A MULTIDISCIPLINARY APPROACH TO

THE REUSE OF

OPEN LEARNING RESOURCES

A thesis submitted in fulfilment of the requirements for the degree of

Master of Engineering (Research)

by

Sergio FRESCHI

School of Electrical and Information Engineering

The University of Sydney

March 2008
Abstract

Educational standards are having a significant impact on e-Learning. They allow for better exchange of information among different organizations and institutions. They simplify reusing and repurposing learning materials. They give teachers the possibility of personalizing them according to the student’s background and learning speed. Thanks to these standards, off-the-shelf content can be adapted to a particular student cohort’s context and learning needs. The same course content can be presented in different languages. Overall, all the parties involved in the learning-teaching process (students, teachers and institutions) can benefit from these standards and so online education can be improved.

To materialize the benefits of standards, learning resources should be structured according to these standards. Unfortunately, there is the problem that a large number of existing e-Learning materials lack the intrinsic logical structure required, and further, when they have the structure, they are not encoded as required. These problems make it virtually impossible to share these materials.

This thesis addresses the following research question:

*How to make the best use of existing open learning resources available on the Internet by taking advantage of educational standards and specifications and thus improving content reusability?*

In order to answer this question, I combine different technologies, techniques and standards that make the sharing of publicly available learning resources possible in innovative ways. I developed and implemented a three-stage tool to tackle the above problem. By applying information extraction techniques and open e-Learning standards to legacy learning resources the tool has proven to improve content reusability. In so doing, it contributes to the understanding of how these technologies can be used in real scenarios and shows how online education can benefit from them.

In particular, three main components were created which enable the conversion process from unstructured educational content into a standard compliant form in a systematic and automatic way.
An increasing number of repositories with educational resources are available, including Wikiversity and the Massachusetts Institute of Technology OpenCourseware. Wikiversity is an open repository containing over 6,000 learning resources in several disciplines and for all age groups [1]. I used the OpenCourseWare repository to evaluate the effectiveness of my software components and ideas. The results show that it is possible to create standard compliant learning objects from the publicly available web pages, improving their searchability, interoperability and reusability.
Acknowledgments

First of all, I would like to thank my supervisor, Dr. Rafael A. Calvo, for his supervision of my thesis, and for giving me the opportunity to pursue this project.

I also want to extend my thanks to Dr. David Levy for his role of co-supervisor of my project. Thanks go to the rest of the Web Engineering Group — in particular Danyu Zhang and Aiman Turani for their constant and selfless support and patience.

On a personal level, I dedicate the effort put into this work to my parents, Maria Rosa and Alberto, my brothers and to my two beloved nephews: Valentin and Thomas. I want also to mention my dear friends, who have always supported me.
Preface

This thesis is the culmination of a Masters project in the Web Engineering Group at the School of Electrical and Information Engineering, University of Sydney.

The project has produced the following outcomes: the first one is this thesis and one publication, and the other one is the application software OCWise, a tool capable of automatically reshaping learning materials available on the Internet.

In order to produce such a tool, research into current Information Extraction techniques has been necessary, as well as research into e-Learning standards for allowing content reusability and interoperability among platforms.

Contributions

During the course of the candidature on which this thesis is based, the following contributions were accomplished:

• The OCWise tool was designed, implemented, tested and released under an open-source license. The release includes documentation and a simple example application using this tool.

• Novel information extraction techniques were implemented, as well as an algorithm which allows users to download learning resources from different public/open repositories and in so doing makes them available as part of the current Learning Management System.

• A paper on the design and applicability of the OCWise tool was published in the proceedings of the 2006 23rd Annual ASCILITE Conference, Sydney [2].
Motivation

As a secondary and university teacher I developed course content for my several subjects. Most subjects shared common topics and they were taught in different schools and universities. One of the problems I encountered was related to exporting and importing courses between different platforms. Since platforms were incompatible between each other - I had to build my courses again from scratch. To make things worse, it was impossible to share and reuse previous content in new courses. This was a time consuming and tedious task. At the same time, I became aware of the latest advances in specifications and standards in the online environment. So, I decided to embark on this project, which I hope will be an important contribution towards improving reusability and interoperability in distance education.

Availability

After submission to the University of Sydney, this thesis document will be available in electronic format at http://weg.ee.usyd.edu.au/people/sergio/Thesis.pdf, and in hardcopy format from the University of Sydney Engineering Library.

Licensing

The OCWise tool is implemented as a set of Perl modules. As is customary with many Perl modules, the tool is distributed under the same licensing terms as the standard version of the Perl interpreter. This means that the user may choose either the GNU General Public License or the Artistic License as the terms of using the software, whichever fits better with their needs. In practical terms, this means that the code is encouraged to be used in research, commercial, educational, or other environments, without the need to pay royalties to the original author of the software. It also means that the software’s inner workings are available to be inspected or modified by other developers for their own projects.

Licenses of the above type are called open source licenses. Their goal is to foster the development and evolution of software by leveraging the user and developer communities as a resource that can feed back into the development cycle.
According to the Open Source Initiative [3], “open source promotes software reliability and quality by supporting independent peer review and rapid evolution of source code.” This aligns very well with the traditional goals of academic research. By making the source code discussed in academic publications available as open source resources, the results can be more easily verified by other researchers.

For more information on open source concepts, please visit http://www.opensource.org/.

This thesis is copyright © 2008 by Sergio FRESCHI. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, v1.0 or later (the latest version is presently available at http://www.opencontent.org/openpub/).
Contents

ABSTRACT ............................................................................................................................................... II
ACKNOWLEDGMENTS .......................................................................................................................... IV
PREFACE .................................................................................................................................................. V
  CONTRIBUTIONS ................................................................................................................................. V
  MOTIVATION ......................................................................................................................................... VI
  AVAILABILITY ....................................................................................................................................... VI
  LICENSING .......................................................................................................................................... VI
CONTENTS .............................................................................................................................................. VIII
LIST OF FIGURES .................................................................................................................................. X
LIST OF TABLES ....................................................................................................................................... XI
1 INTRODUCTION .................................................................................................................................. 1
2 BACKGROUND ...................................................................................................................................... 6
  2.1 OPENCourseWare DEFINITION .............................................................................................. 7
  2.1.1 Analysis of courseware structure and format ....................................................................... 10
  2.2 OPEN E-LEARNING STANDARDS ......................................................................................... 12
  2.2.1 IMS Content Packaging ................................................................................................. 17
  2.2.2 IEEE Learning Object Metadata ................................................................................... 19
  2.2.3 IMS Learning Design ...................................................................................................... 20
  2.3 LEARNING OBJECTS ................................................................................................................ 22
  2.3.1 Challenges ......................................................................................................................... 23
  2.4 LEARNING OBJECTS REPOSITORIES ............................................................................... 24
  2.5 INFORMATION EXTRACTION TECHNIQUES .................................................................... 27
  2.5.1 Challenges in e-Learning ............................................................................................... 30
  2.5.2 Survey of Web IE Systems .............................................................................................. 33
  2.6 EVALUATION MEASURES .................................................................................................... 35
  2.6.1 Definition of common performance measures ................................................................ 36
  2.7 IMPLEMENTATION TECHNOLOGY ..................................................................................... 39
3 OCWIDE DESIGN ............................................................................................................................... 46
  3.1 DESIGN DESCRIPTION ........................................................................................................... 47
  3.2 RETRIEVING EDUCATIONAL MATERIALS FROM THE WEB ........................................ 49
  3.3 EXTRACTION INFORMATION .............................................................................................. 51
  3.4 MAPPING INFORMATION ....................................................................................................... 56
  3.5 WRAPPING-UP PROCESS......................................................................................................... 56
4 SYSTEM IMPLEMENTATION ............................................................................................................. 58
  4.1 IMPLEMENTATION LANGUAGE ............................................................................................ 58
  4.2 OCWIDE INFORMATION FLOW ......................................................................................... 59
  4.3 OCWIDE DEPLOYMENT ........................................................................................................ 60
  4.4 IMPLEMENTATION APPROACH ............................................................................................. 61
  4.5 E-LEARNING STANDARDS IMPLEMENTATION ............................................................... 65
    4.5.1 IEEE Learning Object Metadata .................................................................................... 65
    4.5.2 IMS Content Packaging ................................................................................................ 68
5 SYSTEM EVALUATION ....................................................................................................................... 70
  5.1 DATA SET ................................................................................................................................... 70
    5.1.1 MIT OpenCourseWare .................................................................................................... 70
  5.2 PRELIMINARY TESTING AND EVALUATION ...................................................................... 71
  5.3 QUALITY OF EXTRACTION ................................................................................................... 74
    5.3.1 Efficiency ......................................................................................................................... 74
List of Figures

Figure 1. MIT OpenCourseWare Object Model [30]......................................................... 10
Figure 2. IMS Standards suite.......................................................................................... 13
Figure 3. IMS Content Packaging.................................................................................... 18
Figure 4. Learning Object Model (LOM) ..................................................................... 20
Figure 5. Learning Design UML class diagram.............................................................. 21
Figure 6. IMS Learning Design generation approach..................................................... 22
Figure 7. Learning objects benefits............................................................................... 23
Figure 8. General design of IE systems ....................................................................... 28
Figure 9. Information Extraction Systems Taxonomy..................................................... 29
Figure 10. Evaluation Measures.................................................................................... 36
Figure 11. OCWise System Overview ......................................................................... 46
Figure 12. Retrieval rules............................................................................................. 50
Figure 13. Courseware layout ....................................................................................... 51
Figure 14. Data Extraction ........................................................................................... 52
Figure 15. OCW course home page ............................................................................. 53
Figure 16. Extraction rules – Home page ..................................................................... 54
Figure 17. Extraction rules – Calendar page ................................................................. 54
Figure 18. Extraction rules – Syllabus page ................................................................... 55
Figure 19. Data mappings ............................................................................................ 56
Figure 20. Wrapping process ....................................................................................... 57
Figure 21. OCWise Information Flow .......................................................................... 59
Figure 22. OCWise Deployment Diagram.................................................................... 60
Figure 23. List of processes and functions..................................................................... 62
Figure 24. Process of importing ‘reshaped’ OCW courses............................................. 72
List of Tables

Table 1. List of Learning Objects Repository Systems ............................................. 24
Table 2. Learning Object Repositories Comparison.................................................. 27
Table 3. Confusion matrix........................................................................................ 35
Table 4. Learning Object Metadata categories .......................................................... 66
Table 5. Evaluation results......................................................................................... 73
Table 6. Extraction results for course details ............................................................. 75
1 Introduction

The transformational effect of the Internet is changing education, just as it has changed most other aspects of today’s businesses and social lives. The rapid advance of e-Learning technologies affords new learning experiences including networked learning [4], automatically generated support [5] and many more.

New opportunities come with new challenges, and integrating existing courses into platforms that support these new functionalities is one of the biggest challenges that institutions have. The activities of most existing courses would need to be redesigned to benefit from the opportunities of the online world [6]. The production of new content for distance education is both a time consuming and expensive task. Therefore, it would be very advantageous to be able to reuse legacy learning resources available on the Internet.

A number of international standards describe conceptual structures and XML based schemas that allow learning materials to be portable from one platform to another [7-10]. The learning platforms then support collaborative activities, assessment and much more.

Interestingly, a number of high profile Universities are investing in showcasing and distributing their face to face courses in online form. The Massachusetts Institute of Technology’s (MIT) OpenCourseWare (OCW) [11] was the first, followed by a number more [12]. Regrettably, many of these high quality materials available in the public domain do not have the correct structure or formatting to be used by these standard compliant systems. What is more, they cannot be fully integrated with any Learning Management System (LMS). In addition, as the number of learning resources increases, it is becoming extremely difficult for learners and course authors to find the required and relevant learning resources using conventional search engines. For example, a search engine like Google would not be able to distinguish between a first year Mechanics course and a second year one as it is not able to use the metadata information that describes this attribute.

Static content is of limited value in the overall learning experience. Students learn while they collaborate and participate in activities [13]. Educational researchers
have studied how learning technologies can be used effectively, particularly in higher education [14].

I believe that having a one-stop homogeneous repository of high-quality, multi disciplinary, vast and free learning resources, such as MIT’s OpenCourseWare can benefit a large number of people [15]. Further, being able to manage these materials as standard compliant learning objects will improve their impact.

A manual approach to annotate learning materials is a slow and costly process and therefore not a viable task. Statistical Natural Language Processing techniques have been used to extract information from unstructured documents parsing it into structured format [16].

In this thesis, I develop OCWise, an automated three layered tool, which takes learning materials from public web pages and creates learning objects that enable course authors and educational institutions to share, reuse and integrate open content in Learning Management Systems.

Open content refers to the generation and free distribution of knowledge to the general public on the Internet. For instance, one of the largest open content projects is Wikipedia, an online encyclopedia [17].

Courseware refers to open content in the form of raw educational materials. There are many open courseware projects around the world, mainly run by universities. Among the most visible projects are the MIT OCW [11] and Rice’s Connexions [18]. A full list of OCW projects can be found on the OCW Consortium website [12]. These projects usually adopt the Creative Commons License [19], which allows users (e.g. teachers, learners) to search, use, reproduce and modify content without any restrictions.

This thesis is set out to investigate how to make the most of existing learning resources available on the Web by studying the application of information extraction techniques and the use of the latest open educational standards. It has been realized that open education standards can be used to share content in an efficient and systematic way. They allow course authors and universities great flexibility and platform independence since they facilitate content interoperability, reusability and repurpose.

Two types of benefits drive this research. First, users of OCW and similar websites (e.g. learners, teachers and universities) can benefit from standardization by reusing the content by being able to:
a) Move content from one university to another one very easily. They just export content and import it into a different repository. This can reduce the risk of vendor lock-in.

b) Reuse content which was created previously for a particular course in a particular discipline into a different course and/or discipline. For instance, a physics teacher can explain the concept of entropy and at the same time a Computer Science teacher can use the existing material for his/her own classes.

c) Easily locate learning resources by having the possibility to issue complex search queries. A possible example can be: ‘show me all postgraduate Spanish courses given by Prof. John Sarna’

d) Easily update learning content.

Second, by having standard compliant representations I hope others will explore new ways in which this content can be used. Particularly I intend this content to become part of activities. For example:

a) The course websites describe collaboration tasks where students are asked to engage in discussions about some content. The LMS affords new mechanisms for synchronous discussions and other forms of scripted collaboration [4] that could replace the type of activities that were expected in a face to face environment. This is the type of research carried out by the Learning Design research community [20].

b) Integrate customized versions of the content into activities according to learner’s background, learning speed, language, and so forth, or for example, if a student’s profile shows a lack of knowledge on a particular topic, readings that refer to it would be mandatory instead of ‘optional’. This is the type of research carried out by the Intelligent Tutoring Systems community [21, 22].

Further to these benefits, having content in a structured format (such as XML format) has put universities in a competitive position because they can improve and enhance the teaching and learning process in an online environment.

Complementing the use of learning standards, this project hopes to devise and apply new extraction rules to assist in the process of extraction of semantic data. To
be successful and effective, this multidisciplinary approach must be able to tackle the lack of structure and the heterogeneity of content.

The chosen methodology to tackle the problem of structuring open content involves the combination of different technologies, which makes it innovative. On one hand, I applied information extraction techniques to the education field that have been mostly used in other areas, such as the stock market, weather forecasting and auction sites [23]. I have devised and produced a set of Extraction Rules for extracting semantic information from educational materials. On the other hand, the latest open e-Learning standards, in particular IMS Content Packaging [7], IEEE Learning Object Metadata [10] and IMS Learning Design [8] were studied and analysed in detail, implemented and then tested in real scenarios to show how distance education can benefit from them. This multidisciplinary approach can lead us to the solution of our research questions in a systematic way based on the great potential these technologies offer. Their world wide adoption, continuous improvement and evolving features make them the best choice to tackle our research problems. However, there are a couple of challenges in this approach.

One challenge lies in the fact that metadata should be extracted in a systematic and automatic way and, in this specific project, from different sections within the same course. In addition to that, the format of metadata is diverse (e.g. HTML tags, tables, paragraphs) and sometimes it is non-existent. Another challenge is how to break down course resources into less-coupled units and thus generate Learning Objects, which facilitate reusability and repurpose of learning content.

Other approaches were found in the research literature [24], which will be described in detail in the next chapter.

The significance of this approach is that it facilitates the exchange of knowledge in a better way and makes the best use of these valuable and high quality educational materials by:

- Enhancing reusability and interoperability of digital learning resources among different LMS. For instance, course authors can move all their courses from one institution to another one in a very easy and efficient way. This can be a significant time saving for teachers.
- Building a homogeneous metadata-rich repository with courses from different learning institutions around the world, allowing for a larger
Chapter 1: Introduction

community which can collaborate by revising and updating learning objects.

• Increasing collaboration within and between institutions.

This thesis is divided in six chapters. Chapter 2 presents the background research related to open courseware. In addition to this, it gives a brief description of open educational standards, learning objects, learning objects repositories and information extraction techniques.

Chapter 3 outlines the design of my system architecture and the chosen methodology for my tool design, including different technologies.

Chapter 4 describes the implementation approach and explains technical details in order to share learning objects in different LMS. It also gives ideas about how to get the most of these resources in order to reuse and repurpose them.

In Chapter 5 I present the system evaluation criteria and test my design based on different learning management systems and provide some performance measures.

In Chapter 6, I present my conclusions.
2 Background

In this chapter I present the research literature on the current theories, developments, and standards in regards to reusing open content and its impact on university education. It provides the theoretical background on which the OCWise (Open CourseWare information & structure extractor) tool is based. Section 2.1 introduces the concept of OpenCourseWare and describes most of its characteristics as well as some limitations which triggered some of my research questions. I also describe a case study, which is used a reference model and as an input data set for my system development and evaluation.

Section 2.2 introduces the literature on existing Open e-Learning Standards. Discussed are their benefits, and how they can describe and represent the structure and sequence of course materials in a formal and systematic way, thus enabling LMS and repositories to manage course structure and metadata in an efficient way.

Section 2.3 provides a brief description of the Learning Objects concept, its characteristics, benefits, and importance.

Section 2.4 describes the ‘container’ where learning objects are stored, which is called: Learning Objects Repositories. It also presents their functionality and a list of the most well-known repository systems is shown as well.

Section 2.5 introduces the literature on Information Extraction techniques and how they can be used to improve the reusability of learning materials.

Finally, Section 2.6 provides definitions of the most common performance measures of information extraction techniques that were found in the research literature and are used to evaluate my system’s performance.
2.1 OpenCourseWare definition

According to Baldi et al. [25] courseware consists of “all supporting digital materials for academic purposes, such as presentations slides, case studies and software for educational use.” This author has also highlighted the importance of courseware to university education by describing it as a model for university dissemination of knowledge in the Internet age. “Open courseware is teaching knowledge both in content and structure. It can be of interest to persons willing to learn about the course topic as well as to teachers who want to see how others are teaching specific subjects or want to integrate it into their own teaching material.”

Another definition by Yue et al. [26] states that open courseware is “the open and free publication of course materials, accessible usually through the web.” Attwell [27] emphasizes the real importance and significance of Open Content to the e-Learning community with these points:

- The idea that open content and of the sharing of content promises a potential solution to the biggest issue in e-Learning—the lack of affordable, high quality learning materials in a wide range of subject areas.
- Open content allows innovation. The availability of different learning applications and interfaces, linked to materials repositories can allow the educator to experiment with different pedagogic applications.

An important characteristic of open courseware is that the learning activities are passive, in the sense that students do not get to interact with teachers or other students. This is an important difference to distance learning initiatives. All these OCW projects make clear that they do not replace face-to-face learning. Students are not assessed, so these are not degree granting activities and students do not get formal credits. All course materials are free of charge and users (individuals and institutions) can modify and distribute the content as long as they adhere to its copyright license.

OCW repositories contain courses with more or less the same structure: a syllabus, which describes the course, its aims and expected outcomes; a calendar; and lecture notes and other materials. Few courses include multimedia resources
such as: audio and video files. These repositories are in many ways static, and since they cannot be easily integrated into systems where students can engage in discussions, or participate in activities their learning potential is limited.

Today, courseware initiatives provide the same curriculum structure and content to different learners despite individual differences such as knowledge background, learning style, learning speed, and so on. In the future, these materials would be better used in learner adapted environments.

**MIT OpenCourseWare case study**

One of the most important advances towards sharing educational content with the virtual learning community is the MIT OpenCourseWare project. This idea was very much welcomed and got excellent feedback by institutions, teachers and students. I chose the MIT OCW project as a case study for this project due to the number of courses it makes available, its relevance and importance, as well as its content presentation and organization. I use data from the MIT OCW project as input into my implementation.

In 2001 the Massachusetts Institute of Technology (MIT) decided to share its course materials and make them available on the Internet to the public. This initiative was called *OpenCourseWare* (OCW). MIT defines OCW as:

…MIT course materials that are used in the teaching of almost all undergraduate and graduate subjects available on the web, free of charge, to any user anywhere in the world.

…MIT OpenCourseWare will provide the content of, but is not a substitute for, an MIT education [11].

Its goals are:

- To provide free, searchable, coherent access to MIT’s course materials for educators in the non-profit sector, and students and individual learners around the world.
- To create an efficient, standards-based model which other universities may emulate to publish their own course materials.

MIT courses are being translated by Universia, a consortium of 100 universities in Spain and Portugal and in America, into Spanish and Portuguese, as well as into Mandarin by about 50 universities in mainland China [28]. This is an important
attribute since it will allow the same courses to be produced in different languages. This is a large benefit to teachers and students.

The MIT OCW project was the very first and since then, many other universities have followed this initiative with many more courses becoming available. This is also a really important characteristic since the Internet has been swamped with learning resources from different institutions worldwide and they keep ‘releasing’/adding new courses from time to time. This allows my system to be tested using different OCW projects from around the world and to evaluate my implementation using different input data.

To December 2005 MIT has published 1250 of its 1800 courses. All these courses have been made available under the Creative Commons License (CC). MIT OCW uses four of the eleven available Creative Common licenses—attribution, share alike, non-commercial and exceptions (Creative Commons website [19], MIT website [11]).

As described in [29], Creative Commons is a family of licenses that allows different authors to make different choices. The MIT particular license says you may copy the work and redistribute it, but you must preserve attribution, that is you must say who the original author was. You may not use it for commercial use, and if people make derivative works they have to distribute those derivative works under the same licensing terms as they original work. This characteristic makes this OCW project really attractive and provides the opportunity to reshape these valuable resources while maintaining the same copyright restrictions, which might end up being a benefit or advantage for other related research projects.
2.1.1 Analysis of courseware structure and format

There are several OpenCourseWare projects around the world (see [12] for a full list). For the reasons mentioned above as well as the fact that it provides the most courses, the way courses are divided, the quality of the resources provided as well as copyright restrictions, the MIT OCW was selected as the case study. It is important to note that many other OCW repositories follow the same model.

MIT OpenCourseWare object model

Figure 1 shows the object model for the MIT OCW website.

![Diagram of MIT OpenCourseWare Object Model]

Figure 1. MIT OpenCourseWare Object Model [30]

Each course is composed of many sections such as: Home page, Calendar, Readings, Syllabus, Assignments, Exams, and Projects among others. Some of these sections are always present (such as Home page, Syllabus and Calendar), while others are not (e.g. Exams and Projects). This is something one should take into account when it comes to developing a solution since it has to be able to recognize this difference in structure at runtime.
Chapter 2: Background

The Home page shows the course title, level (e.g. undergraduate or postgraduate), authors’ names, course description, course meeting times and so forth. This page also contains all the keywords given to this course by its author(s). It is important to mention that the keywords are not visible to the users and they are encoded in an html meta tag, which requires a different approach in order to extract the information.

The Syllabus contains, in most cases, learning objectives, grading, and outcomes, among others. Mostly, the format of this information is free text, but sometimes it can be found in a tabular format. This also requires a different approach when it comes to extracting this information.

The Calendar shows all the sessions or classes with their descriptions in a tabular format and sometimes it includes a resource file (such as a PDF file) per class.

All courses contain resource files in different formats such as: PDF, PPT, Word, etc.

**Current research on MIT OCW materials**

Some research is now being conducted on the impact of MIT courses such as the CWSpace project [30]. This project revolves around metadata standards and protocols in order to preserve MIT courses using the DSpace repository system. This project makes use of METS (Metadata Encoding and Transmission Standard) and IMS-CP (IMS Global Learning Consortium, Content Packaging) standards.

According to [30], an OCW course is a statically served, unchanging website, serving as the published record of a particular instance of a given course. No interactivity is found in the course publication, and no student authored content is part of the course materials.

Hannafin et al. [31] agree with MIT’s OCW team – ([30]) and states that most existing learning content cannot be scaled and reused in multiple contexts. It is typically static in nature, having been developed for a single specific teaching purpose.
2.2 Open e-Learning Standards

e-Learning standards are a set of common rules which describe how courses should be created and delivered over multiple different platforms in order to guarantee interoperability and reusability of learning materials. Those standards allow courseware to be moved from one platform to another and reuse materials in a completely new course.

Many organizations around the world have embarked on the challenging task of creating standards and specifications. Some of the main contributors to interoperability standards are:

- IMS Global
- IEEE
- ADL
- SCORM

The above organizations have produced several open interoperability specifications such as content packaging, metadata, learning design, among many others. In this project, I am particularly interested in: IMS Content Packaging, IMS Learning Design and IEEE Learning Object Metadata.

Different aspects of courseware development are taken into consideration by all those three standards. To start with, the Learning Object Metadata specification allows content metatagging, as a way of describing each element of a course. It ensures that learning content will be discovered by search tools (i.e. it ensures content discoverability/searchability). The Learning Design specification provides support for a wide range of instructional design such as Competency based learning (CBL) for example. It allows content authors to structure and sequence content in a pedagogically sound fashion. The Content packaging specification describes how content should be organized so as to produce a self-contained package, which can be transferred between multiple Learning Management Systems (i.e. it ensures data exchange). By having courseware that is compliant to these standards, content users save a great deal of time and money as content is easily located and exchanged between systems. The implementation of such standards requires expert developers and I have contributed towards making it easier for teachers, students and educational institutions. Since this project is very much related to interoperability of
content, it will focus on how to use specifications of information models to exchange data.

**IMS Global Learning Consortium**

Since my implementation is related to IMS content interoperability specifications, I present a description of IMS, an organization whose main aim is to create and develop specifications.

The IMS is an industry-sponsored organization that develops specifications for the learning industry. There are many specifications developed or under development and it is worthwhile visiting the IMS website ([http://www.imsproject.org](http://www.imsproject.org)) to keep current. The most important of these are shown in Figure 2.

![Figure 2. IMS Standards suite](image)
IMS Specifications

The IMS Global Consortium has developed and released over ten specifications. A brief description of each of them is presented next, followed by a detailed overview of the ones used in this project.

- **Content Packaging**
  The IMS Content Packaging Specification provides the means of describing and packaging learning resources such as individual courses or a collection of courses, a course unit, etc into a single self-contained package. OCW courses available on the web lack any structure. This specification could be used to give a new shape to them and thus facilitating the goals of interoperability and reusability.

- **Metadata**
  Metadata means *data about data*. It allows learning materials to be described and tagged so they can be easily discovered by search engines. One of the challenges in this project is to extract as much metadata as possible in an automatic way while metatagging content at the same time.

- **Learning Design**
  The IMS Learning Design specification is a language used to describe different units of study and/or complete courses. This schema allows course authors to specify who does what, when, which resources are available to each activity and what learning objectives have to be met by the learners. The importance of this specification is that it goes beyond the ‘static’ or ‘dead’ content by letting users play important roles during the teaching-learning process. It can also present different learning designs (e.g. Competency Based Learning, Problem Based Learning) to learners according to their learning background, learning speed, level, etc. We could use this schema to give ‘life’ to static content found in OCW courses and improve online education.
• **Simple Sequencing**
  The IMS Simple Sequencing Specification is used to sequence learning activities in a consistent way. The main difference between this schema and IMS LD schema is that Simple Sequencing schema is based on a single learner model, while IMS LD allows for multi-learner experiences.

• **Question and Test Interoperability**
  The IMS Question and Test Interoperability Specification enables course authors to describe questions and tests, which can be presented to learners as part of the required activities. Most OCW courses contain exam papers, projects and labs, normally in PDF format. They could be converted into a standard form using this schema. This would allow for reusability and repurpose.

• **Digital Repositories**
  The main purpose of the IMS Digital Repositories specification is to provide useful recommendations for the interoperability of the most common repository functions such as: import/export.

• **Competency Definitions**
  The Reusable Definition of Competency or Educational Objective (RDCEO) specification can be used to add and exchange competency information between learning management systems, human resources and many other systems. Content designers can use this schema to describe a set of competencies or skills as part of a career plan, pre-requisites or learning outcomes.
• **Learner Information**

The IMS Learner Information Package (IMS LIP) specification addresses the interoperability of Internet based Learner Information systems with other systems that support the Internet learning environment.

**IEEE Learning Technology Standards Committee**

The Institute of Electrical and Electronic Engineers (IEEE) is one of the most important global standards bodies and has taken an active interest in learning standards. The Learning Technology Standards Committee (LTSC) has a number of important standards [32].

The first standard to be formally adopted was for metadata. The IEEE LTSC LOM (Learning Object Metadata) was formally approved in 2002. Work is now underway on an XML binding. This is likely to largely follow the IMS XML binding currently used in SCORM.

**Shareable Content Object Reference Model**

One of the most popular learning standards, the Shareable Content Object Reference Model or SCORM is not actually a standard or even a specification. It is, as its name suggests, a reference model. Developed by the Advanced Distributed Learning Network (ADLNET), an organization initially funded by the US Government (the Department of Defence and the Department of Labour) to ensure that learning resources would be:

- Reusable
- Accessible
- Interoperable
- Durable

SCORM is now the de facto standard for many learning management systems.

In the next section, I provide a short description of the standards I have used and implemented in my project.
2.2.1 IMS Content Packaging

The Content Packaging (IMS-CP) specification [7] was developed by the IMS Global Learning Consortium. It describes how learning materials can be packaged so that they can be easily and consistently moved from one LMS to another. Content Packaging allows teachers and administrators to export a course from one learning content management system (such as WebCT, Sakai, .LRN) and import it into another. In theory, this should be an easy process as long as both systems follow the rules described in the specification. This type of scenario might be significant if an organization decides to change its LMS after a number of years and facilitates sharing of resources between organizations using different systems.

IMS CP schema allows for content interoperability by describing a set of data structures used to exchange content and information between different platforms and applications. One of the benefits of this standard is that content is platform independent. It allows course authors to reuse content in a very easy way. They can import and export content in different Learning Management Systems.

This specification treats content as an aggregation of digital components, allowing content to be re-grouped into different higher-level components. This allows, for instance, having more than one course or into the same package. This package consists of two main elements: a manifest and resource files. A manifest is an XML document which describes how the items are organized and inter-related within the package. Resource files include web pages, text files, evaluation objects and/or any other type of learning material. When both elements (manifest and resource files) are encapsulated together in a single archive file (eg, zip, cab files), the resulting file is called Package Interchange File (PIF). This PIF file can then be uploaded into any Learning Management System.
Figure 3 provides an overview of the IMS CP specification.

Figure 3 graphically depicts an IMS CP Manifest file (imsmanifest.xml), which is made up of different sections. Each section has a particular purpose, thus allowing the mapping of course structure in a very efficient and structured way.

All course resources (e.g. PowerPoint slides, audio and video files, PDF files) are included in the Resources section and they are referenced in other sections.

Another important section is the Metadata section. This standard allows to structure more than one course in the same package and is very flexible.
2.2.2 IEEE Learning Object Metadata

Learning Object Metadata (LOM) is an XML/RDF data model designed by the IEEE to describe learning objects and any other digital resources. This standard facilitates the discoverability, reusability and interoperability of learning materials mainly used by online Learning Management Systems (LMS).

The LOM data model provides a uniform way for tagging learning resources. One of the problems tackled by this thesis involves the extraction of metadata from open learning resources in an automatic way, and then generating an IEEE LOM file as output. I have found this standard very suitable for this project, since it presents a set of elements, which are hierarchically organized, and include all the relevant resource attributes such as course author, title, description. It can also include pedagogical elements such as teaching or interaction style, grade level, prerequisites and so forth. Another reason is that LO metadata can be embedded into other standards such as IMS Content Package, Resource List, QTI (Question and Test Interoperability) among others. Some attributes must always be present while others are optional.

The top level hierarchy presents nine categories. They can contain sub-elements, which can be simple elements or may be aggregate elements (i.e. can contain further elements). Some attributes (e.g. Description and Purpose) can only occur once within each instance of the Classification attribute, while the Classification element may be repeated.

There are other metadata specifications, one such being the Dublin Core Metadata schema. This initiative provides a simpler, more loosely-defined set of elements and some of them overlap with the LOM. I have not chosen this specification since it is not specifically designed for educational materials. Therefore, it lacks any categories describing the pedagogical characteristics of the educational objects.
2.2.3 IMS Learning Design

The IMS Learning Design specification is based on the Educational Modelling Language (EML) and combines other IMS specifications, such as: Content Packaging, Metadata and Simple Sequencing. This specification allows the support of a variety of pedagogical models by having created a formal language which can describe all the elements involved in the teaching-learning process and their interactions. The flow of elements is as follows. Activities are structured in a prescribed sequence by the course author (one of range of possible Roles). Those activities are related to specific objects and services (environment) and they are grouped under what is called ‘Method’. Learner properties, Conditions and Notifications are also included in this standard to enable for personalized learning designs. Figure 5 shows all the elements included in this standard. Learning Design allows for three different levels of implementation: Level A, B and C. Each level is mapped to a separate XML schema. The difference among all three levels is that Level B adds Properties and Conditions to Level A, while Level C adds Notifications to Level B.
As stated in Section 2.1, courseware does not allow for ‘live’ activities (e.g. brainstorming sessions) and learners do not get to interact with other students or teachers. What is more, courses provide the same curriculum structure and content to different learners despite individual differences such as knowledge background, learning style, learning speed, and so forth. This standard offers a myriad of possibilities in regards to structuring course content in a sound pedagogically way. By the implementation of this standard, course authors and universities can enrich learners’ online experiences by personalizing the delivery of educational materials. This is an important contribution to distance education since these materials would be better used in learner adapted environments in the future. The challenge of generating a learning design from OCW websites has been addressed in this project. Figure 6 shows the chosen approach to generate an IMS Learning Design from ‘passive’ activities found in open courseware.
Chapter 2: Background

Figure 6. IMS Learning Design generation approach

The mappings between learning materials and the IMS LD schema are fully described in Chapter 3.

2.3 Learning Objects

Learning Objects have been defined in [10] as follows: "Any entity, digital or non-digital, which can be used, reused and referenced during technology-supported learning". Another definition is given in [33]: "Any digital resource that can be reused to support learning."

For the purposes of this project, a learning object is simply a digitised version of existing learning content found in courseware websites. These learning objects can be an image, text, audio files, and so forth. Several research projects are currently devoted to building reusable learning objects from legacy content—such as in [34]. This project offers some guidelines on how to decompose existing content into different elements for post processing and integration with e-Learning standards such as IMS LD. In this project, I am committed to extending the knowledge in this area and to contributing to a better understanding of issues such as: content granularity, learning activities, test-objects. Learning objects provide instructional designers with important benefits such as reduced costs, customisable content, interoperability, flexibility[35, 36]. Figure 7 shows these benefits.
The IEEE LOM standard can be used to describe learning objects in a systematic way by adding metadata to them [10].

### 2.3.1 Challenges

As stated in [34], the challenges in creating reusable learning materials involve several aspects. One important aspect to be addressed is the determination of the size or granularity of LO. The idea here is to create the smallest possible LO so it can be used and/or reused in many courses and different disciplines. This issue has also been addressed in [33]. Another important aspect is related to the accessibility and self-containedness of LOs.
2.4 Learning Objects Repositories

Learning Object Repositories (LORs) are basically storage and retrieval systems for learning objects. They enable users, either registered or not, to search and retrieve learning materials. Their main focus is on the discovery and delivery of content. They can adopt different technologies to carry out their function such as: IEEE LOM or Dublin Core Metadata. Some LORs are open source while others are commercial. There are different types of LORs: client-server, peer-to-peer, commercial, academic and governmental, among other types. Client-server repositories allow users to search for resources through a web portal. Commercial repositories impose some kind of restriction to the content they make available, for example, by paying a fee per use. Academic or government repositories are developed and maintained by universities or organizations to satisfy the specific needs of their communities. In peer-to-peer repositories, each member can make a contribution towards the learning resources and all members benefit from that.

There is also another type of repository which enables a metadata collection from different sources. These are called ‘repositories of harvested metadata’. They allow for a ‘federated search’, where searches can be submitted to a centralized location and they only perform a new query/search when necessary.

Since I use Learning Objects Repositories to test standards conformant content, I will describe several of them. Others have made extensive reviews of LOR [37] and some of them are listed in Table 1.

<table>
<thead>
<tr>
<th><strong>Repositories</strong></th>
<th><strong>Name</strong></th>
<th><strong>URL</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>HARVESTROAD</td>
<td><a href="http://www.harvestroad.com/">http://www.harvestroad.com/</a></td>
<td></td>
</tr>
<tr>
<td>INTRALLECT</td>
<td><a href="http://www.intrallect.com/">http://www.intrallect.com/</a></td>
<td></td>
</tr>
<tr>
<td>LORS (OpenACS/.LRN)</td>
<td><a href="http://www.openacs.org/">http://www.openacs.org/</a></td>
<td></td>
</tr>
<tr>
<td>Melete (Sakai)</td>
<td><a href="http://etudesproject.org/melete.htm">http://etudesproject.org/melete.htm</a></td>
<td></td>
</tr>
<tr>
<td>edna</td>
<td><a href="http://www.edna.edu.au/">http://www.edna.edu.au/</a></td>
<td></td>
</tr>
<tr>
<td>SMETE</td>
<td><a href="http://www.smete.org/">http://www.smete.org/</a></td>
<td></td>
</tr>
<tr>
<td>MERLOT</td>
<td><a href="http://www.merlot.org/">http://www.merlot.org/</a></td>
<td></td>
</tr>
</tbody>
</table>

*Table 1. List of Learning Objects Repository Systems*
MERLOT

MERLOT stands for Multimedia Educational Resources for Learning and Online Teaching and is a public repository for higher education learning materials. These materials are created by registered members and can be peer reviewed. All materials are licensed under the Creative Commons License. Its main goal is: "to improve the effectiveness of teaching and learning by increasing the quantity and quality of peer reviewed online learning materials that can be easily incorporated into faculty designed courses."

SMETE

The SMETE Digital Library is an online portal where teachers and students can get access to a variety of learning resources for all levels, especially in science, math, engineering and technology.

This project is partially funded by the National Science Foundation, National STEM Education Digital Library program. This repository is IMS compliant. All resources can be downloaded and peer reviewed. Users can also upload content. This repository was built using SOAP and WSDL technologies, and users can send a query to the collection by using the following XML DTD (Document Type Definition):

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SearchResult [ 
<!ELEMENT SearchResult (doc*)> 
<!ELEMENT doc (DID, propList)> 
<!ELEMENT DID (#PCDATA)> 
]> 
```

A complete IEEE LTSC LOM (Learning Object Metadata) element set will be returned for each DID.

This project offers a series of tools for interoperability such as the NEEDS Cataloguing Tool, which allows authors to create metadata for their resources. It also supports a federated search, enabling search over multiple and remote hosted SMETE collections at the same time.
Education Network Australia

Education Network Australia (edna) is a public online resource collection in Australia. It started in 1996 and is being managed by education.au limited. State and territory governments are involved in this initiative with the aim to provide free resources and tools to the education community. This repository is based on the Dublin Core metadata standard in accordance with the Australian Government Locator Service (AGLS).

One important characteristic of edna [38] is its involvement in metadata harvesting, which is defined as the process of gathering metadata records from different related websites and repositories such as universities, libraries, and so forth.

Melete (Sakai)

Melete is an open-source course management tool, developed by the ETUDES project, which can be added to Sakai implementation. It supports IMS Content Packaging specification and courses can be both uploaded and exported in an easy way.

intraLibrary digital repository

intraLibrary is a commercial digital object repository, developed by the Intrallect company. It can manage learning resources such as presentations, videos, whole courses and so on. Users (e.g. educators) can share and reuse their resources in a very simple way. This repository supports the IEEE LOM metadata model.

HarvestRoad Hive

HarvestRoad Hive is a commercial federated digital repository system. This repository allows different repositories (e.g. library systems) around the world to be connected and exchange learning objects. It is standard compliant, including metadata (IEEE LOM), IMS Content Packaging (IMS-CP) and copyright management. It can be integrated with most popular LMS such as: Moodle, Sakai and WebCT.
Learning Object Repositories Comparison

In this section I summarize and provide a non-exhaustive comparative analysis of the content interoperability features of the Learning Object Repositories introduced before. This comparison presents the features and functionality of the standards and specification that these products implement as shown in Table 2.

<table>
<thead>
<tr>
<th></th>
<th>Melete (Sakai)</th>
<th>LORS (.LRN)</th>
<th>HarvestRoad Hive</th>
<th>SMETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project type</td>
<td>Open source</td>
<td>Open Source</td>
<td>Commercial</td>
<td>Open Source</td>
</tr>
<tr>
<td>Metadata</td>
<td>IEEE LOM</td>
<td>IEEE LOM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interoperability</td>
<td>IMS CP</td>
<td>IMS CP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Learning Object Repositories Comparison

2.5 Information Extraction Techniques

In this section, I will introduce several projects which deal with the process of extracting data from the Internet according to what’s been stated in the introduction of this thesis.

The goal of an Information Extraction System (IES) is to locate, extract and transform unstructured input documents (e.g. job advertisements, news stories, real estate listings) into a structured format, which can be used for later post processing by many other applications such as Web mining or searching tools [39].

IE systems can be built manually. However, due to the dynamic and heterogeneous nature of web documents, manual systems are not feasible and most of them work on specific web sites and/or domains [40]. Others apply machine learning or data mining techniques to learn the extraction rules for web documents in a (semi-) automatic fashion [41].

The output generated by IE systems can be either in database structured form or in XML form. Figure 8 presents a general view for an IE system.
In general, there are two types of IE systems: IE systems applied to/used for unstructured text and IE systems applied to (semi-)structured data [42]. In this context, unstructured content refers to 'natural language texts' while (semi)-structured content refers to documents which hold some kind of meta-information such as web pages (i.e. HTML tags). The differences between both types of system are considerable. IE techniques for unstructured documents (classical IE), with roots in the Message Understanding Conferences (MUC), are based on linguistic pre-processing such as syntactic, semantic and/or discourse analysis [41-43]. On the other hand, IE systems for (semi-)structured content (structural IE) takes advantage of the tags and/or delimiters found in web documents as meta-information. These systems usually apply machine learning techniques to extract the data.

It is also important to highlight the difference between IE systems and Information Retrieval (IR) systems. IR systems aim to retrieve all relevant input documents out of a large collection of documents in an automatic way [44]. This has many applications such as document classification and categorization, indexing [45]. More information on IR applied to web documents can be found in [23].
As shown in Figure 9, IE systems can be classified according to three different dimensions: a) task difficulty (i.e. input files), b) techniques used and c) automation degree. In addition to that, they can also be classified according the required user interaction: manual, supervised, semi-supervised and unsupervised IE systems.

Any IE system can take three different types of input files: structured, semi-structured or free text. The definition of these terms can be different depending on the research domain viewpoint. I will define them according to the type of the data sources used in this project. Structured content refers to XML documents since they have a DTD or schema attached to them which describe the data they contain. On the other hand, free text is considered unstructured content because these files require natural language processing. HTML pages are considered to be semi-structured since they make use of HTML tags as embedded data. This project deals with courseware, which is made up of web (HTML) pages usually generated from relational databases using templates.
The ability for an IE system to use different input files or a combination of them determines the degree of difficulty that can be taken to evaluate the system. Different IE systems can apply different techniques in order to extract data from data sources. They include: encoding schemes, scan pass, extraction rule type. For instance, scan pass refers to how many times the extraction rules are applied to the pages. The type of extraction rules can be mainly path expressions (e.g. html.head.title) or delimiters (e.g. HTML tags or words). These rules can follow a top-down or bottom-up approach.

Some IE applications require users to label input documents before data can be extracted, while others do not require this since it is done by the system itself.

Manually constructed systems require users to have much experience in programming since they have to code part of the system for each web site by hand. This makes them expensive systems. Examples of such systems are: Minerva [46], WebOQL [47], W4F [48] and XWRAP [49].

In supervised systems users provide a set of labelled examples as input and the system might suggest new pages to be labelled by the user. Examples include: STALKER [50], SRV [51] and WHISK [43].

Semisupervised systems require user supervision after the learning phase. Examples include IEPAD [52], OLERA [53] and Thresher.

Unsupervised systems do not require input training examples and no user interaction is needed. They can solve both page-level and record-level extraction tasks. Examples include: RoadRunner [54], EXALG [55] and DEPTA [56].

Methodology

Basically, our research methodology is as follows: an HTML page undergoes several steps before data can be extracted from it and mapped to the desired XML schemas. In first place, a web page is retrieved and downloaded from the Internet. Secondly, that web document is transformed into a DOM-tree representation, which consists of different types of nodes: root, other nodes and leaves. Each node is mapped/reconciled to different HTML tags such as: <title>, <img>, <a>, etc. A parsing process separates HTML tags, attributes and content. Efficient regular expressions are applied to locate particular content within the document. Content can be of different types: free text, tabular and multi-value fields. Specific functions were developed to deal with this type of content. Once content has been located and its
type determined, data can be extracted. At this time, the extracted data is enclosed in XML tags for later processing. It is important to highlight that my methodology involves three different schemas (IEEE LOM, IMS-CP and IMS-LD). A specific layer has been devised to handle this task. This methodology is described in detail in the following chapters.

Regarding related work in IE technologies, our proposed approach is in line with the latest developments in the field as stated in [57]. OCWise, a multilayered architecture devised during the course of this dissertation, applies the methodology described in the previous paragraph. OCWise is distinguished from many other IE systems by its novel approach to reusing (educational) content. Compared to other approaches described in [57], I do not use a grammar based approach for extraction but relay on the DOM object model, which gives more flexibility and the ability to navigate a document not only top-down but bottom-up.

Having novel features such as hierarchical navigation and the power of regular expressions, the extraction layer is more robust than other systems. OCWise can be easily applied to other data sources due to its flexible design (configuration files). The way we handle XML is different from other IE systems such as XML-QL since XML files are produced by the extraction layer with no explicit structure.
2.5.1 Challenges in e-Learning

After reviewing recent research on information extraction applied to e-Learning, I identified one main challenge—automatic metadata extraction from learning resources. This important challenge is closely related to the challenge actually faced by courseware, of making courses standard compliant and therefore, reusable. Both challenges complement each other and they are in the line with this research project. One of the main problems in e-Learning is in the difficulty of finding relevant courses. This issue is very important since the production of online courses and learning resources is expensive and time consuming. Therefore, reusing existing course content is in high demand. Most of the existing learning resources cannot be discovered because they were not annotated with semantic information. Doing this manually is not a feasible option due to the amount of courses and time constraints. One solution to this challenge could be the application of information extraction techniques in order to extract metadata from course content available on the Internet in an automatic way [58-61].

Designing and implementing a tool capable of tackling this challenge requires deep understanding of several technologies such as information extraction techniques, e-Learning standards, and the use of different Learning Management Systems. The design and implementation of the tool produced by the author is fully described in Chapters 3 and 4.
2.5.2 Survey of Web IE Systems

GATE

GATE [62] stands for General Architecture for Text Engineering and consists in a graphical software application for text mining. It is an open source project written in Java and it first release was in 1996. It involves three main elements: architecture, a framework and a visual environment.

It supports a number of standard formats such as XML, RTF, and HTML. It uses UNICODE in order to provide multilingual support in 28 different languages. It provides support for different Machine Learning algorithms for text mining, entity recognition and relation extraction. In regards to information extraction support, it includes a system called ANNIE (A Nearly-New IE System). This system includes a tokeniser, sentence splitter among other processing resources.

UIMA

Unstructured Information Management Architecture (UIMA) is an IBM open source project aimed at providing a framework for creating and delivering Unstructured Information Management (UIM) technologies, which are based on Natural Language Processing (NLP), Information Retrieval, etc [63]. Its architecture comprises four different components: Acquisition, Unstructured Information Analysis, Structured Information Access and Component Discovery.
Keyphrase Extraction Algorithm

KEA (Keyphrase Extraction Algorithm) is an algorithm for extracting keyphrases from documents using Naive Bayes machine learning techniques [64].

This algorithm performs two separate processes: training and extraction.

The training process main goal is to identify candidate keyphrases, calculate feature values and choose the best keyphrases. This process applies information extraction techniques to automatically extract data from input documents. Once data has been extracted, it undergoes different steps: first, data is split into tokens. A token is a set of letters and/or digits. Then, a set of rules are applied in order to identify the best keyphrases. Finally, data is case-fold and stemmed using the Lovins method.

This process calculates two features: $TFxIDF$ and $First Occurrence$. $TFxIDF$ is a measure of a phrase's frequency in a document compared to its rarity in general use. General use is represented by document frequency.

The first occurrence value is calculated by counting the number of words which precedes the phrase's first appearance and then dividing it by the number of words in the document. The extraction process uses the values calculated during the training process to determine candidate phrases and feature values in new input documents.
2.6 Evaluation measures

The quality of results produced by an information retrieval and extraction systems can be measured by using a Confusion Matrix shown in Table 3 [65] [66] [67].

<table>
<thead>
<tr>
<th>Answer</th>
<th>Prediction</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>+</td>
<td>True positive</td>
<td>False negative</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>False positive</td>
<td>True negative</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Confusion matrix.

A confusion matrix table shows both the actual and predicted values made by an Information Extraction System. The columns show the predicted values while the rows show the correct values.

The True positive value represents a correct extracted value, while False positive represents an incorrect extracted value. The False negative value in the matrix represents the number of fields that should have been extracted, but for some reason, they could not been extracted. The True negative value is not usually used.

All common performance measures found in the literature, such as recall and precision, can be computed from the confusion matrix.

Before filling the confusion table, one should make a decision about how to determine if an extracted fact or data is correct or not. At this point, there are three different possibilities:

- exact rule: this rule says that a predicted or extracted value is only correct when it is exactly equal to an answer. Thus, if ‘John Smith’ is the given answer, ‘Dr. John Smith’ would not be correct.
- contain rule: this rule allows for extra ‘tokens’ in the extracted value (provided the extracted value contains the right answer) and it is considered correct. Thus, if ‘John Smith’ is the given answer, ‘Dr. John Smith’ would be correct as well.
• *overlap* rule: This rule applies when the extracted value contains part of the right answer along with extra ‘tokens’. Thus, if ‘Dr. John Smith’ would be the given answer, ‘John Smith’ would be correct as well.

Which rule is suitable depends on each specific situation or application. *Contain rules* have been chosen and applied in this project since they allow to extract much data from many documents.

### 2.6.1 Definition of common performance measures

Many different performance measures have been proposed. These measures require a set of input documents and a query. They can be applied to each extracted value or field (e.g., author name, keywords, course description) separately and then calculate the average of all results. It is important to highlight that some fields are easier to extract than others. For instance, an email address is easier to extract than an author name.

The most used evaluation measures in information extraction systems are recall, precision and F-measure. They are shown in Figure 10.

![Figure 10. Evaluation Measures](image-url)
Recall \( (r) \), precision \( (p) \) and accuracy \( (a) \) can be calculated from the confusion matrix:

\[
\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}
\]

\[
\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}
\]

\[
\text{accuracy} = \frac{\text{true positive} + \text{true negative}}{\text{true positive} + \text{true negative} + \text{false positive} + \text{false negative}}
\]

Recall indicates the number of correct extracted values reported by the system (\textit{number of true positives divided by the total number of elements which belong to a particular category}). Precision shows how many of the extracted items or fields are correct (i.e. the percentage of the information reported as relevant by the system that is correct). The traditional F-measure or balanced F-score is a popular performance measure which combines Precision and Recall (i.e. weighted harmonic mean of precision and recall). It defined as follows:

\[
F = \frac{2 \times p \times r}{p + r}
\]

There are other measures which are used in addition to the previously described. They are \textit{fallout} and \textit{overlap}.

Information retrieval provides a measure (called fallout) of the degree to which a system’s performance is degraded by the availability of a large number of irrelevant documents. If Irrelevant is the total number of irrelevant documents, and \textit{false positive} is the number of these which a system inappropriately labels as relevant, then fallout is calculated by:

\[
\text{fallout} = \frac{\text{false positive}}{\text{false positive} + \text{true negative}}
\]
It measures the tendency of a system to be led astray by irrelevant documents. If field instances are relevant objects, and all other fragments of appropriate size—any fragment containing no fewer tokens than the smallest training instance, and no more tokens than the largest—constitute the set of irrelevant objects, then we have one measure of the degree to which a system successfully copes with the inherent difficulty of the extraction problem.

\[
\text{overlap} = \frac{\text{true positive}}{\text{(true positive + false positive + false negative)}}
\]
2.7 Implementation Technology

In this section, I briefly describe the technology used to develop the OCWise system.

XHTML
XHTML stands for eXtensible HyperText Markup Language and is a new version of HTML in the sense that it is stricter and cleaner. HTML will be gradually replaced by XHTML, since it is the latest version of HTML. The characteristics of XHTML that set it apart from HTML are:

- XHTML elements must be properly nested
- XHTML elements must always be closed
- XHTML elements must be in lowercase
- XHTML documents must have one root element

There are three different DTDs in XHTML, Transitional, Strict and Frameset. The Strict DTD is used when the markup language requires clean up. The Transitional DTD is used when support for browsers, which do not support/understand CSS, is needed. The Frameset DTD is used when support for HTML frames in documents is required.

I decided to use this W3C recommendation in this project because it paves the way for an easier and smooth transformation from HTML into XML, and allowed the correction of ill-formed pages as they were downloaded by my crawler. I found that over 50% of the pages contained missing ending tags and invalid characters among other issues.
XML

XML stands for eXtensible Markup Language and it is a markup language similar to HTML. It was designed to store and share structured data, where users can define their own tags. XML files are just plain text files and they can be handled by almost any software application. XML files are meant to be self-descriptive since the author of the file can create and use his/her own tags. For instance, the following example is a course description stored in XML:

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<course discipline="Computer Science">
  <title>Neural Networks</title>
  <authors>
    <author>Dr. Aiman Turani</author>
    <author>Dr. Juan Jose Garcia</author>
  </authors>
  <level>postgraduate</level>
  <language>English</language>
</course>
```

All the tags in the above example were created/defined by the author of the course. This is because in XML language there are no predefined tags. What is more, we can create our own file structure. As seen in the example, XML documents have a tree structure that starts at "the root" and branches to "the leaves".

The structure of this example XML file is as follows. The first line is the XML declaration, which defines the XML version (1.0) and the encoding used (ISO-8859-1 = Latin-1/West European character set). The next line describes the root element of the document. In this case, it reads: "this document is a course". The next lines describe child elements of the root (title, authors, level, and language). And finally, the last line (`</course>`) defines the end of the root element.

An XML file with correct syntax is called well formed XML. An XML file validated against a DTD is referred to as having valid XML. In a case of a valid XML file, it should include a DOCTYPE declaration, which is a reference to an external DTD file. XML DTD will be described in the next section.
For example,

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE course SYSTEM "course.dtd">
<course discipline="Computer Science">
  <title>Neural Networks</title>
  <authors>
    <author>Dr. Aiman Turani</author>
    <author>Dr. Juan Jose Garcia</author>
  </authors>
  <level>postgraduate</level>
  <language>English</language>
</course>
```

One of the main reasons for choosing this W3C recommendation is that it is widely adopted by a large number of applications and vendor software around the globe. Also, it enhances, facilitates and simplifies data storage and sharing because XML is independent of the hardware, software or application used by any user or organization. This is an important feature to highlight for this research since one of the issues focussed on by this project is that of interoperability of learning resources across multiple repositories and learning management systems.

Another important feature about XML is that it separates data from HTML. This point is very important in the decision to choose this standard due to the fact that users (e.g. teachers, students) can deliver content in different fashions (e.g. content layout and presentation), in different languages, to different devices (hand holds, mobile devices), different browsers, different applications, and so forth. Course authors can focus on the data to deliver to students and enhance online education. This technology is also used in all IMS Educational Standards such as IMS CP, IMS LD and IEEE LOM.
**DTD**

An XML DTD (Document Type Definition) file defines the structure of an XML document and specifies a list of legal elements or tags. Using the above example, the following DTD file can be created:

```xml
<!DOCTYPE course [
  <!ELEMENT course (title,author,level,language)>
  <!ELEMENT title     (#PCDATA)>
  <!ELEMENT author     (#PCDATA)>
  <!ELEMENT level      (#PCDATA)>
  <!ELEMENT language   (#PCDATA)>
]>```

PCDATA stands for parsed character data and all elements will be parsed by a parser.

There is an alternative to using DTD, namely XML Schemas. They are described in the next section.

**XML Schema**

An XML Schema defines the valid or legal sections of an XML document. An XML Schema defines the following characteristics:

- elements that can appear in a document
- attributes that can appear in a document
- which elements are child elements
- the order of child elements
- the number of child elements
- whether an element is empty or can include text
- data types for elements and attributes
- default and fixed values for elements and attributes
This is the corresponding XML Schema for the above DTD file:

```xml
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.imsglobal.org"
    xmlns="http://www.imsglobal.org" elementFormDefault="qualified">
  <xs:element name="course">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="title" type="xs:string"/>
        <xs:element name="author" type="xs:string"/>
        <xs:element name="level" type="xs:string"/>
        <xs:element name="language" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

I chose to use XSD schemas since they are much more powerful than DTDs; as they support data types and namespaces, they are written in XML and they allow future additions. All these features allow for the validation of extracted data from different data sources (e.g. OCW websites), and the definition of data patterns. What is more, schemas can be reused in different sources.

When it comes to displaying data to users, XSLT technology comes into action. It is described in the following section.
**XSLT**

XSLT stands for eXtensible Stylesheet Language Transformations and it can be used to transform an XML file into a HTML file for presentation purposes. An example of its usage is as below:

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="simple.xsl"?>
<course discipline="Computer Science">
  <title>Neural Networks</title>
  <authors>
    <author>Dr. Aiman Turani</author>
    <author>Dr. Juan Jose Garcia</author>
  </authors>
  <level>postgraduate</level>
  <language>English</language>
</course>
```

**XMLTH**

XPath is a language used for locating information in an XML document. It allows defining path expressions in order to navigate the different sections or parts in an XML document, such as nodes or node-sets. XPath has over a 100 built-in functions mainly for data manipulation.

**Conditions** can be included in all the queries performed on an XML file. For instance, one might want to retrieve courses in English only or just postgraduate courses. This can be easily done using path expressions. This is a desirable feature to be used in this project because one can perform complex queries on the data and thus retrieve semantic information in a systematic and efficient way. These rules or expressions can be easily adjusted to different data sources without having to modify or update the source code. In order to improve the efficiency and accuracy of the OCWise system, I complemented path expressions with regular expressions such as match and split among others.
Examples

Supposing one wants to retrieve the title of a specific course, then one would use the following path expression:

/course/title

In the case of wanting to select all the authors, use: /course/authors

But one might want to retrieve the first author for that course. In that case, one would use:

/course/authors[1]/author

An example of including conditions to path expressions is:

/course[language="english"]/title
/course[level="undergraduate"]/title

This project mainly deals with open learning resources available on the Internet, as described in the Introduction. These resources lack structure and they are not easily discoverable, reusable or interoperable. Online education can be enhanced and improved by giving them a new shape. This task is far from easy since all these resources are in HTML format and to convert them into an XML format requires the application of different technologies such as the information extraction techniques described previously. Also the learning rules have to be optimized in order to cope with diverse formats and multiple data sources. The underlying rationale behind this process is the following.

First, break each course down into several chunks of data or objects, where each object can then be tagged using the extracted metadata. At this stage, the challenge faced is that not all the courses contain the same structure (e.g. some courses contain Projects and Exams, while some others do not). This is something that I took into account in the system design and it is described in Chapter 3.
3 OCWise Design

This chapter describes in detail the system design for the OCWise. This tool is the result of a comprehensive study on how to make the best use of learning resources available on the Internet by the combination of information extraction techniques and use of open educational standards. The theoretical framework provided in the Background Chapter (Ch.2) forms the basis for this development. The main objective of this tool consists in devising new extraction rules in order to reshape legacy learning content and making a contribution towards enriching online education through enhancing reusability and interoperability of digital learning materials among different Learning Management Systems (LMS) and repositories.

Section 3.1 describes the three layered architecture and its design rationale. Sections 3.2, 3.3, 3.4 and 3.5 explain each layer in our design focusing on the interrelation between layers and their innovative features.

As illustrated in Figure 11, the OCWise tool has a three-tiered architecture based on the latest Web IE system attributes found in the research literature [68], being most of them described in the Background chapter. Examples of such attributes are: accuracy, resiliency, general, extensible and open source. All these attributes were considered in the tool design.

![Figure 11. OCWise System Overview](image-url)
3.1 Design Description

Key design requirements of this project are that of interoperability, reusability, searchability and repurpose of existing educational content. This is in line with current research projects around the globe [60, 61, 69]. In order to deal with the lack of structure and semantic information in learning content, and as well as the diversity of data sources (e.g., OCW repositories), a multi-layered architecture is chosen to achieve these goals. Each layer or module in the tool provides interfaces to the other layers. That makes this tool flexible and cohesive, since each layer can be adapted to new changes in the data sources and/or updated with new rules (retrieval, extraction and/or mapping).

Design Rationale

The system architecture is shown in Figure 11. The design process is relatively simple. First, the user specifies data source(s) parameters such as URL, crawling depth. All this information is stored in a text file. Each parameter in this file is explained in Section 3.2. At this point, no more user interaction is required since all layers run in an automated way. The advantage here is that user interaction is kept to a minimum. This text file is then loaded by the first layer, which starts downloading each course and/or learning resources until all the courses have been downloaded. It also creates a custom directory tree for each course for offline post-processing. Users can specify to download similar content from different sources (e.g. similar/same content in different languages) or content belonging to different disciplines. This provides great flexibility when dealing with several sources. Downloaded courses are automatically processed by the Data Extraction Layer. This process involves ‘correcting’ ill-formed web pages (e.g. missing end tags), inferring course structure, the extraction of semantic information about each course/resource, and activates sequence analysis in a bid to determine the underlying Learning Design among many other tasks. An intermediate text file is then produced as an output providing all the needed information to the next layer. This layer, the Data Mapping layer, produces valid XML files conforming to the following schemas: IMS CP, IMS LD and IEEE LOM. Three different modules have been devised from scratch and implemented according to the best practise guidelines found in the IMS Global Consortium website. All these modules play a vital role towards the conversion
process as well as contributing to the improvement of online education by making the most of Open Educational Resources and assuring the goals of interoperability, reusability and repurpose. The last layer is in charge of generating a valid PIF file which includes the imsmanifest.xml file (this file contains course structure, metadata and sequence of activities) and all the resources belonging to the course. The output produced by this layer is a single self-contained zip file which is ready to be imported into any IMS CP compliant LMS or repository. At this point, a user (teacher, student) should be able to ‘transport’ content from an LMS/repository to another one, reuse and repurpose content. In addition to that, other users can easily find (or locate) the content uploaded into the repository by means of metadata. Every PIF file was carefully tested against four different educational applications (WebCT, Sakai, .LRN and Reload). Test results show a high level of standard adoption which demonstrates the efficiency and accuracy of the tool (refer to Evaluation Chapter 5).

Nevertheless, there are a few main challenges in this design. These challenges are mainly found in the second layer. The first challenge was the complexity of dealing with three different schemas. Each schema imposes a set of strict rules and data types in order to generate a valid XML file. Many sections in those XML files are interrelated. This means that, in some cases, I followed a bottom-up approach in the tool design. In addition to that, some sections in those XML files were left ‘empty’ until extra information was retrieved and could be processed.

Another challenge was defining the right set of extraction rules which would work for all the pages in all the courses for different data sources. I found that different pages belonging to the same course presented different HTML source code when displaying information to the user. For instance, one page might use a ‘<p>’ tag to show specific information to the user, while another page uses a ‘<div>’ tag for the same purpose, and yet a third page uses a ‘<table>’ tag. Some information presented multiple values (e.g. course authors, keywords). As can be seen on any OCW website, courses show different structure. Some of them contain ‘Projects’, while some others do not. This imposes another challenge to be considered and addressed, and means that the tool should be able to detect such layout on the fly.

In order to build a robust tool, I have identified some key issues regarding its design. First, the tool design shows three separate but cohesive layers which offer the possibility of reusing. Second, it acknowledges data contained in a web page may have different granularities. For instance, a web document can contain lists, tables
and so forth. In some other cases, the web document may contain a comma-separated list of values inside a table cell. Therefore, the tool makes use of regular expressions to capture data in such format. Another key design issue was that the tool produces an abstract representation of an HTML document. This object-model provides some advantages when applying data mappings and facilitates document navigation. Another important aspect taken into account in this tool design was to determine the minimum granularity of the extracted content under the premise that a small 'chunk' of content could be reused in a different context or discipline. This design issue is very much in the fore-front of the research literature [70].

3.2 Retrieving educational materials from the Web

The first stage of the tool consists of retrieving learning resources from the Internet for post-processing. This post-processing is carried out in an offline way. This layer sends an HTTP request to a specified remote server (e.g. MIT OCW website) and fetches all the pages which make up a course and then processes the next course until all the courses have been downloaded. This first layer makes use of retrieval rules which specify the method (e.g. GET), the starting URL, the crawling depth and resource types to be downloaded. Examples of such rules are shown in Figure 12.

The method GET sends a request to download a web page from the specified repository. The URL parameter indicates the starting point from which all pages are retrieved. This parameter might also specify the protocol (e.g. http) and port number (e.g., 80, 8080). The Crawling depth parameter specifies the number of slashes ('/') in a URL from its root (URL parameter). This parameter prevents this layer from being drawn into an infinite loop. All documents beyond the specified crawling depth are discarded. The last parameter, Resource Type, lists the allowed MIME types and/or file extensions to be included when crawling a repository.
The above retrieval rules are dependant on the data source structure. Figure 13 shows a basic course layout. After analysing the structure of several data sources, I noticed that almost of all them shared the same basic structure. This characteristic is exploited by this first layer.

A new custom directory structure was devised to hold each part of the course and pave the way for an efficient and faster post-processing. One of the key features developed in this layer is that different retrieval rules can be applied to different data sources (e.g. different repositories). Once all the courses and resources have been successfully retrieved, the next layer comes into action.
3.3 Extracting information

Each web document belonging to a specific course is first transformed into a hierarchical tree according to the Document Object Model (DOM) [71]. This tree representation of an HTML page consists of a root, nodes and leaves. Each node is mapped to an HTML tag (e.g. `<title>`, `<img>`). Figure 14 shows this mapping.
This hierarchical structure allows for navigation of the tree in a very efficient way using path expressions [72]. For instance, the expression 'html/head/title' will lead to the node containing the <TITLE> tag, under the <HEAD> tag. Once a node of interest is located, its contents or value can be extracted. This layer tries to make the most of the HTML structure by combining the DOM object-model and the power of regular expressions in order to extract as much data as possible. For instance, Figure 15 shows a course description taken from an OCW repository.
The image in Figure 15 shows the discipline of the course, its code and title, authors, course level among other information. Such information is found in the course home page only. Other web pages contain extra information about the same course as shown in the basic course structure (See Figure 13 above). As explained in the design section previously, different web pages present data to the user in different formats (e.g. bulleted list, paragraphs, tabular, graphical). In addition to that, the information to be extracted is spread among several parts of the same course. This is dependant on the course structure, which might not be the same for all courses in the same repository. The challenge here was defining the right set of extraction rules which could work successfully for all the pages in the course irrespective of their format, position and course structure. The Figures 16-18 show a set of extraction rules applied to learning materials.
The set of extraction rules shown in Figure 16 enable the extraction of course title, authors, description and keywords. Since a course can have several authors, regular expressions are applied to this field.
As seen in Figures 16-18, extraction rules cover a wide range of HTML tags (e.g. `<div>`, `<table>`, `<meta>`) used in every document belonging to the same course. These rules are assisted by the power of regular expressions to deal with multiple granularities in the data to be extracted. This is an important advantage to highlight in my tool because it ensures the extraction of as much data as possible.

In particular, two operators are applied: *match* and *split*. The match operator allows extracting a string according to a specified string pattern. For example, the pattern `/[a-z]+\s+\d*/` is used when we want to match a lowercase word, at least some space, and any number of digits in a string. The *split* operator returns a list of substrings given a string and a separator as inputs. For instance, the following command `split(/(\?=\w+)/, 'hi there!')` produces the output `'h:i:t:h:e:r:e!'`.

Extraction Rules have been described in Chapter 2.

This layer produces intermediate files with include important information about course structure, sequence of activities, attached resources. All these files are read by the next layer.
3.4 Mapping Information

This last layer uses XML mappings to create XML elements out of intermediate files, provided by the previous layer. A set of mapping rules was devised to assist with the production of XML elements. Figure 19 shows such mappings. A SAX-parser is then used to test the validity of our generated XML files according to XML schemas used in this project.

![Data mappings](image)

**Figure 19. Data mappings**

The Figure 19 is an example of how an HTML file (with no structure at all) can be transformed into an XML file (i.e. a structured file).

3.5 Wrapping-up process

The last step in the design of OCWise tool consists of putting all the pieces together into a self-contained single file. This is a zip file (a type of compressed file) containing course structure, activities sequence and all linked resources along with the extracted semantic information. This makes learning materials easy to discover by other users (students, teachers) as well as by metadata-aware search engines. Figure 20 shows this process.
In this chapter, I have described the design of the OCWise tool in detail and explained how the challenges were addressed. In the next chapter, I will describe the technical details of the implementation of this tool thoroughly.
4 System Implementation

Based on the description of the system design covered in Chapter 3, the OCWise tool has been implemented. It has been released under an open source license as a part of this project. This chapter describes some of the implementation decisions which have been made in OCWise tool and their rationale. Section 4.1 discusses the choice of implementation language for the development. The tool implements three different e-learning specifications, IMS Content Packaging, IEEE LOM Metadata and IMS Learning Design as well as information extraction techniques. Section 4.2 introduces a brief description of the implemented technology and deployment underlying the development. The last section explains the implementation of standards and techniques used in this project.

4.1 Implementation Language

I chose the interpreted scripting language, Perl, for the OCWise project, as it provides the following benefits:

• Perl is widely used powerful text processing tool; hence it should be relatively easy for users of the framework who wish to customize the processing capabilities.

• A large number of contributed Perl modules are freely available for many different tasks on the comprehensive Perl archive network (CPAN), extending the domain of applicability of the framework.

• Perl is an expressive high-level language that allows for rapid prototyping, so the framework developers and application developers can experiment with several alternative designs fairly quickly.

• Perl is widely deployed and is part of all standard Unix distributions. It is available for most platforms that have a C compiler, and because of common high-level interfaces, Perl code written on one platform is often more portable to other platforms than the equivalent C code would be.
4.2 OCWise Information Flow

As illustrated in Figure 21, the user initiates the process by specifying an OCW website or repository, which contains a collection of learning resources in HTML format. This is achieved by the custom web crawler. Then, the Data Extraction stage extracts information from HTML documents and encodes it in XML documents. Finally, the Data Mapping layer transforms any HTML documents into XML documents to conform to the schemas used in this project. Thus, the HTML collection of courses is reshaped into a homogeneous repository of XML documents. This repository can be used by other applications for indexing, formatting, storage or preservation, querying purposes among many other applications.

![Figure 21. OCWise Information Flow](image)

Each stage in the process flow has its own associated package or module. Each package implements particular functions and schemas in a cohesive and decoupled way. The following section describes the tool deployment packages and their relationships.
4.3 OCWise Deployment

As shown in Figure 22, the implementation consists of four PERL packages, each with no User Interface (UI), and which can be embedded into any repository to extend its functionality. The system is platform-independent, thus it can run on Windows, Linux or Mac computers. These modules can work either online or offline. In the case where resources have to be downloaded from the Internet, obviously an Internet connection is required.

Figure 22. OCWise Deployment Diagram

The above UML deployment diagram (Figure 22) shows the main package, OCW, along with its dependent IMSCP, LOM and IMSLD packages. The IMSCP package generates a valid imsmanifest.xml file according to the IMS CP standard. This XML file holds course structure, sequence of activities and resources. The LOM package implements the IEEE LOM standard. All the semantic data, which has been extracted by the Data Extraction Layer, is mapped to this standard and merged within the imsmanifest.xml file. It is important to mention here that the IMS CP package is a big container in the sense that it can accommodate other standards in its structure such as IEEE LOM and IMS LD. This can be seen as an advantage since only one XML file holds all the information. This means there is only one file to maintain (e.g. update metadata, modify learning design, and change course structure). Finally,
the IMLD package implements the IMS LD standard following the IMS LD schema. This schema allows course authors to describe and structure content in a formal way. Many different learning designs (e.g. Competency based learning) can be represented within this schema, providing for great flexibility to course authors in terms of delivering customized content to users (e.g. students) according to their learning background, learning speed, preferences, etc.

Our implementation tries to capture the underlying learning design in an innovative way by inspecting each part of a course, mainly its syllabus and calendar, and performing an analysis on the extracted data to determine the right design among a predefined list of designs. Unfortunately, in many cases, it was not possible to capture the learning design due to a number of factors such as, missing information or the author’s involvement was required.

Each package was carefully developed and implemented following a number of important software engineering design criteria such as modularity, scalability, integration, reusability. These are described in the following sections in detail.

### 4.4 Implementation Approach

The UML diagram in

Figure 23 shows the implementation approach chosen for this tool. It is divided in three columns, Process description, Process flow summary, and Components and functions. This last column shows the associated components and functions of the different packages, their ‘inputs’ and ‘outputs’.
Figure 23. List of processes and functions

The OCM.pm package groups a diversity of functions, ranging from resources download (*DownloadWebPage()*)) to zipping files (*zipFile()*). As information flows from one layer to another, different functions from several packages are invoked and executed in turn, according to each particular learning material structure.

The main entry point function defined in the OCW.pm package is called *CourseConversion*, which takes an HTML course as input and generates a PIF file as output. This main function invokes many other functions in turn, as described and explained in the next section.
The rationale behind this entry function is as follows. The first layer invokes this function and provides a pointer to a *to be* processed course until all the courses have been processed. The very first task is to determine course structure by calling the function \textit{GetCourseStructure()}. This function returns every part of a course (Bearing in mind that each course might have a different structure.). Then, each element is processed in a loop until all of them have been processed. Each element is converted into a DOM-tree structure. This loop invokes three main other functions: \textit{GetResources()}, \textit{ExtractMetadata()} and \textit{GetLDesign()}.

The \textit{GetResources()} function is applied to each node in the input tree CT in a top-down fashion from the root to leaf nodes. For each node in the tree, it determines the resource name, type (e.g. PDF, PPT, video files) and URL. The resource can be either internal or external. All internal resources are downloaded and included in the final PIF file. The user can also specify what type of resources to include, as well as

```plaintext
Function CourseConversion(course) : X

begin
    CS = GetCourseStructure(course)  # Figure out course structure

    For Each Element IN CS
        CT = CreateTree(Element)       # Create a DOM Tree
        Resources = GetResources(CT)
        Metadata = ExtractMetadata(CT) # IE techniques implementation
        If (Element IS 'Calendar') Then
            LDesign = GetLDesign(CT)    # Figure out Learning Design
        End If
        PUSH Resources,Metadata,LDesign INTO W
    Next

    RST = GeneratePIF(W)
    Return RST

end
```
external resources or not. A new node with label RES is created with the following attributes, \textit{name} and \textit{value}.

The \textit{ExtractMetadata()} function implements a set of extraction rules as described in Chapter 3. It is important to highlight that some elements are easier to extract than others. This is explained in detail in Chapter 5.

The main idea behind the \textit{GetLDesign()} function is to tease out the underlying sequence of activities outlined by the author of a particular course. This information is generally found in the Calendar element of a course, and could be complemented by the Syllabus. This presented quite a few challenges during the course of this project. In the context of my research, I tried to contribute towards this goal, where so much research is being conducted. According to the Learning Design specification, authors can describe teaching strategies (pedagogical models) as well as educational goals. This specification requires including the following attributes: roles, learning or support activities, learning objectives and services among many others. In order to generate a consistent IMS LD file (or to meet the requirements of this standard), the \textit{GetLDesign()} function collects the required data from different parts of the course such as: Calendar, Lecture Notes, Labs, Syllabus and Study Materials. As an example, the course 6.004 Computation Structures from the MIT OCW website includes learning objectives, measurable outcomes, prerequisites, grading, and list of lectures. Each lecture has notes, assignments and problem sets along with a list of related resources.

The \textit{GeneratePIF()} function generates a valid PIF file by creating a self-contained zip file using all data gathered by all the previous functions. The validity of this file is tested against the specific schemas provided by IMS and IEEE organizations. This PIF file is ready to be processed by any Learning Management System or repository which supports the standards mentioned earlier in this thesis (see Chapter 2). This function is applied in a bottom-up fashion. Thus, it first processes the resources part of the IMS CP standard, which requires a unique ID. Then, it moves on to the Organization part and finally it processes the Metadata section. It has to be done in such a way since one section relies on another. For instance, the Organization section requires specifying the ID for each resource.
4.5 e-Learning Standards Implementation

The implementation of this tool puts into practice the e-Learning specifications of IMS Content Packaging, IEEE Learning Object Metadata and IMS Learning Design. In the next section, I will discuss their implementation and the problems faced.

4.5.1 IEEE Learning Object Metadata

In Chapter 2 there was a brief description about this standard. In this section an example is presented of metadata found in an OCW course, and it was retrieved in order to tag each course with it.

The objective of the retrieval of metadata from each course is to make the course author’s task easier, and most of all, to guarantee quality and consistency of data, enabling efficient search and reuse of Learning Objects.

Metadata is a vital element used to discover e-Learning materials. It works in a similar way to a library index card system and contains details on the object’s subject, contents, author and copyright status as well as other key indicator information.

The MIT courses specify metadata in different formats and types, such as: the meta HTML tag, the H1 or H2 tags, as well as inside tables. Sometimes metadata is available as whole paragraphs (e.g. syllabus) or as tabular data (e.g. grading). Also, the same metadata can be found in different languages for the same course. In this case, I have applied a different approach in order to retrieve it.

There are nine metadata categories (LOM 1 to 9), each of which has subcategories.
The nine basic Learning Object Metadata (LOM) categories are found in Table 4:

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>A general description of the package including title, language, description, keywords among other descriptors.</td>
</tr>
<tr>
<td>Life Cycle</td>
<td>In this category author’s details, version information and a list of contributors are stored.</td>
</tr>
<tr>
<td>Meta-Metadata</td>
<td>Not available</td>
</tr>
<tr>
<td>Technical</td>
<td>This is used to store technical information on each asset in a learning object. Size, location and requirements for other platforms can be specified.</td>
</tr>
<tr>
<td>Educational</td>
<td>This element allows one to describe the educational values and approaches (active, expositive, mixed or undefined), the type of learning resource (e.g. assessment, open activity), methods of delivery, levels of interactivity, who the resource is aimed at (e.g. teacher, learner), educational level (e.g. primary, secondary), the typical age range the resource is intended for, difficulty levels, typical learning time, and a description of how the resource is intended to be used. This last element complements the resource description placed in the General section.</td>
</tr>
<tr>
<td>Rights</td>
<td>This is where the content developer can store information on the cost of the resource, and all-important information regarding copyright.</td>
</tr>
<tr>
<td></td>
<td>I have included the same copyright note from the OCW website.</td>
</tr>
<tr>
<td>Relation</td>
<td>Not available</td>
</tr>
<tr>
<td>Annotation</td>
<td>Not available</td>
</tr>
<tr>
<td>Classification</td>
<td>We have classified each course according to MIT taxonomy, which is included in the package. It’s important to note that this standard allows for more than one classification framework (or schema). This can be done by a subject expert or a librarian.</td>
</tr>
</tbody>
</table>

Table 4. Learning Object Metadata categories
Example of meta tags found in OCW courses:

```
<meta name="group" content="6-034Artificial-IntelligenceFall2002" />
<meta name="Title" content="Artificial Intelligence" />
<meta name="Description" content="Introduces representations, techniques, and architectures used to build applied systems and to account for intelligence from a computational point of view. Applications of rule chaining, heuristic search, constraint propagation, constrained search, inheritance, and other problem-solving paradigms. Applications of identification trees, neural nets, genetic algorithms, and other learning paradigms. Speculations on the contributions of human vision and language systems to human intelligence. Enrollment may be limited." />
<meta name="Author" content="Winston, Patrick Henry" />
<meta name="Keyword" content="artificial intelligence,applied systems,rule chaining,heuristic search,constraint propagation,constrained search,inheritance,identification trees,neural nets,genetic algorithms,human intelligence,knowledge representation,intelligent systems" />
```

**Technical implementation**

Having the specification requirements in mind, I have implemented a set of functions which provide support for IEEE LOM specification. These functions are able to retrieve all the available metadata on each document and fill in (meta tag/annotate) each category in an automatic way (refer to `ExtractMetadata` function). To start with, I have assigned a unique identifier to each content package. In this case, the course code was chosen as this identifier. Including an identifier ensures that a single object can be aggregated with more objects without any problem, and it allows the objects to be managed properly.
4.5.2 IMS Content Packaging

In Chapter 2 I described this standard, so here I concentrate on its implementation. Each course was broken up into parts and each part was mapped to the corresponding section of this standard.

A specific Perl module was developed to handle the creation of an XML file according to the IMS CP standard. This module takes care of each section as well as adding metadata to the manifest.

Each content package must include a top-level Manifest file (or IMS Manifest File), which should always be named imsmanifest.xml, and always be presented in lower-case characters. This special XML file describes the package itself which is divided into three parts:

- Metadata
- Organization
- Resources

Organization

The Organization is one of the three key elements contained in a content package’s Manifest. It contains a description of how the contents of the learning object are to be presented to the learner, including their presentation order and any particular sequence. Each Manifest must contain an Organization.

Essentially the Organization section organizes the learning object’s contents into a tree structure that complies with the pedagogical approach taken by its designer. In this way, the Organization is used to present and control the hierarchy of the contents.

An example of an Organization is:

```xml
<organizations default="TOC1">
<organization identifier="TOC1">
<title>Artificial Intelligence Course</title>
=item identifier = "ITEM1" identifierref = "RESOURCE1">
<title>Introduction</title>
</item>
</organization>
</organizations>
```
In this chapter, I have discussed the technical details and challenges of implementing IMS Learning Resource Metadata, Content Packaging and Learning Design. I hope the techniques presented in this chapter can be useful for other developers who pursue compliance with these specifications in the future.

In the next chapter, I evaluate my implementation and based on this experience, I provide feedback for the IMS specification developers to help them with future improvement.
5 System Evaluation

In order to evaluate the performance of my system, several aspects have been tested. The two main areas tested are the quality of extraction and efficiency. For testing the quality of extraction and efficiency, performance is measured on different extraction tasks. Section 5.1 describes the data set used during testing. Section 5.2 presents various measurements of how accurately the system performs on the selected data set. Section 5.3 discusses the computational efficiency of the system.

5.1 Data Set

The data set used for evaluation is listed in this section. It is important to note that there are many other data sets available for evaluation purposes. However, I chose this data set according to the following criteria: amount of courses available, copyright restrictions, quality of learning resources, and course structure.

5.1.1 MIT OpenCourseWare

With the intention to expand the description given in Section 3.1 and having in mind the evaluation process, I present precise figures about the MIT OCW repository. This data set is one of the most important and relevant OCW repositories, since it contains 1250 courses (as at December 2005), and more than 6250 HTML documents. There are 32 disciplines. The same course can belong to different disciplines. The total size of the data set is approximately 5 Gb and only some courses can be downloaded from its website. Courses contain different structures or parts, and this in itself constituted one of the challenges in this project. They also display details to be extracted by my tool in different formats (plain text or paragraphs, HTML meta tags, tables etc). This characteristic makes my goal even more difficult since the algorithms I have developed need to be aware of this situation. The resources attached to them are in several formats as well, (such as PPT, PDF, video and audio files) and they are found in different parts of each course. Sometimes, some of these resources are located outside the data set (e.g. at a different website).
5.2 Preliminary Testing and Evaluation

As a preliminary evaluation of the system, I started with the output it produced. This will allow for the determination of how well the system reshaped OCW courses according to the implemented schemas in this project. The methodology followed at this stage was to test the OCWise produced IMS CP compliant courses by importing them into different Learning Object Repository systems. The LOR systems used were: WebCT, LORS (.LRN), Melete (Sakai) and Reload Editor / Player (standalone application). Half of the total number of courses (1250) were chosen in a random way, including courses from each discipline (32 disciplines in total) and uploaded into all four LORs.

This preliminary evaluation allowed me to determine:

a) **Standard adoption**

This test allows determining how well my implementation performed in regards to structuring courses by combining three different standards together. I found that the tool was able to deploy three e-Learning standards (IMS-CP, IEEE LOM and IMS LD) and the output generated by this tool was 100% standard compliant according to the testing done using the schemas provided by IMS website.

b) **How much data could be extracted from each course?**

Regarding this point, the tool could extract much of the data needed according to the aims of this project. However, the parser found it difficult to cope with data that was in a tabular format—being almost impossible to retrieve. One of the main reasons for this is that the tables could be ill-formed, perhaps because they were created with non-specific software. This constitutes one of the limitations of my system.

c) **How well does the extracted data match the original data?**

I also compared the original courses with the ones wrapped by OCWise and looked for differences between them. This test is expanded in the next chapter.

d) **Missing resources**

As described in Chapter 2, OCW courses contain different type of resources such as: PDF, PPT, video and audio files. My system successfully included
all resources in the final wrapped course as specified and required by the IMS CP specification.

e) **How many courses could not be uploaded successfully and why?**

I discovered that some courses contain invalid characters which accounted for the inability to upload them successfully with OCWise. This problem was found in a few courses only.

f) **Test different features of the LORs.**

This last test of the features of the LORs such as: search, update, and export among others, is very useful for people interested in getting/buying or adopting a LOR system as well as for developers, since educational institutions or developers could benchmark them and evaluate their features. I looked for importing and exporting functions, ways of updating course’s content as well as its metadata.

![Figure 24. Process of importing ‘reshaped’ OCW courses](image)

Figure 24 depicts the result of importing OCW courses into a Learning Management System (.LRN LORS) and an editing tool (Reload). Both of these present course details and metadata such as: title, keywords, copyright information as well as course metadata.
Evaluating the quality of the output produced by OCWise

As an example of how the processed learning objects can be used as data for benchmarking software applications, I show the evaluation results for the produced packages imported into different learning object repositories. We examined how well each LOR system dealt with the different imported packages and to what extent these LORs ‘respected’ (or adopted) the IMS Content Packaging specification.

The following Learning Management Systems (LMSs) were used to carry out testing:

a) dotLRN / LORS
b) Sakai / Melete
c) WebCT Campus Edition
d) Reload Editor / Player

Table 5 shows these results.

<table>
<thead>
<tr>
<th>Learning Management Systems (LMSs)</th>
<th>.LRN / LORS</th>
<th>Sakai / Melete v2.1.0</th>
<th>WebCT v4.1</th>
<th>Reload (Editor / Player) v2.0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSS</td>
<td>✓</td>
<td>N/A</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>JavaScript</td>
<td>✓</td>
<td>N/A</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Links</td>
<td>✓</td>
<td>N/A</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Metadata</td>
<td>✓</td>
<td>N/A</td>
<td>N/A</td>
<td>✓</td>
</tr>
<tr>
<td>Size (MB)</td>
<td>N/A</td>
<td>20</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Search ability</td>
<td>✓</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Standard adoption</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

Table 5. Evaluation results

All LORs imported IMS CP packages successfully. However, when it came to presenting the course to the learner, some issues arose. For example, Melete did not handle CSS and JavaScript correctly and all links were broken. The maximum size of the course could not exceed 20 Mb. Some LORs didn’t show the metadata section nor allowed it to be modified (Melete and WebCT). This issue had a negative impact since courses could not be found by their search engines or facilities. Overall,
the percentage of standard adoption was very high (95%), facilitating searchability, interoperability and reusability.

5.3 Quality of Extraction

The MIT OCW content is an important data set for assessing the effectiveness of my system, because it represents a real-world source of legacy learning resources in the educational domain.

5.3.1 Efficiency

In order to evaluate the quality of the results generated by the OCWise application, extraction experiments were carried out on the selected data set described in Section 5.1 using the measures defined in Section 2.6.

The system was run on 6250 HTML documents (1250 courses, 5 HTML pages each) to evaluate the average running time on a single document. The average file size was 4 Mb. The average running time per document was 2 seconds.

5.3.2 Accuracy

The accuracy of the OCWise was evaluated by counting the number of incorrect or missing data in the converted XML file by manually inspecting it against the HTML file. The average number of errors in each XML document was 0.13 and the accuracy was 87%.

5.3.3 Importing Errors

I found that 21 courses out of 1250 (1.68%) could not be imported successfully due to unrecognized characters either for description/keywords metadata fields.

For example, Advanced workshop in writing, 8-01Physics-IFall2003, 18-238Fall2002 (Maths) among others were found to have the above problem.
5.4 Experimental Results

Table 6 shows the results obtained from the data set.

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course Code</td>
<td>99.9</td>
<td>92.5</td>
</tr>
<tr>
<td>Course Title</td>
<td>98.2</td>
<td>82.2</td>
</tr>
<tr>
<td>Course Authors</td>
<td>97.1</td>
<td>98.0</td>
</tr>
<tr>
<td>Discipline</td>
<td>99.9</td>
<td>92.5</td>
</tr>
<tr>
<td>Keywords</td>
<td>99.9</td>
<td>98.5</td>
</tr>
<tr>
<td>Description</td>
<td>96.7</td>
<td>93.4</td>
</tr>
<tr>
<td><strong>F1 Average</strong></td>
<td></td>
<td><strong>92.8</strong></td>
</tr>
</tbody>
</table>

Table 6. Extraction results for course details

From the above table, can be drawn the conclusion that some course details such as: course code, discipline and keywords could be extracted successfully in all cases. However, some other fields such as: course authors and description presented some difficulties when it came to extracting them. The most common error in the course author field was that they contained numbers and non-valid characters. Our results are in the line with some other results found in the literature [61].

As stated previously in this chapter, our proposed approach offers high performance and reliability. One of the relevant measures to further discuss is the system accuracy. OCWise generates XML compliant files as result of a conversion process. In an attempt to measure accuracy, we compare HTML files against XML files. We found out that, during the extraction process, a few fields presented difficulties to be extracted successfully. That might affect the overall system performance. Other issue we came across relates to tables. Generally speaking, most of the open repositories websites were created by an automated process and made available on the Internet. Bearing that in mind, many web pages are ill-formatted which causes our extraction layer to be unable to process specific fields and/or tables (eg, course authors and grading table). We found out invalid characters and/or numbers in the course author field. Regarding different tables containing information
about courses, we spotted missing and/or nested tags which confuses our parser. In an attempt to overcome the above difficulties, OCWise extraction layer is able to automatically trap errors in any to be-processed page. OCWise generates a log file indicating the name of the ill-formed page and the line(s) where the error(s) were found. We can improve our system accuracy by increasing the number of fields processed successfully and implementing a recovery scheme to tackle the problems described above. In addition to that, the extraction layer can be improved with enhanced regular expressions and navigation capabilities to capture finer granularity data.
6 Conclusions

This study aimed at determining how freely available learning resources such as those in MIT’s OCW could be reused by transforming them to standard compliant format. I wanted to address the reuse of them as 1) content and 2) learning designs that include content and activities.

I have presented a tool called OCWise to solve the problem of structuring course content from HTML documents into a repository of homogeneous XML documents. The three components of this tool, web crawler, parser and wrapper, were described in detail in Chapter 3. Some preliminary tests were conducted, some empirical results were presented and comparisons with existing approaches were discussed.

Most existing Learning Management Systems do not provide a way of converting of learning materials found in their repositories into a standard compliant format. This process cannot be done manually mainly because it is a time consuming and expensive process. This process also requires human intervention to be successful. I effectively have overcome this challenge by implementing novel techniques and educational standards and producing a tool capable of solving that problem.

The tool developed has successfully addressed the issue of the interoperability of learning content, as described in Chapter 4—the reshaped courses were successfully uploaded into four different Learning Management Systems, modified and then exported for further use in the same or different LMS. Once they were uploaded, those courses could be discovered by searching facilities/engines, metadata could be updated, course content could be reused and repurposed according to different course author’s needs.

As a part of the evaluation process, OCWise performance was measured according to common performance measures in the field of Information Extraction. OCWise performance was similar to other developments.

The output generated by this tool possesses multiple applications. These applications can be grouped on an educational and/or technological basis. On the technical side, this collection of learning objects can be used by learning objects repositories developers for testing and benchmarking purposes. They can be tested as
to what extent their implementation is standard compliant. They can also test search capabilities offered by their implementation. Course content editing and exporting processes can also be tested. They can also perform a ‘bulk’ uploading and measure how many courses can be uploaded in a certain time.

Students are also potential users of these learning objects for they may be interested in knowing what is taught at different universities and at the same time they can complement or supplement their studies. They can also search for courses in different languages, course level, and so forth.

This set of learning objects can be of interest to researchers in other disciplines such as Artificial Intelligence, and digital libraries among others. They can use these educational materials as data sets for their own projects and produce useful results or statistics.

Educational institutions may also be interested in this system because they can use it to convert/transform their courses into standard compliant formats and thus benefit.

The second objective was more challenging—integrating the content of OCW into learning activities through an LMS.

The Learning design that a lecturer implicitly describes in the materials is not easy to extract automatically. Each teacher describes it in such different ways that manually customising an information extraction layer for each of them would not be feasible. New approaches for the automatic generation of IMS Learning Design (IMS-LD) content will be required. This specification gives course authors the possibility to build a pedagogically neutral set of blocks, which enable many different pedagogies to be expressed and delivered online.

I hope that this work will contribute to simplifying the complex process of moving face-to-face education to an online scenario in a systematic and efficient way in the light of new developments and research in the related fields.
Future work

In this section, we describe some of the possible extensions to OCWise.

On the technical side, we will focus on improving (robustness) some of the rules used by the Extraction layer as well as investigating the concept of Web Services since it offers a higher level interface to web data sources. In addition to that and in a bid to improve performance, we will look into some optimization techniques and recovery schemes.

There are several interesting directions for future works. One major direction of future research is the investigation of reusability of Learning Designs. This will open new opportunities to universities and companies (staff training, certifications), seeking enhancing distance education and training. Another interesting future direction relates to the integration of different IE systems, since the output generated from one system can be the input to another and they all share data in XML format.
# Glossary

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBL</td>
<td>Competency Based Learning</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheet</td>
</tr>
<tr>
<td>DOM</td>
<td>Document Object Model</td>
</tr>
<tr>
<td>DTD</td>
<td>Document Type Definition</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language for displaying text, images, links etc on web pages.</td>
</tr>
<tr>
<td>IEEE LOM</td>
<td>IEEE Learning Object Metadata</td>
</tr>
<tr>
<td>IMS</td>
<td>Global Learning Consortium</td>
</tr>
<tr>
<td>IMS CP</td>
<td>IMS Content Packaging</td>
</tr>
<tr>
<td>IMS LD</td>
<td>IMS Learning Design</td>
</tr>
<tr>
<td>LMS</td>
<td>Learning Management System</td>
</tr>
<tr>
<td>LORS</td>
<td>Learning Object Repository System</td>
</tr>
<tr>
<td>OCWise</td>
<td>Open CourseWare information and structure extractor</td>
</tr>
<tr>
<td>PIF</td>
<td>Package Interchange File</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modelling Language</td>
</tr>
<tr>
<td>XHTML</td>
<td>Extensible HyperText Markup Language</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language similar to HTML</td>
</tr>
<tr>
<td>XPATH</td>
<td>XML Path Language</td>
</tr>
<tr>
<td>XSD</td>
<td>XML Schema Definition</td>
</tr>
<tr>
<td>XSLT</td>
<td>Extensible Stylesheet Language Transformation</td>
</tr>
<tr>
<td>PDF</td>
<td>Portable Document File</td>
</tr>
<tr>
<td>PPT</td>
<td>PowerPoint</td>
</tr>
<tr>
<td>Word</td>
<td>Microsoft Word</td>
</tr>
<tr>
<td>SAX</td>
<td>Simple API for XML</td>
</tr>
<tr>
<td>PERL</td>
<td>Programming language</td>
</tr>
</tbody>
</table>
References


