oriented AEHS.

The IMS SS TM is a collection of dynamic, sequencing state information associated with each node of the activity tree. It includes two types of information that must be maintained by a system that delivers sequencing activities [56]:

- *Objective progress information*: Results of the learner's interactions related to an objective.
- *Activity/Attempt progress information*: Describe the learner's progress on an activity. This information describes the cumulative progress information for an individual activity.

In Table 6.2, Table 6.3 and Table 6.4 we respectively list the objective progress information, activity progress information and attempt progress information.

Table 6.2 Objective progress information defined in the IMS SS TM

Name	Description	Value Space	Default Value
Objective Progress Status	Indicates the objective has a satisfaction value	Boolean	False
Objective Satisfied Status	Indicates the objective is satisfied	Boolean	False
Objective Measure Status	Indicates the objective has a measure value	Boolean	False
Objective Normalized Measure	The measure (e.g., normalized score) for the objective	Real [-1.0..1.0]	0.0

Table 6.3 Activity progress information defined in the IMS SS TM

Name	Description	Value Space	Default Value
Activity Progress Status	Indicates the activity progress information is meaningful for the activity.	Boolean	False
Activity Absolute Duration	The cumulative duration of all attempts on the activity, i.e., the time from the initial start of the activity to the end of the activity.	Duration	0.0
Activity Experienced Duration	The cumulative experienced duration of all attempts on the activity, i.e., the time from the initial start of the activity to the end of the activity, not including any time elapsed while the activity is suspended.	Duration	0.0
Activity Attempt Count	The number of attempts on the activity.	Non-negative Integer	0

Table 6.4 Attempt progress information defined in the IMS SS TM

Name	Description	Value Space	Default Value
Attempt Progress Status	Indicates the attempt progress information is meaningful for the activity attempt.	Boolean	False
Attempt Completion Amount	The measure of the completion of the attempt on the activity.	Real [0..1]	0.0
Attempt Completion Status	Indicates the activity attempt is completed	Boolean	False
Attempt Absolute Duration	The duration of the attempt on the activity, i.e., time from the start of the attempt to the end of the attempt.	Duration	0.0
Attempt Experienced Duration	The experienced duration of the attempt on the activity, i.e., the time from the start of the attempt to the end of the attempt, not including elapsed time while the activity attempt is suspended.	Duration	0.0

The IMS SS TM can be viewed as a common vocabulary of status information on which run-time sequencing decisions can be based. It is associated with each node in the activity tree for each learner with initial values that are defined in the sequencing rules. During a learning experience, the state information of the IMS SS TM data items is updated as the learner interacts with the activities and learning resources. A LMS must be able to receive and maintain the tracking status information for each learning activity defined.

In order to implement the user model and observations functionalities in the standard-oriented AEHS, the IEEE Data Model elements concerning the user model and observations design must first be mapped into the IMS SS TM and IMS SS ASM data elements, through which the standard-oriented AEHS user model can be observed and further managed by run-time environments. Such sort of a mapping can be done following the ADL SCORM, which provides a solution to achieve the mapping between the IEEE Data Model and IMS SS TM/ASM [2].

The NetCoach user model uses a multi-layered overlay model to store individual information about each learner with respect to the documents. In Figure 6.13 we illustrate the NetCoach user model [113].

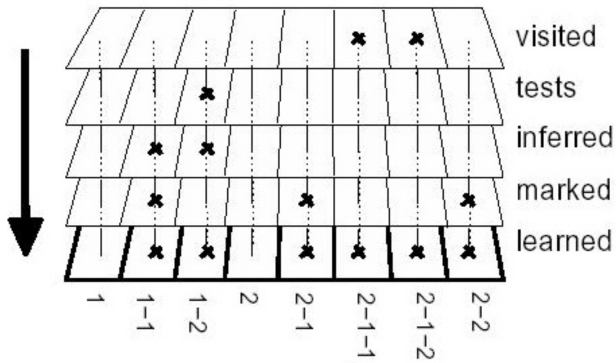


Figure 6.13 NetCoach user model

The first layer of the NetCoach user model describes whether the learner has visited a document. The second layer contains information about which test-items related to this particular document the learner has worked at, and whether the learner has successfully worked on the test-items up to a certain criterion. The third layer describes whether a document could be inferred as known via inference links from more advanced documents the learner has already successfully worked on. Finally, the fourth layer describes whether the learner has marked a document as already known.

NetCoach clearly differentiates between the user model and observations in that everything that is a direct observation about the learner's interactions with the LMS is modelled in observations, whereas all interpreted or processed observations are collected in the user model [47]. The different layers of the NetCoach user model are compiled by making observations about a learner (layer 1, 2 and 4) or by processing these observations (layer 3). As the standard-oriented AEHS user model and observations need to be considered as a whole, it is no longer necessary to differentiate between user model and observations functionalities in the standard-oriented AEHS. Both functionalities are implemented based on the same basis: the IEEE Data Model.

In the comparable design of the standard-oriented AEHS user model and observations, the first layer of the NetCoach user model can be observed through the IEEE Data Model element *completion_status*. According to the SCORM, this data model element is mapped to the "*Attempt Completion Status*" element defined in the IMS SS TM. Due to the nature of the IMS SS TM, the "*Attempt Progress Status*" element in the IMS SS TM is also affected.

The second layer of the NetCoach user model can be observed through the IEEE Data Model element *success_status*. According to the SCORM, this data model element is mapped to the "*Objective Satisfied Status*" element defined in the IMS SS TM. Due to the nature of the IMS SS TM, the "*Objective Progress Status*" element in the IMS SS TM is also affected. Note should be paid that in the standard-oriented AEHS, the *success_status* element is only applied to the test-groups or test-items. As we have mentioned, through the IMS SS SDM objective maps, these test-groups or test-items have already been related to corresponding documents by sharing the same learning objectives. Through tracking the test-groups and test-items we can directly observe and judge the success status of the documents.

For the third layer of the NetCoach user model, as we cannot define the inference-relation in the standard-oriented AEHS document space, we cannot infer the success status of a particular document from the success status of other documents. This implies that the third layer functionality of the NetCoach user model cannot be implemented in the standard-oriented AEHS user model and observations.

For the fourth layer of the NetCoach user model, the success status that is directly marked by the learner can also be observed and managed through the IEEE Data Model element: *success_status*. Supported by the IEEE API, the learner can directly edit this data model element. However, in the standard-oriented AEHS user model and observations, we cannot differentiate how the *success_status* is set up. It may be set up either directly by the learner or by the IMS SS TM.

Standard-oriented AEHS Adaptation Component Vs. Conventional AEHS Adaptation Component. The NetCoach adaptation component can implement two types of adaptation functionalities [47][113]:

- *Adaptive courseware sequencing*. The functionality to provide the learner with the most suitable, individually planned sequence of knowledge units to learn and the sequence of learning tasks to work with. Individual courseware sequencing means that LMSs dynamically compute and suggest which document is best to be visited next according to the learning goal and the learner's learning state of the documents. Learners get a warning if they visit a document with missing prerequisites. Also according to the learner's learning goals, all direct and indirect prerequisites are automatically computed and corresponding documents are suggested.
- *Adaptive link annotation*. The functionality to support the learner in hyperspace orientation and navigation by changing the appearance of visible links according to the learner's current learning state. If all prerequisites of a document have been learned by a learner, the link to this document is marked with a green ball to sign that this document is recommended for reading. If at least one prerequisite of a document has not been learned by a learner, the link to this document is marked with a red ball to sign that this document is not recommended for reading. If the tests corresponding to a document have been successfully passed, the link to this document is marked with a yellow ball to sign that this document has been learned already. If a document is a terminal page and inferred to be known, the link to this document is marked with an orange ball.

As the standard-oriented AEHS document space, user model and observations cannot achieve the same functionalities as in NetCoach, in the comparable design of the standard-oriented AEHS adaptation component, we can only implement part of adaptation functionalities of the NetCoach adaptation component. With regard to the adaptive courseware sequencing functionality in NetCoach, since the standard-oriented AEHS document space is based on a pre-defined hierarchical structure and does not support the dynamic generation of delivery sequence of documents, it is impossible to achieve the same adaptation functionality in the standard-oriented AEHS adaptation component. With regard to the adaptive link annotation functionality, as it is implemented based on the pre-defined document structure, we can implement most of its adaptation functionalities in the comparable design except for the functionalities that heavily depend on the indirect prerequisite-relation in the NetCoach document space, as well as the inference processing in the third layer of the NetCoach user model. Put precisely, the standard-oriented AEHS adaptation component can implement full of the "yellow ball" annotation functionality, part of the "green ball" and "red ball" annotation functionality without considerations on the indirect prerequisite-relation. Due to lack of the inference capability, it cannot implement the "orange ball" annotation functionality.

7. Conclusions

Although the E-Learning standardization is surely not simple universal solution to all the problems we face in E-Learning today, it is a key technology to liberate learning content from local implementations to enable some high-level "abilities" of E-Learning such as accessibility, interoperability, durability and reusability. Starting from the most important component of a generic E-Learning architecture, in this thesis we focus on standardization of the E-Learning content management and propose the standard-oriented E-Learning content management as the first step towards the standard-oriented E-Learning. In order to implement two major functionalities of the standard-oriented E-Learning content management: the standard-oriented learning content discovery and standard-oriented learning content delivery, we have addressed following critical design and implementation issues.

With regard to the standard-oriented learning content discovery, we propose Edutella, an RDF-based E-Learning content management P2P infrastructure, to connect heterogeneous educational repositories and further enable efficient learning content discovery across various back-end systems and learning resource metadata sets. In Edutella, P2P is adopted as one cornerstone technology to cross heterogeneity of educational repositories and further support a flexible E-Learning content management strategy enabling learning content producers to actively participate in a global sharing network without losing the control over their learning content. Leveraging basic facilities of the JXTA platform, Edutella implements a super peer based network topology being able to support ubiquitous and scalable computing and at the same time ensure the efficiency of the learning content discovery by avoiding aimless broadcast. In order to exchange distributed functionalities between Edutella and other distributed computing paradigms built upon Web services, we extend the main framework of Edutella and propose an approach for realizing the service layer interaction between Edutella/JXTA and Web services as the first exploration towards a unified distributed computing architecture converging P2P and Web services for E-Learning.

Besides P2P, Edutella adopts RDF as another cornerstone technology to describe Edutella resources, to define and mediate Edutella P2P functionalities, and to drive Edutella's schema-based P2P networking. Leveraging RDF/RDFS, Edutella specifies RDF-QEL to express and exchange RDF queries and query results, defines ECDM to cross heterogeneity of educational repositories, and uses super peer indices to achieve distributed query processing, routing and mediation. In order to accommodate XML metadata in the RDF-based Edutella, we propose two approaches for integrating XML metadata repositories into Edutella in terms of XML query languages supported. Whereas the XPath-enabled integration approach can only be applied to the DCMES XML binding as well as a limited number of learning resource metadata sets, the XQuery-enabled integration approach provides a generic solution to integrate arbitrary XML metadata sets into Edutella. Furthermore, in order to simplify the RDF-QEL query construction process for complex XML metadata schemas, we propose an approach for querying complex XML data schemas using QBE. Working together, these technologies enable XML and RDF learning resource metadata to be managed in Edutella in a consistent manner.

With regard to the standard-oriented learning content delivery, we propose a reference model being able to standardize two major operations of a learning content delivery process: the learning content organization and run-time environment implementation based on sets of E-Learning standards and specifications including the IMS CP, IMS SS, IEEE Data Model, IEEE API and ADL SCORM. We further develop sets of enabling technologies to implement the reference model based on standard Web technologies and metadata technologies. Going a step further, we apply the reference model to conventional AEHSs and propose an approach for converging the E-Learning standardization and conventional AEHS research to realize the standard-oriented AEHS. The standard-oriented AEHS is expected to be able to combine advantages of both E-Learning standardization and conventional AEHS, ensuring reusability and portability of learning content while achieving adaptability of the learning content delivery.

Taking an engineering approach, the research in this thesis is mainly focused on constructing a testbed for the standard-oriented E-Learning, starting from its most critical component: the standard-oriented E-Learning content management. The testbed is not only purposed to accommodate most of "best practice" of the current E-Learning standardization but also provide the guidance for its future development. We believe that such a testbed can promote the development and distribution of standardized learning content, based on which the standard-oriented E-Learning can start on a journey leading to its final success.

8. References

[1] ACM, ACM Computing Classification System [1998 Version], Valid in 2003, http://www.acm.org/class/1998/

[2] ADL Technical Team, ADL SCORM Version 1.3 Application Profile, March 2003, http://www.adlnet.org

[3] Ahlborn, B., W. Nejdl, and W. Siberski, OAI-P2P: A Peer-to-Peer Network for Open Archives, in Proc. of the Workshop on Distributed Computing Architectures for Digital Libraries (as part of the 31st International Conference on Parallel Processing), Vancouver, Canada, Aug. 2002.

[4] AICC, CMI 001-AICC/CMI Guidelines for Interoperability, Version 3.5, Apr. 2001, http://www.aicc.org/

[5] Apache Software Foundation, Welcome to WSIF: Web Services Invocation Framework, http://ws.apache.org/wsif/

[6] ARIADNE Foundation, ARIADNE Metadata Recommendation, Version 3.2, Feb. 2002, http://www.ariadne-eu.org/

[7] Beckett, D., E. Miller, and D. Brickley, Expressing Simple Dublin Core in RDF/XML, Nov. 2001, http://dublincore.org/documents/dcmes-xml/

[8] Bellwood, T., L. Clément, D. Ehnebuske, A. Hately, M. Hondo, Y. L. Husband, K. Januszewski, S. Lee, B. McKee, J. Munter, and C. von Riegen, UDDI Version 3.0, July 2002, http://uddi.org/pubs/uddi_v3.htm

[9] Berglund, A., S. Boag, D. Chamberlin, M. F. Fernandez, M. Kay, J. Robie, and J. Siméon, XML Path Language (XPath) 2.0, May 2003, http://www.w3.org/TR/xpath20/

[10] Biron, P. V., and A. Malhotra, XML Schema Part 2: Datatypes, May 2001, http://www.w3.org/TR/xmlschema-2/

[11] Boag, S., D. Chamberlin, M. F. Fernandez, D. Florescu, J. Robie, and J. Siméon, XQuery 1.0: An XML Query Language, May 2003, http://www.w3.org/TR/xquery/

[12] Booth, D., M. Champion, C. Ferris, F. McCabe, E. Newcomer, and D. Orchard, Web Services Architecture, Aug. 2003, http://www.w3.org/TR/ws-arch/

[13] Botros, S., and S. Waterhouse, Search in JXTA and Other Distributed Networks, in Proc. of the 1st International Conference on P2P Computing (P2P 2001), Lingköping, Sweden, Aug. 2001.

[14] Box, D., D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer, Simple Object Access Protocol (SOAP) 1.1, May 2000, http://www.w3.org/TR/SOAP/

[15] Brickley, D., and R. V. Guha, RDF Vocabulary Description Language 1.0: RDF Schema, Jan. 2003, http://www.w3.org/TR/rdf-schema/

[16] Brunkhorst, I, H. Dhraief, A. Kemper, W. Nejdl, and C. Wiesner, Distributed Queries and Query Optimization in Schema-Based P2P-Systems, in Proc. of the International Workshop On Databases, Information Systems and Peer-to-Peer Computing, Berlin, Germany, Sept. 2003.

[17] Brusilovsky, P., Adaptive and Intelligent Technologies for Web-based Education. In Rollinger, C. and C. Peylo (Eds.), Künstliche Intelligenz (4), Special Issue on Intelligent Systems and Tele-teaching, 1999.

[18] Brusilovsky, P., Methods and Techniques of Adaptive Hypermedia, in User Modeling and User Adapted Interaction 6 (2-3), Special Issue on Adaptive Hypertext and Hypermedia, 1996.

[19] Brusilovsky, P. and H. Nijhawan, A Framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities, in Proc. of the World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education (E-Learn 2002), Montreal, Canada, Oct. 2002.

[20] Brusilovsky, P., J. Eklund, and E. Schwarz, Web-based Educations for All: A Tool for Development Adaptive Courseware, in Proc. of the 7th International World Wide Web Conference (WWW'98), Brisbane, Australia, Apr. 1998.

[21] Cabrera, F., G. Copeland, T. Freund, J. Klein, D. Langworthy, D. Orchard, J. Shewchuk, T. Storey, Web Services Coordination (WS-Coordination), Aug. 2002, http://www-106.ibm.com/developerworks/library/ws-coor/

[22] CEN/ISSS LTWS, CWA 14643 Internationalisation of the IEEE Learning Object Metadata, Jan. 2003, http://www.cenorm.be/isss/cwa_download_area/cwa14643.pdf

[23] CEN/ISSS LTWS, CWA14645 Availability of Alternative Language Versions of a Learning Resource in IEEE LOM, Jan 2003, http://www.cenorm.be/isss/cwa_download_area/cwa14645.pdf

[24] Chaudhri, A., R. Zicari, and A. Rashid, XML Data Management: Native XML and XML Enabled DataBase Systems, Addison-Wesley, USA, 2003.

[25] Chawathe, S. S., H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. D. Ullman, and J. Widom, The TSIMMIS Project: Integration of Heterogeneous Information Sources, in Proc. of the 10th Meeting of the Information Processing Society of Japan (IPSJ), Tokyo, Japan, Oct. 1994.

[26] Christensen, E., F. Curbera, G. Meredith, and S. Weerawarana, Web Services Description Language (WSDL) 1.1, March 2001, http://www.w3.org/TR/wsdl

[27] Clark, J., XSL Transformations (XSLT) Version 1.0, Nov. 1999, http://www.w3.org/TR/xslt

[28] Clark, J., and S. DeRose, XML Path Language (XPath), Version 1.0, Nov. 1999, http://www.w3.org/TR/xpath

[29] Clip2, The Gnutella Protocol Specification, Version 0.4, March 2001, http://www.clip2.com

[30] Connolly, D., F. van Harmelen, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein, DAML+OIL (March 2001) Reference Description, Dec. 2001, http://www.w3.org/TR/daml+oil-reference

[31] DAML Services Coalition, DAML-S: Semantic Markup for Web Services, May 2003, http://www.daml.org/services/

[32] Druschel, P., and A. Rowstron, PAST: A Large-scale, Persistent Peer-to-Peer Storage Utility, in Proc. of The 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII), Schloss Elmau, Germany, May 2001.

[33] Dublin Core Metadata Initiative, Dublin Core Metadata Element Set, Version 1.1, July 1999, http://dublincore.org/documents/1999/07/02/dces

[34] Dublin Core Metadata Initiative, DCMI Metadata Terms, March 2003, http://dublincore.org/documents/2003/03/04/dcmi-terms/

[35] Duval, E., E. Forte, K. Cardinaels, B. Verhoeven, R. Van Durm, K. Hendrikx, M. Wentland-Forte, N. Ebel, M. Macowicz, K. Warkentyne, and F. Haenni, The ARIADNE Knowledge Pool System, Communications of the ACM, 44 (5), 2001.

[36] Duval, E., W. Hodgins, S. Sutton, and S. L. Weibel, Metadata Principles and Practicalities, D-Lib Magazine, 8(4), April 2002.

[37] Fernández, M., A. Malhotra, J. Marsh, M. Nagy, and N. Walsh, XQuery 1.0 and XPath 2.0 Data Model, May 2003, http://www.w3.org/TR/xpath-datamodel/

[38] Flenner, R., M. Abbott, T. Boubez, F. Cohen, N. Krishnan, A. Moffet, R. Ramamurti, B. Siddiqui, and F. Sommers, Java P2P unleashed, Sams Publishing, USA, 2002.

[39] Foster, I., C. Kesselman, J. M. Nick, and S. Tuecke1, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, June 2002, http://www.globus.org/ogsa/

[40] Friesen, N., S. Fisher, S. Hesemeier, and A. Roberts, CanCore Guidelines Version 1.9, Aug. 2003, http://www.cancore.ca/indexen.html

[41] Garcia-Molina, H., J. D. Ullman, and J. Widom, Database Systems: The Complete Book, Prentice Hall, USA, 2001.

[42] Gavriloaie, R., Personalized Access to Learning Objects in a Distributed RDF-based Repository, Master's thesis, University Politehnica of Bucharest, Romania, 2002.

[43] Gong, L., Project JXTA: A Technology Overview, Oct. 2002, http://www.jxta.org/

[44] Halevy, A. Y., Z. G. Ives, P. Mork, and I. Tatarinov, Piazza: Data Management Infrastructure for Semantic Web Applications, in Proc. of the 12th International World Wide Web Conference (WWW 2003), Budapest, Hungary, May 2003.

[45] Hatala, M., and G. Richards, Global vs. Community Metadata Standards: Empowering Users for Knowledge Exchange, in Proc. of the 1st International Semantic Web Conference (ISWC 2002), Sardinia, Italy, June 2002.

[46] Henze, N., Adaptive Hyperbooks: Adaptation for Project-Based Learning Resources, Ph.D. Thesis, University of Hannover, Germany, 2000.

[47] Henze, N., and W. Nejdl, Logically Characterizing Adaptive Educational Hypermedia Systems, in Proc. of the Adaptive Hypermedia 2003 Workshop, Budapest, Hungary, May, 2003.

[48] Hors, A. L., P. L. Hégaret, L. Wood, G. Nicol, J. Robie, M. Champion, and S. Byrne, Document Object Model (DOM) Level 3 Core Specification, Version 1.0, June 2003, http://www.w3.org/TR/DOM-Level-3-Core/

[49] Horton, W. K., Leading E-Learning, ASTD Press, USA, 2001.

[50] Hwang, J., P. Aravamudham, E., Liddy, J. Stanton, and I. MacInnes, IRTL (Information Resource Transaction Layer) Middleware Design for P2P and Open GRID Services, in Proc. of the 36th Annual Hawaii International Conference on System Science, Big Island of Hawaii, USA, Jan. 2003.

[51] IEEE LTSC, IEEE 1484.12.1: Draft Standard for Learning Object Metadata, July 2002, http://ltsc.ieee.org

[52] IEEE LTSC, IEEE 1484.11.1 Data Model for Content Object Communication, Ballot Draft 2, Aug. 2003, http://ltsc.ieee.org

[53] IEEE LTSC, IEEE 1484.11.2 ECMAScript API for Content to Runtime Services Communication, Ballot Draft 4, July 2003, http://ltsc.ieee.org

[54] IMS Global Learning Consortium Inc., IMS Learning Resource Metadata Specification, Version 1.2.1, Sept. 2001, http://www.imsproject.org

[55] IMS Global Learning Consortium Inc., IMS Content Packaging Specification, Version 1.1.2, Aug. 2001, http://www.imsproject.org

[56] IMS Global Learning Consortium Inc., IMS Simple Sequencing Specification, Version 1.0, Oct. 2002, http://www.imsproject.org

[57] IMS Global Learning Consortium Inc., IMS Question and Test Interoperability Specification, Version 1.2.1, March 2003, http://www.imsproject.org

[58] ISO/IEC, ISO/IEC 15944-1: 2002 Information Technology - Business Agreement Semantic Descriptive Techniques Part 1: Business Operational Aspects of Open-edi for Implementation, Sept. 2002, http://www.iso.org

[59] ISO/IEC, ISO/IEC 16262:2002 Information Technology - ECMAScript Language Specification, June 2002, http://www.iso.org

[60] JXTA Search Engineering Team, JXTA Search Protocol Specification, Version 1.0, June 2001, http://search.jxta.org/protocol.html

[61] Katchaounov, V. J. T., and T. Risch, Scalable View Expansion in a Peer Mediator System, in Proc. of the 8th International Conference on Database Systems for Advanced Applications (DASFAA 2003), Kyoto, Japan, 2003.

[62] Kokkelink, S., and R. Schwänzl, Expressing Qualified Dublin Core in RDF/XML, Apr. 2002, http://dublincore.org/documents/dcq-rdf-xml/

[63] Kreger, H., Web Services Conceptual Architecture (WSCA 1.0), May 2001, http://www-3.ibm.com/software/solutions/webservices/pdf/WSCA.pdf

[64] Kubiatowicz, J., D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, OceanStore: An Architecture for Global-Scale Persistent Storage, in Proc. of the 9th international Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), Cambridge, MA, USA, Nov. 2000.

[65] Lagoze, C., H. V. de Sompel, M. Nelson, and S. Warner, The Open Archives Initiative Protocol for Metadata Harvesting, Version 2.0, June 2002, http://www.openarchives.org/

[66] Lassila, O., and R. R. Swick, Resource Description Framework (RDF) Model and Syntax Specification, Feb. 1999, http://www.w3.org/TR/REC-rdf-syntax

[67] Leymann, F., Web Services Flow Language (WSFL 1.0), May 2001, http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf

[68] Liu, X., K. Maly, M. Zubair, and M. L. Nelson, Arc - An OAI Service Provider for Cross Archiving Searching, in Proc. of the ACM/IEEE Joint Conference on Digital Libraries, Roanoke, VA, USA, June 2001.

[69] Loeser, A., W. Nejdl, M. Wolpers, and W. Siberski, Information Integration in Schema-based Peer-to-Peer Networks, in Proc. of the 15th Conference On Advanced Information Systems Engineering (CAISE 03), Klagenfurt/Velden, Austria, June 2003.

[70] Marsh, J., XML Base, June 2001, http://www.w3.org/TR/xmlbase/

[71] McBride, B., Jena: Implementing the RDF Model and Syntax Specification, in Proc. of the 2nd International Workshop on the Semantic Web (SemWeb 2001), Hongkong, P. R. China, May 2001.

[72] McGuinness, D. L., and F. van Harmelen, OWL Web Ontology Language Overview, Aug. 2003, http://www.w3.org/2001/sw/WebOnt/

[73] Naeve, A., The Concept Browser: A New Form of Knowledge Management Tool, in Proc. of the 2nd European Web-based Learning Environments Conference (WBLE 2001), Lund, Sweden, Oct. 2001.

[74] Nejdl, W., B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch, Edutella: A P2P Networking Infrastructure Based on RDF, in Proc. of the 11th International World Wide Web Conference (WWW 2002), Honolulu, Hawaii, USA, May 2002.

[75] Nejdl, W., M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Brunkhorst, and A. Löser, Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks, in Proc. of the 12th International World Wide Web Conference (WWW 2003), Budapest, Hungary, May 2003.

[76] Nejdl, W., W. Siberski, B. Simon, and J. Tane, Towards a Modification Exchange Language for Distributed RDF Repositories, in Proc. of the 1st International Semantic Web Conference, Sardinia, Italy, June 2002.

[77] Nejdl, W., W. Siberski, and M. Sintek, Design Issues and Challenges for RDF- and Schema-Based Peer-to-Peer Systems, in ACM SIGMOD Record, 32 (3), Sept. 2003.

[78] Nilsson, M, and W. Siberski, RDF Query Exchange Language (QEL): Concepts, Semantics, and RDF Syntax, June 2003, http://edutella.jxta.org/

[79] O'Neill, M., Web Services Security, McGraw-Hill Osborne Media, USA, 2003.

[80] Open Knowledge Initiative, OKI Open Service Interface Definitions, Oct. 2002, http://web.mit.edu/oki/index.html

[81] Papakonstantinou, Y., H. Garcia-Molina, and J. Widom, Object Exchange Across Heterogeneous Information Sources, in Proc. of the 11th International Conference on Data Engineering, Taipei, Taiwan, March 1995.

[82] Polson, M. C., and J. J. Richardson (Eds.), Foundations of Intelligent Tutoring Systems, Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1988.

[83] Qu, C., T. Engel, and C. Meinel, Implementation of a WebDAV-based Collaborative Distance Learning Environment, in Proc. of the Annual Conference of the ACM Special Interest Group on University and College Computing Services (ACM SIGUCCS 2000), Richmond, Virginia, USA, Oct. 2000.

[84] Qu, C., J. Gamper, and W. Nejdl, Collaborative Courseware Authoring and Publishing based on WebDAV, XML, and XSLT, in Proc. of the IEEE International Conference on Trends in Communications (IEEE EUROCON 2001), IEEE Computer Society Press, Bratislava, Slovak Republic, July 2001.

[85] Qu, C., J. Gamper, and W. Nejdl, A Collaborative Courseware Generating System based on WebDAV, XML, and JSP, in Proc. of the 1st IEEE International Conference on Advanced Learning Technologies (IEEE ICALT 2001), Madison, Wisconsin, USA, Aug. 2001.

[86] Qu, C., and W. Nejdl, Constructing a Web-based Asynchronous and Synchronous Collaboration Environment Using WebDAV and Lotus SameTime, in Proc. of the Annual Conference of the ACM Special Interest Group on University and College Computing Services (ACM SIGUCCS 2001), Portland, Oregon, USA, Oct. 2001.

[87] Qu, C., and W. Nejdl, Bridging O-Telos and XML with XML Schema: The Authoring Environment for KBS Adaptive Hyperbook, in Proc. of IEEE International Conference on Information Technology: Coding and Computing (IEEE ITCC 2002), Las Vegas, Nevada, USA, Apr. 2002.

[88] Qu, C., and W. Nejdl, Exploring JXTASearch for P2P Educational Media Discovery, in Proc. of the IEEE International Workshop on Knowledge Media Networking (KMN'02), Kyoto, Japan, July 2002.

[89] Qu, C., and W. Nejdl, Towards Open Standards: the Evolution of an XML/JSP/WebDAV Based Collaborative Courseware Generating System, in Proc. of the 1st International Conference on Web-based Learning (as part of the 28th International Conference on Very Large Databases (VLDB'2002)), Kowloon, Hongkong, P. R. China, Aug. 2002.

[90] Qu, C., and W. Nejdl, Towards Interoperability and Reusability of Learning Resource: a SCORM-conformant Courseware for Computer Science Education, in Proc. of 2nd IEEE International Conference on Advanced Learning Technologies (IEEE ICALT 2002), Kazan, Tatarstan, Russia, Sept. 2002.

[91] Qu, C., W. Nejdl, and H. Schinzel, Integrating Schema-specific Native XML Repositories into a RDF-based E-Learning P2P Network, in Proc. of the 2nd International Conference on Dublin Core and Metadata Applications (DC 2002), Florence, Italy, Oct. 2002.

[92] Qu, C., and W. Nejdl, Searching SCORM Metadata in a RDF-based E-Learning P2P Network using XQuery and Query by Example, in Proc. of the 3rd IEEE International Conference on Advanced Learning Technologies (IEEE ICALT 2003), Athens, Greece, July 2003.

[93] Qu, C., and W. Nejdl, Towards Open Standards: the Evolution of a Collaborative Courseware Generating System, in International Journal of Distance Education Technologies, 1(3), July-Sept. 2003.

[94] Qu, C., and W. Nejdl, Interacting Edutella/JXTA Peer-to-Peer Network with Web Services, in Proc. of the 2004 International Symposium on Applications and the Internet (SAINT 2004), Tokyo, Japan, Jan. 2004.

[95] Qu, C., and W. Nejdl, Integrating XQuery-enabled SCORM XML Metadata Repositories into an RDF-based E-Learning P2P Network, in Journal of Educational Technology & Society, 7(2), Apr. 2004.

[96] Ratnasamy, S., P. Francis, M. Handley, R. Karp, and S. Shenker, A Scalable Content-Addressable Network, in Proc. of the Annual Conference of the ACM Special Interest Group on Data Communications (ACM SIGCOMM 2001), San Diego, CA, USA, Aug. 2001.

[97] Risch, T., V. Josifovski, and T. Katchaounov, Functional Data Integration in a Distributed Mediator System, In Gray, P. M. D., L. Kerschberg, P. J. H. King, and A. Poulovassilis (Eds.), The Functional Approach to Data Management: Modeling, Analyzing and Integrating Heterogeneous Data, Springer-Verlag, Germany, 2003.

[98] Robie, J., The Syntactic Web: Syntax and Semantics on the Web, in Proc. of XML Conference & Exposition 2001, Orlando, FL, USA, Dec. 2001.

[99] Rowstron, A., and P. Druschel, Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems, in Proc. of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, Nov. 2001.

[100] Schlosser, M., M. Sintek, S. Decker, and W. Nejdl, HyperCup - Hypercubes, Ontologies and Efficient Search on P2P Networks, in Proc. of the International Workshop on Agents and Peer-to-Peer Computing, Bologna, Italy, July 2002.

[101] Schmitz, C., Staab, S., Studer, R., Stumme, G., and Tane, J., Accessing Distributed Learning Repositories through a Courseware Watchdog, in Proc. of E-Learn 2002: World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education, Montreal, Canada, Oct. 2002.

[102] Simon, B., Z. Miklos, W. Nejdl, M. Sintek, and J. Salvachua, Smart Space for Learning: A Mediation Infrastructure for Learning Services, In Proc. of the 12th International Conference on World Wide Web, Budapest, Hungary, May 2003.

[103] Stoica, I., R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, in Proc. of the Annual Conference of the ACM Special Interest Group on Data Communications (ACM SIGCOMM 2001), San Diego, CA, Aug. 2001.

[104] Sun Microsystems Inc., JXTA v2.0 Protocols Specification, Feb. 2003, http://www.jxta.org/

[105] Sun Microsystems Inc., Security and Project JXTA, http://www.jxta.org/

[106] Sun Microsystems Inc., JXTA SOAP Binding, http://soap.jxta.org/

[107] Ternier, S., E. Duval, and P. Vandepitte, LOMster: Peer-to-peer Learning Object Metadata. in Proc. of ED-MEDIA'2002 - World Conference on Educational Multimedia, Hypermedia and Telecommunications, Denver, CO, USA, June 2002.

[108] Thacker, S., A. Sheth, and S. Patel, Complex Relationships for the Semantic Web, in Fensel, D., J. Hendler, H. Liebermann, and W. Wahlster (eds.), Spinning the Semantic Web, MIT Press, USA, 2003.

[109] Thaden, U., W. Siberski, and W. Nejdl, A Semantic Web Based Peer-to-Peer Service Registry Network, Technical Report, Learning Lab Lower Saxony, 2003.

[110] Thompson, H. S., D. Beech, M. Maloney, and N. Mendelsohn, XML Schema Part 1: Structures, May 2001, http://www.w3.org/TR/xmlschema-1

[111] Traversat, B., A. Arora, M. Abdelaziz, M. Duigou, C. Haywood, J. C. Hugly, E. Pouyoul, and B. Yeager, Project JXTA 2.0 Super-Peer Virtual Network, May 2003, http://www.jxta.org/

[112] Waterhouse, S., JXTA Search: Distributed Search for Distributed Networks, May 2001, http://search.jxta.org/JXTAsearch.pdf

[113] Weber, G., H. C. Kuhl, and S. Weibelzahl, Developing Adaptive Internet Based Courses with the Authoring System NetCoach, in Proc. of the 3rd Workshop on Adaptive Hypermedia (AH2001), Aarhus, Denmark, Aug. 2001.

[114] Weber, G., and M. Specht, User Modeling and Adaptive Navigation Support in WWW-based Tutoring Systems, in Proc. of the 6th International Conference on User Modeling (UM97), Sardinia, Italy, June 1997.

[115] Wiley, D. A., The Instructional Use of Learning Objects, the Agency for Instructional Technology and the Association for Educational Communications and Technology, 2002.

[116] Wolf, B., Peer-to-Peer Networking for Distributed Learning Repositories, Master thesis, University of Hannover, Germany, 2001.

[117] Wolpers, M., W. Nejdl, and I. Brunkhorst, An O-Telos Provider Peer for the RDF-based Edutella P2P-Network, in Proc. of Semantic Authoring, Annotation and Knowledge Markup Workshop (SAAKM 2002) at the 15th European Conference on Artificial Intelligence, Lyon, France, July 2002.

[118] Zloof, M. M., Query by Example: A Database Language, IBM Systems Journal, 16(4), 1977.

Publication List

2004:

[1] Qu, C., and W. Nejdl, Interacting Edutella/JXTA Peer-to-Peer Network with Web Services, in Proc. of the 2004 International Symposium on Applications and the Internet (SAINT 2004), IEEE Computer Society Press, Tokyo, Japan, Jan. 2004.

[2] Qu, C., and W. Nejdl, Integrating XQuery-enabled SCORM XML Metadata Repositories into an RDF-based E-Learning P2P Network, in Journal of Educational Technology & Society, 7(2), Apr. 2004.

2003:

[3] Qu, C., and W. Nejdl, Searching SCORM Metadata in a RDF-based E-Learning P2P Network using XQuery and Query by Example, in Proc. of the 3rd IEEE International Conference on Advanced Learning Technologies (IEEE ICALT 2003), IEEE Computer Society Press, Athens, Greece, July 2003.

[4] Qu, C., and W. Nejdl, Towards Open Standards: the Evolution of a Collaborative Courseware Generating System, in International Journal of Distance Education Technologies, 1(3), July-Sept. 2003.

2002:

[5] Qu, C., and W. Nejdl, Bridging O-Telos and XML with XML Schema: The Authoring Environment for KBS Adaptive Hyperbook, in Proc. of the IEEE International Conference on Information Technology: Coding and Computing (IEEE ITCC 2002), IEEE Computer Society Press, Las Vegas, Nevada, USA, Apr. 2002.

[6] Nejdl, W., B. Wolf, C. Qu, S. Decker, M. Stintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch, Edutella: A P2P Networking Infrastructure Based on RDF, in Proc. of the 11th International World Wide Web Conference (WWW 2002), ACM Press, Honolulu, Hawaii, USA, May 2002.

[7] Qu, C., and W. Nejdl, Exploring JXTASearch for P2P Educational Media Discovery, in Proc. of the IEEE International Workshop on Knowledge Media Networking (KMN'02), IEEE Computer Society Press, Kyoto, Japan, July 2002.

[8] Qu, C., and W. Nejdl, Towards Open Standards: the Evolution of an XML/JSP/WebDAV Based Collaborative Courseware Generating System, in Proc. of the 1st International Conference on Web-based Learning (as part of the 28th International Conference on Very Large Databases (VLDB'2002)), Springer-Verlag LNCS, Kowloon, Hongkong, P. R. China, Aug. 2002.

[9] Qu, C., and W. Nejdl, Towards Interoperability and Reusability of Learning Resource: a SCORM-conformant Courseware for Computer Science Education, in Proc. of the 2nd IEEE International Conference on Advanced Learning Technologies (IEEE ICALT 2002), IEEE Computer Society Press, Kazan, Tatarstan, Russia, Sept. 2002.

[10] Qu, C., W. Nejdl, and H. Schinzel, Integrating Schema-specific Native XML Repositories into a RDF-based E-Learning P2P Network, in Proc. of the 2nd International Conference on Dublin Core and Metadata Applications (DC 2002), Florence University Press, Florence, Italy, Oct. 2002.

2001:

[11] Qu, C., J. Gamper, and W. Nejdl, Collaborative Courseware Authoring and Publishing based on WebDAV, XML, and XSLT, in Proc. of the IEEE International Conference on Trends in Communications (IEEE EUROCON 2001), IEEE Computer Society Press, Bratislava, Slovak Republic, July 2001.

[12] Qu, C., J. Gamper, and W. Nejdl, A Collaborative Courseware Generating System based on WebDAV, XML, and JSP, in Proc. of the 1st IEEE International Conference on Advanced Learning Technologies (IEEE ICALT 2001), IEEE Computer Society Press, Madison, Wisconsin, USA, Aug. 2001.

[13] Qu, C., and W. Nejdl, Constructing a Web-based Asynchronous and Synchronous Collaboration Environment Using WebDAV and Lotus SameTime, in Proc. of the ACM Special Interest Group on University and College Computing Services (ACM SIGUCCS 2001), ACM Press, Portland, Oregon, USA, Oct. 2001.

2000:

[14] Qu, C., T. Engel, and C. Meinel, Implementation of an Enterprise-level Groupware System Based on J2EE Platform and WebDAV Protocol, in Proc. of the 4th IEEE International Enterprise Distributed Object Computing Conference(IEEE EDOC 2000), IEEE Computer Society Press, Makuhari, Japan, Sept. 2000.

[15] Qu, C., T. Engel, and C. Meinel, Implementation of a WebDAV-based Collaborative Distance Learning Environment, in Proc. of the ACM Special Interest Group on University and College Computing Services (ACM SIGUCCS 2000), ACM Press, Richmond, Virginia, USA, Oct. 2000.

Acknowledgements

I would like to thank my supervisor Prof. Dr. Wolfgang Nejdl for his continual support and foresighted guidance throughout my Ph.D. study. Besides academic research, I have learned from him lots of valuable characteristics which are essential to a qualified researcher such as diligence, conscientiousness, persistence and many more. I am sure all that I have learned will be beneficial to my future career.

I would like to thank all colleagues from the Institute of Information Systems & Learning Lab Lower Saxony, University of Hannover for the cooperation throughout my Ph.D. program. I have benefited a lot from many valuable discussions, research seminars and diverse academic activities. I also appreciate many active cooperation from our Virtual Campus and PADLR project partners coming from Free University of Bozen, Italy, Stanford University, USA, Uppsala University, Sweden, Royal Institute of Technology, Sweden, Karolinska Institute in Stockholm, Sweden, Business University of Vienna, Austria, University of Braunschweig, Germany, University of Karlsruhe, Germany, IWF Göttingen, Germany, Technical University of Darmstadt, Germany, and Fraunhofer ISST, Germany. This thesis has absorbed many great ideas contributed by all these colleagues. Special thanks go to my colleague Mark Painter, who kindly helped me to translate the abstract of this thesis into German.

I would like to thank Prof. Dr. Christoph Meinel, the director of the Institute of Telematics, University of Trier, Germany for the opportunity he gave me to work as research assistant under his supervision from May 1999 to October 2000. The research experience there laid the foundation for my Ph. D. study in computer science.

I would like to thank several students from the Department of Computer Science, University of Hannover: Markus Schneider, Holger Schinzel, Hans Both and Than Vu Quang Chinh for lots of supporting work.

The German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung) and the Ministry of Science and Culture of Lower Saxony (Niedersächsisches Ministerium für Wissenschaft und Kultur) are kindly acknowledged for the financial support.

Last but not least, I would like to thank my wife Dr.rer.nat. Dongjie Gao, who has accompanied me to go through all the hardship and happiness in the past years. My daughter Kaidi Qu, born at 8:34am, April 27, 2004 in Hannover, has doubled my joy of finishing my Ph.D. My father, mother and two younger sisters have always been supporting and encouraging me to keep forging ahead with their great expectation and patience. I hope that I could reward the lease bit of their love with this thesis.

Curriculum Vitae

Name:	Changtao Qu
Birth:	May 24, 1968, Jilin, P. R. China
Nationality:	P. R. China
Personal Data:	Male, Married, One child

Education:

2000.10 - 2004.7	Ph.D. Candidate & Research Assistant, Major: Computer Science Institute of Information Systems & Learning Lab Lower Saxony, University of Hannover, Hannover, Germany
1992.9 - 1995.3	Master Student, Major: Electrical Engineering, Degree: M.Sc. School of Electrical Engineering & Automation, Harbin Institute of Technology, Harbin, P. R. China
1986.9 - 1990.7	Bachelor Student, Major: Electrical Engineering, Degree: B.Sc. School of Electrical Engineering & Automation, Tianjin University, Tianjin, P. R. China

Experiences:

2004.7-	Research Scientist, C&C Research Laboratories, NEC Europe Ltd., Sankt Augustin, Germany
1999.5 - 2000.10	Research Assistant, Institute of Telematics, University of Trier, Trier, Germany
1998.7 - 1999.5	German Language Student, University of Kiel, Kiel, Germany
1996.12 - 1998.7	Technical Editor, China InfoWorld, Beijing, P. R. China
1995.3 - 1996.12	Research Assistant, Institute of Manufacturing Engineering, Tsinghua University, Beijing, P. R. China
1990.7 - 1992.9	Assistant Engineer, Jilin Paper Mill, Jilin, P. R. China