Pradeep Kumar Yalamanchili

eLearning Objects on Mobile Devices

Master thesis based on the examination and study regulations for the Master of Engineering degree programme
Information Engineering
at the Department of Information and Electrical Engineering
of the Faculty of Engineering and Computer Science
of the University of Applied Sciences Hamburg

Supervising examiner: Prof. Dr. Thomas Schmidt
Second examiner: Prof. Dr.rer.nat Hans-Jürgen Hotop

Day of delivery February 27th 2007

Pradeep Kumar Yalamanchili

Title of the Master Thesis

eLearning Objects on Mobile Devices

Keywords

eLearning Objects, Mobile Devices, hylOs, iPOD, Semantic relations, Content adaptation, Learning Content Management System, Semantic Navigation, Linear Navigation.

Abstract

Proliferation of mobile devices increased the demand for mobile eLearning scenarios. Modular content components in the field of online educational content management systems form a suitable basis and combine multimedia content and metadata with context forming structural relations. Mobile access to such eLearning Objects, however requires content and structural adaptations to the device. In this thesis, a software application is designed and implemented which accomplishes the access and adaptation of the eLearning Objects suitable for a dedicated mobile device. Additionally, the adapted content is embedded into the device-specific arrangement structure and made possible for the user to start self-exploration of the learning content in an intuitive fashion.

Pradeep Kumar Yalamanchili

Thema der Masterarbeit

eLearning Objekte für Mobile Endgeräte

Stichworte

eLearning Objekte, mobile Endgeräte, hylOs, iPOD, Semantische Inhaltsbeziehungen, Inhaltsanpassung, Lern-Content-Management Systeme, Semantische Navigation, Lineare Navigation.

Kurzzusammenfassung

Die rasch zunehmende Verbreitung von mobilen Endgeräten läßt die Anforderung an mobile Lernszenarien ebenfalls ansteigen. Modulare Inhaltsbausteine aus dem Bereich der online eLearning Content Management Systeme bilden hierfür eine geeignete Grundlage. Solche eLearning Objekte vereinen multimediale Inhalte und Metadaten mit kontextbildenden Strukturrelationen. Ein mobiler Zugriff auf solche Lernobjekte erfordert jedoch die Anpassung von Inhaltsbausteinen und Struktur an das Endgerät. In dieser Arbeit wird eine Softwareanwendung entworfen und implementiert, welche den Zugriff und die Anpassung von Lernobjekten für ein dediziertes mobiles Endgerät durchführt. Darüber hinaus wird der adaptierte Content in die gerätespezifische Anordnungsstruktur eingebettet und dem Anwender so ein intuitiver, selbstexplorativer Zugang zu den

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Evolution of eLearning started with the supplementation of class room based training by Computer Based Training (CBT). With the advent of World Wide Web, Computer Based Training was transformed into Web-based Training (WBT) and learners were able to access online learning materials from home and as well as from remote locations. eLearning made learning possible from anywhere and anytime, using desktop and laptop computers. The advancements in Information and Communication technology (ICT) which includes digital information, mobile communicational devices, mobile computing, wireless access etc., took eLearning to a further level and permit learning possible from mobile devices, popularly known as Mobile learning [16] [21].

Mobile learning or m-learning, is considered as a part of eLearning and uses mobile devices such as cell phones, PDAs (Personal Digital Assistance), smart phones for presenting educational content[22] [24]. The main advantage of these devices is their small size, portability and anytime, anywhere usage: which allows the users to access them while travelling and utilize the ideal time for learning, reviewing the lecture materials and for other general purposes. Mobile devices such as PDAs, smart phones, game consoles and iPODs are becoming ubiquitous and in general, their usage is significantly high among the students, employees and people of young generation. Also, the growing internet capabilities of mobile devices are making people to choose them as alternatives to desktop computers and many of them are even accessing emails, making bank transactions, buying online products by using such devices. This growing demand is encouraging educational organizations, mobile vendors, developers and other people working in this area, to develop novel applications for these devices.

Much research work is going in the field of mobile learning and many universities and training organizations are developing applications for utilizing mobile devices in education. As a result several eLearning systems were developed providing mobile access to the online learning resources and they include Learning Object Repositories (LORs), Learning Management Systems (LMSs) and Learning Content Management Systems (LCMSs). According to Broisin et al. [4] these systems are differentiated as:

- Learning Object Repository (LOR) - manage learning objects and their metadata.
- Learning Management System (LMS) - focus on management of learning services of end users and store the learning resources in a dedicated space without any metadata.
- Learning Content Management System (LCMS) - functionalities include enabling, creating, delivering of learning content. It stores the learning resources independently in a LOR and provides a transparent environment for sharing and reusing learning objects.

Presently, the content from the eLearning systems are available as HTML Web pages or PDF scripts. In contrast to accessing the Web content from desktop devices, accessing it with a mobile device provides difficult experience for the user, due to small screen sizes, limited supported formats and several other limitations. Hence, to provide easy access of Web learning content to the mobile device and enhance its usability, considerable device dependent adaptation is required before delivering it to the device.

In this thesis content adaptation, navigation and usability issues are addressed and, developed techniques to adapt the content to the device and enhance the usability of online learning content access on mobile devices by providing suitable navigation to content. As a part of this work, a software application is designed to access the learning content from an online educational content management system, adapt the content according to the characteristics of the mobile device and transfer the content to the mobile device. Additionally, the software application organizes the content in a smart way inside the device, providing the user with a navigation for attaining self-exploration of the learning content. In the implementation part, eLearning content of hylOs [1] are chosen for rendering on iPOD[2].

[1] http://www.hylos.org/

[2] http://www.apple.com/itunes/

This thesis is organized as follows. Chapter 2 presents an overview of the background issues required for this topic. Chapter 3 gives a detailed description about the challenges to be faced in developing mobile eLearning applications and other related problems. In Chapter 4, the related works are discussed followed by proposed concepts and approach for adaptation of eLOs in Chapter 5. The System design and implementation are detailed in the Chapter 6. The results obtained are tested and evaluated in the Chapter 7. Finally conclusion and future work are given in the Chapter 8.

Chapter 2

Mobile eLearning Background

2.1 Mobile Web Initiative

Growing demand for content support on mobile devices and increasing popularity of mobile Web access, made researchers in this field to focus on addressing the usability and interoperability issues of the Web content. To overcome these problems and make the mobile web access easy, World Wide Web Consortium (W3C) launched a Mobile Web Initiative (MWI)[1], focusing mainly on the best practices and device descriptions.

Many Web sites available today are designed for accessing on desktop size displays. Mobile users find difficulty in navigating the Web content due to the limitations imposed on the display size, keypad, and format. It is not easy for the content developers to build Web sites that works well on all types of mobile devices. Hence, a feasible regulation is necessary for creating content to mobile devices. MWI combined with other active members in this field, addressed interoperability and usability issues that are making mobile Web access difficult. As a result, a set of guidelines and best practices are specified by MWI BP in [23], for the content developers, to help in building Web sites that work well on all kinds of mobile devices. The best practices includes statements that are related to the:

- Content pages and how they lay out
- Technical construction of pages
- Navigation and linking mechanisms
- Input

[1]http://www.w3.org/Mobile/

Along with the best practices, researchers are also developing technical trustmarks for the Web content to indicate adherence to the best practices. These trustmarks or labels are human readable and machine readable. According to Owen et al. [26] content created with these labels have several advantages:

- Content developers can advertise their content to consumers with the mobileOK trust marks
- Search engines and other content locating engines can give importance for locating those content and much more

Widely varying characteristics of mobile devices make it difficult for a Web site to provide an acceptable user experience across a significant range of devices. For example different devices support different markup features and different displays may demand different sized images. Therefore, when delivering content to the mobile devices, it is required to adapt the details of markup, size and format of image etc., depending on the characteristics of the device. In the absence of adaptation, Web sites can be developed targeting the default delivery context specification provided in MWI BP

According to McCathieNevile and Rabin [23] adaptation can be performed via several methods. One method is to determine the device type and selecting an appropriate set of previously prepared content that suits the device characteristics. In the other method, adaptation can be implemented dynamically, by formatting the content at the time of retrieval, taking into account not only the statistically determined properties, such as screen resolutions, but also dynamically determined properties, such as the temporary attachment of a keyboard. The success of such dynamic adaptations depends greatly on the knowledge of the device characteristics available to the individuals or systems performing the adaptation. Hence to develop high-quality Web applications for mobile devices, a list of essential device attributes were developed by Pearce et al. [27] in MWI Device Description Landscape, which include:

- Screen dimensions
- Supported and preferred markup
- Supported and preferred image formats
- Size limitations

- Color Support
- Other browser features
- Non markup object support

The relevant standard organizations for these descriptions are W3C's Composite Capabilities / Preferences Profile (CC/PP) [25] and OMA's User Agent Profile (UAProf) [2].

Device descriptions are useful for content adaptation which can be performed in several stages. At the server side, if the server can identify the clients mobile device, based on the provided device description information, it can decide on how the content to be generated to it .For example, a server can easily switch between WML, CHTML or XHTML according to browser's known behavior. The format of the objects can be adapted and resolution of the images can be chosen at the server side itself. Content adaptation is also possible along the delivery chain between server and the client. Device Descriptions for adapting content in the delivery chain is similar to the need within the content server. Content is altered on the fly as it passes through different entities across the mobile network which includes WAP gateways, proxies and dedicated transcoding platforms. Client side adaptation includes, the device accepting the content and adapting the content according to its device characteristics. This technique generally involves a browser using the information about its host device to decide how to best layout and display the content. The browser only requires information about the host device and needs to adapt only the content that has reached the device[27].

2.2 Aspects of Mobile devices

The adaptation of eLearning Objects on mobile devices requires a thorough review of the device capabilities. This section describes the technical capabilities of the mobile devices which are popularly used among people.

2.2.1 PlayStation Portable

Sony PlayStation Portable (PSP)[3], a hand held gaming device, provides many functionalities besides its core gaming features. It offers resolution

[2] http://www.wapforum.org/what/technical/SPEC-UAProf-19991110.pdf
[3] http://playstation.com/

(480 x 272 pixels) and display size (4.3inch) that are relatively higher than other mobile devices and gaming consoles. PSP has the ability to display images, play audio and video files. The internet browser provided in the system is highly compatible and provides a reasonable browsing experience to the user. It is better than most of the mobile phone browsers and is effective with the sites built for the PSP. Interaction devices like D-Pad and the Square button provided in PSP make navigating the Web fast and easy.

PSP supports flash content and there it is possible to access interactive elements present in the online learning content using the joystick. Multimedia content is accessible from a PSP, if proper adaptations are made to the content according to the restrictions imposed on the display size, resolution, memory capacity, audio and video formats of the device. Web browser support and wireless connectivity to Internet, allows it to access the online learning materials from anywhere, anytime. In addition to the play-back capability of video and music files from the available device dependent drive, PSP is adequately provided with input devices such as active buttons, keypad for game controlling etc. The available controls provide a facility for adopting them for navigation and interaction of Web based hypermedia learning materials as well.

Additionally, PSP's RSS Channel feature[4], allows adding link information to a channel list, enabling PSP to link to RSS or podcast-distributed content available on Web pages. Therefore, when the system is connected to the Internet, the list of content that has been previously added is automatically updated and plays the new content on the system. An option for memory card is also available and hence the RSS or podcast content can be saved on the card for offline usage.

2.2.2 Personal Digital Assistant

Personal digital assistant or shortly PDA[5] is a handheld device that is basically designed for organizing the personal information. They are also called as palmtops or hand-held computers or pocket computers. The most significant feature of the PDA is the combination of input and output functionality as a touch-sensitive screen, which provides flexible interaction with the device to navigate through the content without necessary to press keys. Also, PDAs are equipped with a compact ASCII keyboard. They are also provided with static and dynamic storage devices. Additionally, Secure

[4]http://www.playstation.com/manual/psp/rss/en/index.html
[5]http://en.wikipedia.org/wiki/Personal_digital_assistant

Digital (SD) and Flash cards can be used in the expansion slots. Better processing power and larger storage capabilities, are allowing PDAs to become multimedia compatible. Backup and storage methods employed by PDAs are either by synchronizing to PC or by using memory cards. Rather than streaming the media files available in larger sizes, users can download the files to the PC and transfer them to PDA. The image formats supported by PDAs include GIF, JPEG, PNG etc.,

Palm OS[6] and Windows Mobile[7] are the leading operating Systems for PDAs. Palm OS provides wide variety of mobile software for PDAs. A Software Development Kit (SDK) is provided for devices with Palm OS platform which includes a full set of File Browsing API, Web Browsing API, Navigating API and much more. It allows developers to manipulate the input events by writing appropriate call back functions and to customize user specified applications. Microsoft Windows Mobile provides SDK and a variety of development tools (e.g Microsoft embedded Visual C++, Compact .NET framework) for the software application developers. These tools can be used in conjunction with SDK consisting of a comprehensive set of API for developing software applications

Moreover, wireless-enabled PDAs are provided with Wireless cards, for connecting to the internet, and users can subscribe to the services providing internet connection through access points in major areas (e.g . restaurants, business centers) via their Wi-Fi (802.11) equipped devices. PDAs with Web-enabled digital phone features, called as Smart-Phones are usually WAP enabled.

2.2.3 Mobile Phones and Smart Phones

Mobile phones have become an essential thing for people around the world. Providing communication between people and portability are the main design concerns of mobile phones; hence they are confined to small screens and low resolutions. But present day mobile phones are becoming more compatible to audio, video and other multimedia data.

Mobile phones which are exhibiting PDA features are commonly called as Smart Phones. The data enabled features of PDA are integrated into the compact wireless telephones, allowing the user to operate with one hand. Compared to desktop devices, mobile devices like PDAs, hybrid PDAs or Smart Phones are limited in screen size, color and in Web content delivery.

[6] https://pdn.palm.com/regac/pdn/index.jsp
[7] http://msdn.microsoft.com/windowsmobile/

To address these problems several browser technologies exists for hybrid PDAs and mobile phones. For example, Opera browser[8] can automatically fit the Web pages to the small screens, rescales the larger images and reformat the html tables. It works on all platforms of mobile devices.

Smart phones run popular operating systems like Palm OS and Windows Mobile and provide all development features similar to PDAs. They also come with Symbian OS platform which is especially for mobile phones. Its 32-bit preemptive multitasking is designed for devices running flash applications. Mobile phones with Symbian OS[9] are provided with a rich set of APIs for developing independent software applications.

MobiPocket reader

eBook or electronic book is the digital version of a normal text book. They contain additional features like magnification, text search, copy and paste. Mobipocket Reader[10] is a Universal eBook Reader that can be installed on multiple platforms of PDAs and Smart phones for reading eBooks. Mobipocket format, a binary format for distribution of eBooks, is based on HTML and is reflowable. Mobipocket supports most features of standard HTML to format and layout text and images. Mobipocket Reader provides a bookshelf listing and a reader. A book's text can be orientated in any direction and an auto scroll feature is provided which avoids the need to turn pages. Scrolling speed is adjustable using the up and down hard key buttons. Images can be embedded in different sizes. Depending on the screen resolution, the most adapted image is displayed. Images in the text, which are scaled to fit the screen, can be opened in a viewer for more detailed examination.

2.2.4 iPOD

Apple iPOD is a portable media player with a user interface design. In addition to playing audio files, iPODs of fifth-generation can also play videos, display image and note files. It stores media on an internal hard drive which can hold large amounts of data (30 GB to 80 GB) similar to a PC. Display sizes of 5G iPOD is better than many mobile phones but are relatively less than Sony PSP. To display images of high resolutions on 5G iPOD, it is not necessary to rescale them as it was in the case of PSP, PDA

[8] http://www.opera.com/
[9] http://www.symbian.com/developer/
[10] http://www.mobipocket.com/dev/

and mobile phones. iPOD automatically rescales the image size to fit the display screen, but videos and audios have to be converted to the formats supported by iPOD which can be done by using Apple iTunes[11] .

The multimedia content including photos, audios and videos can be imported from the internet to a PC and then synchronized to the iPOD via iTunes media player. iPOD supports mpeg audio formats which may contain meta descriptors and the navigation within the iPOD is built upon these meta descriptors, achieving a contextual hierarchical navigation with respect to meta tags (e.g., author, title). In 5G iPODs, textual presentations are produced by using a limited number of HTML tags supported in the Notes section, which browses a hierarchical file system within the device. The Notes section can hold note files and images in JPEG format. Furthermore, custom menus can be created to display categories that users can navigate to view note files. Any note file in the Notes section can be linked to audio/video files synchronized to iPOD and also to other note files, images and sub folders stored in the root Notes folder. By transforming the custom content to the Notes section, an automated self-paced learning guide can be created with sophisticated interactive multimedia presentations. The user interaction and navigation of menus and files within the device is carried out by means of its trend setting Click Wheel and five buttons under it. As in the case of Windows Mobile, Palm OS and Symbian, Apple do not provide any programmable environment for developing software applications for iPOD. It provides iTunes COM API[12] which is useful for accessing iPOD data base and controlling iTunes (e.g for creating, managing play lists).

The comparison of the essential features of above discussed mobile devices is shown in the table 2.1

[11] http://www.apple.com/itunes/
[12] http://developer.apple.com/sdk/itunescomsdk.html

Devices	Application	Display resolution (in pixels)	Development tools	Interaction Devices
Sony Psp	Gaming	480 x 272	Closed	Directional buttons, Analog stick
Tungsten C (Palm OS)	Organizer	320 x 320	Open	Touch Screen, 5-way buttons, Qwerty Keyboard
CN3 (Windows Mobile OS)	Organizer	240 x 320	Open	QWERTY keyboard
Nokia N80 (Symbian OS)	Phone and Organizer	352 x 416	Open	Scroll wheel with 4 buttons, center select
iPOD (5G)	Media Player	320 x 240	Closed	4 buttons Scroll wheel and a center select

Table 2.1: Comparision Table : Essential features of mobile devices

2.3 eLearning Objects and standards

From the time eLearning was invented there has been a dramatic change in the learning methods, which in turn brought changes to the development and delivery of learning contents. Among the educational institutions and other organizations, major attention is drawn towards developing Learning objects. The concept of learning objects is similar to the Object-Oriented concepts of computer science and the main objective is to create self-contained learning materials that can be reused in multiple contexts. The definition of learning object is defined by many people. According to Polsani [28], a learning object should be independent of delivery media (PDF, HTML, audio, video etc) and knowledge management systems, tagged with meta data and are usable in different learning contexts. Several organizations are involved in developing standards for supporting low-cost, high quality sharable learning contents and one such standard is IEEE Learning Object Metadata (IEEE LOM)[8] . According to Duval et al. [8] the definition of an eLearning Objects is ”an entity -digital or non-digital that may be used for learning education or training”. It can be composed of:

- small atomic information units suitable for publishing online and having no restrictions on its proportion or media types.
- a set of meta data information properties of the object in a standardized vocabulary.
- an option for named interrelations expressing structural information of the content and its educational semantic.

In the eLearning field, one of the main standards for meta data is IEEE LOM, which describes meta data instance of a learning object as, the relevant characteristics of learning object to which it applies and these characteristics are grouped under nine different categories namely life, cycle, meta-metadata, educational, technical, rights, relation, annotation, and classification. A brief explanation of these categories according to IEEE LOMv1.0 Base Schema [8] is given below :

1. The General category, groups the general information that describes the learning object as a whole which includes the details about the identifier, title, structure of the learning object. A textual description of the learning object and language used within it for communicating to the intended user, keywords are also mentioned within this category.
2. The Lifecycle category describes the history and current state of the learning object and those entities that have affected this learning object during its evolution. It mentions the present status and the version of the learning object. And also, it gives the information about the contributions of people and organizations to the learning object, which includes their information, identity, date of contribution and their role in it

3. The Meta-Metadata category describes about this metadata record itself (rather than the learning object that this record describes). It describes how the metadata instance can be identified, who created it, how, when and with what references.

4. The Technical category describes the technical requirements and characteristics of the learning object which includes the format, size, location, necessary technical capabilities for using this learning object and other essential details.

5. The educational category describes the key educational or pedagogical characteristics of the learning object which include:

   (a) Interactivity Type, specifying whether the learning object supports active learning or passive learning.
   (b) Learning Resource Type, specifying the kind of learning object (e.g. Exercise, simulation, questionnaire, graph, diagram etc)
   (c) Interactivity Level, the degree of interactivity to which the learner can influence the behavior of the learning object (very low, medium, high).
   (d) Semantic Density, the degree of conciseness of a learning object, which is estimated in terms of its size, span or duration.
   (e) Intended End User Role, specifying the user for which the learning objects was designed (teacher, author, learner, manager).
   (f) Context, environment for which the learning object is intended (school, higher education, training or other).
   (g) Typical Age Range.
   (h) Difficulty level for the typical intended target audience (very easy, easy, medium, difficult, very difficult).
   (i) Typical Learning Time it takes to work with this learning object.
   (j) Description specifying how the learning object is to be used.
   (k) The Language used by the typical intended user of the learning object (it depends on the language used in the General category).

6. The Rights category describes the intellectual property right and conditions of the use for the learning object. It specifies whether the user needs to pay for copyright or other restrictions when using the learning object.

7. The Relation category defines the relationship of the learning object with other learning objects. It includes the nature of the relationship (based on Dublin Core relations[13] for e.g. 'ispartof', 'isbasedon' etc.,) between the learning object and the target learning object. It also contains information about the target learning object.

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

9. Classification describes where the learning object falls within a particular classification system.

This metadata information decorates the content and is helpful for automatic processing. It simplifies the discovery and exchange of eLOs over the Web. Another important standard in the field of eLearning is Sharable Content Object Reference Model (SCORM)[29]. It encompasses a many learning standards in a model to provide accessibility, interoperability, durability and reusability to the learning content. The main objective of SCORM is to foster the creation of reusable learning content as instructional objects within a common technical framework for computer-based and Web-based learning. Adopting the work of different eLearning standards, SCORM describes this technical framework with a set of guidelines, specifications and standards. Some of the main underlying concepts in SCORM are as follows:

- SCORM describes a mechanism for assembling Sharable Content Objects (SCORM defined Learning Objects) using a Content Aggregation Model (CAM). It includes the metadata information to enable search and discovery of the content, content packaging component representing the learning experience that will be presented to the user and the packaging of the components for exchange between system to system. And it also provides the rules for sequencing the components.

- SCORM provides Run-Time Environment specifications for Learning Management Systems (LMS) to deliver, display the SCORM conformant content to the learners and track the content through a common SCORM RTE Data Model.

[13] http://dublincore.org/documents/relation-element/

- A Sequencing and Navigation (SN) model is described by SCORM for delivering the learning content dynamically based on the learner needs[29].

2.4 eLearning Content Management Systems

A Content Management System (CMS)[2] designed for higher education purpose is often called as a Learning Content Management System (LCMS) [13]. It facilitates authoring, maintenance, production and publishing of the Learning content. LCMSs which maintain learning content on the Web and publish the content in the form of Web pages for the users, are known as eLearning Content Management Systems or online educational content management systems e.g. LON-CAPA [14] , an educational Learning Content Management System.

In general, a LCMS is comprised of a content repository, an Authoring environment, a delivery interface and administrative applications as shown in the figure 2.1.

Instructional designers, subject matter experts and all the authors having access to the authoring environment can create and/or assemble the course materials such as lecture notes, assignments, conceptual questions etc., using available Web based authoring tools. The entire learning content and its metadata are stored and managed in a content repository. The repository can be shared among the organizations which are having access to the content. Delivery interface provides the facility to deploy the learning materials in different output formats such as HTML Web pages, PDF scripts, printed materials or using CD-ROM to the user. Additionally, the tasks involving the monitoring of learners track record, helping instructors and learners with course administration are carried out by administrative applications which is also a part of LCMS. A general eLearning Content Management System provides the following features:

- Create, maintain, manage and publish the learning content in different formats (e.g. HTML , PDF).
- Enable instructors to create learning materials and share resources with colleagues across the institution in a simple and efficient manner.
- Support for creation of more interactive content.

[14] http://www.loncapa.org/

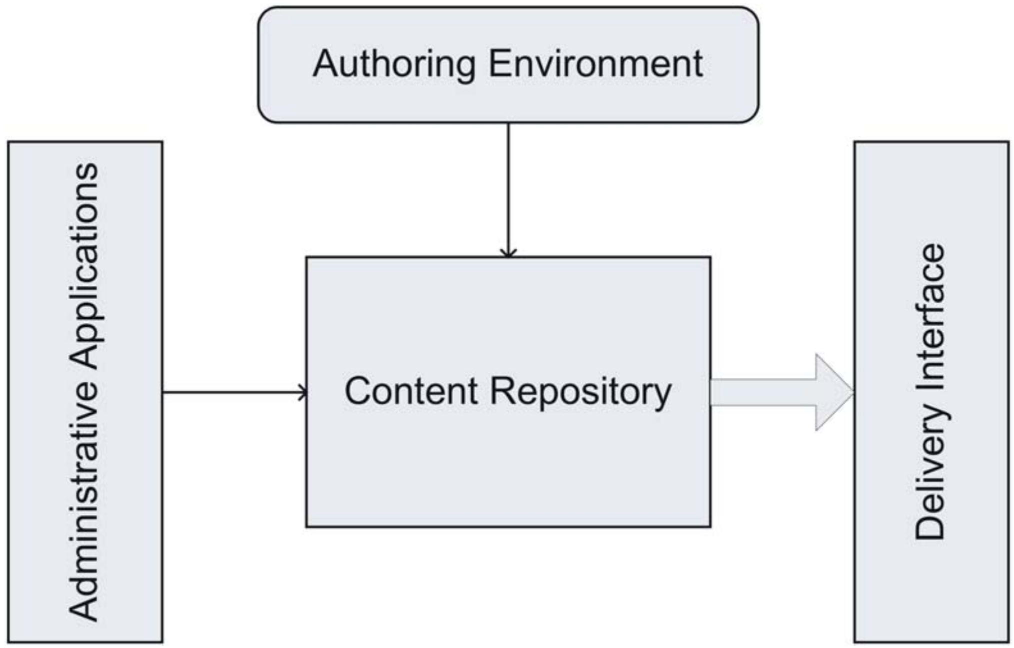


Figure 2.1: Elements of a Learning Content Management System.

- Maintain the learning content repositories which provides content replication, versioning, catalog information etc.
- Facilitate cross-media publications.
- Support adaptive learning and asynchronous collaborative learning.
- Customize student learning experience depending on their feedback and thereby facilitate interactive education.
- Maintain learners track records and manage other course related administrative functions.

Despite the above mentioned features, the increase in off-campus and mobile learning are demanding self-consistent online materials to students at the university level. To meet the increasing requests in the open distant learning, the eLearning content management systems need to meet the following requirements[9]:

- coherence and timeliness of information

- re-use of simple, compound and fragmented content material
- dynamic content structuring and arrangement with coherent presentation
- ease in authoring and updating of content constituents
- flexible options of content decoration with meta data
- semantically guided content access and retrieval.

2.5 Hypermedia Learning Object System (hylOs)

hylOs is an adaptive eLearning Content Management System and a runtime environment based on eLO information model. hylOs is build upon more general Media Information repository (MIR)[15] and the MIR adaptive context linking environment (MIRaCLE) [10], its linking environment. Grounded on a powerful media object model, MIR was designed as an open system supporting XML[16] and JNDI[17] standards to provide easy modeling and implementation of complex multimedia applications. All the eLearning Objects reside within MIR and are composed of rich media components decorated with IEEE LOM metadata, interconnected by qualified relations. hylOs use Extensible Markup Language (XML) technology for storing all the data. It allows, the generation of various content access paths and online views from the same collection of re-usable eLOs, individually adapted to specific context of teachers and learners.

The modeling of hylOs content is as follows: All the metadata and particular content entities such as titles, authors, keywords or information about courses etc., are singled out. The separation of entities in this way provides content normalization and allows easy updating of content for authors and automatic generation of navigational overviews. The remaining content units are organized in paragraph objects which are sub structured according to its XML schema. By means of external structuring the paragraph units are collected to pages and hence, content entities of paragraph dimensions are easily re-used by applying multiple structural references in a static or dynamic way. hylOs implements these concepts for practical implementation by using XML technologies and provides the consistent separation of content, structure, logic and design elements [9].

[15] http://www.rz.fhtw-berlin.de/content/Projekte/MIR/index.html
[16] http://www.w3.org/XML/
[17] http://java.sun.com/products/jndi/

hylOs offers different content access views to the learners, with each view presenting the content of the underlying eLearning Objects according to a specific learning pattern or publication context. A linear path hierarchy composed by an instructional designer using appropriate eLOs from a knowledge repository is one perspective. Another view which is formed by content authors is a primary content structure with root eLO giving most common description and leaves with detailed information. Besides the above mentioned views, a constructivist view is also provided to the learner, which is based on educational semantic nets formed by the eLOs depending on their qualified relations. In this view, a set of constructivist tools are provided to the learners for supporting self-exploration of the eLOs. In contrast to the above accessing methods that are more focused on eLO structures, it is more concentrated on the eLO-centred view. Upon accessing any eLO in this view, user will be supplied with a list of related eLOs. User can access the eLOs of interest and navigate along any learning paths by switching between the three perspectives at any point [11].

hylOs constructs educational semantic nets based on the named inter-object relations and present them to the learners for navigation and knowledge exploration, as well as to the author and instructional designer. The relations specified in the relation section of IEEE LOM metadata, taken from Dublin Core library metadata set is limited to administrative approach, hence the relations are redefined in a semantic meaning. Additionally four pair-wise inverse, qualified relations, which are missing in LOM standards are chosen to provide educational semantics. Adding a new eLO into a repository requires manual identification and updating of appropriate relations with large amount of already existing entries. To avoid this manual netting, relation semantics are encoded in OWL ontology and processed by an inference engine, which is provided with a set of inference rules accounting for logical dependencies between related properties. Consequently any newly inserted object or relation will lead automatically to a chain of subsequent link placements within the hylOs system, thereby forming a dense educational semantic mesh [12]. This helps the learners to visualize the educational semantic net providing a transparent view to the related eLOs of subject area as shown in the figure 2.2.

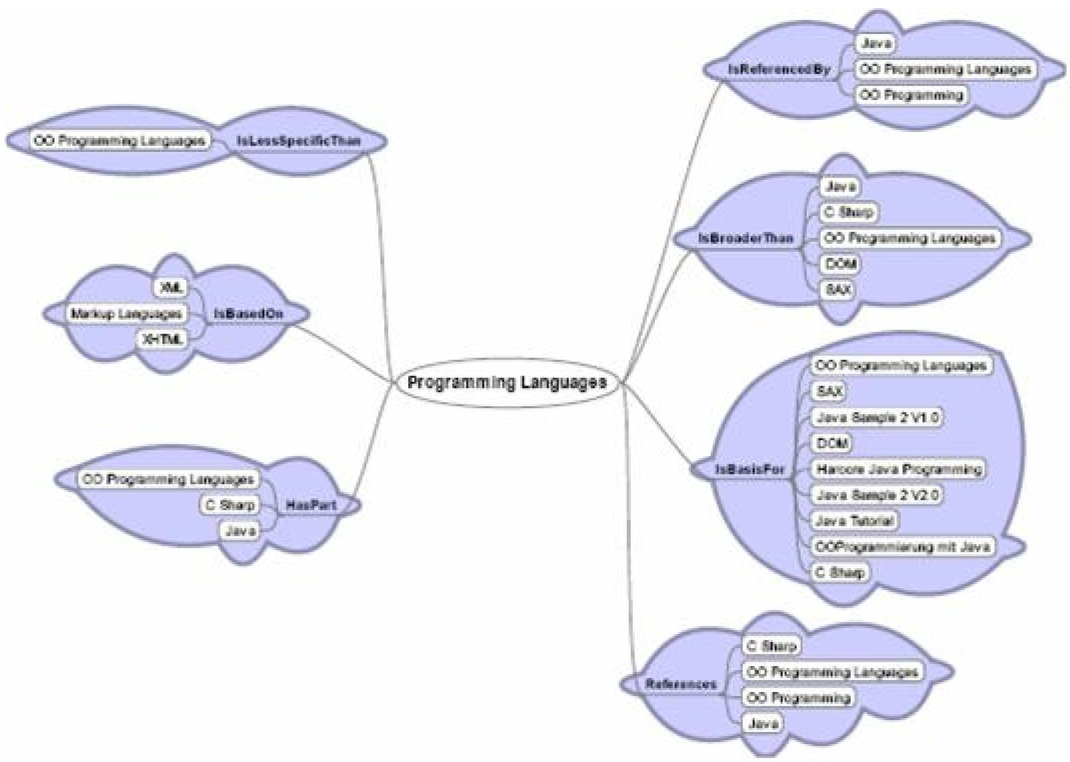


Figure 2.2: An example of an eLO centered view : Mind map navigation[12]

Chapter 3

Mobile eLearning Challenges

The content present in the online educational content management systems is accessed by learners using wide range of devices. Originally the content was developed targeting desktop users but the proliferation of mobile devices increased the demand for content support on mobile devices as well. To develop and present the content on different types of devices, several challenges have to be faced by content providers, application designers and device manufacturers. This chapter discuss some of the challenges by grouping them under different categories.

3.1 Device Limitations

As discussed in 2.1, display size, format support, processing power etc are some of the limitations associated with mobile devices and they have a major impact on accessing Web content. Among them display size is considered to be the predominant factor. The amount of viewable content at a time on a single page is very less on small screen devices. eLearning content presented as Web pages contain large amount of data. They can be viewed on desktop sized screens with out having any problem. Accessing the same content from a small screen device (e.g. PDAs with screen sizes about 20 times lesser than that of a desktop screen) require considerable amount of horizontal and vertical scrolling, which is not comfortable for the user.

For example, a Web page with the overview of hylOs Courses[1] accessed from a mobile device and desktop size screens are shown in the figure 3.1.

[1] http://hylosdev.fhtw-berlin.de/index.xhtml

(a) xda screen. (b) Desktop Screen.

Figure 3.1: hylOs Web page view from a mobile device (xda) and desktop screen.

Desktop devices are provided with Gigabytes of memory, whereas many mobile devices have relatively little memory and low processing powers. Operations associated with rendering Web page content, such as parsing long style sheet documents, handling invalid markup, laying out tables etc., needs more computational space. With limited processing power, processing media files in the Web content takes more time to complete and therefore introduces more delay. Another significant usability factor is input devices. For example, some eLearning systems provide feedback channels for which user require entering text via a keyboard. Some mobile devices do not have keyboard which is a major problem. Even if the mobile devices are provided with keypads, they are small in size and uncomfortable to use. eLearning materials are composed of textual data, graphical information, animations, and audio/video content, which need user interaction. In order to browse the entire content or to interact with the data or to browse external links, a user need to press more number of keys which is uncomfortable for the user, in the absence of a pointing device. Some devices like PDA and iPOD are provided with touch screen and scrolling wheel respectively, but nevertheless data input remains difficult compared to desktops.

3.2 Internet Access

Internet access is one of the main problems of mobile device compared to Personal Computers (the actual medium for delivering e-learning). Different telecommunication companies among worldwide use different wireless networks (e.g. 3G, GMS, and GPRS). Hence, when users having global roaming access mobile phones move around the world, learning becomes interrupted which results in loss of data and online access, due to the frequent disconnections occurring with the change of networks. Hence replication mechanisms have to be developed in the case of such involuntary disconnections or alternative solutions have to be investigated. In addition the bandwidth offered in mobile networks is also low.

3.3 Content Providers

Content providers face many problems while developing the Web content that is accessible from different devices. Some of the mobile devices do not allow to run external applications (e.g. iPOD). As the number of devices accessing the content increases, the problem of creating the presentation for every type of device becomes difficult. Hence authors have to develop rich content with description of all application and presentation aspects which allows to implement automatic adaptation to any delivery device. The adaptation software developed should create a presentation to match the capabilities of the device in question.

3.3.1 Online/Offline Learning

Users access the learning content according to their convenience. When accessing online learning content at home, they normally use desktop computers. In case if any additional reference materials are required, they connect to the Internet, access the required information and continue learning. Whereas, if the user is outside or traveling, the same content can be accessed via mobile device provided it is internet-enabled. When the device is not having Internet access or when it is more expensive to access content directly online, users prefers to download the content to the personal computer and then synchronized to the mobile device for offline usage. But in the case of offline learning, users donot have access to the remote data. Such situations have to be addressed by content providers, while providing content to mobile device. The following factors have to be considered while providing offline content :

- Content providers need to develop synchronizing applications for offline content.
- The applications should be developed making it sure that they are portable across different devices.
- Redundant information have to be provided for offline content whereas online content can be produced dynamically on the fly.
- Additionally, caching is to be maintained for offline content at the client-side.

3.4 Content Compatibility: adaptation

Designing Web content that is compatible on all mobile devices is difficult. Different devices support different standards. Wireless Markup Language (WML), based on XML, is a content format specifically devised for small screen devices that implement WAP protocol. Presently, Wireless Access Protocol (WAP) specification is supporting mobile users to access HTML and XML content dynamically from the Web. Using WAP the content can be adapted according to the device characteristics if the mobile devices are provided with device profiles (e.g UAProf or CC/PP etc.,), but unfortunately not all the mobile devices are provided with such profiles. Existing Internet standards like HTML, HTTP, Transport Layer Security (TLS) and Transport Control Protocol (TCP) would require huge amount of text based data to be sent, but mobile wireless networks are limited in bandwidth.

Much of the learning materials available in the Internet are not interoperable across different devices. Even if the content is presented in the standard formats, mobile devices neither support such formats nor SCORM players. Instead, they demand appropriate presentation and patterns to their usage. eLearning content is embed with rich multimedia content which includes text, audio, video, graphics and animations. Different devices support different formats and different markup languages. Hence, specific adaptation of the eLearning content depending on the characteristics of the mobile device is required prior to their rendering. It includes:

- High resolution images and videos have to be scaled
- Media formats have to be converted

- The structure of the documents have to be adapted to the device capabilities
- Tables have to be transformed to the device supported markup.

3.5 Navigation and Interactivity

In online hypermedia learning systems, users experience self-exploration of the learning content which gives the responsibility and the power to the users. According to Botafogo et al. [3] the main problem of explorative learning (e.g hypertext or hypermedia system) is the 'lost-in-hyperspace'. This leads to frustration and a possible abortion of the learning session. Within the adaptation of a hypermedia system for a mobile learning application, the presentation of the content has to be considered.

Any flexible, self- adaptive solution to the mobile content generation problem, requires adjustment in interactivity and navigation complexity to be manageable by device-specific players, which then handle hardware and software controls in an appropriate fashion. Hence, to master certain modifications are required. A semantic notion of interactive elements is required by a system to accomplish meaningful transformations. The whole idea of a reference work in learning materials is that at any time user can look up to the desired learning objects. For example as discussed in 2.5 eLOs in hylOs are connected by inter-object relations forming a dense mesh and a user can visualize a mind map navigation with the semantic structures. Traversing such complex semantic nets on a mobile device is difficult. To provide interaction and navigation on mobile device for accessing such educational semantic nets, application developers have to face a challenge in determining the set of user interface components and their features. Further more, adopting them to different navigations of the content requires a detailed study.

Chapter 4

Related Work

A lot of research is going on in the field of eLearning and m-learning. Many researchers from educational institutions, training organizations and mobile content providers are developing applications to provide eLearning content to the mobile device with enhanced navigation and user interaction. This chapter discusses, the work done by several researchers in this area categorized under specific groups.

4.1 Navigation and Interaction

A navigation interface for m-learning application was proposed by dos Anjos and de Oliveira [7] based on concept maps with the capability of adapting to learner's characteristics. Concept maps are spatial representations of learning objects and their interrelationships that are stored in memory.These spatial representations enable learners to interrelate the learning objects they are studying, as multidimensional networks of concepts [14]. Using this interface, a personalized interaction with the application is achievable to the learner. The concept maps used in the proposed interface model were used to represent the domain knowledge structure in educational activities. For a chosen domain, each concept of the map is a link to the corresponding learning object, containing the detailed presentation of the concept information. A user navigating though the map can visualize an active node (central node), and if exists, nodes above and below to the active node. A small prototype application was also implemented by dos Anjos and de Oliveira [7] which provides two sets of buttons for content and navigation respectively. Content Buttons are used for viewing the text, image and animation of the learning object. Navigation Buttons allow the user to return to the main node, return to the node, one level above in the hierarchy of nodes and for accessing the supported resources which help the user to find the current position in the map. figure 4.1

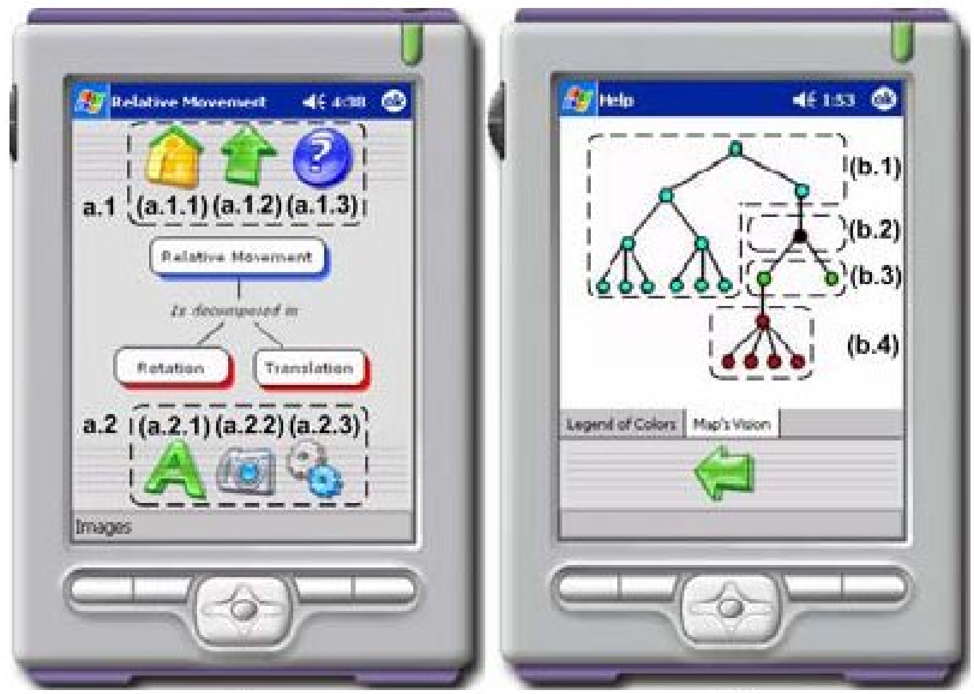


Figure 4.1: User Interface.

Different colors are used to indicate the access status of each concept and the decision for such annotation is taken by the adaptation model with in the application architecture. A node which is not accessed yet and can be accessed is indicated with a green shadow. A node which has been accessed and can be accessed again is indicated with blue shadow and a node with red shadow indicates that it cannot be accessed yet. Additionally, the color annotation of the nodes and a miniature of the concept map showing the users current position in the map are presented to the user.

4.2 Podcasts

Podcasts are the new aspect of audio/video broadcasts available on the internet. Real Simple Syndication (RSS)[1] which is an XML-based format for content distribution is used for publishing podcasts on a Web site. Podcasting is a process by which listeners subscribed to the RSS feeds are notified of new programs by their aggregators, which then download the programs and transfer to the listener's PC or MP3 audio players or iPods. An example is the Education Podcast Network[2] which contains wide range of podcast programming that are helpful to teachers looking for content to teach with and about, and to explore issues of teaching and learning.

Podcasting has become a new learning paradigm in numerous institutions. Many universities[3][4] [5] published foreign language courses, music and other audio/video recordings via podcasts for academic purpose. Stanford on iTunes [6] provides access to a wide range of audio and video related learning content via iTunes Music Store. Under this project two sites were developed. A public site is targeted mainly at alumni, which includes learning content, music, sports, faculty lecture notes and more. Another site with restricted access is available for students to deliver course-based materials. Any student having access, need the iTunes application installed on the desktop computer to download the podcasts. After downloading these recording can be synchronized to an iPOD player and use them for offline learning.

4.3 Adaptation

In many situations, eLearning content do not match with the mobile device requirements and demand adaptations. As discussed in 2.1 content adaptation can be done at the Server Side, In-Network and Client Side and it can be achieved by programming, using scripting languages, appropriate style sheets [XSLT][6] etc. In general, adaptations includes media content transformations which includes image and video transcoding and structural transformations which includes transforming the structure of the documents. This section discuss upon the work done by researchers for adapt-

[1] http://blogs.law.harvard.edu/tech/rss
[2] http://epnweb.org/
[3] http://dukecast.oit.duke.edu//
[4] http://itunes.stanford.edu/
[5] http://webcast.berkeley.edu/podcastabout.html
[6] http://itunes.stanford.edu/

ing the content and structure of the documents to be available for mobile access.

4.3.1 Digester System

Digester System developed by Girgensohn and Sullivan [15] provides an automatic reauthoring of Web documents to make them fit on small screen devices. All the Images in a page are transformed by predefined scaling factors and the reduced images hypertext links back to the originals. In addition to the image reduction, elision transforms are developed, where elided images are replaced with their alternative text or with a standard icon. If the screen space is too limited or the client device cannot display images, Digester will remove them from the document. If elided images were used as anchors for hypertext links, then Digester incorporates a transform that extracts the hypertext links from such images and formats them into a text (extracted from ALT tags of images or from part of the URL of the link) list of link anchors.

A indexed segmentation transform is applied by Digester, which divides a long Web page into a sequence of small subpages that fit small screen displays of handheld devices. The transform tries to find logical elements, such as text blocks or lists, by analyzing syntactic information on the Web page. It sequentially arranges the identified elements in the transcoded page until it can properly display the new page on the handheld device.The transform then creates a sequence of sub pages, each connected.

The table transform identifies a table in a Web page and checks whether the mobile device present the Web page properly or not. If the table is too wide or too long, the transform unrolls it and creates one sub page per cell in a top-down, left-right order. The tables nested within the tables were processed in the same manner

4.3.2 Content adaptation system

A content adaptation system for mobile devices was designed by Lemlouma and Laya [20], which enables the presentation of multimedia content on mobile devices. The system provides two components: an adaptation component at the proxy side and a formatter module at the client side. The adaptation component processes the interaction requests of the user and applies a structural and media adaptation on the original content. The result of the adaptation is transmitted to the client formatter that presents the different parts of the adapted content.

The media objects are displayed on the allocated layout regions on the mobile terminals. Synchronized Multimedia Integration Language (SMIL) [18] was used for defining the regions and the position, size and scaling of the media objects are controlled by SMIL region elements. To provide flexibility while allocating display size for the objects, Lemlouma and Laya [20] defined variable regions over a fixed interval. A set of events were associated with the regions and upon user interaction the corresponding event triggers and the content was presented with in the region. The user can select a particular region, and, via the proxy, navigate in different parts of the content (linear navigation) and for getting more details of the content, a hierarchical navigation can be obtained with the users request.

After a client request, the adaptation module (located on the proxy side) parses the different media objects nodes from the original document which can be a SMIL or an XHTML document. Depending on the text length and the client's device characteristics, the text node is decomposed into a set of sub nodes and sent to the client's formatter. For the first sub-node, the textual extraction is applied from the beginning of the text content; the extraction length depends on the clients region. The next extractions were calculated depending on the user interactions and all the decompositions were achieved in streaming. The user can navigate to the different parts of the original text using the direction keys of the device (Left and Right keys).

The content of an image node is resized according to the corresponding region. The user can select an image region and navigate in the corresponding image. After the user interaction (using the direction keys: Left, Right, Up and Down) only a partial area of the original image is resized and rendered in the selected region as shown in the figure 4.2. Left and Right direction keys enable linear navigation between sibling content blocks whereas Up and Down keys allow a hierarchical navigation. Such navigation applications are found in the medical sector which involves the access to radiography images.

Figure 4.2: Hierarchical Navigation within regions

The formatter module presents the content of the different regions and adds listeners in order to intercept the user interactions. Each user interaction triggers a request sent to the adaptation module of the proxy. A request includes a set of parameters such as: the selected region and the part of the content currently displayed together with the user interaction event. The proxy replies to the client request by an adapted content portion extracted from the original content. Consequently, the different parts of the content can be navigated progressively and are only sent when necessary. To adapt an area (wo,ho) of an original image to a region (wr,hr) while maintaining the image aspect ratio, the image area was resized to the new adapted width and height: wadapted = a.wo and hadapted = a. ho, where a = Min(wr/wo, hr/ho). For the textual adaptation, the text is extracted from the original document according to the region dimensions and the width and height of the destination font.

4.3.3 Fern Universität's Project

Fern Universität's Mobile Learning project was developed to extend the cross-media publishing features of the learning content management system, FuXML[7], from paper and laptop to mobile devices[19]. FuXML is delivering learning content as XHTML documents for online access from desktop and laptop computers and as PDF documents to be printed for offline usage. For accessing the content from a mobile device, an additional output channel was created at the production side (server-side) of FuXML,

[7] http://eleed.campussource.de/archiv/87

offering preprocessed presentation versions of learning materials suitable for PDAs, smart phones and cell phones running a HTML-capable browser.

The adaptation of the entire content is carried out at the server-side of FuXML. The presentation output of FuXML is converted into plain XHTML pages for which they developed special templates and style sheets and particular parameters were set in the configuration files. The resultant presentation of FuXML courses and the navigation page on the Palm PDA is shown in the figure 4.3

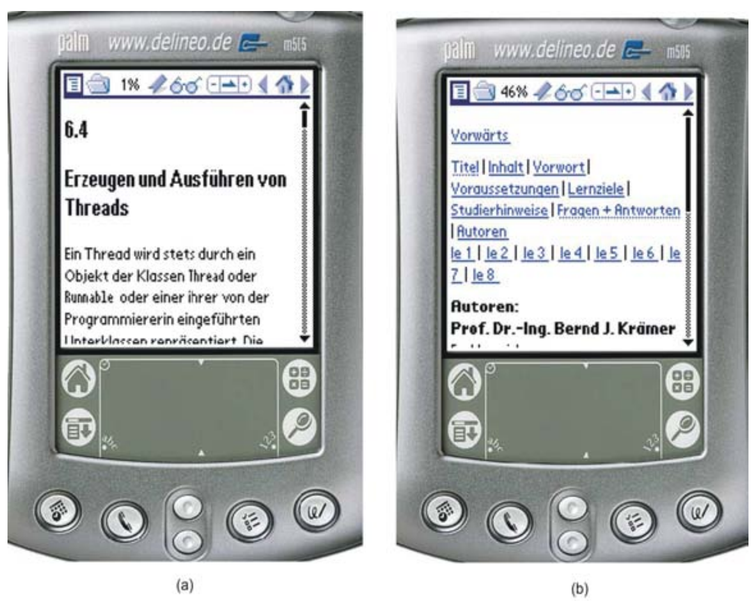


Figure 4.3: (a) Viewing the HTML version of course on a Palm PDA . (b) Navigation page on a Palm PDA

Furthermore, images are stored with high and low-resolutions separately for PC and mobile device access. Within the pages at the top and bottom of each page, a scarce navigation is provided with all the required links to the lectures and other information. Additionally, some of the language lectures and other courses are provided as online audio files for accessing on Apple's iTunes Music Store[8] . Users are allowed to download these audio files and access them on iPOD's, mp3 players and PC's.

[8]http://www.apple.com/itunes/store/

4.3.4 M-library

Another mobile learning project was developed by Cao et al. [5] at Athabasca University of Canada to extend the university library facility from desktop to mobile devices. The format of the AU's M-library Web site is designed in such a way that the format changes automatically in response to the device accessing the Web site. In addition to the generic CSS style sheet, they used mobile CSS style sheet suitable for accessing the Web site from a mobile device. But when a user access external Web sites that are linked from the m-library Web site, considerable scrolling is required for accessing the data because some pages or images do not fit to the screen and are unreadable. The library Web site has links to learning resources in PDF format. As there is no native PDF support on some mobile devices, all the documents were reformatted into HTML pages for accessing on such devices. They used a system on the server side for detecting whether the device is desktop or a portable device and also a software module called as AirPac[9] runs on the server to send out Java Server Pages formatted for the mobile devices requesting the information. At the moment, m-library supports only text data and in their ongoing work, they are trying to make audio/video files present in their resources accessible on mobile devices.

4.4 Other Mobile eLearning Projects

4.4.1 Mobile Learning Engine (MLE)

Mobile Learning Engine (MLE)[10] is a computer-aided multimedia based learning application created for running on Smart phones. Instead of accessing the traditional Learning Objects in the e-Learning environment, MLE has access to learning objects hosted in a central internet server, Mobile Learning Platform(MLP). MLE is used in combination with the learning platform and can also realize a hypermedia system i.e it can implement a knowledge network or any kind of hypertext or hypermedia system.The communication between MLE and MLP is carried out using open standards like HTTP and XML. Users can download learning objects directly from the server with the MLE installed on their mobile devices. Once the learning objects are downloaded on to the device, offline learning is achievable[17].

[9] http://www.iii.com/mill/webopac.shtml
[10] http://www.elibera.com/

Figure 4.4: Media-bar for the play back of video in MLE

The main means of navigation(up, down, left, right, active) is achieved by using Joystick which avoids the necessity of pressing any keys for using the application. All the learning objects for MLE are written in XML, which makes it not restricting to any particular kind of learning content. Formatted text, integrated images in the text, fonts and colors, hyperlinks with specific action, audio and video bars for play back, interactive questions and intelligence help, are some of the features that can be realized from the learning objects accessible with this application. For example, a media-bar with play back video of learning objects is shown in the figure 4.4

MLE was developed using Java 2 Micro Edition (J2ME)[11] to realize a platform independent application that can be used on different operating systems for Smart phones

4.4.2 Mobile Delivery Server (MDS)

Mobile Delivery Server (MDS) is a middle layer service between a LMS and a mobile device, which is responsible for adapting the learning objects to a mobile device depending on the requirements of different mobile devices [1]. The dynamic and automatic mapping of different XML formats and its content is a core functionality of the MDS. MDS uses a customized XML writer and parsing modules and able to map the learning contents coming

[11] http://java.sun.com/javame/index.jsp

from the LMS to any mobile content player that operates on the basis of XML.

MDS maintains a list of different mobile classes and based on the list, the content and format is transformed to be interpretable by the content players software which was provided for each mobile device. A two step transformation of learning content is proposed by the researchers. First step involves the downscaling of images or the substitution of videos by images or textual content. In the second step XML structure used by the LMS is mapped to the XML structure used by the content player on the mobile device. The basic idea of mapping used by MDS is, at first it extracts the content from the XML received by LMS and stores it according to the restrictions of the mobile device. The content suitable for a mobile player is written in XML structure understood by mobile device player. The content players developed with in the project work provides enhanced usability for the users. They are implemented using J2ME[12] for the mobile phones with Symbian OS and CrEme (JVM for Windows CE) for the PDA's with Windows CE.

The implementation was tested with a Web-based LMS, called e-tutor and it provides an interface called DirectLogin for communication with the LMS. The interface uses HTTP as the communication protocol and returns XML formatted data. If the data consists of multiple files, a compressed file is returned. The solution of MDS for transforming the learning content from the LMS to mobile devices was applicable only for textual data. For other multimedia content the work is still going. For handling involuntary disconnections, a retransmission protocol with error handling capabilities was also designed. The basic idea is while receiving the data, mobile device counts the transferred packages (e.g. number of bytes) and in the case of interruption it saves the position of the last packet completely transmitted. When the connection is re-established it sends a new request to the LMS. Optionally a checksum can be provided by the server after transferring each package and mobile device can compare it with the calculated check sum on the device for detecting any errors occurred during transmission

[12]http://java.sun.com/javame/index.jsp

Chapter 5

Adaptation of eLearning Objects

The main objective of this project is to access eLearning Objects available in the hylOs repository, adapt the content according to the characteristics of a mobile device(iPOD in our case) and render it on iPOD for offline usage. A software application is designed to perform this task and fulfill the following basic requirements.

- It should be able to access eLearning Objects in the hylOs repository, download them to the user′s personal computer and transfer them to the mobile device.
- Before transferring, the content present in the eLearning Objects should be adapted according to the device specifications and requirements.
- All the eLOs should be properly organized i.e the interconnected eLOs in the educational semantic nets have to be arranged in a proper way to provide user with immediate inter object navigation while accessing the related topics for a particular eLO.
- The content should be made accessible on the device with enhanced usability (e.g reducing the number of clicks) with in the device limitations.

Proxy Based Approach

According to W3C[1], for adapting the Web content, three possible approaches exist: server-side, intermediate and client side. The main difference among the three approaches is what the content is and what provides the content. In server-side approach, servers supply directly adapted content, whereas in intermediate (or proxy approach) and client side approach, the input content is obtained from a server or a proxy. Client side adaptation is to be done on users mobile device.

An intermediate approach is chosen to accomplish our task and all the required adaptations (content and structural) are carried out by the Intermediate Proxy (software application) designed in this project. The underlying factors in choosing Proxy-based approach are explained below.

- Adaptation of eLearning content for mobile devices need more computational space and high processing power. Performing these tasks directly on a mobile user agent is not effecitive.
- It allows the content in the repository to be unchanged.
- The mobile device chosen in this project (iPOD) do not have internet connectivity and do not support any programmable interface for communicating with hylOs. Therefore an intermediate service is certainly essential to access the hylOS eLOs and perform all the adaptations before synchronizing them to the device.

Furthermore, the proxy based approach enables the porting of our application into two parts. Depending on the concerned aspects of the mobile device and hylOs server, part of the application can be ported on hylOs and the remaining on a mobile device.

5.1 Adaptation

5.1.1 Transformation of readable content

As discussed in 2.5 the content present in the eLOs of hylOs repository is provided in the Web as different kind of XML documents and can be transformed to desired output formats as HTML Web pages and PDF scripts using XML style sheet programming. To present such content viewable on iPOD display, content adapations have to be done before rendering them to

[1]http://www.w3.org/

the device. As discussed in 2.2.4, iPOD note files support only text content with limited html tags and do not support any kind of colors and fonts, hence in the adaptation process, the raw text content is retrieved from each XML element. A few tags (XML elements) are replaced with iPOD supported html tags (e.g. break line and paragraph tags). While retrieving the content, it is decorated with some special characters wherever necessary in order to distinguish normal text and headings.In this manner, the data present in the XML documents are adapted to the plain text format.

5.1.2 Image Conversion and handling media files

eLOs in hylOs contain media files like image, audio, video and animation. The major advantage of iPOD is its high storage capacity and automatic scaling of images, hence the images are downloaded with original resolutions as available in the hylOs repository. eLOs can have images in various formats. In our thesis we handle only images present in BMP, JPG and PNG formats. iPOD supports images only in JPG formats while storing in its Notes section. Therefore, all the different formats of images are converted to JPEG format using ImageIO[2]. It loads the images from an external image format into Java 2D's internal BufferedImage format and provides a simple way to save images in a variety of image formats.

Audio files in MP3 formats and video files in MPEG 4 formats are also rendered to the mobile device. They are accessed without any conversion. If the audio files are present in other formats then open source mp3 encoders like LAME[3] can be used to encode the audio files to mp3 format, but it has to be done manually by the user.

5.1.3 Table transformation

To represent the table content present in eLOs on iPOD, they have to be transformed to a plain text format. For transforming the tables into iPOD supported notes format with limited html tag support, the following factors have to be considered:

- proportional fonts are used for displaying text on iPOD screens
- number of displayable characters per line varies

[2]http://java.sun.com/docs/books/tutorial/2d/images/saveimage.html
[3]http://lame.sourceforge.net/index.php

Hence, the 'Line width' is fixed to the total average length of allowable maximum width characters and minimum width characters (for both upper case and lower case) per line. The length of each column width is fixed to columnwidth = (Line width / no. of columns in a table). Table is partitioned into rows and columns, such that each cell can be stored into a dimension separately. The cell information retrieved from hylOs XML tables are broken down into an array, the size of which is equivalent to the precalculated 'columnwidth' and the table header is considered as the first row.The boundaries of each cell is displayed using pipeline characters and padding is applied to each cell to align its content equal to the size of the cell which is of maximum length in the table.

The approach followed in transforming the table to text is applicable for all devices independent of the markup language they support. Whereas the solution provided by Girgensohn and Sullivan [15] was implemented for the devices with full markup support. But the main advantage of their solution is, it is applicable for tables with any number of columns and handles nested tables also, whereas in our approach at the moment we tested only for plain tables containing three columns and 'n' number of rows.

5.2 Interaction and Navigation: Deduction of educational semantic net

As discussed in 2.5, any eLO in hylOs is arranged in a guided learning path and attains meaningful relations to the related eLearning objects. Applying mind map based navigation, the semantic structure of an eLO can be visualized as shown in the figure 2.2. Traversing such mind maps using mobile devices requires tying the mind map traversal to the buttons (or any other input devices) available on the device by implementing event handlers. But as discussed in 2.2 iPOD is not provided with any programmable environment for catching events to the buttons. Moreover accessing all the available related topics of any eLO from such small devices is not effective compared to desktop devices. Hence, in this project the complexity of each eLO's semantic net is reduced and while providing the related topics (for any eLO) for offline usage on a mobile device only the most significant topics are selected and fed to the device. To achieve this, the following approach is developed in this thesis.

Approach

A simplified structure of educational semantic net from a single eLO perspective is as shown in the figure 5.1.

An eLO A (say current eLO A) is connected to 'n' number of eLOs in the repository using any of R_i semantic relations ($1 \leq i \leq 18$) and exihibits a $O(n)$ complexity in the first depth of the semantic net of current eLO A. In the figure 5.1 a, b, c... s are subsets of 'n'. Each subset is formed independently by gathering the related eLO's that are having same relations with eLO A.

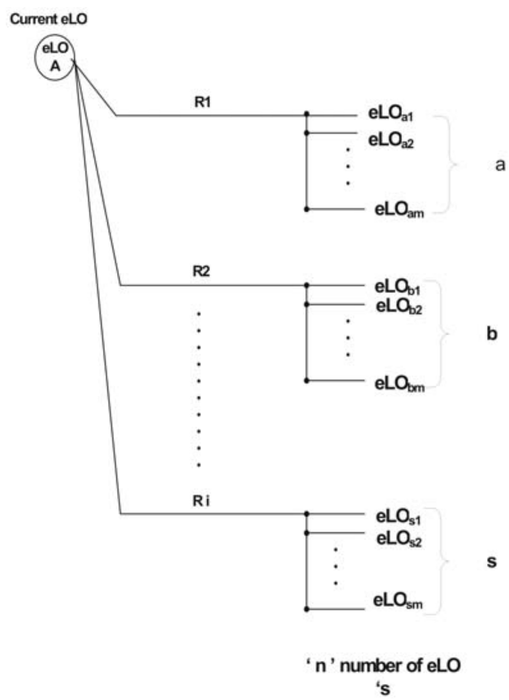


Figure 5.1: Simplified structure for the semantic net of current eLO A

The following points are to be known in advance while working with the eLO's semantic nets shown in the figure 5.1.

- In any of two subsets of 'n', same eLOs may be present if their corresponding relations of the subsets are not inverse pair relations.
- If any eLO (belonging to the subset of 'n') have relations to other eLO's, then it also exhibits the similar structure shown for eLOA and

Navigation	**InterObject relations in hylOs**
Theory	isBasedOn Requires isNarrowerThan illustrates
Apply	isBasisFor isRequiredBy isBroaderThan
Examples	isillustratedBy
References	references

Table 5.1: Projective mapping of semantic relations to navigation fields

thereby providing a $O(n^2)$ complexity in the second depth of the semantic net of current eLOA.

To reduce the navigational complexity of the mesh, the relational pointers are grouped under associate categories and a most significant eLO is extracted under each category. Finally, while rendering the content to iPOD, each eLO is provided with the reference topic (if exists) placed under associated categories within its notes file.

The reduction of semantic net complexity is explained in detail as follows: At first, the meanings of each of the extended LOM relations are carefully examined from the semantics defined for each relation by Engelhardt et al. [12]. Semantically similar relational pointers are selected, grouped and projected onto associative categories. In this process, we realized that the relations ’isBasedOn’, ’Requires’, ’illustrates’, and ’isNarrowerThan’ are used to interconnect the current eLO with the eLO that provide the basic conceptual details and background information for the current eLO. Hence, a category is defined to group these relations and it is given a name ’Theory’. Our reduction algorithm is used to extract the most significant eLO for this category. Similarly, ’isBasisFor’, ’isRequiredBy’, and ’isBroaderThan’ relations are grouped under the category ’Apply’ and the relations ’isillustratedBy’ and ’References’ under the categories ’Examples’ and ’References’ respectively. For each category, a most significant eLO is extracted using the reduction algorithm. The grouped semantic relations and the resultant associate categories are displayed in the table 5.1.

Algorithm

The following assumption is made, before extracting the most significant related eLOs from the available related eLOs of the current eLO A.

- In a general learning scenario, while learning a course, if a related topic is to be selected from a set of topics for any current eLO, then choice is made for a topic, which is fundamental (or required by) or useful for further learning (or for more number of eLOs).

Hence in our approach, following the above assumption, the most significant eLOs are discovered for any eLO from the available educational semantic net. To do this we examined all the 18 relational pointers and determined four relations namely 'IsRequiredBy', 'IsBasisFor', 'IsReferencedBy' and 'IsBroaderThan', which satisfies the above mentioned requirement i.e if any eLO contains these relational pointers then it implies, the corresponding eLO will be useful or basis or required for other eLOs. Therefore, these relations are selected and given weights according to their priorities. For example, 'isBasisFor' relates an eLO carrying fundamental knowledge to another eLO and 'isRequiredBy' denotes an obligatory content dependence in the sense that eLOA cannot be understood without the knowledge of eLOB. Hence, after having a close look at the meaning of these two relations, more weightage is given to 'IsRequiredBy'. A similar approach is followed for all the relations and depending on their priority the weights are assigned to them. In our approach, the priority order of the relations starting from low priority relation to high priority relation and their corresponding weights are given below.

- IsBroaderThan - w1
- IsReferencedBy - w2
- IsBasisFor - w3
- IsRequiredBy - w4

The weights are chosen in such a way that, if the lowest priority relation is assigned a weight w1, then the next priority relations will be assigned weights w2=w1(k+1), w3=w2(k+1), w4=w3(k+1), where 'k' is the maximum number of eLO's that are to be considered under each relation. Thus, to avoid confliction, only equal number of eLO's are considered under each relation and the weights are assigned accordingly. Therefore, in the next step, any eLO containing any of the four relations ('IsRequiredBy', 'IsBasisFor', 'IsReferencedBy' and 'IsBroaderThan'), is assigned with a weight equal to the total weight assigned to the relations which it contains. In this procedure all the related eLOs of the current eLO are assigned weights. The resulting structure of the semantic net with weights assigned to the related eLOs is shown in the figure 5.2.

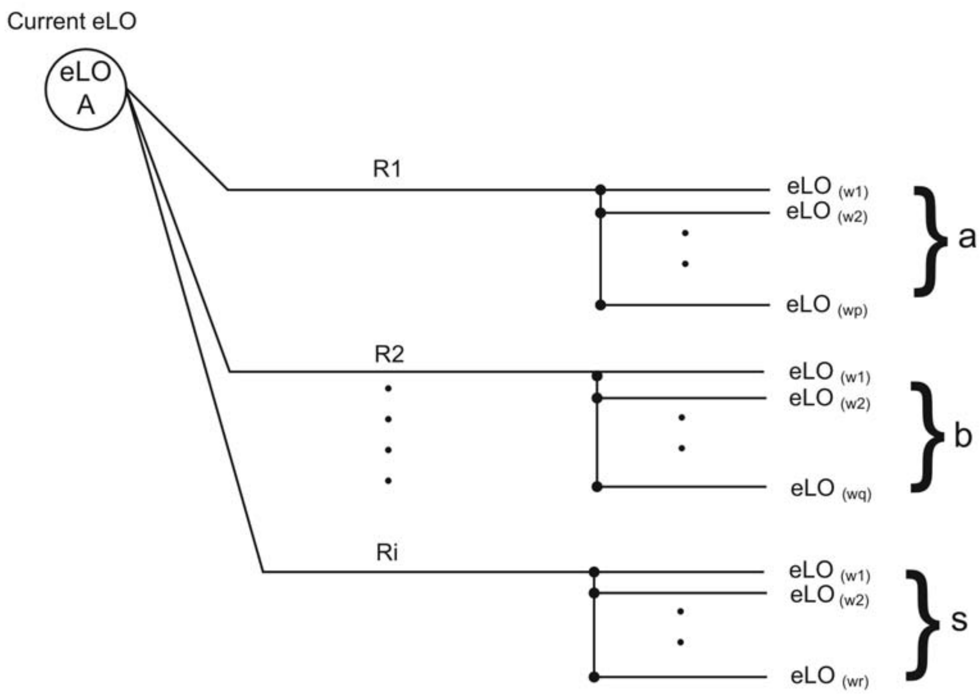


Figure 5.2: Simplified structure for the semantic net of current eLO A with weights assigned to its related eLO's

After assigning weights to all the eLOs, a maximum weighted eLO is extracted under each relational pointer and the resulting structure is as shown in the figure 5.3

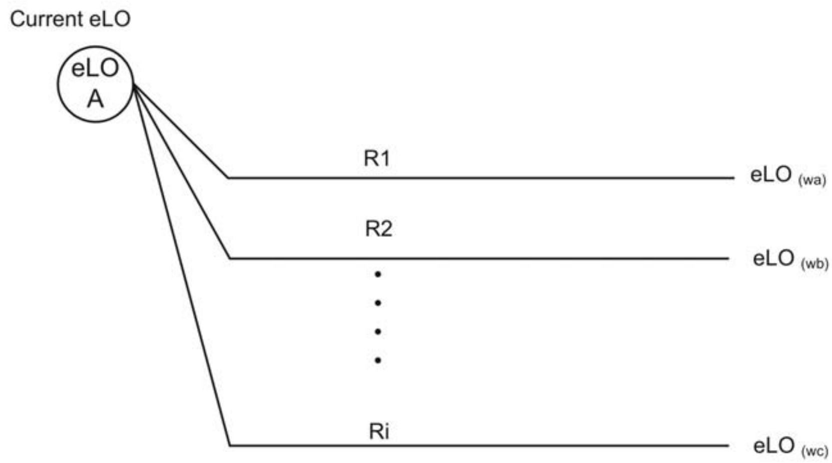


Figure 5.3: Simplified structure for the semantic net of current eLO A with a max weighted eLO extracted under each relational pointer

After extracting an eLO under each relation, the relational pointers are grouped to the associate categories as shown in table 5.1, which results in, each category containing one or more eLOs (maximum eLOs in a category is equal to the number of relations under that category) along with their assigned weights. Now, from each category an eLO with maximum weight is chosen as the most significant eLO for that category. In this process, at any stage if more than one eLO exists with same weight, a random selection is done at that point. The final structure of reduced semantic net from an eLO centered view is as shown in the figure 5.4

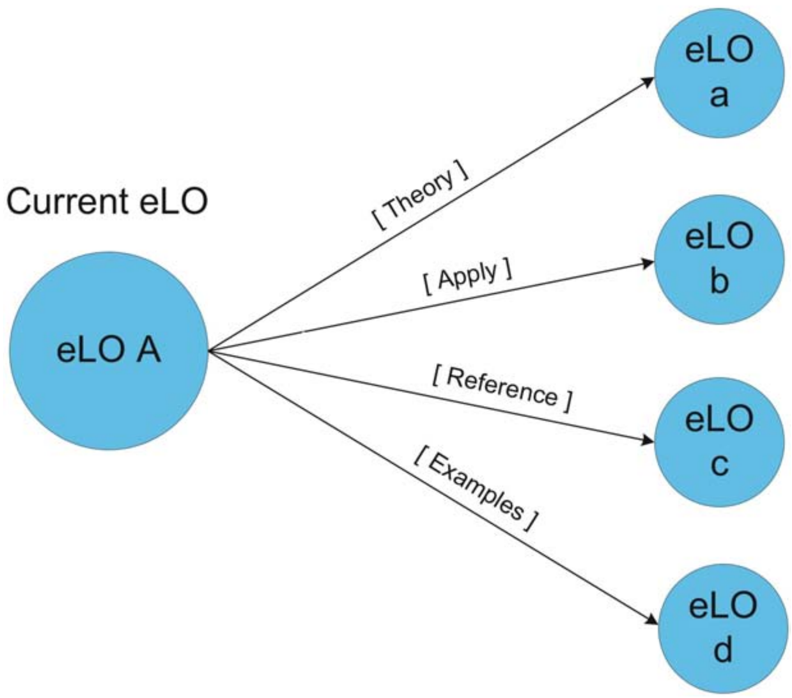


Figure 5.4: Final structure of current eLO A after extracting most significant related topics under associate categories

Additional relations like 'hasVersion/IsVersion', 'IsFormatOf' are not considered in our approach as they donot play any significant role in this context. Other relations like 'hasPart/IsPartOf' do not provide any meaning related to the content, but they express the structural relation of nesting eLOs and can become effective in structuring the eLOs for any course while providing to the user. In our approach, the linear navigation of eLOs present in any course is obtained by following the hierarchy of eLOs in that course as defined by the instructor, hence, 'hasPart/isPartOf' are also not considered. Therefore, while implementing the algorithm the relations 'hasVersion/IsVersion', 'IsFormatOf', 'hasPart/IsPartOf', 'IsAlternativeTo' and 'IsMoreSpecificThan/IsLessSpecificThan' will not be considered and the Semantic navigation and most significant related topics to the eLO of reference are obtained without using these relations.

The approach of deducing the semantic net and providing user with knowledge browsing in an intuitive fashion is advantageous for avoiding the problem of traversing the semantic net through all the branches and deciding upon which related topic to choose. The concept maps based interface provided in the 4.1 is also an option for representing the knowledge information of each eLO but it is used to represent only a chosen domain of knowledge and dedicated to the devices having better user interfaces, whereas our approach is implemented to any number of eLOs that are semantically interconnected and stored in the repository. Furthermore, reducing the net and providing only the most essential information while navigating across the eLOs is useful for devices with less interaction capabilities and other limitations.

Chapter 6

System Architecture and Implementation

6.1 Overview of the System Architecture

The figure6.1 shows an overview of the system architecture.

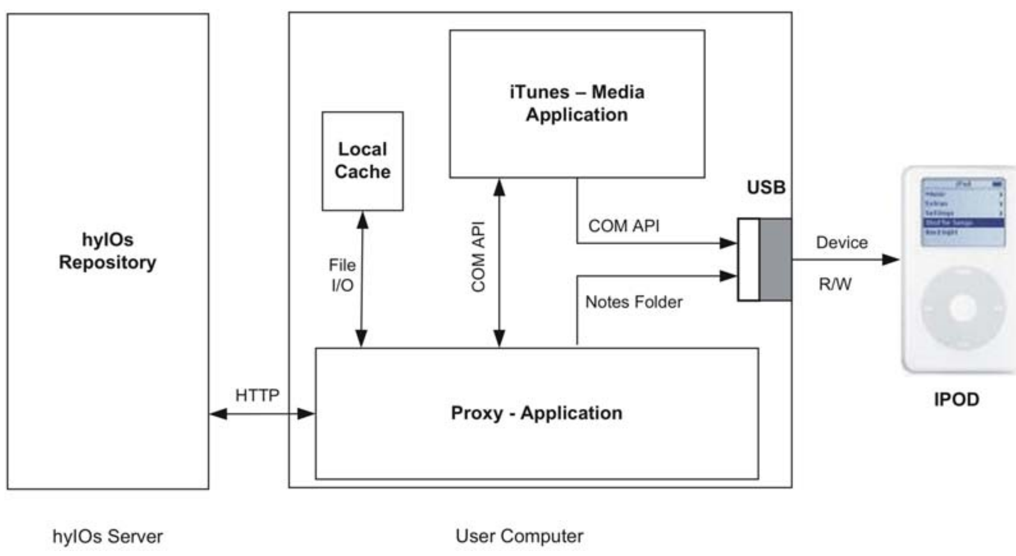


Figure 6.1: An Overview of System Architecture

The main components of the system are

- hylOs Repository
- Proxy
- iTunes Media Player
- Local Cache
- iPOD Notes section

6.1.1 hylOs Repository

hylOs Repository is located at the server side of hylOs and inside the hylOs, an object oriented approach is followed for representing the learning object structure and metadata. To provide the data on Web, hylOs dynamically generates XML documents on the fly. Different kind of XML documents are produced, out of which some representing course indices, course structures, single learning objects or full content trees. In our project the XML documents are accessed from the Proxy and by processing the documents, available content is transformed to the desired output format. Before going into the details about Proxy functionalities, a list of some (generic) URLs from which the generated XML documents are accessible is given below:

a. Course Overview:

http://hylosdev.fhtw-berlin.de/index.debug

This URL generates an XML document which contains the list of courses provided from the repository. For each course, a path for the root eLO of the course is provided with in the <hylos:nav-lo> element. This path together with other generic URL's are used for accessing the remaining XML documents. The XML document with an overview of sample courses is shown in the figure6.2

Figure 6.2: Course Overview XML

A similar URL is shown below which is actually used in this project to access the sample courses (or eLOs) that are stored in the repository.

http://hylosdev.fhtw-berlin.de/tryouts.debug

b. Course Structure as Tree (used for navigation):

http://hylosdev.fhtw-berlin.de/navtree/<path-to-lo>.debug

Using this generic URL, an XML document containing the tree structure of any course is obtained, for which the '<path-to-lo>' field has to be replaced with the corresponding navigation path of the eLO (e.g path present in the <hylos:nav-lo> element of course overview XML document).

c. Content Pages:

http://hylosdev.fhtw-berlin.de/content/<path-to-lo>.debug

Using this generic URL, an XML document containing the content of individual eLO is obtained, for which the '<path-to-lo>' field has be replaced with the corresponding navigation path of the eLO (e.g path present in the <hylos:path> element of any eLO present in the tree structure of a course).

d.Semantic Relations:

http://hylosdev.fhtw-berlin.de/navcons/<path-to-lo>.debug

Using this generic URL, an XML document containing the semantic relations of an individual eLO are obtained, for which the '<path-to-lo>' field has be replaced with the corresponding navigation path of the eLO (e.g path present in the 'hylos:path' element of any eLO present in the tree structure of a course).

6.1.2 Proxy

The Proxy is located on a computer at the client-side. The main operations performed by the proxy are given below.

- It communicates with hylOs through HTTP and automatically access XML documents containing the content, structure and semantic relation of an eLO by using the corresponding URLs.
- Access the eLOs content and writes the content in appropriate files created dynamically on the local disk(local cache) of clients PC by using required File I/O operations.
- While downloading the data it performs adaptation of the eLOs content and structure suitable for accessing on iPOD.
- Transfer the note files, image files to Notes folder of iPOD and media files are synchronized to iPOD via iTunes media player.

An overview of the Proxy-application is explained by an use case diagram shown in the figure 6.3

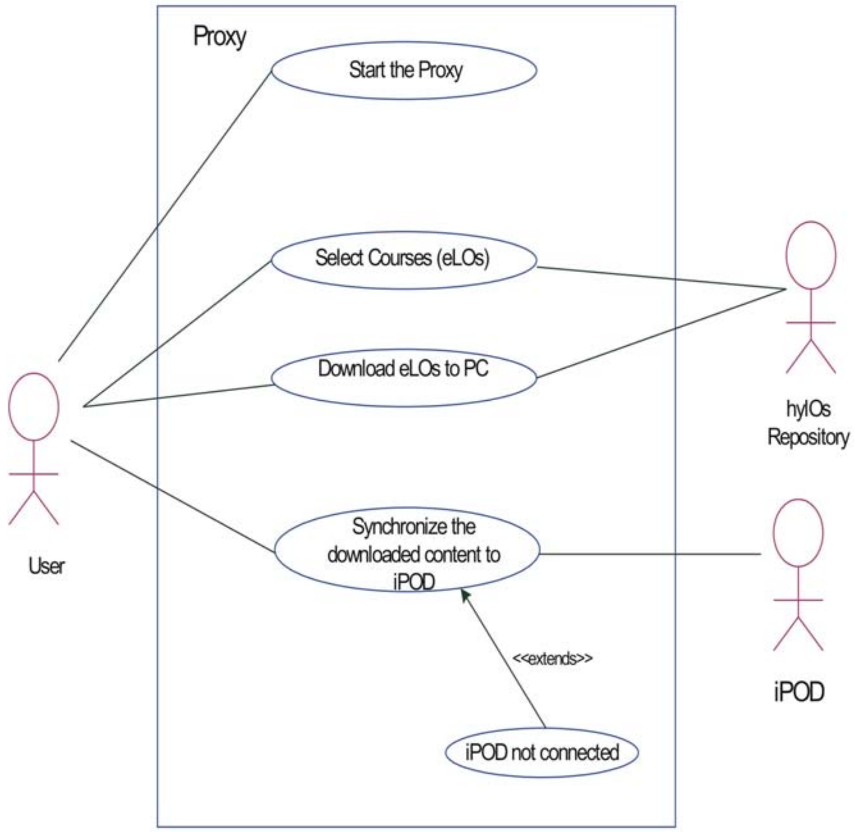


Figure 6.3: UML Use case diagram for Proxy

Upon starting the Proxy-application, user can choose a directory on the Personal Computer for storing the data. The proxy will access the XML documents generated from the hylOs repository, process them and provide the user with a list of courses(eLOs) available in the repository. User can select the courses one by one and download them to the local disk on PC. Before feeding the content to iPOD, Proxy performs the following operations based on the characteristics of iPOD:

- Transformation of readable content to plain text format decorated with suitable html tags.
- Transformation of table content to plain text format.
- Image Conversion and handling media files.

The adapted content of all the eLOs are stored in the local disk on clients PC in a Notes file hierarchy supported by iPOD Notes section. After connecting iPOD to the computer, Proxy transfers the note files, image files to the iPOD Notes section and contols the iTunes application to import the media files the local disk to iTunes Library. The media files are then synchronized automatically to the iPOD via iTunes media player.

6.1.3 Local Cache

Instead of downloading the content directly to iPOD, an intermediate local cache is maintained on the user's PC to store the content of eLO's. The cache is filled with a set of files and folders for storing the content of eLOs in a notes hierarchy similar to iPODs Notes section. The readable content is stored as note files and images and media files are stored in the formats supported by iPOD (JPG, MP3 and MPEG-4). Tranferring of the content from local disk to the iPOD can be enabled only after downloading the entire content on local disk.

6.1.4 iTunes Media Player

iTunes Media Player is used for synchronizing the audio files download on the local disk to iTunes library and from the library to iPOD. The copying of audio files into iTunes music library is initiated by Proxy using COM API. The audio files are copied from iTunes library to iPOD automatically and are arranged in music playlists of iPOD.

6.1.5 iPOD Notes section

The basic components of iPOD Notes section [1], that are useful for storing eLO content are given below:

- **note**: a plain text file which typically end with a .txt extension. These not files supports limited HTML tags.
- **folder**: a regular folder, just as in Mac or PC, that may contain files and/or other folders.
- **Notes folder**: a specific folder, named Notes, that is present on the iPod, which serves as the root or home folder for all Notes feature content on iPod.

[1] http://developer.apple.com/hardwaredrivers/ipod/

- **media file:** a standard-format graphic, audio, or videofile.
- **link:** a hyperlink logically connecting one note to a folder, a media file, or another note.
- **markup:** a set of plain-text character sequences used to specify HTML-style commands to affect the behavior of the Notes feature.
- **.linx note:** A note used to specify a menu.

6.2 Implementation

The concepts and functionalities of the Proxy, as described in the section 5 and 6.1.2, have been implemented as a software application. JAVA and C Sharp technologies are used on Eclipse and Visual Studio.NET platforms for our implementation. A user interface is designed using Java Swing for accessing, downloading and synchronizing the courses available in the hylOs repository. All the XML documents carrying the content pages of an eLO and other essential data were processed in JAVA with DOM parser [2] . Also, SAX and JDOM[3] parsers are utilized wherever necessary. This section presents a detailed description of how the implementation of all the concepts and functionalities discussed in 5 was done.

6.2.1 Adaptation

As discussed in 5.1, the adaptation includes the transformation of readable content, table transformation, image conversion and handling of media files. hylOs uses the following XML elements for storing the eLO's content:

- <content> element : It consists of the the eLO content placed under the elements<section>, <subsection>, <subsubsection>(which contains data to be presented as headings, sub headings in an eLO), <paragraph>, <table> and <richmedium > elements.
- <paragraph> element : It contains the textual content decorated with fonts, break lines and other XML elements. With in the <paragraph> element, the information about the images are embedded in an <image> element

[2]http://java.sun.com/
[3]http://www.jdom.org/

- <table> element : With in this element, the content to be displayed in table format is stored using <header>, <row> and other elements
- <richmedium> element : It consists of information regarding the media content (e.g references to audio/video and animated files)

All the XML documents, were processed by using DOM parser. It loads the document into the computer's memory and manipulates it using the DOM. It treats everything in an XML document as a node and the entire document as node tree. The following transformations were done while parsing the XML document and the resultant content obtained from each transformation was dynamically stored together in a note file on the local disk.

6.2.2 Transforming the data present in <content> element

The XML schema for the 'content' is shown in the figure6.4

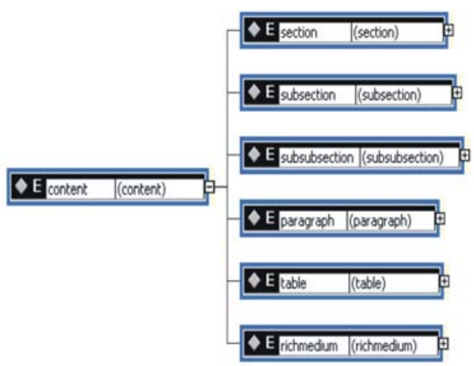


Figure 6.4: XML schema for content

The <content> node tree was traversed and by looping through all of its child nodes, the text content present in the each node was extracted and decorated with special characters. The child nodes like <section>, <subsection> and <subsubsection> contains the data to be displayed as headings. While traversing the tree if any of these nodes are found in the XML document, then its text content was retrieved and appended with a special character (e.g ':') and a break line tag. The manipulation of data present in other nodes of content is explained below.

6.2.3 Transforming the data present in <paragraph> element and Handling Image files present in it

The XML schema for the paragraph is as shown in the figure 6.5

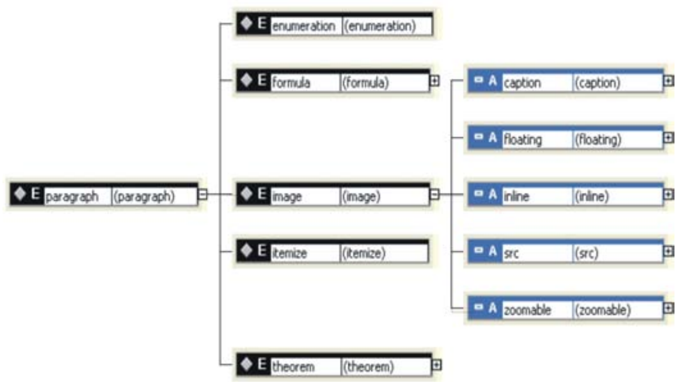


Figure 6.5: XML schema for paragraphs

Manipulation of the child nodes of <paragraph> node and their data is explained as follows: The text content present in the <enumeration> or <itemize> nodes is retrieved node by node and a break line tag was placed every time an <item> node appears. Presence of any font elements were ignored and only their raw text content is retrieved, because iPOD note files donot support any fonts and colors.

<image> node is provided with attributes which contains the caption and the source path of the image. By traversing the image node, the source path of the image was retrieved from the attribute 'src' and appended to the following generic URL to obtain an URL for retrieving the image for that eLO.

http://hylosdev.fhtw-berlin.de/data<ImgsrcPath>

While retrieving the image from the URL , the conversion of images to JPG format was performed and saved on the users computer. The conversion of images to JPG format was implemented by using Java Image IO package [4] as explained in 5.1.2. The following code snippet shows how the image conversion was done.

[4] http://java.sun.com/docs/books/tutorial/2d/images/saveimage.html

```
URL sUrl = new URL(srcUrl);
URLConnection urlCon = sUrl.openConnection();
DataInputStream dis = new DataInputStream(urlCon.getInputStream());
File f_out = new File(localFileName);
BufferedImage bi = ImageIO.read(dis);
ImageIO.write(bi, "jpg", f_out);
```

The downloaded image was transferred to the iPOD Notes folder. In the note file of the eLO, a link for the image was given by creating an anchor element with a link tag pointing to the filename of the image file. A sample link path is shown below

```
<A HREF="/ImagefileName">Image Title</A>
```

In the above link 'ImagefileName' was replaced with the name given to the image file when downloade to local disk. The 'Image Title' field was filled with the title of the eLO which was obtained by traversing the 'title' node of the eLO and obtaining the text content of it.

The link thus obtained from the above procedure was placed in the note file of the corresponding eLO. Other nodes in the 'paragraph' node tree like 'theorem' and 'mir:link' were not handled in this thesis.

6.2.4 Transforming the data present in the <richmedium> Element and Handling Media files present in it

The XML schema for the rich medium node is as shown in the figure 6.6

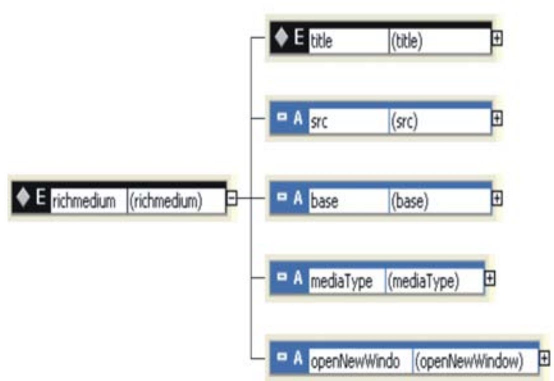


Figure 6.6: XML schema for richmedium node

This node consists of the attributes specifying the type of the media it is holding and the source path of that media file. While traversing the tree if this node appears, the parser looks for the 'mediaType' attribute value and if it detects a string with the value 'audio' or video, then its reference path is retrieved and appended to the URL given below.

http://hylosdev.fhtw-berlin.de/data

The resultant URL was used to access the media file and download the file to the local disk on the user computer. The following code snippet shows how media files are downloaded.

```
URL url = new URL(srcUrl);
OutputStream outstr = new BufferedOutputStream
      (new FileOutputStream( localFileName));
URLConnection urlCon = url.openConnection();
InputStream in = urlCon.getInputStream();
byte[] buffer = new byte[1024];
long numWritten = 0; int numRead;
while ((numRead = in.read(buffer)) != -1){
    outstr.write(buffer, 0, numRead);
    numWritten += numRead;
}
```

The media files downloaded to the local disk were transferred via iTunes to the iPOD. These media files were accessible on the iPOD in the audio and video menus. But it is not effective to access the media files by manually navigating the audio and video menus. Hence we provided a linking to the media files from the note file of the corresponding eLO creating the anchor element similar to the one created for Image. The sample anchor element for linking to the audio file is as shown below.

<A HREF="song='Audio/Video' filename">Audio/Video Title</A>

The 'Audio/Video filename' was replaced with the corresponding name given when file downloaded to the disk and 'Audio/Video Title' were replaced with the title of the corresponding eLO. For audio files, the ID3 tags used in the mp3 audio files were edited. The 'title' field was set to the eLO title (and also the album field was set to 'hylOs'). To set the ID3 tags of the mp3 audio files, we used Java ID3 library[5].

[5] http://jid3.blinkenlights.org/

Transfering Media files to iTunes

All the media files were downloaded to a directory created in the user selected path on local disk. To transfer both audio and video files from local disk to iPOD, we used iTunes media player. The reason for transferring media files via iTunes is, iPOD donot have any linking scheme for the media files, if they are stored in its Notes section. Hence all the media files stored in the local disk were transferred one by one to iTunes library and then they were synchronized automatically to iPOD. To control and perform all the required actions on iTunes media player, iTunes COM API[6] was used and the coding for all these operations was written in C sharp[7]. The following code snippet was written to in C sharp to launch the iTunes application and import the audio files from local disk to iTunes library.

```
//ITunes Object is created
iTunesApp apl = new iTunesAppClass();
//IITLibraryPLaylist Object is created
IITLibraryPlaylist lib = apl.LibraryPlaylist;
DirectoryInfo dir = new DirectoryInfo(MedFilesPath);
if (dir.Exists)
{
    FileInfo[] files = dir.GetFiles();
    for (int i = 0; i < files.Length; i++)
    {
        string file = files[i].Name;
        lib.AddFile(MedFilesPath + "\\" + file);
        //add media files to iTunes lib
    }
}
```

6.2.5 Transforming the data present in the <table> element:

The XML schema for the table node is as shown in the figure 6.7

[6] http://developer.apple.com/sdk/itunescomsdk.html
[7] http://msdn2.microsoft.com/en-us/vcsharp/default.aspx

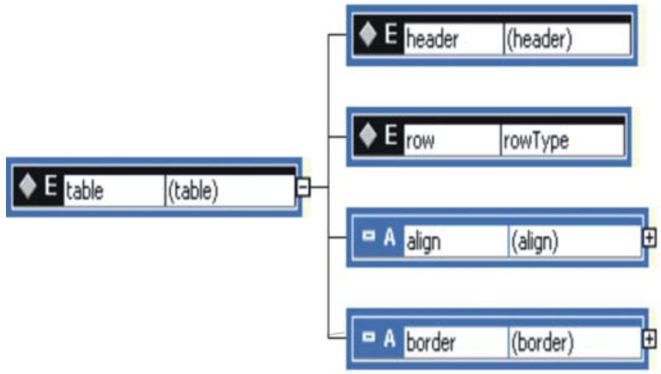


Figure 6.7: XML schema for table

The approach explained in 5.1.3 was implementing while traversing the child nodes of the 'table' node tree. The actual content along with the special characters used for padding and cell separation ('asterisk' and 'pipeline' placed in appropriate places) were appended in a string buffer. After processing the entire XML document carrying the content pages of an eLO, the string buffer contains the transformed content and it was written to a note file in iPOD Notes section.

6.2.6 Implementation of Reduction Algorithm

The XML document of an eLO containing the semantic relations with its related topics and their reference paths is as shown in the figure 6.8

```
<?xml version="1.0" encoding="UTF-8" ?>
- <hylos:hylos-navigation xmlns:hylos="http://hylos.fhtw-berlin.de">
  + <hylos:lo
      hylos:path="/HyLOs/content/Data/Hypermedia_History/TheWeb/TheWeb"
      hylos:title="The Web">
  - <hylos:lo-list hylos:title="Requires">
    + <hylos:lo
        hylos:path="/HyLOs/content/Data/Hypermedia_History/WhatHyper/WhatHyper"
        hylos:title="What is Hypermedia?">
    </hylos:lo-list>
  - <hylos:lo-list hylos:title="IsPartOf">
    + <hylos:lo
        hylos:path="/HyLOs/content/Data/Hypermedia_History/HyperHisto/HyperHisto"
        hylos:title="Hypermedia History">
    </hylos:lo-list>
  - <hylos:lo-list hylos:title="HasPart">
    + <hylos:lo
        hylos:path="/HyLOs/content/Data/Hypermedia_History/TheWeb:OpenIssue/TheV
        hylos:title="The Web: Open Issues">
    </hylos:lo-list>
  - <hylos:lo-list hylos:title="References">
    + <hylos:lo
        hylos:path="/HyLOs/content/Data/Hypermedia_History/PioneHyperSyste/PioneH
        hylos:title="Pioneers of Hypermedia Systems">
    + <hylos:lo
        hylos:path="/HyLOs/content/Data/Hypermedia_History/HyperHisto/HyperHisto"
        hylos:title="Hypermedia History">
    </hylos:lo-list>
  </hylos:hylos-navigation>
```


Figure 6.8: Semantic Relations XML document of an eLO of reference

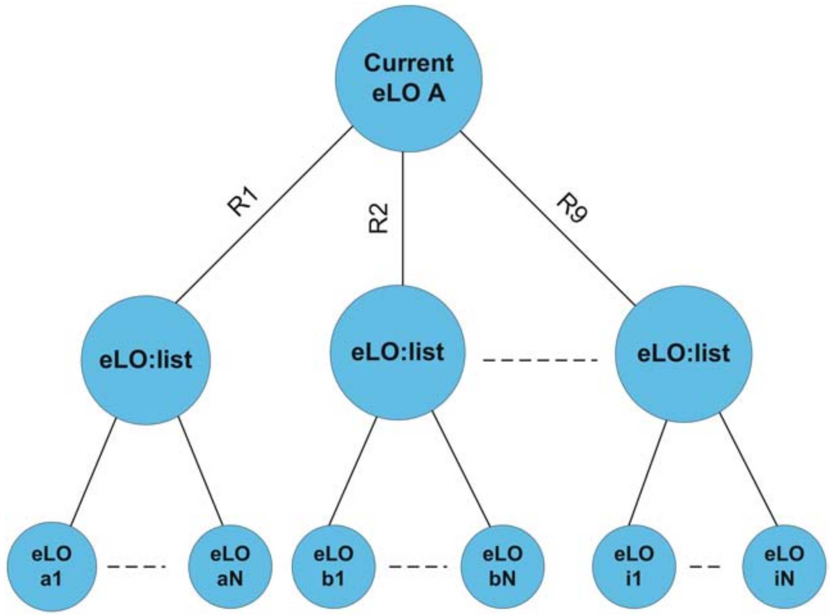
In the XML document shown in the figure 6.8, the related topics of the current eLO of reference are stored under 'hylos:lo-list' independently for each relation. Each related topic again contains similar XML document with its related topics and the semantic relations. To obtain the most significant topics for the current eLO of reference, we used data structures in our implementation. The implementation of our algorithm was done in 3 steps

STEP 1:

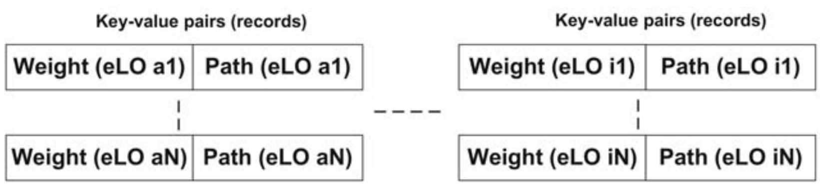
All the related topics (or eLO's) of the current eLO were assigned weights (depend on the qualified relations they hold) as explained in 5.2.

STEP 2:

In the second step, for each set of eLO's present under each relation of current eLO A, required number of hashtables are used which store the assigned weights and reference paths of corresponding eLOs (with weight as key). Figure 6.9a shows the tree structure of the eLO of reference with its relations and related topics and figure6.9b shows the hashtable with the key value pairs(records) of the related eLOs



(a) Tree strcuture of eLO of reference with relations and related eLOs

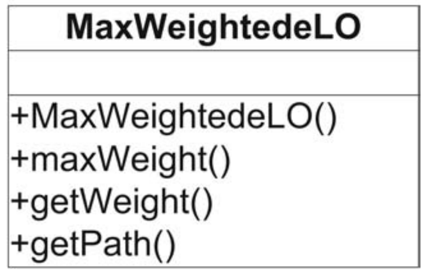


(b) Hashtables containing weights and reference paths of each eLO.

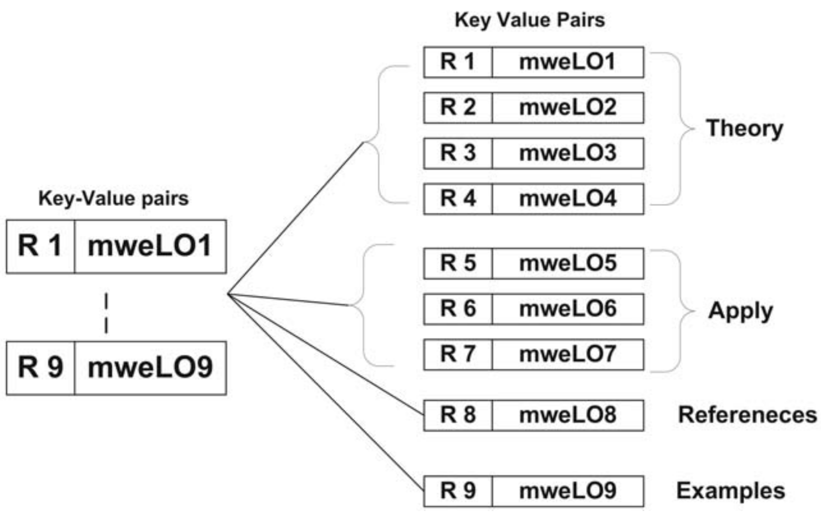
Figure 6.9

STEP 3:

The weights present in each hashtable were stored separately in arrays and a maximum weight is obtained from each array. Using the maximum weight extracted from each array, the corresponding path was obtained from the respective hashtable records. Both weight and the path obtained from each hashtable were stored separately. A Java class was written for performing all the operations mentioned in this step and an Object was instantiated every time to find the maximum weight from each hashtable, the corresponding maximum weight and path was retrieved whenever necessary by using the instantiated Objects. The UML class diagram for that class is shown in the figure6.10a. Each relation of the current eLO A and an



(a) MaxWeightedeLO class



(b) Hashtables containing Relations of eLO A and instantiated objects of MaxWeightedeLO class, grouped under different categories based on the relations.

(c) Hashtables containing weights and paths of related eLOs projected under associate categories.

Figure 6.10

instantiated object of the Java class containing weight and path of the resultant maximum weighted eLO(extracted from the set of eLOs), were stored as key value pairs in another hashtable as shown in the figure6.10b and also shows how the resultant hashtables were grouped into associate categories based on the approach explained in 5.2. Now for each category of hashtables there exists a set of instantiated Objects of the class *MaxWeightedeLO* . From this class, the weights and paths of the maximum weighted eLO extracted under each relation were retrieved and stored them as a key value pairs in another hashtable as shown in the figure 6.10c

By storing all the weights again in an array, a maximum weight was extracted and based on the weight (since it is the key), the corresponding path is extracted from the class *MaxWeightedeLO* using the instantiated object and placed under the respective category. Thus, a most significant related eLO was extracted and placed under each category of the current eLO A and the resultant simplified structure of the current eLO A is as shown in the figure 5.4.

The implementation of deducing the semantic nets for each eLO and other important concepts implemented in our application is also explained by means of an UML class diagram in the last section of this chapter.

6.2.7 Arrangement of eLOs content in hierarchical files and folders

The content present in eLOs were stored on the users local disk by creating two folders. One is a 'Notes' folder which contains the images and note files of eLO and the other one is a 'DLoad' folder containing the media files (audio and video). All the images present in the eLO content were stored in the root Notes folder. The text content was stored as note files in the sub directories created in the root Notes folder with a directory path name same as the eLO's corresponding reference path in hylOs repository. For each eLO the corresponding reference path is available in the content page XML document and was retrieved by traversing through corresponding nodes in the document. This reference path was processed and the note file having the eLO content was stored in a hierarchy as defined in the hylOs reference path. This process was repeated for each eLO by using its corresponding reference path. Figure6.11 shows the hierarchy of files and folders created for an eLO with the reference path given below.

"/HyLOs/content/Data/masters/kumar/DOM/DOM"

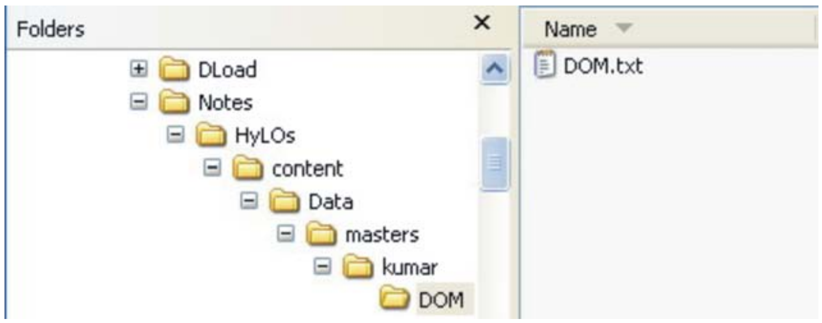


Figure 6.11: An example of Notes hierarchy created for an eLO

Customized menus for hylOs courses

To provide course menus in iPOD Notes folder, the XML document containing the list of courses in hylOs repository was parsed and from each course node, the title of the course and reference path of its root eLO were retrieved and the list of courses were presented to the user on a window in the swing interface. Now, every time user selects a course, a note file was created for each selected course with a '.linx' extension. A 'main.linx' file was created in which anchor elements were written such that each anchor element contains a link tag pointing to the '.linx' file of the course selected by user. And in each '.linx' file of the course, an anchor element was written with a link tag pointing to the the root eLO of the course. Now to display all the course menus in iPOD Notes folder, all the '.linx' files were stored in the root Notes folder.

Linking of note files:

The whole linking scheme adopted for the eLO note files, folders and media files was based on the components of iPOD Notes section discussed in 6.1.5 . An anchor element was used to link one note file to another note file in iPOD. A sample element is given below

"<A HREF="/note file Path">note file name.</A>"

In one eLO note file the above anchor element is placed with the 'note file name' and 'note file Path' replaced by the path and name of another eLO note file that is stored in the Notes folder hierarchy. In this way we linked two eLO's note files. This linking scheme was implemented in providing navigation links for both linear and semantic navigation as shown in figure6.2.7

Linear Navigation

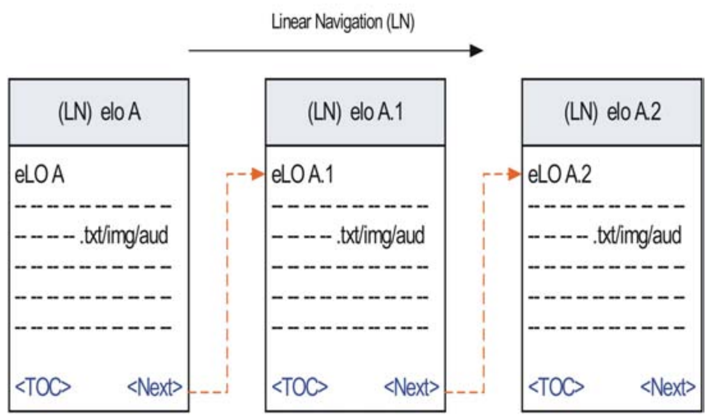
The eLOs of a course were structured as a tree as defined by the author. To navigate between all the eLOs of a course a linear navigation was provided for which the course tree XML document was traversed and each eLOs node was parsed and the title and reference paths were obtained from its attributes and child nodes. All the titles and paths of the eLOs were stored in a hash table and while writing content to each note file of the eLO, an anchor element containing a link tag pointing to the reference path of the next eLO stored in the hashtable was written. And for the last eLO note file, the anchor element was written with a link tag pointing to the root eLO of that course. Additionally, for each note file an anchor element containing link tag pointing to the 'main.linx' file was written in order to provide the navigation to the index of courses that are synchronized to iPOD. A title element was written into the note file of each each eLO as shown below.

<TITLE> (LN)title </TITLE>

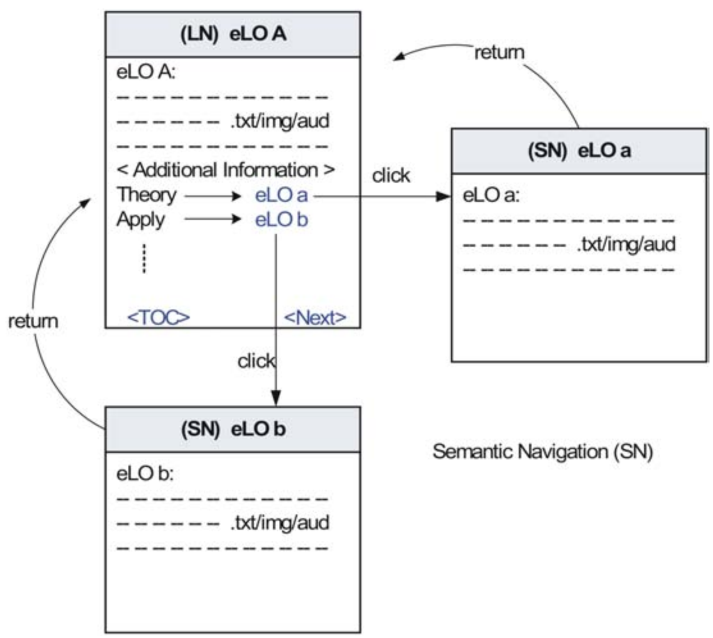
The 'title' was replaced by the corresponding title of the eLO and a string '(SN)' was appended at the beginning of the title to indicate, the user is in Linear navigation and accessing a eLO present in the hierarchy of a course as defined by author. The entire text written in the title element will be displayed on iPOD as the title of the corresponding note file, instead of the actual filename given to the note file. Finally, the expected linear navigation on iPOD for an eLO present in the XML tree of a course is as shown in the figure6.12a with hyperlinks to the next eLOs, root eLOs and course index.

Semantic Navigation: Handling note files of related eLOs

After determining the most significant eLO's for any eLO from the 6.2.6, the corresponding hylOs reference path was retrieved and processed to obtain the URL for the XML document containing the content page of the corresponding eLO . Using the URL the content Page XML document was parsed and the content was written to a notes file. The resultant note file was stored in the directory of the eLO of reference. Figure6.13 shows an eLO 'DOM' and its related topic 'XML' stored in the same directory of eLO 'DOM'



(a) Linear Navigation of eLOs with in a course A



(b) Semantic Navigation of the root eLO of the course A.

Figure 6.12

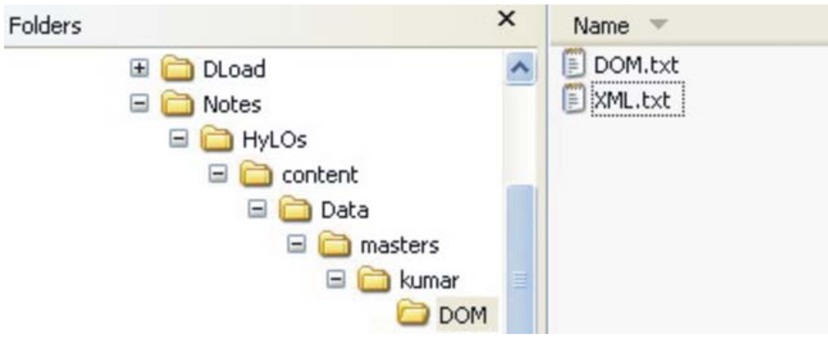


Figure 6.13: An example of Notes hierarchy created for an eLO with the note file of the related topic stored in the same directory

Now, for each eLO which contains the related topics, the hyperlinks for the semantic navigation were providing by writing an anchor element into its note file with link tags pointing to the note files of the related topics that are stored in the same directory of the eLO of reference. Additionally, a title element was written into the note file of each related topic as shown below.

<TITLE> (SN)title </TITLE>

The 'title' was replaced by the corresponding title of the related eLO and a string '(SN)' was appended at the beginning of the title to indicate, the user is in semantic navigation and accessing a related topic. The entire text written in the title element will be displayed in iPOD as the title of the corresponding note file, instead of the actual filename given to the note file. The expected semantic navigation for an eLO containing related topics is as shown in the figure6.12b, in which only the content is displayed and no links are provided for further linear or further semantic navigation of that particular related eLO page.

Reason for storing the note file of an eLO and the note file of its related topics in the same directory : As shown in the figure 6.13. instead of placing the note file of the related topic in a directory path obtained from its hylOs reference path, they were stored in the same directory as the note file of eLO of reference. The main reason for it is explained as follows:

- If an eLO is existing as one of the several eLOs that are present in the tree hierarchy of a course and if the same eLO was extracted as the most significant related topic for an eLO, then it is not possible to provide only a single note file for that eLO with all the required navigation links one time(for linear navigation) and with no navigation links another time(for semantic navigation).

Hence, to address such situations we first created note files of eLOs with all the content and required navigation links in it and then the eLOs which are the most significant related topics for any particular eLO were determined and then another note file (duplicate note file if it already exists) was created for that eLO with the required content written to it, excluding the navigation links (because we do not provide any further linear or semantic navigation for these related eLOs). Similarly the media files present in the related eLOs were stored independently in the DLoad directory on users local disk, but all the images were stored in the root Notes folder distinguished by names.

6.2.8 Class diagram

Figure 6.14 shows the UML Class diagram for the important java classes written in our implementation. For abstract explanation of the classes, attributes were not mentioned. The parameters and return types of the methods were not listed in the figure, but they are mentioned in the explanation.

- Class *DownloadeLOs* consists a method *createFolders()*, which creates required note files and folders for the user selected eLOs. *DownloadeLOs* has a one way aggregation association with the classes *ReductionAlg* and *ContentParser*. The latter classes can be accessed with the object names 'redAlg' and 'conParser' instantiated in the *DownloadeLOs*. The content received from the class *ContentParser* and class *ReductionAlg* are written to the eLO note files by *filehandling()*, which receives the directory path name for the files from the *createFolders()* and writes the content into the files created.

- Class *ContentParser* has one way aggregation association with the class *FileDownload* and access it by the object 'fdLoad'. The *getContent()*, parses the XML document containing the individual content pages of eLOs and transforms the content into plain text format. Whenever media files or image files are detected, it sends the source and destination paths of those files to the *FileDownload*. Finally, it returns the transformed text content (decorated with the links created with some html tags for accessing the image, media files when presented to iPOD) as a string buffer to the *DownloadeLOs*.

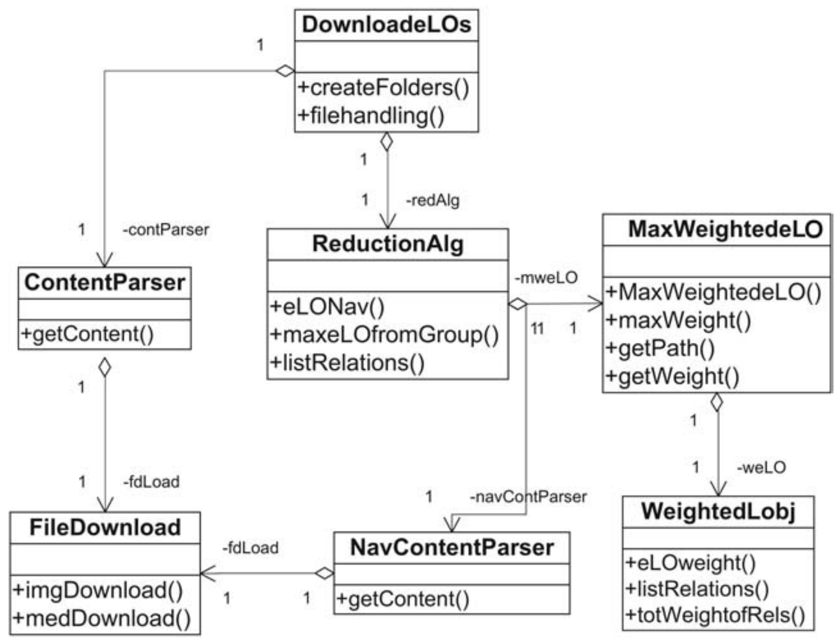


Figure 6.14: UML class diagram: Important Classes describing the implementation of our software application

- Class *FileDownload* contains *imgDownload()* method which reads the image file from the source path it received, converts the images to JPG and downloads the resultant image to a destination directory. It also contains *medDowload()* method which downloads the media files directly into the local disk with out any conversions.

- Class *NavContentParser* retrieves an eLO's reference path from the class *ReductionAlg* and performs the similar operations as *ContentParser*, but stores the content (related eLO content) in the directory of the eLO for which the related topics were extracted from the class *ReductionAlg*. It has a one way aggregation association with the *FileDownload* and access it by the object fdLoad for retrieving and downloading the images, media files present in the eLO content.

- Class *ReductionAlg* has one way aggregation association with the class *MaxWeightedeLO*. The *listRelations()* parses the XML document containing the qualified relations of an individual eLO. *eLONav()* method forms the associate navigation categories by grouping the relations and the most significant eLO paths obtained for each relation from *MaxWeightedeLO*. *maxeLOfromGroup()* extracts the maximum weighted eLO from each group and returns the reference path of corresponding eLO to the *ReductionAlg*

- Class *MaxWeightedeLO* has a one way aggregation with the class *WeightedLobj*. The latter class can be accessed with the Object name weLO from the same *MaxWeightedeLO* . The constructor MaxWeightedeLO() receives a list of eLO's reference paths from the *ReductionAlg* . For each eLO, it retrieves a weight from the *WeightedLobj* and calculates the maximum weight of all the eLOs by using the *maxWeight()*. The *getWeight()* and *getPath()* returns the maximum weight of eLO and its path which was set by the *maxWeight()*.

- Class *WeightedLobj* has a method *eLOWeight()* which receives eLO reference path from *MaxWeightedeLO*. It parses the semantic relations XML document of the corresponding eLO. It checks for a specified relation and pass the count of number of eLOs under that relation as an integer to the *totWeightofRels()* and receives the weight for that relation depending on the number of eLOs under that relation. It accumulates the weights of the specific relations and assigns as the total weight for eLO containing those relations. The *totWeightofRels()* calculates weights for relations based on the priority of the relations and based on the integer it received and returns the calculated weight to the *eLOweight()*, which accumulates weight of the relations of eLO and assigns the total weight of its relations as eLO's weight and returns the weight to *MaxWeightedeLO*.

Chapter 7

Test and Evaluation

7.1 Functionality testing

In this section the functionality of the software application is shown. Upon starting the application, a swing interface appears with a 'FilechooserDemo' window providing the user with a button to select a local directory. The eLOs of hylOs that will be accessed in the next step will be downloaded to the local disk under the selected directory. The 'FilechooserDemo' window is as shown in the figure7.1

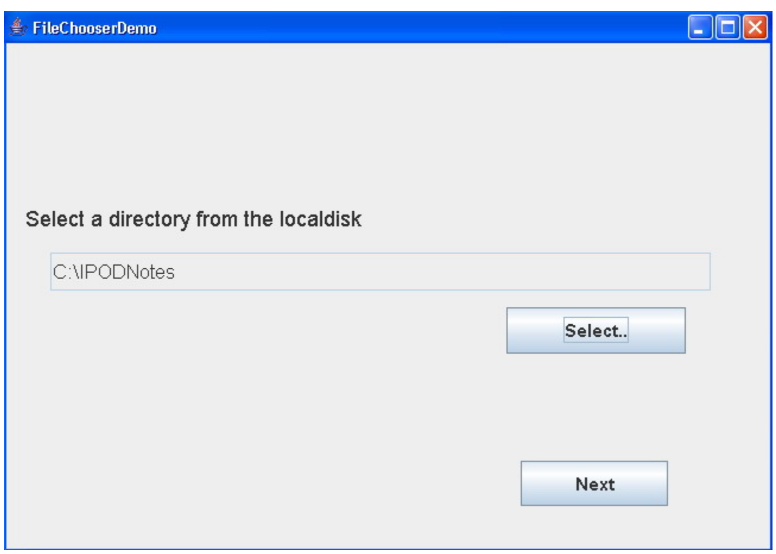


Figure 7.1: Filechooser window

After choosing a directory and pressing 'Next' button hylOs Courses window appears as shown in the figure 7.2.

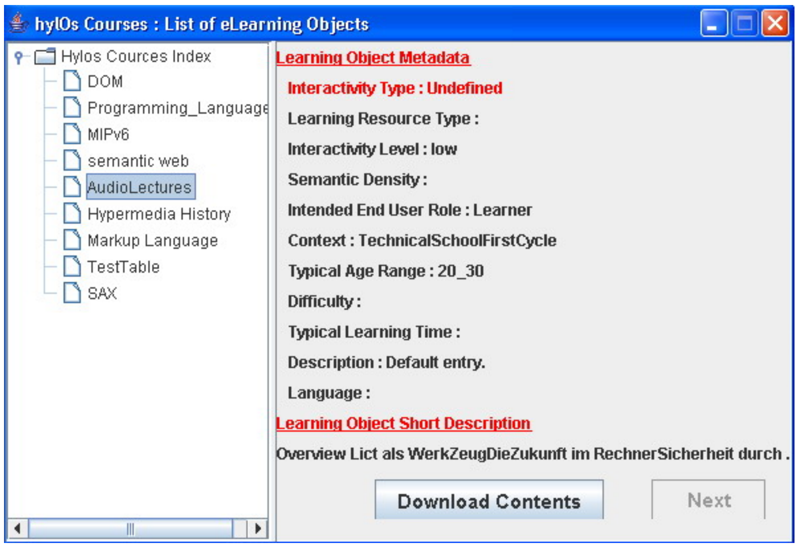


Figure 7.2: hylOs Courses

It displays the available courses(eLOs) in a tree structure on the left side of the window. From the course tree, any course can be selected. The metadata information of the selected course is displayed on the right side of the window. After selecting a course, and pressing the 'Download' button, the content of the course will be downloaded to the selected directory(from the filechooser window) on the local disk.

File hierarchy:

While downloading the content present in the eLOs, they are arranged in the local disk in a hierarchy of files and folders. It is shown in the figures 7.3 and 7.4 for selected courses.

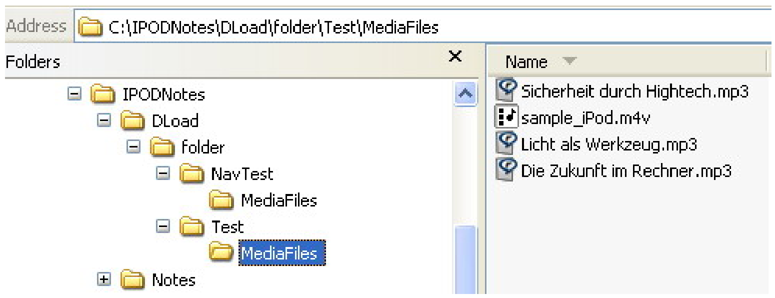


Figure 7.3: An example : Files and folders created along with DLoad directory on the local disk for storing media files

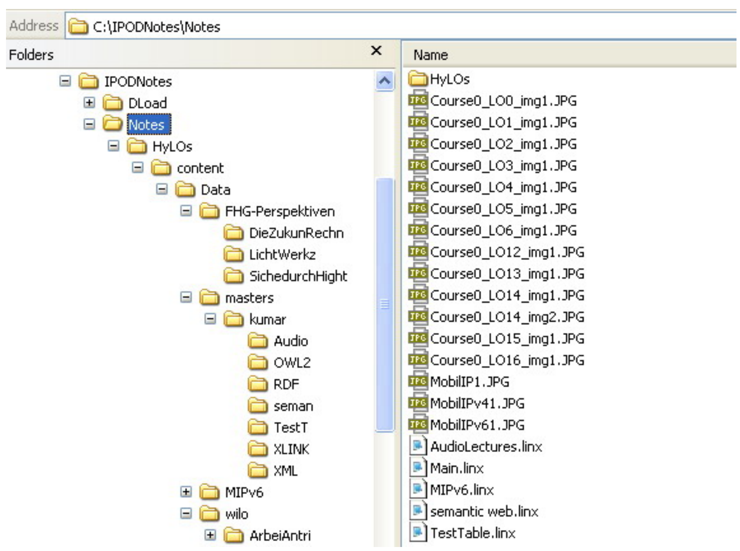


Figure 7.4: An example: hierarchy of files and folders along with Notes directory on user local disk for storing images and note files

Now after downloading the eLOs and pressing the button 'Next' in the hylOs courses window, another window appears which provide the user to enable transferring of the downloaded content to iPOD.

When the window appears, user is supposed to connect the iPOD and press the 'submit' button. If the user press the 'submit' button, without connecting the iPOD or connecting some other device, then a status messages will be displayed to inform the user that iPOD is not connected and hence it do not enables the button 'synchronization to iPOD'. If the user connects the iPOD properly, then the status message informs the user that the synchronization can be done. The content transfer window with status messages displaying for both the situations is shown in the figures 7.5 and 7.6

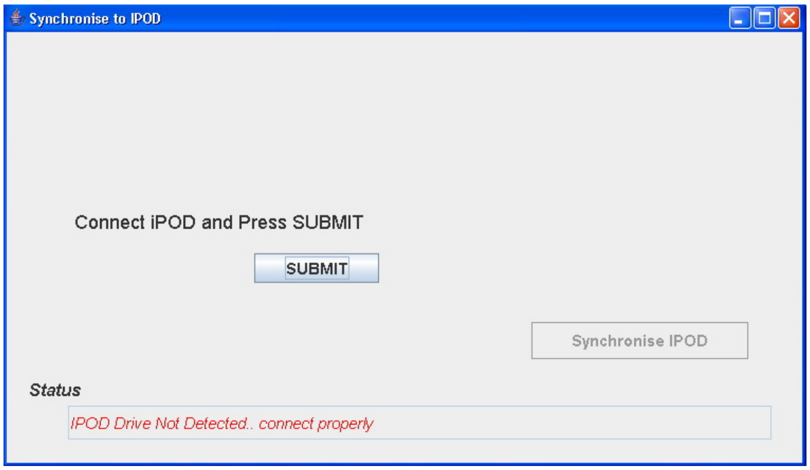


Figure 7.5: Content transfer window : when the iPOD is not detected

Figure 7.6: Content transfer window : when the iPOD is detected

When the iPOD drive is detected properly, the 'Synchronization to iPOD' button becomes active. When the user selects the synchronization, then the note files and images dowloaded in the Notes folder on local directory are transfered to the iPOD Notes section. Simulataneously, if there are any media files downloaded to the DLoad directory then the button press of 'synchronization to iPOD' launches the iTunes application and import the media files to iTunes library. Finally the imported files are automatically synchronized to iPOD and iTunes displays a message 'iPOD update is complete', as soon as it finishes transferring of media files to iPOD. Figure7.7 shows the audio files imported to iTunes Library and figure 7.8 shows the video files imported to the iTunes library, along with the message shown by iTunes after synchronizing the files to iPOD.

Figure 7.7: iTunes Library: audio files of hylOs eLOs in music section

Figure 7.8: iTunes Library: a sample video file of hylOs eLO in movie section

7.2 Results: Interaction and Navigation in iPOD

The courses synchronized to iPOD are displayed as menu items inside the Notes menu. It is as shown in the figure.7.9

Figure 7.9: Custom menus for hylOs courses in iPOD

7.2.1 Linear Navigation

All the eLOs that are present in the tree structure of a course are presented in iPOD as a sequence of eLOs, which are accessible in a linear fashion by pressing the hyperlink 'Next.Lesson' as shown in the figure7.10

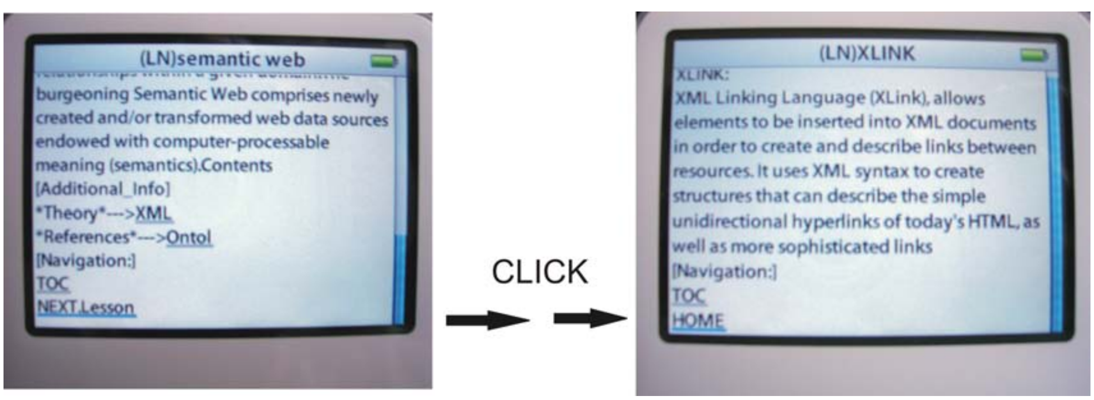


Figure 7.10: Linear navigation in iPOD

When the last eLO object of the course is reached, a hyperlink with the name 'HOME' will be displayed and pressing this link will guide the user to the root eLO of that course. And the 'TOC' hyperlink in the linear navigation will guide the user to the index of courses that are synchronized after downloading the courses from hylOs repository.

7.2.2 Semantic Navigation

At anytime, while accessing the eLO in linear navigation, if the additional information is displayed in the content page of the eLO with hyperlinks to the corresponding related topics, then user can click on the links and navigate the semantically related topics provided. The corresponding semantic navigation in iPOD is as shown in the figure 7.11

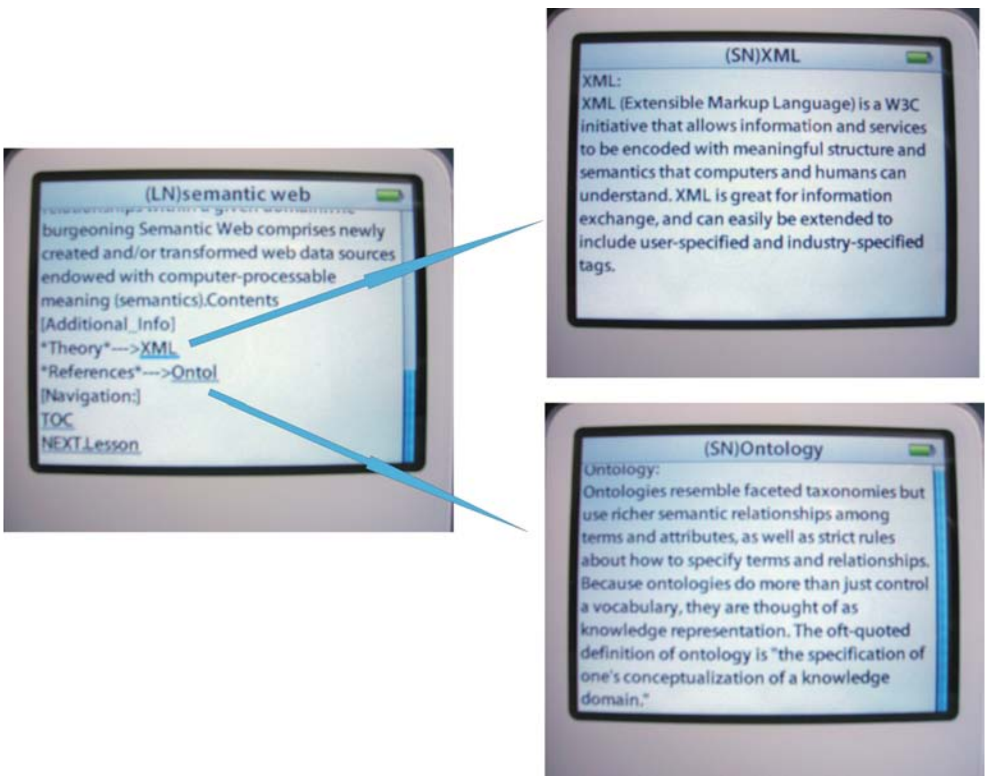


Figure 7.11: Semantic navigation in iPOD

7.2.3 Multimedia access

The images and media files transferred to the iPOD along with the readable text content are accessed by means of hyperlinks as shown in the figure 7.12 and 7.13.

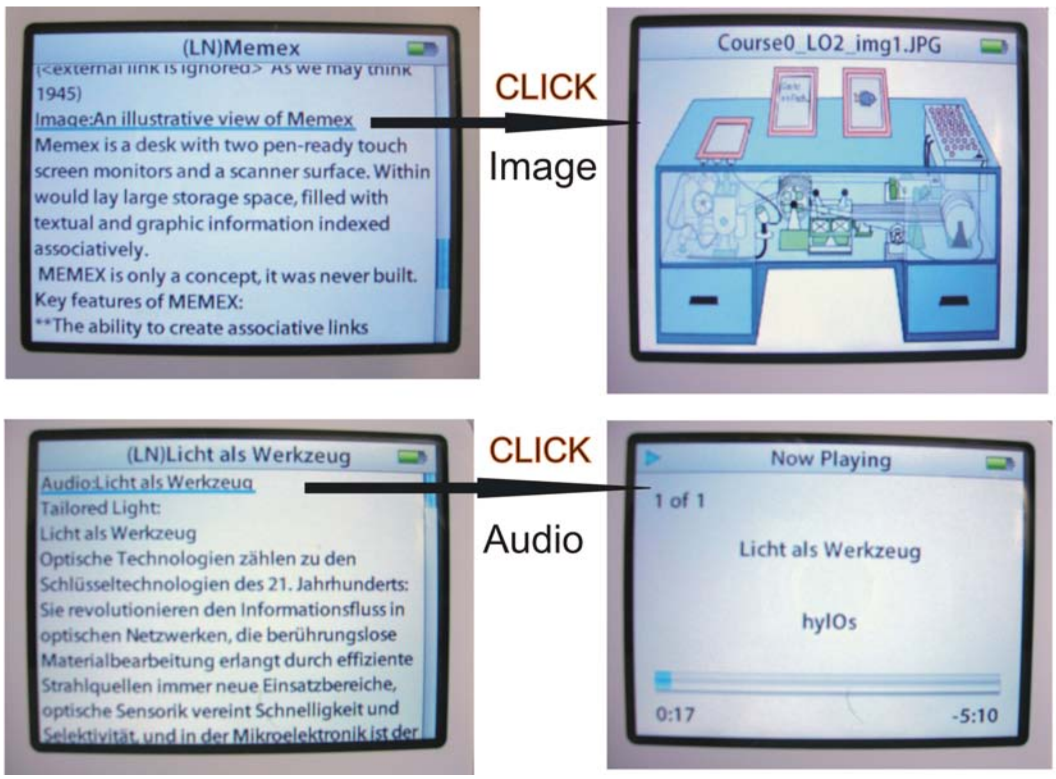


Figure 7.12: eLO content page on iPOD display: accessing Image and Audio files

Figure 7.13: eLO content page on iPOD display: accessing Video files

7.3 Evaluation of results:

The following evaluation is done on the results obtained :

- Linear Navigation : The navigation of eLOs in a linear way was obtained as expected
- Semantic Navigation : The navigation provided to access the semantically related topics was also obtained as expected.
- Multimedia Access : All the images and media files present in the hylOs eLOs were accessible from iPOD without any problem.
- Table transformation: The transformation of table content to plain text format didnot produced the results as we expected. The content was not properly displayed on iPOD display. The main reason for it is, iPOD takes the characters as variable-width, and donot support fixed-width for characters. Hence, the number of allowable characters per line varies, depending on the character width present in the content. In our implementation we have tried to provide reasonable presentation of table content , based on the approach as explained in 5.1.3, but we donot succeeded.
- Content feed: The downloaded content on the local disk was tranferred successfully to iPOD, provided the iPOD is already synchronized to the iTunes in use. An exception occurs while transferring the

media files, if the iPOD is not previously synchronized to that iTunes and it was not solved in this thesis. It is left open for future work.

7.4 User Feedback

In this thesis, we provided learning content with the related topics that are extracted from a set of eLOs in the semantic net. As this learning content is used for offline learning on iPOD, the user donot get any external information other than the content fed to the device. Therefore it is certainly necessary to see the response of people to know, how effective is our solution in providing most significant related eLOs, navigation and multimedia access on iPOD. Hence, a user feedback is taken from different students and provided in this section, which will be useful for further development of our application in future.

User: Madan Gopal

Location :TUHH, Hamburg, Germany

Feedback:

It is really nice to access the learning content along with the images, audio and video altogether on iPOD. Particularly, the availability of related topics along with the actual content is impressive. It would have been better if a feedback channel is also provided, atleast on the Proxy

User: Rajat Singh

Location :HAW-Hamburg, Germany

Feedback:

Navigation provided for the learning content is nice. Accessing additional information for the content is really helpful for self learning and multimedia content available along with the learning content was also handful while learning.

User: Vijay Raghavan

Location:University of Lueneburg, Germany

Feedback:

Accessing images and media files with learning content is nice. I am really impressed with the navigation provided for accessing the related topics and the actual learning content. Its really useful to have some additional information along with the original content which avoids the necessary to refer outside resources.

User: Bala Ranganadh

Location:TUHH, Hamburg

Feedback:

Providing learning content on iPOD is a good move. Navigation is effective. Regarding the additional information provided with the learning content, it would have been much useful, if atleast two or three topics are provided for each category.

User : Raju

Location:TUHH, Hamburg

Feedback:

Accessing learning content with images and audio files is nice. Navigation is good, but its better if a list of all the lessons for a particular course is provided at the begining of the course. Anyway its nice to see some additional information for the lessons which becomes useful for getting detailed information about that lesson.

Chapter 8

Conclusion

In this project a Software application was designed and implemented as a proxy between the hylOs and a desktop PC system. The sytem retrieves eLearning Objects that are semantically connected with qualified inter-object relations, stored in the online repository of hypermedia learning Objects system (hylOs). While downloading the eLOs, the content and structure of the eLOs are dynamically adapted to the iPOD device and stored in customized Note files on the device. During this process the following tasks are accomplished.

1. The relational complexity of the semantic nets of eLOs is reduced by the proposed reduction algorithm and as a result of this each eLO is provided with a set of related topics grouped under associate categories to achieve immediate interobject navigation.

2. All the files and folders which hold the eLO content are arranged in the Notes file hierarchy supported by the iPOD Notes section. Note files of eLOs are customized and a smart navigation is provide to the user, to achieve a self-exploration of the learning content. The iPOD interface is customized by creating the custom menus to display the courses, that users navigate to view the underlying eLOs present in that course.

3. Prior to transferring the readable content from hylOs to Notes on the iPOD, the content is transformed to the iPOD compatible markup and fragmented by following the paragraph-wise content segmentation of hylOs.

4. Images present in eLOs in different formats are converted to JPG and stored in iPOD Notes section.

5. Audio and video files present in eLOs content are retrieved and synchronized to iPOD using the iTunes COM API.

6. To access the multimedia content along with the Note files on iPOD, the required hyperlinks are embedded in the respective note files before transfering to iPOD.

Further issues and the areas worth investigating are give below:

1. A subscription feature on the proxy to subscribe to the courses available on hylOs to keep track of courses.

2. Automatic updates for the subscribed courses using the time stamps provided in the XML documents of each eLO.

    - Implementing a logging facility for the proxy to enable error recovery and consistent update operations between hylOs and mobile device

3. Provide feedback channel for rating purpose or commenting on the topics.

4. Improving the proxy features for adapting the eLOs to other mobile devices.

Bibliography

[1] Dieter Averberg, Julien Boppert, and Kai Holzweißig. Mobile delivery server (mds) a solution for resolving problems and limitations in mobile e-learning scenarios. In *ICN/ICONS/MCL*, page 212, 2006.

[2] Stefan Bergstedt, Stefan Wiegreffe, Jochen Wittmann, and Dietmar Moller. Content management systems and e-learning-systems a symbiosis? *icalt*, 00:155, 2003. doi: http://doi.ieeecomputersociety.org/10.1109/ICALT.2003.1215047.

[3] Rodrigo A. Botafogo, Ehud Rivlin, and Ben Shneiderman. Structural analysis of hypertexts: identifying hierarchies and useful metrics. *ACM Trans. Inf. Syst.*, 10(2):142–180, April 1992. ISSN 1046-8188. doi: 10.1145/146802.146826. URL `http://portal.acm.org/citation.cfm?id=146826`.

[4] Julien Broisin, Philippe Vidal, Michael Meire, and Erik Duval. Bridging the gap between learning management systems and learning object repositories: Exploiting learning context information. In *AICT-SAPIR-ELETE '05: Proceedings of the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop (AICT/SAPIR/ELETE'05)*, pages 478–483, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2388-9. doi: http://dx.doi.org/10.1109/AICT.2005.33.

[5] Yang Cao, Tony Tin, Rory McGreal, Mohamed Ally, and Sherry Coffey. The athabasca university mobile library project: increasing the boundaries of anytime and anywhere learning for students. In *IWCMC '06: Proceeding of the 2006 international conference on Communications and mobile computing*, pages 1289–1294, New York, NY, USA, 2006. ACM

Press. ISBN 1-59593-306-9. doi: http://doi.acm.org/10.1145/1143549.1143808.

[6] James Clark. XSL transformations (XSLT) version 1.0. W3C recommendation, W3C, November 1999. http://www.w3.org/TR/1999/REC-xslt-19991116.

[7] Laercio Sambuc dos Anjos and Jose M. Parente de Oliveira. A navigation interface for adaptive m-learning applications. In *ICALT '06: Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies*, pages 61–62, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2632-2.

[8] Erik Duval, Tom Wason, and Wayne Hodgins. Ieee 1484.12.1-2002, learning object metadata standard. Technical report, July 2002.

[9] M. Engelhardt, A. Hildebrand, A. Karparti, T. Rack, and T.C. Schmidt. Educational Content Management – A Celluar Approach. In Michael E. Auer and Ursula Auer, editors, *Interactive computer aided learning (ICL) 2002. International Workshop. Blended Learning*, Villach (Austria), September 2002. Kassel University Press.

[10] Michael Engelhardt and Thomas C. Schmidt. Semantic Linking – a Context-Based Approach to Interactivity in Hypermedia. In Robert Tolksdorf and Rainer Eckstein, editors, *Berliner XML Tage 2003 – Tagungsband*, pages 55–66, Humboldt Universität zu Berlin, September 2003.

[11] Michael Engelhardt, Arne Hildebrand, Alexander Lang, Thomas C. Schmidt, and Mathias Werlitz. A Constructivist Content Exploration based on a Hypermedia eLearning Object System. In Michael E. Auer and Ursula Auer, editors, *Proceedings of the International Conference Interactive Computer aided Learning ICL 2004. The Future of Learning*, Villach (Austria), October 2004. Kassel University Press.

[12] Michael Engelhardt, Arne Hildebrand, Dagmar Lange, and Thomas C. Schmidt. Reasoning about eLearning Multimedia Objects. In Jacco Van Ossenbruggen, Giorgos Stamou, Raphal Troncy, and Vassilis Tzouvaras, editors, *Proc. of WWW 2006, Intern. Workshop on Semantic Web Annotations for Multimedia (SWAMM)*, May 2006. URL `http://image.ntua.gr/swamm2006/resources/paper06.pdf`.

[13] Michael Feldstein. What's important in a learning content management system. *eLearn*, 2002(5):4, 2002. doi: http://doi.acm.org/10.1145/566786.566792.

[14] Thanasis Giouvanakis, Garyfallos Fragidis, Eyaggelos Kehris, and Haido Samaras. Exploiting concept mapping in a semantic web environment. In *ICALT '05: Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies*, pages 432–434, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2338-2. doi: http://dx.doi.org/10.1109/ICALT.2005.146.

[15] T. Bickmore A. Girgensohn and J.W. Sullivan. "web page filtering and reauthoring for mobile users," computer journ., vol. 42, no. 6, 1999, pp. 534-546.

[16] Andreas Holzinger, Alexander Nischelwitzer, and Matthias Meisenberger. Lifelong-learning support by m-learning: example scenarios. *eLearn*, 2005(11):2, 2005. doi: http://doi.acm.org/10.1145/1125280.1125284.

[17] Andreas Holzinger, Alexander Nischelwitzer, and Matthias Meisenberger. Mobile phones as a challenge for m-learning: Examples for mobile interactive learning objects (milos). In *PERCOMW '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 307–311, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2300-5. doi: http://dx.doi.org/10.1109/PERCOMW.2005.59.

[18] Philipp Hoschka. Synchronized multimedia integration language (smil) 1.0 specification w3crecommendation, w3c, june 1998, w3crecommendation, http://www.w3.org/tr/1998/rec-smil-19980615/. Technical report.

[19] Bernd J. Krämer. Mobile learning: The next generation of learning. research report 5, fernuniversität in hagen, faculty of mathematics and computer science, hagen, germany, may2005. fernuniversitäts contributions to the 2nd year of the leonardo project mlearn2.

[20] Tayeb Lemlouma and Nabil Laya. Content interaction and formatting for mobile devices. In *DocEng '05: Proceedings of the 2005 ACM symposium on Document engineering*, pages 98–100, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-240-2. doi: http://doi.acm.org/10.1145/1096601.1096627.

[21] Chi-Hong Leung and Yuen-Yan Chan. Mobile learning: A new paradigm in electronic learning. *icalt*, 00:76, 2003. doi: http://doi.ieeecomputersociety.org/10.1109/ICALT.2003.1215030.

[22] Moira J. McAlister and Peng Hui Xie. Using a pda for mobile learning. In *WMTE '05: Proceedings of the IEEE International Workshop on Wireless and Mobile Technologies in Education*, pages 282–284, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2385-4. doi: http://dx.doi.org/10.1109/WMTE.2005.70.

[23] Charles McCathieNevile and Jo Rabin. Mobile web best practices 1.0. W3C proposed reccommendation, W3C, November 2006. http://www.w3.org/TR/2006/PR-mobile-bp-20061102/.

[24] Boaz Mermelstein and Edna Tal. Using cellular phones in higher education: Mobile access to online course materials. *wmte*, 0:285–287, 2005. doi: http://doi.ieeecomputersociety.org/10.1109/WMTE.2005.71.

[25] Hidetaka Ohto, Luu Tran, Mark H. Butler, Graham Klyne, Franklin Reynolds, Johan Hjelm, and Chris Woodrow. Composite capability/preference profiles (CC/pp): Structure and vocabularies 1.0. W3C recommendation, W3C, January 2004. http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/.

[26] Sean Owen, Phil Archer, and Jo Rabin. W3C mobileok scheme 1.0. W3C working draft, W3C, July 2006. http://www.w3.org/TR/2006/WD-mobileOK-20060712/.

[27] James Pearce, Argogroup, Matt Womer, and France Telecom. W3c device description landscape, w3c working draft, februaury 2006, url http://www.w3.org/tr/dd-landscape/. Technical report.

[28] P. R. Polsani. Use and abuse of reusable learning objects. *Journal of Digital Information*, 3(4), February 2003.

[29] ADL Technical Team. Sharable content object reference model (scorm), specification v 1.2, advanced distributed learning (adl). Technical report.