

STATUS OF THESIS

Title of thesis

**SWA-KMDLS : An Enhanced e-Learning Management System Using
Semantic Web and Knowledge Management Technology**

I **AHMAD MUKHLASON**

hereby allow my thesis to be placed at the Information Resource Center(IRC) of
Universiti Teknologi PETRONAS(UTP) with the following conditions:

- 1. The thesis becomes the property of UTP.
- 2. The IRC of UTP may take copies of the thesis for academic purposes only.
- 3. The thesis is classified as:

Confidential

Non-confidential

If this thesis is confidential, please state the reason:

.....-

The contents of the thesis will remain confidential for-..... years.

Remark on disclosure:

.....-

Endorsed by

AHMAD MUKHLASON
RT 01 RW 05 Wringinpitu,
Plampangrejo Cluring Banyuwangi,
Jawa Timur
INDONESIA 68482

DR AHMAD KAMIL MAHMOOD
Department of Computer and Information Sciences
Universiti Teknologi PETRONAS, Seri Iskandar,
Tronoh, Perak MALAYSIA 31750

Date: _____

Date: _____

UNIVERSITI TEKNOLOGI PETRONAS

Approval by Supervisor (s)

The undersigned certify that they have read, and recommend to The Postgraduate Studies Program for acceptance, a thesis entitled “**SWA-KMDLS : An Enhanced e-Learning Management System Using Semantic Web and Knowledge Management Technology**” submitted by **Ahmad Mukhlason** for fulfillment of the requirements for the degree of Master of Science.

Date

Signature : _____

Main Supervisor : Dr Ahmad Kamil Mahmood

Date : _____

Co-Supervisor 1 : Ms Nooren Izza Arshad

Co-Supervisor 2 : Mr Ahmad Izuddin Zainal Abidin

TITLE PAGE

UNIVERSITI TEKNOLOGI PETRONAS

SWA-KMDLS: An Enhanced e-Learning Management System
Using Semantic Web and Knowledge Management Technology

By
Ahmad Mukhlason

A THESIS
SUBMITTED TO POSTGRADUATE STUDIES PROGRAM
AS REQUIREMENT FOR THE
DEGREE OF MASTER OF SCIENCE
IN INFORMATION TECHNOLOGY
BANDAR SERI ISKANDAR
PERAK
MARCH, 2009

DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Signature : _____

Name : AHMAD MUKHLASON

Date : _____

DEDICATION

This thesis is dedicated to my parent who always tirelessly support and pray for me. For my sisters Ima, and Iva. For all my teachers anywhere. On every grudge and every fight you are my everlasting inspiration and I miss you all day and night. There is light on every night, there is hope on every fright. I hope this thesis gives a contribution for a better life.

ACKNOWLEDGEMENTS

I would like to thank Allah subhanahu wata'alaah, Muhammad shallahu 'alaihi wasallam the most inspiration of my life. This work could not have been possible without the advice and support of so many people. First and foremost, I Would like to thank my research supervisors the honorable Dr Ahmad Kamil Mahmood, Ms Nooren Izza Arshad, and Mr. Izuddin Zainal Abidin for their nice and full-dedicated guidance during conducting my research project, Universiti Teknologi PETRONAS for providing well-equipped facilities to conduct this research.

I am also deeply grateful to my parents, Mujiyono and Rodliyah, whose support went well beyond kindly words of encouragement and finally become for me an affirmation that I really did have something worthwhile to contribute for others. I am deeply thankfull to my family and friends who support me to pursue my master studies. For my soul mate in Indonesia who always support and pray me every time, my 'brother' and 'sister' in Indonesia who always keep in touch and motivate me even though we are so far each other: Andri, Arbi, Fajar, Ibnu, Azis, Iyan, Mahendra, Ndop, Dhidy, Ley, Nur laili, etc and my friends (room mate : "Isal"; housemate: Kiki, Candra, Pak Jodi, Bang Surya, Rizal, Aryo, Fikri, Dancom, Arif –Bojon, Teguh-Telor, etc.; course mate: Pak Arif, Pak Bayu, Bang Oqi, Aini, Pak TBA, Pak Dani, Mbak Lail, Intan, Tito, Inung, Yunita, Aidel, etc.; country mate –PPI UTP: epul, bastian, ihtatho, etc;) who make my life in UTP more alive. PGO Staff (Pn. Kamaliah, Kak Aida, Kak Nurul, Cik Burhan, Cik Kahar), Cik Asfu thanks a lot. Shaqifah and Khail Malik thank for translating my abstrack into Bahasa Malaysia and to everyone who has given contribution to this thesis directly or indirectly I really appreciate it.

ABSTRACT

In this era of knowledge economy in which knowledge have become the most precious resource, surveys have shown that e-Learning has been on the increasing trend in various organizations including, among others, education and corporate. The use of e-Learning is not only aim to acquire knowledge but also to maintain competitiveness and advantages for individuals or organizations. However, the early promise of e-Learning has yet to be fully realized, as it has been no more than a handout being published online, coupled with simple multiple-choice quizzes. The emerging of e-Learning 2.0 that is empowered by Web 2.0 technology still hardly overcome common problem such as *information overload* and *poor content aggregation* in a highly increasing number of learning objects in an e-Learning Management System (LMS) environment.

The aim of this research study is to exploit the Semantic Web (SW) and Knowledge Management (KM) technology; the two emerging and promising technology to enhance the existing LMS. The proposed system is named as *Semantic Web Aware-Knowledge Management Driven e-Learning System (SWA-KMDLS)*. An Ontology approach that is the *backbone* of SW and KM is introduced for managing knowledge especially from learning object and developing automated question answering system (Aguas) with *expert locator* in SWA-KMDLS. The METHONTOLOGY methodology is selected to develop the Ontology in this research work.

The potential of SW and KM technology is identified in this research finding which will benefit e-Learning developer to develop e-Learning system especially with *social constructivist* pedagogical approach from the point of view of KM framework and SW environment. The (semi-) automatic ontological knowledge base construction system (SAOKBCS) has contributed to knowledge extraction from learning object semi-automatically whilst the Aguas with *expert locator* has facilitated knowledge retrieval that encourages knowledge sharing in e-Learning environment.

The experiment conducted has shown that the SAOKBCS can extract concept that is the main component of Ontology from text learning object with precision of 86.67%, thus saving the expert time and effort to build Ontology manually. Additionally the experiment on Aquas has shown that more than 80% of users are satisfied with answers provided by the system. The expert locator framework can also improve the performance of Aquas in the future usage.

Keywords: semantic web aware – knowledge e-Learning Management System (SWA-KMDLS), semi-automatic ontological knowledge base construction system (SAOKBCS), automated question answering system (Aquas), Ontology, expert locator.

ABSTRAK

Dalam era k-ekonomi di mana ilmu telahpun menjadi sumber terpenting, kajian telah menunjukkan bahawa penggunaan sistem e-Learning telah meningkat dalam bidang pelajaran dan korporat. Penggunaan e-Learning bukan sahaja bertujuan untuk mendapatkan ilmu tetapi untuk mengekalkan daya saing sama ada untuk individu mahupun organisasi. Namun, potensi awal e-Learning masih belum dapat direalisasikan kerana ia hanya dianggap sebagai handout yang diterbit secara online, ditambah dengan kuiz. Walaupun e-Learning 2.0, yang dibantu teknologi Web 2.0, sudahpun wujud, masalah seperti information overload dan poor content aggregation masih tidak dapat ditangani dalam suasana e-Learning Management System atau LMS.

Tujuan kajian ini adalah untuk mengekstrak potensi Semantic Web (SW) and teknologi Knowledge Management (KM) dengan sepenuhnya. SW dan KM merupakan dua teknologi yang baru dikenalpasti dapat memperbaiki LMS yang sedia ada. Cadangan nama untuk sistem baru ini adalah Semantic Web Aware-Knowledge Management Driven e-Learning System (SWA-KMDLS). Pendekatan ontologi yang merupakan tulang belakang SW dan KM dikenalkan untuk mengurus pengetahuan terutamanya daripada objek pembelajaran dan pembangunan sistem penjawab soalan automatik atau dalam Bahasa Inggeris, *automated question answering system* (Aquas). Kaedah METHONTOLOGY dipilih untuk membangunkan ontology dalam kajian ini.

Potensi teknologi SW dan KM yang dikenal pasti dalam kajian ini dapat membantu pemaju e-Learning untuk mencipta sistem e-Learning yang menggunakan pendekatan pedagogi *constructivist* dari sudut pandangan KM dan SW. Sistem semi-automatic ontological knowledge base (atau SAOKBCS) telah membawa kepada pengekstrakan ilmu daripada objek pembelajaran secara separa-automatik manakala Aquas dengan expert locator dapat membantu pengekstrakan ilmu yang mengalakkan perkongsian ilmu dalam suasana e-Learning.

Eksperimen yang dijalankan menunjukkan bahawa SAOKBCS dapat mengekstrak konsep bahawa komponen utama ontology dari objek pembelajaran yang berasaskan teks

dapat beroperasi dengan ketepatan 86.67%. Hal ini dapat menjimatkan masa dan mengurangkan beban dalam proses membina ontology secara manual. Tambahan lagi, eksperimen yang dijalankan terhadap Aquas menunjukkan bahawa lebih dari 80% pengguna e-Learning berpuas hati terhadap jawapan yang diberi oleh sistem. Penggunaan *expert locator framework* dapat meningkatkan prestasi Aquas di masa depan.

TABLE OF CONTENTS

STATUS OF THESIS.....	i
TITLE PAGE.....	iii
DECLARATION.....	iv
DEDICATION.....	v
ACKNOWLEDGEMENTS.....	vi
ABSTRACT.....	vii
ABSTRAK.....	ix
TABLE OF CONTENTS.....	xi
LIST OF TABLES.....	xiv
LIST OF FIGURES.....	xv
LIST OF ABBREVIATION AND SYMBOL USED.....	xvii
LIST OF EQUATIONS.....	xix
CHAPTER 1 INTRODUCTION.....	1
1.1 Backgrounds.....	1
1.2 Problem Statement.....	3
1.3 Objectives and Scope.....	7
1.4 Thesis Contribution.....	7
1.5 Thesis Organization.....	8
CHAPTER 2 LITERATURE REVIEW.....	9
2.1 Emerging Problems and Requirements in e-Learning System.....	9
2.1.1 Emerging Requirement in e-Learning System.....	9
2.1.2 Semantic Web and e-Learning.....	11
2.1.3 Knowledge Management (KM) and e-Learning.....	15
2.1.4 Knowledge Management and <i>Social Constructivism</i> in e-Learning System	18
2.2 e-Learning, SW, and KM Where is their Meeting Point?.....	21
2.3 Summary.....	23
CHAPTER 3 METHODOLOGY.....	24
3.1 General Research Design.....	24
3.1.1 Development of Ontologies.....	24
3.1.2 Design the Framework and Architecture of SWA-KMDLS.....	26
3.1.3 Development and Implementation of SWA-KMDLS.....	26
3.1.4 Experimental testing and evaluation.....	26
3.1.5 The Experimental Validation.....	29
3.2 Comparative Studies of Ontology Development Methodology.....	30
3.3 METHONTOLOGY.....	32
3.3.1 Ontology Development Process.....	32
3.3.2 Ontology Life Cycle.....	34
3.3.3 Ontology Development Technique: Ontology Tool and Language.....	42
3.4 Summary.....	42
CHAPTER 4 DESIGN FRAMEWORK AND ARCHITECTURE OF SWA-KMDLS ..	44
4.1 Ontological Learning Object Management.....	44
4.1.1 SAOKBCS Framework.....	45
4.1.2 The Ontology Population.....	47

4.1.3	The Ontology Learning Module	47
4.1.4	The Ontology Mapping, Merging and Alignment	54
4.2	Automated Question Answering System (Aquas)	59
4.2.1	Statistical-based Automatic Question Answering System (S-Aquas)	60
4.2.2	Ontology-driven Automatic Question Answering System (Onto-Aquas)	63
4.3	Expertise Finding and Mapping Framework	65
4.3.1	Expert Locator Framework	65
4.3.2	Expert User Ontology	66
4.4	Summary	67
CHAPTER 5	DEVELOPMENT AND IMPLEMENTATION	69
5.1	The Implemented Environment: Moodle e-Learning Management System	69
5.1.1	Modular Approach in Moodle	69
5.1.2	Creating a New Module in Moodle	70
5.2	Initial Ontology Development (Ontology Population)	71
5.2.1	Initial Domain Ontology Development	73
5.2.3	Initial Application Ontology	75
5.2.3	Ontology Testing	77
5.3	SAOKBCS Implementation	79
5.3.1	SAOKBCS User Interface	79
5.3.2	SAOKBCS Algorithm and Implementation	81
5.4	Aquas Implementation	82
5.4.1	Aquas User Interface	83
5.4.2	Aquas Algorithm and Implementation	85
5.5	Expertise Finding and Mapping Framework Implementation	90
5.6	Summary	92
CHAPTER 6	THE EXPERIMENT, EVALUATION, AND DISCUSSION	93
6.1	The Experiments	93
6.1.1	Experimental Environment	93
6.1.2	Experimental Scenarios	94
6.1.3	Experimental Result	94
6.2	Evaluation and Discussion	101
6.3	Summary	103
CHAPTER 7	CONCLUSION	105
7.1	Addressing the Research Objectives	105
7.2	Contributions	107
7.3	Limitations, Challenges and Open Issues	108
7.4	Future Works	109
REFERENCES	111
PUBLICATIONS	123
APPENDIX I	E-LEARNING	125
I.1	Definition and Scope of e-Learning	125
I.2	e-Learning Standards and Pedagogical Approaches	128
I.3	e-Learning Tools and Technologies: State Of The Art	130
I.4	e-Learning Technology Trend	133
APPENDIX II	SEMANTIC WEB	135
II.1	The Fundamental of SW	135

II.1.1 SW Definition	135
II.1.2 The SW Vision	136
II.1.3 The SW Goals	138
II.1.4 The SW Component	138
II.2 Ontology	140
II.2.1 Definition and Scope	140
II.2.2 Classification of Ontology	143
II.2.3 Ontology Representation Languages	145
II.3 Tool For SW	148
II.3.1 Ontology editors	148
II.3.2 Ontology Management System (OMS)	149
II.3.3 Annotation Tools	150
II.3.4 Visualization Tools	150
II.3.5 Ontology Search Engines	151
II.3.6 Semantic Browser	151
APPENDIX III KNOWLEDGE MANAGEMENT	153
III.1 Fundamental of KM	153
III.2 KM Tools and Techniques	158
III.2.1 Content Management System (CMS)	158
III.2.2 Knowledge Taxonomies	159
III.2.3 Groupware	159
III.2.4 Online Communities of Practice (CoP)	159
III.2.5 Portals	159
III.2.6 Social Network Analysis (SNA) and Design	160
III.2.7 e-Learning	160
III.2.8 Storytelling and Narratives	160
III.2.9 Wireless Platform for Knowledge Mobilization	160
III.2.10 Innovation Management Tools	160
III.2.11 Inter-Organizational Knowledge Sharing Platforms	161
III.3 Knowledge Management System (KMS)	161
APPENDIX IV DOMAIN ONTOLOGY	164
IV.1 Domain Ontology in Class Tree	164
IV.2 Domain Ontology in Domain/Range Relation	166
IV.3 Domain Ontology in OWL (Web Ontology Language)	167
APPENDIX V APPLICATION ONTOLOGY	194
V.1 Application Ontology in Class Tree	194
V.2 Application Ontology in Domain/Range Relation	196
V.3 Application Ontology in OWL (Web Ontology Language)	197

LIST OF TABLES

Table 2- 1 Differences between Traditional Learning (Training) and e-Learning	10
Table 2- 2 Benefits of using Semantic Web as a technology for e-Learning	12
Table 2- 3 Prior Work On Investigating Semantic Web for e-Learning.....	13
Table 2- 4 Relationship between KM enablers/techniques and KM Processes in e-Learning	16
Table 3- 1 The SAOKBCS experimental scenarios.....	26
Table 3- 2 The Aquas Experimental Scenarios.....	28
Table 3- 4 Query Result Classification.....	29
Table 3- 3 Comparative Studies of Ontology Development Methodologies	31
Table 4- 1 Ontology Learning Task based on Ontology Artifact	50
Table 4- 2 The Text2Onto Ontology Primitive Model & Gruber Frame Ontology	53
Table 4- 3 Algorithm For Primitive Model Extraction.....	53
Table 5- 1 Compulsory Files in Each Module in Moodle	70
Table 6- 1 Experiment on SAOKBCS Result.....	95
Table 6- 2 ANR of Ontology Learning Result.....	96
Table 6- 3 Experimental result in finding best-fit answer	98
Table 6- 4 Experimental result in finding most-relevant learning material.....	99
Table 6- 5 Precision, Recall, and F Measure of SAOKBCS	101
Table I- 1 e-Learning Categorization.....	127
Table I- 2 Pedagogical Approaches of e-Learning	129
Table I- 3 LMS/CMS Features	131
Table II- 1 The OWL Language Level	148

LIST OF FIGURES

Figure 1- 1 Research Area	3
Figure 2- 1 e-Learning and KM Integration based on Enterprise Content and Knowledge	18
Figure 2- 2 e-Learning, Semantic Web, KM: the Meeting Point.....	23
Figure 3- 1 General Research Flow Design.....	25
Figure 3- 2 Ontology Learning Algorithm.....	27
Figure 3- 3 METHONTOLOGY: Ontology Development Process	32
Figure 3- 4 METHONTOLOGY: Ontology Development Process Life Cycle	36
Figure 3- 5 Methodology, Tool, and Language for Ontology Development.....	42
Figure 4- 1 Ontology Approach for Learning Objects Management.....	45
Figure 4- 2 SAOKBCS Framework.....	46
Figure 4- 3 Prototype LO's Context Ontology	48
Figure 4- 4 Ontology Learning Layer Cake.....	49
Figure 4- 5 Text2Onto Architecture	51
Figure 4- 6 Ontology Mapping, Merging, and Alignment.....	54
Figure 4- 7 Context Ontology Merging Case Scenario	56
Figure 4- 8 PROMPT Algorithm	57
Figure 4- 9 Proposed Aquas for Knowledge Reusing and Sharing Framework in e- Learning Environment	59
Figure 4- 10 S-Aquas Framework.....	62
Figure 4- 11 Onto-Aquas Framework.....	63
Figure 4- 12 Expert Locator Architecture.....	65
Figure 4- 13 Expert User Ontology	67
Figure 5- 1 Moodle LMS Front Page.....	70
Figure 5- 2 Creating a New Module in Moodle.....	71
Figure 5- 3 Protégé –OWL 4.0 Screenshot.....	72
Figure 5- 4 Graphical Representation of Domain Ontology in Class Hierarchy Diagram	74
Figure 5- 5 Graphical Representation of Domain Ontology in Domain-Relation-Range Diagram.....	75
Figure 5- 6 Graphical Representation of Application Ontology in Class Hierarchy Diagram.....	76
Figure 5- 7 Execute FaCT++ Reasoning Tool in Protégé	77
Figure 5- 8 Racer Reasoning Tool.....	78
Figure 5- 9 Classification and Consistency Testing Using Racer.....	78
Figure 5- 10 Add New Learning Resource in IPSP course	80
Figure 5- 11 Uploading New Learning Resources	81
Figure 5- 12 Ontologies Construction Pseudocode	82
Figure 5- 13 User interface Aquas in moodle.....	83
Figure 5- 14 User interface AQUAS – entering question.....	84
Figure 5- 15 User interface AQUAS – respond from system.....	85
Figure 5- 16 Pseudocode for Aquas.....	86
Figure 5- 17 The Pseudocode of S-Aquas	87

Figure 5- 18 TFxIDF Pseudocode.....	88
Figure 5- 19 Onto-Aquas Pseudocode	88
Figure 5- 20 Ontology Similarity Algorithm Pseudocode.....	89
Figure 5- 21 SPARQL Query For expert User	90
Figure 5- 22 User interface Aquas – Question alert for expert user	91
Figure 5- 23 User interface Aquas – User expert answering question	91
Figure 6- 1 Graphics of Experimental SAOKBCS Ontology Learning Result	96
Figure 6- 2 Chart of experimental result in finding best-fit answer	98
Figure 6- 3 Comparison chart of experimental result in finding best-fit answer.....	99
Figure 6- 4 Chart of experimental result in finding the most relevant material	100
Figure 6- 5 Comparison chart of experimental result in finding the most relevant material	100
Figure 6- 6 Comparison of Precision, Recall, and F Measure for SAOKBCS.....	102
Figure I- 1 Evolution of E-learning Technology	130
Figure I- 2 Evolution of E-learning Content	133
Figure II- 1 Tim Berner Lee, SW Vision	137
Figure II- 2 The Semantic Web Layer Cake	138
Figure II- 3 Ontologies and Their Relatives	141
Figure II- 4 Graphical Ontology Example: Human Resource	142
Figure III- 1 KM Life Cycle	155
Figure III- 2 Characteristic of KMS	162
Figure III- 3 Software Use for KMS in Companies.....	163

LIST OF ABBREVIATION AND SYMBOL USED

Aquas	Automated question answering system.
§	Section.
CIO	Chief Information Officer.
CKO	Chief Knowledge Officer.
CMS	Content Management System.
CoP	Community Of Practice.
CV	Characteristic Vector.
ED	Euclidian Distance.
FAQ	Frequently Asked Question.
GATE	General Architecture For Text Engineering.
HTML	Hyper Text Markup Language.
HTTP	Hyper Text Transfer Protocol.
IP	Inner Product.
IR	Information Retrieval.
ISO	International Organization for Standardization.
KM	Knowledge Management.
KMS	Knowledge Management System.
LCMS	Learning Content Management System.
LMS	e-Learning Management System.
LO	Learning Object.
LOM	Learning Object Metadata.
LOR	Learning Object Repository.
OMS	Ontology Management System.
Onto-Aquas	Ontology-driven automatic question answering system.
OWL	Web Ontology Language.
RDF(S)	Resource description framework (Schema).
RQL	RDF Query Language.
SAOKBCS	(Semi-)Automatic Ontological Knowledge Base Construction System.

S-Aquas	Statistical-based automatic question answering system.
SNA	Social Network Analysis.
SWA-KMDLS	Semantic Web Aware – Knowledge Management Driven e-Learning System.
TM	Topic Map.
URI/L	Uniform Resource Identifier/Locator.
WWW(C)	World Wide Web (Consortium).
XML	Extensible Markup Language.

LIST OF EQUATIONS

$P = \frac{tp}{(tp + fp)}$	Eq. 3- 1	29
$R = \frac{tp}{(tp + fn)}$	Eq. 3- 2	29
$F = \frac{2PR}{(P + R)}$	Eq. 3- 3	29
$R_{i,q} = \sum_{termj \in q} \left(0.5 + 0.5 \frac{TF_{i,j}}{TF_{i,max}} \right) IDF_j$	Eq.4- 1	60
$IDF_j = \log \left(\frac{N}{\sum_{i=1}^N C_{i,j}} \right)$	Eq.4- 2	60
$D_i = \sum_{k=1}^n W_{kx} W_{ik}$	Eq.4- 3	61
$Di = \sqrt{\sum_{k=1,n} (W_k - W_{ik})^2}$	Eq.4- 4	61
$W_{ij} = \frac{f_{ij}}{\sum_{k=1}^m f_{kj}}$	Eq.4- 5	61
$ANR = \left(\frac{T}{T + F + D} \right) \times 100$	Eq. 6- 1	95

CHAPTER 1

INTRODUCTION

1.1 Backgrounds

In this era of knowledge economy in which knowledge become the most precious resource, e-Learning makes more sense than ever before as a media not only to acquire knowledge but also maintain competitiveness and advantages for individuals or organizations. e-Learning has become a trend that has also been discussed in the same breath with e-business and e-commerce area. A recent study by International Data Corporation (IDC) reported that the worldwide e-Learning market reached \$15.9 billion in 2007 and was predicted to continue growing (Anderson 2008). The American Society for Training and Development (ASTD) estimated that U.S. organizations spent \$134.39 billion on employee learning and development in 2007 (Paradise 2008). In corporate sectors such as banking, finance, and insurance, e-Learning is gradually upstaging conventional training methods to gain benefits of e-Learning which offers cost saving and employee productivity.

However, the early promise of e-Learning has yet to be fully realized. e-Learning, for many years, has been no more than a handout which is published online, coupled with simple multiple-choice quizzes (Downes 2007). Users' feedbacks are rather disappointing and e-Learning investments lack of intangible result in organizations (Brodsky 2008).

The Semantic Web (SW) promises advancement in the intelligent e-Learning systems design and development by enabling more appropriate representation for the learners, in term of learning goals, learning materials, contexts of usage, as well as more efficient access and navigation to the learning resources. The SW and SW technologies offer a new approach to managing information and processes; the fundamental principle of SW is the creation and use of semantic metadata. At its simplest, the results can be clustered

by meaning instead of a search providing a linear list of results. The use of semantic metadata is also crucial in integrating information from heterogeneous sources, within as well as across organization(Davies et al. 2006).

Knowledge Management (KM), on the other hand, has been developed within the business context. The recent interest in organizational knowledge has prompted the use of KM in order to process and manage the knowledge to the organization's benefit (Alavi and Leidner 2001). KM outcomes fall into three categories: knowledge creation, retention, and transfer (Argote et al. 2003). Knowledge creation occurs when new knowledge is generated in organization. Knowledge retention involves embedding knowledge in a repository so that it exhibits some persistence over time. Knowledge transfer is evident when experience acquired in one unit affects another. These three categories are closely related.

These two new promising approaches, namely SW and KM have driven the interest of implementing them both in an e-Learning system. The integration of e-Learning, SW, and KM proposed to enable the creation of great synergies in organizations and business applications. Thus, the context of this dissertation is the amalgamation of e-Learning, SW, and KM as depicted in Figure 1-1. The SW and KM shall be exploited to improve the legacy e-Learning system in which e-Learning is viewed separately from KM and developed in SW-unaware environment toward SW-aware KM driven e-Learning management system (SWA-KMDLS).

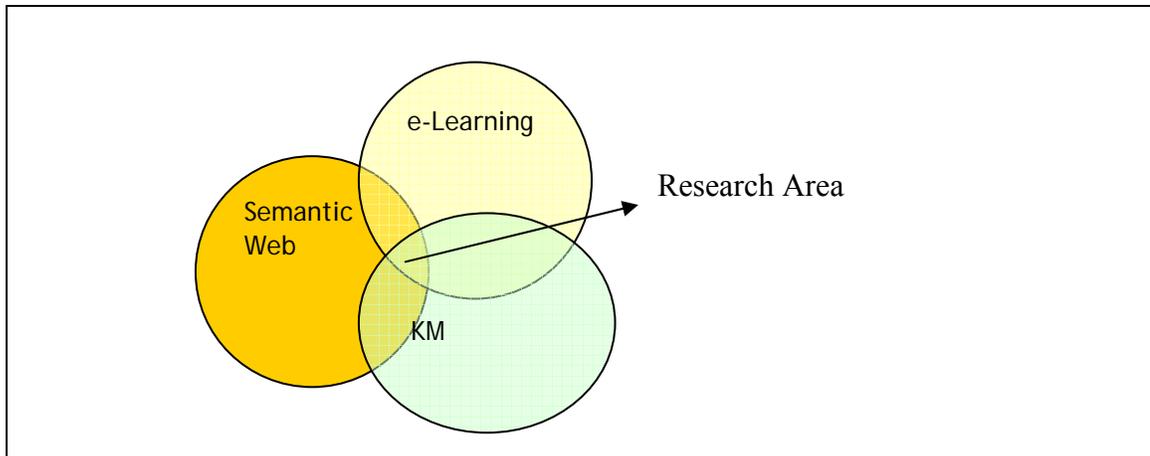


Figure 1- 1 **Research Area**

1.2 Problem Statement

This research is motivated by the experience of the author as a tutor for programming class at Universiti Teknologi PETRONAS. Although this class has used e-Learning system as supporting media for the actual class, the e-Learning is only used for uploading the learning material presented in class, assignment materials, and assignment mark or test score for the class. Furthermore, during the semester, students continued to question related to the programming class material to the tutor either by messenger or by personal contact. The problem was in many cases students asked the trivial questions that had been presented in learning material or question that had been asked by other students. Indeed, it was a tedious work to answer those kind of questions and this drives the author to imagine an improved e-Learning system to overcome this problem, a new system that can answer the students' question automatically from the latent knowledge in the e-Learning system.

Since e-Learning is yet to be mature in implementation, it faces many problems to be an effective and efficient tool for learning, yet lifelong learning. It will still evolve in the future. Effective and efficient e-Learning methods are generally required to ensure that learners are equipped with the latest knowledge in timely manner. The prior research by

Piccoli et al. identified two major components affected e-Learning effectiveness in virtual learning environment: the human dimension and the design dimension (Piccoli et al. 2001). The technology in this framework is geared toward providing effective e-Learning. For example, content management is one of the factors in the design dimension, which includes factual knowledge, procedural knowledge, and conceptual knowledge that has a positive relationship with e-Learning effectiveness. The design of interaction enhances the knowledge sharing among learners, and between learners and the instructor. This thesis stressed on design and technology dimension, particularly it will highlight on the problems which deals with organizing learning object and knowledge retrieval or reuse in e-Learning.

(i) Organizing Learning Objects

The increasing number and use of learning objects (LO) in e-Learning system make LOs become increasingly valuable and, at the same time, the management of LOs repository becomes complicated. The LOs that are not semantically inter-related one to another caused the problem in LO retrieval more complicated.

The previous work to overcome this problem is by creating learning object metadata (LOM). LOM is a data model, usually encoded in XML, used to describe a learning object and similar digital resources used to support learning. The purposes of LOM initiative are the creation of well structured descriptions of learning resources that will help in facilitating the discovery, location, evaluation and acquisition of learning resources by students, teachers or automated software processes in e-Learning environment such as that (Barker 2005):

- Sharing of learning resource descriptions between resource discovery systems. This should lead to a reduction in the cost of providing services based on high quality resource descriptions.
- Tailoring of the resource descriptions to suit the specialized needs of a community. This may include choosing suitable controlled vocabularies for classification, reducing the number of elements that are described or adding new ones from other resource description schemas.

- Creators and publishers may use the LOM along with other specifications to “tag” learning resources with a description that can be associated with the resource. This will provide information in a standard format similar to that found on the cover and fly-page of a textbook.

There have been metadata standards for learning objects, such as Dublin Core (Core 2008), IEEE Learning Technology Standard Committee (LTSC) (IEEE 2008), and IMS global learning consortium (IMS 2008). These standards are used to represent individual LOs at the collection level, which is similar to library catalogue systems. However, to use LOs to support teaching and learning at the knowledge sharing level for a specific field; knowledge schema must be applied to the LOs repository for the domain. This is because LOs can be organized in a variety of ways depending upon complex intra-context and inter-context.

When a virtual LOs repository is huge and distributed on the Internet, the use of meta-data and keywords only to search the needed LOs is inefficient and ineffective since much potential associations with various learning aspects are bypassed (Wang 2008). Furthermore, although LOM enables the indexing, location management and searching of learning objects in a LO repository by extended sharing and searching features. However, LOM has a deficiency in semantic-awareness capability (Lee et al. 2006). This has led to approaches to SW applications that model the relationships between learning objects using formal Ontologies.

(ii) Knowledge Retrieval

Since the huge increasing number of LO in e-Learning environment, users face difficulty to find the exact LO that they really need. This situation usually termed as “Information Overload” or “Information Glut”. To cope with this condition there is a need for a system that intelligently understand what the user need and find the LO that matched with the users’ need.

The experiences of e-Learning practice particularly in the university where the researcher is, showed that the question students ask to their lecturer is the very common. The problem is usually student tends to ask the same question which previously has been asked by another student and has been answered by the lecturer. It is a tedious work for lecturer to re answer the same question a “reinventing the wheel” activity. Therefore, there is a need to cope with this kind of problem by providing an automatic question answering system that can answer the question that previously has been asked by student and has been answered by lecturer.

In Workshop On Learning Object Repository as Digital Libraries (Harris and Thom 2006) has identified three major barriers in the retrieval and reuse of LOs: organizational, cultural and technical. Organizational barriers are those that exist because of the structure of institutions. Cultural barriers exist because of the experiences or attitudes of academic staff in terms of the contribution of material to a LO repository (LOR) or to the reuse of material already contributed. Several technical problems also exist; these include design of appropriate user interfaces for both the submission and the retrieval of material; the choice of file formats appropriate for reuse; and the storage of the LOs in a manner that allows for efficient access.

In most information retrieval systems, users’ cognitive ability on what they need is a basic system assumption. However, in e-Learning systems, users may have no idea on what they want and what the LOs' metadata is (Lee et al. 2006). Therefore, there is a need to adopt semantic-aware LO retrieval approach and not only using the traditional keyword-based retrieval.

Considering those problems in existing legacy e-learning system, there are two fundamental questions needing to be answered:

- How SW and KM technology can enhance the legacy LMS?
- How to exemplify and evaluate the benefit of implementation SW and KM technology in LMS?

1.3 Objectives and Scope

The objectives and scope of this thesis are outlined as below:

1. To design framework of the SWA-KMDLS : Semantic Web Aware and Knowledge Management Driven e-Learning System (Chapter 4).
2. To develop and implement the SWA-KMDLS (Chapter 4).
3. To develop Ontology that models the SWA-KMDLS environment (Chapter 5).
4. To develop Ontology that models the knowledge in SWA-KMDLS (Chapter 5).
5. To implement the framework of SWA-KMDLS (Chapter 5).
6. To evaluate and validate the Ontology development using common evaluation measurement (Chapter 6).
7. To test and evaluate the performance of the SWA-KMDLS (Chapter 6).

1.4 Thesis Contribution

The main contributions of this thesis surrounds the exploitation of Semantic Web and KM and its technology to enhance an e-Learning Management System where the e-Learning is viewed as part of more general framework, Knowledge Management, and developed in Semantic Web environment the promising future of web technology. It will benefit e-Learning developer in designing an e-Learning system from point of view of KM as broader framework as well as SW (the future of web technology)-aware environment. It also benefits the educational practitioner to implement *social constructivist* pedagogical approach in a *courseware* development.

The contribution of this thesis also surrounds the implementation of SW and KM technology for enhancing LMS particularly in representing learning material content and organizing learning repositories, enabling shareable LOs and learner models, knowledge representation and retrieval, LOs representation and retrieval, encouraging knowledge sharing in e-Learning environment.

1.5 Thesis Organization

This thesis is divided into Seven chapters. It begins by highlighting some background and motivation of the research followed by objectives, scope and contributions. The structure of the rest of this thesis is as follows:

Chapter Two elaborates on the fundamental concept from literature review in the context of research area where this research is conducted namely e-Learning, SW, and KM. The underlying technology of these concept and selected previous works related to this thesis are also presented. This chapter ends by presenting the emerging problem, limitations and requirements in the existing e-Learning and the technology underlying e-Learning that motivates this thesis to overcome these shortcomings.

Chapter Three explains the methodology how the research is conducted. This chapter presents the general design of research work and highlights on Ontology development and Ontology engineering. Ontology is the heart of this thesis work.

Chapter Four focuses on the design architecture and framework of the proposed SWA-KMDLS framework that is composed of a set of components. Each component is responsible for a key aspect of the framework, and the interactions between these components.

Chapter Five stresses the development and implementation of the proposed system in an LMS that is followed by experimental result and evaluation of the proposed system in Chapter Six. Finally the conclusion of this thesis: contributions, limitations, challenges, open issues, and future works are highlighted in Chapter Seven.

CHAPTER 2

LITERATURE REVIEW

The aim of this chapter is to investigate prior research on the current issues, challenges, emerging problems and requirements in e-Learning. This chapter also discusses how the Semantic Web (SW) and Knowledge Management (KM) as two emerging and promising technology can address the current challenges, emerging problems, and requirements in order to make sense of amalgamation of e-Learning, SW and KM. Before going further in these issues, the fundamental concepts and state of the art underlying technology in e-Learning, SW, and KM are elaborated in Appendix I, Appendix II, and Appendix III respectively.

2.1 Emerging Problems and Requirements in e-Learning System

This section is intended to stress on how the SW and KM technology cope with the emerging problems and requirements in e-Learning system. Firstly, the current issues, emerging problems and requirements from previous works is reviewed (§ 2.1.1 and § 2.1.2). Then, an investigation on how SW and KM technology can potentially be exploited for enhancement of e-Learning will be presented (§ 2.1.3 and § 2.1.4).

2.1.1 Emerging Requirement in e-Learning System

e-Learning system has characteristics that make it different and better with traditional learning. The characteristics of e-Learning comparing with traditional learning are presented in Table 2-1.

Table 2- 1 Differences between Traditional Learning (Training) and e-Learning (Drucker 2000)

Dimensions	Traditional Learning (Training)	e-Learning
Delivery	Push – Instructor determines agenda.	Pull – Student determines agenda.
Responsiveness	Anticipatory – Assumes to know the problem.	Reactionary – Responds to problem at hand.
Access	Linear – Has defined progression of knowledge.	Non-linear – Allows direct access to knowledge in whatever sequence makes sense to the situation at hand.
Symmetry	Asymmetric – Training occurs as a separate activity.	Symmetric – Learning occurs as an integrated activity.
Modality	Discrete – Training takes place in dedicated chunks with defined starts and stops.	Continuous – Learning runs in the parallel to business tasks and never stops.
Authority	Centralized – Content is selected from a library of materials developed by the educator.	Distributed – Content comes from the interaction of the participants and the educators.
Personalization	Mass-produced – Content must satisfy the needs of many.	Personalized – Content is determined by the individual user’s needs and aims to satisfy the needs of every user.
Adaptivity	Static – Content and organization / taxonomy remains in their originally authored form without regard to environmental changes.	Dynamic – Content changes constantly through user input.

Elearningeurope Bony Consortium, a well known e-Learning Consortium in Europe, suggest for future work on e-Learning development (elearningeurope 2008). The suggestions include improving the quality of the learning process and content. This improvement can be achieved by selecting the adequate expert for each subject within the community, enhancing intellectual skills and abilities of all participants and making their learning experience more effective as they are active players in the social network.

2.1.2 Semantic Web and e-Learning

As emerging technology that has given a new approach on how information in the Web is represented, SW has great potential to be implemented for enhancing the e-Learning system. Stojanovic et al. (Stojanovic et al. 2001) identified some of SW potentials for realizing the emerging e-Learning requirements (see Table 2-2).

Investigation on exploring Semantic Web for e-learning have been conducted by prior work such have been discussed in Journal of Educational Technology & Society special issues on Ontologies and the Semantic Web for e-Learning (Sampson et al. 2004). Table 2-3 is presented to sum the discussion on this special issues up and to highlight the prior works contribution.

Another suggestion come from Bony Consortium (elearningeurope 2008) that initiated SW potential to improve e-Learning process such as:

1. Ontological representation of content, SW-based approach for knowledge sharing, that allows collaborative creation and maintenance of educational contents.
2. User profiling and intelligent recommending systems, that allow automatic assessment of skills and competences into different areas of interest, so as the automatic definition of a formative path depending on trainees' objectives and pre-existent knowledge.
3. Social network analysis finalized to the creation of a cooperative model in order to select automatically training or research partners from the established network. This analysis includes research in special algorithms elaboration, creating e-Learning Social Network keys features such as the "Expert Finding" and the "User Reputation system".

Table 2- 2 Benefits of using Semantic Web as a technology for e-Learning (Stojanovic et al. 2001)

Dimension	e-Learning	Semantic Web
Delivery	<i>Pull</i> – Student determines agenda.	Knowledge items (learning materials) are distributed on the Web, but they are linked to commonly agreed ontology(ies). This enables construction of a user specific course, by semantic querying for topics of interest.
Responsiveness	<i>Reactionary</i> – Responds to problem at hand.	Software agents on the Semantic Web may use commonly agreed service language, which enables co ordination between agents and proactive delivery of learning materials in the context of actual problems. The vision is that each user has his own personalized agent that communicates with other agents.
Access	<i>Non-linear</i> – access knowledge in whatever sequence makes sense to the situation at hand.	User can describe situation at hand (goal of learning, previous knowledge) and perform semantic querying for the suitable learning material. The user profile is also accounted for. Access to knowledge can be expanded by semantically defined navigation.
Symmetry	<i>Symmetric</i> – Learning occurs as an integrated activity.	The Semantic Web (semantic intranet) offers the potential to become an integration platform for all business processes in an organization, including learning activities.
Modality	<i>Continuous</i> – Learning runs in parallel and never stops.	Active delivery of information (based on personalized agents) creates a dynamic learning environment.
Authority	<i>Distributed</i> – Content comes from the participants and the educators.	The Semantic Web will be as decentralized as possible. This enables an effective co-operative content management.
Personalization	<i>Personalized</i> – Content is determined by the individual user’s needs and aims to satisfy the needs of every user	A user (using personalized agent) searches for learning material customized for her/his needs. The Ontology is the link between user needs and characteristics of the learning material.
Adaptivity	<i>Dynamic</i> – Content changes constantly through user input, experiences, new practices, business rules and heuristics.	The Semantic Web enables the use of knowledge provided in various forms, by semantically annotation of content. Distributed nature of the Semantic Web enables continuous improvement of learning materials.

Table 2- 3 Prior Work On Investigating Semantic Web for e-Learning

Prior Work	Contribution
(Devedžic 2004)	<p>Exploiting SW, interm of Web Intelligent (WI) for tackling new and challenging research problems especially adaptivity and learner comfortability in AIED such as:</p> <ul style="list-style-type: none"> • Enabling course sequencing and material presentation not only according to the learner model, but also according to the most up-to-date relevant content from the Web. • Automatic discovery, invocation, and composition of educational Web services can free the learner from many time-consuming activities that often disrupt the learning process itself. • Ontology-supported learning process greatly increases automation of a number of learners', teachers', and authors' activities related to Web-based learning environments.
(Cristea 2004)	<p>A conversion method from Adaptive Hypermedia to the Semantic Web. It demonstrated how LAOS, an Adaptive Hypermedia (authoring) framework can be used in the context of the Semantic Web.</p>
(Aroyo and Dicheva 2004)	<ul style="list-style-type: none"> • Identified state-of-the-art research a realistic approach towards the Educational Semantic Web. • Proposed a modular semantic-driven and service-based interoperability framework and related ontology-driven authoring tools. • Identified the challenge of the next generation web-based educational systems such as user-friendly, structured and automated authoring, balancing between

	exploiting explicit semantic information for agreement and exchange of educational information, and on the other hand, collecting and maintaining the information semantics.
(Yang et al. 2004)	Proposed a framework for ontology enabled annotation and knowledge management in collaborative learning environments that consisted of personalized annotation, real-time discussion, and semantic content retrieval.
(Henze et al. 2004)	Framework for personalized e-Learning in the semantic web A logic-based approach to educational hypermedia using TRIPLE
(Abel et al. 2004)	An ontology building process An Ontology-based Organizational Memory for e-learning
(Moreale and Vargas-Vera 2004)	Proposed an architecture for a student semantic portal providing semantic services, including a student essay annotation service.
(Papasalouros et al. 2004)	CADMOS-D: A Hypermedia Design Method Method for transformation of UML modeling to Semantic Description

Joint Information Systems Committee Centre for Educational Technology Interoperability Standards (JISC CETIS) University of Bolton, UK (Campbell and Currier 2005) in their research report gave recommendations to exploit the potential of SW technologies in e-Learning:

- Explore the applicability of various SW technologies including RDF, SKOS, OWL, TopicMaps, etc. to the teaching and learning domain.
- Establish links between the SW community within the UK and the JISC community of educational practitioners and e-Learning technologists.
- Evaluate the ability of SW technologies to represent pedagogical practice.

2.1.3 Knowledge Management (KM) and e-Learning

The integration of knowledge management and e-Learning actually is an elaboration of KMS and e-Learning system. This collaboration is designed to support the rapid capture, delivery, and measurement of knowledge in Web-based fashion. Based on (Dongming and Huaiqing 2008) research framework, an e-Learning system is considered as one of KM enablers that provides the infrastructure necessary for the Learning organization to increase the efficiency of knowledge process. However, from a broader point of view, the KM enabler and the KM processes in e-Learning are correlated in different levels as presented by Table 2-4.

The growing synergies between KM and e-Learning and the convergence of work and learning are leading to the importance of e-Learning as knowledge scaffolding in the 21st century. Learning in KM-driven organizations can take place via mentoring in face-to-face CoPs (Community of Practices), e-Learning in digital environments, or blended learning. Web services and next generation Internet technologies will further enmesh knowledge and learning processes. Standardization is proceeding, thanks to consortia like the Workflow Management Coalition, the HR-XML Consortium, OASIS (Organization for the Advancement of Structured Information Standards), and GKEC (the Global Knowledge Economics Council). The blended of KM and e-Learning can change the view of e-Learning from course-centric to knowledge course-centric (Rosenberg 2006).

Table 2- 4 Relationship between KM enablers/techniques and KM Processes in e-Learning (Dongming and Huaiqing 2008)

		KM Processes in e-Learning				
		Knowledge Creation	Knowledge Storage	Knowledge Retrieval	Knowledge Transfer	e-Learning Application
KM Enablers / Techniques in e-Learning	Content Management	+	++	-	-	++
	Course Management	+	+	-	+	++
	Administration and Operation	-	+	++	++	+
	Evaluation Management	+	+	+	+	++
	Interaction/Collaboration	++	+	+	++	+

Note : - less related; + correlated; ++ highly correlated

In order to take advantage of knowledge mapping and the functions connected with it, software tools capable of supporting and managing interaction in a network environment is needed. The tool must be designed and created on the basis of guidelines aimed at pursuing not only the ergonomics, efficiency and efficacy of use, but also flexibility and completeness during the phases of construction, updating and analysis of the relational fabric (Pedroni 2007).

Simple development and delivery of learning resources can not satisfy requirements of information society of professionals skills achievement, knowledge sharing and exchange, and gaining competencies in specific domains of science and real life necessary for individuals and organizational success and prosperity. That is why education has to be a process of sharing and acquirement of knowledge, skills and competencies. Advantages of KM are very useful for that process. KM is indivisible part

of teams training so capturing of knowledge process is very similar to the processes related to selection of most appropriate learning content in e-learning. Outcome of effective learning process should be not only knowing facts for a separate subject but having practical skills and developing competency in the given domain so acquiring knowledge is more precise definition of the learning outcomes instead of learning facts for different related to domain topics. Therefore KM processes should be more deeply and successfully integrated in learning content delivery and learning activities support (Yordanova 2007).

Prior work on implementing KM for e-learning has been conducted by some researchers. (Woelk and Agarwal 2002) proposed a framework to integrate e-Learning and KM based on enterprise content and knowledge (See Figure 2-1). As depicted by Figure 2-1 the proposed framework is consisted of integrated knowledge management system, content management system, and learning object repository. The strength of this proposed framework is the more more intelligent and well organized authoring system. It encouraged employees to create and transfer knowledge with knowledge manager and software program as knowledge organisator and instructional designer as learning of knowledge organisator. In this scenario, knowledge managers and instructional designers will be able to create 1000 LOs per month.

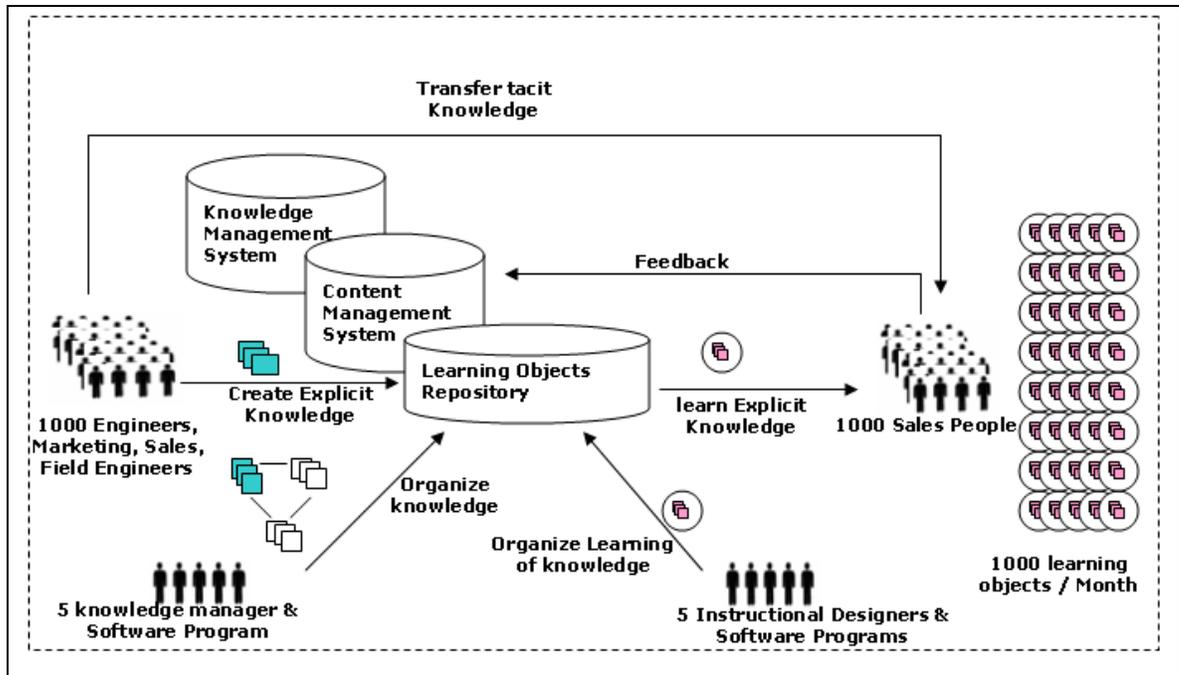


Figure 2- 1 e-Learning and KM Integration based on Enterprise Content and Knowledge (Woelk and Agarwal 2002)

Another prior work on implementing KM for e-Learning is conducted by (Morales et al. 2005). They focused on qualitative research for managing reusable LO using KM approach. Based on (Morales et al. 2005) there are some aspects that must be considered to KM for e-learning systems together with LOs and standards implication namely LOs quality characteristics, LOs normalization, and LOs quality evaluation according to a context.

2.1.4 Knowledge Management and *Social Constructivism* in e-Learning System

Another aspect that needs to be pondered when developing e-Learning system to be successful is the pedagogical approach. One of the pedagogical approach that become emerging perspective on learning, teaching, and technology today is *Social Constructivism* (Kim 2008). Social constructivism emphasizes the importance of culture and context in understanding what occurs in society and constructing knowledge based on this understanding (McMahon 1997; Derry 1999). This perspective is closely associated with many contemporary theories, most notably the developmental theories of Vygotsky and Bruner, and Bandura's social cognitive theory (Kim 2008).

Social constructivism is based on specific assumptions about reality, knowledge, and learning. To understand and apply models of instruction that are rooted in the perspectives of social constructivists, it is important to know the premises that underlie them:

- *Reality*. Social constructivists believe that reality is constructed through human activity. Members of a society together invent the properties of the world (Kukla 2000). For the social constructivist, reality cannot be discovered: it does not exist prior to its social invention.
- *Knowledge*. To social constructivists, knowledge is also a human product, and is socially and culturally constructed (Gredler 1997; Ernest 1999). Individuals create meaning through their interactions with each other and with the environment they live in.
- *Learning*. Social constructivists view learning as a social process. It does not take place only within an individual, nor is it a passive development of behaviors that are shaped by external forces (McMahon 1997). Meaningful learning occurs when individuals are engaged in social activities.

Social constructivists see as crucial both the context in which learning occurs and the social contexts that learners bring to their learning environment. There are four general perspectives that inform how the learning within a framework can be facilitated in terms of social constructivism namely cognitive tools perspective, idea-based social constructivism, pragmatic or emergent approach, and transactional or situated cognitive perspectives (Gredler, 1997).

Cognitive tools perspective focuses on the learning of cognitive skills and strategies. Students engage in those social learning activities that involve hands-on project-based methods and utilization of discipline-based cognitive tools. Together they produce a product and, as a group, impose meaning on it through the social learning process.

Idea-based social constructivism sets education's priority on important concepts in the various disciplines (e.g. part-whole relations in mathematics, photosynthesis in science, and point of view in literature). These "big ideas" expand learner vision and become important foundations for learners' thinking and on construction of social meaning.

Social constructivists with pragmatic or emergent approach perspective assert that the implementation of social constructivism in class should be emergent as the need arises. Its proponents hold that knowledge, meaning, and understanding of the world can be addressed in the classroom from both the view of individual learner and the collective view of the entire class.

The transactional or situated cognitive perspective focuses on the relationship between the people and their environment. Humans are a part of the constructed environment (including social relationships); the environment is in turn one of the characteristics that constitutes the individual. When a mind operates, its owner is interacting with the environment. Therefore, if the environment and social relationships among group members change, the tasks of each individual also change. Learning thus should not take place in isolation from the environment.

This social constructivism perspective on e-Learning makes sense with the spirit of KM that look at knowledge as a social construct. There are four perspective from KM that support this social constructivism on e-Learning namely ontological, epistemological, commodity, and community perspective (Jakubic 2007).

The ontological perspective suggests that knowledge is concerned with the nature of reality. Criteria of measurement and evaluation have to do with understanding the nature of the knowledge and the reality. Whether external or internal and the objective is. Ontological views of knowledge look at two subcategories: (a) Social (b) Individual.

The epistemological view looks at knowledge management from a scientific perspective, and seeks to develop systems of classification that incorporate the logic(s) of science, including a "grammar" of science. Whether the knowledge is tacit or explicit.

In commodity case, knowledge is an asset. As such, its value depends on its utility. In this case, instead of relying on the logic(s) of science, one would look at economics, particularly microeconomics.

Community is the most commonly addressed in contemporary discussions of knowledge and knowledge management, this view suggests that knowledge is a function of community interactions. The underpinning concept is the social construction of reality, and the notion that knowledge is socially determined. While this is undoubtedly useful as a knowledge management philosophy, there are limitations, particularly when needing to create classification schemes that respond to an object's usefulness.

KM is a dynamic, quickly evolving field. As collaboration, peer review, sharing, and modification occur in web applications and information manipulation and retrieval, then it will be necessary to develop more schemes and to refine them in an ongoing way.

2.2 e-Learning, SW, and KM Where is their Meeting Point?

E-Learning, SW, and KM as elaborated in Appendix I, Appendix II, and Appendix III, has its own potential. The next question is how to meet these three concept up together. Prior work on collaborated e-Learning, SW, and KM have been conducted by some researchers such as (Qi and Sun 2006) and (Liu and Hu 2006).

Qi and Sun (2006) proposed Ontology approach for knowledge representation and Topics Map to Link Knowledge with Content in e-Learning community. They also proposed

Knowledge Management System (KMS) Based on Semantic Web in e-Learning Community. The proposed KMS is made up of knowledge collecting subsystem, knowledge management subsystem, and knowledge sharing subsystem. The whole proposed KMS provided the support of environment and tools for the management of learner and knowledge as core objects.

Liu and Hu (2006) developed a knowledge management system named R-ELKM for e-Learning. Some SW technologies such as modeling, automatic mining, discovering and processing techniques have been migrated into R-ELKM. R-ELKM offers a multi-model based framework for knowledge management and several intelligent services like automatic knowledge expansion, clustering and dynamic knowledge representation. It meets the requirement of multiple knowledge domains in practical e-Learning environment.

In this research work, the researcher argued that these three concepts should be put together in order to get the maximum benefit of them. Thus, it can be concluded that ideally to be successful in this *knowledge age* environment e-Learning should be viewed as part of KM spirit in Organization instead of viewed e-Learning out of concern from KM. KM is viewed as broader framework where e-Learning inside it. E-Learning is considered one of tools to capture, organize, and store knowledge –the most important asset and key competitive success factor for an organization. Then, the SW as future of Web technology which provides a common framework that allows data to be shared and reused across application, enterprise and community boundaries is considered to be the key enabler of the successful KM implementation. In other words, it can be concluded that SW is the technology enabler for KM driven e-Learning implementation in an organization. Figure 2-2 depicts our conclusion: where the meeting point of e-Learning, Semantic Web, and KM is.

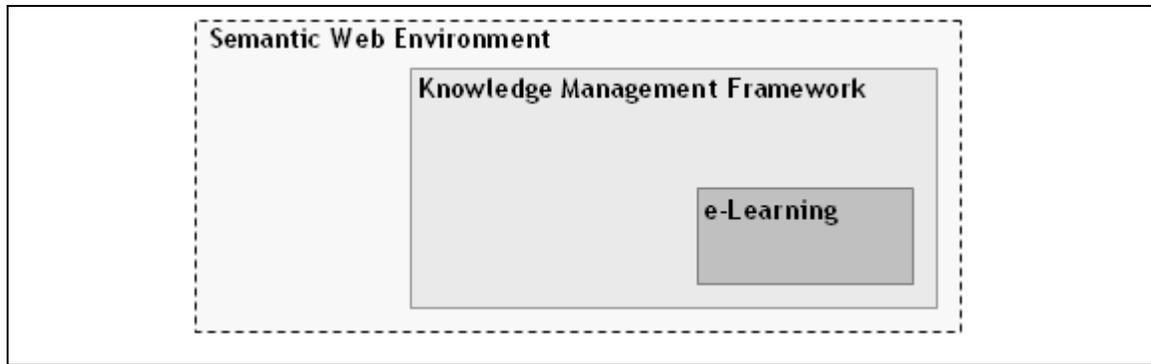


Figure 2- 2 e-Learning, Semantic Web, KM: the Meeting Point

Regarding to the amalgamation of three fundamental concept as depicted by Figure 2-2 the proposed e-Learning system is named as *Semantic Web Aware- Knowledge Management Driven e- Learning System (SWA-KMDLS)*.

2.3 Summary

This chapter has reviewed the meeting point of e-Learning, SW and KM. It shown how e-Learning, SW and KM can be integrated to enable better synergies. Current issues, emerging problems and requirement on e-Learning are also reviewed. It is followed by investigation on how the Semantic Web and KM Technology can meet the emerging problems and requirement in the existing e-Learning system.

The following chapter discusses the methodology that is used to conduct this research work. The heart of this research work is the development of Ontology, because Ontology is the fundamental aspects in either SW or KM and the main concern of this thesis. In SW Ontology is the backbone of SW application, where the successful of SW depend on the successful of Ontology development. In KM Ontology is the representation model for modeling knowledge. Therefore, in the following chapter the Ontology development methodology is highlighted.

CHAPTER 3

METHODOLOGY

This chapter presents the methodology how the research is conducted. This chapter is divided into two parts. The first part (§ 3.1) presents the general research design and the second part (§ 3.2 and § 3.3) present the methodology for developing Ontology. Ontology, as described in prior chapter is the backbone of SW and the fundamental aspect in KM for modeling and representing knowledge. In this research work, METHONTOLOGY is selected. METHONTOLOGY is the most mature Ontology development methodology compared with others as it is discussed in § 3.2 METHONTOLOGY framework that includes Ontology development process, Ontology life cycle and Ontology development technique shall be elaborated in § 3.3

3.1 General Research Design

Concisely, Figure 3-1 depicts the general flow of the thesis research work in block digram. As depicted by Figure 3-1, the research flow is consisted of four main steps namely development of Ontology, design the framework and architecture of SWA-KMDLS, development and implementation of SWA-KMDLS, and Experimental testing and evaluation. The following subsection detail out the rationale, description, purpose, application, assumptions (if applicable), and parameter of each step.

3.1.1 Development of Ontologies

The first step in this research work is Ontologies development. To develop the Ontologies METHONTOLOGY methodology is selected . The rationale on why METHONTOLOGY is selected and how the processes of Ontology development using this methodology is detail out in the following section § 3.2 and § 3.3 respectively. The Ontology is developed using tool : Protégé (Protégé 2008).

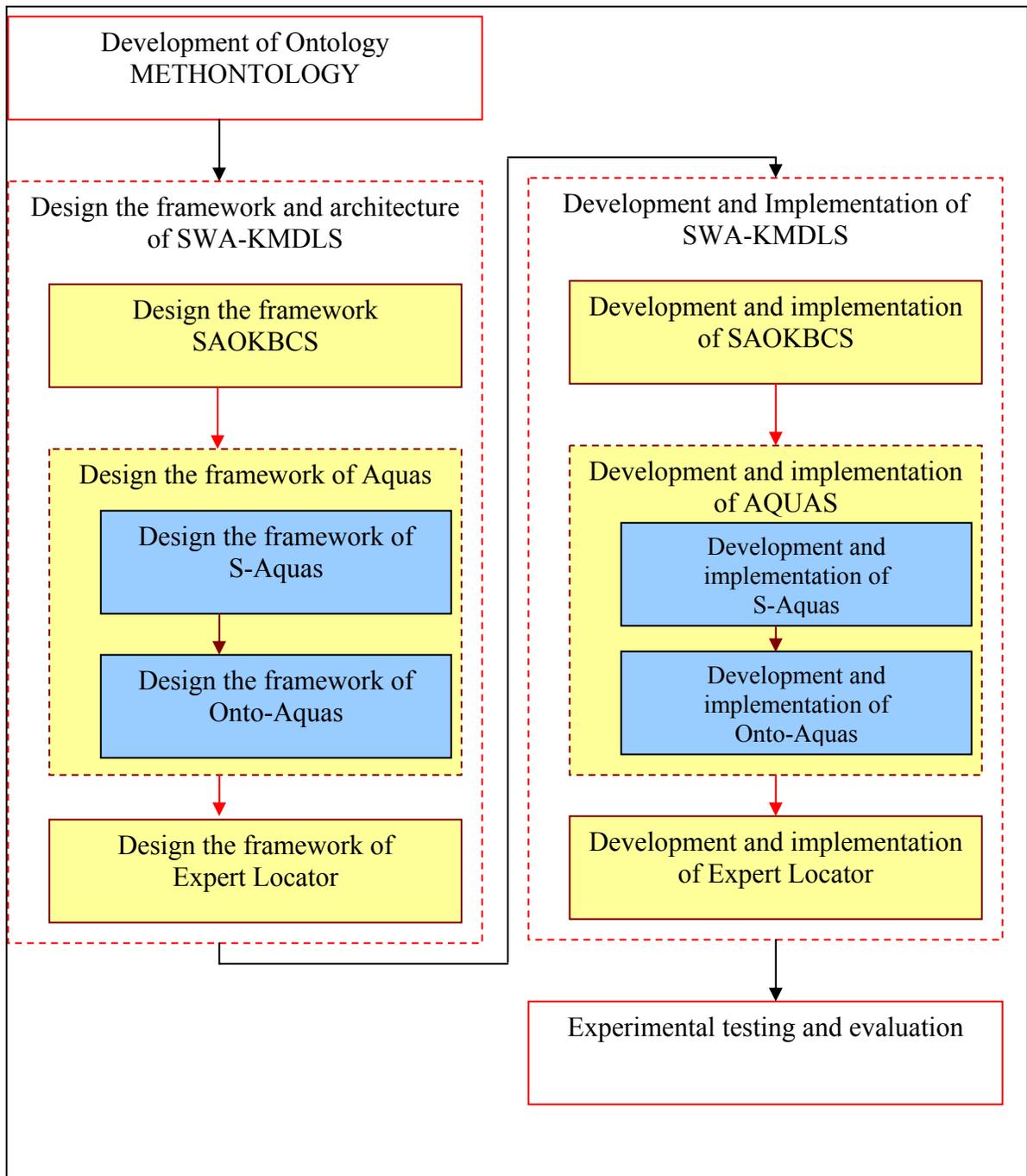


Figure 3- 1 General Research Flow Design

3.1.2 Design the Framework and Architecture of SWA-KMDLS

The second step of the research is Designing of the SWA-KMDLS. In this step, the conceptual design of SWA-KMDLS is figured in framework and architecture. The design of SWA-KMDLS is divided into three folds. The first fold is the SAOKBCS for LO management. The second fold is Aquas for information, LO, and knowledge retrieval. Aquas is designed using two approach namely statistical approach (S-Aquas) and Ontological approach (Onto-Aquas). The third fold is Expert Locator that is intended to improve the performance of Aquas and encourage knowledge sharing culture in e-learning environment.

3.1.3 Development and Implementation of SWA-KMDLS

In this step the a prototype based on framework and architecture of SWA-KMDLS is developed and implemented. The SWA-KMDLS is developed using PHP and Java programming language and implemented in Moodle content management system (Moodle 2008). The difference platform between PHP and Java is addressed using PHP-Java bridge.

3.1.4 Experimental testing and evaluation.

The aim of the experiment in this research work is to assess the performance of SAOKBCS and Aquas. The experimental scenario for SAOKBCS and Aquas is presented in Table 3-1 and Table 3-2 respectively.

Table 3- 1 The SAOKBCS experimental scenarios

No.	Scenarios
1	Course creator uploads Learning Material 1(LM 1) in Introduction To Problem Solving and Programming (IPSP) Course.
2	Course creator uploads Learning Material 2 (LM 2) in Introduction To Problem Solving and Programming (IPSP) Course.

The experiment for SAOKBCS as presented in Table 3-1 involved course creator user in virtual e-Learning system (SWA-KMDLS). The LM1 presented IPSP learning material in which its content explored the fundamental concepts in programming. It is consisted of 285 words and 1697 characters. The LM2 presented IPSP learning material in which its content introduced C Programming concepts. It is consisted of 275 words and 1621 characters. Both LMs are in text format. Usually these text input is named as *Corpus*.

From these two learning materials, the *Ontology Learning* engine of SAOKBCS namely the Text2Onto constructs *contextual Ontology* (semi-)automatically. The algorithms that are used by the Text2Onto engine are set up as depicted by Figure 3-2.

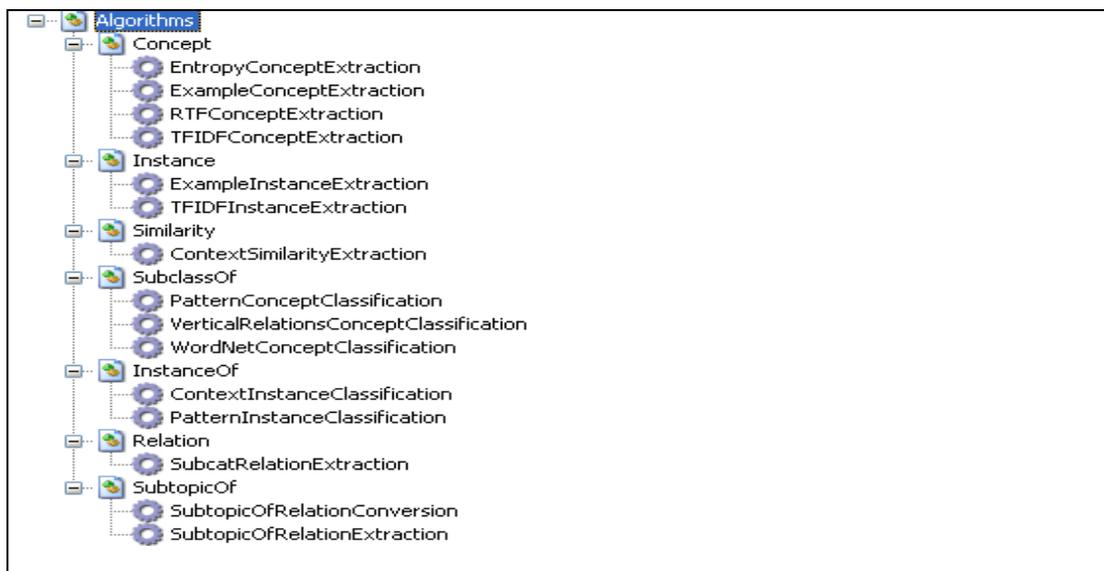


Figure 3- 2 Ontology Learning Algorithm

As depicted by Figure 3-2, the Ontology Learning engine extract *concepts*, *instance*, *similarity* between concepts and instance, *subclassOf* relation between concepts, *instanceOf* relation between concept and instance, *relation* and *subtopicOf* from the corpus.

The experiment for Aquas as presented in Table 3-2 is involved students in Introduction To Problem Solving and Programming (IPSP) e-Learning virtual class consist of 62 students. In this IPSP e-Learning class, students are asked to enter their question using Aquas. In which the students’s questions can be classified into three scenarios as presented in Table 3-2.

The experiment was conducted over C Programming FAQs database adopted from (Summit 2008) containing 300 FAQs. The experiment was also conducted over learning resource base containing 100 learning materials. In the experiment, two versions of data were generated from the FAQ database. The first version is the original data from the FAQ database. The second version is classified data based on six interrogative words i.e. “What”, “Who”, “Where”, “When”, “Why”, and “How.

Table 3- 2 The Aquas Experimental Scenarios

Scenario	Explanation
Scenario 1 (S1)	The student’s entire question is similar to the question in the FAQs database (i.e. such questions have been asked before).
Scenario 2 (S2)	The student’s entire question is quite different to the question in the FAQs database (i.e. such questions have not been asked before, but the answers of the question exist in FAQ database).
Scenario 3 (S3)	The student’s question is randomly selected from Scenario 1 and Scenario 2

This classification intends to measure the performance of Aquas in two different cases. The first case is the student asks a trivial question, a question that has been asked previously and stored in FAQ database. Moreover, the second case is the student asks a non-trivial question, a question has not been asked previously and not stored in FAQ database yet.

3.1.5 The Experimental Validation

The experimental validation for Ontology development has been discussed in the previous section. Ontology development validation is included in evaluation stage as support activities in Ontology development lifecycle. It clearly explained in § 3.3.2 For the other experiment such as automated question answering system, in this research work common evaluation measurement in information or knowledge retrieval namely: *Precision(P)*, *Recall(R)*, *F-Measure(F)* are used. *Precision* is defined as the fraction of relevant documents that are retrieved. *Recall* is defined as the number of relevant documents. *F-Measure* is defined as a single measure that trades off precision versus recall is the, which is the weighted harmonic mean of precision and recall (Manning et al. 2008).

Suppose that the result of query is classified into as presented in Table 3-3 below:

Table 3- 3 Query Result Classification

	Relevant	No Relevant
Retrieved	True Positive (tp)	False Positive (fp)
Not Retrieved	False Negative (fn)	True Negative (tn)

The Precision, Recall, and F Measure can be formulated as below:

$$P = \frac{tp}{(tp + fp)} \quad \text{Eq. 3- 1}$$

$$R = \frac{tp}{(tp + fn)} \quad \text{Eq. 3- 2}$$

$$F = \frac{2PR}{(P + R)} \quad \text{Eq. 3- 3}$$

3.2 Comparative Studies of Ontology Development Methodology

During the process of developing Ontology, one of the most important aspects need to be considered is what methodology should be use. Since ontological engineering is still an immature discipline, there is no consensus between research groups on the Ontology development methodology even Ontology engineering is similar with software engineering. Each research group has their own methodology. Some of the most popular Ontology development methodologies are Ushold & King (Ushold et al. 1995), TOVE (Gruninger and Fox 1995), METHONTOLOGY(Lopez et al. 1999), KACTUS (Bernaras et al. 1996), SENSUS (Swartout et al. 1997) and On-To-Knowledge (Sure and Studer 1999).

A comparative studies of some Ontology development methodologies is undertaken by (Casely-Hayford 2005). This comparative study is based on some criteria: detail of methodology, recommendation for formalization, strategy for building applications and identifying concepts, and recommended life cycle (See Table 3-4). Another comparative studies of methodologies for building Ontologies is under taken by (Corcho et al. 2003) that compare based on features of project management process, Ontology development-oriented process, and integral process. (AhKim and YoungChoi 2007) also compared some Ontology development methodologies using CMM-I (Ahern et al. 2001).

All previous comparative studies agreed that METHONTOLOGY is the most mature and comprehensive methodology among the others for developing Ontology from scratch, reusing other Ontologies as they are, or by a process of re-engineering them, even it has been recommended by For Intelligent Physical Agents (FIPA) for the Ontology construction task methodology. METHONTOLOGY has been widely used to develop Ontologies and applications in different domains (Lopez 1999). Another merit of METHONTOLOGY is its application independent. Thus, it gives more flexibility in Ontology development process.

Table 3- 4 Comparative Studies of Ontology Development Methodologies (Casely-Hayford 2005)

Criteria	Uschold & King	TOVE (Gruninger & Fox)	METHONTOLOGY (Lopez)	KACTUS (Bernaras et al.)	SENSUS (Swartout et al.)	On-To-Knowledge (Sure and Studer)
Detail of Methodology	Very little	Little	A lot	Very little	Very little	Very little
Recommendation for formalization	None	Logic	None	None	None	None
Strategy for building applications	Application Independent	Application semi-Independent	Application Independent	Application Independent	Application semi-Independent	Application Independent
Strategy for identifying concepts	Middle-out	Middle-out	Middle-out	Top-Down	Top-Down	Top-Down
Recommended life cycle	None	None	Yes	None	None	Yes

Considering prior comparative studies on Ontology development methodologies that all concluded and agreed that METHONTOLOGY is the most mature methodology, in this research work METHONTOLOGY is chosen. How this methodology guides the Ontology development process is presented in the following section.

3.3 METHONTOLOGY

METHONTOLOGY is a methodology created in the Artificial Intelligence Lab from the Polytechnic University of Madrid (PUM). It is used for building Ontologies either from scratch or by a process of reusing or re-engineering other existing Ontologies. The METHONTOLOGY framework enables the construction of Ontologies at the knowledge level. It includes the identification of the Ontology development process (§ 3.3.1), Ontology life cycle (§ 3.3.2), and particular techniques to carry out each activity (§ 3.3.3) (Casely-Hayford 2005).

3.3.1 Ontology Development Process

The Ontology development process identifies which tasks or activities should be performed when building Ontologies. It identifies three categories of activities namely: management, technical and support as depicted by Figure 3-3 (Corcho et al. 2005).

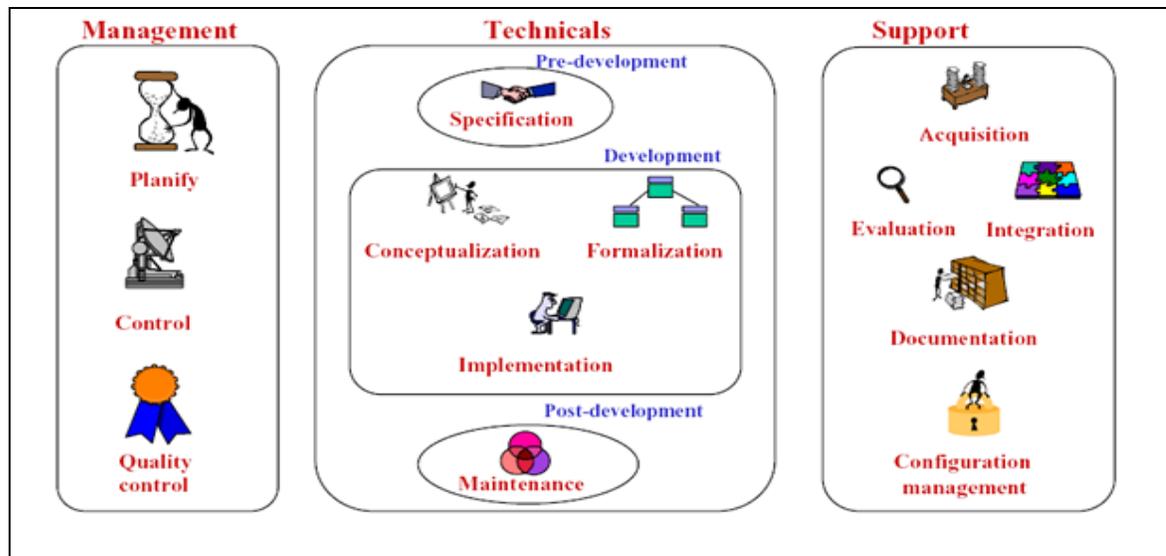


Figure 3- 3 METHONTOLOGY: Ontology Development Process (Corcho, Fernández-López et al. 2005)

Each activity in Ontology development process is detailed bellows:

(i) Management Activities

The first category of Ontology development activities is management activities. These activities include planning or scheduling, control and quality assurance:

- The *planning* or *scheduling* activity identifies the tasks to be performed, their arrangement and the time and resources needed.
- The *control* activity guarantees completion of tasks as intended.
- The *quality assurance* activity checks the quality of each methodology output (Ontology, software and documentation).

(ii) Technical Activities

The second category of Ontology development activities is technical activities. These activities are grouped in pre-development, development and post-development activities:

- *Pre-development*, in this state the environment where the Ontology will be used are studied and feasibility study is under taken to create requirement specification. This step also identifies the purpose of the Ontology, including the intended users, scenarios of use, the degree of formality required, etc., and the scope of the Ontology including the set of terms to be represented, their characteristics and the required granularity. The output of this phase is a natural-language Ontology specification document.
- *Development*, this state include these activities:
 - The *conceptualization* activity structures the domain knowledge as meaningful models at the knowledge level, identifies domain terms as concepts, instances, verbs relations or properties and each are represented using an applicable informal representation.
 - The *formalization* one transforms the conceptual model into a formal or semi-computable model.
 - Finally, the computable models are built in *implementation* in which the Ontology is formally represented in a language, such as OWL.

- *Post-development*, this state includes maintenance activity that updates and corrects the Ontology if needed and it can be reused by other Ontologies or applications.

(iii) Support Activities

The last category of Ontology development process is support activities. These activities are performed at the same time as the development activities. During support stage, the following activities take place:

- The knowledge *acquisition* activity aims to acquire knowledge from experts or by (semi-)automatic Ontology learning.
- *Evaluation* activity aims to judge the developed Ontologies, software and documentation against a frame of reference.
- *Integration* to obtain some uniformity across Ontologies, definitions from other Ontologies activity if other Ontologies are reused possibly in conjunction with *merging* or *alignment* activities if multiple Ontologies are reused and need to be combined. Merging produces a new Ontology from the combination while alignment establishes mappings that preserve the original Ontologies.
- *Documentation* details each completed stage and product.
- *Configuration management* records Ontologies, software and documentation versions in order to control changes.

The previous Ontology development process identifies the activities to be performed. It does not say anything about how it should be scheduled. This is determined by the other part of the methodology, the Ontology life cycle that establishes the stages through which the Ontology moves during its lifetime and the activities to be performed in each stage.

3.3.2 Ontology Life Cycle

The Ontology life cycle identifies the stages through which the Ontology passes during its lifetime, as well as the interdependencies with the life cycle of other Ontologies. It schedules the Ontology development activities detailed previously, although not all of them are currently considered by the METHONTOLOGY life cycle. The life cycle is

cyclic, based on evolving prototypes (López et al. 2000). It allows an incremental development of the Ontology that enables earlier validation and readjustment. Each cycle starts with the scheduling activity that identifies the tasks to be performed, their arrangement, their temporal extent and the resources they need. After that, the development activities are engaged, starting with specification. Simultaneously, the management activities, control and quality assurance, and the support activities, knowledge acquisition, integration, evaluation, documentation and configuration management, are launched. They take place in parallel with the development activities.

In each cycle, the current prototype Ontology moves along the development activities, from specification through conceptualization, formalization and implementation until maintenance, although it is not necessary to pass through all of them. Eventually, the prototype might be mature enough for evaluation purposes and a new cycle can be engaged considering the conclusions from this evaluation. If a development cycle is completed, these are the steps to be performed:

1. Specify the Ontology prototype.
2. Build a conceptual model from pieces provided by the knowledge acquisition activity, which is mainly run during the conceptualization.
3. Formalize the conceptual model.
4. Implement the formalized conceptual model. This can be automatic if the formalization can be translated automatically to an Ontology implementation language.
5. Maintain the resulting Ontology, which might lead to a new development cycle if unsatisfied or new requirements are detected.

The Ontology life cycle activities are depicted by Figure 3-4.

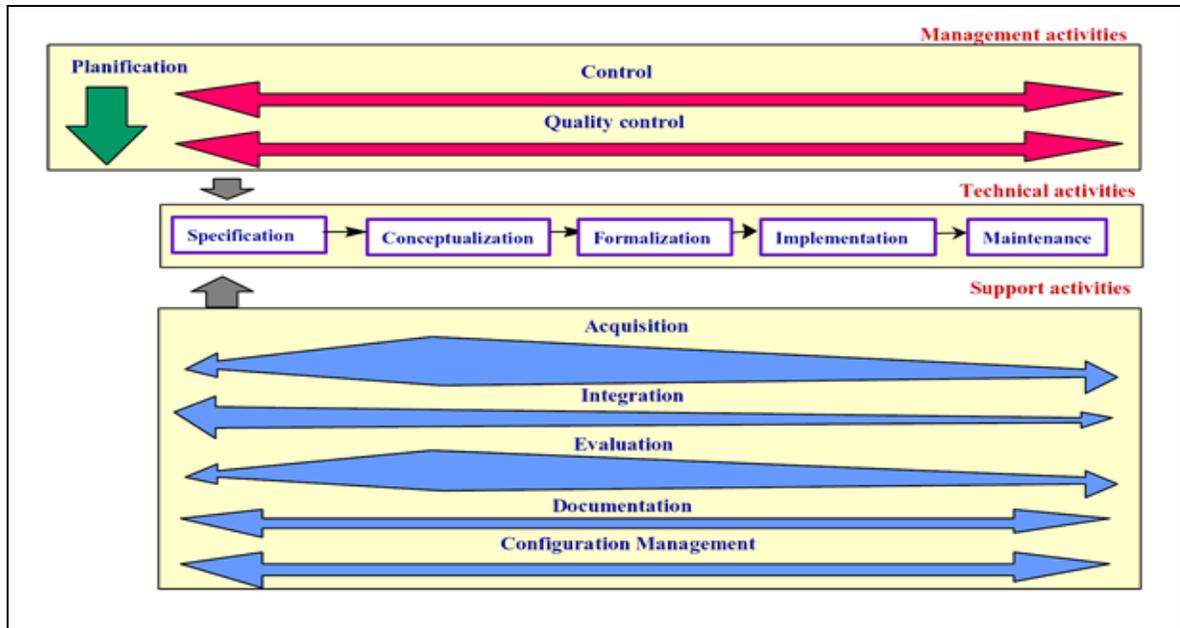


Figure 3- 4 METHONTOLOGY: Ontology Development Process Life Cycle (Corcho et al. 2003)

As depicted by Figure 3-4, the activities in the management and support processes take place simultaneously with the technical activities. The efforts applied to the support activities are not uniform along the life cycle. Knowledge acquisition, integration and evaluation are greater during Ontology conceptualization. This is due to most knowledge is acquired at the beginning of the development, Ontologies are integrated at the conceptual level before implementation and it is better to accurately evaluate the conceptualization as earlier as possible in order to avoid propagating errors.

It is important to note that all the relationships between activities detailed until this point are intra-dependencies, i.e. there are relationships between activities from the same Ontology development process. Intra-dependencies define the Ontology life cycle. METHONTOLOGY considers also that activities for the development of Ontology may involve activities for other Ontologies already built or under construction. These are called inter-dependencies and defined crossed life cycles of Ontologies. They are necessary, for instance, because it is usually necessary to perform some changes before Ontology is integrated with the Ontology currently under development.

Now, there are more detailed descriptions of the activities scheduled by the Ontology life cycle in METHONTOLOGY. They are organized in the management, development and support processes. The pre-development activities (environment and feasibility studies) and the post-development activity use are not detailed, as they are not included in the current life cycle. Moreover, the maintenance activity is moved to the development process.

The Ontology life cycle activities as depicted by Figure 3-4 are detailed as bellows:

(i) Management Activities

The management activities are responsible for the project management issues (González 2005). These activities include:

- *Planning or Scheduling*, it is the first activity of the Ontology life cycle. The objective of this activity is to plan the main tasks to be done and how they will be arranged. It also determines the required resources, i.e. people, software and hardware.
- *Controlling*, it is performed along the whole Ontology life cycle in order to survey that there are not undesired deviations from the initial schedule.
- *Quality Control*, it is responsible for checking that the quality of each methodology output (Ontology, software and documentation) is assured.

(ii) Technical Activities

The technical activities includes all the activities that produce the successive prototype refinement stages towards the desired Ontology (Lopez et al. 1999). The process starts with specification that produces an informal output that then evolves increasing its level of formality, as it passes through the different activities, towards the final computable model, which can be directly understood by the machine. Each stage in the development process or the technical activities is detailed below:

ii.1 Specification

The specification establishes the Ontology purpose and scope. Why the Ontology is being built, what are the intended uses and end-users. The specification can be informal, in natural language, or formal, e.g. using a set of competence questions (González 2005).

ii.2 Conceptualization

The objective of this activity is to organize and structure the knowledge acquired during knowledge acquisition using external representations that are independent of the knowledge representation and implementation paradigms in which the Ontology will be formalized and implemented next. An informally perceived view of a domain is converted into a semi-formal model using intermediate representations based on tabular and graph notations. These intermediate representations (concept, attribute, relation, axiom and rule) are valuable because domain experts and Ontology developers can understand them. Therefore, they bridge the gap between people's domain perception and Ontology implementation languages.

In order to build a consistent and complete conceptual model, the conceptualization activity defines a set of tasks that should be executed in succession. These tasks increase, gradually, the complexity of the intermediate representations used to build the conceptual model. This way, it is easier to ensure a consistent and complete conceptual model (González 2005):

1. First, it is necessary to build a *glossary of terms* to be included on the Ontology, their natural language definition and their synonyms and acronyms. Terms are identified following a middle-out strategy. The core of basic terms is identified first and then they are specialized and generalized as required. This strategy provides a balanced set of terms because detail only arises as necessary and higher-level categories are built naturally.
2. Then, the terms are classified into one or more *taxonomies of concepts*, where a concept is an abstraction for one or more terms. The subclass of taxonomic relation is used, where: C subclass of D $\Leftrightarrow \forall i$ instance of C \rightarrow i instance of D.

3. *Binary relations* are used to define the ad hoc relations between concepts of the Ontology and with concepts of other Ontologies. Relations are determined by their name and the source and target concepts.
4. The *concept dictionary* is built. It describes each concept by stating the relations that have it as their domain and the concept instance and class attributes. Class *attributes* have the same value for all instances of the concept, while instance attributes have different values for each instance of the concept. Moreover, it might be necessary now to define some concept *instances*, although it is more usual to create them during Ontology use, after its construction.
5. The concept dictionary is detailed. For each relation, it is specified its cardinality, inverse relation and mathematical properties (symmetric, transitive, functional, etc.). Instance and class attributes are also described in terms of their concept, value type, measurement unit, range, cardinality, value, related axioms and rules that infer the value of this attribute or use it to infer other attributes. Moreover, there is a *constants* table that defines immutable aspects of the domain of knowledge.
6. Once concepts, taxonomies, attributes and relations have been defined, formal axioms and rules are used for constraint checking and for inferring values for attributes. *Axioms* are logical expressions that are always true and are normally used to specify constraints. They are defined informally in textual form and formally in first order logic. Moreover, all the concepts, relations and attributes used in the definitions are highlighted. *Rules* are generally used to infer knowledge in the Ontology, such as attribute values, relation instances, etc. Rules are also defined informally and formally and the related concepts, relations and attributes are highlighted. The "if conditions then consequent" rule template determines rules definitions. In order to avoid inference complexity problems, it is usual to restrict the conditions to a conjunct of atoms and the consequent to a single atom. These restrictions might be relaxed if inferences that are more complex are needed.

ii.3 Formalization

The goal of this activity is to formalize the conceptual model. There are Ontology development tools that automatically implement the conceptual model into several Ontology languages using translators. Therefore, formalization is not a mandatory activity.

ii.4 Implementation

This activity builds computable models using Ontology implementation languages. There are many Ontology languages such as RDF(S), OWL, etc; and they do not have the same expressiveness nor do they reason the same way.

ii.5 Maintenance

This activity updates and corrects the Ontology if needed due to the necessities of the current development process or other processes that reuse this Ontology in order to build other Ontologies or applications.

(iii) Support Activities

The support activities are performed in parallel with the development-oriented activities. These activities include:

iii.1 Knowledge Acquisition

First, the source knowledge must be captured using knowledge elicitation techniques. The sources of knowledge are listed giving a description and specifying the elicitation techniques used in each case. The techniques used to extract knowledge from sources can be partially automatic by means of natural language analysis and machine learning techniques (Lopez 1999).

iii.2 Evaluation

The evaluation activity judges the developed Ontologies, software and documentation against a frame of reference. Ontologies should be evaluated before they are used or reused. There are two kinds of evaluation, the technical evaluation, which is carried out by developers, and users' evaluation.

Ontology evaluation includes (González 2005):

- Ontology *verification* refers to building the Ontology correctly, that is, ensuring that its definitions implement correctly the requirements or function correctly in the real world.
- Ontology *validation* refers to whether the Ontology definitions really model the real world for which the Ontology was created.
- Ontology *assessment* is focused on judging the Ontology from the user's point of view. Different types of users and applications require different means of assessing Ontology.

The criteria for Ontology evaluation are:

- *Consistency*, which checks if all individual definitions are consistent and no contradictory knowledge, can be inferred from other definitions and axioms. Some consistency problems are circular definitions, common classes or instances in disjoint decompositions and partitions, external instances in exhaustive decompositions and partitions and semantic errors.
- *Completeness*. All that is supposed to be in the Ontology is explicitly stated in it, or it can be inferred. Some common completeness errors are incomplete concept classification, disjoint knowledge omission and exhaustive knowledge omission.
- *Conciseness*. Ontology is concise if it does not include unnecessary definitions, explicit redundancies between definitions do not exist and redundancies cannot be inferred. Some redundancies are redundant subclass of or instance of relations and identical formal definitions of classes or instances.

iii.3 Integration, Merging and Alignment

The integration activity is needed if other Ontologies are reused (González 2005). There are two options when Ontology is integrated in the current ontological framework. *First*, there is Ontology alignment that consists in establishing different kinds of mapping between the Ontologies, hence preserving the original Ontologies. *Second*, Ontology merging that produces a new Ontology from the combination of the input Ontologies.

iii.4 Documentation

Documentation details at each completed stage and product.

iii.5 Configuration Management

Configuration management records Ontologies, software and documentation versions in order to control changes (González 2005).

3.3.3 Ontology Development Technique: Ontology Tool and Language

Finally, the methodology specifies the techniques used in each activity, the products that each activity outputs and how they have to be evaluated (Corcho et al. 2003).

Besides deciding what methodology used in developing Ontology, another aspects need to be considered is development tool and the Ontology Language. By considering previous works on comparative studies for Ontology methodology, tools, and language (Corcho et al. 2003; Casely-Hayford 2005; AhKim and YoungChoi 2007), Protégé 4.0 and POWL are used as Ontology development tool, and OWL Ontology language as depicted by Figure 3-5.

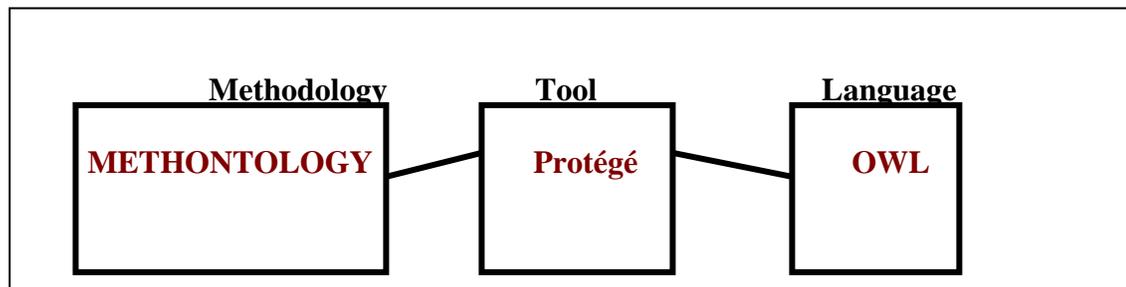


Figure 3- 5 Methodology, Tool, and Language for Ontology Development

3.4 Summary

This chapter has discussed the selected Ontology development methodology namely METHONTOLOGY, that give a guideline on Ontology development process, Ontology life cycle, and Ontology development technique. Prior to METHONTOLOGY, the reason to select METHONTOLOGY was presented. This chapter ended with experimental

validation method to evaluate the experimental result in this thesis research work. The following chapter explains the design level of the proposed SWA-KMDLS. It consists of the architecture and framework where the Ontologies are applied and implemented in the SWA-KMDLS system.

CHAPTER 4

DESIGN FRAMEWORK AND ARCHITECTURE OF SWA-KMDLS

This chapter presents the design framework and architecture of SW and KM technology-driven e-Learning system. This system is called as “SWA-KMDLS”, which is contraction of Semantic Web Aware - Knowledge Management Driven e-Learning System. The distinctive point of SWA-KMDLS compared with the existing legacy LMS consist of three folds. The first fold is the Ontological approach for LO management by developing “(Semi-) Automatic Ontological Knowledge Base Construction System” (abbreviated SAOKBCS) that is presented in § 4.1. The second fold is the Ontological approach for knowledge reusing and LO retrieval by developing “Automated question answering system” (abbreviated Aquas) that is presented in § 4.2 and the last fold is the “Expertise Finding and Mapping Locator” framework to encourage knowledge sharing in e-Learning environment that is presented in § 4.3.

4.1 Ontological Learning Object Management

One of the main issues faced by e-Learning practice is the Learning Objects (LOs) management (refers to representing, organizing, indexing and reusing LOs). This problem becomes more critical when the number of LOs increase dramatically in e-Learning system that has caused difficulties in reusing LOs. As it has been discussed in chapter two, Learning Object Metadata (LOM) is insufficient to cope with this problem, a new approach known as Ontological approach is used for managing LOs in the proposed SWA-KMDLS as depicted by Figure 4-1. The LO is represented in Ontology instead of creating the traditional LOM.

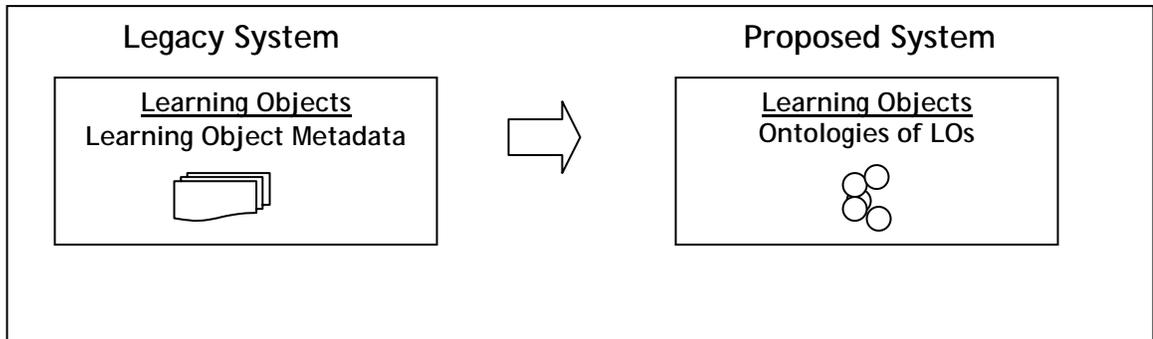


Figure 4- 1 Ontology Approach for Learning Objects Management

The idea of Ontological approach for LO management is realized by developing (Semi-) Automatic Ontological Knowledge Base Construction System (SAOKBCS) in the proposed SWA-KMDLS. The aim of SAOKBCS is to construct Ontology (semi-) automatically from LO uploaded in SWA-KMDLS. The constructed Ontology models the knowledge either the context or the content of the uploaded LO. Following subsections shall give a more detail discussion on how SAOKBCS framework works.

4.1.1 SAOKBCS Framework

Although Ontologies has shown its merits in knowledge representation, but manual Ontology development has proven to be hard and tedious task. Therefore, there is a need to develop a tool that can (semi-)automatically construct Ontology from scratch. In the SWA-KMDLS, a framework that can generate Ontological knowledge base from uploaded LOs (semi-)automatically is developed. The SAOKBCS framework is depicted in Figure 4-2.

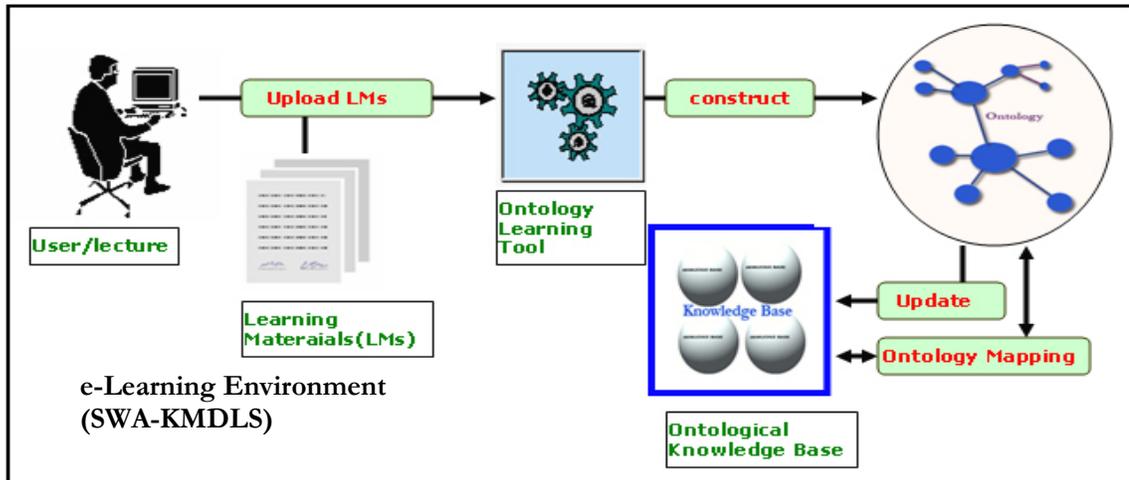


Figure 4- 2 SAOKBCS Framework

As depicted in Figure 4-2 the (semi-)automatic Ontological knowledgebase framework in raw granularity can be explained as bellows:

- (1) *Lecturer (User)*: upload Learning Materials (LMs). This framework starts when lecturer or any other user who has the authority (such as the tutor or administrator) upload learning materials. The learning materials that will be processed should be in any text file format (e.g. .doc, .pdf, .txt, .htm, .html). Other format such as images, audios, videos will not be processed.
- (2) Invoke *Ontology Learning* tool. When the LMs are successfully uploaded to the e-Learning system, the Ontology Learning tool is invoked. This Ontology Learning tool will construct new Ontology (semi-)automatically from the uploaded LMs. How the Ontology Learning tool works is detailed in the following section (§ 4.1.3)
- (3) New *Ontology* is constructed. The output of the Ontology Learning tool is a new Ontology that is represented in OWL (Web Ontology Language) which is one of the most expressive Ontology representation languages.
- (4) Invoke *Ontology Mapping* tool. Since the initial Ontological knowledge base has been existed in the system, there is a need to merge and map the new Ontology with the existing one. The aim of Ontology mapping is to avoid inconsistency and redundancy on concepts and relations in the Ontological knowledge base. This

Ontology Mapping is undergone by an Ontology mapping tool. How the Ontology Mapping tool works is detailed in the following section (§ 4.1.4).

- (5) Update the existing *Ontological Knowledge Base*. Finally, after completing the Ontology Mapping phase, the new Ontology will update the existing knowledge base.

From the above general framework of SAOKBCS, some important parts need to be stressed and discussed in more detail. They are Ontology population, Ontology Learning, Ontology Mapping, Ontology Merging and Ontology Alignment. These important parts will be presented bellows:

4.1.2 The Ontology Population

The aim of Ontology population is to create the initial knowledge base. The initial knowledge base is developed by creating Ontology manually. The process of defining and instantiating a knowledge base is referred to as *knowledge markup* or *Ontology population*. Ontology population must be done first, before invoking Ontology learning module.

4.1.3 The Ontology Learning Module

The main part of SAOKBCS framework is the Ontology Learning (OL) module. Ontology learning simply can be defined as any (semi-) automatic support for Ontology development. In the context of Semantic Web, Ontology learning is primarily concerned with knowledge acquisition from and for Web content and is thus moving away from small and homogeneous data collections to tackle the massive data heterogeneity of the World Wide Web instead (Buitelaar et al. 2003). This research work emphasized Ontology learning as (semi) automatic support for Ontology development from LOs namely text Learning Materials (LMs).

The OL module in the framework is intended to construct two kinds of Ontology from its input corpora i.e. LMs. The first Ontology is *Context Ontology*, which models the context of LOs for example who is the author, what subject the LMs is intended for, what is the

format, etc. The second Ontology is *Content Ontology*, which models the latent knowledge inside the particular LMs, for example in the case of LM containing introduction to programming, the content Ontology models some fundamental concepts and relations between concepts in programming. The following subsections give more detail discussion on how both context and content Ontology are constructed.

(i) Constructing Context Ontology from LOs

In Constructing Ontological knowledge base, OL module has subtask to construct an Ontology that describe (model) the context of the particular uploaded LO. This Ontology is constructed automatically. The prototype of LO’s context Ontology is depicted by Figure 4-3.

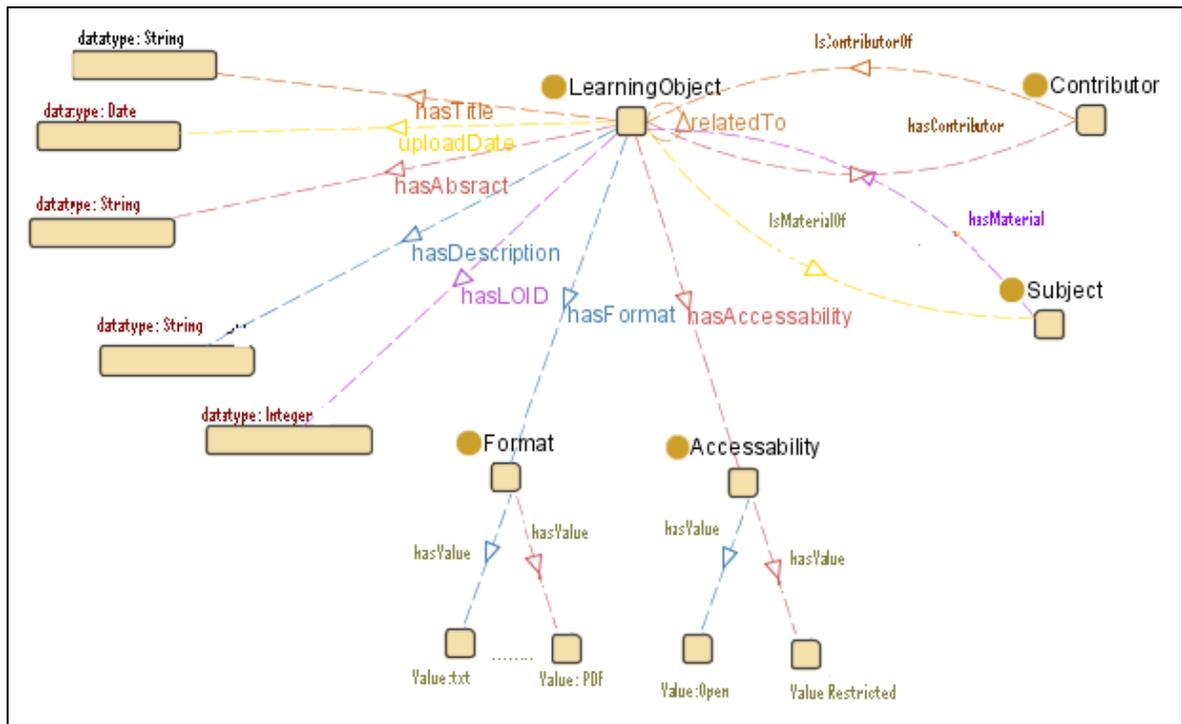


Figure 4- 3 Prototype LO’s Context Ontology

As shown in Figure 4-3, the context of LO is identified by:

- (i) *Subject*, it means to what subject or course the LO is intended to.
- (ii) *Contributor*, it means who is the contributor or author of the LO.
- (iii) *Accessibility*, whether LO is *Open* or *Restricted* to authorized user only.
- (iv) *Format*, indicates the file format of LO such as: txt, ppt, doc, htm/html, or pdf.
- (v) LO is also identified by its *Title, Description, Abstract, and relation* with other LO. The relation means what other prior LOs that related to the particular LO.

(ii) Constructing Content Ontology from LOs

The construction of content Ontology from LO constitutes subtasks concerning with the definition of terms, synonyms, concepts, concept hierarchy, relations, relation hierarchy, axiom schemata and general axioms that representing knowledge in the LO content. This Ontology artifacts and its example are illustrated as “Ontology Learning Cake” that is depicted in Figure 4-4. This “Ontology Learning Cake” is derived from (Buitelaar et al. 2003; Cimiano 2006).

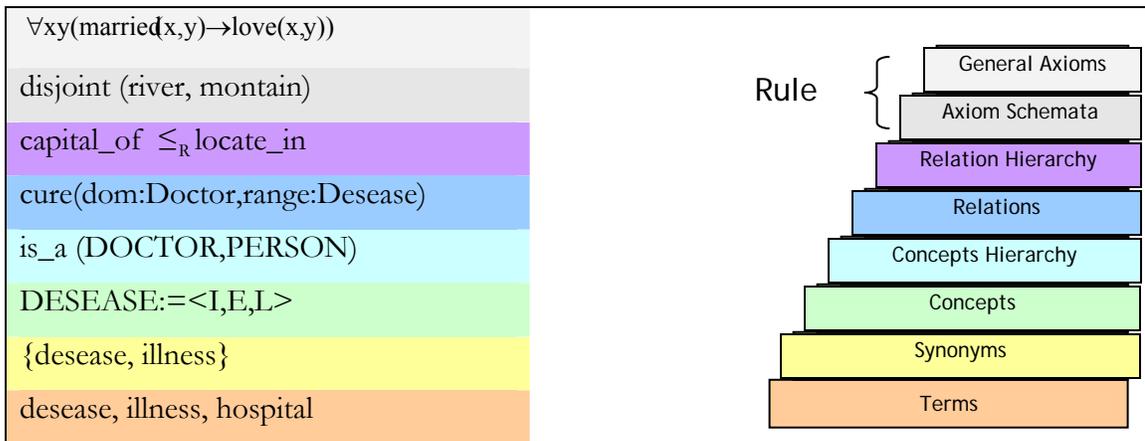


Figure 4- 4 Ontology Learning Layer Cake

The subtask of OL module while developing the LO’s content Ontology in extracting each Ontology artifact concisely is detailed in Table 4-1

Table 4- 1 Ontology Learning Task based on Ontology Artifact

Ontology Artifact	Ontology Learning (OL) Task
<i>Terms</i>	<i>Terms</i> are any single word or multi-word compound relevant and specific for the domain. Thus, the task of Ontology Learning (OL) is to extract a set of strings S_C and S_R representing terms that will be used as signs for concept and relations, respectively.
<i>Synonym</i>	Finding words which denote the same concept and which thus appear in the same set $Ref_C(c)$ for a given concept c .
<i>Concept</i>	Finding a triple $\langle i(c), [[c]], Ref_C(c) \rangle$ where $i(c)$ is the intention of concepts, $[[c]]$ its extension and $Ref_C(c)$ describes its lexical realization in a corpus.
<i>Concept hierarchy</i>	Finding the hierarchical relation between concepts.
<i>Relation</i>	Finding relation identifier or labels r that binary relate domain $dom(r)$ and $range(r)$.
<i>Axiom Schemata</i>	Not learning the axiom schemata itself. But, learning which concepts, relations, or pairs of concepts the axiom in the system apply to, i.e. which pairs of concepts are disjoint, which relations are symmetric, the cardinality of relation, etc.
<i>General Axiom</i>	Deriving more complex relationships and connections between concepts and relations.

To do this work, the proposed framework adapted Text2Onto architecture(Cimiano and Vöölker 2005) (See Figure 4.3). Text2Onto has three main features that distinguish it

from state-of-the-art Ontology learning frameworks such as TextToOnto (TextToOnto 2008), the OntoGen (OntoGen 2006), the OntoBuilder (Roitman and Gal 2006; OntoBuilder 2008), OntoLearn (Navigli and Velardi 2004) or OntoLT (Buitelaar and Sintek 2004; Paul Buitelaar1 2004; OntoLT 2008). First, by representing the learned knowledge at a meta-level in the form of instantiated modeling primitives within a so called Probabilistic Ontology Model (POM), make it independent of a concrete target language while being able to translate the instantiated primitives into any (reasonably expressive) knowledge representation formalism. Second, user interaction is a core aspect of Text2Onto and the fact that the system calculates the confidence value for each learned object allows designing sophisticated visualizations of the POM. Third, by incorporating strategies for data-driven change discovery, it avoids processing the whole corpus from scratch each time it changes, only selectively updating the POM according to the corpus changes instead. Thus, besides increase its efficiency the evolution of the Ontology with respect to the changes in the underlying corpus can be traced.

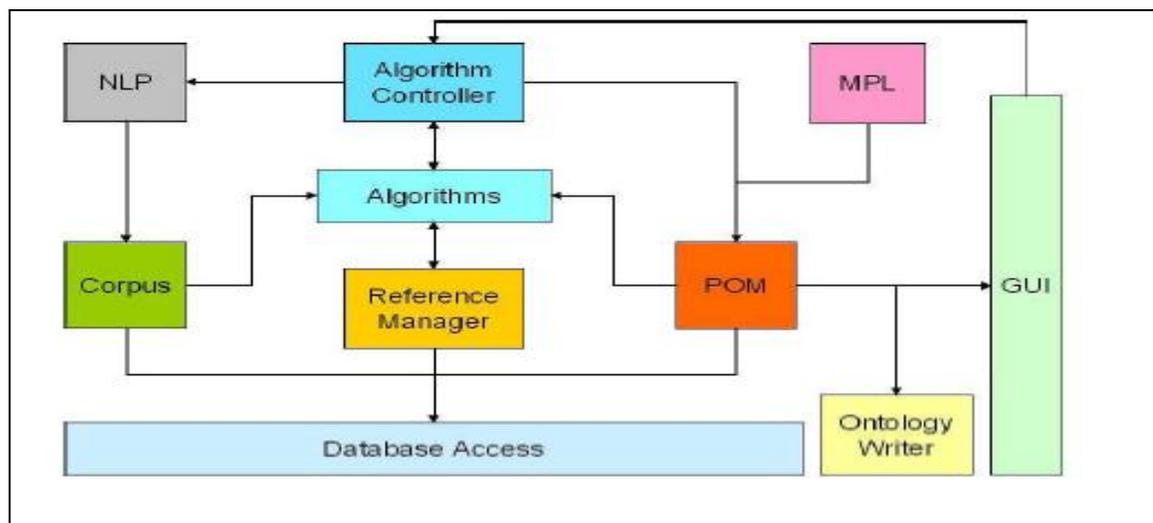


Figure 4- 5 Text2Onto Architecture (Cimiano and VÄolker 2005)

The architecture of Text2Onto in Figure 4-5 concisely can be explained as below:

- (i) The text file as input in the architecture is called a *Corpus*, from which the Ontology is constructed (semi-)automatically.
- (ii) Before being invoked to the further process, the corpus is pre-processed by *Natural Language Processing (NLP)* module such as tokenization, sentence splitting, lemmatizing or stemming and shallow parsing. It is based on GATE (Gate 2000) and JAPE (Cunningham et al. 2000) framework.
- (iii) The *algorithms* are initialized by a *controller*, the purpose of which is: to trigger the linguistic preprocessing of the data (NLP), to execute the Ontology learning algorithms in the appropriate order, and to apply the algorithms' change requests to the POM (Algorithms detailed in Table 4-3).
- (iv) The execution of each *algorithm* consists of three phases: *notification phase* in which the algorithm learns about recent changes to the corpus; *computation phase* in which the changes are mapped to changes with respect to the reference repository where all kinds of knowledge about the relationship between the Ontology and the data (e.g. pointers to all occurrences of a concept) are stored; and *result generation phase* in which requests for POM changes are generated from the updated content of the reference repository.
- (v) The center of the architecture is *Probabilistic Ontology Model (POM)*, which stores the results of the different Ontology learning algorithms. POM is a collection of instantiated modeling primitives (see Table 5-2) which are independent of a concrete Ontology representation language. In fact, Text2Onto includes a *Modeling Primitive Library (MPL)* which defines these primitives in a declarative fashion.
- (vi) So called *Ontology writers* are responsible for translating instantiated modeling primitives from POM into a specific target knowledge representation language such as RDFS (RDFS 2004), OWL (OWL 2004), and F-Logic (Bruijn 2007).

Table 4- 2 The Text2Onto Ontology Primitive Model & Gruber Frame Ontology

Text2Onto Primitive Model	Gruber Frame Ontology
Concepts	CLASS
Concept inheritance	SUBCLASS-OF
Concept instantiation	INSTANCE-OF
Properties/relations	RELATION
Domain and range restrictions	DOMAIN/RANGE
Mereological relations (part-of)	
Equivalence	

Table 4- 3 Algorithm For Primitive Model Extraction

POM	Algorithm
Concepts	Relative Term Frequency (RTF), TFIDF (Term Frequency Inverted Document Frequency), Entropy, and the C-value/NC-value method
Subclass-of Relations	Algorithms for exploiting the hypernym structure of WordNet (Miller 2006), matching Hearst patterns ¹ (Hearst 1992) , and Linguistic heuristics.
Part of	JAPE expressions.
General Relations	Shallow parsing strategy to extract sub categorization frames enriched with information about the frequency of the terms appearing as arguments.
Instance-of Relations	Similarity-based approach extracting context vectors for instances and concepts from the text collection and assigning instances to the concept corresponding to the vector with the highest similarity with respect to their own vector.
Equivalence	Similarity algorithm between terms based on contextual features extracted from the corpus, whereby the context of a terms varies from simple word windows to linguistic features extracted with a shallow parser. This corpus-based similarity is then taken as the probability for the equivalence of the concepts in question.

¹ Technique for exploiting certain lexico-syntactic patterns to discover is-a relations between two given concepts

The final output of OL module is an LO's content Ontology represented in OWL (Web Ontology Language).

4.1.4 The Ontology Mapping, Merging and Alignment

Ontology mapping is the process, which needs to be done after successfully constructing new Ontology (semi-)automatically from the uploaded LO. The aim of this Ontology mapping is to conform, establish correspondences and determines the set of overlapping concepts, concepts that are similar in meaning but have different names or structure, and concepts that are unique to each source, between the new Ontology with the existing Ontology in knowledge base. This work is either done to create a single coherent Ontology that includes the information from all the sources (*merging*) or if the sources must be made consistent and coherent with one another but kept separately (*alignment*). Figure 4-6 illustrates the difference between Ontology Mapping, Ontology Merging and Ontology Alignment; the yellow and purple box represents two different Ontologies.

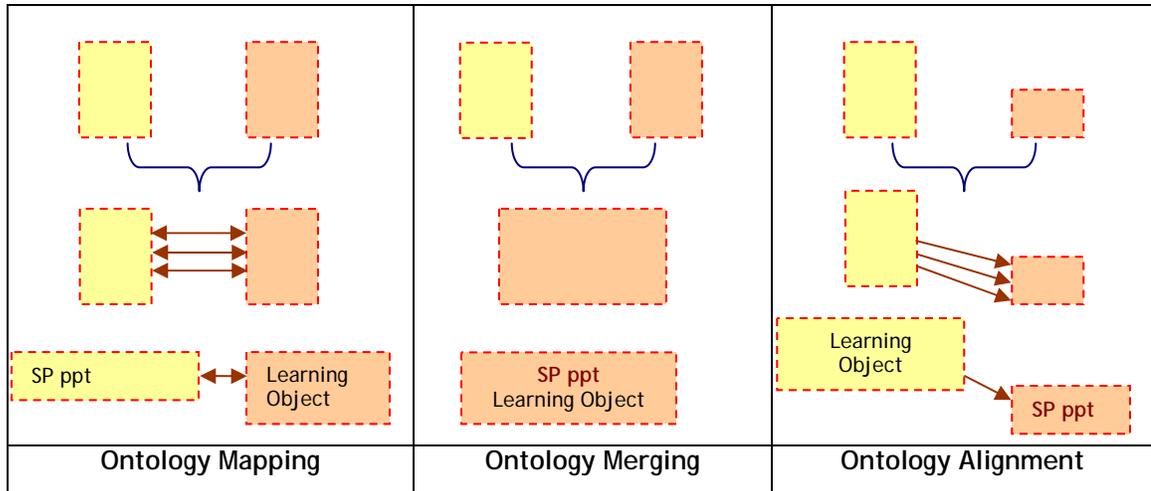


Figure 4- 6 Ontology Mapping, Merging, and Alignment

Because there are two kinds of Ontology in this SAOKBCS, the Ontology Mapping in our process consists of two folds. The first fold is Context Ontology Mapping and the second fold is content Ontology Mapping. The context Ontology refers to Ontology that model anything describing the LO in e-Learning environment. The content Ontology

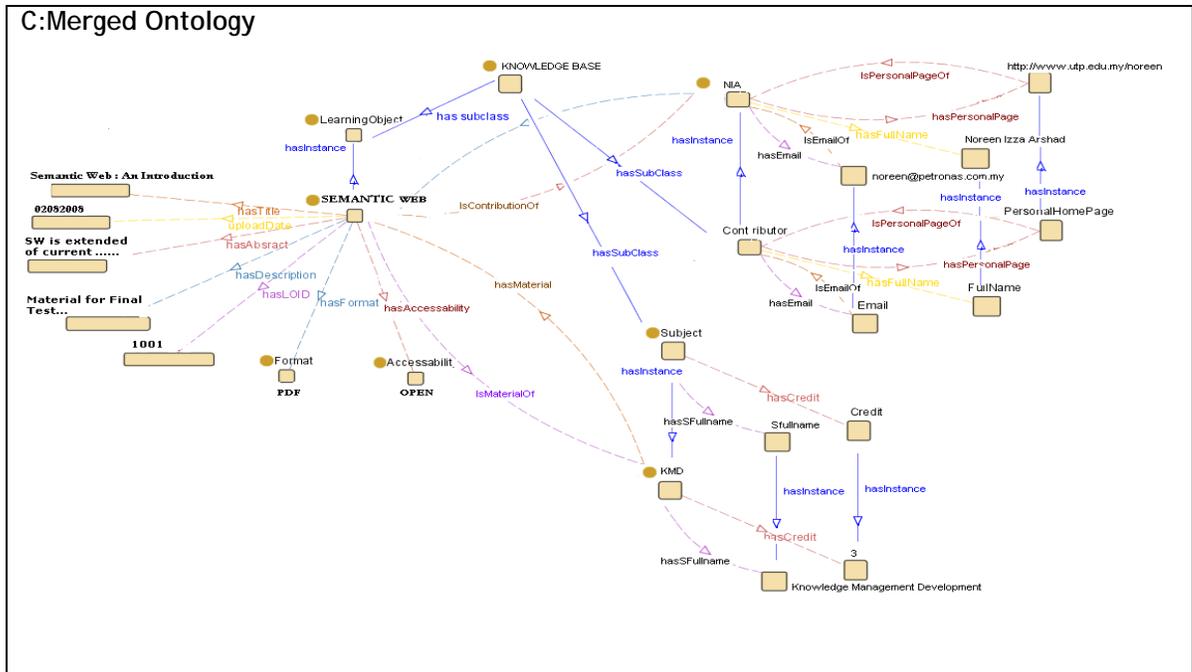


Figure 4- 7 Context Ontology Merging Case Scenario

The part A of Figure 4-7 is a new Ontology which is constructed automatically from the uploaded LO i.e. text Learning Materials, part B is the existing Ontology in knowledge base before a new Ontology (part A) is added, and part C is the Ontology in knowledge base after a new Ontology (part B) is added. Part C is showing how the new Ontology (part A) is to be mapped to the existing Ontology. As depicted in Figure 4-7 Part C, it is found that *Semantic Web* is mapped as subclass of *Learning Object*, *NIA* is mapped as instance of *Contributor*, and *KMD* is mapped as instance of *Subject*.

The second part of Ontology mapping, merging, and alignment is under taken for content Ontology. To doing so, PROMPT framework is adopted to our SAOKBCS. The PROMPT algorithm is depicted by Figure 4-8.

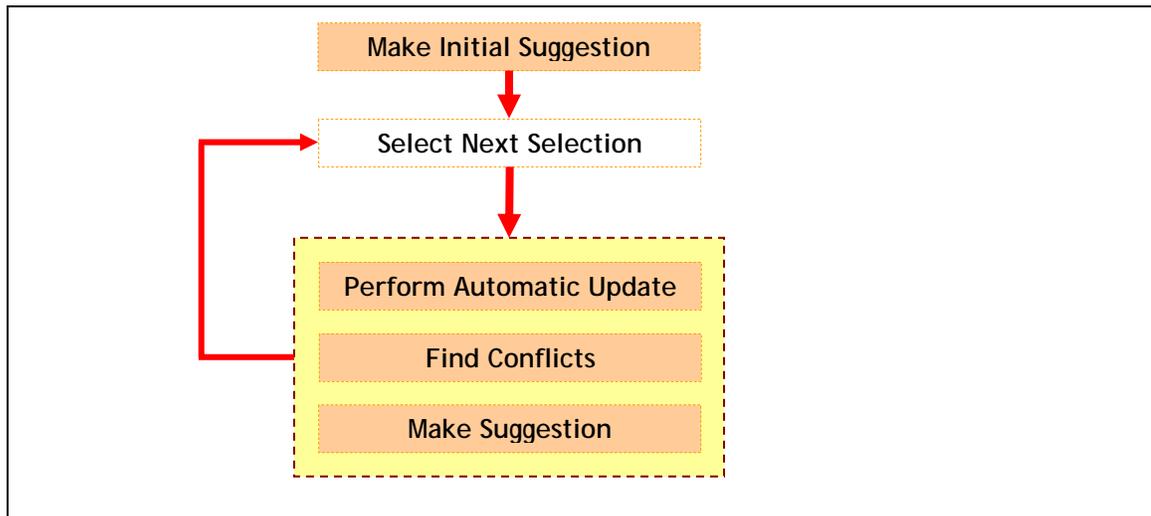


Figure 4- 8 PROMPT Algorithm (Natalya et al. 2000)

Figure 4-8 illustrates the PROMPT Ontology-merging and Ontology-alignment algorithm. PROMPT takes two Ontologies (the new Ontology and the existing Ontology in knowledge base) as input and guides the user in the creation of one merged Ontology as output. First PROMPT creates an initial list of matches based on class names. Then the following cycle happens:

- (1) The user triggers an operation by either selecting one of PROMPT’s suggestions from the list or by using an Ontology-editing environment to specify the desired operation directly; and
- (2) PROMPT performs the operation, *automatically* executes additional changes based on the type of the operation, generates a list of *suggestions* for the user based on the structure of the Ontology around the arguments to the last operation, and determines *conflicts* that the last operation introduced in the Ontology and finds possible solutions for those conflicts.

The set of Ontology-merging operations that Ontology mapping module done consists operations such as: merge classes, merge slots, merge bindings between a slot and a class; perform a deep copy of a class from one Ontology to another (includes copying all the parents of a class up to the root of the hierarchy and all the classes and slots it refers to);

perform a shallow copy of a class (just the class itself, and not its parents or the classes and slots it refers to).

For example, suppose in the case of merging two Ontologies and it performs a merge-classes operation for two classes A and B to create a new class M. PROMPT then performs the following actions:

- (i) For each slot S that was attached to A and B in the original Ontologies, attach the slot to M with the same value type and other facets. If S did not exist in the merged Ontology, create S.
- (ii) For each *superclass* of A and B that has been previously copied into the merged Ontology, make that copy a *superclass* of M (thus restoring the original relation). Do the same for subclasses.
- (iii) For each class C in the original Ontologies to which A and B referred (that is, for each *superclass*, *subclass*, slot value, and class restricting a slot value of A and B), if C has not been copied to the merged Ontology, suggest that it is copied to the merged Ontology.
- (iv) For each class C that was a facet value for A or B and that has not been copied to the merged Ontology, declare a dangling-reference conflict.
- (v) For each pair of slots for M that have linguistically similar names, suggest that the slots are merged. Later, if the user chooses to merge the slots, suggest that the classes restricting the values of these slots are merged as well.
- (vi) For each pair of *superclasses* and *subclasses* of M that have linguistically similar names, suggest that they are merged: these classes have similar names and, in addition, they were both superclasses (or subclasses) for A and B, which the user declared to be similar.
- (vii) Check for redundancy in the parent hierarchy for M: If there is more than one path to any parent of M (other than the root of the hierarchy), suggest that one of M's parents is removed.

4.2 Automated Question Answering System (Aquas)

The second fold of the main issue tackled in the proposed SWA-KMDLS is knowledge reusing and LO retrieval in e-Learning environment. For knowledge reusing and LO retrieval issue an ontological approach for Automated question answering system (Aquas) is proposed. It let users find knowledge they need easily in their natural language (English) than traditional keyword based search engine. The idea of this Aquas is illustrated in Figure 4-9.

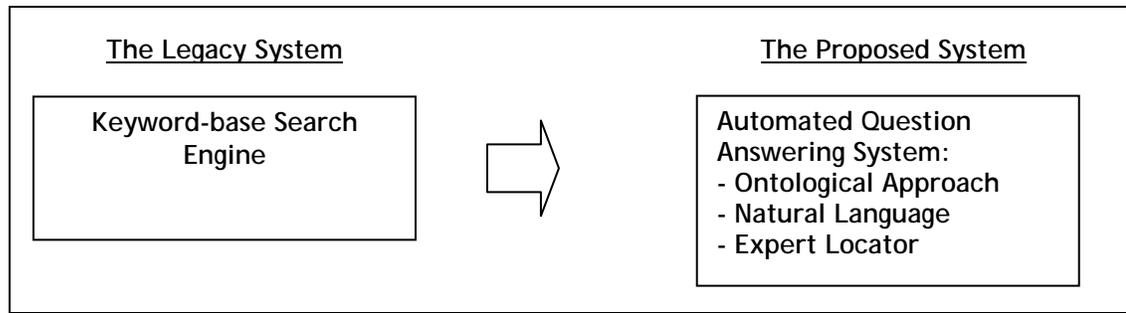


Figure 4- 9 Proposed Aquas for Knowledge Reusing and Sharing Framework in e-Learning Environment

Principally, Aquas is similar to information retrieval that has been a popular research area for along years ago. The input of this automated question answering system is expressed in Natural Language (i.e. English) instead of key word based that commonly used in information retrieval system. Aquas is designed to be consisting of two automated question answering system:

- (i) Statistical-based Automatic Question Answering System (S-Aquas).
- (ii) Ontology-driven Automatic Question Answering System (Onto-Aquas).

The former is intended to Information Retrieval (IR) that will draw answer based on traditional information base (i.e. FAQ database and LO database) and the latter is intended to Knowledge Retrieval (KR) that will draw answer based on proposed

knowledge base. The differences of these two approaches will be described in the following sub section (§ 4.2.1 and § 4.2.2).

4.2.1 Statistical-based Automatic Question Answering System (S-Aquas)

The heart of the S-Aquas is the concept of Term Frequency x Inverse Document Frequency (TFxIDF) and Similarity Comparison Algorithm. The Term Frequency (TF) measures how well an index term describes the document content by assigning weight to the term that frequently occur in a document. The Inverse Document Frequency (IDF) on the other hand measures how well the index term discriminates between relevant and non-relevant documents in the collection by giving high weights to rare terms. The original TFxIDF formulas is given as presented in the following equation 4-1 and equation 4-2 (Manning et al. 2008):

$$R_{i,q} = \sum_{termj \in q} \left(0.5 + 0.5 \frac{TF_{i,j}}{TF_{i,max}} \right) IDF_j \quad \text{Eq.4- 1}$$

$$IDF_j = \log \left(\frac{N}{\sum_{i=1}^N C_{i,j}} \right) \quad \text{Eq.4- 2}$$

Where D_i is i -th document for $1 < i < N$, Q_j is j -th query term in q for $1 < j < M$, TF_{ij} is number of occurrence for Q_j in D_i , $TF_{i,max}$ is maximum number of occurrences for the key term in D_i and $C_{i,j}=1$ if D_i Contains Q_j , and $C_{ij}=0$ otherwise.

The backbone of S-Aquas are the keyword base, learning resource base and traditional (non-ontological) knowledge base. Keyword base is constructed from any keywords that describe any important concept either in learning resource base or traditional knowledge base. Learning resource base is constructed from learning materials uploaded into e-Learning environment and its corresponding Characteristic Vector (CV). Traditional knowledge base is constructed from FAQ database and its corresponding CV. CV is a

vector of keyword probability occurrence in the domain (i.e. LO or FAQ database) which is constructed by using *TFxIDF* algorithm as presented in equations (4.1) and (4.2).

For the similarity comparison algorithm, in this framework either Inner Product (IP) or Euclidian Distance (ED) algorithm is adopted. Suppose that user Question is Q , i Documents (D), and n Keywords (k), the similarity between Q and D can be computed using either IP or ED formula which is presented in equation (4.3) and (4.4) respectively.

$$D_i = \sum_{k=1}^n W_k W_{ik} \quad \text{Eq.4- 3}$$

$$D_i = \sqrt{\sum_{k=1,n} (W_k - W_{ik})^2} \quad \text{Eq.4- 4}$$

W_k is weight of keyword k in Q , and W_{ik} is weight of keyword k in D . Weight of keyword represent the probability occurrence of k in Q or D which can be computed using simplified *TFxIDF* algorithm as presented in equation (4-5).

$$W_{ij} = \frac{f_{ij}}{\sum_{k=1}^m f_{kj}} \quad \text{Eq.4- 5}$$

Where K_j represents j th keyword and f_{ij} is the number of occurrences of K_j in Question Q_i , $\sum_{k=1}^m f_{kj}$ is the total number of occurrences of K_j in the FAQ database. How the CV and similarity algorithm are implemented will be clearly explained in development phase (Chapter Five).

The framework of the proposed S-Aquas is depicted by the following Figure 4-10.

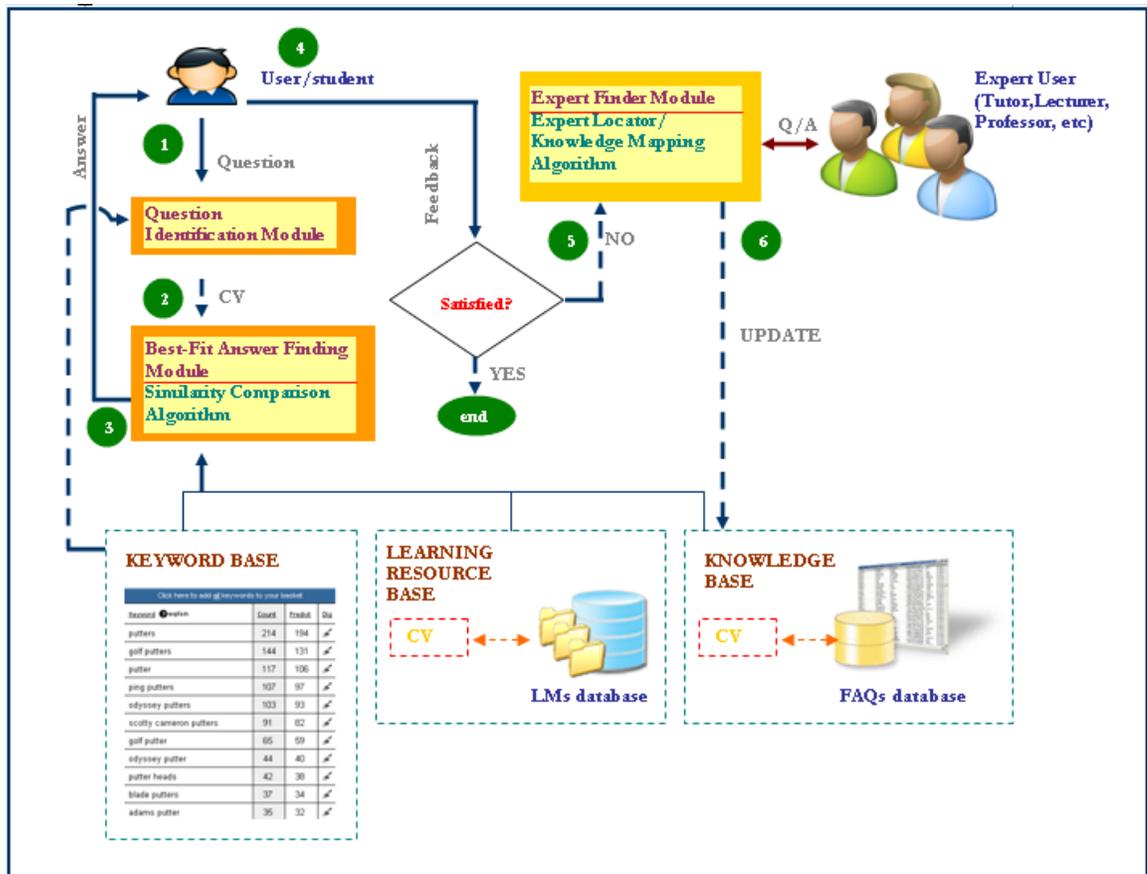


Figure 4- 10 S-Aquas Framework

The framework of the proposed S-Aquas depicted by Figure 4-10 concisely can be explained as follows:

- (1) User enter question expressed in natural language (English), differ from traditional search engine which is keyword based since the S-Aquas can handle input in natural language (English). Accordingly, invoke the Question Identification Module. It extracts the question entered by user into CV describing the question by decomposing the question into several terms defined in keyword database.
- (2) Invoke the Best-Fit Answer Finding Module to select the best answer from knowledge base-consist of FAQs database and corresponding CV-and select the most relevant learning materials (LMs) from learning resource base-consist of LMs and their corresponding CV by using similarity comparison algorithm.
- (3) Reply the answer and relevant learning material(s) to the user.
- (4) User gives feedback whether the answer given is satisfied or unsatisfied.

- (5) If the user is not satisfied with the answer, the S-Aquas invoke the expert finder module (expert locator). This module find user (i.e. tutor, lecturer) who is the expert according to the question area and send the question to the expert user. The expert finder module will be detailed in the following section (§ 4.4).
- (6) Once the question answered by the expert user the FAQs database will be updated, such that the S-Aquas will be able to answer the similar question automatically.

4.2.2 Ontology-driven Automatic Question Answering System (Onto-Aquas)

The key different between Onto-Aquas and S-Aquas is where the answer is drawn from. In Onto-Aquas, the answer is drawn from ontological knowledge base instead of traditional knowledge base. The Onto-Aquas framework is depicted in Figure 4-11. In this framework Aqualog data model (Lopez et al. 2007) is adopted.

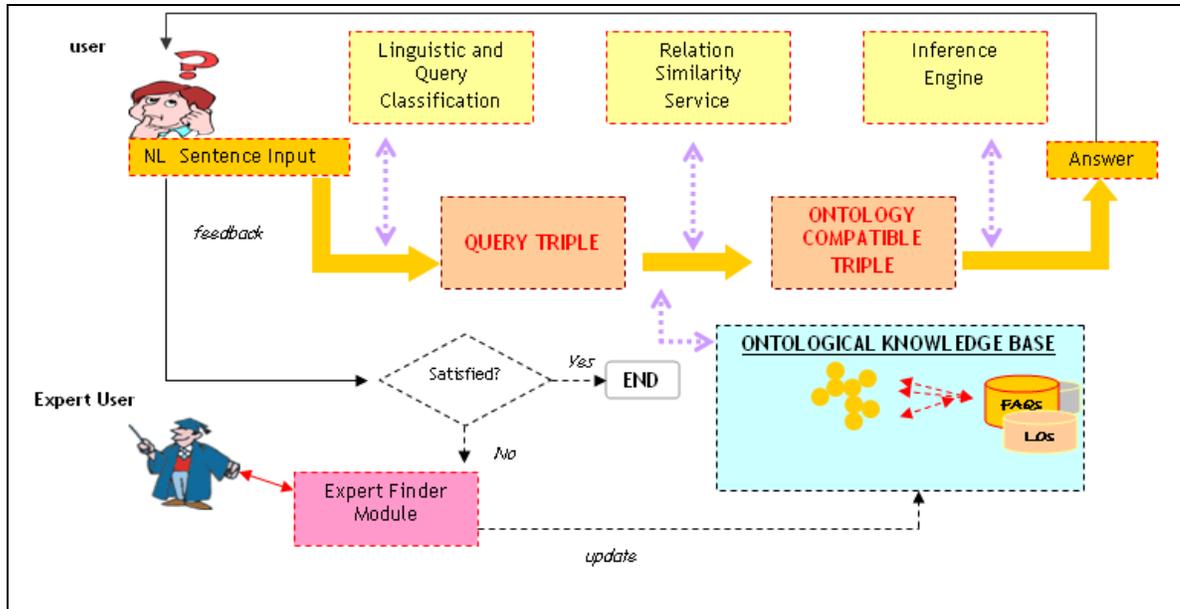


Figure 4- 11 Onto-Aquas Framework

At a coarse-grained level of abstraction, the Onto-Aquas framework can be described as follows:

- (1) User input a question using Natural Language (NL) expression (i.e. English) instead of keywords as in traditional search engine.
- (2) NL query is translated by the *Linguistic Component* into a set of intermediate triple-based representations, which are referred to as the Query-Triples.
- (3) Then, *the Relation Similarity Service (RSS)* component takes as an input these Query-Triples and further processes them to produce the Ontology-compliant queries from ontological knowledge base, called Onto-Triples.
- (4) Finally, the *inference engine* will infer from the Ontology compatible triple into compatible and relevant answer.
- (5) To enhance the performance of Onto-Aquas, the system will ask for feedback from user to know whether user is satisfied with the answers.
- (6) If the user is unsatisfied, an expert finder module will be invoked. The duty of expert finder module is to find e-Learning user which is an expert in the area of user question (see §4.4).
- (7) Finally, responses from expert user will update the ontological knowledge base.

There are some aspects that are important to be noted in Onto-Aquas framework such as the data model. The data model is triple-based; it takes the form of (subject, predicate, object). There are two main reasons for adopting a triple-based data model. First of all, as pointed out by (Katz et al. 2002), although not all possible queries can be represented in the binary relational model, in practice these exceptions occur very infrequently. Secondly, RDF-based knowledge representation (KR) formalisms for the SW, such as RDF itself or OWL also subscribe to this binary relational model and express statements as (subject, predicate, object).

Hence, it makes sense for a query system targeted at the SW to adopt a triple-based model that shares the same format as many millions of other triples on the SW.

4.3 Expertise Finding and Mapping Framework

The third fold of the main issue tackled in the proposed SWA-KMDLS is knowledge sharing in e-Learning environment. For knowledge sharing issue an ontological expert expertise finding and mapping framework (expert locator) is proposed. This expert locator framework is intended to either improve the performance of Aquas in future use or push knowledge sharing culture in e-Learning system environment. The purpose of this expert finding and mapping framework is to locate user in the e-learning environment, which is an expert with the problem being asked in the automated question answering system. Thus, encourage expert user to share their expertise and knowledge. The framework of this Ontology-based expert locator is depicted in Figure 4-12.

4.3.1 Expert Locator Framework

The heart of our expert locator framework is an Ontology that models the knowledge of expert users in e-Learning environment (see Figure 4-12). Detail of this expert user Ontology is discussed in § 4.3.2.



Figure 4- 12 Expert Locator Architecture

The framework of expert locator module in Figure 4-12 concisely can be explained as follows:

- *Source layer*: this layer contains data sources that are relevant in identifying the expert user (professor, lecturer, assistant/tutor). This layer consists the Learning Materials (LMs), technical report, publications authored by expert them, e-Learning Management System (LMS) database that store information about courses taught by them, and their personal homepage which includes personal contact information, research interest, associated research group(s), and recent publication.
- *Mediating layer*: this layer maintains conceptual model i.e. expert Ontology and identifies which data sources are relevant to the query. The Ontology is depicted in Figure 4-13.
- *Query interface*: this layer translated query from AQUAS (Automatic Question Answering System: S-Aquas and Onto-Aquas) into Ontology query language i.e. OWL-QL (Web Ontology Language-Query Language) and return the result to AQUAS.
- *Application layer*: this layer is AQUAS.

4.3.2 Expert User Ontology

The purpose of developing expert user Ontology is to model how the expert users are represented and how to find an expert user in the e-Learning environment. Figure 4-13 depicted the graphical view of the expert user Ontology that is developed using Protégé.

expert finder framework (expert locator) to encourage knowledge sharing between expert users in e-Learning environment. The following chapter explains how this design level is developed in the application level.

CHAPTER 5

DEVELOPMENT AND IMPLEMENTATION

This chapter describes the proof-of-concept, prototype development and implementation based on proposed design framework and architecture in the Chapter Four. Firstly, the environment where the proposed SWA-KMDLS is implemented will be presented in section 5.1 and followed by initial Ontology development, the backbone of the SWA-KMDLS presented in section 5.2. The implementation of learning object management with SAOKBCS is presented in section 5.3. Then, section 5.4 and 5.5 present the implementation of Aquas and expertise finding and mapping framework (expert locator) respectively.

5.1 The Implemented Environment: Moodle e-Learning Management System

The proposed systems are implemented in an e-Learning (Course) Management System so-called Moodle (Moodle 2008). Moodle is chosen because it is a free and open source software package designed using sound social constructivist pedagogical principles in which knowledge is constructed through interaction with one another and with learning materials in a social way. Its open source state makes it easy to be modified and reconstructed. The Moodle front page is depicted in Figure 5-1.

5.1.1 Modular Approach in Moodle

The word Moodle was originally an acronym for Modular Object-Oriented Dynamic Learning Environment. Modular means that Moodle is developed by divided it under some modules. Moodle is designed using object oriented approach that has merit in reusing ability and easing modification. To add functionality in Moodle, it is easily can be made by create a new module that is separated from another module of another functionalities. Thus, it can avoid disrupting functionalities. Discussion on how to create new module in Moodle is presented in the following section (§ 5.1.2).

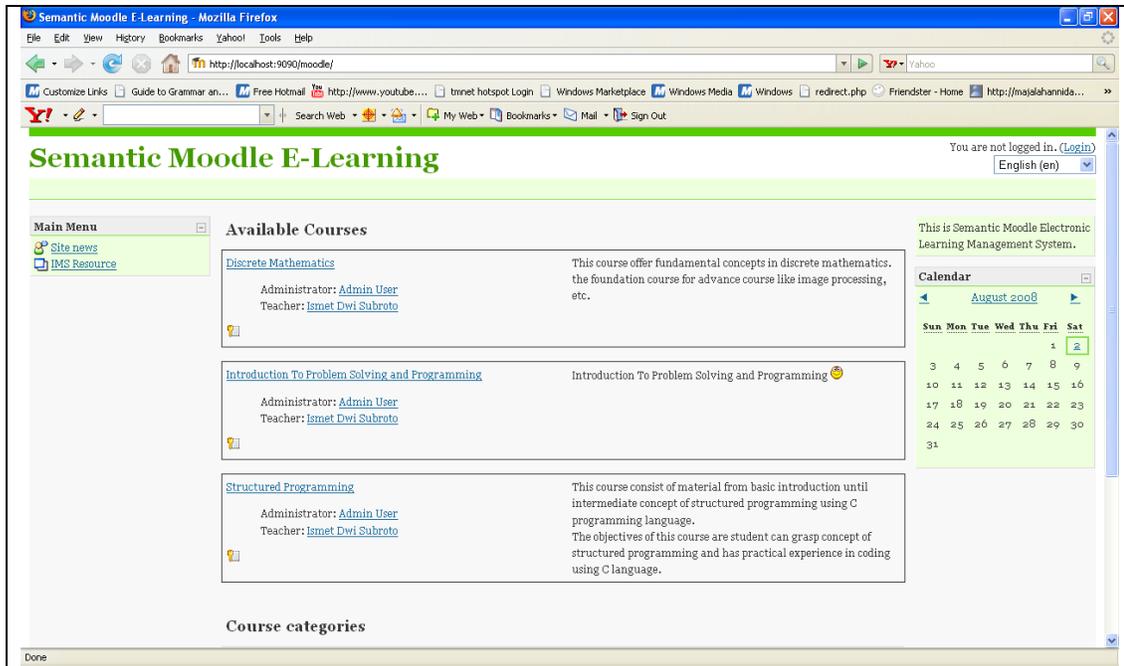


Figure 5- 1 Moodle LMS Front Page

5.1.2 Creating a New Module in Moodle

Its modular state in which all functionalities are represented in a specific module, make Moodle easily to be modified or added with new functionalities. Creating a new module can be done easily by creating a new folder inside folder *mod* in the moodle folder (see Figure 5-2). Each module must have some mandatory file as presented in Table 5-1.

Table 5- 1 Compulsory Files in Each Module in Moodle

Name of File	Description
mod.html	A form to setup/update a module instance.
version.php	Defines some meta-info and provides upgrading code.
icon.gif	A 16x16 icon for the module.
db/mysql.sql	An SQL dump of all the required database tables and data.
index.php	A page to list all instances in a course.
view.php	A page to view a particular instance.
lib.php	Any/all functions defined by the module should be in here. Constants should be defined using <code>MODULENAME_XXXXXX</code> and each function should be defined using <code>modulename_XXXXXX</code> .

The mod folder usually located under directory moodle as depicted by Figure 5-2.

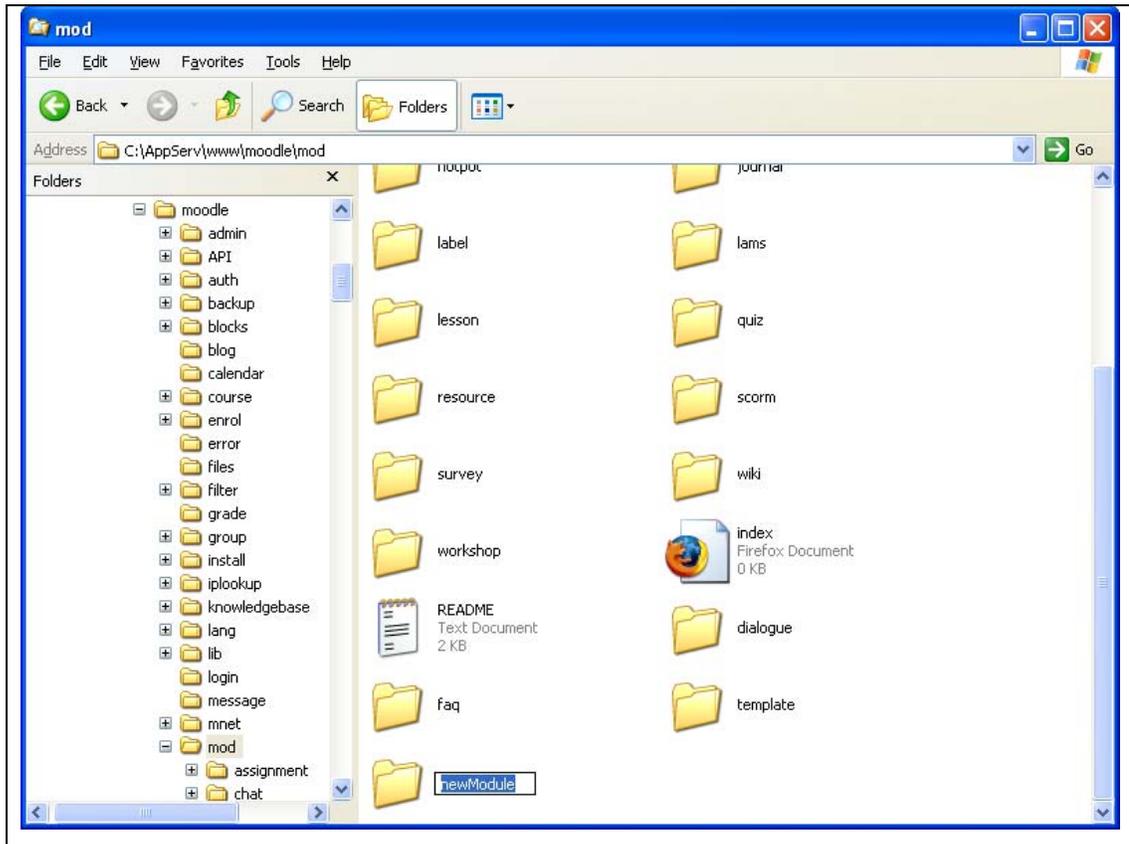


Figure 5- 2 Creating a New Module in Moodle

5.2 Initial Ontology Development (Ontology Population)

As mentioned in prior chapter, the backbone of the proposed SWA-KMDLS is Ontology development. There are two main purposes of Ontology development in the proposed system:

- (1) To model the general concepts in the e-Learning system environment (in our SWA-KMDLS Moodle LMS is adopted), it is named as *Domain Ontology*.
- (2) To model the knowledge from learning material/object, it is named as *Application Ontology*.

At the beginning of Ontology development lifecycle, there is a need to initiate the initial Ontology for both types of Ontologies *Domain Ontology* and *Application Ontology* as it has been explained in § 4.1.1.1. This initial Ontology development also known as *Ontology population*. The initial Ontology is developed manually based on guideline as stated in METHONTOLOGY methodology (chapter three) using Protégé. Protégé is a free, open source Ontology editor and knowledge base framework from Stanford University called Protégé (Protégé 2008) . The Protégé especially its Protégé-OWL editor feature that supports the OWL code generation is required to develop the Ontology from scratch. OWL is the Ontology representation language used in this research work because it is the most recent developed Ontology language standard endorsed by the W3 Consortium to promote the SW vision.

In addition to Ontology development tool in which the latest version of Protégé i.e. Protégé 4.0 is used, for Ontology visualization, OWL Viz (Horridge 2005) and Jambalaya (Storey et al. 2002) Plugin are used. To evaluate the developed Ontology reasoning tool called FaCT++ (Tsarkov and Horrocks 2006) and Pellet 1.5 (Clark and Parsia 2006) are also used. The Protégé 4.0 screenshot is depicted by Figure 5-3.

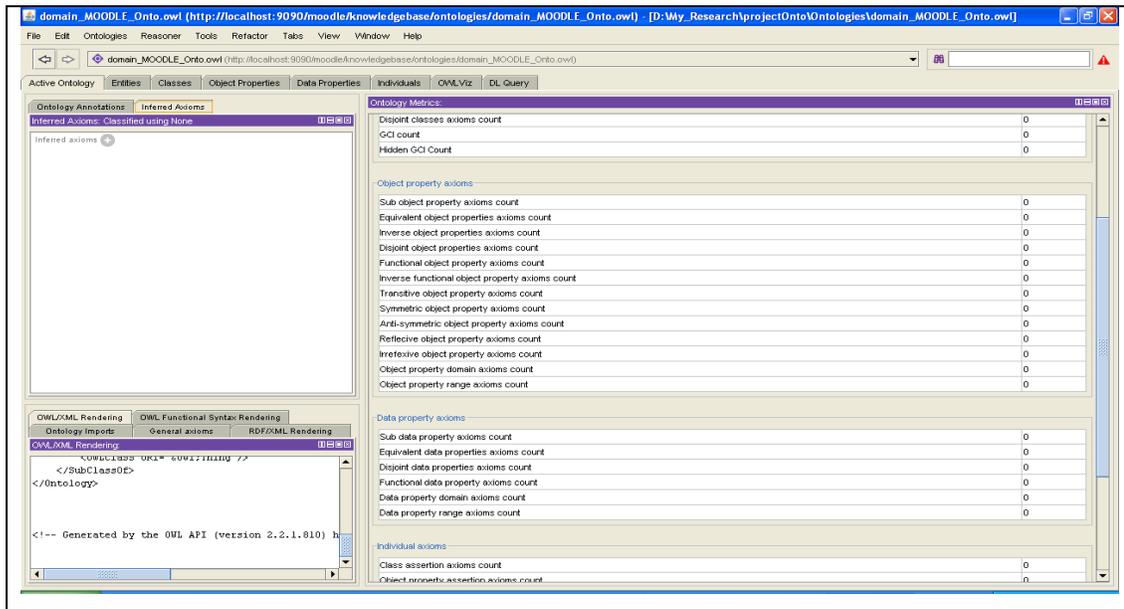


Figure 5- 3 Protégé –OWL 4.0 Screenshot

Details Ontology development process in this research is presented in (Mukhlason et al. 2008). The next section presents the two types of initial Ontologies i.e. *Domain* Ontology and *Application* Ontology is presented in § 5.2.1 and §5.2.2 respectively. The Ontology testing that will evaluate the developed Ontologies using the reasoning tool is presented in § 5.2.3.

5.2.1 Initial Domain Ontology Development

The *Domain* Ontology development aims to model the SWA-KMDLS e-Learning environment that adopted from Moodle LMS. The graphical representation of this domain Ontology is depicted by Figure 5-4 and Figure 5-5 that are generated by using OWL Viz Plugin.

Figure 5-4 depicts Domain Ontology in class hierarchy diagram in which “is a” relation mean that the corresponding classes or concepts has superclass–subclass relationship. For example, “Assignment” is subclass of “Activity” and vice versa, “Activity” is superclass of “Assignment”.

Figure 5-5 depicts Domain Ontology in domain-relation-range diagram. Each relation in Ontology has particular domain and range. For example, “Journal” and “JournalName” is related by “hasJournalName” relation. It means that for relation “hasJournalName”, the domain is “Journal” and the range is “JournalName”. More detail of the Figure 5-4 and Figure 5-5 can be seen in Appendix IV.1 and IV.2 respectively. The OWL representation of this domain Ontology is depicted in Appendix IV.3.

As depicted by Figure 5-4 the main concepts (or class) in Domain Ontology consists of User of e-Learning, Courses and Activities. User of e-Learning refers to user who has authority to access courses in the LMS, which has hierarchical level such as administrator, lecturer, tutor, student, etc. Courses refer to subject that is available to be accessed in e-Learning System. Each course has some activities and resources, which are the learning material for the pertaining course.



Figure 5- 4 Graphical Representation of Domain Ontology in Class Hierarchy Diagram



Figure 5- 6 Graphical Representation of Application Ontology in Class Hierarchy Diagram

As depicted by Figure 5-6, the Application Ontology, which is IPSP Ontology, consists all fundamental concepts in Programming such as variable, data type, function, statement, operator, etc. The Ontology also models how these concepts are related one to another such as operator includes binary, arithmetic, equality, logical, and unary. In other words, operator has subclass binary, arithmetic, equality, logical and unary.

5.2.3 Ontology Testing

After developing the Ontologies, the next step need to do is the Ontology testing. The purpose of the Ontology testing is to evaluate that the Ontologies are logically correct. In this research work, the Ontology testing is conducted by using two reasoning tools called FaCT++ and pellet. FaCT++ and pellet have been bundled with protégé 4.0 Alpha version. FaCT++ and pellet can be executed from Reasoner Tool in Protégé 4.0 (See Figure 5-7). Classification and Consistency testing is conducted using RACER Pro (See Figure 5-8). The result of the Ontology testing is depicted by Figure 5-9.

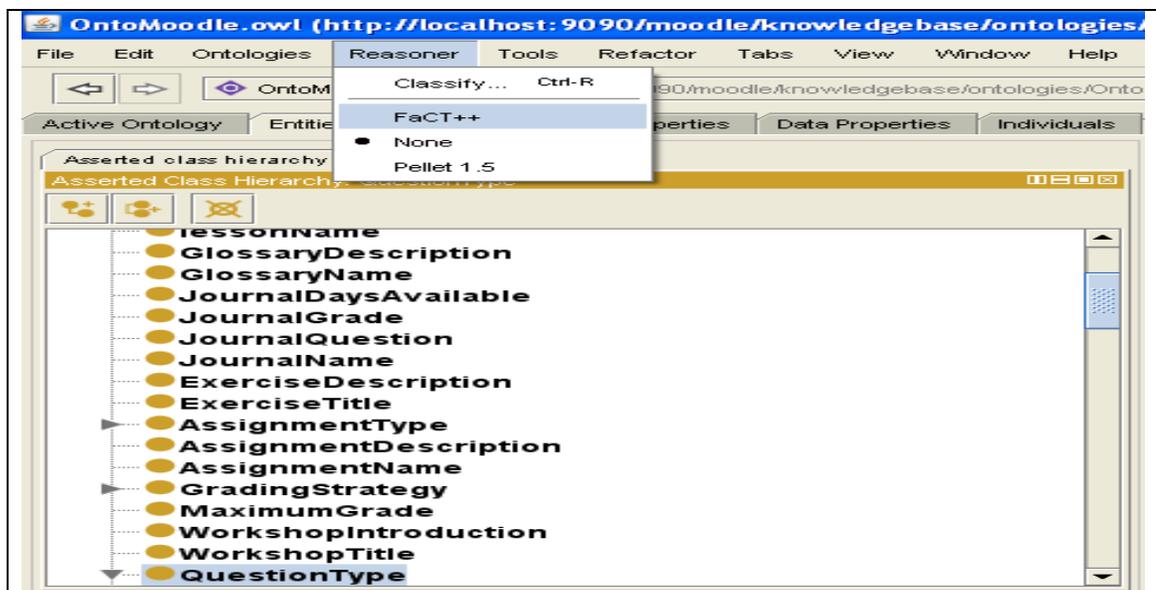


Figure 5- 7 Execute FaCT++ Reasoning Tool in Protégé

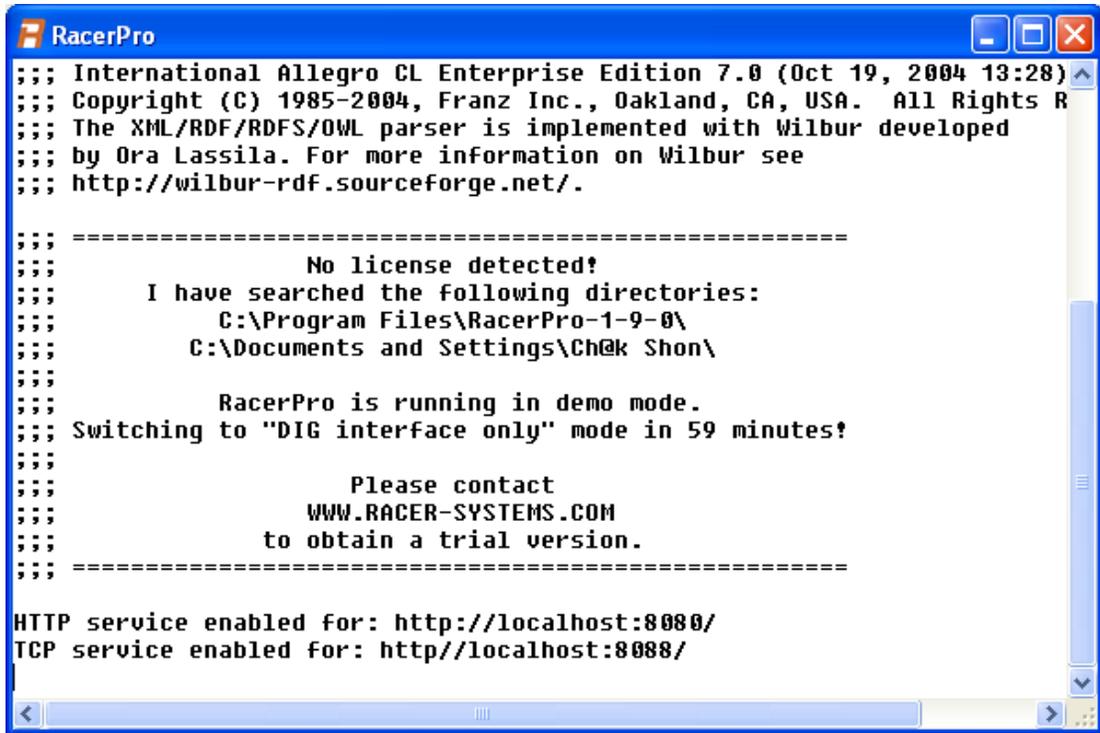


Figure 5- 8 Racer Reasoning Tool

As depicted by Figure 5-9, the result of Ontology testing using Racer Reasoner for either domain Ontology or application Ontology shows no error. The Ontology testing using FaCT++ also showed the same result. It means that the Ontology that was developed has been classified correctly and there is no inconsistency in the developed Ontology.

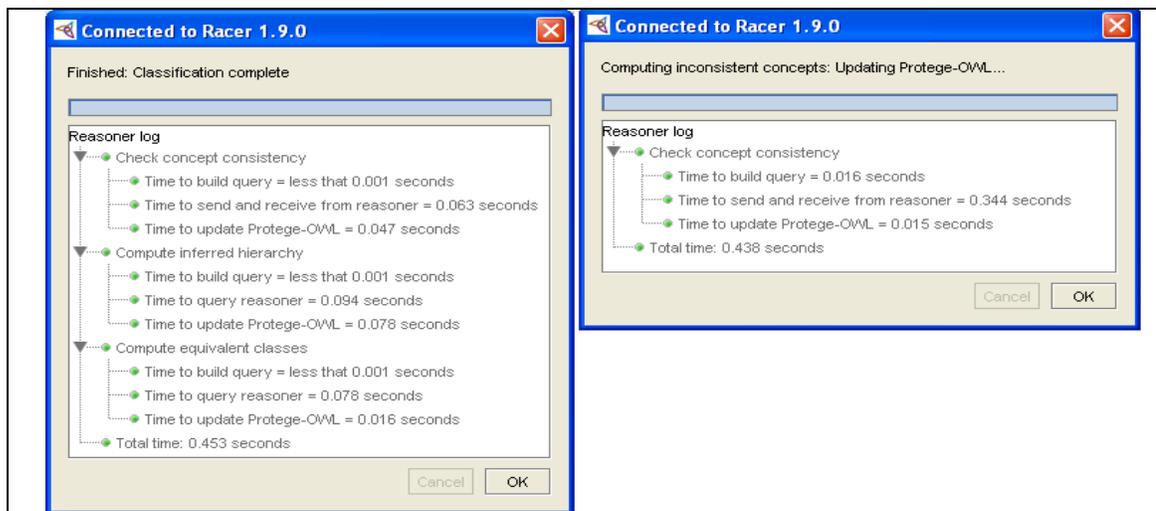


Figure 5- 9 Classification and Consistency Testing Using Racer

5.3 SAOKBCS Implementation

This SAOKBCS is the implementation of the proposed LO Management. This thesis introduces Ontological knowledge base to collect, organize, and share either the LO itself or the knowledge inside the LO. Ontological knowledge base is a knowledge base where the knowledge is represented in Ontologies. In this research work, the ontological knowledge base is constructed (semi-)automatically. This (semi-) automatic Ontology construction process usually called as *Ontology Learning* as explained in § 4.1.1.3.

The SAOKBCS starts working when a user (administrator, course creator, and teacher) in the e-Learning system adds or uploads a new learning resource in a particular course (refer to the SAOKBCS framework in Figure 4-2 in Chapter Four). For example, a Teacher of Introduction to Problem Solving and Programming (IPSP) adds a new File as a learning resource as depicted by the user interface section in Figure 5-10. Then the process of ontological knowledge base construction is taken place in the black box. What in the black box is explained in section 5.3.2, the algorithm and implementation section.

This SAOKBCS implementation section is divided into two parts. The first part focuses on User Interface (§ 5.3.1). In addition, the second part stresses on algorithm and its implementation (§ 5.3.2).

5.3.1 SAOKBCS User Interface

The user interface to add new LO is depicted in Figure 5-10. The LO (usually refers to learning material, learning resource) can be in the format of:

- Label, to insert text or other HTML elements into the content area.
- Text page, a simple plain- text page with little formatting.
- Web site.
- Link to file or Website.
- Directory of files.
- LMS content package, it is a packaging standard for educational content such as SCORM (Sharable Content Object Reference Model).

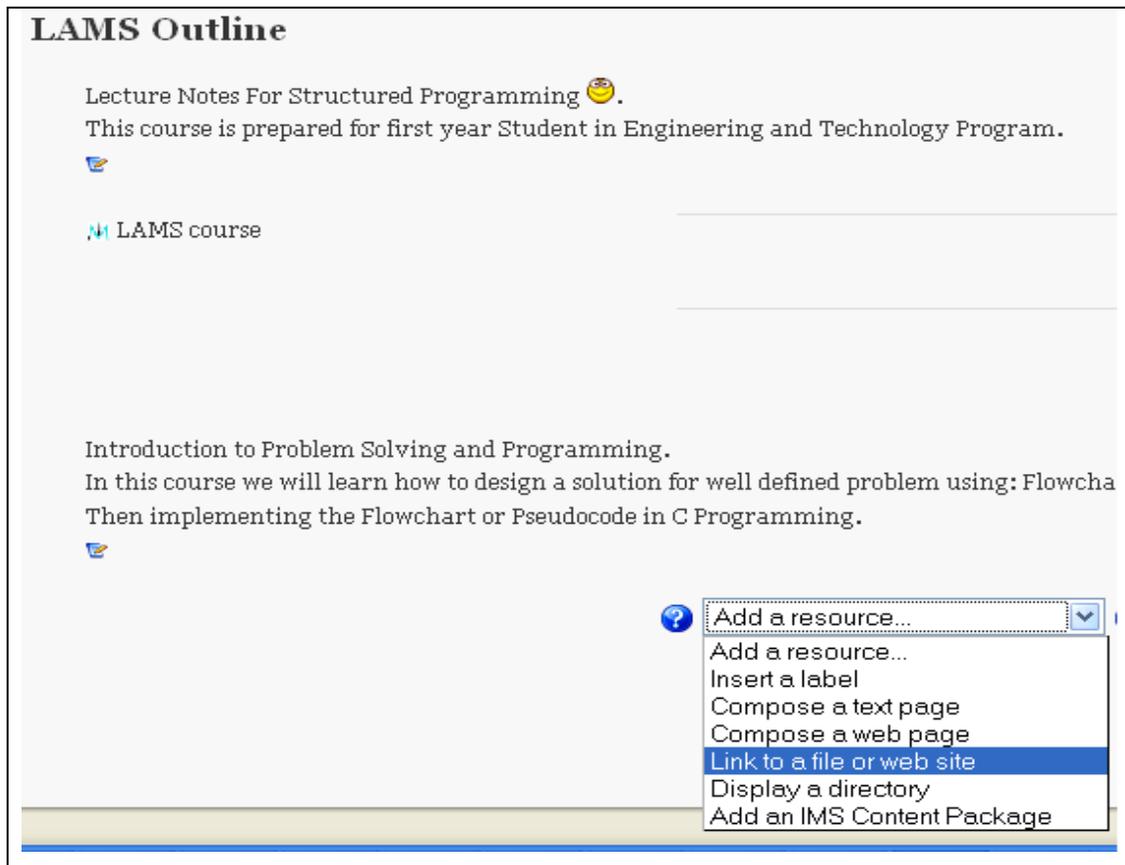


Figure 5- 10 Add New Learning Resource in IPSP course

Supposed an administrator or a course creator want to upload a new learning resource that is a link to a file, the user interface is depicted in Figure 5-11.

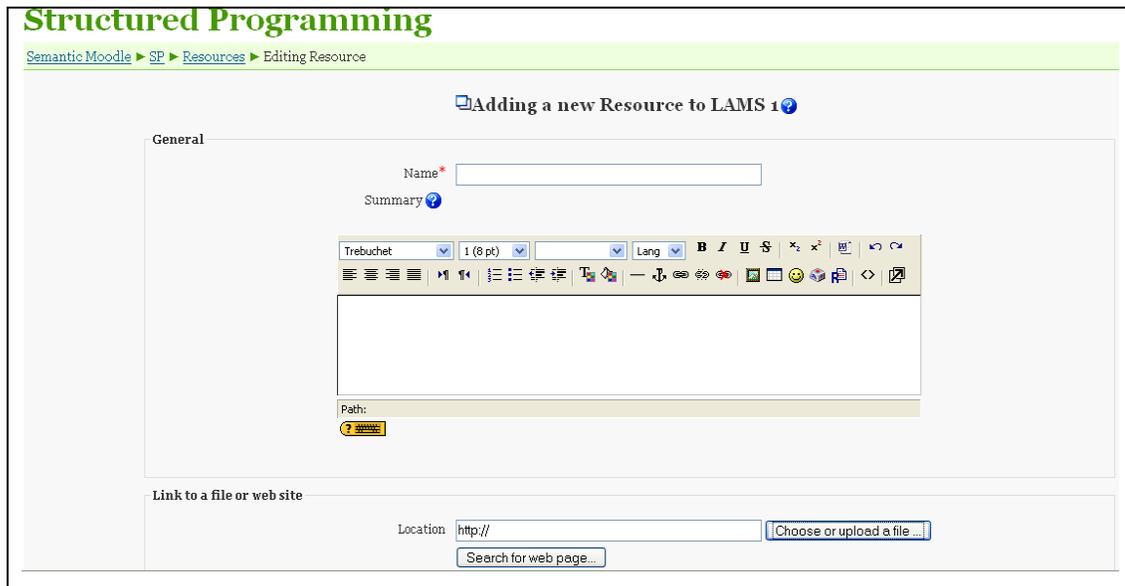


Figure 5- 11 Uploading New Learning Resources

5.3.2 SAOKBCS Algorithm and Implementation

The Ontological knowledge base which is constructed from learning resources consists of two kind of Ontology which are (i) *contextual* Ontology that model the context of the learning resources and (ii) *content Ontology* that model the concept inside the learning material. Therefore, as it is explained in the chapter four (§ 4.1.3) the *Ontology learning* process is consists of two steps. The first step is the contextual Ontology development, which is constructed automatically by *File Writer function* in PHP and the second step is content Ontology development that is constructed (semi-) automatically by *Text2Onto* from the uploaded learning material/object/resource. Finally as explained in the chapter four (§ 4.1.4) the new Ontology need to be mapped, merged, and aligned with the existing Ontologies. To do this Ontology mapping, merging, and alignment PROMPT tool is implemented. The algorithm of this (semi-) automatic Ontologies development or the Ontology learning is illustrated in *pseudocode* as shown in Figure 5-12.

```

1.0  Start
2.0  New Learning Resource Uploaded Successfully
3.0  Call php File_Open function to open the existing contextual
      Ontology
4.0  Call php File_Writer to edit and add entries in the existing
      contextual Ontology
5.0  The existing Contextual Ontology is updated
6.0  Close existing Contextual Ontology
7.0  Invoke Text2Onto function
8.0  Text2Onto read the learning resource
9.0  New Content Ontology is constructed
10.0 Invoke PROMPT function
11.0 The new Content Ontology and existing content Ontology are
      read by PROMPT
12.0 PROMPT results the mapped, merged, and aligned Ontology.
13.0 End

```

Figure 5- 12 Ontologies Construction Pseudocode

As depicted by Figure 5-12, constructing context Ontology from LOs process is implemented in line 3.0 – 6.0. The File_Writer is a function in PHP language to generate a file, which in this case to generate OWL file. Then, constructing content Ontology from LOs process is implemented in line 7.0 – 9.0 which is conducted by Text2Onto followed by Ontology mapping, merging, and alignment in line 10.0-11.0 which is conducted by PROMPT. Text2Onto and PROMPT are developed in JAVA language. These two different platform are integrated using *PHP-Java Bridge* (PJB 2008).

5.4 Aquas Implementation

As it has been explained in Chapter Four (§ 4.2), the Aquas consists of two categories of automated question answering approach. The first one is Statistical based automated question answering (S-Aquas); it is dedicated to answer trivial question from the user. Trivial question is a question that has been answered and stored in the FAQ database. The second one is Ontology-driven automated question answering system (Onto-Aquas); it is

dedicated for non-trivial question from user. Non-trivial question is a question that has not been answered and stored in the FAQ database, thus need to infer from learning resources. Then, to improve the performance of the Aquas for future use, the expertise finding and mapping framework is introduced (§ 5.5).

This Aquas implementation section is divided into two parts. The first part focuses on User Interface (§ 5.4.1) and the second part stresses on algorithm and its implementation (§ 5.4.2).

5.4.1 Aquas User Interface

The Aquas in SWA-KMDLS is implemented as an Activity for particular course in e-Learning environment. Thus, the Aquas can be course specific. The user interface to access the Aquas is depicted in Figure 5-13.

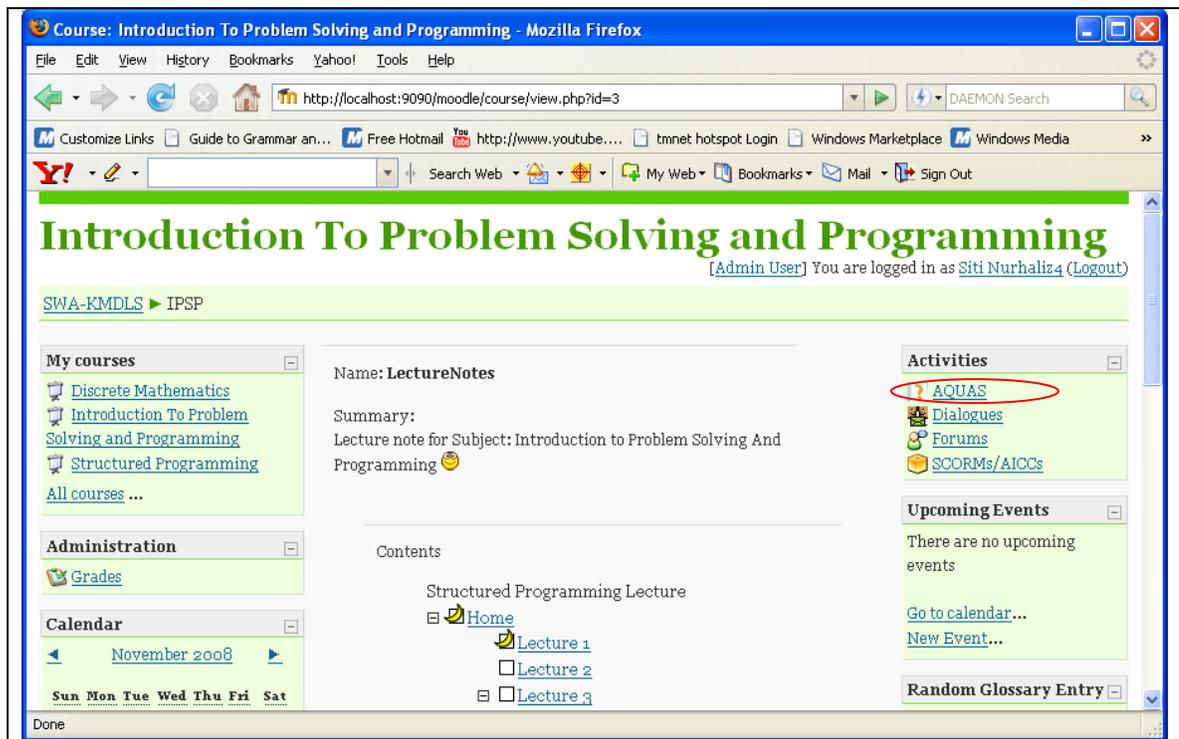


Figure 5- 13 User interface Aquas in moodle

The users key in their question in Natural Language that is English instead of keying in keyword as in common search engine (see Figure 5-14).

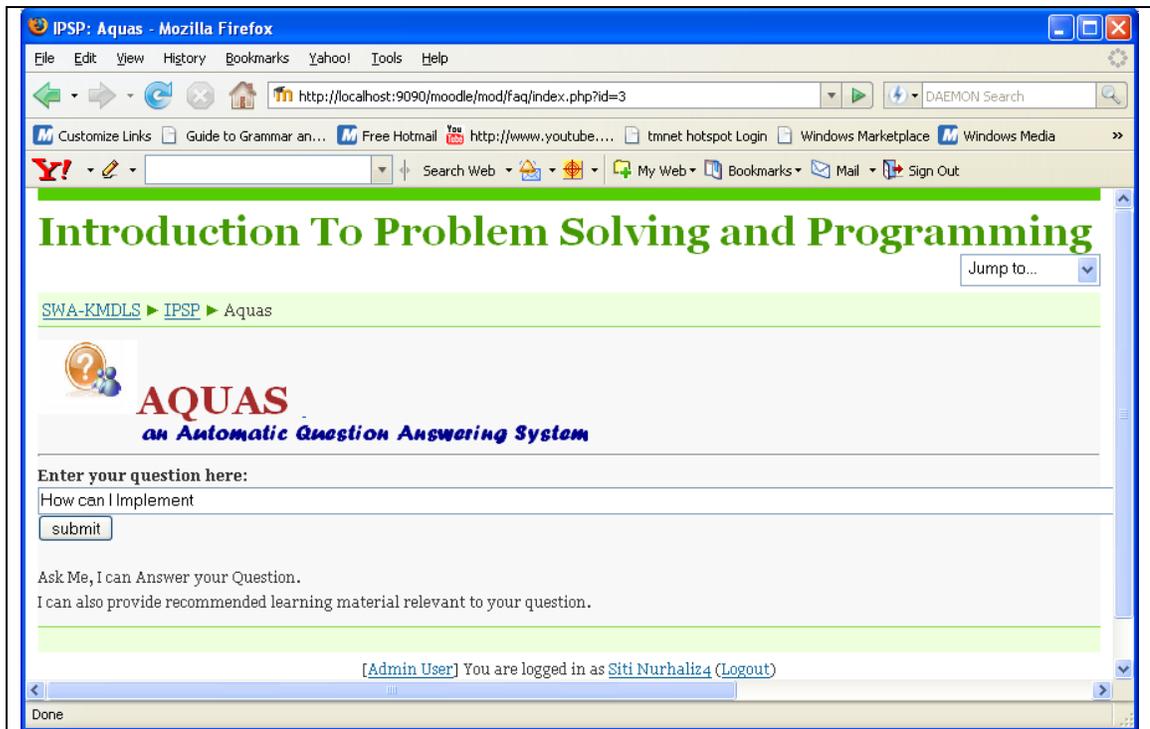


Figure 5- 14 User interface AQUAS – entering question

After the user submits the question, the system will automatically provide the answers. The system not only gives the answer, but also recommends the relevant LOs to user. To assess the user satisfaction on the answer given, users are asked to give their feedback on whether they are satisfied or not (See Figure 5-15).

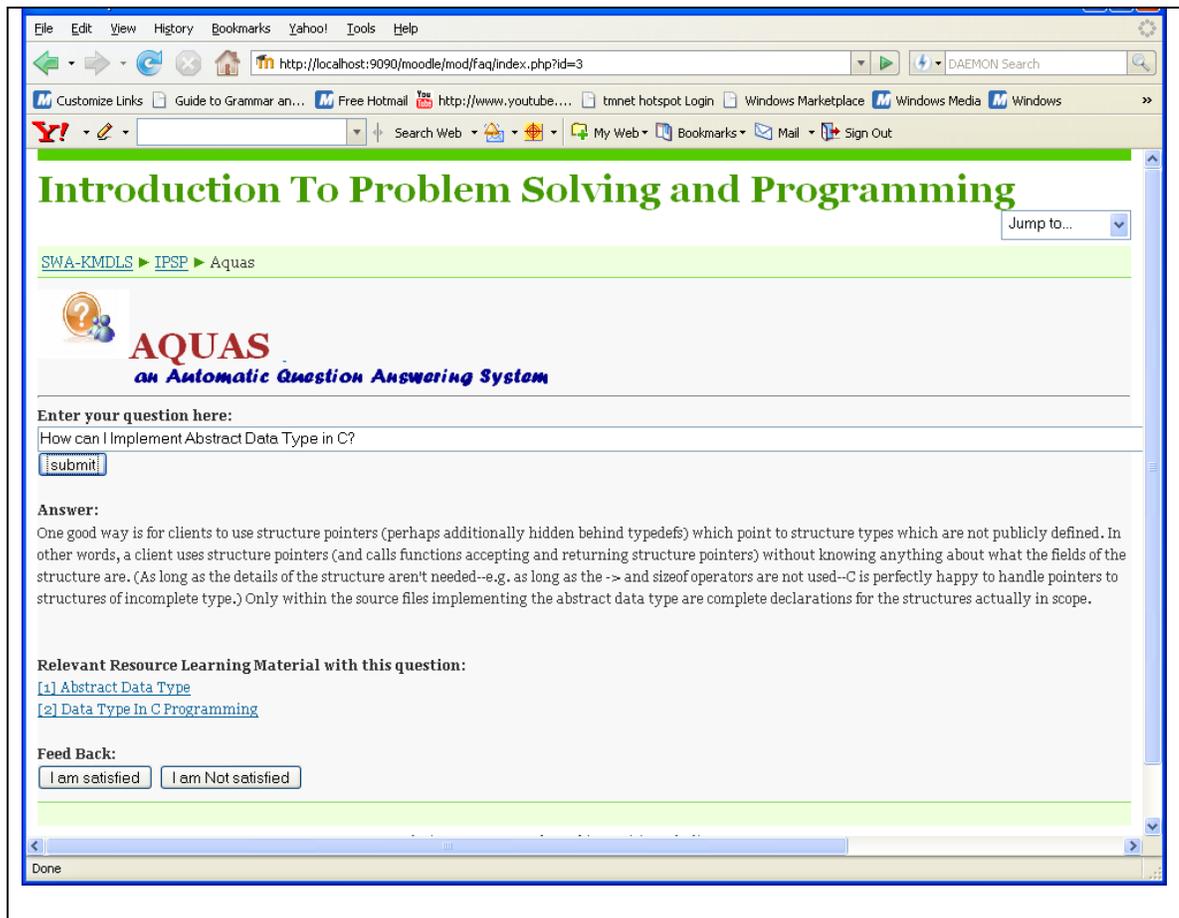


Figure 5- 15 User interface AQUAS – respond from system

5.4.2 Aquas Algorithm and Implementation

The Aquas algorithm, which is consisted of S-Aquas and Onto-Aquas, in nutshell, is presented by pseudocode in Figure 5-16. As depicted by Figure 5-16 for Natural Language Processing such as stemming, lemmatization, etc. GATE (Gate 2000) architecture is implemented, and for query expansion such as for finding the synonym adopted wordnet library is adopted. TreshHold is determined to justify whether the answer given by the Aquas engine is eligible to pass to user. If the result from the Aquas engine (either S-Aquas or Onto-Aquas) is under the TreshHold or the user feedback is “Unsatisfied” then the expert locator engine is invoked. The expert locator engine will

find list of user that is an expert in the area related to the question and pass the question to the list of expert.

```
1.0  Start
2.0  User enter a question in Natural Language
3.0  Call Gate and Wordnet engine for Natural Language Processing.
4.0  Call S_Aquas Engine -> S_result_metric
5.0  Determine TreshHold
6.0  If S_result_metric >= TreshHold Goto 9.0
7.0  Call Onto_Aquas Engine -> Onto_result_metric
8.0  If Onto_result_metric < TreshHold Goto 10.00
9.0  Display answer to the user, then Goto 16.0
10.0 Display message "System Can not drawn answer from system"
11.0 User key in Feedback ->u_feedback
12.0 If u_feedback == 'satisfied' then Goto 16.0
13.0 Call expertise finding engine ->list_experts
14.0 Send the user question to list_experts
15.0 Get the feedback from list_experts
16.0 End
```

Figure 5- 16 Pseudocode for Aquas

From the Figure 5-16, there are some engines that are important to note such as GATE and Wordnet. GATE and Wordnet engine are developed in JAVA environment, and integrated in Moodle, which is in PHP environment using PHP_JAVA Bridge.

From Figure 5-16 in line 4.0, the S_Aquas engine algorithm of S-Aquas, which is fully implemented in PHP, is depicted by Figure 5-17. The S_Aquas engine algorithm can be explained concisely by generating Characteristic Vector (CV) for question entered by user and each question in FAQ. This CV is generated by using TFxIDF Algorithm. Then, calculate the similarity between CV of question from user and CV of each question in FAQ. In this research work, the CV similarity is calculated by using Inner Product (IP) and Euclidian Distance (ED) method.

```

1.0  start
2.0  User key in a question in Natural Language Processing (NLP)
3.0  Calculate and Construct user question Characteristic
      Vector(CV) using TFxIDF Algorithm -> uq_CV
4.0  Calculate and Construct FAQ Characteristic Vector (CV) based
      on keyword database using TFxIDF Algorithm -> faq_CVs
5.0  Calculate the Similarity Comparison between uq_CV and faq_CVs
      using either Euclidian Distance or Inner Product (IP) ->
      SC_CV_ED and SC_CV_IP
6.0  Determine Treshhold
7.0  If SC_CV_IP< Treshhold  or SC_CV_ED > Treshhold Then call
      Onto_Aquas Engine
8.0  Select The Answer from FAQ with Maximum value of SC_CV_IP or
      Minimum value of SC_CV_ED.
9.0  Return the Answer to User
10.0 Determine lo_treshold
11.0 Calculate and Construct Learning Objects Characteristic Vector
      (CV) based on keyword database using TFxIDF Algorithm ->
      lo_CVs
12.0 Calculate the Similarity Comparison between uq_CV and lo_CVs
      using either Euclidian Distance or Inner Product (IP) ->
      SC_LO_ED and SC_LO_IP
13.0 Select The learning object from learning resource base with
      value of SC_LO_IP >=lo_treshold or with value of SC_LO_ED
      <=lo_treshold.
14.0 Return the selected Learning Object to the user
15.0 End.

```

Figure 5- 17 The Pseudocode of S-Aquas

As depicted by Figure 5- 17 the heart of the S-Aquas is TFxIDF algorithm. Figure 5-18 depicts the pseudocode of this TFxIDF algorithm.

```

1.0 Assume: t.idf gives the idf of any term t
2.0 q.tf gives the tf of any query term
3.0 Begin
4.0 Score[] -> 0
5.0 For each term t in Query Q
6.0 Obtain posting list l
7.0 For each entry p in l
8.0 Score[p.docid] = Score[p.docid] + (p.tf * t.idf)(q.tf * t.idf)

```

Figure 5- 18 TFXIDF Pseudocode

Further, the Ontology-driven Automatic Question Answering System (Onto-Aquas) algorithm is depicted by Figure 5-19.

```

1.0 start
2.0 get the user input
3.0 call linguistic and query classification engine ->
    query_triple
4.0 call Relation and Similarity Service engine -> onto_triple
5.0 invoke Ontology similarity algorithm between onto_triple and
    Ontological knowledge base -> Ontology-Compatible_triple
6.0 Invoke Inference engine -> answer
7.0 return answer to user
8.0 get user feed back -> feedback
9.0 if feedback == 'satisfied' then end else
10.0 call expert_finder engine
11.0 end

```

Figure 5- 19 Onto-Aquas Pseudocode

The heart of Onto-Aquas is Ontology Similarity Comparison Algorithm, in this case an algorithm by (Culmone et al. 2002) is adopted as depicted in Figure 5-20.

```

1.0  Given a concept  $c \in C1$ :
2.0  Algorithm FindSimilar(Concept :  $c$ ) returns Concept
3.0  { $rc$  = Node corresponding to the concept  $c$  on platform 1}
4.0  begin
5.0  extractSubGraph( $rc$ );
6.0  for each not primitive concept  $s \in C2$  on platform 2 do
7.0  {  $rs$  = Node corresponding to the concept  $s$  on platform 2} begin
8.0  extractSubGraph( $rs$ );
9.0  Compare subgraphs  $S_c$ 
10.0 1 and  $S_s$ 
11.0 2: compute  $F(rc, rs)$ ;
12.0 end
13.0  $max = \max$ 
14.0  $s \in C2$ 
15.0  $F(rc, rs)$ ;
16.0 end
17.0 {  $\bar{s}$  such that  $F(rc, \bar{s}) = \max$  is the concept similar to  $c$  on
18.0 platform 2} return  $\bar{s}$  such that  $F(rc, \bar{s}) = \max$ ;
19.0 end
20.0 Algorithm extractSubGraph(Node :  $rc$ ) returns Subgraph
21.0 begin
22.0  $S_{ck}$ 
23.0 = Subgraph corresponding to concept  $c$  on platform  $k$ -th;
24.0 add the node  $rc$  to  $S_{ck}$  ;
25.0 for each arc  $(rc, j)$  outgoing from  $rc$  do
26.0 begin
27.0 add the arc  $(rc, j)$  to  $S_{ck}$ ;
28.0 add the node  $j$  to  $S_{ck}$ ;
29.0 if  $j$  not primitive then
30.0 extractSubGraph( $j$ );
31.0 end
32.0 return  $S_{ck}$ ;
33.0 end
34.0 Algorithm  $F(Nodes : rc, rs)$  returns value in  $[0, 1]$ 
35.0 begin
36.0 for each arc  $(rc, j)$  outgoing from  $rc$  do
37.0  $sum = sum + f((rc, j), rs)$ ;
38.0  $F = sum / (size(outgoingArcs(rc)) + size(outgoingArcs(rs)) - sum)$ 
39.0 return  $F$ ;
40.0 end
41.0 Algorithm  $f(Arc : a = (rc, j), Node : rs)$  returns value in  $[0, 1]$ 
42.0 begin
43.0 for each arc  $h = (rs, k)$  outgoing from  $rs$  do
44.0 if  $(\delta(a) = \delta(h))$  then
45.0 if  $(\lambda(j) = \lambda(k))$  then
46.0 if  $j, k$  primitive then
47.0  $f = 1$ ;
48.0 else if  $j, k$  not primitive then
49.0  $f = F(j, k)$ ;
50.0 else
51.0  $f = 0$ ;
52.0 return  $f$ ;
53.0 end

```

Figure 5- 20 Ontology Similarity Algorithm Pseudocode

5.5 Expertise Finding and Mapping Framework Implementation

As has been explained in chapter four (§ 4.3) to improve the performance of Aquas in the future usage, Ontology-driven expert locator is introduced. The expert locator which represents the expertise finding and mapping framework is implemented by executing an Ontology Query Language called OWL-QL and SPARQL from the expert user Ontology. The expert user Ontology has been discussed in § 4.3. An Example of SPARQL query is depicted by Figure 5-21.

```
PREFIX us: <http://localhost:9090/Ontologies/user#>
PREFIX lo: <http://localhost:9090/Ontologies/learningobject#>
SELECT ?username ?email
WHERE { ?x lo:summary ?'Semantic Web' .
        ?x us:expertise ?'Semantic Web' . }
```

Figure 5- 21 SPARQL Query For expert User

The user interface of expertise finding module is depicted by Figure 5-22 and Figure 5-23. Figure 5-22 illustrated how a user expert and Figure 5-23 is the user interface for user expert to give response to the question alert achieve the question alert.

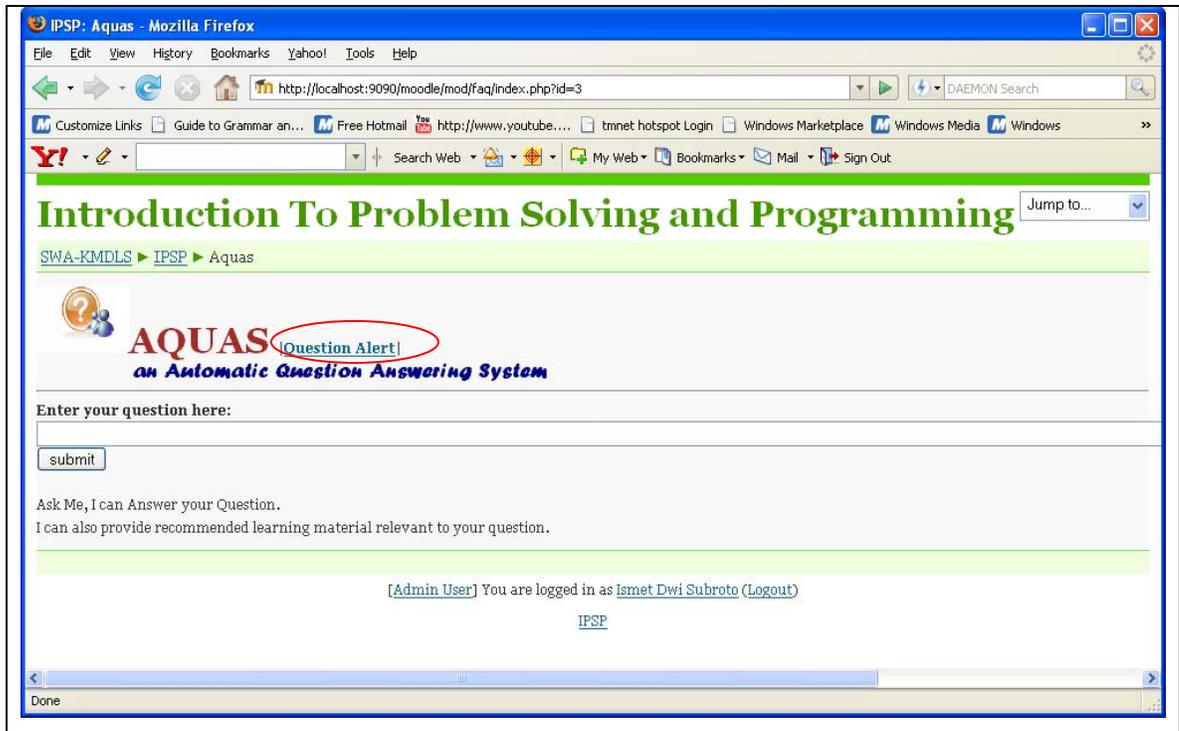


Figure 5- 22 User interface Aquas – Question alert for expert user

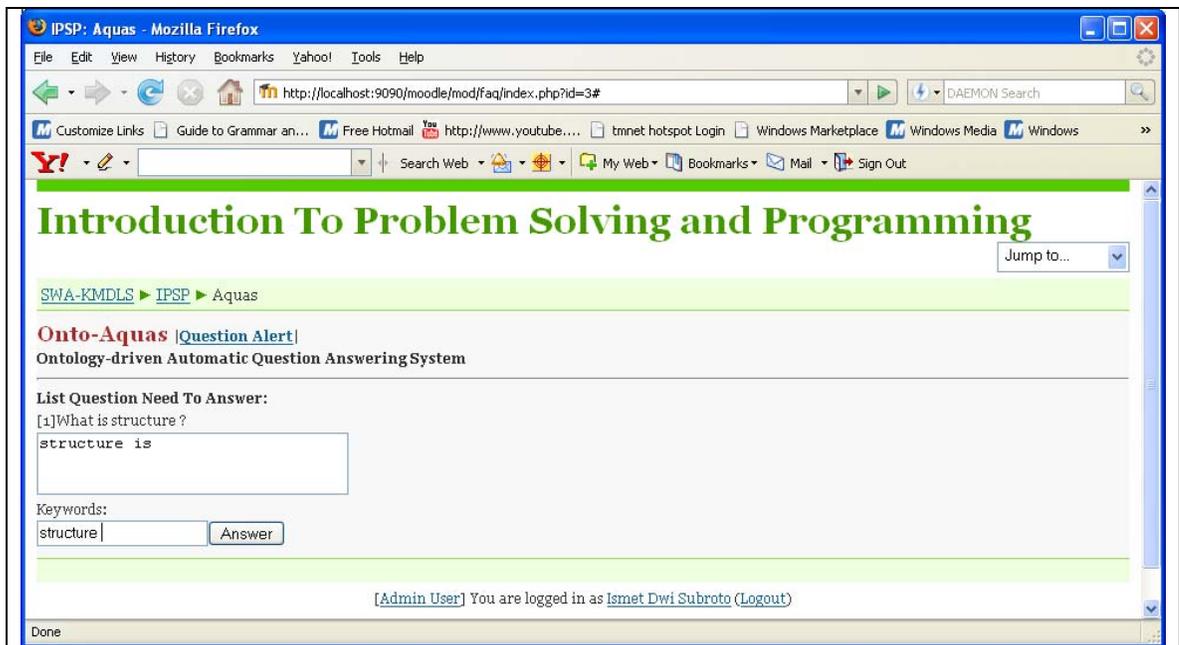


Figure 5- 23 User interface Aquas – User expert answering question

5.6 Summary

This chapter has discussed the development of Ontologies the backbone of the proposed SWA-KMDLS. Two kinds of initial Ontologies development namely Application and Domain Ontology were presented. Thereafter, the development and implementation of SWA-KMDLS' main components namely SAOKBCS, Aquas, and expert locator were presented. SAOKBCS that intended for learning object management, Aquas for knowledge retrieval as well as Ontology-driven expert locator for encouraging knowledge sharing in e-Learning environment were clearly discussed by presenting their user interface and implemented algorithms in pseudocode. The following chapter discusses the experiment, experimental result evaluation and discussion of the developed system.

CHAPTER 6

THE EXPERIMENT, EVALUATION AND DISCUSSION

This chapter presents the experiments conducted for the SWA-KMDLS: SAOKBCS and Aquas. This chapter is divided into two sections. The first section (§ 6.1) presents the experiment its self and the second section (§ 6.2) focuses on evaluation and discussion on experimental result. The experiment section shall present the experimental environment, which explains where the experiment was conducted; the experimental scenario, which explains how the experiment was conducted; and experimental result. The evaluation and discussion section shall discuss evaluation on experimental result to asses the performance of the tested system and discuss some important thing to be noted based on the experiment.

6.1 The Experiments

The aim of these experiments is to assess the performance of the proposed system: SAOKBCS and Aquas. The experiment environment, experimental scenarios, and experiment results are described as follows:

6.1.1 Experimental Environment

The proposed system: SAOKBCS and Aquas were implemented in:

1. SWA-KMDLS Moodle based e-learning management system.
2. Running on PC Server with AMD Turion™ MK-38 2.20 Ghz CPU and 512 MB RAM.
3. Developed using PHP 5 and Java SE.
4. Operating System: Microsoft Windows XP SP2.
5. Web Server: Apache 2.2 Web server.
6. Database Management System (DBMS): MySQL 5.0.
7. Java SDK version 1.6.0_03.

6.1.2 Experimental Scenarios

The experimental scenarios that describes how the experiment is conducted has been clearly explained in Chapter Three (Methodology). Please refer to section experimental testing and evaluation (§ 3.1.4).

6.1.3 Experimental Result

This section presents the experimental result based on the scenarios in the previous section. Firstly, the experiment on SAOKBCS is presented followed by the experimental result on Aquas.

(i) Experimental Result on SAOKBCS

Based on scenarios in § 6.1.2, the experiment has been conducted on SAOKBCS. The result of this experiment is presented in Table 6-3. The values in Table 6-3 represented amount of seven Ontology notions (concepts, instance, similarity, subclassof, instanceof, relation and subtopicof) that have been successfully extracted by Ontology learning engine. T, F, and D indicate TRUE, FALSE and DON'T KNOW respectively which means that in the perspective of domain/human expert, the corresponding Ontology notion extracted by Ontology learning engine is correct, incorrect, and user/domain expert can not determine whether it is correct or vice versa. In other words, in this experiment the one who can justify T, F, D for Ontology notion extracted by Ontology learning is the domain expert or human expert, in this case the domain/human expert is the authors.

For example, based on Table 6-1 amount of the concept that can be extracted from LM1 is 60, where 54 out of 60 are T(correct), 4 out of 60 are F (wrong) , and 2 out of 60 are D (don't know). Column “%” in Table 6-1 represents the values in percentage or known as acceptance notion rate (ANR) using Eq. 6-1.

$$ANR = \left(\frac{T}{T + F + D} \right) 100 \quad Eq. 6-1$$

Table 6- 1 Experiment on SAOKBCS Result

Onto Notion	LM1						LM2					
	T		F		D		T		F		D	
	N	%	N	%	N	%	N	%	N	%	N	%
Concept	54	90	4	7	2	3	40	83.34	7	14.58	1	2.08
Instance	0	0	2	100	0	0	2	28.57	5	71.43	0	0
Similarity	0	0	0	0	0	0	0	0	0	0	0	0
SubClassOf	11	10	94	87	3	3	3	5.08	55	93.24	3	5.08
InstanceOf	0	0	0	0	0	0	0	0	1	100	0	0
Relation	1	14	6	86	0	0	1	50	1	50	0	0.20
SubTopicOf	11	3	300	84	50	13	1	0.28	355	98.89	3	0.83

In the Table 6-1 LM1 is learning material from experiment in scenario 1 and LM2 is learning material from experiment in scenario 2. Furthermore, “N” is the number of Ontology notions that successfully extracted from learning material and “%” is the percentage of Ontology notions that successfully extracted from learning material or ANR value. From the experimental result in Table 6-1, the average value of ANR from LM 1 and LM 2 can be calculated and presented in Table 6-2.

Table 6- 2 ANR of Ontology Learning Result

Onto Notion	LM		
	T	F	D
Concept	86.67	10.79	0.72
Instance	14.285	85.715	0
Similarity	0	0	0
SubClassOf	7.54	90.12	4.04
InstanceOf	0	50	0
Relation	32	68	0.1
SubTopicOf	1.64	91.445	6.915

Figure 6-2 depicts the graphical representation of experimental Ontology learning result that is previously presented in Table 6-2. The vertical bar in the figure represents percentage or ANR value of Ontology notion that can be extracted from Learning Materials. The result as depicted in Figure 6-1 shows that the domain experts agreed 86.67% of concepts extracted by SAOKBCS Ontology learning engine. However, still less than 15% for instance and relation extraction between concepts (similarity, subclassof, instanceof, relation, and subtopicof) the domain expert agrees those.

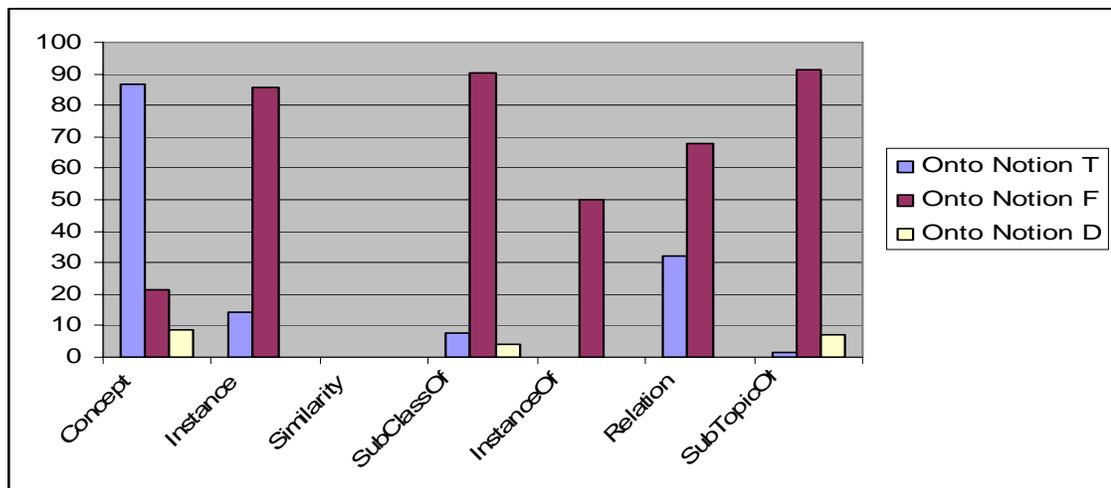


Figure 6- 1 Graphics of Experimental SAOKBCS Ontology Learning Result

(ii) Experimental Result on Aquas

To evaluate the performance of the Aquas, the experiment has been conducted based on experimental scenario in § 6.1.2. The experiment was conducted over C Programming FAQs database adopted from (Summit 2008) containing 300 FAQs. The experiment was also conducted over learning resource base containing 100 learning materials. In the experiment, two versions of data were generated from the FAQ database. The first version is the original data from the FAQ database. The second version is classified data based on six interrogative words i.e. “What”, “Who”, “Where”, “When”, “Why”, and “How.

To evaluate the performance of different approaches, the experiment is divided into three scenarios as described in experiment design (Chapter Three : § 3.1.4). The three scenarios are as follows:

- a. The first scenario is student enter questions that are similar to the ones being asked before.
- b. The second scenario is student enter questions that are quite different from any question being asked before.
- c. The third scenario is the combination of two previous scenarios.

The Aquas experimental results are divided into two parts. The first part is result of experiment to find the best-fit answer and the second part is result experiment to find the most relevant learning material.

ii.1 Experiment to Find Best Fit Answer

For finding best-fit answer, the experimental result is presented in Table 6-3. As presented in Table 6-3 the experimental result is divided into two categories. The first category is experimental result with classified FAQ data and the second one is experimental with unclassified data. For each category, the experimental result is divided

into experimental result with Euclidian Distance (ED) and experimental result with Inner Product (IP). ED and IP are Characteristic Vector (CV) similarity measurement method that has been explained in § 5.4.2. From Table 6-3, it can be seen that the average ratio of correct answer is greater than 88.07%. The experimental result also revealed that Aquas is able to answer questions that have not been asked before.

Table 6- 3 Experimental result in finding best-fit answer

Scn.	Unclassified		Classified	
	ED(%)	IP(%)	ED(%)	IP(%)
S1	97.01	98.05	98.43	98.31
S2	82.06	83.01	83.22	79.85
S3	87.04	86.07	88.01	87.98
Avg	88.07	89.04	89.88	88.71

To compare the performance of Aquas between using EP method and using IP method, the experimental result is depicted in chart by Figure 6-2. Then, to compare the performance of Aquas between within classified data and within unclassified data, the experimental is depicted in chart by Figure 6-3.

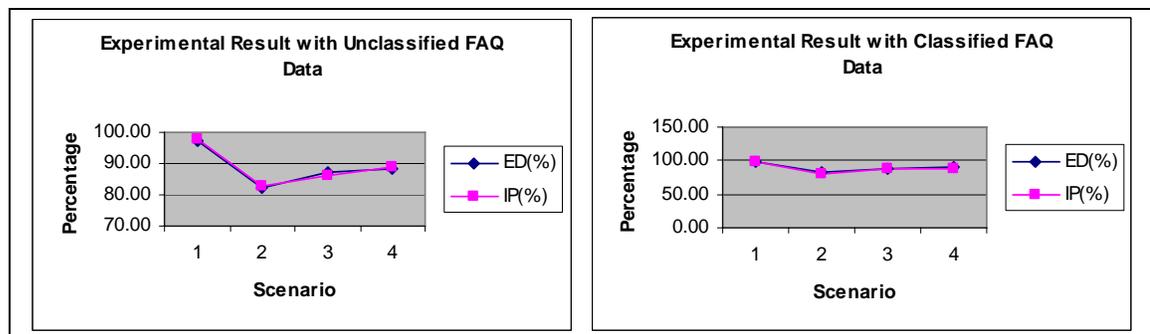


Figure 6- 2 Chart of experimental result in finding best-fit answer

As depicted by Figure 6-2, the experimental result with unclassified FAQ data is presented on the left and classified data in right. It shows that IP method (89.04%) has better performance than ED method (88.07%) with unclassified FAQ data, and ED

method (89.88%) has better performance than IP method (88.71%) with classified data. The classification of FAQ data affect on the performance of Aquas in finding the best-fit answer but not too significant as depicted in Figure 6-3.

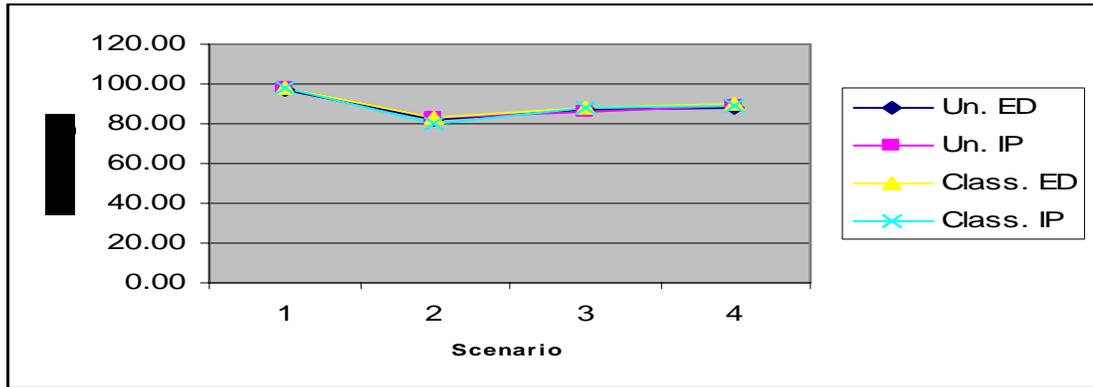


Figure 6- 3 Comparison chart of experimental result in finding best-fit answer

ii.2 Experiment to Find Most Relevant Learning Material

The second part of Aquas experiment aims to find the most relevant learning material. As well as the experiment to find best-fit answer, the experiment is conducted over either unclassified FAQ data or classified FAQ data and using either ED or IP method. The experimental result is presented in Table 6-6. As presented in Table 6-4, the experimental result shows that ED method (89.99% and 91.19%) has higher precision than IP method (87.88% and 86.06%) over either unclassified data or classified data.

Table 6- 4 Experimental result in finding most-relevant learning material

Scn.	Not Classified		Classified	
	ED(%)	IP(%)	ED(%)	IP(%)
S1	98.11	98.05	98.50	95.01
S2	85.22	80.32	86.86	81.78
S3	86.65	85.26	88.20	83.01
Avg	89.99	87.88	91.19	86.06

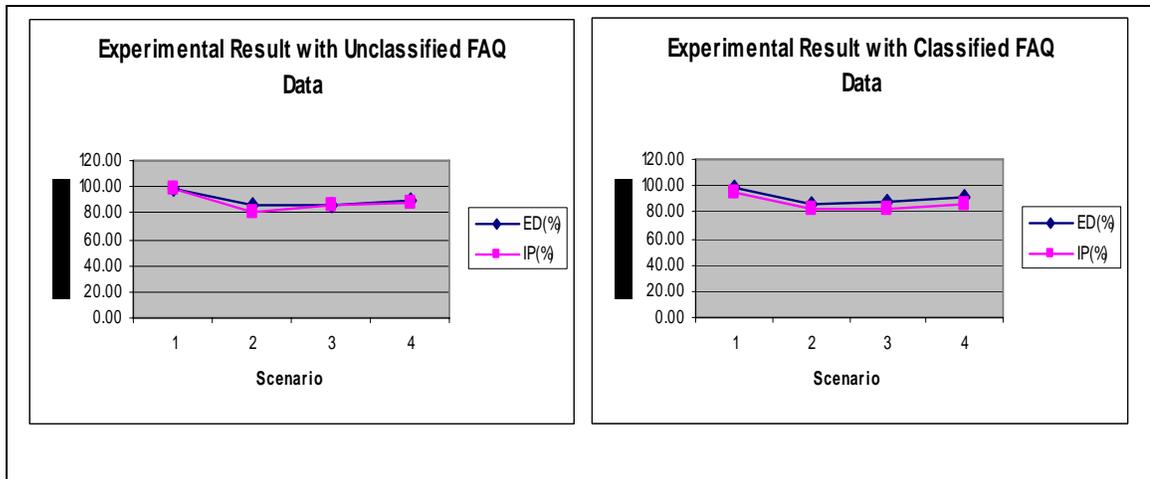


Figure 6- 4 Chart of experimental result in finding the most relevant material

Figure 6-4 presents the comparison between ED and IP method performance in finding the most relevant learning material with either classified FAQ data or unclassified FAQ data. It shows that ED method has better performance than IP method with either unclassified FAQ data or classified data. The classification of FAQ data affect on the performance of Aquas in finding the most relevant learning material but not too significant as depicted in Figure 6-5.

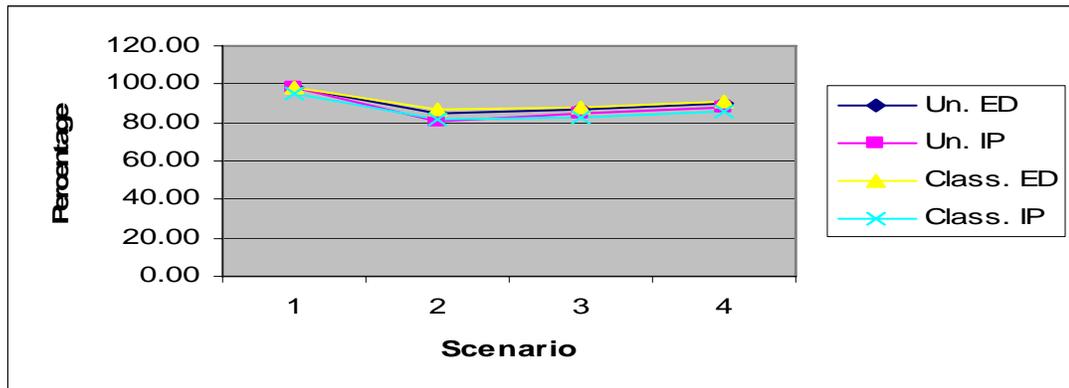


Figure 6- 5 Comparison chart of experimental result in finding the most relevant material

6.2 Evaluation and Discussion

Based on METHONTOLOGY as explained in § 3.2 the Ontology evaluation includes *verification, validation, and assessment*. Since Ontology verification and validation are something difficult to measure, in this research work only Ontology assessment is taken to evaluate the constructed Ontology from SAOKBCS. The Ontology assessment aims to judge whether the Ontology is correct in the user's point of view. The user's point of view judgment then can be measured by using common measurement tool in information retrieval area namely *Precision (P)*, *Recall (R)*, and *F-Measure (M)* which has been explained in § 3.3. From the experimental result in § 6.1.3 the P, R, and F can be calculated as presented in Table 6-5.

Table 6- 5 Precision, Recall, and F Measure of SAOKBCS

<i>Ontology Notion</i>	Precision (P)	Recall (R)	Precision (P)
Concept	0.87	0.92	0.89
Instance	0.14	0.17	0.15
Similarity	0.00	0.00	0.00
SubClassOf	0.08	0.58	0.13
InstanceOf	0.00	0.00	0.00
Relation	0.32	0.20	0.23
SubTopicOf	0.02	0.47	0.03

The comparison between P, R, and F of SAOKBCS experimental result from Table 6-5 is depicted by Figure 6-6.

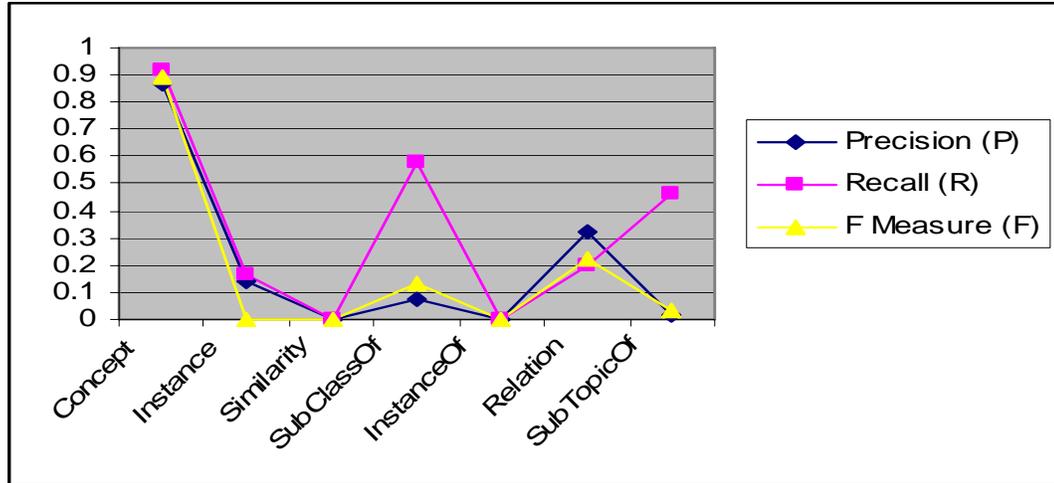


Figure 6- 6 Comparison of Precision, Recall, and F Measure for SAOKBCS

Based on the experimental result of SAOKBCS as presented § 6.1.3.1 and its evaluation using precision, recall, and F Measure as presented in Table 6-5 and depicted by Figure 6-6 it is known that a human expert agreed with a very large fraction (86.67 %) of *concept* in Ontology that Ontology learning engine in SAOKBCS produced. The precision and recall for *concept* extraction also present high values, near to 0.9 where 1.0 is the highest value. It was able to perform a large number of concept extractions, thus saving the expert time and effort to do it manually.

However, the Ontology learning engine in our SAOKBCS framework still demonstrates a low performance in instance and relation between concepts (similarity, subclassof, instanceof, relation, and subtopicof) extraction. It is understandable that it is very difficult to know the relation between concept, because it is not syntactic problem only but also semantic problem that is still difficult understandable by machine. Therefore, it needs more effort in future work that collaborated cross-research area such as semantic linguistic, NLP and text mining.

To evaluate the performance of Aquas the same measurement such as Precision, Recall and F Measure can be used. However, for simplicity the performance of Aquas can be evaluated based on user feedback as the result is depicted by Figure 6-2 for finding best-

fit answer experiment and Figure 6-4 for finding the most relevant learning material. In Figure 6-2 and Figure 6-4 although the result for each scenario is inter-connected each other by line, but each scenario is independent. The line in chart aims reminds the different between scenarios.

In both Figure 6-2 and Figure 6-4 founded that both chart has the same pattern. Scenario 1 is the highest, then scenario 2 is extremely lower than scenario 1 and scenario 3 is between scenario 1 and scenario 2. Scenario 3 is higher than scenario 2 but still lower than scenario 3. It is obviously amenable because in scenario 1 the *question* is set up to be similar with the question in FAQ data and in scenario 2 the *question* is set up to be quite different with the question in FAQ data. Scenario 3 is randomly selected from scenario 1 and scenario 2.

The important thing to note is the experimental result in scenario 2 in which despite the *question* is being set up to be quite different from the questions in the FAQ data, the Aquas still can answer the question with acceptable precision. Scenario 4 is the average result of scenario 1, scenario 2, and scenario 3. Another point important to note is the expert locator framework, which is obviously amenable, can improve the performance of Aquas. Each question finally will be answered by the Aquas, because the expert locator framework keep on looking for expert user and encourage him to share his knowledge by answering the *question*.

6.3 Summary

The experiment of proposed SWA-KMDLS: SAOKBCS and Aquas as well as expert locator have been presented in this chapter, which included the experimental environment, experimental scenario, and the experimental result. The evaluation and discussion on the experimental result were also presented. Experimental result on SAOKBCS showed good performance on concept extraction from learning object in which more than eighty percent extracted concepts are agreed by domain expert. Experimental result on Aquas also showed good performance in which more than eighty

percent users are satisfied by the answer given by Aquas. The following chapter is the last chapter that will conclude the entire content of thesis.

CHAPTER 7

CONCLUSION

This chapter concludes all the previous chapters of the research thesis, which is motivated by author's experience in the ineffectiveness use of e-Learning system caused by ill-managed knowledge in the e-Learning environment at the University where this research is conducted. This chapter reviews how the research objectives of this thesis are addressed and highlight the thesis contributions, the limitations, challenges, open issues, and recommendation for future works.

7.1 Addressing the Research Objectives

This section presents how the research objectives are addressed in this thesis by directing to the corresponding chapter(s), allowing the reader to see their respective contributions.

1. *To design framework of the SWA-KMDLS : Semantic Web Aware and Knowledge Management Driven e-Learning System.*

The framework and architecture of the SWA-KMDLS that is Semantic Web and Knowledge Management enhanced e-Learning system was presented In *Chapter 4: Design Framework and Architecture of SWA-KMDLS*. There are three main contributions of this research work: Ontology-based learning object management, automated question answering system, and expertise finding and mapping system. In Ontology-based learning object management design of (semi-) automatic ontological knowledge base construction system which can construct automatically Ontology (semi-) from learning object was presented. Framework of automated question answering system (Aguas) which consists of statistical-based Aquas and Ontology-driven Aquas that is collaborated with Ontology-driven expert locator framework also presented in chapter 4.

2. *To develop the SWA-KMDLS.*

In *Chapter 5: Development and Implementation* clearly described how each component of SWA-KMDLS namely SAOKBCS, Aquas, and expert locator are developed. SWA-KMDLS components are developed in web-based application using PHP and Java. How each process in architecture and framework is developed clearly presented in pseudocode.

3. *To develop Ontology that models the SWA-KMDLS environment.*

In *Chapter 5: Development and Implementation* have been described how the Domain Ontology is developed in this research work. Domain Ontology is aimed to model the SWA-KMDLS environment namely Moodle e-Learning Management system. The main concepts and relation between concepts in Moodle are formally specified in Domain Ontology. The result of Domain Ontology is completely presented in Appendix IV.

4. *To develop Ontology that models the knowledge in SWA-KMDLS.*

In *Chapter 5: Development and Implementation* have been described how the Application Ontology is developed in this research work. Application Ontology is aimed to model the knowledge from and inside Learning Objects (LOs). The main concepts and relation between concepts from LOs and inside LOs are formally specified in Application Ontology. The result of Application Ontology is completely presented in Appendix IV.

5. *To implement the framework of SWA-KMDLS.*

In *Chapter 5: Development and Implementation* started with discussion on the implemented environment where the proposed system was implemented namely Moodle e-Learning management system. Continued with presenting

the initial Ontology development which consists of domain and application Ontology, including the Ontology testing for the developed Ontology. The implementation of (semi-)automatic ontological knowledge base construction system framework (SAOKBCS) and Aquas with expertise finding and mapping framework were also discussed by presenting some critical aspect in the implementation such as its user interface and algorithm.

6. *To evaluate and validate the Ontology development using common evaluation measurement.*

In *Chapter 5: Development and Implementation* have been described how the initial Ontologies that are developed manually are validated using reasoning tools namely Racer and Fact+. In *Chapter 6: The experiment, Evaluation, and Discussion* also have been explained how the Ontologies that are developed semi-automatically from SAOKBCS is evaluated using ANR(Acceptable Notion Rate), Precision, Recall, and F-Measure.

7. *To test and evaluate the performance of the SWA-KMDLS (Chapter 6).*

In *Chapter 6: The experiment, Evaluation, and Discussion* also have been explained how the performance of SWA-KMDLS is assed. The performance of Aquas in SWA-KMDLS is measured by calculating the percentage of answer from Aquas that are acceptable by the user. The result of SWA-KDLS performance assesment was also presented and discusses.

7.2 Contributions

In general, the main contributions of this thesis are the exploitation of Semantic Web and KM and its technology to enhance an e-Learning Management System where the e-Learning is viewed as part of more general framework, Knowledge Management, and developed in Semantic Web environment the promising future of web technology. It will

benefit e-Learning developer in designing an e-Learning system in the point of view KM as broader framework; and SW (the future of web technology)-aware environment. It also benefits the educational practitioner to implement *social constructivist* pedagogical approach in a *courseware* development.

In term of e-Learning practices, the contribution of this thesis can be highlighted as below:

1. Using Ontology approach for managing (indexing and organizing) Learning Objects in e-Learning environment by developing a (semi-)automatic Ontological knowledge base construction system (SAOKBCS) from learning objects. It will benefit for (i) Representing learning material/object/resource content, learner model, and knowledge in general using Ontology. (ii) Organizing learning repositories and enabling shareable learning objects.
2. The SAOKBCS constructed Ontologies (semi-)automatically from uploaded learning object. Thus, saving the expert time and effort to do it manually.
3. New approach in Knowledge Retrieval by developing an Ontology-driven automated question answering system (Aquas). It will benefit for knowledge retrieval and particularly learning object retrieval in e-Learning environment. In this semantically related environment, users of e-Learning can easily find knowledge to address problem at hand and find the exact learning object that they really need. It is an important step toward adaptive and personalized e-Learning system.
4. Improving the performance of Information and Knowledge Retrieval by proposing an Expert Locator Framework to encourage expert users to share their knowledge in e-Learning environment. As the Knowledge Management driven e-Learning is not solely technology problem, but also aware of people and culture, it is very important to initiate knowledge sharing culture between users in e-Learning environment.

7.3 Limitations, Challenges and Open Issues

Although the experimental results in (semi-) automatic Ontological Knowledge Base system (SAOKBCS) from Learning objects shows significance in Concepts extraction in which more than eighty percent extracted concepts are agreed by domain expert, but it still shows low significance in extracting the relations between concepts in which only less than thirty percent extracted concepts are agreed by domain expert.

The proposed system assumed that there are no changes in the Ontology, for instances there are no learning resource, which is referred by the Ontological knowledge base deleted from e-Learning environment. It challenges new effort on how to address this matter in term of Ontology reengineering. An Ontology reengineering adopted from software engineering, which is more mature discipline, that aware on Ontology changes can be argued to solve this problem.

From the implementation of Knowledge Management and Semantic Web in e-Learning system, it is encouraged to open new issues in implementing or collaborating Knowledge Management and Semantic Web especially the Ontology to Service Oriented Architecture (SOA), such as Ontology-driven SOA.

7.4 Future Works

Considering the limitations, some recommendations for future works are as follows:

1. For researcher in *Ontology Learning*, since our proposed system showed good performance only on *concept* extraction, there is a need to improve the method and algorithm to extract the relations between concepts in text such as *subclassof* and *instanceof* that collaborate inter-discipline research area such as Linguistic and Text Mining which not only consider syntactic aspect but also the more important thing is Semantic aspect.
2. For researcher in Question Answering system, although an Ontology approach potentially can improve the precision and recall, but query on Ontology needs improvement and further research for finding novel method on Ontology similarity measurement which until now it still difficult to calculate.

3. Since Ontology changes are considered critical aspect, the Ontology Re-Engineering method is strongly needed to cope with the changes in the resources of Ontology.
4. The context of this thesis is e-Learning in academic scenario, in the future work it can be investigated the implementation of Semantic Web and KM technology for e-Learning in professional scenario such as knowledge society e-Learning for profit oriented learning organization.

REFERENCES

- ABEL, M.-H., BENAYACHE, A., LENNE, D., MOULIN, C., BARRY, C. & CHAPUT, B. (2004). "Ontology-based Organizational Memory for e-learning." *Journal of Educational Technology & Society* 7(4): 98-111.
- ADL. (2008). "Advanced Distributed Learning Project." Retrieved 15-07-2008, from <http://www.adlnet.gov>.
- AHERN, D., CLOUSE, A. & TURNER, R. (2001). *CMMI Distilled* Addison-Wesley.
- AHKIM, J. & YOUNGCHOI, S. (2007). Evaluation of Ontology Development Methodology with
CMM-i. 5th ACIS International Conference on Software Engineering Research, Management & Applications (SERA) 1: 823 - 827.
- AICC. (2008). "Aviation Industry CBT Committe." Retrieved 15-07-2008, from <http://www.aicc.org>.
- ALAVI, M. & LEIDNER, D. E. (2001). "Review: Knowledge Management and Knowledge Management System: Conceptual Foundations and Research Issues." *MIS Quarterly* 25(1): 107-136.
- ALLEN, M. W. & ALLEN, M. (2002). *Michael Allen's Guide to E-Learning*. New Jersey, John Wiley.
- ANDERSON, C. (2008). *Worldwide and U.S. Corporate eLearning 2008–2012 Forecast*, IDC: International Data Corporation Research.
- ARGOTE, L., MCEVILY, B. & REAGANS, R. (2003). "Managing Knowledge in Organizations: An Integrative Framework and Review of Emerging Themes." *Management Sciences* 49(4): 571-582.
- AROYO, L. & DICHEVA, D. (2004). "The New Challenges for E-learning: The Educational Semantic Web." *Journal of Educational Technology & Society* 7(4): 59-69.
- BARKER, P. (2005). *What is IEEE Learning Object Metadata / IMS Learning Resource Metadata? Joint Information Systems Committee Centre for Educational Technology and Interoperability Standard Research Report*.
- BERGERON & BRYAN, P. (2003). *Essentials of knowledge management*. New Jersey, John Wiley & Sons.

BERNARAS, I., LARESGOITI & CORERA, J. (1996). Building and reusing ontologies for electrical network applications. European Conference on Artificial Intelligence (ECAI'96), Budapest, Hungary.

BERNERS-LEE, T., HENDLER J. & Lassia, O. (2001). "The Semantic Web." Journal Of Scientific American 284(1): 34-43.

BJØRKE, Å. (2008). "Pedagogical Approaches " Retrieved 10-10-2008, from <http://gvu.unu.edu/workshops/Suez/LinkedDocuments/pedagogical%20approaches.doc>.

BLACKBOARD (2008). "Blackboard Learning System." Retrieved 02-03-2008, from http://www.blackboard.com/products/academic_suite/learning_system/index.

BORGO, S., MASOLO, C., GANGEMI, A., GUARINO, N. & OLTRAMARI, A. (2002). Research Preliminary Report WonderWeb Deliverable D17 The WonderWeb Library of Foundational Ontologies.

BRODSKY, M. W. (2008). E-learning Implementation: Your Roadmap to Success. Ulysses Learning, Ulysses Learning.

BRUIJN, J. D. (2007). "KWTR: f-logic." Retrieved 26-07-2008, from http://semanticweb.org/wiki/KWTR:_f-logic.

BUILDER, M. C. (2008). "Macromedia Course Builder : eLearning - CourseBuilder for Dreamweaver Extension." Retrieved 10-04-2008, from http://www.adobe.com/resources/elearning/extensions/dw_ud/coursebuilder/.

BUITELAAR, P., CIMIANO, P. & MAGNINI, B. (2003). Ontology Learning from Text: An Overview, IOS Press.

BUITELAAR, P. & SINTEK, M. (2004). OntoLT Version 1.0: Middleware for Ontology Extraction from Text. 3rd International Semantic Web Conference (ISWC2004), Hiroshima Prince Hotel, Hiroshima, Japan.

CAMPBELL, L. M. & CURRIER, S. (2005). JISC Pedagogical Vocabularies Project: Priorities, Issues and Recommendations to JISC. Deane Road Bolton, JISC CETIS, Institute for Educational Cybernetics (IEC), University of Bolton.

CASELY-HAYFORD, L. (2005). "A comparative analysis of methodologies, tools and languages used for building Ontologies." CCLRC Council for the Central Laboratory of the Research Councils Daresbury Laboratories Publication UK.

CHEW, L. K. (2006). "E-learning Standards and Pedagogical Approaches." Journal Of Synthesis 2(1): 1-10.

CIMIANO, P. (2006). *Ontology Learning and Population From Text Algorithms, Evaluation And Applications*. New York, Springer.

CIMIANO, P. & VÄOLKER, J. (2005). *Text2Onto A Framework for Ontology Learning and Data-driven Change Discovery*. 10th International Conference on Applications of Natural Language to Information Systems (NLDB), Alicante, Spain, Springer.

CIO. (2001). "CIO.COM - Business Technology Leadership." Retrieved 10-05-2008, from <http://www.cio.com>.

CIORĂSCU, C., CIORĂSCU, I. & STOFFEL, K. (2003). *knOWLer Ontological Support for Information Retrieval Systems*. 26th Annual International ACM SIGIR Conference, Workshop on Semantic Web, Toronto, Canada.

CLAREY, J. (2007). *E-Learning 101: An Introduction to E-Learning, Learning Tools, and Technologies*. Sunnyvale, CA., Brandon Hall Research

CLARK & PARSIA (2006). "Pellet: The Open Source OWL DL Reasoner." Retrieved 02-08-2008, from <http://pellet.owldl.com/>.

CLARK, R. C. & MAYER, R. E. (2002). *e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning*, Pfeiffer.

COLE, J. (2005). *Using Moodle*, O'Reilly Publishing.

CONRAD, K. (2000). *Instructional Design for Web-based Training (Paperback)*. Amherst, Massachusetts, HRD Press.

Conzilla. (2008). "Conzilla - The Concept Browser." Retrieved 15-08-2008, from <http://www.conzilla.org/>.

CORBY, O., DIENG-KUNTZ, R. & FARON-ZUCKER, R. (2004). *Querying the Semantic Web with the CORESE search engine*. 16th European Conference on Artificial Intelligence (ECAI'2004), Valencia, IOS Press.

CORCHO, O., FERNANDEZ-LOPEZ, M. & GOMEZ-PEREZ, A. (2003). "Methodologies, tools, and Languages for building ontologies where is their meeting point?" *Sciencedirect: Journal Of Data & Knowledge Engineering* 46(1): 41-64.

CORCHO, O., FERNÁNDEZ-LÓPEZ, M., GÓMEZ-PÉREZ, A. & LÓPEZ-CIMA, A. (2005). *Building Legal Ontologies with METHONTOLOGY and WebODE*. *Law and the Semantic Web*. Berlin / Heidelberg, Springer 3369/2005: 142-157.

CORE, D. (2007). "Dublin Core : Metadata Initiative." Retrieved 18-07-2008, from <http://dublincore.org/>.

CRISTEA, A. I. (2004). "What can the Semantic Web do for Adaptive Educational Hypermedia?" *Journal of Educational Technology & Society* 7(4): 40-58.

CULMONE, R., ROSSI, G. & MERELLI, E. (2002). An ontology similarity algorithm for BioAgent. NETTAB Workshop on Agents in Bioinformatics, Bologna.

CUNNINGHAM, H., MAYNARD, D. & TABLAN, V. (2000). JAPE: A Java Annotation Patterns Engine (Second Edition). Technical report CS--00--10, Department of Computer Science, University of Sheffield.

DACONTA, M. C., OBRST, L. J. & SMITH, K. T. (2003). *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. Indiana, Wiley Publishing.

DAML. (2002). "DAML Ontology Language Feature." Retrieved 14-08-2008, from <http://www.daml.org/language/features.html>.

DAVIES, J., STUDER, R. & WARREN, P. (2006). *Semantic Web Technologies Trends and Research in Ontology-based Systems*. West Sussex, England, John Wiley.

DERRY, S. J. (1999). A Fish called peer learning: Searching for common themes, A. M. O'Donnell & A. King.

DEVEDŽIC, V. (2004). "Web Intelligence and Artificial Intelligence in Education." *Journal of Educational Technology & Society* 7(4): 29-39.

DIENG-KUNTZ & DIENG-KUNTZ, R. (2004). "Capitalisation des Connaissances via un Web Sémantique d'Entreprise. ." In I. Boughzala et J-L. Ermine (eds.), *Chapitre 12 de Management des Connaissances en Entreprise*, Hermès.

DONGMING, X. & HUAIQING, W. (2008). "Integration of Knowledge Management and e-Learning." *Knowledge Management Concepts, Methodologies, Tools, and Application* 5(5): 2292-2300.

DOWNES, S. (2007). "E-learning 2.0." Retrieved 10-07-2008, from <http://www.elearnmag.org/subpage.cfm?section=articles&article=29-1>.

DOWNES, S. (2007). "Learning Networks in Practice." *BECTA: Emerging Technologies for E-learning* 2(1): 19-27.

DRUCKER, P. (2000). "Need to Know: Integrating e-Learning with High Velocity Value Chains. ." A Delphi Group White Paper.

DZBOR, M., MOTTA, E. & STUTT, A. (2005). "Achieving Higher-level Learning Through Adaptable Semantic Web Applications." *International Journal of Knowledge and Learning* 1(1).

EBNER, M. (2007). *E-Learning 2.0 = e-Learning 1.0 + Web 2.0*. IEEE: Second International Conference on Availability, Reliability and Security 2007 (ARES 2007): 1235 – 1239.

ELEARNINGEUROPE (2008). "Babylon & ONtology: the European Educational Social Network." Retrieved 21-07-2008, from http://www.elearningeuropa.info/directory/index.php?page=doc&doc_id=11592&doclng=6.

ERNEST, P. (1999). *Social Constructivism as a Philosophy of Mathematics: Radical Constructivism*. New York, State University of New York Press.

FOSNOT, C. T. (1996). *Constructivism: Theory, Perspective and Practice*. New York., Teachers College Press.

GATE (2000). "Gate-General Architecture For Text Engineering." Retrieved 12-03-2008, from <http://www.gate.ac.uk/>.

GONZÁLEZ, R. G. (2005). *A Semantic Web approach to Digital Rights Management*. Phd Thesis in Computer Science and Digital Communication Department of Technologies. Barcelona, Universitat Pompeu Fabra: 286.

GOTTSCHALK, P. (2005). *Strategic Knowledge Management Technology*. London, Idea Group Publishing.

GREGLER, M. E. (1997). *Learning and instruction: Theory into practice* (3rd ed). Upper Saddle River, NJ, Prentice-Hall.

GRUBER, T. (1993). "A Translation Approach to Portable Ontology Specifications." *Journal Of Knowledge Acquisition* 5(1): 199-110.

GRUNINGER, M. & FOX, M. S. (1995). *Methodology for the design and evaluation of ontologies*. Workshop on Basic Ontological Issues in Knowledge Sharing. Montreal.

GUARINO, N. (1998). *Formal Ontology in Information Systems*. First International Conference (FOIS '98), Trento, Italy, IOS Press.

HARRIS, M. C. & THOM, J. A. (2006). *Challenges Facing the Retrieval and Reuse of Learning Objects*. Workshop on Learning Object Repositories as Digital Libraries: Current Challenges, Alicante, Spain.

HEARST, M. A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora Fourteenth International Conference on Computational Linguistics, Nantes, France.

HENZE, N., DOLOG, P. & NEJDL, W. (2004). "Reasoning and Ontologies for Personalized E-Learning in the Semantic Web." *Journal of Educational Technology & Society* 7(4): 82-97.

HONEBEIN, P. C., DUFFY, T. & FISHMAN, B. (1993). *Constructivism and the Design of Learning Environment: Context and Authentic Activities for Learning*. New York, Springer-Verlag.

HORRIDGE, M. (2005). "OWLViz A visualisation plugin for the Protégé OWL Plugin." Retrieved 02-08-2008, from <http://www.co-ode.org/downloads/owlviz/>.

HORTON, W. (2000). *Designing Web-based Training: : How to Teach Anyone Anything Anywhere Anytime*. Newyork, John Wiley & Sons.

HORTON, W. & HORTON, K. (2003). *E-learning Tools and Technologies: A consumer's guide for trainers, teachers, educators, and instructional designers*, Joe Wikert.

IBM. (2007). "IBM Integrated Ontology Development Toolkit." Retrieved 20-07-2008, from <http://www.alphaworks.ibm.com/tech/semanticstk>.

IEEE. (2008). "IEEE Learning Technology Standard Committee." Retrieved 18-07-2008, from <http://ieeeltsc.org/>.

IMS. (2008). "IMS Global Learning Consortium." Retrieved 15-07-2008, from <http://www.imsglobal.org>.

IsaViz. (2008). "IsaViz: A Visual Authoring Tool for RDF." Retrieved 15-07-2008, from <http://www.w3.org/2001/11/IsaViz/>.

JAKUBIC, M. (2007). "Exploring the Knowledge Landscape: Four Emerging Views of Knowledge." *Journal of Knowledge Management* 11(4): 6-19.

JENA (2008). "Jena: Semantic Web Framework for Java." Retrieved 15-07-2008, from <http://jena.sourceforge.net/>.

KAHIGI, E.K., EKENBERG, L., HANSSON, H., TUSUBIRA, F.F., & DANIELSON, M. (2008). "Exploring the E-learning State of Art." *Electronic Journal of E-learning* 6(2).

KAON2 and B. Motic. (2008). "KAON2." Retrieved 15-07-2008, from <http://kaon2.semanticweb.org/>.

KATZ, B., FELSHIN, S., YURET, D., IBRAHIM, A., LIN, J., MARTON, G., MCFARLAND, A. J.

& TEMELKURAN, B. (2002). Omnibase: Uniform Access to Heterogeneous Data for Question Answering. 7th International Workshop on Applications of Natural Language to Information Systems (NLDB), Stockholm, Sweden.

KEWL. (2008). "KEWL : The Knowledge Environment for Web-based Learning." Retrieved 09-06-2008, from <http://sourceforge.net/projects/kewl/>.

KIM, B. (2008). "Social Constructivism : Emerging perspectives on Learning, Teaching and Technology." Retrieved 18-08-2008, 2008, from http://projects.coe.uga.edu/epltt/index.php?title=Social_Constructivism.

KMI. (2005). "Magpie: The Semantic Filter." Retrieved 15-08-2008, from <http://kmi.open.ac.uk/projects/magpie/main.html>.

KUKLA, A. (2000). Social Constructivism and the Philosophy of Science. New York, Routledge.

LEE, M.-C., TSAI, K. H. & WANG, T. I. (2006). An Ontological Approach for Semantic-Aware Learning Object Retrieval. IEEE: Sixth International Conference on Advanced Learning Technologies 2006. 1: 208 - 210

LIU, B. & HU, B. (2006). Migration of the Semantic Web Technologies into E-Learning Knowledge Management. Rough Sets and Knowledge Technology. Berlin, Springer 638-642.

LOPEZ, M. F., GOMEZ, P. A. & SIERRA, J. P. (1999). Workshop on Ontologies and Problems-Solving Methods (KRR5). IJCAI-99, Stockholm, Sweden.

LOPEZ, M. F., GOMEZ, P. A. & SIERRA, J. P. (1999). "Building a Chemical Ontology

Using Methontology and the Ontology Design Environment." IEEE: Intelligent Systems and Their Applications 14(1): 37 - 46

LÓPEZ, M. F., GÓMEZ-PÉREZ, A., DOLORES, M. & AMAYA, R. (2000). Ontology's Crossed Life Cycles. Knowledge Engineering and Knowledge Management Methods, Models, and Tools, Springer.

LOPEZ, V., UREN, V., MOTTA, E. & PASIN, M. (2007). "Aqualog: An Ontology-driven Question Answering System for Organizational Semantic Intranets." Sciondirect Journal of Web Semantics: Science, Services and Agents on the World Wide Web 5(2): 72-105

- MACSTRAVICK, P. (2006). Worldwide and US corporate eLearning 2006-2010 forecast. IDC.
- MAIER, R. (2007). Knowledge Management Systems: Information and Communication Technologies for Knowledge Management. Berlin, Springer.
- MANNING, C. D., RAGHAVAN, P. & SCHÜTZE, H. (2008). Introduction to Information Retrieval. Cambridge, Cambridge University Press.
- MCENTIRE, R., KARP, P. & ABERNTHY (1999). An Evaluation of Ontology Exchange Languages for Bioinformatics. Eight International Conference on Intelligent Systems for Molecular Biology, La Jolla, California.
- MCMAHON, M. (1997) . Social Constructivism and the World Wide Web - a Paradigm for Learning. ASCILITE conference. Perth, Australia.
- MILLER, G. A. (2006). "WordNet: a lexical database for the English language." Retrieved 26-07-2008, from <http://wordnet.princeton.edu/>.
- MISSIKOFF, M. & TAGLINO, F. (2003). SymOntoX: Web-ontology Tool for ebusiness domains. Fourth International Conference on Web Information Systems Engineering 2003 (WISE 2003), Rome, Italy.
- MOODLE (2008). "Moodle - A Free, open Source Course Management System for Online Learning." Retrieved 01-08-2008, 2008, from <http://moodle.org/>.
- MORALES, E., GARCÍA, F., REGO, H., MOREIRA, T. & BARBOSA, H. (2005). Knowledge management for e-learning based on learning objects: a qualitative focus. 6th International Conference on Information Technology Based Higher Education and Training, 2005 (ITHET 2005).
- MOREALE, E. & VARGAS-VERA, M. (2004). "Semantic Services in e-Learning: an Argumentation Case Study." Journal of Educational Technology & Society 7(4): 112-128.
- MUKHLASON, A., MAHMOOD, A. K. & ARSHAD, N. I. (2008). Building Two Ontologies: As Knowledge Base for Automatic Question Answering System (Onto-Aquas) in Moodle: eLearning Management System. Persidangan Kebangsaan Capaian Maklumat dan Pengurusan Pengetahuan 2008 (CAMP'08), Universiti Pertahanan Nasional Malaysia (UPNM).
- MURRAY, T. (2003). "MetaLinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks." International Journal of Artificial Intelligence and Education 13(1).

NAEVE, A. (2001). a new form of Knowledge Management Tool. 2nd European Web-based Learning Environments Conference (WBLE 2001). Lund, Sweden.

NATALYA, NOY, F. & MUSEN, M. A. (2000). PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. Seventeenth National Conference On Artificial Intelligence (AAAI-2000).

NAVIGLI, R. & VELARDI, P. (2004). "Learning Domain Ontologies from DocumentWarehouses and Dedicated Web Sites." *Journal of Computer Linguist* 30(2): 151–179.

NOVAK, J. D. & CAÑAS, A. J. (2006). The Theory Underlying Concept Maps and How to Construct Them. Technical Report IHMC CmapTools 2006-01, Florida Institute for Human and Machine Cognition.

ONTOBROKER (2008). "Ontobroker." Retrieved 15-07-2008, from <http://ontobroker.semanticweb.org/>.

ONTOBUILDER (2008). "OntoBuilder: Fully Automatic Extraction and Consolidation of Ontologies from Web Sources." Retrieved 11-04-2008, from <http://iew3.technion.ac.il:8080/OntoBuilder/>.

ONTOGEN (2006). "OntoGen: semi-automatic ontology editor." Retrieved 08-05-2008, from <http://ontogen.ijs.si/>.

OntoLT. (2008). "Middleware for Ontology Extraction from Text." Retrieved 11-05-2008, from <http://olp.dfki.de/OntoLT/OntoLT.htm>.

O'REILLY, T. (2005). *What is Web 2.0 – Design Patterns and Business Modells for the Next Generation Software*, O'Reilly.

OWL. (2004). "OWL Web Ontology Language Overview W3C Recommendation 10 February 2004 " Retrieved 25-07-2008, from <http://www.w3.org/TR/owl-features/>.

PAPASALOUIROS, A., RETALIS, S. & PAPASPYROU, N. (2004). "Semantic Description of Educational Adaptive Hypermedia based on a Conceptual Model." *Journal of Educational Technology & Society* 7(4): 129-142.

PARADISE, A. (2008). *ASTD: 2008 State of the Industry Report*: 32.

PEDRONI, M. (2007). *E-learning and Knowledge Management: Context Structuration*. 2007 Informing Science and IT Education Joint Conference CARID University of Ferrara, Italy.

- PICCOLI, G., AHMAD, R. & IVES, B. (2001). "Web-based virtual learning environments: A research framework and a preliminary assessment of effectiveness in basic IT skill training." *MIS Quarterly* 25(4): 401-425.
- PJB. (2008). "PHP/Java Bridge." Retrieved 03-11-2008, from <http://php-java-bridge.sourceforge.net/pjb/>.
- Protégé. (2008). "Protégé Ontology Editor." Retrieved 01-02-2008, from <http://protege.stanford.edu/>.
- QI, H. & SUN, S. (2006). Knowledge Management System Based on Semantic Web in e-Learning Community. *Technologies for E-Learning and Digital Entertainment*. Berlin, Springer 3942/2006: 188-192.
- RAO, M. (2005). Knowledge Management Tools and Techniques: Practitioners and Experts Evaluate KM Solutions. Elsevier Butterworth-Heinemann 200 Wheeler Road, Burlington, MA 01803, USA, Elsevier Inc.
- RDFS. (2004). "RDF Vocabulary Description Language 1.0: RDF Schema W3C Recommendation 10 February 2004." Retrieved 04-04-2008, from <http://www.w3.org/TR/rdf-schema/>.
- ROITMAN, H. & GAL, A. (2006). OntoBuilder: Fully Automatic Extraction and Consolidation of Ontologies from Web Sources using Sequence Semantics. *EDBT 2006 Workshops*.
- ROSENBERG, M. J. (2006). Beyond E-learning: Approaches and Technologies to Enhance Organizational Knowledge, Learning, and Performance. San Francisco, CA., Pfeiffer.
- SAMPSON, D. G., LYTRAS, M. D., WAGNER, G. & DIAZ, P. (2004). "Ontologies and the Semantic Web for E-learning." *Journal of Educational Technology & Society* 7(4): 26-28.
- SCADAMALIA, M. (2004). "CSILE/Knowledge Forum." *education and Technology: An encyclopedia* (pp. 183-192). Santa Barbara: ABC-CLIO.
- SESAME (2008). "Sesame: a Java framework for storing, querying and inferencing for RDF." Retrieved 15-07-2008, from <http://sourceforge.net/projects/sesame/>.
- SHADBOLT, N., HALL, W. & BERNERS-LEE, T. (2006). "The Semantic Web Revisited " *IEEE: Journal Of Intelligent Systems* 21(3): 6.
- SKOS. (2008). "Simple Knowledge Organisation System (SKOS) W3C Working Draft 25 January 2008." Retrieved 06-09-2008, from <http://www.w3.org/TR/2008/WD-skos-reference-20080125/>.

SNOBASE (2003). "IBM Ontology Management System." Retrieved 20-07-2008, from <http://www.alphaworks.ibm.com/tech/snobase>.

SOWA, J. F. (2000). Knowledge Representation: Logical, Philosophical and Computational Foundations Brooks Cole Publishing, Pacific Grove Publishing, Inc.

STOJANOVIC, L., STAAB, S. & STUDER, R. (2001). eLearning based on the Semantic Web. World Conference on the WWW and Internet (WebNet2001). Orlando, Florida, USA.

STOREY, M.-A., NOY, N. F., MUSEN, M., BEST, C., FERGERSON, R. & ERNST, N. (2002). Jambalaya: An Interactive Environment for Exploring Ontologies. 7th International Conference on Intelligent User Interfaces, San Francisco, California, USA.

STUTT, A. & MOTTA, E. (2004). "Semantic Learning Webs." Journal of Interactive Media in Education 1(1).

SUMMIT, S. (2008). "C Programming FAQs: Frequently Asked Questions." Retrieved 12-05-2008., from <http://c-faq.com/>.

SUN (2008). "JDBC (Java Database Connectivity) Introduction." Retrieved 04-04-2008, from <http://java.sun.com/docs/books/tutorial/jdbc/overview/index.html>.

SURE, Y. & STUDER, R. (1999). Content-Driven Knowledge Management Tools through Evolving Ontologies : On-To-Knowledge Methodology. On-To-Knowledge Research Publication : Institute AIFB, University of Karlsruhe.

SWARTOUT, B., RAMESH, P., KNIGHT, K. & RUSS, T. (1997). Toward Distributed Use of Large-Scale Ontologies. AAAI Symposium on Ontological Engineering. Stanford California USA.

SWOOP. (2008). "SWOOP Hypermedia-based OWL Ontology Browser and Editor." Retrieved 05-05-2008, from <http://www.mindswap.org/2004/SWOOP/>.

TEXTTOONTO (2003). "TextToOnto : Text To Ontology Tool." Retrieved 25-07-2008, from <http://ontoware.org/projects/text2onto/>.

TIWANA, A. (1999). Knowledge Management Toolkit, The: Practical Techniques for Building a Knowledge Management System, Prentice Hall PTR.

TRIPLE (2008). "Triple:an RDF query, inference, and transformation language for the Semantic Web." Retrieved 15-07-2008, from <http://triple.semanticweb.org/>.

TSARKOV, D. & HORROCKS, I. (2006). FaCT++ Description Logic Reasoner: System Description. the International Joint Conference on Automated Reasoning (IJCAR 2006) Lecture Notes in Artificial Intelligence, Springer.

UREN, V., CIMIANO, P., IRIA, J., HANDSCHUH, S., VARGAS-VERA, M., MOTTA, E. & CIRAVEGNA, F. (2005). "Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art." *Journal of Web Semantics* 4(1).

USHOLD, M., USCHOLD, R. M. & KING, M. (1995). Towards a Methodology for Building Ontologies. IJCAI95: Workshop on Basic Ontological Issues in Knowledge Sharing. Montreal.

W3C. (2001). "W3C Semantic Web Activity." Retrieved 11-07-2008, from <http://www.w3.org/2001/sw/>.

W3C. (2004). "Web-Ontology (WebOnt) Working Group." Retrieved 15-08-2008, from <http://www.w3.org/2001/sw/WebOnt/>.

W3C. (2008). "Web World Web Consortium." Retrieved 15-07-2008, from <http://www.w3.org>.

WANG, S. (2008). "Ontology of Learning Objects Repository for Pedagogical Knowledge Sharing." *Interdisciplinary Journal of E-Learning and Learning Objects* 4(1).

WOELK, D. & AGARWAL, S. (2002). Integration of e-Learning and Knowledge Management. E-Learn 2002 World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education, Montreal, Canada.

YANG, S. J. H., CHEN, I. Y.-L. & SHAO, N. W. Y. (2004). "Ontology Enabled Annotation and Knowledge Management for Collaborative Learning in Virtual Learning Community." *Journal of Educational Technology & Society* 7(4): 70-81.

YORDANOVA, K. (2007). Integration of Knowledge management and E-learning – commonfeatures. International Conference on Computer Systems and Technologies - CompSysTech'07.

PUBLICATIONS

Year	Title	Publication In
2007	Enhanced Moodle CMS using Semantic Web Technology: Developing an Architecture of Ontologydriven Automatic Question Answering System. Co Author : Ahmad Kamil Mahmood, Noreen Izza Arshad	MMU International Symposium on Information And Communication Technologies 2007 : 19th - 20th November 2007 (http://m2usic.mmu.edu.my/07/index.html) Accepted for Publication (but not published): <ul style="list-style-type: none"> • Conference On Teaching and Learning in Higher Education (CTLHE) 2007, Universiti Putra Malaysia. • Rural ICT Development Conference (RICTD) 2007, Universiti Utara Malaysia. • International Conference on Enterprise Information Systems and Web Technologies 2008, Orlando, FL, USA.
2008	Building Two Ontologies: As Knowledge Base for Automatic Question Answering System (Onto-Aquas) in Moodle eLearning Management System. Co Author : Ahmad Kamil Mahmood, Noreen Izaa Arshad	Persidangan Kebangsaan Capaian Maklumat dan Pengurusan Pengetahuan 2008 (CAMP'08), Universiti Pertahanan Nasional Malaysia (UPNM) (http://www.upnm.edu.my/CAMP08/). Accepted for Publication (but not published): <ul style="list-style-type: none"> • e-CASE 2008 The 2008 International Joint Conference on e-Commerce, e-Administration, e-Society, and e Education, Bangkok, Thailand. • International Conference on Enterprise Information Systems and Web Technologies 2008, Orlando, FL, USA.
2008	Developing an Architecture of Ontology-driven Automatic Question Answering System (Onto-Aquas) in Moodle eLearning Management System. Co Author : Ahmad Kamil Mahmood, Noreen Izaa Arshad	NATIONAL POSTGRADUATE CONFERENCE ON ENGINEERING, SCIENCE AND TECHNOLOGY 2008 (NPC 2008) (www.utp.edu.my/npc08/).
2008	Semi Automatic Ontological Knowledge Base Construction From Learning Materials in eLearning Management System.	IEEE: International Symposium On Information Technology 2008 (itsim 2008) (http://www.itsim.org/).

	Co Author : Ahmad Kamil Mahmood, Noreen Izaa Arshad	Accepted for Publication (but not published): <ul style="list-style-type: none"> • MWAIS (Midwest Association for Information System) 2008 Conference , University of Wisconsin, Eau Claire, USA.
2008	Knowledge Management Driven E-Learning System: Developing An Automatic Question Answering System with Expert Locator Feature Co Author : Ahmad Kamil Mahmood, Noreen Izaa Arshad, Ahmad Izzudin	Journal of KM Practice Vol. 9 No. 3 2008 available for viewing at http://www.tlinc.com/jkmpv9n308.htm September 30, 2008

APPENDIX I

E-LEARNING

There are various spelling of e-Learning, e.g., E-Learning, E-learning, e-Learning, e-learning, eLearning and Elearning. In this thesis, the term ‘e-Learning’ is used. Many terms are also used to define e-Learning. For example, some of these terms include online learning, virtual learning, distance learning, e-Learning, Web-based learning, Web-based training and computer-based training. Online learning and Web-based training seem to be the most popular interchangeable terms for e-Learning. This section shall provide some definitions, scope and fundamental aspects, state of the art underlying technology of e-Learning.

I.1 Definition and Scope of e-Learning

The common definition of e-Learning is instruction accessed electronically on a computer. This instruction could be a class, a course or a discussion and could look like a book, a movie, a Web page, a game or a combination of those things. Some other notable definitions of e-Learning are listed below:

1. e-Learning is instruction that is delivered electronically, in part or wholly – via a Web browser, through the Internet or an intranet, or through multimedia platforms such as CD-ROM or DVD (Horton 2000).
2. e-Learning is a structured, purposeful use of electronic system or computer in support of the learning process (Allen and Allen 2002).
3. e-Learning is training delivered on a computer (including CD-ROM, Internet or Intranet) that is designed to support individual learning or organizational performance goals (Clark and Mayer 2002).
4. Web-based training (an alternate term for e-Learning) is the integration of instructional practices and Internet capabilities to direct a learner towards a specified level of proficiency in a specified competency (Conrad 2000).

5. e-Learning is defined as the use of Internet technologies to create and deliver rich learning environment that includes a broad array of instruction and information resources and solutions, the goal of which is to enhance individual and organizational performance (Rosenberg 2006).
6. e-Learning also can be defined broadly as any use of Web and Internet technologies to create learning experiences. Such an inspirationally open-ended definition, though, does little to help you narrow in on the specific tools needed for an individual project (Horton and Horton 2003).

There are some types of e-Learning. (Horton and Horton 2003) categorized e-Learning into six types. The six types of e-Learning are presented in Table I-1.

Table I- 1 e-Learning Categorization

Type	Aims
Learner-led e-Learning (standalone or self directed e-Learning)	Delivering highly effective learning experiences to independent learners. Its contents (Web page, multimedia, etc.) housed in Web server and accessed using Web browser.
Facilitated e-Learning	Combining the reliance on Web content found in learner-led e-Learning with the collaborative facilities found in <i>instructor-led e-learning</i> .
Instructor-led e-Learning	Using Web technology to conduct conventional classes with distant learners. These classes use a variety of real-time technologies, such as video and audio conferencing, chat, screen-sharing, polling, whiteboards, and the plain old telephone.
Embedded e-Learning	Providing just-in-time training. It is usually embedded in computer programs, help files, Web pages or network applications. It may even be a component of an Electronic Performance Support System (EPSS).
Tele Mentoring	Using the latest technologies for one of the oldest forms of learning. They use video conferencing, instant messaging, Internet telephones and other collaboration tools to help mentors guide the development of protégés tend to be long term and focus on career development.
e-Coaching.	Same like Tele Mentoring but has a more short-term, project-specific goal.

I.2 e-Learning Standards and Pedagogical Approaches

To be successful in implementation, learning object standards and pedagogical aspects need to be considered in e-Learning system. A learning object is a unit of digital resource that can be shared to support teaching and learning (Wiley 2000). The existing learning object standards are IMS Global Learning Consortium (IMS 2008), AICC (AICC 2008) and ADL's (Advanced Distributed Learning) SCORM (Shareable Content Object Reference Model)(ADL 2008), just to name a few. Pedagogical aspect consists of elements that attempt to define structures or units of educational material. For example, this could be a lesson, an assignment, a multiple choice question, a quiz, a discussion group or a case study. Some of existing pedagogical approaches are constructivism (Honebein et al. 1993; Fosnot 1996), active learning, inquiry-based learning, learning by doing, learning by discovery, collaborative learning, associative learning, and etc. (see Table I-2).

Table I- 2 Pedagogical Approaches of e-Learning (Bjørke 2008)

Approach	Description
Instructive	Transmission of theoretical units of information in a traditional classroom situation: the teacher in front lectures the students facing the teacher. Communication between the students is discouraged in this approach.
Constructivism	This approach argues that people have to be active learners and construct knowledge themselves based on what they already know (learner centred). The knowledge is seen as more subjective, dynamic and expanding rather than objective and static. The main tasks here are processing, understanding of information and making sense of the surrounding world. The learner has a clear responsibility for his/her own learning.
Social Constructivism or Socio-cultural pedagogy	Social constructivism means that the student joins a knowledge-generating community and in collaboration with others to solve real problems and assignments in an authentic context as part of their study. In a social constructivist environment, the teacher will be him/her-self, though an “old-timer” (a master), to some extent be a learner together with his/her students, as the generic skills of collaboration, problem-solving and creating new knowledge are important goals by themselves.

I.3 e-Learning Tools and Technologies: State Of The Art

(Kahiigi et al. 2008) identified the e-Learning technologies evolution start from 1995 to 2007. These technologies include CD-ROM media, Learning Management System (LMS) / Content Management System (CMS) / Learning Content Management System (LCMS), multimedia/virtual communities, game authoring tool, and personalized tools (see Figure I-1).

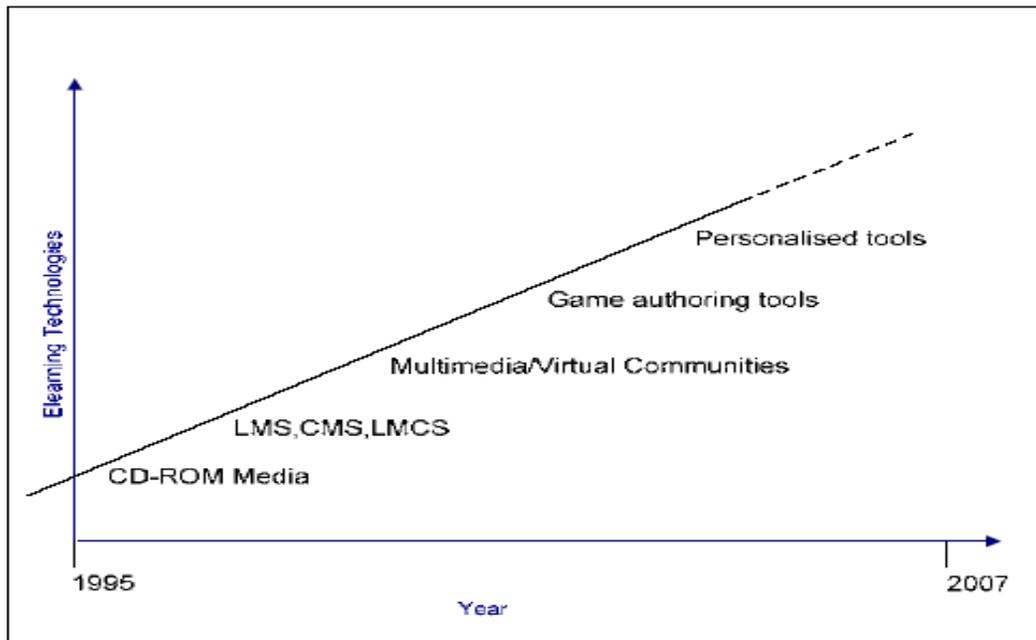


Figure I- 1 Evolution of E-learning Technology (Kahiigi et al. 2008)

CD-ROM Media have been used to deliver learning material to students on distance programs. This media was adopted mainly in the early 1990's and supports learning content in text or multimedia formats. Use of CD-ROM media encourages independent learning where learners learn by executing special training programs on the computer irrespective of internet connectivity. This tool is commonly used for computer based training such as those usually offered as tutorial with new software and tutorials for learning foreign languages.

LMS are a whole range of information systems and processes that contribute directly or indirectly to learning and to the management of that learning. They are primarily

developed to provide online learning services for students, teachers and administrators. Examples of LMS include (KEWL 2008) and (Blackboard 2008). The former example is open source and the latter is commercial.

CMS such as (Moodle 2008) are developed to facilitate the collaborative creation of content, organization, control and to manage the publication of documents in a centralized environment.

LCMS are mostly Web-based systems that combine the management and administrative functionalities of LMS and CMS to author, approve, publish and manage learning content. An example of such technologies is the (Builder 2008). The common features offered by LCMS are presented in Table 2-3.

Table I- 3 LMS/CMS Features (Cole 2005)

Feature	Blackboard	WebCT	Moodle
<i>Upload and share documents</i>	Y	Y	Y
<i>Online discussions</i>	Y	Y	Y
<i>Grade discussions / participation</i>	N	Y	Y
<i>Online chat</i>	Y	Y	Y
<i>Student peer review</i>	N	N	Y
<i>Online quizzes / surveys</i>	Y	Y	Y
<i>Online grade book</i>	Y	Y	Y
<i>Student submission of documents</i>	Y	Y	Y
<i>Self-assessment of submission</i>	N	N	Y
<i>Student workgroups</i>	Y	Y	Y
<i>Lessons with paths</i>	Y	Y	Y
<i>Student journals</i>	N	N	Y
<i>Embedded glossary</i>	N	N	Y

* Y = Yes (available), N= No (not available)

Multimedia Communities and Virtual Worlds have transformed e-Learning environments from disseminating only text based to one that incorporates multimedia content. It is the online delivery of information, communication education and training providing a new set of technologies that can add to all the traditional learning modes CD-ROM, and traditional computer based training. An example of multimedia communities is the CSILE/Knowledge Forum (Scadamalia 2004).

Game Authoring Technologies aim at enhancing and facilitating the students' learning process through built-in simulations and interactions. Online games range from text based games to games that incorporate graphics and virtual worlds populated by many players simultaneously. They create social communities that facilitate knowledge sharing and creation; a concept commonly referred to as "edutainment".

Personalized learning is a learning approach that facilitates and supports individualized learning. Each learner has a learning path that caters for learners learning needs and interests in a productive and meaningful way. One of the attributes of personalized learning is the ability to dictate the students' learning.

The development of e-Learning content has been highly facilitated by advances in e-Learning technologies. Content development has been transformed from text based to include multi-media supported content that caters for personalized learning, as illustrated in Figure I-2.

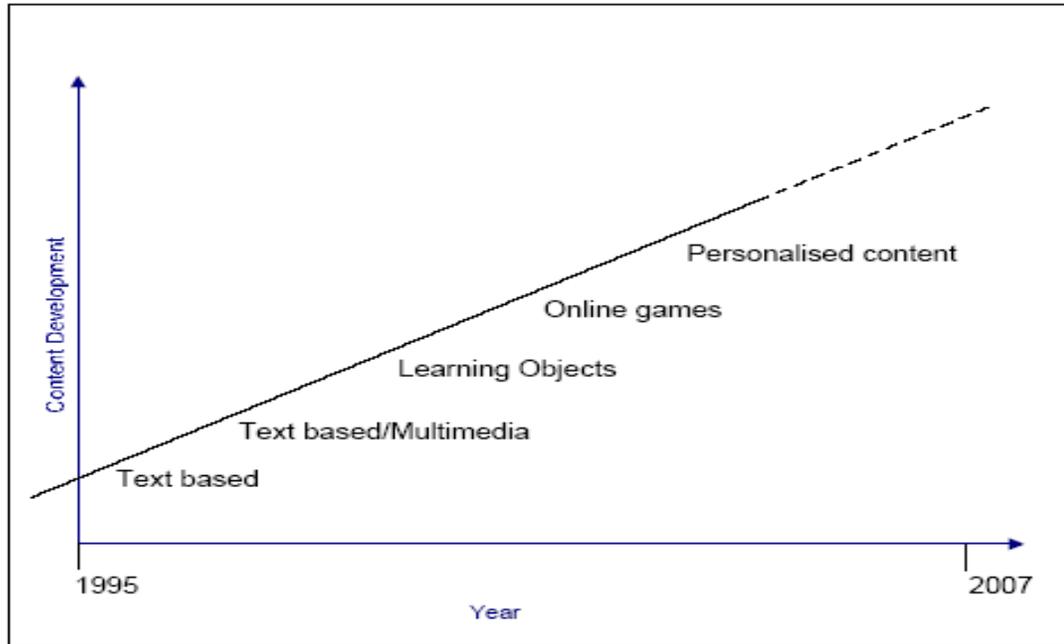


Figure I- 2 Evolution of E-learning Content (Kahiigi et al. 2008)

I.4 e-Learning Technology Trend

Since most of e-Learning technology is Web based. The e-Learning technology followed Web technology development. The Web technology that is known as Web 1.0 and Web 2.0 has derived e-Learning technology jargon like e-Learning 1.0 and e-Learning 2.0.

e-Learning 1.0 (Ebner 2007) can be characterized as: in the beginning of e-learning one of the first theories was to create a perfect content. With the help of “new media” interactions, animations, simulations and similar were added to the traditional learning material. With other words, the traditional book is replaced by a multimedia accumulation. Comparable to the typical classroom there are a teacher and there are students (learners). The teaching person provides the content in high quality. The learning material is accessible via a LMS. The “new” (in relation to the traditional face-face teaching) components are further tools, like communication tools or interactive exercises. With other words, in many cases the old traditional education form was accomplished with new media. The role of the LMS seems to be on the one side an administrative and on the other side a simple lecture add on.

e-Learning 2.0 is used to refer to new ways of thinking about e-Learning inspired by the emergence of Web 2.0. Web 2.0 its self is first coined by Tim O'Reilly in 2005 (O'Reilly 2005). He described Web 2.0 as architecture of participation: a grassroots user base creates a self-regulating collaborative network. Web 2.0 is characterized by the use of social network such as Weblogs, Wikis, Podcasts, Social Book marking and Web sharing. e-Learning 2.0 perception for the first time by stephen downes (Downes 2006), he mentioned in his article "For all this technology, what is important to recognize is that the emergence of the Web 2.0 is not a technological revolution, it is a social revolution". Therefore Marten Ebner (Ebner 2007) argued that $e\text{-Learning } 2.0 = f^2$ (e-Learning 1.0, Web 2.0, human factor) instead of $e\text{-Learning } 2.0 = e\text{-Learning } 1.0 + \text{Web } 2.0$.

² Function of

APPENDIX II SEMANTIC WEB

The development of internet and World Wide Web (WWW) has changed the way people communicate with each other namely how information is disseminated and retrieved, and how business is conducted. Views on the next stage of the WWW's evolution vary greatly. Many technologists, journalists, and industry leaders have used the term "Web 3.0" to hypothesize about a future wave of Internet innovation. The Semantic Web (SW) is believed as promising next internet technology that will dramatically improve the current WWW and its usage. This section shall demystify this emerging technology and technologies underlying the SW.

II.1 The Fundamental of SW

This section shall describe the fundamental idea of the SW, its definitions and technologies underlying it.

II.1.1 SW Definition

The SW is one of activities from WWW consortium (wwwc 2008), a forum for information, commerce, communication and collective understanding which concern in developing interoperable technologies (specifications, guidelines, software and tools) to lead the Web to its full potential. Tim Berners-Lee, the man behind the WWW, HTTP and HTML technologies, firstly introduced it.

W3C SW activity stated that SW is Web of data that provides a common framework allowing data to be shared and reused across application, enterprise and community boundaries. It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework (RDF), which integrates a variety of applications using XML for syntax and Uniform Resource Identifiers (URIs) for naming. The SW is about two things. The first thing is

common formats for integration and combination of data drawn from diverse sources where on the original Web mainly concentrated on the interchange of documents. The second thing is about language for recording how the data relates to real world objects that allows a person or a machine to start off in one database and then move through an unending set of databases which are connected not by wires but by being about the same thing (W3C 2001). The SW is only a promising technology in its infancy, yet today the SW is attainable (Shadbolt, Hall et al. 2006).

II.1.2 The SW Vision

The first original article about SW appeared in 2001 and was published in *Scientific American* (T. Berners-Lee 2001). Tim Berners Lee presented what the SW is and his vision of the “Semantic Web”. The SW is an extension of the current Web in which information is given in well-defined meaning, better enabling computers and people to work in co-operation. To summary, Tim Berners-Lee has two-part vision for the future of the Web. The first part is to make the Web a more collaborative medium. The second part is to make knowledge in the Web understandable and thus process able by machines. Figure II-1 is Tim Berners-Lee’s original diagram of his vision.

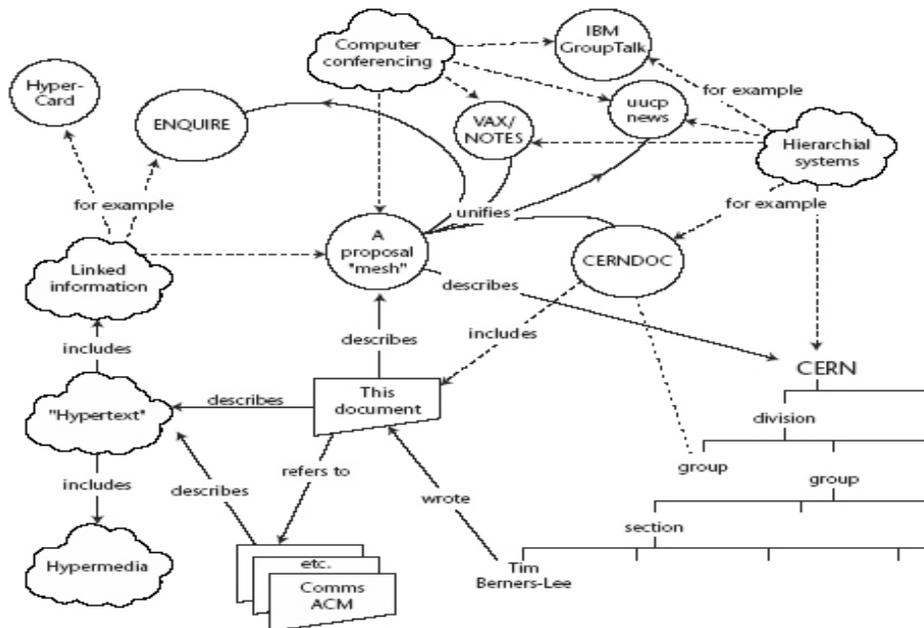


Figure II- 1 Tim Berner Lee, SW Vision (Daconta, Obrst et al. 2003)

Figure II-1 depicts the relations between information items like “includes”, “describes”, and “wrote”. Unfortunately, these relationships between resources are not currently captured on the Web. The technology to capture such relationships is called the RDF that will be described in the following section of this chapter. The key point to understand about Figure II-1 is that the original vision encompassed additional Meta data beyond what is currently in the Web. This additional Meta data is needed for machines to be able to process information on the Web.

Giving information in the Web understandable and process able by machine, Michael C., et.al. (Daconta, Obrst et al. 2003) identified several key problems facing current information technology architectures that will be resolved by SW technology:

1. Information Overload

Information overload is the most obvious problem in need of a solution, and technology experts have been warning this issue for 50 years because of rapid rate of growth in the amount of information available with the propagation of the Internet, email, and now instant messaging.

2. Stovepipe System

A *stovepipe system* is a system where all the components are hardwired to only work together. Therefore, information only flows in the stovepipe and cannot be shared by other systems or organizations that need it. For example, the client can only communicate with specific middleware that only understands a single database with a fixed schema.

3. Poor Content Aggregation

Putting together information from disparate sources is a recurring problem in a number of areas, such as financial account aggregation, portal aggregation, comparison shopping and content mining.

II.1.3 The SW Goals

The goal of SW and SW Technology in its fundamental principle is the creation of semantic metadata that will offer new approach in managing information and process. This semantic metadata is key enabler to organize and find information based on meaning, improve information presentation and integrate information from heterogeneous sources.

II.1.4 The SW Component

The SW components are usually represented in a “layered cake”, which initially presented by Tim Berner Lee to describe the main layer of the SW design and vision (T. Berners-Lee 2001).

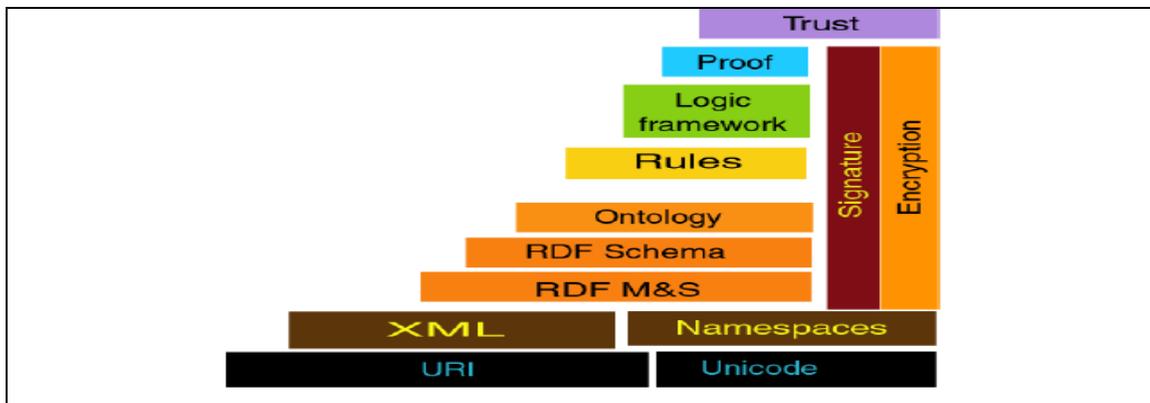


Figure II- 2 The Semantic Web Layer Cake (T. Berners-Lee 2001)

As depicted in Figure II-2 at the bottom layered cake is XML. XML is the syntactic foundation layer of the SW that lets one writes structured Web documents with user-defined vocabulary. The technologies that XML is built upon are Unicode characters and URIs. The Unicode characters allow XML to be authored using international characters. URIs are used as unique identifiers for concepts in the SW.

RDF is a basic data model (it is like the entity-relationship model) for writing simple statements about Web objects (resources). The RDF data model does not rely on XML, but RDF has an XML-based syntax. Based on this RDF, a logical assertion - the smallest

expression of useful information- can be made by modeling the key parts of a sentence (connecting a subject to an object with a verb).

RDF Schema provides modeling primitives for organizing Web objects into hierarchies. Key primitives are classes and properties, subclass and sub property relationships, and domain and range restrictions. RDF Schema is based on RDF. RDF Schema can be viewed as a primitive language for writing Ontologies. However, there is a need for more powerful Ontology languages that expand RDF Schema and allow the representations of more complex relationships between objects. The logic layer is used to enhance the Ontology language further and to allow the writing of application-specific declarative knowledge.

The next upper layer is a rule. With XML, RDF and inference rules, the Web can be transformed from a collection of documents into a knowledge base. An inference rule allows you to derive conclusions from a set of premises.

The proof layer involves the actual deductive process as well as the representation of proofs in Web languages (from lower levels) and proof validation. Finally, the Trust Layer will emerge with digital signatures and other kinds of knowledge, based on recommendations by trusted agent or on rating and certification agencies and consumer bodies. Sometimes “Web of trust” is used to indicate that trust will be organized in the same distributed and chaotic way as the WWW itself. Being located at the top of the pyramid, trust is a high-level and crucial concept: the Web will only achieve its full potential when users have trust in its operations (security) and in the quality of information provided.

II.2 Ontology

As depicted by Semantic Web layer cake in previous section (Figure II-2), at the heart of all SW applications is the use of Ontologies. Thus, the successful of SW depends on the successful of Ontology development. This section shall give a more detail understanding on Ontology.

II.2.1 Definition and Scope

The term “Ontology” is known not only in computer sciences discipline but also in other disciplines such as philosophy, linguistics, etc. Bellows are some definition of Ontology from point of view of computer science discipline:

1. An Ontology defines the common words and concepts (the meaning) used to describe and represent an area of knowledge (Daconta, Obrst et al. 2003).
2. An Ontology is an engineering product consisting of a specific vocabulary used to describe a part of reality, plus a set of explicit assumptions regarding the intended meaning of that vocabulary (Guarino 1998).
3. In other words, Ontology is the specification of a conceptualization (Gruber 1993).
4. Ontology is a catalog of the types of things that are assumed to exist in a domain of interest D from the perspective of a person who uses a language L for the purpose of talking about D . The types in the Ontology represent the *predicates*, *word senses*, or *concept and relation types* of the language L when used to discuss topics in the domain D (Sowa 2000).

Thus, to sum up prior definitions of Ontology, Ontologies can be defined as a formal specification, shared understanding and machine executable model of a domain of interest.

Ontology also related but different with other concepts like controlled vocabulary, taxonomy, thesauri, Topic Maps (TMs), Simple Knowledge Organization Systems (SKOS), semantic network, predicate logic and extended ER-Models (See Figure II-3).

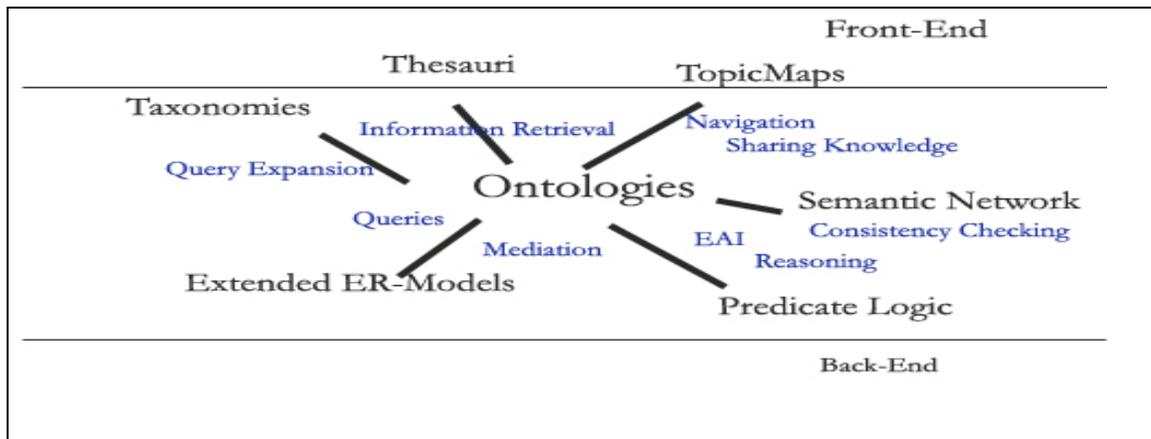


Figure II- 3 Ontologies and Their Relatives

A controlled vocabulary is a lightweight Ontology that simply lists a set of terms. Taxonomy is a set of terms that are arranged into a hierarchy. It is an extension of a controlled vocabulary. Taxonomy does not define attributes of these terms, nor does it define relationships between the terms. Thesauri are extensions of taxonomy and allow for expression of some relationships between terms. TMs are an ISO standard for the representation and interchange of knowledge, with an emphasis on the find ability of information. TMs are an alternative URI based Ontology language to the W3C’s RDF and RDFS/OWL. TMs provide powerful search and index functionalities (R. McEntire and Hodnett 1999). SKOS provides a model for expressing the basic structure and content of concept schemes (thesauri, classification schemes, subject heading lists, taxonomies, terminologies, glossaries and other types of controlled vocabulary)(SKOS 2008).

An example of Ontology in graphical representation can be seen in Figure II-4. It models knowledge or domain of interest of an organization. Some important of concepts in organization are identified such as “management_employee”, “employee”, “manager”, “president”, “department”, “division”, “group” and etc. Each concept has some attributes. For example, concept “employee” has attributes: “employee_no”, “employee_of”, “organization” and name (string). The Ontology also identify relations between concepts

such as “manages” that relate “management_employee” and employee; and “is a” that relate “employee” and “Staff_Employee”.

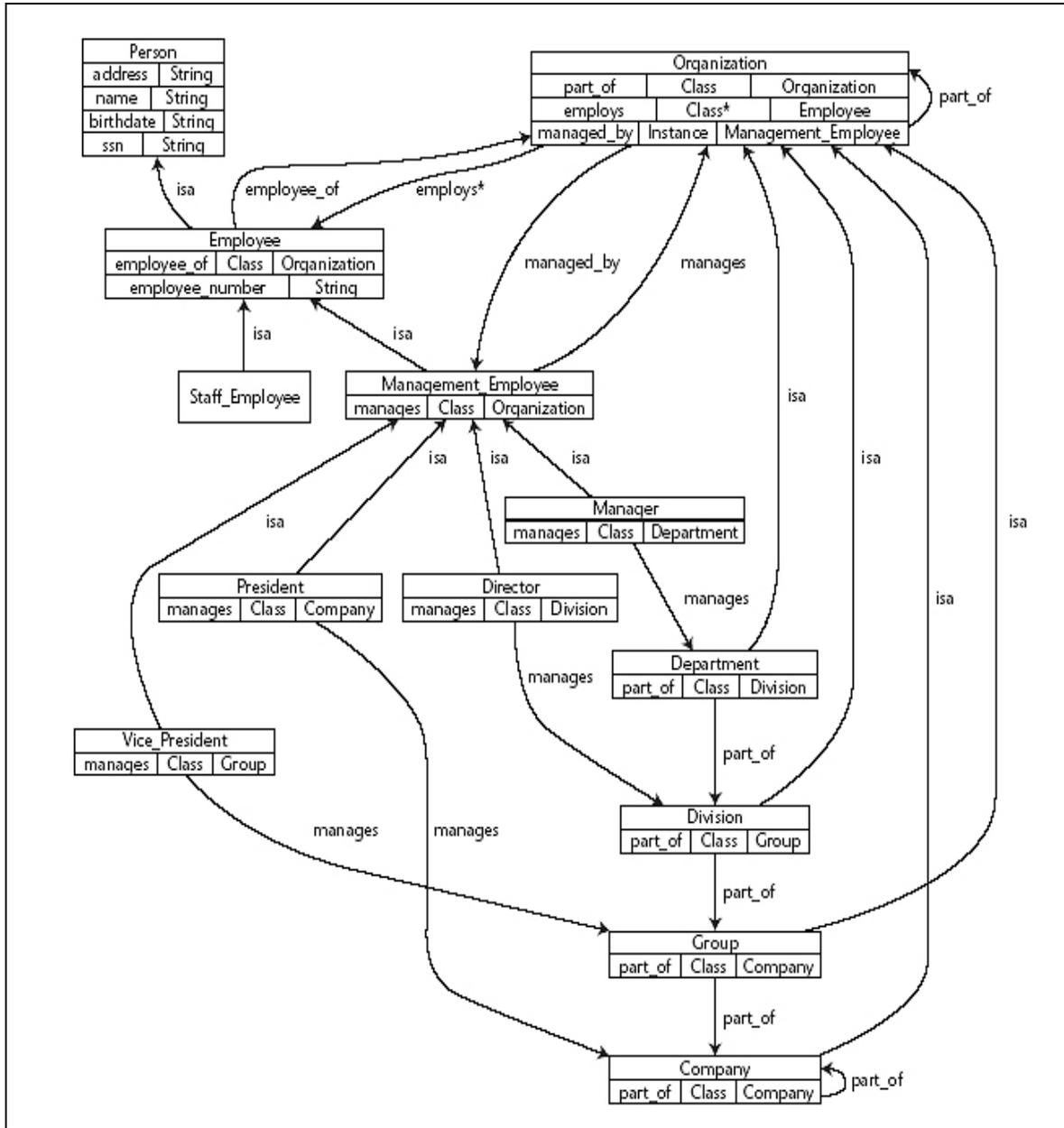


Figure II- 4 Graphical Ontology Example: Human Resource (Daconta, Obrst et al. 2003)

II.2.2 Classification of Ontology

Ontology can be classified based on some criteria such as Ontology development purpose, expressiveness, specificity and formality. Bellows are detailed explanation of this classification.

1. Classification According to Purpose

Borgo et al.(Stefano Borgo 2002) classify Ontology according to its purpose into:

- a. *Application Ontology* used during run time of a specific application putting constraints on the axiomatization for the terminological service i.e. the reasoner.
- b. *Reference Ontology* used during development time of applications for mutual understanding and explanation between (human or artificial) agents belonging to different communities, for establishing consensus in a community that needs to adopt a new term or simply for explaining the meaning of a term to somebody new to the community.

2. Classification according to expressiveness

- a. *Heavyweight* Ontologies try to specify the intended meaning of a vocabulary as precisely as possible. Their primary motivation is to enable mutual understanding in a heterogeneous environment. Their drawback is that they may be hard to develop and to reason with, both because of the number of axioms and the expressiveness of the language adopted. Heavyweight Ontologies are extensively axiomatized and, thus, represent ontological commitment explicitly. The purpose of the axiomatization is to exclude terminological and conceptual ambiguities, which are due to unintended interpretations. Every heavyweight Ontology can have a lightweight version. As with all dimensions, the borderline between lightweight and heavyweight is not clearly delimited.
- b. *Lightweight* Ontologies, on the other hand, may consist of a minimal set of axioms written in a language of limited expressiveness. Such Ontology may support only a limited set of specific services, intended to be shared

among users who already agree on the underlying conceptualization. Ontologies are often reduced to a simple taxonomy of concepts and a small number of associations. Such Ontologies are classified as lightweight Ontologies because they are hardly axiomatized, as opposed to heavyweight Ontologies. Lightweight Ontologies are used when the intended meaning of the concepts used by the community is more or less known in advance by all members and the Ontology can be limited to those structural relationships among concepts that are considered as relevant.

3. Classification according to specificity

The classification according to specificity introduces three layers: generic, core and domain Ontologies (Guarino 1998).

- a. *Generic Ontology*, the concepts defined by this layer are considered generic across many fields. Typically, generic Ontologies (synonyms are "upper level" or "top-level" Ontology) define concepts such as state, event, process, action, component, etc.
- b. *Core Ontology*, defining concepts that are generic across a set of domains. Therefore, they are situated in between the two extremes of generic and domain Ontologies. The borderline between generic and core Ontologies is not clearly defined because there is no exhaustive enumeration of fields and their conceptualizations. However, the distinction is intuitively meaningful and useful for building libraries.
- c. *Domain Ontology*, expressing conceptualizations that are specific for a universe of discourse. The concepts in domain Ontologies are often defined as specializations of concepts in the generic and core Ontologies.

4. Classification according to formality

- a. *An informal Ontology* may be specified by a catalog of types that are either undefined or defined only by statements in a natural language.

- b. *A formal Ontology* is specified by a collection of names for concept and relation types organized in a partial ordering by the type-subtype relation. Formal Ontologies are further distinguished by the way the subtypes are distinguished from their super types:
 - i. *An axiomatized Ontology* distinguishes subtypes by axioms and definitions stated in a formal language such as logic or some computer-oriented notation that can be translated to logic.
 - ii. *Prototype-based Ontology* distinguishes subtypes by a comparison with a typical member or *prototype* for each subtype. Large Ontologies often use a mixture of definitional methods: formal axioms and definitions are used for the terms in mathematics, physics, and engineering; and prototypes are used for plants, animals, and common household items.

II.2.3 Ontology Representation Languages

There are some emerging SW languages for representing Ontologies. Some of them have been W3 consortium recommendation. These languages differ on their expressiveness capability. The widely used Ontology representation languages include the RDF and RDFS (usually abbreviated RDF/S or RDF(S)), Defense Advanced Research Projects Agency (DARPA) Agent Markup Language (DAML) + Ontology Inference Layer (OIL) (usually abbreviated DAML+OIL) and the Web Ontology Language (OWL). A more detail explanation of these Ontology languages is described below:

(i) DAML+OIL

DAML is a SW Ontology language that was developed as part of the DARPA DAML program, which originated in 2000 and continues to the present. Soon after the initial United States-based DAML language version had emerged, DAML researchers and the comparable European Union-based OIL language researchers became aware of each other's effort. There have subsequently been two versions of the combined language, now called DAML+OIL: December 2000 and March 2001. More recently, the DAML-Service

(DAML-S) extension has emerged. DAML-S is really a collection of Ontologies represented in DAML+OIL that address the semantics of Web services, including services modeled as processes, resources, service profiles, service models and service groundings (i.e., the concrete realization of the abstractly specified service components, and comparable to the Web Service Description Language's notion of binding). The features of DAML+OIL can be seen in DAML site (DAML 2002).

(2) RDF(S)

RDF is general-purpose language for representing information in the Web. The fundamental concepts of RDF are resources, properties and statement. Resource can be defined as an object, a “thing” people want to talk about. Resources may be student, course, lecturer, book, author, and so on. Every resource has URI. A URI can be URL (or Web address) or some other kind of unique identifier; an identifier does not necessarily enable access to resource. URI schemes have been defined not only for Web-locations but also for such diverse objects as telephone numbers, ISBN numbers and geographic locations.

Properties are a special kind of resources. They describe relations between resources, for example “written by”, “age”, “title”, and so on. Properties in RDF are also identified by URIs (and in practice by URLs). This idea of using URIs is to identify “things” and the relations between them is quite important. This choice gives us in one stroke a global, worldwide, unique naming scheme. The use of such a scheme greatly reduces the homonym problem that has plagued distributed data representation until now.

Statements assert the properties of resources. A statement is an object-attribute-value triple, consisting of a resources, a property, and a value. Values can be either resources or literals. Literals are atomic values (strings), the structure of which no needs to be discussed further.

RDF is universal language that lets users describe resources using their own vocabularies. RDF does not make assumptions about any particular application domain, nor does it define the semantics of any domain. It is up to the user to do so in RDF Schema (RDFS).

(iii) OWL (Web Ontology Language)

Web Ontology Language (sometimes referred to as Ontology Web Language) is the most expressive of the Ontology languages currently defined or being defined for the SW. Unlike DAML+OIL, OWL is originating as a World Wide Web Consortium (W3C) sponsored language (W3C 2004). The W3C's Web Ontology Working Group was formed in November 2001, and the first official version of OWL is anticipated to be available in early 2003. The OWL developers began with DAML+OIL as the initial candidate for an expressive Web Ontology language, and evaluated DAML+OIL with respect to its known problems and the sufficiency of its semantic expressivity for developing Ontologies usable on the Web. Initially, use cases were developed to drive out requirements, and then the requirements for an Ontology language were codified. An abstract syntax and semantics, then the full language syntax (at least, up to this point; there are still some issues under discussion), and its semantics were defined.

OWL has three levels of language: OWL Lite, OWL DL (for Description Logic) and OWL Full. These three levels are in increasing order of expressivity. The higher levels of the language contain the lower levels and so are said to extend the lower levels. A valid conclusion in OWL Lite is still a valid conclusion in OWL DL and OWL Full and a valid conclusion in OWL DL is a valid conclusion in OWL Full, but not necessarily in OWL Lite. A valid conclusion in OWL Full is not necessarily a valid conclusion in either OWL DL or OWL Lite. Table II-1 presents the levels of language in OWL.

Table II- 1 The OWL Language Level

Language Level	Description
OWL Full	The complete OWL. For example, a class can be considered as both a collection of individuals and an individual itself.
OWL DL	Slightly constrained OWL. Properties cannot be individuals, for example. More expressive cardinality constraints.
OWL Lite	A simpler language, but one that is more expressive than RDF/S. Simple cardinality constraints only (0 or 1).

II.3 Tool For SW

One of the key ideas of SW is interoperability of formalized knowledge. Knowledge is represented in standard formalisms with defined semantics that can be exchanged and used in different applications. All the agents can rely on the standard formalisms to implement this interoperability. It is especially important for tools to respect those standards. Dieng et. al. distinguish five categories of tools or standard components, necessary to set up a “Semantic Web”: first, for the formalization of Ontologies, then for the expression of knowledge and its exploitation through visualization, search and navigation (Dieng-Kuntz and Dieng-Kuntz 2004). Some of these Ontology tools are presented as below:

II.3.1 Ontology editors

To be exchanged and manipulated on the SW, knowledge must be formalized through shared Ontologies. Ontology editors for OWL like Protégé (Protégé 2008) or SWOOP (Swoop 2008) offer dedicated interfaces to manually author Ontologies and generate OWL files. Those Ontology editors share the following characteristics:

- They allow browsing the Ontology structure, usually making use of hierarchical tree views.
- They offer full edition of the Ontology.

Both mentioned tools are generic editors. They require a good knowledge of OWL formalism and experts or so-called “knowledge engineers” can only use them. When the Ontologies have to be built directly by other categories of users, such as domain experts, dedicated interfaces and editors are necessary.

II.3.2 Ontology Management System (OMS)

An OMS is equivalent to a database management system and allows an application to manipulate and query Ontology without worrying about how the Ontology is stored and accessed, how queries are processed and how query results are retrieved. Ontology editing is not viewed as the most important part of an OMS and in instances where editing capabilities are absent it can be used in connection with an Ontology editor such as Protégé 2000 (Casely-Hayford 2005).

An example of OMS is IBM OMS. It (also known as SNOBASE that is contraction of Semantic Network Ontology Base) is a framework for loading Ontologies from files and via the Internet and for locally creating, modifying, querying and storing Ontologies (snobase 2008) and at present it has been merged into IBM Integrated Ontology Development Toolkit -an Ontology toolkit for storage, manipulation, query and inference of Ontologies and corresponding instances(IBM 2007).

The IBM OMS provides a mechanism for querying Ontologies with a user friendly programming interface for interacting with vocabularies of standard Ontology specification languages such as RDF, RDF Schema, DAML+OIL, and OWL. The system uses an inference engine, an Ontology persistent store, an Ontology directory and Ontology source connectors. Applications are able to query against the created Ontology models and the inference engine deduces the answers and returns results sets similar to JDBC (JavaTM Data Base Connectivity) (Sun 2008). Despite the fact that an OMS may not completely solve the main problems involved in deployment, they are step in the direction of a more integrated environment, which addresses more aspects of the

Ontology life cycle than previously described tools. Other examples of OMSs include SymOntoX (Symbolic OMS) and knOWLer.

II.3.3 Annotation Tools

On the SW not only documents can be annotated, but also any resource identified by a URI. That means agents, places, concepts, etc. In order to perform this annotation of the resources, a number of tools are again necessary. The tools can be regrouped under the generic term of annotation tools. A review of annotation tools for KM in general is proposed by (Uren V. 2005). Seven key points are identified:

- The editor must provide standard format for input and output.
- It must be integrated within the environment where users manipulate the documents, and in the global collaborative process.
- It should support multiple Ontology references.
- Multiple document formats must be supported.
- Evolution of the document must be taken into account.
- Annotation can be stored either in the document itself or in a separate container.
- Facilities for automation have to be provided.

Some of these points are still hot research topics, but they give a good overview of the complexity of the task. For a list of exiting annotation tools, please consult (Uren V. 2005).

II.3.4 Visualization Tools

Ontologies are commonly represented by graphs structures. Tools like IsaViz or plugins for Protégé, allow visualizing the graph nature of RDF or OWL. However, RDF does not specify physical coordinates for displaying the nodes and arcs is encoded. Tools must rely on automatic algorithms for the placement of nodes and arcs on a two dimension level. As explained in the concept map theory (Novak J. D. 2006), the physical location may carry some semantics that is not encoded in RDF.

Another difficulty raised by RDF is the definition of the RDFS language in RDF itself. When flattened on a graph view it might be difficult to separate the different levels of

modeling. In the case of Ontology editors like Protégé, the class hierarchy is clearly separated from the instances. It facilitates comprehension. Globally specific presentation patterns must be adapted to efficiently visualized RDF graphs. This is illustrated by the “stylesheet” approach of Isaviz (IsaViz 2008).

II.3.5 Ontology Search Engines

One of the main functionalities of Web applications for KM is information retrieval (IR). IR on the SW takes advantage of Ontologies and annotations to allow users to ask Ontology-based queries. Semantic search engines are necessary to perform this task. Those search engines take Ontologies in RDFS or OWL and annotations in RDF to answer queries expressed in query languages like RQL, SPARQL, etc. Semantic search engines are not standalone applications but provide the semantic middleware that needs to be connected with dedicated interfaces. Bellows are some existing platforms:

- Ontobroker (Ontobroker 2008), which is now a commercial product, and KAON2 (KAON2 and Motic 2008) based on DL reasoning.
- The Corese semantic search engine (Corby O. 2004), internally based on conceptual graphs. It has been embedded in a complete Web platform called SEWESE (Semantic Web SErver).
- Other tools that include the Jena framework, developed by HP (Jena 2008), Sesame (Sesame 2008), Triple (Triple 2008).

II.3.6 Semantic Browser

Another way to exploit formalized knowledge and annotated documents is to follow the analogy with the actual Web and offer browsing of the available knowledge. This is called “conceptual browsing”, and finds a direct expression in the context of the SW. A number of “conceptual browsers” with different navigation paradigms supports the idea.

“Conceptual navigation” as presented by (Naeve A. 2001) may consist in visualizing graphs representing the conceptual knowledge and accessing related documents. An example of conceptual browser for conceptual navigation is Conzilla browser (Conzilla 2008). In the Conzilla browser, users visualize what is called a “knowledge manifold”.

Another type of browser is proposed by the Magpie plug-in (Dzbor M. 2005; KMI 2005). Instead of visualizing a conceptual structure and the associated annotations, the browser relies on classical hypertext navigation, enhanced by automatic annotation. Terms on a Web page identify concepts from a selected Ontology. By right clicking on the term a list of available services relative to this concept are proposed. Navigation is then driven by the concepts found in the pages and not only by hyperlinks.

The tools and associated use presented above are generic. Most of these ideas can be very interesting in the context of e-Learning. For example, as experimented with earlier “semantic hypertexts” (Murray 2003), “conceptual navigation” may help learners in exploiting a course document. In the next part, the actual state of the research on SW that actually focuses on learning is presented.

APPENDIX III

KNOWLEDGE MANAGEMENT

Many management researchers have stressed the importance of knowledge. They argued that knowledge is not merely like another resource such as labor and capital, but it is considered the most important resource today. Knowledge is the source of the highest-quality power and is the key to the power shift that lies ahead. That is why how well an organization managing the knowledge that contributes to their core competencies is a key competitive factor for the organization in this knowledge economy era (where service and expertise are the main business outcome). This section shall provide some definitions, fundamental aspects, tool and techniques underlying KM.

III.1 Fundamental of KM

KM is still a young field with multidisciplinary roots. Thus, there are many definitions and still no fixed definition on it. There is considerable variety in the perspectives taken and there is no consensus yet what exactly KM is all about and how to proceed. Bellows are some definition in what KM is:

1. A wide range of practices aimed at capturing, organizing and storing the knowledge and experiences of individuals and groups within an organization and making it available to others in the organization. (Brandon Hall, 2003)
2. The process of identifying, capturing, organizing and disseminating the intellectual assets those are critical to the organization's long-term performance. (Shelda Debowski, 2006)
3. The ability to selectively capture, archive and access the best practices of work-related knowledge and decision making from employees and managers for both individual and group behaviors (Bergeron and Bryan 2003).
4. A deliberate and systematic business optimization strategy that selects, distills, stores, organizes, packages and communicates information essential to the

business of a company in a manner that improves employee performance and corporate competitiveness (Bergeron and Bryan 2003).

5. Management of organizational knowledge for creating business value and generating a competitive advantage. KM enables the creation, communication and application of knowledge of all kinds to achieve business goals (Tiwana 1999).
6. KM is defined as the management function responsible for the regular selection, implementation and evaluation of goal-oriented knowledge strategies that aim at improving an organization's way of handling knowledge either internal or external to the organization in order to improve organizational performance. The implementation of knowledge strategies comprises all person-oriented, organizational and technological instruments. The knowledge strategies are supposed to be suitable to dynamically optimize the organization-wide level of competencies, education and ability to learn of the members of the organization as well as to develop collective intelligence (Maier 2007).

Instead of knowing what KM is, it is essential to know what KM is not about. Amrit Tiwana stated that what KM is not about:

1. *KM is not knowledge engineering.* Knowledge engineering has been a vital part of computer science but is barely even related to KM. KM is a business problem and falls in the domain of information systems and management, not in computer science. KM needs to meld information systems and people in ways that knowledge engineering has never been able to.
2. *KM is about process, not just digital networks.* Management of knowledge has to encompass and improve business processes.
3. *KM is not about building a "smarter" intranet.* A KM system can use your company's intranet as its front end, but one should never be mistaken for the other. Saying that your intranet is your KM system is something as senseless as saying that a jetliner is the cockpit.
4. *KM is not about a one-time investment.* KM, like any other future-oriented investment, requires consistent attention over a substantial period even after it begins to deliver results.

5. *KM is not about enterprise-wide "infobahns"*. While enterprise integration helps the primary focus of KM is on creating, getting, importing, delivering, and most importantly helping the *right* people apply the *right* knowledge at the *right* time. KM solutions must, therefore, reflect the way individuals and organizations have managed and shared information, albeit more effectively.
6. *KM is not about "capture."* Document management vendors would have you believe otherwise, but KM is not about capturing "knowledge." An inevitable loss of context occurs when documents are "sanitized" for use across the company. While a document management system lacks context, experience, and insight, it still has a marginal place in KM technology. Knowledge, in its entirety, cannot be captured.

From those above definition of KM, the KM life cycle can be depicted as Figure III-1. The KM life cycle consist of 8 stages: (i) Knowledge creation or acquisition, (ii) Knowledge modification, (iii) Immediate use, (iv) Archiving, (iv) Transfer, (v) Translation/repurposing, (vi) User access, (vii) Disposal.

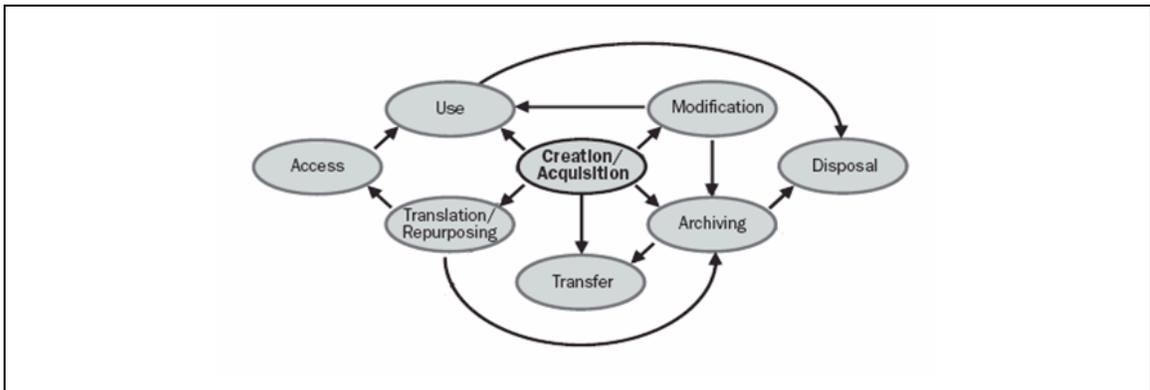


Figure III- 1 KM Life Cycle (Bergeron and Bryan 2003)

As illustrated in Figure III-1 each phase of KM life cycle can be explained as bellows:

In the creation and acquisition phase of the KM life cycle, information is authored internally by knowledge workers, acquired through outsourcing, or purchased from an outside source. Then in the modification phase, the information is modified to suit the

immediate and likely future needs of knowledge workers and management. Next phase is use, where the information is employed for some useful purpose. The range of potential uses for information is virtually unlimited, and depends on the industry and the needs and activities of knowledge workers within the organization.

In the archiving phase of the KM life cycle, information are stored in a form and format that will survive the elements and time and still be accessible and usable by knowledge workers in the organization. Then, the transfer or communications of information from one person or place to another is a prerequisite for an efficient KM System. In the translation/repurposing phase, information is translated from its original form into a form more suitable for a new purpose. A characteristic of most KM Systems is information hiding, in that not all information in the corporation is openly available to everyone. The ultimate destiny of information, whether from the scourges of time, short-term benign neglect, or intentional disposal, is destruction.

In practice, most KM practices fall short of this ideal. This is primarily because it's virtually impossible to capture the thoughts, beliefs and behaviors of a manager or employee in a way that is both economical and complete enough to provide another person—or machine—with enough quality information to make the same decisions, exhibit the same leadership principles, or perform the same complex tasks at the same level of performance.

There is also confusion caused by terminology borrowed from the academic community regarding the use of knowledge in artificial intelligence research, much of which does not apply to KM.

Bellows are the essential key concepts in KM stressed by knowledge researcher/practitioner (Bergeron and Bryan 2003) :

1. *Leadership is essential.* Someone in senior management has to own the KM effort. This manager is often termed the chief knowledge officer (CKO) if the task is all-encompassing, or the chief information officer (CIO) or other senior

manager may take it on as an additional responsibility. Regardless of who takes the role, it involves achieving buy-in at all levels in the organization.

2. *KM works.* The potential benefits of KM are numerous and can potentially benefit every type of business, especially those involved in the information technology and service industries. KM promises reduced costs, improved service, increased efficiencies, and retention of intellectual assets.
3. *KM requires training.* Employee and manager education is fundamental to the proper operation of every phase of the KM process. As the story illustrates, employees and managers have to be trained to focus on the overall process even while they are attending to specific problems.
4. *Expectations must be managed.* Implementing a KM program involves fundamental changes in how employees and managers interact, communicate, command and get things done. Before reporting lines, responsibilities and management directives shift to meet the KM demands of the corporation, employees and managers must be prepared for the change. However, since most people fear change, especially if it means disrupting a way of life that they have grown accustomed to, productivity can suffer unless employee expectations are managed proactively.
5. *Practical KM is technology dependent.* Each of the steps in the KM process, as well as tracking knowledge assets, can be enhanced by information technologies. For example, the process of information creation is supported by the ubiquitous word processor running on a PC, and painless acquisition is made possible by the Web and associated networking hardware. Similarly, database servers make storing and manipulating huge stores of data possible and software, and getting data in the hands of users benefits from handheld devices and wireless networks that provide anytime, anywhere access to information.
6. *KM is a process, not a product.* KM is a dynamic, constantly evolving process and not a shrink-wrapped product. Knowledge is an organizational process rather than a static collection of data that can be stored in a database. Typical KM practices in a modern corporation include acquiring knowledge from customers, creating new revenues from existing knowledge, capturing an employee's

knowledge for reuse later, and reviewing the predictors of a successful KM initiative.

7. *KM Components*. KM is composed of three interconnected and interdependent components: information repositories, communities and networks, and experts and expertise.

III.2 KM Tools and Techniques

This section profiles the following key sets of KM tools, as described in action in organizations around the world: content management, taxonomies, groupware, online communities of practice, portals, social network analysis, e-learning, storytelling, wireless platforms, innovation management tools and inter-organizational knowledge sharing platforms which is drawn from first-hand interviews with KM practitioners in the Americas, Europe, Asia, and Australia; consulting assignments for KM initiatives; and participation and reports from over two dozen KM conferences and workshops around the world in 2002 and 2003 by (Rao 2005).

III.2.1 Content Management System (CMS)

A growing number of companies today have instituted content management system for best practice, lesson learned, product development knowledge, customer knowledge, human resource management knowledge and methods-based knowledge.

III.2.2 Knowledge Taxonomies

Automated tools like computer-generated taxonomies can also assist in KM practices, especially in conjunction with human inputs and refinements. The info-glut or “digital sprawl” on corporate intranets has led to users not being able to find relevant information in time and numerous taxonomy development tools are coming to the rescue.

III.2.3 Groupware

Desirable features for collaborative tools in the context of KM include affinity building, knowledge mapping, threading, polling, group document creation, rating, anonymity, notification and access management.

III.2.4 Online Communities of Practice (CoP)

Online communities constitute a growing part of the organizational landscape of 21st century global players, but businesses are still at the early stages of individual and organizational optimization of Web-based communities.

III.2.5 Portals

The enterprise portal is becoming the IT platform of choice for the interlinked workspace. Context and aggregation need to be built around the role of each knowledge worker. Further personalization should be made possible via customization features.

III.2.6 Social Network Analysis (SNA) and Design

Within organizational settings, SNA is emerging as a powerful tool for mapping knowledge flows and identifying gaps. SNA can be used to reinforce existing flows and to improve knowledge integration after activities like mergers and acquisitions. Methods used can be qualitative (e.g., employee surveys) or quantitative (e.g., analysis of transactions like e-mails or phone calls or information artifacts like documents and search strings).

III.2.7 e-Learning

One interesting emerging development on the KM front is the growing convergence of viewpoints between the KM community and the e-learning community.

III.2.8 Storytelling and Narratives

Building communities in business has become a priority in knowledge-sharing organizations around the world. Storytelling has become an effective way to mobilize change, according to Seth Kahan, President of the Performance Development Group and internal communications consultant to the World Bank.

III.2.9 Wireless Platform for Knowledge Mobilization

One of the most notable emerging trends in workforce connectivity is the increasing use of mobile technologies to take “KM” to another dimension “knowledge mobilization”—by bringing relevant knowledge directly to the fingertips of a company’s road warriors and field-workers via cell phones, PDAs, industry-specific handheld devices, Wireless Local Area Network (WLAN), and Radio Frequency Identification (RFID) tags.

III.2.10 Innovation Management Tools

Online idea management systems have been deployed at companies like Bristol Myers Squibb, Cadbury-Schweppes and Mott’s Apples. Managing an innovation pipeline, promoting an “idea central” or ideas marketplace, and creating the “hundred headed brain” are some creative approaches being adopted by KM pioneers, according to Mark Turrell, CEO of Imaginatik Research.

III.2.11 Inter-Organizational Knowledge Sharing Platforms

Online services such as dial-up bulletin boards and Web communities have actually helped network communities of interest across the globe for years. The World Bank has leveraged a strategy of “global knowledge, local adaptation” for brokering global knowledge exchanges. This includes Web-based resources like case studies, Web casts and video clips from experts, world development indicators, knowledge toolkits (on topics like business climate and corruption), distance learning services, knowledge assessment methodology, and forums like development gateway.

III.3 Knowledge Management System (KMS)

A KMS is an application that collects stores and makes information available among individuals in an organization. This system's primary purpose is to capture a company's collective knowledge and then make it simple to retrieve and reuse. A KMS can help companies avoid reinventing the wheel. It can also enhance the exchange and dissemination of understanding within an enterprise and can increase the level of collaboration between employees (Clarey 2007).

KMS is an ICT system in the sense of an application system or an ICT platform that combines and integrates functions for the contextualized handling of both explicit and tacit knowledge, throughout the organization or that part of the organization that is targeted by a KM initiative. A KMS offers integrated services to deploy KM instruments for networks of participants, i.e. active knowledge workers, in knowledge-intensive business processes along the entire knowledge life cycle. Ultimate aim of KMS is to support the dynamics of organizational learning and organizational effectiveness (Maier 2007).

Goals of using KMS are for example to generate, share and apply knowledge, to locate experts and networks, to actively participate in networks and communities, to create and exchange knowledge in these networks, to augment the employees' ability to learn and to understand relationships between knowledge, people and processes (Maier 2007).

Figure III-2 depicts the characteristic of KMS. The characteristics of KMS can be used as requirements in order to judge whether an actual system is a KMS or not.

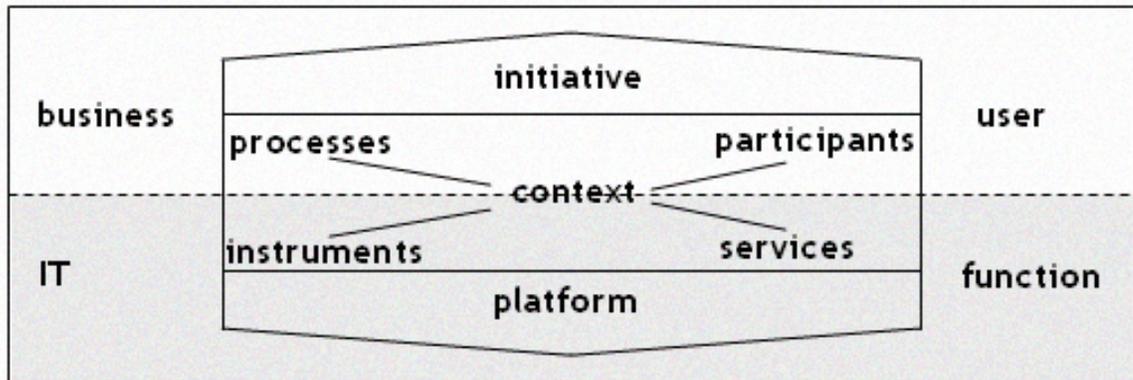


Figure III- 2 Characteristic of KMS (Ronald Maier, 2007)

KMS has to be aligned (1) with the business environment, i.e. the knowledge intensive business processes that are affected, (2) the user environment with the expectation of a rich user experience and personalized on-demand KMS services, (3) the IT infrastructure environment, which determines the technical base, and (4) the function environment that determines the service interfaces for KMS design.

KM Systems offer some benefits in which IT is applied in KM for several important reasons (Gottschalk 2005):

1. IT is an enabler of improved individual performance among knowledge workers.
2. IT is an enabler of improved organizational performance by new business processes.
3. IT is an enabler of improved inter-organizational performance by effective knowledge networks.

In 2001 a survey research has been conducted to know the use of software concerning with KM systems. Responding companies ranked software based on dollar amount to be spent as presented in Figure III-3 (CIO 2001) :

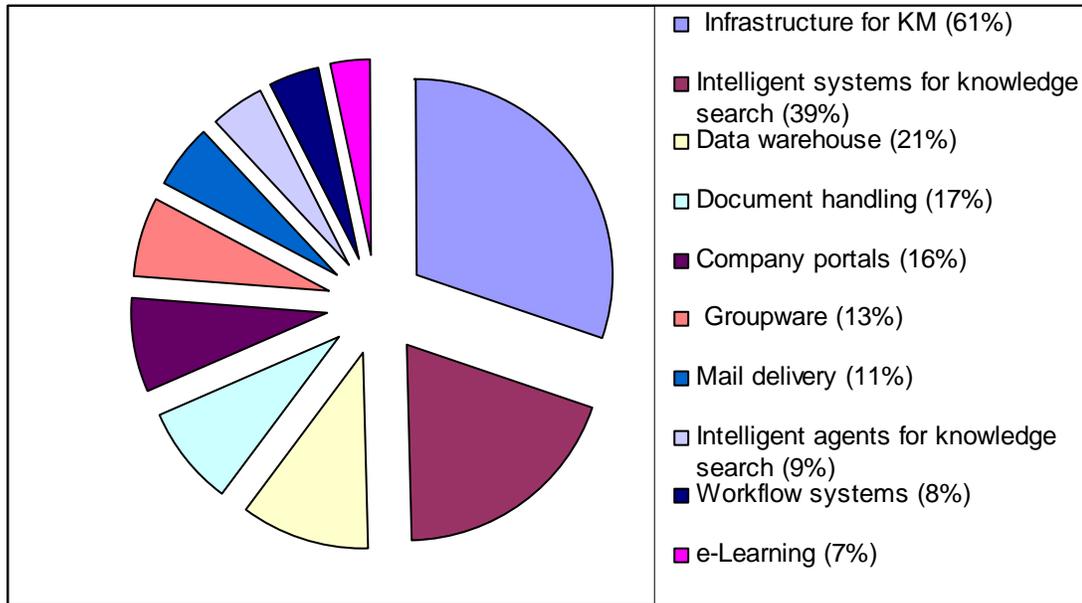


Figure III- 3 Software Use for KMS in Companies

APPENDIX IV DOMAIN ONTOLOGY

This appendix presents the domain Ontology in graphical view and Ontology Representation Language (OWL: Web Ontology language. Section IV.1 presents the domain Ontology in Hierarchy diagram, Section IV.2 presents the domain Ontology in Concepts Relationship Diagram, and Section IV.3 presents the domain Ontology in OWL.

IV.1 Domain Ontology in Class Tree



Figure IV- 1 Graphical Domain Ontology Representation in Class Tree Diagram

IV.2 Domain Ontology in Domain/Range Relation

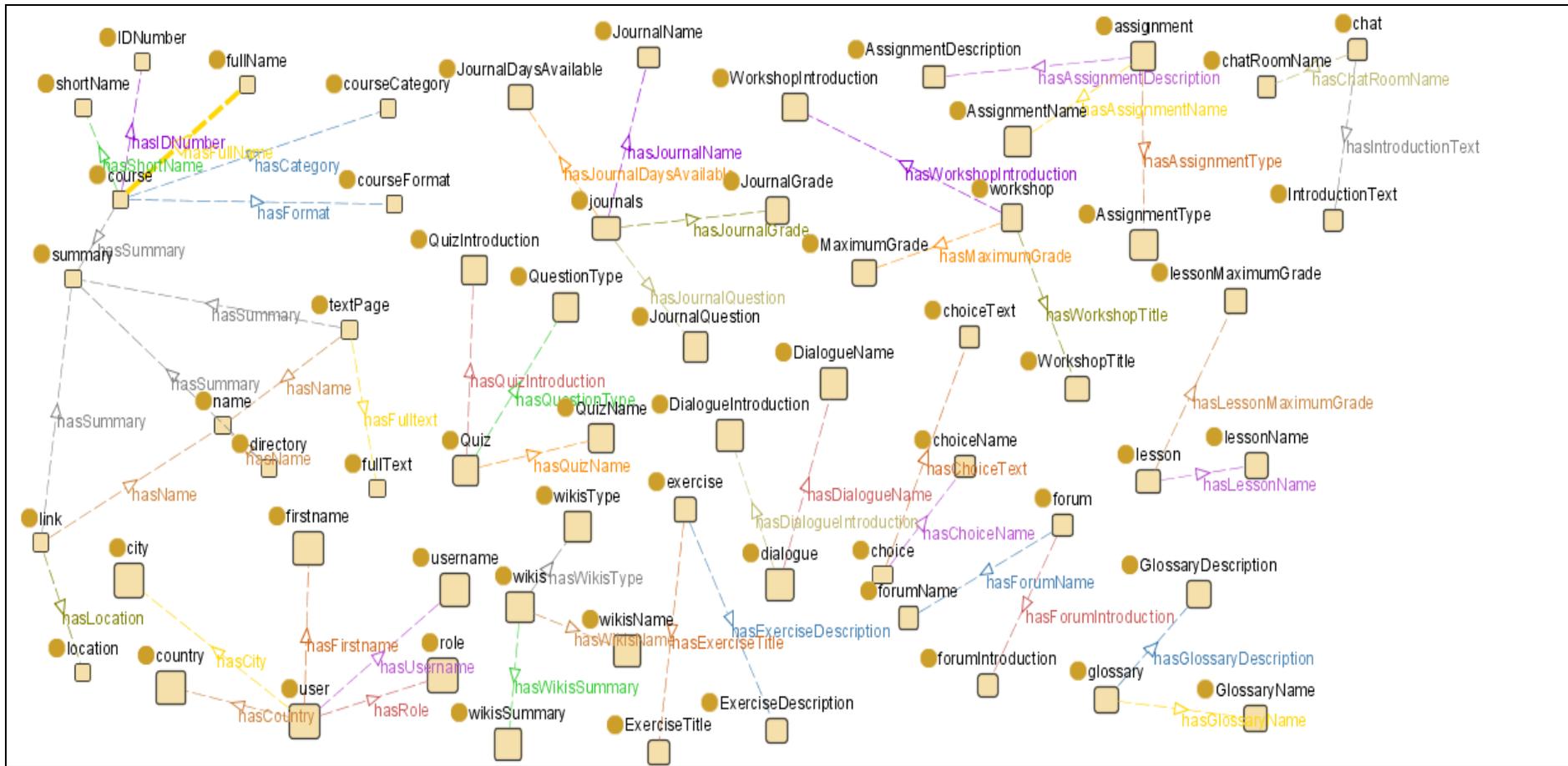


Figure IV- 2 Graphical Representation Domain Ontology in Domain/Range Relation

IV.3 Domain Ontology in OWL (Web Ontology Language)

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY OntoMoodle_v2 "http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#" >
] >

<rdf:RDF xmlns="http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#"
  xml:base="http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:OntoMoodle_v2="http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#" >
  <owl:Ontology rdf:about=""/>

  <!--
  ////////////////////////////////////////////////////////////////////
  //
  // Object Properties
  //
  ////////////////////////////////////////////////////////////////////
  -->

  <!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#hasCountry -->

  <owl:ObjectProperty rdf:about="#hasCountry">
    <rdf:type rdf:resource="#owl:FunctionalProperty"/>
    <rdfs:range rdf:resource="#Country"/>
  </owl:ObjectProperty>
</rdf:RDF>
```

```
<rdfs:domain rdf:resource="#User" />
</owl:ObjectProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#hasCourseFormat -->
<owl:ObjectProperty rdf:about="#hasCourseFormat">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
  <rdfs:domain rdf:resource="#Course" />
  <rdfs:range rdf:resource="#CourseFormat" />
</owl:ObjectProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#hasDialogueType -->
<owl:ObjectProperty rdf:about="#hasDialogueType">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
  <rdfs:domain rdf:resource="#Dialogue" />
  <rdfs:range rdf:resource="#DialogueType" />
</owl:ObjectProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#hasForumType -->
<owl:ObjectProperty rdf:about="#hasForumType">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
  <rdfs:domain rdf:resource="#Forum" />
  <rdfs:range rdf:resource="#ForumType" />
</owl:ObjectProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#hasGlossaryType -->
<owl:ObjectProperty rdf:about="#hasGlossaryType">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
  <rdfs:domain rdf:resource="#Glossary" />
  <rdfs:range rdf:resource="#GlossaryType" />
</owl:ObjectProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#hasLanguage -->
```

```

<owl:ObjectProperty rdf:about="#hasLanguage">
  <rdf:type rdf:resource="#owl:FunctionalProperty" />
  <rdfs:domain rdf:resource="#Course" />
  <rdfs:range rdf:resource="#Language" />
</owl:ObjectProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#hasSurveyType -->

<owl:ObjectProperty rdf:about="#hasSurveyType">
  <rdf:type rdf:resource="#owl:FunctionalProperty" />
  <rdfs:domain rdf:resource="#Survey" />
  <rdfs:range rdf:resource="#SurveyType" />
</owl:ObjectProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#hasVisibility -->

<owl:ObjectProperty rdf:about="#hasVisibility">
  <rdf:type rdf:resource="#owl:FunctionalProperty" />
  <rdfs:domain rdf:resource="#Activity" />
  <rdfs:domain rdf:resource="#Resource" />
  <rdfs:range rdf:resource="#Visibility" />
</owl:ObjectProperty>

<!--
////////////////////////////////////
//
// Data properties
//
////////////////////////////////////
-->

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#assignmentDescription -->

<owl:DatatypeProperty rdf:about="#assignmentDescription">
  <rdf:type rdf:resource="#owl:FunctionalProperty" />
  <rdfs:domain rdf:resource="#Assignment" />
  <rdfs:range rdf:resource="#xsd:string" />
</owl:DatatypeProperty>

```

```
<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#assignmentName -->

<owl:DatatypeProperty rdf:about="#assignmentName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Assignment"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#availableFrom -->

<owl:DatatypeProperty rdf:about="#availableFrom">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Assignment"/>
  <rdfs:range rdf:resource="&xsd:dateTime"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#chatIntroduction -->

<owl:DatatypeProperty rdf:about="#chatIntroduction">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Chat"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#chatRoomName -->

<owl:DatatypeProperty rdf:about="#chatRoomName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Chat"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#choiceName -->

<owl:DatatypeProperty rdf:about="#choiceName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
```

```

    <rdfs:domain rdf:resource="#Choice" />
    <rdfs:range rdf:resource="&xsd:string" />
  </owl:DatatypeProperty>

  <!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#choiceText -->

  <owl:DatatypeProperty rdf:about="#choiceText">
    <rdf:type rdf:resource="&owl;FunctionalProperty" />
    <rdfs:domain rdf:resource="#Choice" />
    <rdfs:range rdf:resource="&xsd:string" />
  </owl:DatatypeProperty>

  <!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#city -->

  <owl:DatatypeProperty rdf:about="#city">
    <rdf:type rdf:resource="&owl;FunctionalProperty" />
    <rdfs:domain rdf:resource="#User" />
    <owl:equivalentProperty rdf:resource="#town" />
    <rdfs:range rdf:resource="&xsd:string" />
  </owl:DatatypeProperty>

  <!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#courseFullName -->

  <owl:DatatypeProperty rdf:about="#courseFullName">
    <rdf:type rdf:resource="&owl;FunctionalProperty" />
    <rdfs:domain rdf:resource="#Course" />
    <rdfs:range rdf:resource="&xsd:string" />
  </owl:DatatypeProperty>

  <!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#courseIDNumber -->

  <owl:DatatypeProperty rdf:about="#courseIDNumber">
    <rdf:type rdf:resource="&owl;FunctionalProperty" />
    <rdfs:domain rdf:resource="#Course" />
    <rdfs:range rdf:resource="&xsd;ID" />
  </owl:DatatypeProperty>

```

```

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#courseShortName -->

<owl:DatatypeProperty rdf:about="#courseShortName">
  <rdf:type rdf:resource="#owl:FunctionalProperty" />
  <rdfs:domain rdf:resource="#Course" />
  <rdfs:range rdf:resource="#xsd:string" />
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#courseStartDate -->

<owl:DatatypeProperty rdf:about="#courseStartDate">
  <rdf:type rdf:resource="#owl:FunctionalProperty" />
  <rdfs:domain rdf:resource="#Course" />
  <rdfs:range rdf:resource="#xsd:date" />
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#courseSummary -->

<owl:DatatypeProperty rdf:about="#courseSummary">
  <rdf:type rdf:resource="#owl:FunctionalProperty" />
  <rdfs:domain rdf:resource="#Course" />
  <rdfs:range rdf:resource="#xsd:string" />
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#customIntro -->

<owl:DatatypeProperty rdf:about="#customIntro">
  <rdf:type rdf:resource="#owl:FunctionalProperty" />
  <rdfs:domain rdf:resource="#Survey" />
  <rdfs:range rdf:resource="#xsd:string" />
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#databaseIntroduction -->

<owl:DatatypeProperty rdf:about="#databaseIntroduction">
  <rdf:type rdf:resource="#owl:FunctionalProperty" />
  <rdfs:domain rdf:resource="#Database" />
  <rdfs:range rdf:resource="#xsd:string" />
</owl:DatatypeProperty>

```

```

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#databaseName -->

<owl:DatatypeProperty rdf:about="#databaseName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Database"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#dialogueIntroduction -->

<owl:DatatypeProperty rdf:about="#dialogueIntroduction">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Dialogue"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#dialogueName -->

<owl:DatatypeProperty rdf:about="#dialogueName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Dialogue"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#dueDate -->

<owl:DatatypeProperty rdf:about="#dueDate">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Assignment"/>
  <rdfs:range rdf:resource="&xsd:dateTime"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#emailaddress -->

<owl:DatatypeProperty rdf:about="#emailaddress">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>

```

```

    <rdfs:domain rdf:resource="#User"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#everyoneCanViewPastChatSession -->

<owl:DatatypeProperty rdf:about="#everyoneCanViewPastChatSession">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Chat"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#firstname -->

<owl:DatatypeProperty rdf:about="#firstname">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#User"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#forumIntroduction -->

<owl:DatatypeProperty rdf:about="#forumIntroduction">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Forum"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#forumName -->

<owl:DatatypeProperty rdf:about="#forumName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Forum"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#fullText -->

```

```

<owl:DatatypeProperty rdf:about="#fullText">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#TextPage"/>
  <rdfs:domain rdf:resource="#WebPage"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#glossaryDescription -->

<owl:DatatypeProperty rdf:about="#glossaryDescription">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Glossary"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#glossaryName -->

<owl:DatatypeProperty rdf:about="#glossaryName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Glossary"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#grade -->

<owl:DatatypeProperty rdf:about="#grade">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Assignment"/>
  <rdfs:range rdf:resource="&xsd:integer"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#journalName -->

<owl:DatatypeProperty rdf:about="#journalName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Journal"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```
<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#journalQuestion -->

<owl:DatatypeProperty rdf:about="#journalQuestion">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
  <rdfs:domain rdf:resource="#Journal" />
  <rdfs:range rdf:resource="&xsd:string" />
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#labelText -->

<owl:DatatypeProperty rdf:about="#labelText">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
  <rdfs:domain rdf:resource="#Label" />
  <rdfs:range rdf:resource="&xsd:string" />
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#lessonName -->

<owl:DatatypeProperty rdf:about="#lessonName">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
  <rdfs:domain rdf:resource="#Lesson" />
  <rdfs:range rdf:resource="&xsd:string" />
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#location -->

<owl:DatatypeProperty rdf:about="#location">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
  <rdfs:domain rdf:resource="#Link" />
  <rdfs:range rdf:resource="&xsd:anyURI" />
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#maxNumberOfAnswer -->

<owl:DatatypeProperty rdf:about="#maxNumberOfAnswer">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
```

```

    <rdfs:domain rdf:resource="#Lesson" />
    <rdfs:range rdf:resource="&xsd;integer" />
  </owl:DatatypeProperty>

  <!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#preventLateSubmission -->

  <owl:DatatypeProperty rdf:about="#preventLateSubmission">
    <rdf:type rdf:resource="&owl;FunctionalProperty" />
    <rdfs:domain rdf:resource="#AdvancedUploadingOfFile" />
    <rdfs:domain rdf:resource="#Assignment" />
    <rdfs:range rdf:resource="&xsd;boolean" />
  </owl:DatatypeProperty>

  <!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#quizClose -->

  <owl:DatatypeProperty rdf:about="#quizClose">
    <rdf:type rdf:resource="&owl;FunctionalProperty" />
    <rdfs:domain rdf:resource="#Quiz" />
    <rdfs:range rdf:resource="&xsd;dateTime" />
  </owl:DatatypeProperty>

  <!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#quizIntroduction -->

  <owl:DatatypeProperty rdf:about="#quizIntroduction">
    <rdf:type rdf:resource="&owl;FunctionalProperty" />
    <rdfs:domain rdf:resource="#Quiz" />
    <rdfs:range rdf:resource="&xsd;string" />
  </owl:DatatypeProperty>

  <!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#quizName -->

  <owl:DatatypeProperty rdf:about="#quizName">N
    <rdf:type rdf:resource="&owl;FunctionalProperty" />
    <rdfs:domain rdf:resource="#Quiz" />
    <rdfs:range rdf:resource="&xsd;string" />
  </owl:DatatypeProperty>

```

```

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#quizOpen -->

<owl:DatatypeProperty rdf:about="#quizOpen">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Quiz"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#resourceName -->

<owl:DatatypeProperty rdf:about="#resourceName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Resource"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#resourceSummary -->

<owl:DatatypeProperty rdf:about="#resourceSummary">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Resource"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#submissionTitle -->

<owl:DatatypeProperty rdf:about="#submissionTitle">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Workshop"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#surname -->

<owl:DatatypeProperty rdf:about="#surname">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#User"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#surveyName -->

<owl:DatatypeProperty rdf:about="#surveyName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Survey"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#timeLimit -->

<owl:DatatypeProperty rdf:about="#timeLimit">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Lesson"/>
  <rdfs:range rdf:resource="&xsd:integer"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#timed -->

<owl:DatatypeProperty rdf:about="#timed">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Lesson"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#town -->

<owl:DatatypeProperty rdf:about="#town">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#User"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#username -->

<owl:DatatypeProperty rdf:about="#username">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>

```

```

    <rdfs:domain rdf:resource="#User"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#wikiName -->
<owl:DatatypeProperty rdf:about="#wikiName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Wiki"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#wikiSummary -->
<owl:DatatypeProperty rdf:about="#wikiSummary">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Wiki"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#workshopDescription -->
<owl:DatatypeProperty rdf:about="#workshopDescription">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Workshop"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!--
////////////////////////////////////
//
// Classes
//
////////////////////////////////////
-->

```

```

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Activity -->

<owl:Class rdf:about="#Activity">
  <rdfs:subClassOf rdf:resource="#owl:Thing"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#AdvancedUploadingOfFile -->

<owl:Class rdf:about="#AdvancedUploadingOfFile">
  <rdfs:subClassOf rdf:resource="#Assignment"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Assignment -->

<owl:Class rdf:about="#Assignment">
  <rdfs:subClassOf rdf:resource="#Activity"/>
  <rdfs:comment
    >A basic task with which you can describe what you want the students to do or record a grade. You can also have the
students upload a response and score it later.</rdfs:comment>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Chat -->

<owl:Class rdf:about="#Chat">
  <rdfs:subClassOf rdf:resource="#Activity"/>
  <rdfs:comment
    >A group chat room where people can meet at the same time and send text messages.</rdfs:comment>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Choice -->

<owl:Class rdf:about="#Choice">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Country -->

```

```
<owl:Class rdf:about="#Country">
  <rdfs:subClassOf rdf:resource="#owl:Thing"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Course -->
<owl:Class rdf:about="#Course">
  <rdfs:subClassOf rdf:resource="#owl:Thing"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#CourseFormat -->
<owl:Class rdf:about="#CourseFormat">
  <rdfs:subClassOf rdf:resource="#owl:Thing"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Database -->
<owl:Class rdf:about="#Database">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Dialogue -->
<owl:Class rdf:about="#Dialogue">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#DialogueType -->
<owl:Class rdf:about="#DialogueType">
  <rdfs:subClassOf rdf:resource="#owl:Thing"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Directory -->
```

```
<owl:Class rdf:about="#Directory">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Forum -->

<owl:Class rdf:about="#Forum">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#ForumType -->

<owl:Class rdf:about="#ForumType">
  <rdfs:subClassOf rdf:resource="#owl:Thing"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Glossary -->

<owl:Class rdf:about="#Glossary">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#GlossaryType -->

<owl:Class rdf:about="#GlossaryType">
  <rdfs:subClassOf rdf:resource="#owl:Thing"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#HotPotatoesQuiz -->

<owl:Class rdf:about="#HotPotatoesQuiz">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>
```

```
<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Journal -->
<owl:Class rdf:about="#Journal">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Label -->
<owl:Class rdf:about="#Label">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Language -->
<owl:Class rdf:about="#Language">
  <rdfs:subClassOf rdf:resource="#owl:Thing"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Lesson -->
<owl:Class rdf:about="#Lesson">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Link -->
<owl:Class rdf:about="#Link">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#OfflineActivity -->
<owl:Class rdf:about="#OfflineActivity">
  <rdfs:subClassOf rdf:resource="#Assignment"/>
</owl:Class>
```

```
<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#OnlineText -->
<owl:Class rdf:about="#OnlineText">
  <rdfs:subClassOf rdf:resource="#Assignment"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Quiz -->
<owl:Class rdf:about="#Quiz">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Resource -->
<owl:Class rdf:about="#Resource">
  <rdfs:subClassOf rdf:resource="#owl:Thing"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Role -->
<owl:Class rdf:about="#Role">
  <rdfs:subClassOf rdf:resource="#owl:Thing"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Survey -->
<owl:Class rdf:about="#Survey">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#SurveyType -->
<owl:Class rdf:about="#SurveyType">
  <rdfs:subClassOf rdf:resource="#owl:Thing"/>
</owl:Class>
```

```
<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#TextPage -->
<owl:Class rdf:about="#TextPage">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#UploadSingleFile -->
<owl:Class rdf:about="#UploadSingleFile">
  <rdfs:subClassOf rdf:resource="#Assignment"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#User -->
<owl:Class rdf:about="#User">
  <rdfs:subClassOf rdf:resource="#&owl;Thing"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Visibility -->
<owl:Class rdf:about="#Visibility">
  <rdfs:subClassOf rdf:resource="#&owl;Thing"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#WebPage -->
<owl:Class rdf:about="#WebPage">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Wiki -->
<owl:Class rdf:about="#Wiki">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>
```

```

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Workshop -->

<owl:Class rdf:about="#Workshop">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#wikiType -->

<owl:Class rdf:about="#wikiType">
  <rdfs:subClassOf rdf:resource="#owl:Thing"/>
</owl:Class>

<!-- http://www.w3.org/2002/07/owl#Thing -->

<owl:Class rdf:about="#owl:Thing"/>

<!--
////////////////////////////////////
//
// Individuals
//
////////////////////////////////////
-->

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#ATTLS -->

<SurveyType rdf:about="#ATTLS"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Administrator -->

<Role rdf:about="#Administrator">
  <rdfs:comment
    >Administrators can usually do anything on the site, in all courses.</rdfs:comment>

```

```
</Role>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Arabic -->
<Language rdf:about="#Arabic"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#COLLES -->
<SurveyType rdf:about="#COLLES"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#CourseCreator -->
<Role rdf:about="#CourseCreator">
  <rdfs:comment
    >Course creators can create new courses and teach in them.</rdfs:comment>
</Role>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#CriticalIncident -->
<SurveyType rdf:about="#CriticalIncident"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#EachPersonPostOneDiscussion -->
<ForumType rdf:about="#EachPersonPostOneDiscussion"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#English -->
<Language rdf:about="#English"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Everybody -->
<DialogueType rdf:about="#Everybody"/>
```

```
<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Groups -->
<wikiType rdf:about="#Groups" />

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Guest -->
<Role rdf:about="#Guest">
  <rdfs:comment
    >Guests have minimal privileges and usually can not enter text anywhere.</rdfs:comment>
</Role>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Hide -->
<Visibility rdf:about="#Hide" />

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#India -->
<Country rdf:about="#India" />

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Indonesia -->
<Country rdf:about="#Indonesia" />

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Indonesian -->
<Language rdf:about="#Indonesian" />

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#LAMS -->
<CourseFormat rdf:about="#LAMS" />
```

```
<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#MainGlossary -->
<GlossaryType rdf:about="#MainGlossary"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Malaysia -->
<Country rdf:about="#Malaysia"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Malaysian -->
<Language rdf:about="#Malaysian"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Myanmar -->
<Country rdf:about="#Myanmar"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#NonEditingTeacher -->
<Role rdf:about="#NonEditingTeacher">
  <rdfs:comment
    >Non-editing teachers can teach in courses and grade students, but may not alter activities.</rdfs:comment>
</Role>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#QandAForum -->
<ForumType rdf:about="#QandAForum"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#SCORM -->
<CourseFormat rdf:about="#SCORM"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#SecondaryGlossary -->
```

```

<GlossaryType rdf:about="#SecondaryGlossary"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Show -->
<Visibility rdf:about="#Show"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#SingleSimpleDiscussion -->
<ForumType rdf:about="#SingleSimpleDiscussion"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Social -->
<CourseFormat rdf:about="#Social">
  <rdfs:comment
    >This format is oriented around one main forum, the Social forum, which appears listed on the main page. It is useful
for situations that are more freeform. They may not even be courses. For example, it could be used as a departmental notice
board.</rdfs:comment>
  </CourseFormat>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#StandardForumForGeneralUse -->
<ForumType rdf:about="#StandardForumForGeneralUse"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Student -->
<Role rdf:about="#Student">
  <rdfs:comment
    >Students generally have less privileges within a course.</rdfs:comment>
  </Role>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#StudentToStudent -->
<DialogueType rdf:about="#StudentToStudent"/>

```

```

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Students -->
<wikiType rdf:about="#Students"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Teacher -->
<Role rdf:about="#Teacher">
  <rdfs:comment
    >Teachers can do anything within a course, including changing the activities and grading students.</rdfs:comment>
</Role>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#TeacherToStudent -->
<DialogueType rdf:about="#TeacherToStudent"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Teachers -->
<wikiType rdf:about="#Teachers"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Thailand -->
<Country rdf:about="#Thailand"/>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Topics -->
<CourseFormat rdf:about="#Topics">
  <rdfs:comment
    >Very similar to the weekly format, except that each &quot;week&quot; is called a topic. A &quot;topic&quot; is not
restricted to any time limit. You don't need to specify any dates.</rdfs:comment>
</CourseFormat>

<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Vietnam -->
<Country rdf:about="#Vietnam"/>

```

```
<!-- http://localhost:9090/moodle/knowledgebase/Ontologies/OntoMoodle_v2.owl#Weekly -->

<CourseFormat rdf:about="#Weekly">
  <rdfs:comment
    >The course is organised week by week, with a clear start date and a finish date. Each week consists of activities. Some
of them, like assignments, may have &quot;open windows&quot; of, say, two weeks after which they become unavailable.</rdfs:comment>
  </CourseFormat>
</rdf:RDF>

<!-- Generated by the OWL API (version 2.2.1.810) http://owlapi.sourceforge.net -->
```

APPENDIX V

APPLICATION ONTOLOGY

This appendix presents the application Ontology in graphical view and Ontology Representation Language (OWL: Web Ontology language). Section V.1 presents the application Ontology in Hierarchy diagram, Section V.2 presents the application Ontology in Concepts Relationship Diagram and Section V.3 presents the application Ontology in OWL.

V.1 Application Ontology in Class Tree

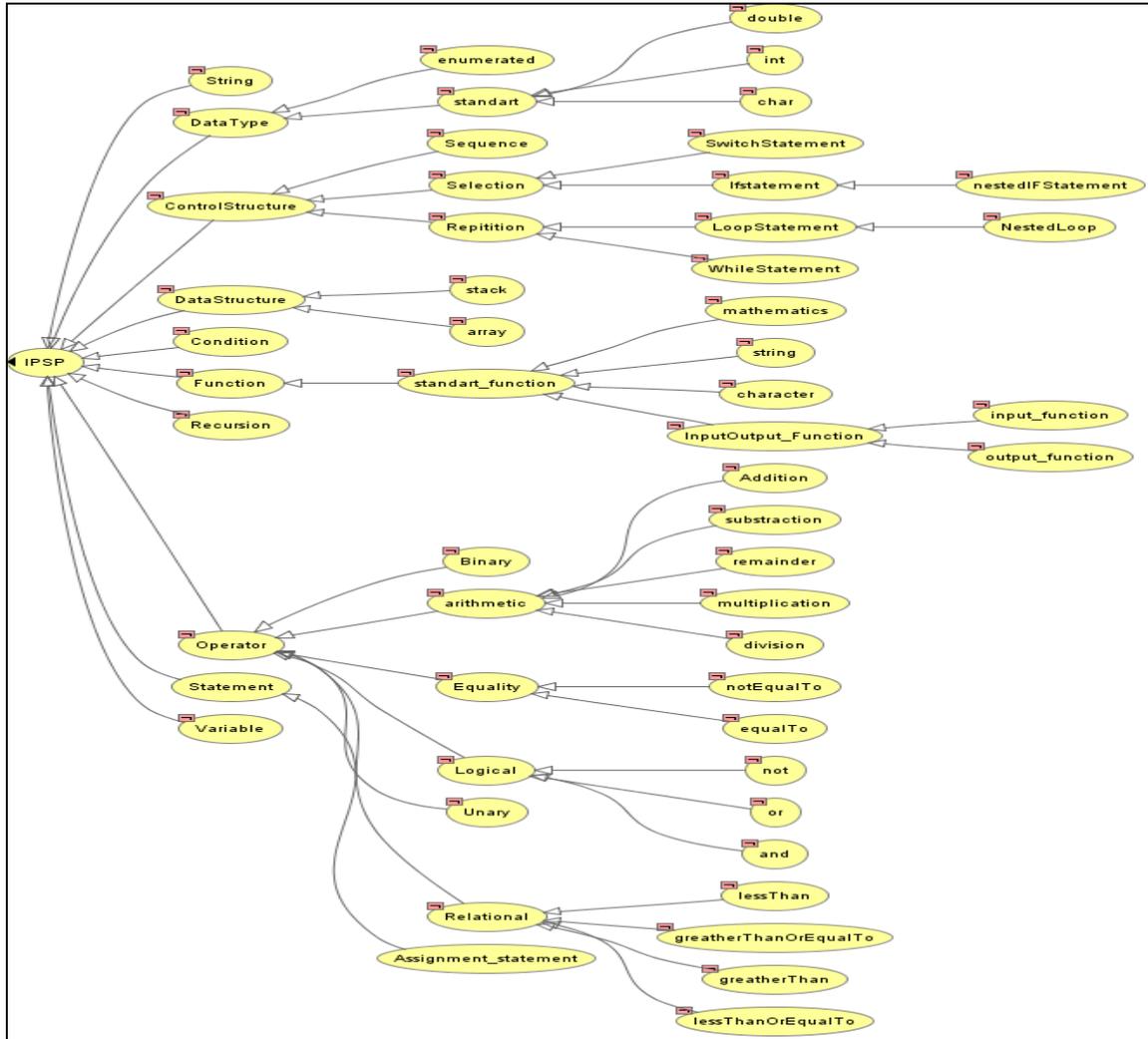


Figure V- 1 Graphical Representation Application Ontology In Class Tree Diagram

V.2 Application Ontology in Domain/Range Relation

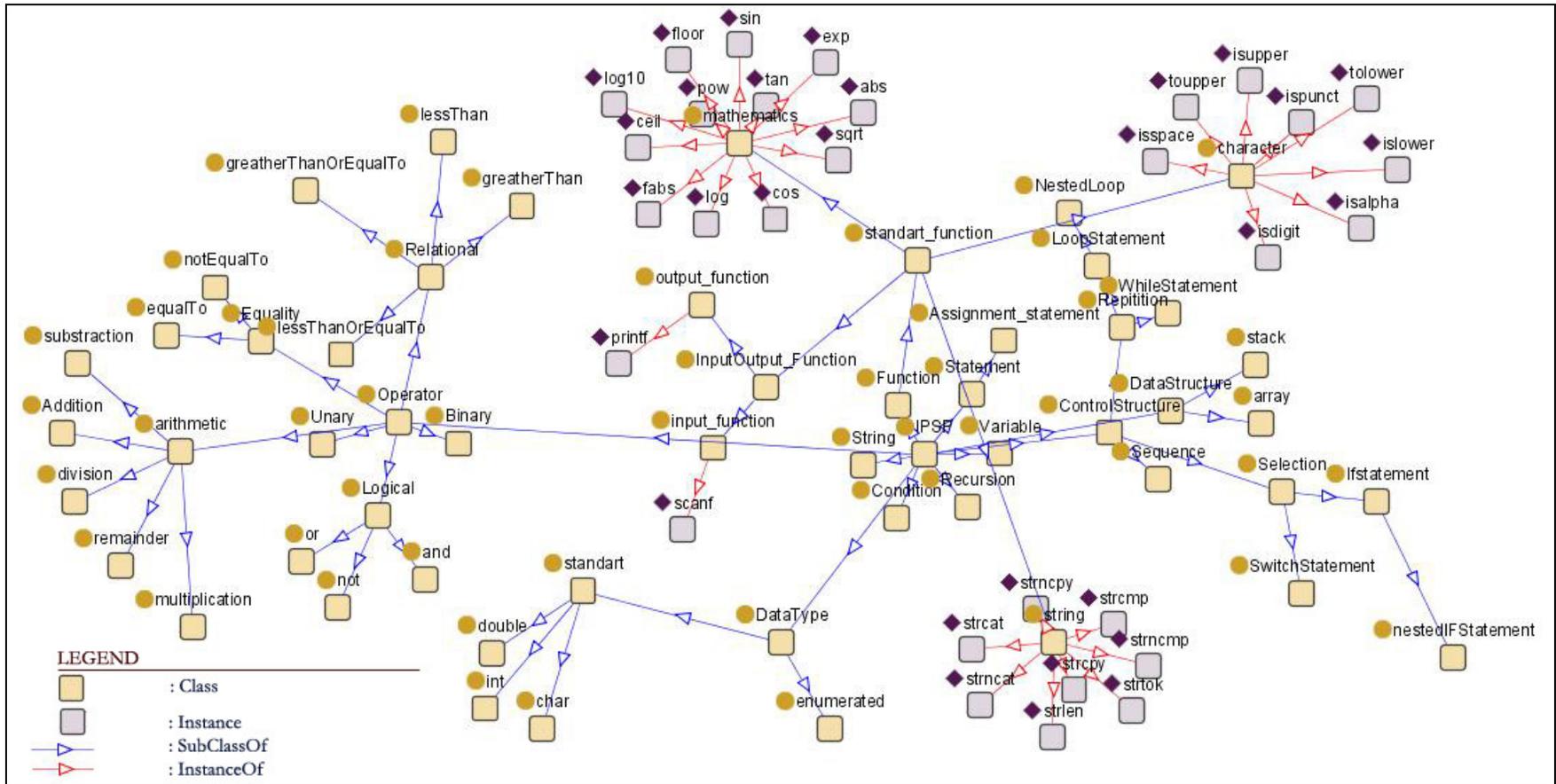


Figure V- 2 Graphical Application Ontology Representation in Domain/Range Relation Diagram

V.3 Application Ontology in OWL (Web Ontology Language)

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns="http://localhost/Ontologybase/OntoMoodle001.owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:pl="http://www.owl-ontologies.com/assert.owl#"
xml:base="http://localhost/Ontologybase/OntoMoodle001.owl">
<owl:Ontology rdf:about="" />
<owl:Class rdf:ID="Variable">
  <owl:disjointWith>
    <owl:Class rdf:ID="DataStructure" />
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="IPSP" />
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="ControlStructure" />
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="Condition" />
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="DataType" />
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="String" />
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="Recursion" />
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="Function" />
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="Statement" />
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="Operator" />
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="Unary">
```

```

<rdfs:subClassOf>
  <owl:Class rdf:about="#Operator" />
</rdfs:subClassOf>
<owl:disjointWith>
  <owl:Class rdf:ID="arithmetic" />
</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:ID="Binary" />
</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:ID="Equality" />
</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:ID="Logical" />
</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:ID="Relational" />
</owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="array">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#DataStructure" />
  </rdfs:subClassOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >a collection of data items of the same type.</rdfs:comment>
  <owl:disjointWith>
    <owl:Class rdf:ID="stack" />
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="InputOutput_Function">
  <owl:disjointWith>
    <owl:Class rdf:ID="character" />
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="mathematics" />
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="string" />
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="standart_function" />
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="notEqualTo">
  <owl:disjointWith>
    <owl:Class rdf:ID="equalTo" />
  </owl:disjointWith>

```

```

<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>!=</rdfs:comment>
<rdfs:subClassOf>
  <owl:Class rdf:about="#Equality"/>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="input_function">
  <rdfs:subClassOf rdf:resource="#InputOutput_Function"/>
  <owl:disjointWith>
    <owl:Class rdf:ID="output_function"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="SwitchStatement">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Selection"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="Ifstatement"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:about="#Operator">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#IPSP"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:about="#Function"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Recursion"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Condition"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#ControlStructure"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#DataStructure"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#Variable"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#Statement"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#DataType"/>
  </owl:disjointWith>
  <owl:disjointWith>

```

```

    <owl:Class rdf:about="#String"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:about="#Logical">
  <rdfs:subClassOf rdf:resource="#Operator"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#arithmetic"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Binary"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Equality"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Relational"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#Unary"/>
</owl:Class>
<owl:Class rdf:about="#equalTo">
  <owl:disjointWith rdf:resource="#notEqualTo"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Equality"/>
  </rdfs:subClassOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >=</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="division">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  ></rdfs:comment>
  <owl:disjointWith>
    <owl:Class rdf:ID="Addition"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="multiplication"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="remainder"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="subtraction"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#arithmetic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#subtraction">

```

```

<owl:disjointWith>
  <owl:Class rdf:about="#Addition"/>
</owl:disjointWith>
<owl:disjointWith rdf:resource="#division"/>
<owl:disjointWith>
  <owl:Class rdf:about="#multiplication"/>
</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:about="#remainder"/>
</owl:disjointWith>
<rdfs:subClassOf>
  <owl:Class rdf:about="#arithmetic"/>
</rdfs:subClassOf>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#remainder">
  <owl:disjointWith>
    <owl:Class rdf:about="#Addition"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#division"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#multiplication"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#substraction"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#arithmetic"/>
  </rdfs:subClassOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >%</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="greatherThanOrEqualTo">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&gt;=</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Relational"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="greatherThan"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="lessThan"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="lessThanOrEqualTo"/>
  </owl:disjointWith>
</owl:Class>

```

```

<owl:Class rdf:ID="Sequence">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#ControlStructure"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="Repetition"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Selection"/>
  </owl:disjointWith>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >same with compound statement: a group of statements bracketed by { and } that are executed sequentially.</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#DataStructure">
  <owl:disjointWith>
    <owl:Class rdf:about="#Function"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Condition"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Recursion"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#IPSP"/>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#Variable"/>
  <owl:disjointWith rdf:resource="#Operator"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#ControlStructure"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Statement"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#String"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#DataType"/>
  </owl:disjointWith>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >a composite of related data items stored under the same name.</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#arithmetic">
  <rdfs:subClassOf rdf:resource="#Operator"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#Binary"/>

```

```

</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:about="#Equality"/>
</owl:disjointWith>
<owl:disjointWith rdf:resource="#Logical"/>
<owl:disjointWith>
  <owl:Class rdf:about="#Relational"/>
</owl:disjointWith>
<owl:disjointWith rdf:resource="#Unary"/>
</owl:Class>
<owl:Class rdf:about="#multiplication">
  <rdfs:subClassOf rdf:resource="#arithmetic"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#Addition"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#division"/>
  <owl:disjointWith rdf:resource="#remainder"/>
  <owl:disjointWith rdf:resource="#substraction"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >*</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="double">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >real number.</rdfs:comment>
  <owl:disjointWith>
    <owl:Class rdf:ID="char"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="int"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="standart"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="not">
  <owl:disjointWith>
    <owl:Class rdf:ID="and"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="or"/>
  </owl:disjointWith>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >!</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Logical"/>
</owl:Class>
<owl:Class rdf:about="#or">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```

```

>||</rdfs:comment>
<owl:disjointWith>
  <owl:Class rdf:about="#and" />
</owl:disjointWith>
<owl:disjointWith rdf:resource="#not" />
<rdfs:subClassOf rdf:resource="#Logical" />
</owl:Class>
<owl:Class rdf:ID="enumerated">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#DataType" />
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:about="#standart" />
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:about="#stack">
  <owl:disjointWith rdf:resource="#array" />
  <rdfs:subClassOf rdf:resource="#DataStructure" />
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >a data structure in which only the top element can be accessed.</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#Relational">
  <rdfs:subClassOf rdf:resource="#Operator" />
  <owl:disjointWith rdf:resource="#arithmetic" />
  <owl:disjointWith>
    <owl:Class rdf:about="#Binary" />
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Equality" />
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#Logical" />
  <owl:disjointWith rdf:resource="#Unary" />
</owl:Class>
<owl:Class rdf:about="#Recursion">
  <owl:disjointWith>
    <owl:Class rdf:about="#Condition" />
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#IPSP" />
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#Operator" />
  <owl:disjointWith>
    <owl:Class rdf:about="#ControlStructure" />
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#String" />
  </owl:disjointWith>

```

```

<owl:disjointWith rdf:resource="#Variable"/>
<owl:disjointWith>
  <owl:Class rdf:about="#Statement"/>
</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:about="#Function"/>
</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:about="#DataType"/>
</owl:disjointWith>
<owl:disjointWith rdf:resource="#DataStructure"/>
</owl:Class>
<owl:Class rdf:about="#Equality">
  <owl:disjointWith rdf:resource="#arithmetic"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#Binary"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#Logical"/>
  <owl:disjointWith rdf:resource="#Relational"/>
  <owl:disjointWith rdf:resource="#Unary"/>
  <rdfs:subClassOf rdf:resource="#Operator"/>
</owl:Class>
<owl:Class rdf:about="#Binary">
  <owl:disjointWith rdf:resource="#arithmetic"/>
  <owl:disjointWith rdf:resource="#Equality"/>
  <owl:disjointWith rdf:resource="#Logical"/>
  <owl:disjointWith rdf:resource="#Relational"/>
  <owl:disjointWith rdf:resource="#Unary"/>
  <rdfs:subClassOf rdf:resource="#Operator"/>
</owl:Class>
<owl:Class rdf:about="#string">
  <owl:disjointWith>
    <owl:Class rdf:about="#character"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#InputOutput_Function"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#mathematics"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#standart_function"/>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="#Selection">
  <owl:disjointWith rdf:resource="#Sequence"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#Repitition"/>

```

```

</owl:disjointWith>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>a control structure that chooses among alternative program statements. same as condition.</rdfs:comment>
<rdfs:subClassOf>
  <owl:Class rdf:about="#ControlStructure"/>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#IPSP">
  <owl:disjointWith>
    <owl:Class rdf:about="#DomainConcept"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="LoopStatement">
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Using For</rdfs:comment>
<owl:disjointWith>
  <owl:Class rdf:ID="WhileStatement"/>
</owl:disjointWith>
<rdfs:subClassOf>
  <owl:Class rdf:about="#Repetition"/>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#lessThanOrEqualTo">
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>&lt;=</rdfs:comment>
<owl:disjointWith>
  <owl:Class rdf:about="#greaterThan"/>
</owl:disjointWith>
<owl:disjointWith rdf:resource="#greaterThanOrEqualTo"/>
<owl:disjointWith>
  <owl:Class rdf:about="#lessThan"/>
</owl:disjointWith>
<rdfs:subClassOf rdf:resource="#Relational"/>
</owl:Class>
<owl:Class rdf:about="#ControlStructure">
  <owl:disjointWith rdf:resource="#DataStructure"/>
  <owl:disjointWith rdf:resource="#Recursion"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#DataType"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#String"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#Operator"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#Function"/>
  </owl:disjointWith>

```

```

<owl:disjointWith>
  <owl:Class rdf:about="#Statement" />
</owl:disjointWith>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>control the flow of the execution in a program or function. a combination individual instructions into a single logical unit
with one entry point and one exit point.</rdfs:comment>
<rdfs:subClassOf rdf:resource="#IPSP" />
<owl:disjointWith rdf:resource="#Variable" />
<owl:disjointWith>
  <owl:Class rdf:about="#Condition" />
</owl:disjointWith>
</owl:Class>
<owl:Class rdf:about="#DataType">
<rdfs:subClassOf rdf:resource="#IPSP" />
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>a set of values and operations that can be performed on those values.</rdfs:comment>
<owl:disjointWith>
  <owl:Class rdf:about="#Statement" />
</owl:disjointWith>
<owl:disjointWith rdf:resource="#Variable" />
<owl:disjointWith>
  <owl:Class rdf:about="#String" />
</owl:disjointWith>
<owl:disjointWith rdf:resource="#ControlStructure" />
<owl:disjointWith rdf:resource="#Operator" />
<owl:disjointWith rdf:resource="#DataStructure" />
<owl:disjointWith>
  <owl:Class rdf:about="#Function" />
</owl:disjointWith>
<owl:disjointWith rdf:resource="#Recursion" />
<owl:disjointWith>
  <owl:Class rdf:about="#Condition" />
</owl:disjointWith>
</owl:Class>
<owl:Class rdf:about="#WhileStatement">
  <owl:disjointWith rdf:resource="#LoopStatement" />
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Repetition" />
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Function">
  <owl:disjointWith rdf:resource="#DataStructure" />
  <owl:disjointWith>
    <owl:Class rdf:about="#String" />
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#Variable" />
  <owl:disjointWith rdf:resource="#Operator" />

```

```

<owl:disjointWith>
  <owl:Class rdf:about="#Condition"/>
</owl:disjointWith>
<owl:disjointWith rdf:resource="#ControlStructure"/>
<owl:disjointWith>
  <owl:Class rdf:about="#Statement"/>
</owl:disjointWith>
<owl:disjointWith rdf:resource="#Recursion"/>
<owl:disjointWith rdf:resource="#DataType"/>
<rdfs:subClassOf rdf:resource="#IPSP"/>
</owl:Class>
<owl:Class rdf:ID="nestedIFStatement">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Ifstatement"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Statement">
  <rdfs:subClassOf rdf:resource="#IPSP"/>
  <owl:disjointWith rdf:resource="#Function"/>
  <owl:disjointWith rdf:resource="#DataStructure"/>
  <owl:disjointWith rdf:resource="#Operator"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#String"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#Recursion"/>
  <owl:disjointWith rdf:resource="#Variable"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#Condition"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#DataType"/>
  <owl:disjointWith rdf:resource="#ControlStructure"/>
</owl:Class>
<owl:Class rdf:about="#mathematics">
  <owl:disjointWith>
    <owl:Class rdf:about="#character"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#InputOutput_Function"/>
  <owl:disjointWith rdf:resource="#string"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#standart_function"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#lessThan">
  <owl:disjointWith>
    <owl:Class rdf:about="#greatherThan"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#greatherThanOrEqualTo"/>

```

```

<owl:disjointWith rdf:resource="#lessThanOrEqualTo"/>
<rdfs:subClassOf rdf:resource="#Relational"/>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>&lt;/rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#char">
<rdfs:subClassOf>
  <owl:Class rdf:about="#standart"/>
</rdfs:subClassOf>
<owl:disjointWith rdf:resource="#double"/>
<owl:disjointWith>
  <owl:Class rdf:about="#int"/>
</owl:disjointWith>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>represents an individual character value: a letter, a digit, or a special symbol.</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#Repetition">
<owl:disjointWith rdf:resource="#Sequence"/>
<owl:disjointWith rdf:resource="#Selection"/>
<rdfs:subClassOf rdf:resource="#ControlStructure"/>
</owl:Class>
<owl:Class rdf:about="#standart">
<rdfs:subClassOf rdf:resource="#DataType"/>
<owl:disjointWith rdf:resource="#enumerated"/>
</owl:Class>
<owl:Class rdf:about="#DomainConcept">
<owl:disjointWith rdf:resource="#IPSP"/>
</owl:Class>
<owl:Class rdf:about="#Addition">
<rdfs:subClassOf rdf:resource="#arithmetic"/>
<owl:disjointWith rdf:resource="#division"/>
<owl:disjointWith rdf:resource="#multiplication"/>
<owl:disjointWith rdf:resource="#remainder"/>
<owl:disjointWith rdf:resource="#subtraction"/>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>+</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="NestedLoop">
<rdfs:subClassOf rdf:resource="#LoopStatement"/>
</owl:Class>

<owl:Class rdf:about="#and">
<owl:disjointWith rdf:resource="#not"/>
<owl:disjointWith rdf:resource="#or"/>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>&amp;&amp;</rdfs:comment>
<rdfs:subClassOf rdf:resource="#Logical"/>

```

```

</owl:Class>
<owl:Class rdf:about="#standart_function">
  <rdfs:subClassOf rdf:resource="#Function"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >functions bundled in C.</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#Ifstatement">
  <rdfs:subClassOf rdf:resource="#Selection"/>
  <owl:disjointWith rdf:resource="#SwitchStatement"/>
</owl:Class>
<owl:Class rdf:about="#character">
  <owl:disjointWith rdf:resource="#InputOutput_Function"/>
  <owl:disjointWith rdf:resource="#mathematics"/>
  <owl:disjointWith rdf:resource="#string"/>
  <rdfs:subClassOf rdf:resource="#standart_function"/>
</owl:Class>
<owl:Class rdf:about="#int">
  <rdfs:subClassOf rdf:resource="#standart"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >whole number</rdfs:comment>
  <owl:disjointWith rdf:resource="#char"/>
  <owl:disjointWith rdf:resource="#double"/>
</owl:Class>
<owl:Class rdf:about="#Condition">
  <owl:disjointWith>
    <owl:Class rdf:about="#String"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#Function"/>
  <owl:disjointWith rdf:resource="#Statement"/>
  <owl:disjointWith rdf:resource="#Recursion"/>
  <owl:disjointWith rdf:resource="#DataStructure"/>
  <owl:disjointWith rdf:resource="#Operator"/>
  <owl:disjointWith rdf:resource="#ControlStructure"/>
  <owl:disjointWith rdf:resource="#Variable"/>
  <rdfs:subClassOf rdf:resource="#IPSP"/>
  <owl:disjointWith rdf:resource="#DataType"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >an expression that is either false(represented by 0) or true (usually represented by 1)</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#String">
  <rdfs:subClassOf rdf:resource="#IPSP"/>
  <owl:disjointWith rdf:resource="#ControlStructure"/>
  <owl:disjointWith rdf:resource="#Operator"/>
  <owl:disjointWith rdf:resource="#Recursion"/>
  <owl:disjointWith rdf:resource="#Function"/>
  <owl:disjointWith rdf:resource="#DataStructure"/>
  <owl:disjointWith rdf:resource="#DataType"/>

```

```

    <owl:disjointWith rdf:resource="#Statement" />
    <owl:disjointWith rdf:resource="#Variable" />
    <owl:disjointWith rdf:resource="#Condition" />
  </owl:Class>
  <owl:Class rdf:about="#greaterThan">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >&gt;</rdfs:comment>
    <owl:disjointWith rdf:resource="#greaterThanOrEqualTo" />
    <owl:disjointWith rdf:resource="#lessThan" />
    <owl:disjointWith rdf:resource="#lessThanOrEqualTo" />
    <rdfs:subClassOf rdf:resource="#Relational" />
  </owl:Class>
  <owl:Class rdf:about="#output_function">
    <rdfs:subClassOf rdf:resource="#InputOutput_Function" />
    <owl:disjointWith rdf:resource="#input_function" />
  </owl:Class>
  <owl:Class rdf:about="#Interactive">
    <rdfs:subClassOf rdf:resource="#CourseMaterial" />
    <owl:disjointWith rdf:resource="#Social" />
    <owl:disjointWith rdf:resource="#Static" />
  </owl:Class>
  <owl:ObjectProperty rdf:ID="hasCategory">
    <rdfs:range rdf:resource="#CourseCategory" />
    <rdfs:domain rdf:resource="#Course" />
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasRole">
    <rdfs:range rdf:resource="#Role" />
    <rdfs:domain rdf:resource="#User" />
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasActivity">
    <rdfs:domain rdf:resource="#Module" />
    <rdfs:range rdf:resource="#Activity" />
  </owl:ObjectProperty>
  <character rdf:ID="toupper" />
  <Activity rdf:ID="Lesson" />
  <rdf:Description rdf:ID="NoneditingTeacher">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Non-editing teachers can teach in courses and grade students, but may not alter activities.</rdfs:comment>
  </rdf:Description>
  <Activity rdf:ID="Wiki" />
  <Activity rdf:ID="Assignment" />
  <Activity rdf:ID="Label" />
  <string rdf:ID="strncat" />
  <mathematics rdf:ID="floor" />
  <Role rdf:ID="Guest">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Guests have minimal privileges and usually can not enter text anywhere.</rdfs:comment>

```

```

</Role>
<string rdf:ID="strncpy"/>
<character rdf:ID="isdigit"/>
<output_function rdf:ID="printf"/>
<mathematics rdf:ID="sin"/>
<Role rdf:ID="CourseCreator">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Course creators can create new courses and teach in them.</rdfs:comment>
</Role>

<string rdf:ID="strcpy"/>
<Activity rdf:ID="SCORM"/>
<mathematics rdf:ID="cos"/>

<mathematics rdf:ID="abs"/>
<mathematics rdf:ID="log10"/>
<string rdf:ID="strncpy"/>
<input_function rdf:ID="scanf"/>
<mathematics rdf:ID="tan"/>
<character rdf:ID="isalpha"/>
<mathematics rdf:ID="log"/>

<mathematics rdf:ID="exp"/>

<string rdf:ID="strlen"/>

<mathematics rdf:ID="pow"/>
<mathematics rdf:ID="sqrt"/>
<mathematics rdf:ID="fabs"/>

<string rdf:ID="strcmp"/>

<mathematics rdf:ID="ceil"/>
<character rdf:ID="islower"/>
<character rdf:ID="isspace"/>

<string rdf:ID="strtok"/>

<character rdf:ID="tolower"/>

<string rdf:ID="strcat"/>
<Role rdf:ID="Teacher"/>
<character rdf:ID="ispunct"/>
</rdf:RDF>
<!-- Created with Protege (with OWL Plugin 3.3, Build 418) http://protege.stanford.edu -->

```

