A FRAMEWORK FOR AN ADAPTABLE AND PERSONALISED E-LEARNING SYSTEM BASED ON FREE WEB RESOURCES

Eiman Aeiad

Thesis Submitted in Partial Fulfilment of the Requirements of

the Degree of Doctor of Philosophy

School of Computing, Science and Engineering

University of Salford, UK

2017

Table of Contents

List of Figures

List of Table

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Farid Meziane for the continuous support throughout my PhD, for his patience, motivation, and immense knowledge. His guidance helped me through all the time of the research and the writing process of this thesis. I could not have imagined having a better advisor and mentor for my PhD study. Similar great gratitude goes to all the experts who were willing to participate in the study and gave their valuable feedback. A special thanks to my family. Words cannot express how grateful I am to my parents who raised me with a love of science and supported me in all my pursuits. I pass all my success to my father's soul, and deeply appreciate my mother sacrifices that she has made on my behalf. Thanks to my brothers and sisters, your prayers and support for me was what sustained me this far. I owe thanks to my friends and colleagues for their support and unconditional help whenever I needed. At the end, I would like to express tremendous appreciation to my beloved wife for her continuous and unfailing love, support and understanding during my pursuit of PhD degree that made the completion of thesis possible. You were always around at times I thought that it is impossible to continue, you helped me to keep things in perspective. I appreciate my little girls Mayar and Menatallah for abiding my ignorance and the patience they showed during my thesis writing and hard times throughout my PhD. Words would never say how grateful I am to both of you. I consider myself the luckiest in the world to have such a lovely and caring family, standing beside me with their love and unconditional support.

List of Publications

The following papers have been published as part of this research. Parts of the papers have been included in the thesis.

Aeiad, E. and Meziane, F (2014), An Adaptable and Personalized E-learning System Based On Freely Available Resources on the WWW, UK Poster Abstract Research Showcase College of Science and Technology MediaCity UK, Wednesday 18th June 2014. University of Salford.

Eiman Aeiad and Farid Meziane (2015), An Adaptable and Personalised E-Learning System Based on Free Web Resources, in Biemann, C., Handschuh, S., Freitas, A., Meziane, F., Métais, E. (Eds.), Natural Language Processing and Information Systems, Proceedings 20th International Conference on Applications of Natural Language to Information Systems, NLDB 2015, Passau, Germany, June 17-19, Springer.

Eiman Aeiad and Farid Meziane (2016), Validating the Learning Outcomes of an E-Learning System Using NLP, in E. Métais, F. Meziane, M. Saraee, V. Sugumaran and S. Vadera (editors), Natural Language Processing and Information Systems, Proceedings of the 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016, Salford, UK, June 22-24, LNCS 9612 Springer, pp: 292-300.

Eiman Aeiad and Farid Meziane (2017), An Adaptable and Personalised E-Learning System Applied to Computer Science Programmes Design. International Journal of Artificial Intelligence in Education, 1-19. Submitted awaiting for decision

List of Abbreviations

ACM	Association for Computing Machinery
AHS	Adaptive Hypermedia System
AI	Artificial Intelligent
APELS	Adaptable and Personalised E-learning System
BoK	Body of Knowledge
CS	Computer Science
HTML	Hypertext Markup Language
IEEE	Institute for Electrical and Electronic Engineers
IE	Information Extraction
IR	Information Retrieval
IRS	Information Retrieval System
ITS	Intelligent Tutoring System
KAs	Knowledge Areas
KUs	Knowledge Units
LMS	Learning Management System
LSI	Learning Style Inventory
NLP	Natural Language Process
OWL	Ontology Web Language
PHP	Hypertext Preprocessor
PoS	Part of Speech
NER	Named Entity Recognition
VARK	Visual Auditory Read/write Kinaesthetic
XML	EXtensible Mark-up Language
XHTML	Extensible Hypertext Markup Language

Abstract

An adaptable and personalised E-learning system (APELS) architecture is developed to provide a framework for the development of comprehensive learning environments for learners who cannot follow a conventional programme of study. The system extracts information from freely available resources on the Web taking into consideration the learners' background and requirements to design modules and a planner system to organise the extracted learning material to facilitate the learning process. The process is supported by the development of an ontology to optimise and support the information extraction process. Additionally, natural language processing techniques are utilised to evaluate a topic's content against a set of learning outcomes as defined by standard curricula. An application in the computer science field is used to illustrate the working mechanisms of the proposed framework and its evaluation based on the ACM/IEEE Computing Curriculum.

A variety of models are developed and techniques used to support the adaptability and personalisation features of APELS. First, a learner's model was designed by incorporating students' details, students' requirements and the domain they wish to study into the system. In addition, learning style theories were adopted as a way of identifying and categorising the individuals, to improve their on-line learning experience and applying it to the learner's model. Secondly, the knowledge extraction model is responsible for the extraction of the learning resources from the Web that would satisfy the learners' needs and learning outcomes. To support this process, an ontology was developed to retrieve the relevant information as per users' needs. In addition, it transforms HTML documents to XHTML to provide the information in an accessible format and easier for extraction and comparison purposes. Moreover, a matching process was implemented to compute the similarity measure between the ontology concepts that are used in the ACM/IEEE Computer Science Curriculum and those

extracted from the websites. The website with the highest similarity score is selected as the best matching website that satisfies the learners' request.

A further step is required to evaluate whether the content extracted by the system is the appropriate learning material of the subject. For this purpose, the learning outcome validation process is added to ensure that the content of the selected websites will enable the appropriate learning based to the learning outcomes set by standard curricula. Finally, the information extracted by the system will be passed to a Planner model that will structure the content into lectures, tutorials and workshops based on some predefined learning constraints.

The APELS system provides a novel addition to the field of adaptive E-learning systems by providing more personalized learning material to each user in a time-efficient way saving his/her time looking for the right course from the hugely available resources on the Web or going through the large number of websites and links returned by traditional search engines. The APELS system will adapt better to the learner's style based on feedback and assessment once the learning process is initiated by the learner. The APELS system is expected to develop over time with more users.

Chapter 1 : Introduction

1.1 Introduction and Motivation

Learning is greatly influenced by the development of Information and Communication Technologies (ICTs) and advanced digital media. Learning using these new media is referred to as E-learning. It allows access to education to those who find it difficult to be physically present in the traditional classroom based learning (Uhomoibhi, 2006) or complement it.

The education process is usually referred to as the knowledge transfer process in which classes, lectures or workshops are conducted by the instructor to transfer the information and knowledge to the apprentice (Sloman, 2001). In today's era, the way education is delivered has changed with the use of new multimedia means like videos and audios making it captivating and alluring (Stotz et al., 2017, Yarkova et al., 2017). Although these advanced techniques are attractive, the learning process held in the class setup in which learners are taught by educators through dialogues seems to be the preferred-instructional approach. In this approach, the process of education is held through face to face interaction and is restricted only to the learners present in the classroom. With this approach, it becomes difficult for students who live in far-flung areas or working to be physically in the class and attend, as a result, they struggle to follow and complete their education. These problems are handled commonly through the introduction of E-learning. This technique helps learners to study at any time and place. People who are not able to take out time from their regular routines can easily get an education through E-learning (Yieke, 2005).

Oblinger and Hawkins (Oblinger and Hawkins, 2005) stated that the number of adult learners is increasing and E-learning is a worthy choice for these people to carry out family duties, work and education equally. The flexible nature of E-learning helps them to complete their courses easily. Other than flexibility, there are lots of other advantages which E-learning offers including cost saving for both travelling to the learning place and time spent missing work

(Cantoni et al., 2004). In addition, various interaction media offered by E-learning systems aid in developing interest groups where members can learn from each other through participation in discussions, different points of view, understanding new concepts and learning from each other's mistakes (Cantoni et al., 2004).

Education is perceived by each individual as per his/her own needs, specific learning ways and interests. Personalisation has that trait because of which it could be considered as the innate feature of E-learning. Learners can have an easy approach to the material available on E-learning platforms at any time and any place and can show the completion of particular tasks and learning outcomes, but it is not guaranteed that the easy approach to teaching material can result in effective learning and education outcomes (Henderson et al., 2009). Currently, the E-learning platforms do not offer personalised learning options to learners as identical content and practices are available for all users. The term personalisation is defined as introducing the learning knowledge to various learners based on the evaluation of their skills, knowledge and learning priorities (Bittencourt et al., 2008). The main objective of personalised learning approaches is to customise the teaching according to the specific skills and needs of the learner. Certain barriers like location, time etc. to access the education are eradicated by the personalised learning approach (Sampson and Karagiannidis, 2010).

1.2 Rationale

Nowadays, various E-learning systems have been used by many people all over the world. However, the high diversity of learners on the Internet poses new challenges to the traditional "one-size-fits-all" learning model, in which a single set of learning resources is provided to all learners (Li and Chang, 2005). As a result of this, the Adaptive and Personalised E-learning System (APELS) will be introduced to extend the current understanding and use of conventional E-learning resources, by using freely available resources on the Web to design and deliver content for individual learners. The system can be used by individual learners, universities that may not have the resources and the expertise to develop learning resources and anyone who wishes to learn a specific filed. At this stage, the academic learning that is conducted via APELS is based on resources available freely on the Web and does not require the involvement of field experts. The APELS system will enable users to design their own learning material based on internationally recognised curricula and contents. Using standard search engines to find learning material that is suitable for individual learners is time consuming and may not lead to a suitable outcome. The major contribution of this research is therefore the development of an intelligent system to support online course design based solely on freely available resources on the Web.

Furthermore, the APELS system will address three main issues. The first issue is the identification of the learner's requirements, such as learning style and field of interest. The user can then specify a particular model to allow the system to provide the exact information required. The second issue is structuring the knowledge domain of a pre-selected area using an ontology, in order to extract relevant learning resources from the Web. Furthermore, defining the major similarity between the learning outcomes as defined by standard curricula and the content extracted from relevant websites, to enable the appropriate learning of the subject will be addressed by this system. The last issue to be addressed is the ability of the APELS system to adapt and modify the content and learning style based on the interactions of the users with the system over a period of time. In addition, the information extracted by the system will be passed to a Planner model to structure the content into lectures/tutorials and workshops based on some predefined constraints such as time.

1.3 Research Question

Is it possible to provide personalised learning material with the content automatically extracted from freely available resources on the Web to an individual learner according to his/her learning needs and style? The idea behind the knowledge extraction of the APELS system is illustrated in Figure 1-1. APELS will select learning material from an enormous number of freely available resources on the Web (dots in Figure 1-1) according to student's constraints, which are the student's need, learning style and learning outcomes for the chosen knowledge domain.

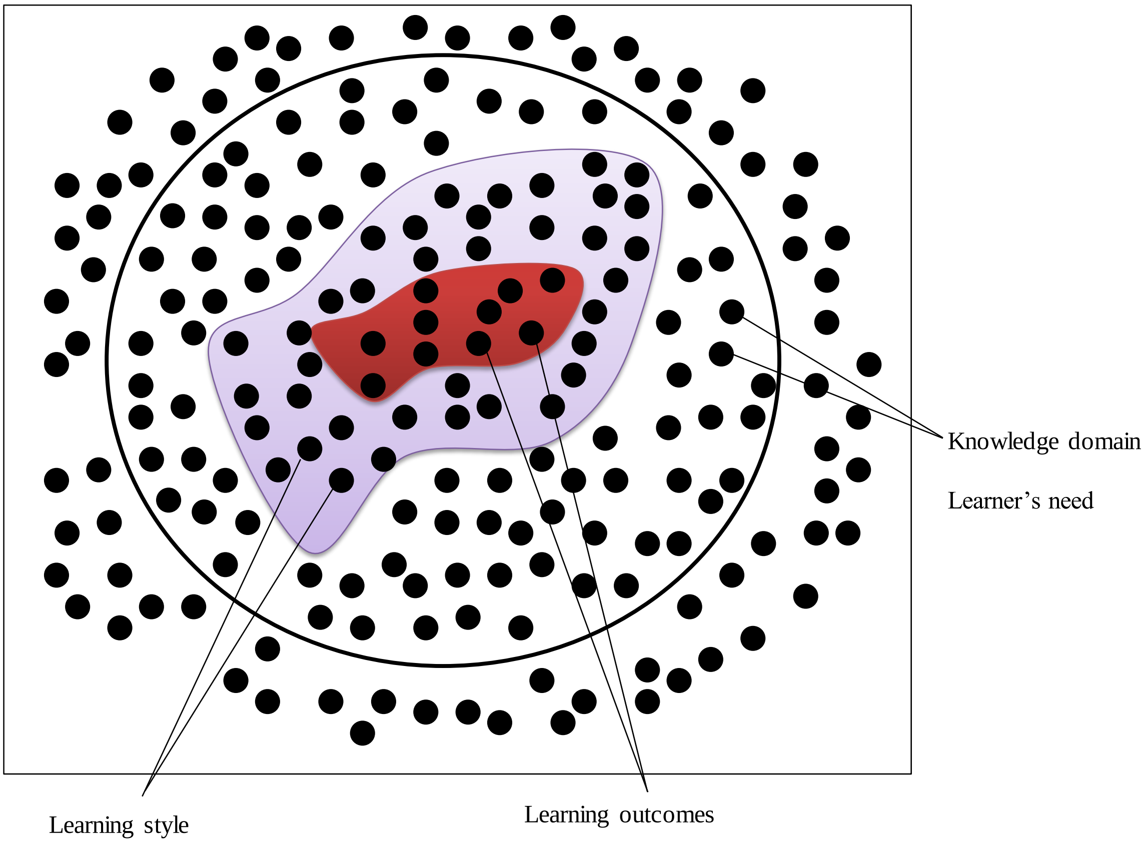

Figure 1-1 Knowledge extraction based on students' constrains

1.4 Aim and Objectives

The project aim, concerns the development of an intelligent system to support online course design based solely on the information extracted from available resources on the Web. The overall architecture of the system will be designed to allow the application of the system to various educational fields.

The objectives of the research will include:

1. Review the existing work on automatic knowledge extraction from the Web.
2. Review the current platforms of E-learning systems.

3. Review learning styles and their use in curriculum design.
4. Design the architecture of the Adaptable and Personalised E-learning System (APELS).
5. Implement the proposed architecture using a specific field of learning with a well-defined curriculum content.
6. Integrate the components of the APELS system and produce a computer based system that can be used by the learners.
7. Evaluate the proposed system by experts from the field and education.
8. Evaluate the experience learned during design and implementation of the APELS system, and discus future improvements.

1.5 Research Contribution

Through the analysis, design, creation and evaluation of the APELS system, this research has provided novel contributions to the E-learning area and the computer science field.

In the E-learning area the contributions can be summarised as follows:

- APELS defines the major similarities between the learning outcomes as specified by the standard curricula and the content of the extracted websites to enable the appropriate learning of the subject.
- APELS provides a very important addition to the world of adaptive learning as it enables the portability of the system to other domains by reusing the same architecture and rules without changing the system but by only replacing the ontology.
- APELS can be used by individual learners as well as educational institutions that may lack the expertise to develop learning resources cost-effectively; the academic learning that is conducted via APELS is based on resources available freely on the Web and does not require the involvement of field experts.

- APELS enables users to design their own learning material based on internationally recognised curricula and contents, which will be presented in their preferred learning style making learning a more interesting and enjoyable experience.

In the computer science field:

- A new linguistic method was developed to evaluate the learning material of a particular module against a set of pre-defined learning outcomes, based on determining the linguistic patterns in NLP.
- Adaptability as an important intelligence feature was considered in APELS. The system can adapt and modify the content and the learning styles based on the interactions of the users with the system over a period of time. This is based on calculating a satisfaction score that is fed back to the system.

1.6 Limitations of the Current Research

Throughout this project there have been a number of limitations that could be summarised as follows:

1. Given the strict duration of the PhD program it was not possible to evaluate the APELS system with real learners during a substantial period of the time that will allow the evaluation of their learning experience and the full adaptability mechanism of the system.
2. The system will need a continuous updating of the synonyms of the concepts in the domain ontology and this is not automatic in the current version of the system.
3. Expanding the system to include other media learning resources such as video and audio files as the current version of the system considers only text based resources.

1.7 Phases of the Research

The purpose of this thesis is to provide a comprehensive intelligent organisation of materials available on the Web based on student's constraints which are student's need, learning style and learning outcomes. Therefore, this study is conducted in sixth phases (Figure 1-2):

The first phase, is concerned with the review and analysis of a number of diverse approaches in implementing personalised E-learning systems in order to present the differences between these systems and highlight the shortcomings of the existing models.

The second phase, concerns the design of the interface in order to identify learners' requirements such as learner's background and learner's need. Also, the system uses the VARK learner model in order to identify the learner's learning style. Then, the system introduces two ways of presenting the content for the learner to choose from according to their preferences.

The third phase, concerns the information extraction approach, which is used to extract learning resources from the Web that are suitable for the learner. This approach is based on an ontology in order to retrieve relevant information as per user request. In addition, it transforms HTML document to XHTML to provide the information in a friendly accessible format and easier for extraction and comparison. Moreover, a matching process will be implemented to search for websites that have the highest probability of satisfying the learners requirements. After the matching process, a further step is required to evaluate the content against a set of learning outcome as defined by standard curricula.

In the fourth phase, the information extracted by the system will be passed to a Planner that will structure it into lectures/tutorials and workshops based on some predefined learning times.

The fifth phase, concerns the validation of the designed courses with standard curriculum with computer science as the learning domain will be performed.

The sixth phase, involves the validation of the course content by education specialists.

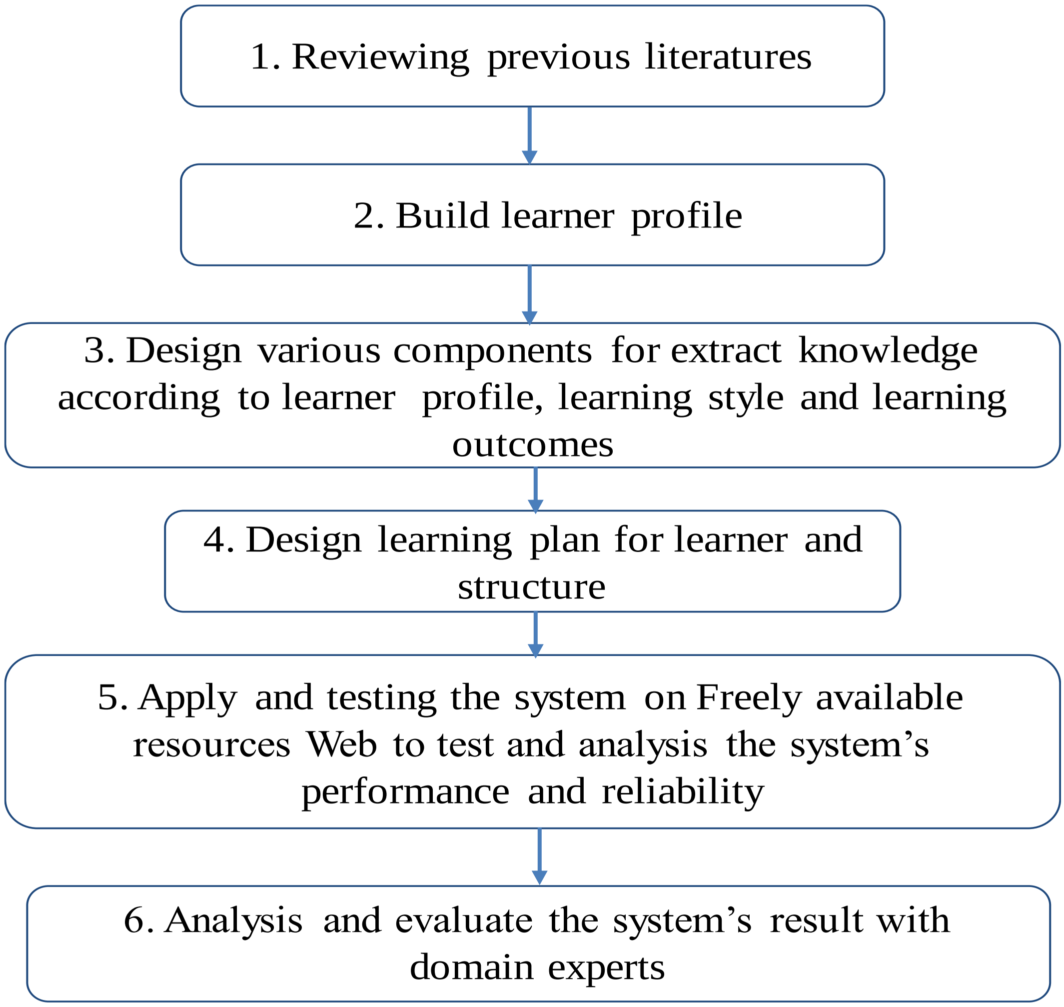

Figure 1-2 Phases of the research

1.8 Thesis Organisation

The remaining of the thesis is organised as follows:

Chapter 2 reviews and provides the background of the natural language processing (NLP) field, its applications and knowledge resources. In addition, it presents the information extraction and information retrieval techniques.

Chapter 3 reviews the background and design of E-learning systems, and how they develop into adaptive E-learning systems. Furthermore, it presents an overview of learning theory and learning style. Finally, the chapter reviews some related works by outlining different methods that are used in personalised E-learning systems and their evaluation.

Chapter 4 describes the architecture of the APELS system which is based on three main models that will form the main components of the system. Firstly, it presents the learner's model which contains all the needed information about the learner. Secondly, it introduces the information extraction model that is used to extract learning resources from the Web. Finally, the chapter presents the content of the delivery model that generates and structures the learning plan for the module including the content.

Chapter 5 illustrates the working mechanism of the APELS system, how it would be used in practice including implementation of its rules and approaches using the ACM/IEEE Computer Society Computer Science Curriculum as case study.

Chapter 6 discusses the system evaluation to test the research hypothesis from the perspectives of experts. It describes the setting of the experiment, which includes various phases such as testing the system usability, evaluating the quality of the produced content, and general discussion.

Chapter 7 presents the conclusion, including reflection on the extent to which the research objectives have been met and the potential directions for future works.

Chapter 2 : Natural Language Processing and Knowledge Engineering Background

This chapter introduces the natural language processing (NLP) field, its applications and knowledge resources that are used for developing the APELS system. Furthermore, it reviews the NLP tools that have been selected and used for this research and justifies the motivational reasons behind choosing such techniques. In addition, the chapter gives an overview of information retrieval and information extraction processes focusing on the important tasks of information retrieval, which are used throughout this research.

2.1 An Overview of Natural Language Processing

Natural Language (NL) is known for its ambiguity and openness to many and different interpretations. This is due to lexical ambiguities, such as words written in the same way but having different meanings or grammatical as verbs phrases modifying different parts of a sentence. Moreover, the interpretation of natural language statements often depends on the context. The Internet is a very rich source of information that is exploited and used by many applications and users. This information is often expressed in NL and referred to in the literature as unstructured data. Examples of such unstructured data is made available in online sources such as news articles, blogs, social media and surveys all presented as dynamic or static webpages. To exploit this plethora of unstructured data, tools that are able to analyse, disambiguate and process NL resources are needed (Manning and Schütze, 1999, Jackson and Moulinier, 2007, Jurafsky and Martin, 2009, Collobert et al., 2011). Some of these tools are attempting to transform the unstructured data into a structured one as they are much easier to process.

In addition, the interactions that occur between NL and computers are analysed through Natural Language Processing (NLP), which is a specific field in Artificial Intelligence (AI) (Russell et al., 2003). This utilises software together with the most advanced systems for linguistic pattern analysis, in order to generate and determine vital rules that are evident in specific texts. For instance, NLP demonstrates fixed rules of grammar that show how sentences are constructed. This is achieved by applying a group of NLP tasks usually supported by tools.

2.1.1 Natural Language Processing Tasks

The linguistic analysis of different grammatical structures that are referred to as parts of speech (PoS), grammatical structure, or semantic information are the basis for NLP tasks and these are used for processing unstructured texts. As a result, the texts become structured, as relevant features are extracted from parts of the text, and these individual parts are then used to detail syntactic information (Manning and Schütze, 1999, Jackson and Moulinier, 2007, Jurafsky and Martin, 2009, Collobert et al., 2011).The main NLP tasks include tokenisation (Webster and Kit, 1992), part of speech (PoS) tagging (Brill, 1992), lemmatisation (Porter, 1980), named entity recognition (NER) (Jurafsky and Martin, 2009), machine translation (MT) (Manning et al., 2014), and co-reference resolution (Stoyanov and Eisner, 2012). These tasks are introduced in the following subsections.

2.1.1.1 Tokenisation

Tokenisation is a common process in NLP tasks, in which a source text is initially split into smaller units or tokens for subsequent processing and analysis (Webster and Kit, 1992). For instance, documents are broken into paragraphs, paragraphs into sentences, and sentences into individual tokens. Tokens can be words, symbols, numbers, punctuation or the space token. In addition, during the tokenisation process, word boundaries are detected. The ending point of a word and the beginning of the next word is referred to as word boundaries.

2.1.1.2 Part of Speech Tagging

Part of Speech Tagging (PoS) provides a syntactic analysis for each word in the text (e.g. ‘drive’: verb, ‘girl’: noun, ‘by’: preposition, ‘usually ‘: adverb, ‘young ‘: adjective etc.). This has been widely advocated by many researchers to be one of the main tasks in syntactic text analysis, as it is useful for linguistic analysis and is commonly undertaken on texts that are tokenised (Brill, 1992, Hepple, 2000). Moreover, the linguistic features of each word as extracted by a PoS tagger are useful for automatic keyword and key phrases extraction. For instance, Hulth (2003) used a supervised learning system for keyword extraction from abstracts, using a combination of lexical and syntactic features. According to Hulth, keyword extraction from abstracts is more widely applicable than from full texts, since many documents on the Internet are not available as full-texts, but only as abstracts. In her work, adding linguistic knowledge such as part of speech improved keywords extraction and the experiment of this work showed that the accuracy of the system almost doubled by adding linguistic knowledge to the term representation. Furthermore, an improved keyword extraction method (Extended TF) was proposed by Hong and Zhen (2012). They utilised linguistic features of keywords like word frequency, part of speech, syntactical function of words, and the position where the word appears and word's morphology. On the basis of the characteristics of each feature, weights were ascribed to different features and the Super Vector Machine (SVM) model was utilised to optimize the results of key words extraction.

2.1.1.3 Lemmatisation

Lemmatisation (morphological analysis), is the process of analysing the structure of the word in order to identify its root such as is, are → be, bicycles, bicycle's, bicycles' → bicycle. For grammatical reasons, different documents and texts will use different forms of the words. Therefore, Lemmatization would be useful when searching for a specific form of a word to return all the documents that contains the other structures that are derived from the same root (Porter, 1980).

2.1.1.4 Named Entity Recognition

Named Entity Recognition (NER) is the task of identifying certain types of named expressions in unstructured text and classify them into a predefined set of categories. These expressions can be personal and geographic named expressions, as well as temporal and numeric ones. NER systems may utilises heuristic rules that rely on the syntactic structures of the surrounding context. NER is a crucial constituent of many NLP applications (Jurafsky and Martin, 2009). Examples of these applications include Machine Translation, Text Summarization, Opinion Mining, and Semantic Web Searching (Benajiba et al., 2009).

Several studies have been applied to NER to identify important entities. For example, Ek et al. (2011) developed an approach for a NER system to recognize named entities for short text messages (SMS) in Swedish that runs on a mobile platform. Moreover, Jiang et al. (2010) examined the use of Hidden Markov Model (HMM) to extract named entities related to events or activities from SMS messages in Chinese. In addition, Polifroni et al. (2010) proposed an approach that employs logistic regression to recognize name, location, date, and time entities from spoken or typed messages. They built a corpus from transcribed utterances of English SMS messages from real users in a laboratory setting. The problem of identifying proper names is not straight forward for some languages. For example, it is particularly difficult for Arabic, since names in the Arabic language do not start with capital letters like in English. Therefore, Abuleil (2004) presented an approach to extract names from text by building a database and graphs to represent the words that might form a name and the relationships between them. Similarly, Zayed and El-Beltagy (2015) proposed an approach that adopts a rule-based model combined with a statistical model. The statistical model is based on association rules and is built by employing unsupervised learning of context patterns that indicate the presence of a persons' name in Arabic Tweets.

2.1.1.5 Machine Translation

Machine Translation (MT) is the task of automatically converting one natural language text into another, preserving the meaning of the input text, and producing fluent text in the output language (Manning et al., 2014). While machine translation is one of the oldest subfields of artificial intelligence research, the recent shift towards large-scale empirical techniques has led to very significant improvements in translation quality. MT implements various approaches that determine the rules of linguistics that interpret a text's words linguistically through the process of analysis on all the different linguistic features via two main approaches: rule-based and statistical Machine Translation. The rule-based approach utilises rules and structures of the grammar, together with dictionaries in order to ascertain linguistic information that consequently creates the target language from the source language. Comparatively, the statistical approach develops translations via the process of advancing previously translated large text corpora that translates texts which are similar or original (Locke and Booth, 1955, Hutchins, 2007).

2.1.1.6 Co-reference Resolution

Co-reference resolution is the task of finding all expressions in a text that refer to the same entity. For instance, "Mark went to the school yesterday, he was on a holiday for three days", the word "he" and "Mark" in the given sentence refer to the same person and the task of co-reference is to link these two words. There are many open source tools available for co-reference resolution that explore entities and tacitly assume a correspondence between them in the text such as the GuiTAR system (Poesio and Kabadjov, 2004), BART (Versley et al., 2008), JAVARAP (Qiu et al., 2004), and the Reconcile system (Stoyanov et al., 2010). For example, Reconcile is the most common co-reference resolution system, which can be run on a piece of plain text to produce annotated co-reference resolution (Stoyanov et al., 2010). It employs supervised machine learning classifiers from the Weka toolkit (Hall et al., 2009), as well as

NLP tasks such as sentence splitting, tokenisation, and PoS tagging in the OpenNLP tool (Baldridge, 2005) and Stanford named entity recognition (Manning et al., 2014) to facilitate rapid creation of co-reference resolution systems.

2.1.2 Natural Language Processing Tools

NLP is the computational technique for analysing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications (Liddy, 2001). Most NLP tools are influenced by relatively newer areas such as Machine Learning, Computational Statistics and Cognitive Science, in order to perform annotations on words and terminologies to identify real world objects, and their relationships in the text (Madnani and Dorr, 2010). There are many open source tools available for NLP for semantic annotation of textual documents such as OpenNLP (Baldridge, 2005), NLTK (Loper and Bird, 2002), GATE (Cunningham et al., 2002), and Stanford CoreNLP (Manning et al., 2014). In the following paragraphs, some of these NLP tools are described in more detail.

2.1.2.1 OpenNLP

The OpenNLP tool is a free toolkit, which creates textual linguistic annotations that are based on maximum entropy use of statistical models (Baldridge, 2005). It integrates many NLP tasks including tokenization, sentence segmentation, PoS tagging, NER, chunking and co-reference resolution.

The OpenNLP toolkit has been used in different studies and domains. For example, the OpenNLP system was used for the development of a biomedical corpus for performing noun phrase chunking (NP) by Kang et al. (2011). Separately, an approach in relation to named entity detection in an SMS corpus in the Swedish language was described by Ek et al. (2011). They used the OpenNLP toolkit to annotate the SMS corpus through applying part-of-speech (PoS) tagging tasks and tokenisation on the corpus. However, the OpenNLP toolkit (Baldridge, 2005) is provided as a full package, making it hard to use it in our system if only few components are needed.

2.1.2.2 NLTK

The Natural Language Toolkit (NLTK) is a Python package for natural language processing (Loper and Bird, 2002). NLTK provides and enables interfaces for the purpose of text processing, linguistic structure analysis and the access to large corpora collections. NLTK includes libraries and programs of NLP components such as tokenization, PoS tagging, parsing, chunking, semantic analysis, classification and clustering.

Many researchers have utilised NLTK in the process of analysing natural language and knowledge extraction. For example, Mckenzie et al. (2010) introduced a novel application for information extraction by extracting data from helicopter maintenance records to populate a database. They used NLTK to implement a partial parsing of text by way of hierarchical chunking of the text. Additionally, Stoyanchev et al. (2008) developed a question answering system that employed the NLTK toolkit in order to analyse questions linguistically. Moreover, Sætre (2006) presented an approach used to find biological relevant information on protein interactions from the internet. This was developed using NLTK components that include data selection, tokenization, PoS tagging and stemming. The fact that NLTK (Loper and Bird, 2002) is an open source package implemented in Python is the main disadvantage in this tool. That is because Python is not powerful enough for most standard NLP tasks despite having most of the functionality needed to perform simple NLP tasks (Madnani, 2007).

2.1.2.3 GATE

The General Architecture for Text Engineering (GATE) (Bontcheva et al., 2004, Cunningham et al., 2009) architecture is implemented in Java and developed at the University of Sheffield for processing natural language is a publicly available system.(Gosling et al., 2005). It is an independent platform that helps in text processing, annotating, defining ontologies and using them for semantic annotation. The GATE architecture is developed in the IE component set called ANNIE (A Nearly-New Information Extraction). ANNIE contains a set of processing resources that apply algorithms for extracting information from unstructured text. Various major processing resources presented by the ANNIE plug-in are: English tokenizer, Gazetteer, sentence splitter, part-of-speech (PoS) tagger, named entity (NE) transducer, Java Annotations Pattern Engine (JAPE) transducer, and orthographic co-reference (Cunningham et al., 2009).

The advantages of using the GATE architecture have been showed by a number of different researches. An idea developed by Feilmayr et al. (2009) that a rule/ontology-based IE system can be used for analysing tourism websites and extracting structured data from accommodation webpages based on the use of the GATE system. An ontology-based IE system for the business domain based on the use of the standard and adapted processing resources from GATE was created by Saggion et al. (2007). Joshi et al. (2012), used the GATE for IE which shows the advantages of using social networking sites like twitter in the marketing domain. A domain specific NER for classifying named entities in Twitter posts from buyers and sellers was also created in this study. Accordingly, these posts (a collection of tweets) are analysed and processed by using the GATE components (e.g. English Tokenizer, Sentence Splitter, PoS-tagger, Gazetteer and NE transducer) so that data is acquired for farmers and merchants for giving them useful ideas. However, the main drawback of the GATE tool (Bontcheva et al., 2004, Cunningham et al., 2009) is that it can only be used as a full package, therefore, it is not suitable for our system as only some components of it are needed to be implemented in APELS.

2.1.2.4 Stanford CoreNLP

The Stanford CoreNLP is an open source toolkit that is composed of a set of NLP tools that are used for processing English texts (Manning et al., 2014). It can give the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize dates, times, and numeric quantities, mark up the structure of sentences in terms of phrases and word dependencies, indicate which noun phrases refer to the same entities, indicate sentiment, extract particular or open-class relations between mentions, etc.

Stanford CoreNLP is designed to be highly flexible and extensible. With a single option you can change which tools should be enabled and which should be disabled. It is very easy to apply a bunch of linguistic analysis tools to a piece of text.

Stanford CoreNLP integrates many of Stanford's NLP tools, including the part-of-speech (PoS) tagger, the named entity recognizer (NER), the parser, the coreference resolution system, sentiment analysis, bootstrapped pattern learning and the open information extraction tools. Moreover, an annotator pipeline can include additional custom or third-party annotators. CoreNLP's analyses provide the foundational building blocks for higher-level and domain-specific text understanding applications.

A variety of studies have actually used the Stanford CoreNLP tool. For instance, Ahmed et al. (2009) proposed the BioEve system to extract the Bio-Molecular events from Text. They used Stanford parser a simple tool to provide typed-dependency relationships between these words in form the dependency parse. In addition, Poria et al. (2014) developed an algorithm to exploit the relationship between words and obtained the semantic relationship between words based on dependency parsing. The Stanford Chunker component is used as the first step in the algorithm to chunk the input text. Moreover, Trupti and Deshmukh (2013) presented an approach for building an ontology from heterogeneous text documents using the Stanford CoreNLP parser. They parsed the text file using Stanford parser, which generates XML file that tags words as noun, verbs, adjectives, pronouns etc. Then OWL ontology, which contains classes and concepts, is generated by converting identified PoS words in XML file. Likewise, Siddharthan (2011) presented a system for text regeneration tasks such as text simplification, style modification or paraphrase. The system applied transformation rules specified in XML files, to a typed dependency representation obtained from the Stanford Parser. Furthermore, Pal et al. (2010) introduced a system to automatically classify the semantic relations between nominals. The system achieves its best performance using lexical features such as nominalization of WordNet and syntactic information such as dependency relations of Stanford Dependency Parser. Likewise, Kern et al. (2010) built a Word Sense Induction and Discrimination (WSID) system that exploits the syntactic and semantic features based on the results of a natural language parser component. They applied the Stanford Parser in order to provide a context-free phrase structure grammar representation and a list of grammatical relations (typed dependencies) of a given sentence. Moreover, Uryupina (2010) presented Corry – a system for co-reference resolution in English. He relied on the Stanford NLP toolkit for extracting named entities and parse trees for each sentence. The Corry system has shown the best performance level among four well-known co-reference resolution systems. Finally, Berend and Farkas (2010) introduced a novel approach which includes a set of features for the supervised learning in order to extract key phrases from scientific papers. They applied syntactic tagging using the Stanford parser on each sentence. Taken together, Stanford CoreNLP has been widely and effectively used tool for text processing, information extraction, therefore, it was implemented in our system for the same purpose.

2.1.3 Natural Language Processing Resources

The huge quantities of textual information and the need for NLP resources require the investigation into the utility of linguistic knowledge from available resources. The most common knowledge resources are described in the following subsections.

2.1.3.1 Machine-Readable Dictionaries

A Machine-readable dictionary is commonly used in NLP in order to provide helpful information, such as word meaning, grammatical word categories and relations (Manning and Schütze, 1999). Two specific examples of machine-readable dictionaries are in the form of

WordNet lexicon (Miller et al., 1990, Fellbaum, 1998) and the Longman Dictionary of Contemporary English (LDOCE) (Mayor, 2009).

2.1.3.2 Thesaurus

A thesaurus enables the definition of information that relates to the correlation between words, which produce synonyms (same meanings), as well as antonyms (opposing meanings) (Kilgarriff and Yallop, 2000). Moreover, a thesaurus also provides the overall word meaning. One example of a thesaurus that has been used in the NLP field on many occasions comes in the form of the Roget's International Thesaurus (Roget, 1911).

2.1.3.3 Ontologies

An ontology is composed of a set of concepts within a specific domain and the relationships between these concepts in order to extract knowledge (Gruber, 1995). In addition, ontologies are used in AI and NLP mainly in the domain of intelligent information integration (Seng and Kong, 2009), cooperative information systems (Ouksel and Sheth, 1999), information retrieval and extraction (Müller et al., 2004), and database management systems (Necib and Freytag, 2003, Snae and Brueckner, 2007).

Ontologies have a vital part in the process of information retrieval. The data belonging to a document has closely linked to the concepts of ontology. If the users agree upon the conceptualization of the ontologies, then the retrieval process would benefit better. The traditional information retrieval systems had to invent strategies that would semantically improve the queries and use similarity measures to match documents since the initiation of the ontology's (Saruladha, 2012). According to Lee et al. (1993), it is necessary to utilise domain knowledge and semantic similarity measures to efficiently match the queries and the documents in the information retrieval domain.

The content of data in traditional information retrieval techniques was made up using a lot of keywords. The issue with this format was that the data did not display semantic relations with the words and neither did it show the meaning of the words which caused for the retrieval data to be less accurate. The users found it hard to show their data need as the intent of the user is less understood by the system. So, to make the information retrieval system better, it is essential that external domain knowledge is required to augment the keyword based queries (Müller et al., 2004).

2.2 Information Retrieval and Information Extraction

Information Retrieval (IR) is the task of retrieving documents to help a user find the right answer (Goker and Davies, 2009) . IR presents a comparison of terms from the query, alongside index terms that are evident from the documents. Indeed, the most well-known IR applications in data retrieval are web search engines (e.g. Google). Google asks the user to provide a query with some words that are expected to occur in the relevant documents. Then the document list can be ordered according to how many times each search word occurs, how close the different search words are clustered in the document and so on.

Information Extraction (IE) is the task of automatically extracting knowledge from text. IE systems utilised NLP tasks to analyse unstructured text in order to extract information about pre-specified types of events, entities or relationships. IE has been used in some studies to identify names of proteins in biological documents (Fukuda et al., 1998), to extract medical information (Hahn et al., 2002) , and to ascertain information of businesses from webpages (Sung and Chang, 2004). Named Entity Recognition (NER) is known as the principle generic task in the process of IE, as well as the task of template elements and identifying co-references (Appelt, 1999, Moens, 2006). Furthermore, various IE systems are concerned with learning linguistic patterns or extracting rules automatically from training examples such as AutoSlog (Riloff, 1996) , RAPIER (Mooney, 1999), and CRYSTAL (Soderland et al., 1995).

Various IE systems are available for keywords extraction such as automatic indexing, text summarization, information retrieval, classification, clustering, filtering, topic detection and tracking, information visualization, report generation, and web searches (Bracewell et al., 2005). There are several methods for Automatic Keyword Extraction and these can be divided into four categories: statistical methods, machine learning methods, linguistic methods and other methods.

Statistical methods: These methods are based on statistical features derived from text such as term frequency, inverse document frequency and position of a keyword. For instance, Cohen (1995) developed a model to represent index terms from text. It does not utilise any stop list, stemmer, or any language and domain specific component, allowing for easy application in any language or domain with slight modification. The model uses n-gram counts, which results in a function similar and more general than a stemmer. Moreover, Herrera and Pury (2008) addressed the problem of finding and ranking the relevant words of a document by using statistical information referring to the spatial use of the words.

Machine learning methods: These methods employ the extracted keywords from training documents to train a model and then apply the model to find keywords from new documents. These methods include Naive Bayes and Support Vector Machine. For instance, Uzun (2005) applied a naive Bayesian classifier, utilizing features such as the TFxIDF score, distance of the word from the beginning of the text, paragraph and the sentence to identify keywords in the text. It has been assumed that keyword features are normally distributed and independent.

Linguistic methods: These methods utilise linguistic features of the words mainly sentences and documents. The linguistic methods include lexical analysis, syntactic analysis, discourse analysis and so on. For instance, Hulth (2003) utilised linguistic knowledge (i.e., PoS tags) to identify candidate sets of nouns or phrases. Potential PoS patterns were utilised to identify candidate phrases from the text. It was shown that, employing a PoS tag as a feature in candidate selection results in a considerable improvement in the key phrase extraction. Similarly (Yang et al., 2009, Hu and Wu, 2006) extracted the keywords by defining the noun phrase from the corpus. These keywords are defined based on their linguistic features (PoS). Moreover, Ercan and Cicekli (2007) used semantic word features based on lexical chains of words to determine important keywords in the document.

Other methods: Other methods about keyword extraction essentially combine the methods mentioned above or utilise some heuristic knowledge in the task of keyword extraction, such as their position, length, layout feature of words, HTML tags around the words, etc. For instance, Krulwich and Burkey (1996) employed heuristics for extracting key phrases from a corpus. The heuristics are syntactic ones, such as italicization, the presence of phrases in section headers and the use of acronyms. Additionally, the key phrase or keywords can be extracted from text by finding the relationships between them using dependencies. For instance, Poria et al. (2014) proposed a Concept Net-based semantic parser as a tool based on dependency between phrases, which has been used to extract concepts from heterogeneous texts with high accuracy. Similarly, Gelfand et al. (1998) proposed a method based on the Semantic Relation Graph to extract concepts from a whole document. They used the relationship between words, extracted on a lexical database to form concepts.

2.2.1 Information Retrieval Tasks

Informtion Retrieval (IR) deals with the problem of finding and presenting documents of unstructured data that satisfy an information need (query) from within collections of documents (Manning et al., 2008). A user information need, also referred to as query, must be 'translated' in order for an IR system to process it and retrieve information relevant to its topic. This 'translation' is usually made by extracting a set of keywords that summarise the description of an information need. The information can be presented in such a way that facilitates the user to find the documents that s/he is interested in.

IR is the process by which a collection of data is represented, stored, and searched for the purpose of knowledge discovery as a response to a user request (query) (Makris et al., 2009).

This process involves various stages initiated with representing data and ending with returning relevant information to the user. The intermediate stage includes filtering, searching, matching and ranking operations. The main objective of IR system is to find relevant information or a document that satisfies user information needs. To achieve this objective, IR systems usually implement the following two main processes (Indexing and Matching) as shown in Figure 2-1 (Croft, 1993). First, the information need (query) and the document will undergo query formulation or indexing respectively. The formulated query and the indexed document will be matched to find the similarities and the matched information will be retrieved.

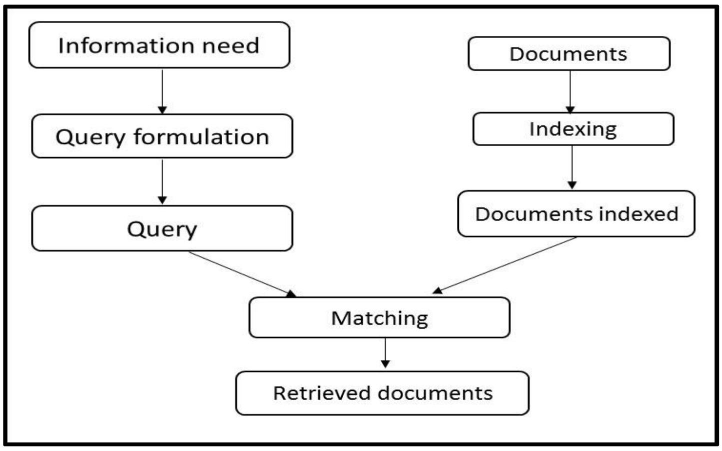

Figure 2-1 Information retrieval processes (Croft, 1993)

2.2.1.1 Indexing

The purpose of indexing is to optimize speed and performance in finding relevant documents for a search query. The index is built from the information stored with the data and the method by which the information is indexed. The process of indexing consists of the following set of stages:

1. Collecting the documents to be indexed.
2. Tokenize the documents and turn them into sets of tokens.
3. Eliminating all special characters from documents like "\ "," #", "$", "% "and stop words such as "a", "and"," or", "the", "of" that appears many times in documents. For humans stop words may be relevant in understanding the meaning of the sentence. However, when

it comes to text processing these words only make the process slow and take space. Furthermore, in text processing will be less precise since these words affect the weighting of words and their occurrence is taken into account in the similarity measure.

4. Stemming is an important step in many of the Information Retrieval (IR) and Natural Language Processing (NLP) tasks. The procedure of eliminating the prefixes or suffixes from words is called stemming. In terms of information retrieval, stemming is used to conflate word forms to avoid mismatches that may hinder the recalling process. For example, while looking for a document "How to write", if the user issues the query "writing" there will be no match with the title but if the query is stemmed, such that "writing" becomes "write", then the retrieval of this specific document will be triumphant. In many languages, stemming is imperative for the retrieval performance. Stemming is an absolute essential for the retrieval performance in many different languages like in Hebrew, stemming increases the number of documents retrieved by between 10 and 50 times (Krovetz, 1993). There are less drastic results for stemming in English but still it has proved to give betterment to retrieval process (Krovetz, 1993, Hull, 1996, Harman, 1991). Porter (1980) was the one who gave the most widely cited stemming algorithm. The Porter stemmer applies a set of rules to iteratively remove suffixes from a word until none of the rules apply.
5. In IR systems, the user's information need is represented by a query formulation. With a Web search engine for instance, the user is presented with a simple interface where he inputs some words stating his request for information. The IR system is expected to return an ordered list of documents, where the most relevant to the query should be on top. In order to match the query with the indexed terms of documents, the query is also represented in the same way as during the indexing stage.

6. The Vector Space Model, also called term vector model is a mathematical model for representing text documents as vectors of identifiers, like terms or tokens (Salton et al., 1975). The term depends on what is being compared to but are normally single words, keywords, phrases or sentences. The query $\boldsymbol{q}$ and document $\boldsymbol{d}$ are represented as an $\boldsymbol{m}$-dimensional vectors, where each dimension corresponds to a distinct term and $\boldsymbol{m}$ is the total number of terms used in the collection of documents and query.

$$d_i = (w_{1,i}, w_{2,i}, \dots, w_{t,i}) \quad (2.1)$$

$$q_i = (w_{1,q}, w_{2,q}, \dots, w_{n,q}) \quad (2.2)$$

The document vector $\boldsymbol{d_i}$ is represented by equation (2.1) where $\boldsymbol{w_i}$ is the weight that refers to the number of occurrences of the term $\boldsymbol{t}$ in the document $\boldsymbol{d}$. If a document $\boldsymbol{d}$ does not contain term the $\boldsymbol{t_i}$ then the weight $\boldsymbol{w_i}$ is set to zero.

2.2.1.2 Similarity Measures

Measuring similarities between words, sentences, paragraphs and documents is an important role in various applications such as information retrieval, document clustering, word-sense disambiguation, automatic essay scoring, short answer grading, machine translation and text summarization (Gomaa and Fahmy, 2013).

The comparison of the query against the document representations is called the matching or similarity process. The matching process usually results in a ranked list of documents in search of the information they need. Ranked retrieval will hopefully put the relevant documents towards the top of the ranked list, minimising the time the user has to invest in reading the documents.

Moreover, the similarity between the documents and the query is often calculated using a distance metric, such as the Jaccard similarity, or the cosine function of the angles between the two vectors. The documents are ranked on the basis of this similarity measure, and the list is returned to the end user. A similarity measure is a function which computes the degree of similarity between a pair of text objects. Furthermore, the similarity measures rely heavily on terms occurring in both the query and the document. If the query and document do not have any terms in common, then similarity score is very low, regardless of how topically related they actually are. The similarity measure functions described below are commonly used in information retrieval techniques.

2.2.1.2.1 Cosine Similarity

The Cosine similarity is one of the most popular similarity measures, which is applied to text documents in numerous information retrieval (Baeza-Yates and Ribeiro-Neto, 1999) and clustering applications (Larsen and Aone, 1999). It is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. If the angle $\emptyset$ between the document vectors is 0, then cosine $\emptyset$ is 1 and both documents are the same or identical. If the angle is 90° , this means that cos $\emptyset$ is 0 are no similarities between the two vectors. Figure 2-2 shows the angle in a two-dimensional space.

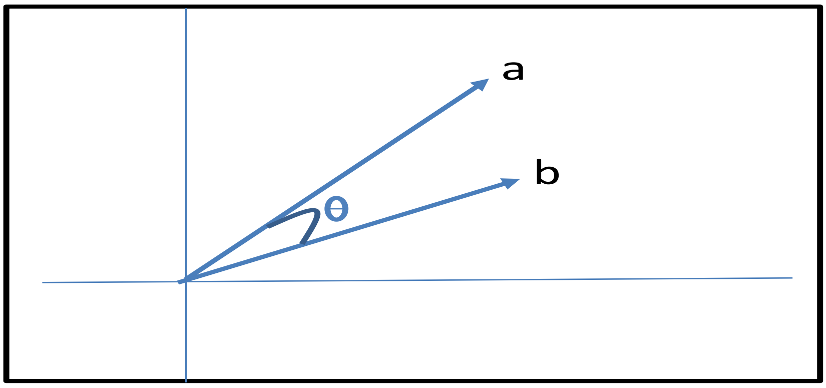

Figure 2-2 Angle between documents *a* and *b*

The standard way of measuring the similarity between two documents x_1 and x_2 is to compute the cosine similarity of their vector representations $\vec{V}(x_1)$ and $\vec{V}(x_2)$ and given by equation (2.3).

$$\text{Sim}\,(x_1, x_2) = \frac{\vec{V}(x_1)\,.\,\vec{V}(x_2)}{|\vec{V}(x_1)|\;|\vec{V}(x_2)|} \tag{2.3}$$

where $\vec{V}(x_1).\vec{V}(x_2)$ represents the dot product (also known as the inner product), which is a simple multiplication of each component from the vectors (x_1) and (x_2) and added together, while the denominator is a multiplication of the magnitude (Euclidean length) of the vector (x_1) with the magnitude of the vector (x_2).

To illustrate this process, suppose that A and B are vectors such that

A = {1, 1, 1}

B = {2, 2, 1}

The dot Product: A·B = ($x_1 \cdot x_2$) + ($y_1 \cdot y_2$) + ($z_1 \cdot z_2$) = (1·2) + (1·2) + (1·1) = 4

The magnitude length of A = $\sqrt[2]{{x_1}^2 + {y_1}^2 + {z_1}^2}$ = $\sqrt[2]{1^2 + 1^2 + 1^2}$ = 1.73

The magnitude length of B= $\sqrt[2]{{x_2}^2 + {y_2}^2 + {z_2}^2}$ = $\sqrt[2]{2^2 + 2^2 + 1^2}$ = 3

|A|·|B| = (1.73) · (3) =5.19

sim = cosine (A, B) = $\frac{A\,.\,\mathrm{B}}{|A|\ |B|}$ = $\frac{4}{5.19}$ =0.7707

According to the Cosine Similarity A and B are 77.07 % similar.

2.2.1.2.2 The Jaccrad Coefficient

The Jaccard coefficient is another similarity measure that measures similarities between finite sample sets (Jaccard, 1901), and is defined as the size of the intersection divided by the size of the union of the sample sets. For example, as shown in Figure 2-3, the two sets X and Y have four common elements, hence their intersection is 4, and there are 9 elements in their union. So, Sim (X, Y) = 4/9.

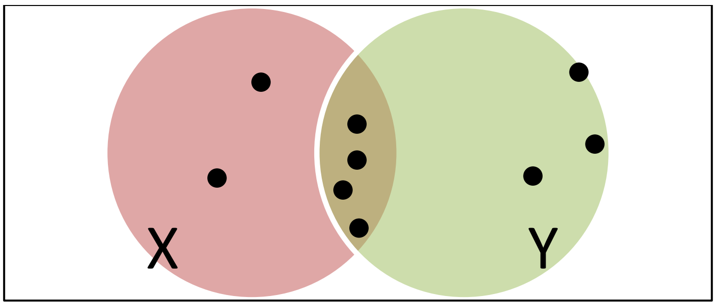

Figure 2-3 Two sets with Jaccard similarity 4/9

For text documents, the Jaccard coefficient is used to compare the total weight of shared items with the total weight of terms that are present in either of the two documents and not shared terms. For example, suppose that we have a set of documents D = $\{d_1, d_2, d_3, \ldots, d_n\}$. Each document d_i can be included in a set t_i, where t_i is as a set of all the terms.

Then d_1 is represented as $d_1 = \{t_1, t_2, t_3, \ldots\ t_n\}$. The similarity of any two documents d_i and d_j can be computed using the Jaccard coefficient as J $(d_i, d_j) = \frac{|d_i \cap d_j|}{|d_i \cup d_i|}$

According to Jaccard coefficient, the similarity measure ranges between 0 and 1. If the value is 0 then the documents are dissimilar and a value 1 means that the documents are identical. The value also represents the probability of similarity between the documents.

2.2.1.2.3 Euclidean Distance

Euclidean distance is a standard metric for geometrical problems. It is the ordinary distance between two points and can be easily measured with a ruler in a two- or three-dimensional space. Euclidean distance is widely used in clustering problems, including clustering text (Anderberg, 1973, Jiawei and Kamber, 2001). The Euclidean distance of the two documents A and B is defined by equation (2.4).

$$\text{D (A, B)} = \sqrt{\sum_i^n (A_i, B_i)^2} \tag{2.4}$$

2.2.1.2.4 The Dice Coefficient

The dice coefficient is another similarity measure that measures similarities between two samples (Dice,1945) , which is commonly used in information retrieval application (Lin, 1998, Duarte et al., 1999). The measure similarity of two documents A and B is performed by normalising the size of their intersection over the average of their sizes. The Dice's coefficient of the two documents A and B is defined by equation (2.5).

$$\text{Dice's coefficient similarity (A, B)} = \frac{2\,|A \cap B|}{|A| + |B|} \tag{2.5}$$

Although Dice coefficient is shown to be identical to Jaccard coefficient, it has shown in many studies to have better efficiency in identifying correlation between the vectors (Thada and Jaglan, 2013, Duarte et al., 1999, Anuar, N. and Sultan,2010).

2.3 Summary

In order to identify the appropriate tools and techniques to be used in our system, it was important to review the common and available NLP tasks and tools. After this detailed search and review, the following decisions were made.

For text processing, linguistic structure analysis and the access to large corpora collections, this chapter has shown that there are many open source tools for NLP available for semantic annotation of textual documents which could be used such as OpenNLP (Baldridge, 2005), NLTK (Loper and Bird, 2002), GATE (Bontcheva et al., 2004, Cunningham et al., 2009), and Stanford CoreNLP (Manning et al., 2014). For the purpose of this study, Stanford CoreNLP has been selected for many reasons. Firstly, it has been applied effectively and extensively for processing texts, extracting useful information, and generating text annotations (Ahmed et al., 2009, Pal et al., 2010, Uryupina, 2010, Poria et al., 2014). Moreover, among the available state-of-the-art NLP technologies, Stanford CoreNLP is one of the important NLP tools due to its integrated toolkit with a good range of grammatical analysis tools, support for a number of major (human) languages as well as its design which is highly flexible and extensible. In addition, Stanford CoreNLP tool involves a set of tasks that can be implemented to perform on unstructured texts converting it into structured text in order to facilitate the extraction process; in this case, the flexibility of Stanford CoreNLP is obvious as these tasks can be applied individually or combined together. Interestingly, Stanford CoreNLP tool can also read various forms of plain text input and generate different data format such as XML, JSON, and CoNLL. For example, XML files give linguistic features of each word, which are useful for automatic keyword and key phrases extraction (Siddharthan, 2011).

Secondly, in this research, an ontology, which is composed of a set of concepts within a specific domain and the relationships between these concepts, has been used as a knowledge resource for extracting the required domain knowledge from the Web. In addition, reviewing the literature suggested that ontologies play a crucial role in formation retrieval. Traditionally, the information retrieval process is based on the keywords description of the information, which neither reflects the semantic relationships among the words nor their meaning. Hence the retrieved information is less precise and relevant making the ontology a valid substitute for more effective knowledge extraction process (Müller et al., 2004, Saruladha, 2012).

Finally, this chapter has covered the importance of using similarity measures such as Cosine similarity, Jaccrad Coefficient, Euclidean Distance, and Dice Coefficient to find out which one suits the purpose of our system. Although cosine is a very common similarity measure in information retrieval due to its good performance, it is not suitable for this system because its computational complexity is very high, when applied to very large data sets (Rawashdeh and Ralescu, 2015). Furthermore, it relies on measuring the metrics and also tends to propose less frequent words as similar suggesting frequency bias by cosine (Erk and Padó, 2008). While Jaccard and Dice coefficient are very close similarity measures, Dice was chosen in this study because of its ease of use and its superiority in finding the best fit to the query and document, therefore it was selected to be used to find the intersection between ontology domain concepts and entities on the Web.

Chapter 3 : A Review of the Development of E-learning Systems

3.1 Introduction

This chapter presents an overview of learning theories, which are crucial for developing personalised E-learning applications. It also reviews the background and platforms for E-learning systems, and how they developed into adaptive E-learning systems. Furthermore, it presents some common learning styles and discusses the suitable learning style for the APELS system. Finally, the chapter reviews some related works by outlining different methods that are used in personalised E-learning systems and their evaluation.

3.2 Learning Theories

Before introducing any addition to the field of education, it is essential to understand what is learning. Learning is a process that enables cognitive, emotional, and environmental influences to be brought together, alongside experiences that demonstrate acquisition, enhancement, or changes in an individual's skills, knowledge, values, and global views (Illeris, 2009). Therefore, learning theories are required to deliver an explanation into how this process is undertaken. This section reviews the three important learning theories that have had a major impact in learning and personalised E-Learning applications and these are behaviourism, cognitivism, and constructivism.

3.2.1 Behaviourism

Behaviourism or 'behavioural theory' focuses on human behaviour, although it does not evaluate those mental processes that are performed on the mind, as they are deemed inaccessible (Graham and Bechtel, 1998). Behavioural theorists have stated that learning is when the original behaviour is acquired. Hence, a learner is passive when acquiring knowledge, and thus, the correct behaviour needs to be reinforced by a teacher. Indeed, it is shown that behaviour is malleable and observable alterations in the behaviour define behaviourism, which results in learning being a stimuli and a process of response (Chen, 2009). Consequently, behavioural theorists determine that adaptive E-learning systems that assist adaptation need to create stimuli that cause learners' behaviour to result in successful learning.

The main point of this theory is the reward or punishment of a new behaviour for both animal and human which means that if someone is rewarded for a particular behaviour, s/he is encouraged to act in a similar way in similar situations. On the contrary, if s/he is punished, s/he is less likely to act similarly. The behavioural theorists believe that knowledge must be displayed in a predefined order by the teachers in the case of traditional learning and the systems in the case of E-learning.

3.2.2 Cognitivism

The cognitivism theory was created to make mental processes as a primary object of learning as opposed to behaviourism theory, which only views the learning process in a passive way. Learning has been deemed as an internal process and memory is an active processor by cognitivists. The ability of people to learn depends on their previously acquired study and the amount of mental effort expended during the learning process. They view knowledge as symbolic mental constructions. Hence, as might be seen, the learning process changes this previously acquired information which represents symbolic metal constructions (Ausubel, 1960, Craik and Lockhart, 1972).

Cognitive scientists state that there exists external reality in environment. When a person is paying attention to detail, he can sense many various things in the environment which they can acquire through their senses. This sensed information can then be converted into pre-existing cognitive structure through integration which is then converted to knowledge and then kept safe in the memory. The information in the memory can be kept for retrieval when remembered in the future.

Teachers play a vital role in increasing the learner's attention and motivation and are also responsible for managing the content of learning activities to develop conceptual knowledge even though it is the general belief of the cognitivists that a learner has an important role in the learning process. Both the cognitivists and behaviourists have different views to learning even though they have a similar perspective on knowledge that it is given and is complete and whole. Additionally, from the E-learning perspective both cognitivists and behaviourists agree that computers should present information to be learnt and learners practice until they understand it.

On the other hand, cognitivists are also taken to be active mental processes of the learner. The learning process, according to Piaget (1985), is very repetitive since every new data is merged with the previous knowledge of the learner and the previous knowledge is varied with reference to the new ones gathered. So, to implement cognitivism into instruction in adaptive E-learning systems, cognitivist instructional designers must take into account the previous information gathered before acquiring the new. They should know that all learners do not have the same prior knowledge or learn in the same way. The lesson must be broken into segments according to its difficulty level according to the learners previously acquired information (Piaget, 1985).

3.2.3 Constructivism

Constructivism demonstrates that knowledge needs to be constructed instead of being transmitted, although the theorists agree with cognitivists that attention, encoding and retrieval of knowledge are the same process. Constructivism shows that an individual's previous knowledge is a vital factor in learning, as it can prove to be both positive and negative in the development of new concepts of learning. As new concepts are implemented, they need to be correlated together with the structural knowledge of a learner, and how this advances depend on the manner that the new concepts link with structural knowledge, which was developed previously in the mind of the learner (Reynolds and Muijs, 2005). Hence, new concepts are developed by the learner through a connection of new and past knowledge, which means that well-organised previous knowledge can help a learner to develop new concepts in an easy and rapid manner (Ausubel et al., 1978).

Maintaining a constructivist context, the teacher is required, as their responsibility to establish scenarios for students where they are motivated to create the desired mental constructions rather than using material to simply share the knowledge. It is also necessary to adopt a suitable format of information for the students and this format would be in accordance with their understanding abilities and knowledge. Keeping in mind the adaptive E-learning context, the information brought forward presently must be related to the information that has been stated earlier (Henze et al., 1999). Through this information, the learner should be able to integrate the knowledge and real world capabilities. Concepts can be thoroughly explained and their importance stated through information explanation of other spheres. Learning effectiveness can also be enhanced through inquiring-oriented learning, problem solving applied for restructuring teaching and identification of the mathematical thinking of the learner (Glasersfeld, 1995). It also stressed upon by the constructivists, that a student centred approach must be followed rather than the traditional teacher centred one. Usually, the learners perform much better if they are part of the process and not just attaining information from the computer or teacher. Hence, the framework should be such that the learners are able to construct their own knowledge representations for learning as well as problem solving. They should be allowed to integrate new concepts within the present knowledge system. The teachers must not be non-existent but their courses should be designed in a manner that learning is focused and effective through activities.

3.2.4 Conclusion and Discussion

The three important learning theories discussed above impact on the design and use of personalised learning applications by providing insights into how individuals may learn best using such systems. Behaviourism is based on behavioural changes. It focuses on the repetition of a new behavioural pattern until that pattern becomes permanent. Cognitivism, on the other hand, is based on the thought process behind the behaviour. Normally, changes in behaviour are observed, but only as an indicator of what is going on in the learner's mental model. Finally, constructivism is based on the premise that human beings construct their own view of the world based on individual experiences. This view focuses on preparing the learner to solve problems in a variety of situations.

Constructivists stress that people learn more with a teacher than from a teacher. Reeves (1998) also agree that students learn more with a computer than from a computer. Therefore, computers may simply be used as a tool to empower students and instructors. Constructivism is concerned with learners' creation of meaning and connection of new concepts to existing knowledge. Both of these processes involve a large degree of autonomy and initiative.

Reward or punishment online seems to be the very in-frequently practiced, because it stems from the behaviourist approach. However, many of these ideas have been incorporated by cognitivism. Conlan (2005) notes that cognitivism is concerned not only with a student's observable behaviour, but also with his non-observable mental processes. The approach taken by constructivists is different from that taken by others, because it contends that learners construct their own view of information. This implies that constructivism may be better suited to self-motivated learners. A balance between cognitivism and constructivism is achieved as a learner moves from being a novice (prescriptive learning experience) to becoming an expert (more control over learning) (Conlan, 2005).

3.3 E-learning

E-learning is a modality of learning that uses Information and Communication Technologies (ICTs) and advanced digital media (Wentling et al., 2000). It offers education to those who cannot access face to face learning. Uhomoibhi (2006) added that E-learning offers a platform for instant response and promotes the interaction of learners with instructors and other online users through discussion boards, chat rooms, e-mail, immediate messaging which are all efficient ways of developing interaction among E-learners. The interactivity offered by E-learning helps in maintaining concentration through quizzes, games etc. E-learning can also support large numbers of learners that can be administered by an online system (Uhomoibhi, 2006) .

Additionally, with regards to traditional learning, the content available in an E-learning environment is more adaptable to personal needs. For example, students have more authority on the learning process, they tend to understand the content quicker than the learning held in classrooms, and the facility of altering the content according to personal aptness and specification is given to the instructor and learner (Kirsh, 2002). Moreover, E-learning decreases the leading and administering ability of instructors and allows the usage of already formulated content of the course and provides time to the instructors for conducting research and learns about the significant goals of learning (Cantoni et al., 2004). However, E-learning has some disadvantages; as per the concepts of education, learners, who learn through these techniques can feel isolation and separation from classroom and instructors, and they might feel the deficiency of the direct human contact, which in turn adversely affects the learning of the students (Bleimann, 2004). In addition, the E-learning process lacks in providing learning about the practical side of education, which is essential in some courses such as those involving laboratory work, therefore, the teaching of such kind of material on the Internet is not possible. Furthermore, the E-learner requires more self-discipline and E-learning needs reliability from the learner as it lacks confined learning routine and process. Additionally, the cost of producing E-learning content is about 100 times the cost of face to face lectures (Bleimann, 2004).

Since the era of E-learning started in the middle of 1990, many platforms have been developed to overcome some of E-learnings disadvantages and the support the learning process. Many educational institutions across the world have focused on bringing E-learning to the individual user, by the use of several different commercial on-line educational mediums. For example, modern E-learning is dominated by E-learning platforms or Learning Management Systems (LMS) such as Blackboard (Stephen Gilfus, 1997), which comprises tools that support students and teachers within an E-learning environment. These tools can be used to provide different ways of on-line education, and address several learning contexts, ranging from the conventional, classroom delivery to off-line, distance learning and on-line learning.

Modular Object-Oriented Dynamic Learning Environment (Moodle) is another E-learning platform that is commonly used in many educational institutions around the world. Moodle is an open source software, offering course management for learning resources. It also integrates communication tools, supports timed quizzes, manages assignment submissions etc. (Dougiamas, 2002). Furthermore, The Sakai CLE (Collaboration and Learning Environment) was developed by a community of academic institutions, commercial organisations and individuals. Like Moodle, it is an open source software platform, used for teaching and learning; portfolio support; ad-hoc collaboration and research (Farmer and Dolphin, 2005). Although LMSs provide a variety of features to support teachers in creating, administering, and managing online courses, they typically do not consider individual differences of learners and treat all learners equally regardless of their personal needs and characteristics.

Importantly, learning is carried out traditionally as well as through advanced technologies and the learners play a vital role during the process. All individual learners have their specific characteristics like motivation, learning styles, cognitive abilities, prior knowledge and many more. The learning process is affected by these differences which are why some learners may find it easy to attain specific knowledge and some may find it complex (Grabowski and Jonassen, 1993). Additionally, adapting the content according to the learner's progress/needs is another feature not provided by the available LMS systems, which is a key feature that will be addressed in our system.

3.4 Adaptive Systems

In the context of E-learning systems, the concept of “adaptable” refers to the property of changing system parameters where the user is able to change the behaviour of the system. In other words, the user is able to modify the system in specified ways to fit his/her needs, whereas, the term “adaptive” means the automatic tailoring of the system to the user. The system adapts to the users automatically based on the system’s assumptions about the users’ needs (Oppermann and Rasher, 1997).

An adaptive system adapts itself to various circumstances, which are mainly based on user’s goals and preferences. First, these properties of the user are stored in a user model. The user model is held by the system and provides information about the user such as his/her knowledge, goals, preferences, etc., therefore, it gives the possibility to distinguish between users and provides the system with the ability to tailor its reaction depending on the model of the user (Brusilovsky and Maybury, 2002). In the context of E-learning, Adaptive Systems are more specialized and focus on the adaptation of learning content and the presentation of this content. According to Gütl et al. (2004), an adaptive system focuses on how the knowledge is learned by the student and pays attention to learning activities, cognitive structures and the context of the learning material. In Figure 3-1, the structure of an adaptive system according to Brusilovsky and Maybury (2002) is shown. The system intervenes at three stages during the process of adaptation. It controls the process of collecting data about the user, the process of building up the user model (user modelling) and during the adaptation process. The adaptive feature of E-learning systems has developed swiftly over the past few decades, and intelligent tutoring systems (ITS) and adaptive hypermedia are among the key methods in the field of education, which will be discussed here in details.

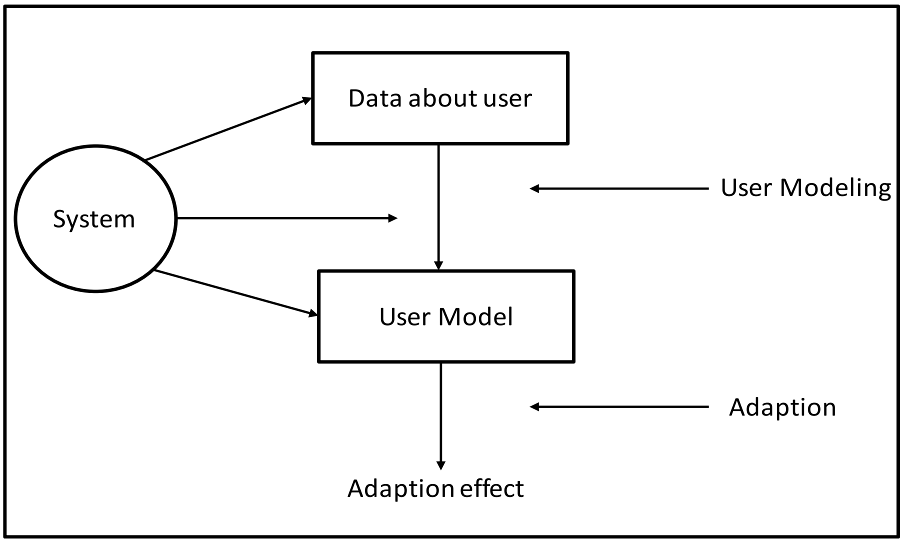

Figure 3-1 The structure of an adaptive system (Brusilovsky and Maybury, 2002)

3.5 Intelligent Tutoring Systems

Intelligent Tutoring Systems ITSs are adaptive instructional systems applying artificial intelligence (AI) techniques. The objective of ITS is to provide automatic one-to-one instruction in a cost-effective manner (Shute and Psotka, 1996). There are learning content components part of the ITS along with instructional and teaching strategies that would help to recognize how much knowledge a student attains. An *expertise module, user interface module, student-modelling module and the tutoring module* is present within the ITS that arranges these components (see Figure 3-2) (Brusilovskiy, 1994). With the help of the *expertise module*, the student performance can be assessed and instructional content is presented. The present knowledge of the user is stated by the *student-modelling module* along with estimating their conceptions and reasoning strategies. The ITS makes use of this valuable information to indicate how the progress of the teaching process should be made. The instructional material selection is also proposed after assessing this information within the *tutoring module*. With the help of the information contained in the *student-modelling module,* it is possible to state how and when the material should be presented. The interaction between the system and the student is carried out through the communication component which is the *user interface module*.

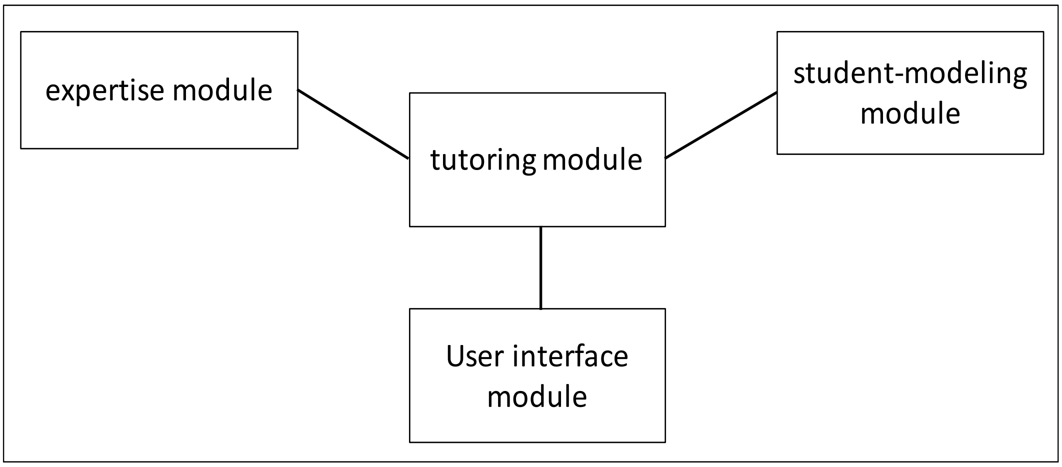

Figure 3-2 Components of an ITS (Brusilovskiy, 1994)

The micro-adaptive approach has been applied by the ITS since it integrates two processes which are the prescriptive and diagnostic processes. In the first, diagnostic process, the learner aptitude or beforehand knowledge is assessed to characterize him/her and carry out the activity. The interaction takes place as part of the second step where learning content is adapted to the aptitude of the learner in order to optimize the task (Modritscher, 2004).

A variety of AI techniques are used to represent the learning and teaching process. For instance, some rules of the ITS system include the expertise specific to a topic. The ITS attains the ability to create issues on the fly, implement rules which would help resolve the issue, analyse the understanding of the learner by software reasoning comparison and present the participants with software solutions. However, a vital issue of ITS is creating an expert system that would comprehensively cover the subject material. The Adaptive Hypermedia Systems (AHSs) were developed after inspiration from the ITS and their objective was to integrate educational hypermedia with ITS (Brusilovskiy, 1994).

The development of AHSs can be traced back to the early 1990s. AHSs are environments which supply an experience for users composed of free navigation in a large hyperspace of information. However, a problem with these systems is that users can easily get disorientated when looking for information and lose their sense of location due to cognitive overload. AHSs are based mainly on two techniques, which are explained thoroughly in the following sections.

3.6 Adaptive Hypermedia System

Generic learners have made use of the E-learning systems but they are unsuccessful in meeting the requirements of the individual students. These learners are not provided with customized or personalised resources or learning plans. The “one- size- fits-all” traditional learning approach has been replaced by the Adaptive hypermedia systems (AHSs). The objective is to attain knowledge related to a specific learner and create a learning path based on this information.

With the help of the adaptive system, information related to the users is collected and the system behaviour is changed accordingly. For instance, the presentation given by an adaptive educational hypermedia system would be based on the specific students’ subject knowledge and links on how he may further progress. It is quite possible that mistakes would be made by the system to assess the preferences of the user. Hence, it is necessary for the user to have control upon system adaptability (Fink et al., 1996). There are two techniques used by the AHS which are explained in the following subsections.

3.7 Methods of Adaptation in AHS

Adaptation is performed by the adaptive system using two techniques which are adaptive presentation and adaptive navigation support (Brusilovsky, 2001). Different content portions for a variety of users can be displayed using the adaptive presentation. The links presented are adapted for the user in adaptive navigation. The learning characteristics of the user are mentioned in the user model which is used by the adaptive presentation to customize the content. Static pages are not used within an adaptive presentation system but the information pieces are accumulated based on the request of the user to extract pages according to each individual user’s preference. For instance, the additional explanation is provided to novice users and detailed or through information provided to advanced users (Brusilovsky, 1998).

Adaptive navigation support techniques guide users to find the most relevant information by adapting link presentation to users' needs (Brusilovsky, 2007). Adaptive link annotation, adaptive link sorting, adaptive link hiding and direct guidance are the commonly used to adapt presentation of links techniques. The link presentation and structure within the learning environment is changed through the technique considering it as the vital navigational method. This can be compared to the adaptive presentation where the content is changed according to learner's requirements.

3.8 Learning Style

Students often learn better when the process of learning is adapted to their own preferred approach, which is referred to as their "learning style". Learning styles have been described by Keefe (1979) as the combination of distinctive cognitive, physiological, and affective factors that tend to be comparatively stable indicators of the way a learner identifies, communicates, and reacts to the learning environment. This means that a learning style is a way through which a person prefers to identify and understand the information when interacting with the learning environment. It has been observed through research on learning styles that determining and taking into account the learning style of the learner when preparing instructions could enhance the effectiveness of the learning. Thus, it is expected that when learning material is adapted to the learner's learning style, it will lead to a strong personalised mechanism. Such personalisation could lead to a better understanding of learning materials with a shorter learning duration (Dwyer, 1998).

During this research, it has been found that a number of learning style models are being used for the purpose of personalisation and adaption within E-learning. Examples of learning styles systems include: The index of learning styles Felder and Silverman (1988), Learning Styles Inventory (Kolb, 1984), The Manual of Learning Styles (Honey and Mumford, 1992), VARK Learning Style (Fleming, 2001) and Teaching Students through Their Individual Learning

Styles (Dunn and Dunn, 1978). A detailed overview of the most commonly known learning style models that are used in adaptive E-learning systems is presented in the following sections.

3.8.1 Kolb Learning Style

Learning is a process through which knowledge is created by ways of changing experiences (Kolb, 1984). Learning Style Inventory (LSI) is proposed by Kolb who stated that knowledge is derived from the integration of understanding of experiences and transforming it. LSI has four sequential stages. First concrete experiences provide a basis for observation. In the next step the learner reflects on these observations and builds a theory of what this information might mean. Next, the learner creates abstract concepts based on their hypothesis. Finally, the implications of these concepts are tested in new situations. Then the process cycles back to the first stage of the experiential process. A learner must complete the cycle of learning through all parts to fully understand the topic. Four kinds of learning styles were determined that defined the concepts of conceptualisations, concrete experiences, reflective observations, and active experimentation, which functioned to distinguish the most beneficial style for each learner. These learning styles are defined as follows:

i. Converging (Abstract/Active)

This type of learning style is one in which learners like to experiment with new ideas and to work with practical applications. They prefer technical tasks and will use learning for problem-solving and decision-making. They have abilities in the areas of abstract conceptualisation and active experimentation. Learners with a converging learning style have abilities to find practical uses for theories and ideas. They like to work actively on well-defined tasks. They prefer to deal with technical tasks and problems rather than with social and interpersonal discussions. They like interactive instruction, not passive.

ii. Diverging (Concrete /Reflective)

This style type comprises of imaginative people. They have abilities in the areas of concrete experience and reflective observation. They are best in generating new ideas such as brainstorming. People with the diverging style prefer to work in groups. They are best in viewing things from different perspectives. They prefer to use imagination to solve problems. They also tend to be strong in the arts such as artists and musicians.

iii. Assimilating (Abstract/Reflective)

This style type is inclined towards reflective observation and abstract conceptualisation. They need clear explanation rather than practical applications. They prefer inductive reasoning and logical approach for organising a wide range of information. They are more interested in abstract ideas and creating theoretical models and less focussed on interaction with others. People, who assimilate into new surroundings, are more concerned with abstract concepts and ideas rather than practical applications.

iv. Accommodating (Concrete/ Active)

This style type includes learners who prefer to do things, prepare their plans, and be part of new experiences. They also prefer to work in groups to complete tasks. People with this learning style rely on other people's analysis rather than perform their own analysis. They like to work in technical fields or get jobs requiring action such as sales and marketing.

LSI is the assessment tool for Kolb's theory and comprises of 12 sentences which define learning. The learner is supposed to rank these definitions in order to best determine how they best learn.

3.8.2 Felder-Silverman Learning Style

According to Felder and Silverman (1988), learners have various methods of obtaining and processing information which shows that they learn in various ways. The learner's style of learning is rated on a scale of four dimensions in the Felder-Silverman learning style model. A questionnaire was developed in 1991 by Felder and Soloman (1991) in order to identify the learning style of every learner. There are 44 questions in the questionnaire which categorise each learning style as per the dimensions mentioned below.

i. **Sensing – intuitive**

Sensing and intuition are two ways in which people tend to perceive the world. Sensor learners have good memory and thus remember facts and are also careful people, but could be a little slow. They prefer data, facts, experimentation, and solving problems. Intuitive learners have the capability of easily understanding new content and could be quick but careless. They prefer theories, principles, and determining possibilities.

ii. **Visual-verbal**

Visual learners have good visionary skills. They could forget something that was said to them but remember what they had seen such as pictures, sights, symbols, diagrams, films, and demonstrations. Verbal learners are more prone towards listening skills as they remember more of what was said to them. They prefer words, sounds, and hearing. They understand more through discussions; prefer verbal explanation over visual description. They learn better by explaining to others.

iii. **Active-reflective**

Active learners like to try out something to see if it is effective. They are not able to learn well in situations where they have to be passive. Active learners work better in groups. Reflective learners first think through the matter before taking action. They are not able to learn well in situations where they are not able to first go through the information being provided. Reflective learners work better on individual basis.

iv. **Sequential-global**

When solving problems, sequential learners use linear reasoning processes. They like to work step-by-step through the material in order to logically proceed towards a solution. They learn the most when the material is given to them in a consistent way in accordance with difficulty and complexity levels. Global learners can take intuitive decisions and might not be able to explain how they reached certain solutions. They like to focus on the bigger picture and work intuitively. At times, they perform better by moving directly to more difficult and complex material.

3.8.3 VARK Learning Style

An inventory was developed by Fleming (2001) which used VARK learning styles to facilitate students to understand more about what their learning preferences are. The VARK learning model does not have any impact on the structure or sequence of the learning material. It only affects the form and nature of the delivered learning material. VARK stands for Visual, Aural, Read/write, and Kinaesthetic. There are 16 questions in the questionnaire suggested by Fleming (see Appendix C) regarding the way the learner likes to learn. The learner will then be given their learning style in accordance with their answers. The four different preferences of the VARK learning style are:

i. **Visual (V)**

Visual learners are those who prefer diagrams, flow charts, graphs, labelled diagrams and all the symbolic arrows, allow learners to interpret data in a logical manner. Learners who prefer this type of learning see the information presented in a visual rather than in written form. Moreover, Visual learners prefer to use images, pictures and maps to organise information.

ii. **Auditory (A)**

Auditory learners are those who prefer to use audio recorders, explain new concepts to others, discuss different topics, be part of discussion groups and attend lectures. Learners with this particular learning style remember most things they are told. Auditory individuals benefit from background music when they work. They are also able to debate and discuss with one another in a group setting.

iii. **Read/Write (R)**

These are the learners who prefer using reports, essays, textbooks, hand-outs, lists, webpages, and manuals, etc. They prefer this modality best understand information displayed as words. Learners with this type of learning style prefer strongly text-based learning materials. Likewise, they like to read widely and write the material learned in a structured form.

iv. **Kinaesthetic (K)**

These are the learners who prefer the approach of trial and error, like field trips, doing things, using their senses, and employing hands on approaches, etc. Learners with this type of learning style tend to understand the information via demonstrations, simulations.

3.8.4 Learning Styles Assessment for the APELS System

Peter et al. (2008) undertook a study that describes commonly used learning styles and how they are currently being used within the area of adaptive E-learning. They evaluated a suitable learning styles using criteria proposed by Sampson and Karagiannidis (2002) in order to select a suitable learning methodology for the iLearn E-learning platform. Sampson and Karagiannidis (2002) described the evaluation criterion for learning styles in measurability, as well as their time effectiveness, and descriptiveness. This criterion was also used in this research to choose the most appropriate learning style so that adaptability and personalisation could be offered to the learner.

i. **Measurability**

According to Sampson and Karagiannidis (2002), measurability is the ability of learning style model to measure how a learner belongs to a particular category, specifically, to evaluate how will this be appropriate for a personalised learning environment. There are adequate corresponding questionnaire tools for all the learning styles that were analysed for this research. The number of questions ranges from 12 (Kolb) questions to 44 questions (Felder & Silverman).

ii. **Time Effectiveness**

The learning styles were evaluated based on the time it would take the learner to fill in the questionnaire and the pertinence of those questions with the specific personalisation that is required. The results for this category are defined as high for the most time effective method to Low as the least time effective. Specifically, the Learning Styles were assessed on how long it was felt the learner would have to take to undertake the questionnaire and the relevance of the questions to the particular personalisation required.

iii. **Descriptiveness**

This category describes how learners are classified and the way in which these categories can be adjusted for the particular learner in the system. For instance, learner styles are categorised as Visual/Verbal, Sensing/Intuitive, Active/Reflective, and Global/Sequential as per the Felder Silverman's tool categories

A summary of the results of this evaluation is presented in Table 3-1. The learning styles with the least amount of questions were defined by the evaluation, which included Kolb at only 12 questions, followed by Fleming's VARK at a total of 16 questions. These questionnaires would therefore take the least amount of time for the learner to fill. With regard to the pertinence of the questions, VARK questions were observed to be the most pertinent and concise.

Table 3-1 Learning styles assessment (Peter et al., 2008)

Learning Style	Measurability	Time Effectiveness	Descriptiveness
Kolb's Learning Style Inventory	12 Questions	Low	Converging, Diverging, Assimilating, Accommodation
Felder-Silverman's Leaning Style	44 Questions	Medium	Sensing-intuitive, Visual-verbal, Active-reflective, Sequential-global
Fleming's VARK	16 Questions	High	Visual, Aural, Read/Write, Kinaesthetic

With regard to personalisation, it was stated that questions posed in Kolb might not be pertinent. This was specifically for questions that attempt to understand how the learner prefers to learn. Keeping this in mind, it leads to low time effectiveness for Kolb because some more questions would have to be included for this learning style questionnaire to achieve the required outcomes. Felder & Silverman (1998) included pertinent and concise questions, but they achieved a lower ranking because of the high number of questions included in the questionnaire (44 questions).

A result of evaluation was that VARK learning style will be the most suitable learning style methodology for use in our system for offering personalisation and adaptability for the learner. This tool is quite relevant and take least amount of time for the learner to fill the questionnaire. It has also all the essential questions to be used to determine the learning style of the user. Moreover, the tool helps in clearly determining and mapping the kind of learning materials that are required.

3.9 Related Work

Personalised E- learning systems have attracted a great interest in the area of technology-based education, where their main aim is to offer to each individual learner the content that suits her/his learning style, background and needs. Personalised E-learning systems have been developed in order to include a variety of techniques which show contrasting forms of teaching.

Intelligent tutoring systems and adaptive hypermedia systems are among the earliest E-learning systems. The most significant aim of an Intelligent Tutoring System (ITS) is to support adaptive learning by using knowledge about the domain, the learner and teaching strategies (Brusilovsky, 1998). For instance, AutoTutor is an intelligent tutoring system developed to help students learn about physics and computer literacy (Cai et al., 2015), at the Institute for Intelligent Systems at the University of Memphis. AutoTutor helps students learn by holding a conversation in natural language. It also tracks the cognition and emotions of the student and responds in a manner that adapts to the student. A system called InterBook, which is a web-based education program that uses one specific model of a learner's knowledge and applies it in order to provide adaptive guidance, navigation, support and help for the user, was originally proposed by Eklund and Brusilovsky (1999). As a result, this system determines educational material that is subsequently made into a set of electronic textbooks. Moreover, the ElmArt system provides intelligent tutoring, which enables support to a Lisp course that ranges from concept presentation to debugging programmes, and was advocated as an on-line intelligent textbook that included an integrated problem-solving environment(Weber and Brusilovsky, 2001). Likewise, ActiveMatch is based on intelligent tutoring systems, which is a generic web-based learning system that dynamically generates interactive (mathematical) courses adapted to the student's goals, preferences, capabilities, and knowledge (Melis et al., 2001). A knowledge tree system was proposed by Brusilovsky (2004). This is a learning support portal that enables access to different resources within a course objective hierarchy that has been detailed by a teacher. In general, the system's interface is static, although educational material from a variety of servers can be retrieved. The system also monitors the activities of the learners and adapts to the individual's knowledge level. The adaptive systems' limitations were stated by Brusilovsky (2004) that stem from the definition that ITSs relate to their architecture and not from their performance. In fact, the system should be shown in its entirety and not utilised in parts. Furthermore, a separate limitation is related to its flexibility, which was also affected by being ‘teacher directed’. Another limitation of the discussed ITS such as Elmart, Interbook, ActiveMatch, AutoTutor and knowledge tree system have no learning style. Therefore, incorporating a model of learning style has been considered in a variety of adaptive E-learning systems in order to influence learning content personalisation. For instance, one of the ITS that have incorporated learning style is INtelligent System for Personalised Instruction in a Remote Environment (INSPIRE) system, which utilises the Honey and Mumford’s learning style (Honey and Mumford, 1992) and adapts the presentation to the learner based on their learning style, was proposed by Papanikolaou et al. (2003), in order to create diverse lessons that fit individual learners that would meet their objectives. The learner initially completes the Honey and Mumford style questionnaire where different categories are recorded by the model for the learner: activist, pragmatist, reflector and theorist. It is 80-items questionnaire in order to give comprehensive analysis of learning style and suggestions for action in more depth, which makes it time- consuming. Moreover, the Learning Style Adaptive System (LSAS) was introduced by Bajraktarevic et al. (2003), which assimilated the combined sequential and global dimension of Index of learning style Felder-Solman (Felder and Soloman, 1997) and provided two individual user interface templates: sequential and global. In relation to sequential learners, small sections of information were presented on each page that showed pure text instead of separate links, as the ‘forward’ and ‘back’ were the only provided link buttons, and they enabled the learners to pursue a linear path of learning. Contrastingly, more navigational freedom is acquired by global learners. Different pages would comprise of individual parts, i.e. a table of contents; a summary and an overview; supplemental links; as well as links shown within the text. Consequently, learners were provided with a topic overview, which enabled the possibility to navigate with freedom through the course content. Another system that used the Index of Learning Styles (ILS) model (Felder & Silverman, 1988) is Oscar Conversational

Intelligent Tutoring System (CITS) (Latham et al., 2014). Oscar has also used natural language interface in order to allow learners constructing their own knowledge through discussions. Oscar CITS mimics a human tutor by detecting and adapting to student's learning style whilst directing the conversational process.

More recently VARK learning style has been one of the most widely used learning style. For instance, many well established E-learning systems such as Arthur system (Gilbert and Han, 1999), SACS (Style-based Ant colony system) (Wang et al., 2008), and AEHS-LS (An approach to Adaptive E-Learning Hypermedia System based on Learning Styles) (Mustafa and Sharif, 2011), have adopted the VARK learning style model (Fleming, 2001) to support the adaptability of these systems and provide suitable learning environment for each user. VARK model has been selected to be used in our system not only because of its wide use, but also because it is concise and uses a specific questionnaire that can provide preliminary learning style for each leaner in time-efficient way. In addition, Conlan et al. (2002) presented a pedagogical learning environment (OPAL) system that utilise Kolb/McCarthy's learning style models to categories learners into continuums (e.g. abstract/concrete and active/reflective). A separate investigation proposed the Tangow system that adapts the content of learning from an adaptation of the rules, combined with the model by Felder-Silverman (Carro et al., 2001). Nonetheless, this system presents a limitation in the level of ease of adaption rule modification, which needs to alter the implementation code through the developer of the system. As a consequence, insufficient flexibility and extensibility is offered through this system.

Ontologies are also becoming a great tool for developing personalised E-learning systems. For example, Yarandi et al. (2012) proposed an approach for developing personalised E-learning systems for learning content using ontology. They build the hierarchical and navigational relations between different parts of the learning material and how these can be determined based on users' profiles. Likewise, Sudhana et al. (2013), proposed an approach that includes domain ontology for organizing learning materials and learner-model ontology to manage the personalised delivery of learning material. Additionally, the development of learning content can be simplified from the use of ontologies to benefit authors and instructors, as this advances the developed personalisation levels and enables a combination of shared knowledge and enhanced reusability (Sicilia et al., 2011). The Curriculum Content Sequencing System was proposed by Chi (2009). He uses ontology purely to represent content sequencing and course materials in an abstract style. Furthermore, Boyce and Pahl (2007) emphasise that the utilisation of ontologies in E-learning systems are highly successful in a course's knowledge domain and permits a significant increase in an organisation's detail and the adaptation of students' learning paths. Moreover, Cassin et al. (2004) used the notion of ontology extraction for educational knowledge, which aims to help a student who needs to learn about some specific topic. They developed a system architecture for extracting the ontological information from raw webpages and into a useful structured form. That leads us to use ontology as an information retrieval tool to enable extracting relevant information through giving more organized and classified information about the domain knowledge. However, the major weakness of this approach is that developing an ontology from texts is an extremely difficult task and there is no guarantee that the ontology is complete or correct.

In this work, the first to support the adaptability of the system at the learners' levels, background, learning style and needs will be addressed by using questionnaires, which will be performed before the learning process starts. Subsequently, their progress will be tested following each level, and they will not be able to proceed until they have met the learning outcomes that could be under the form of taking an assessment for example. To make the learning process more enjoyable, the learning style of each student will be assessed using VARK questionnaire because it is concise and approachable. Furthermore, this system will have an ontology to help in extracting the required domain knowledge from the Web, in order to improve information retrieval, organise and update learning resources specific to the user. Although most of the available adaptive systems use a large number of rules to guide the learners in their learning process, these rules are created for a specific domain and cannot not be applied if the domain is changed. The APELS provides a very important addition in the world of adaptive learning as it enables the feature of adding/changing the ontology for a particular domain, as it will not require a complete change of the whole system. Furthermore, adding a resources feature to compare the produced material against known and standard curricula, such as the ACM/IEEE Computing Curriculum (Sahami et al., 2013), example used in this research, will ensures the quality of the produced content.

3.10 Summary

In order to develop a personalised and adaptable E-learning system that satisfies learner's need, it is important to understand the concept of learning. Furthermore, it was important to introduce different learning theories that have significant influence on learning and personalised E-Learning applications. Thus, the common learning theories are behaviourism, cognitivism and constructivism, which were described in this chapter.

It also reviewed the context of E-learning systems. The research has shown that there are many of frameworks of E-learning, which try to provide mechanisms that encourage the learning experience to be more pleasurable and a student-focused. The importance of the current E-learning platforms such as Blackboard, Moodle, and Sakai CLE were highlighted; these represent integrated systems which offer support for a wide area of activities in the E-learning process. Thus, teachers can use LMS for the creation of courses and test suites, for communicating with the students, for monitoring and evaluating their work. The problem is that LMS do not offer personalised services, all the students being given access to the same set of educational resources and tools, without taking into account the differences in knowledge level, interests, motivation and goals. Therefore, AHSs are a vital alternative to the traditional "one-size-fit-all" for learning model. AHSs collect information about its users and adapt the system's behaviour accordingly. The issues with these systems include flexibility, reuse and integration problems due to the nature of the architecture (Brusilovsky, 2004).

In addition, learning styles were also reviewed within this chapter as an approach of learning that allow the student to facilitate their acquisition of knowledge, skills or attitudes through study or experience. Therefore, this chapter reviewed some main learning style models that may be pertinent for adaptation and personalisation, and different learning styles were assessed for their suitability for the APELS system. The evaluation was based on the criteria proposed by Sampson and Karagiannidis (2002) and Peter et al. (2008). The evaluation demonstrated that Fleming's VARK would be a suitable learning style to incorporate within a model offering personalisation and adaptability to the learner. The main reasons for this choice are that the VARK learning style offers a concise questionnaire for a learner to complete comprising of a minimum number of relevant questions and that the learning style categories map clearly to learning object file types.

Moreover, ontology, as an important information retrieval tool, was reviewed and its importance to APELS was discussed. The evidence showed that it can be effectively used as a knowledge resource for extracting the required domain knowledge from the Web. In respect to the four main features (ontology, learning style model, adaptability and learning outcomes validation), APELS is compared to a number of available E-learning systems in (Table 3-2). This comparison explicitly shows what sets APELS apart from other recent developments in E-learning field is its adaptability. That is because tools used in this system can be easily re-used for any other domain of knowledge without the need to design a whole new system. Another important addition introduced by APELS in contrast to the available E-learning platforms is the learning outcome validation approach, which will be explained thoroughly in the next chapter, where the extracted learning material is validated against a set of pre-defined learning outcome.

Table 3-2 A comparison of some E-learning systems

System's name	Developed	Using learning style	Using ontology	Adaptability	Learning outcomes validation
Interbook	(Eklund and Brusilovsky ,1999)	No	No	No	No
Elmart	(Weber and Brusilovsky, 2001)	No	No	No	No
ActiveMatch	(Melis et al., 2001)	No	No	No	No
Knowledge tree system	(Brusilovsky, 2004)	No	No	No	No
INSPIRE	Papanikolaou et al. , 2003)	Yes	No	No	No
LSAS	Bajraktarevic et al. , 2003)	Yes	No	No	No
SACS	(Wang et al., 2008)	Yes	No	No	No
Tangow	(Carro et al., 2001)	Yes	No	No	No
Curriculum Content Sequencing	(Chi , 2009)	No	Yes	No	No
Rule-PAdel	(Yarandi et al. , 2013)	Yes	Yes	No	No
OSCAR	(Latham et al. , 2014)	Yes	No	No	No
AutoTutor	(Cai et al. , 2015)	No	No	No	No
APELS	(Aeiad, 2016)	Yes	Yes	Yes	Yes

Chapter 4 : A Framework for an Adaptable and Personalised E-Learning System

4.1 Introduction

This chapter describes the architecture of APELS which is based on three main models that will form the main components of the system. These include the Learner's model, the Knowledge extraction model and the Content delivery model. The learner model will be used to collect and manage the information about individual learners that will include the learner's background, needs, learning style and the history of the contents and resources used and preferred by the learner to support the adaptability and personalisation process of the E-learning system. The information collected from the learner's model will be used as an input for the information extraction model that is used to extract the learning resources from the Web that are suitable for the learner's needs and learning style. The Information extraction model comprises two phases; the Relevance phase and the Ranking phase. The relevance phase uses an ontology to retrieve the relevant information as per users' needs. The ranking phase, will be used to evaluate and validate the learning outcome from the content of the extracted resources against a set of learning outcomes as defined by standard curricula. This model has two advantages; it saves time searching for appropriate material on the web and ensures the suitability of the extracted material to the query. The third model is the Content delivery model which contains the planner model. The planner is responsible for generating and structuring the learning material of a specific module in a similar way as a lecturer would organise the teaching material of a module. In addition, the planner uses some adaptation rules for content adaptation based on the learner's content preference during the learning process.

Overall APELS system is an important addition to E-Learning, as institutions or individual learners may not have expertise or resources to design or develop learning material. At this stage, the academic learning that is conducted via APELS is based on resources available freely on the Web and does not require the involvement of field experts. Also, using standard search engines is time consuming and may not lead to suitable outcomes. These problems can be overcome by the APELS system. In addition, the adaptability of the system can be easily re-used for any other domain of knowledge without the need to design a whole new system but only replacing the ontology and adapting some rules.

4.2 System Architecture

The purpose of the APELS System is to deliver recommended learning materials to learners who may have different backgrounds, learning styles and learning needs. The architecture is based on three main models that will form the basis of the system. These include: learner model, knowledge extraction model, and content delivery model as shown in Figure 4-1.

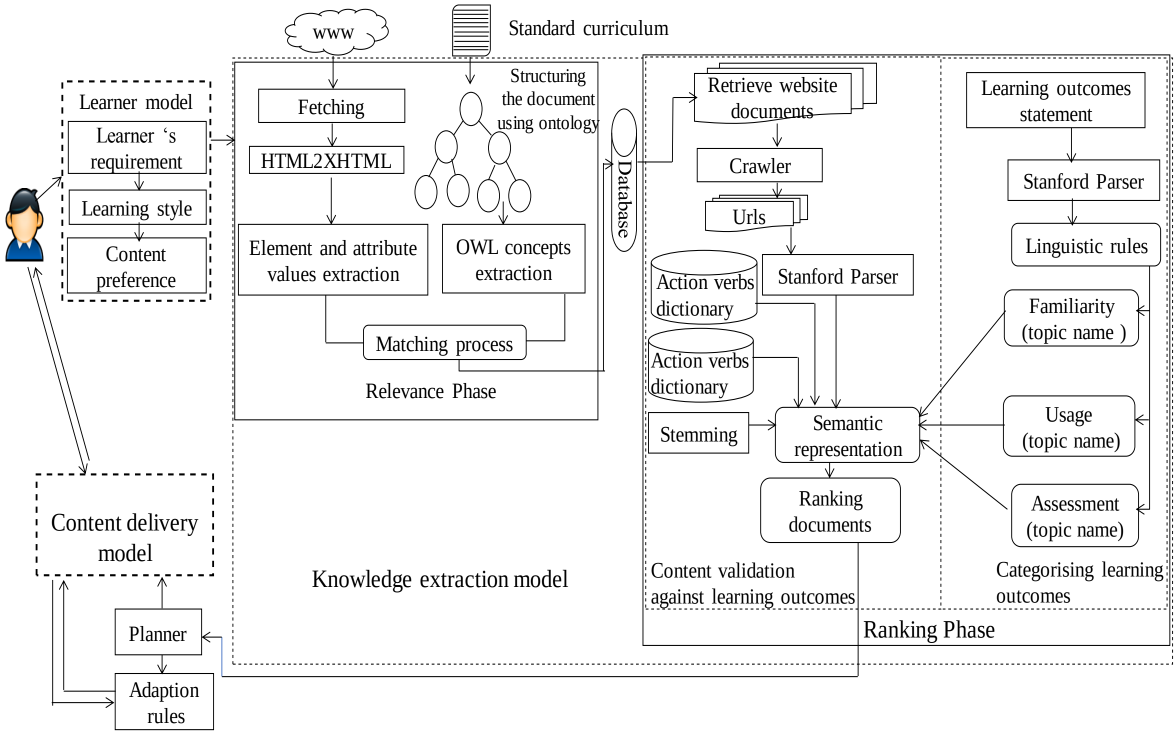

Figure 4-1 System architecture

4.3 The Learner Model

The learner model contains all the information about the learner in order to adapt to the learner's needs. When the system has collected this information, it can then provide the learner with the appropriate course content. The information collected by this model, will then be used to build a learner's model that represents his/her profile. During the learning process, the system updates the learner's model. The system can also provide suggestions after the learner finishes the course as to what they can do next. Furthermore, the learner model keeps the details of the student's profile to track their progress as they can return to the course when required. The learner model contains four components: Personal Information, Prior Knowledge, Learning Style, and Content Preference as shown in Figure 4-2.

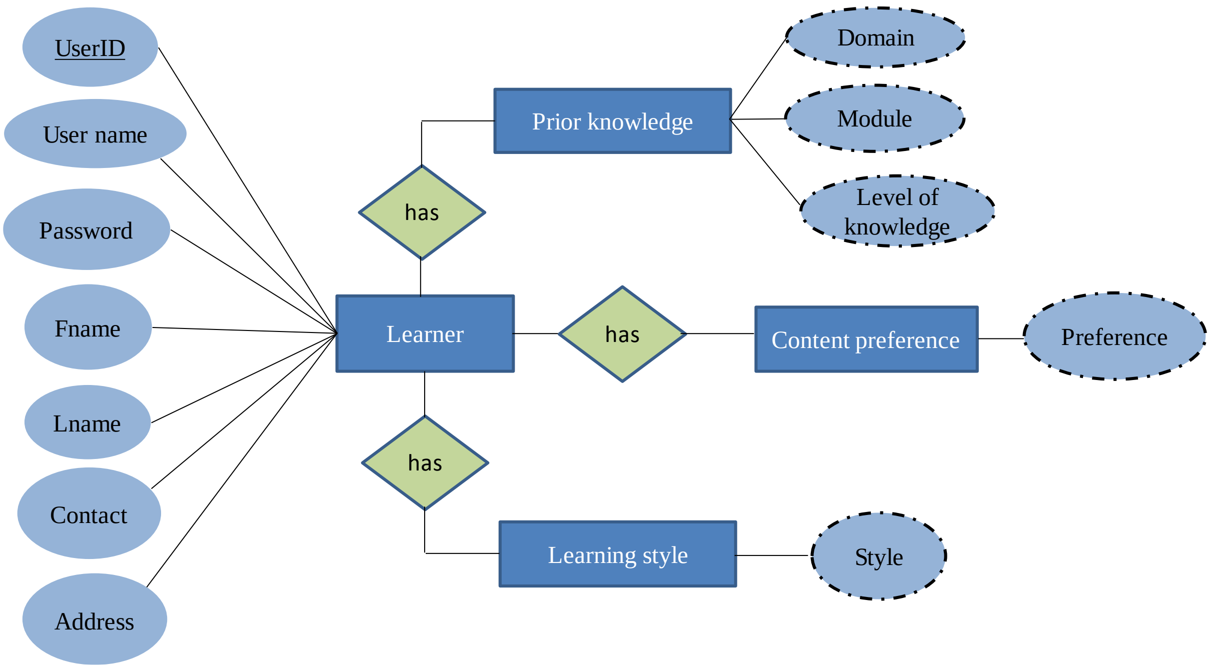

Figure 4-2 Entity-Relation Diagram (ERD) representation of learner model

4.3.1 Personal Information

This component will gather some personal information of the learner such as first name, last name, contact, and address. The learner will then be prompted to enter other information such as their user name and password to create an account in the system.

4.3.2 Prior Knowledge

Determining the level of the learners' knowledge is a crucial task in the personalisation process. Furthermore, the information that is related to the learner's background and previous knowledge are an important factor in learning as it can help or obstruct the learner in learning new concepts (Ausubel et al., 1978). Thus, after creating an account, in this step the learner will first choose a specific domain, and then s/he selects a module s/he wishes to study and their level of knowledge (Beginner – Intermediate - Advanced). For example, if the learner wishes to study Algorithms and Data Structures module, then s/he first selects the computer science domain, after that the system provides list of modules covered in computer science, finally, the learner selects his/her module (Algorithms and data structure) and the level of this module.

4.3.3 Learning Style

In education, when learning styles are applied, the learners are able to enhance their learning experiences as the course content would be presented in a manner that can be retained in the most appropriate way. Learning styles can be used to allow the student to facilitate their acquisition of knowledge, skills or attitudes through study or experience in accordance to their preference learning style (Sadler-Smith, 1996).

As described in Chapter 2 section 3.8.4, the VARK learning style has been chosen in this research as it is found to be more relevant as the associated tool has all the necessary questions to identify a user's learning style. It does not have any impact on the structure or sequence of the learning material. In addition, the tool can clearly identify and map the type of learning materials requested.

The latest version (7.8) of the VARK questionnaire, which was developed by Fleming (2016), was used in this research to determine the learning style preferences of the learners (see Figure 4-3). There are 16 questions in the questionnaire suggested by Fleming regarding the way learner likes to learn to analyse the suitable learning style. The list of the questions is given in Appendix C and a screenshot of the interface implementing this questionnaire is given in Figure 4-3.

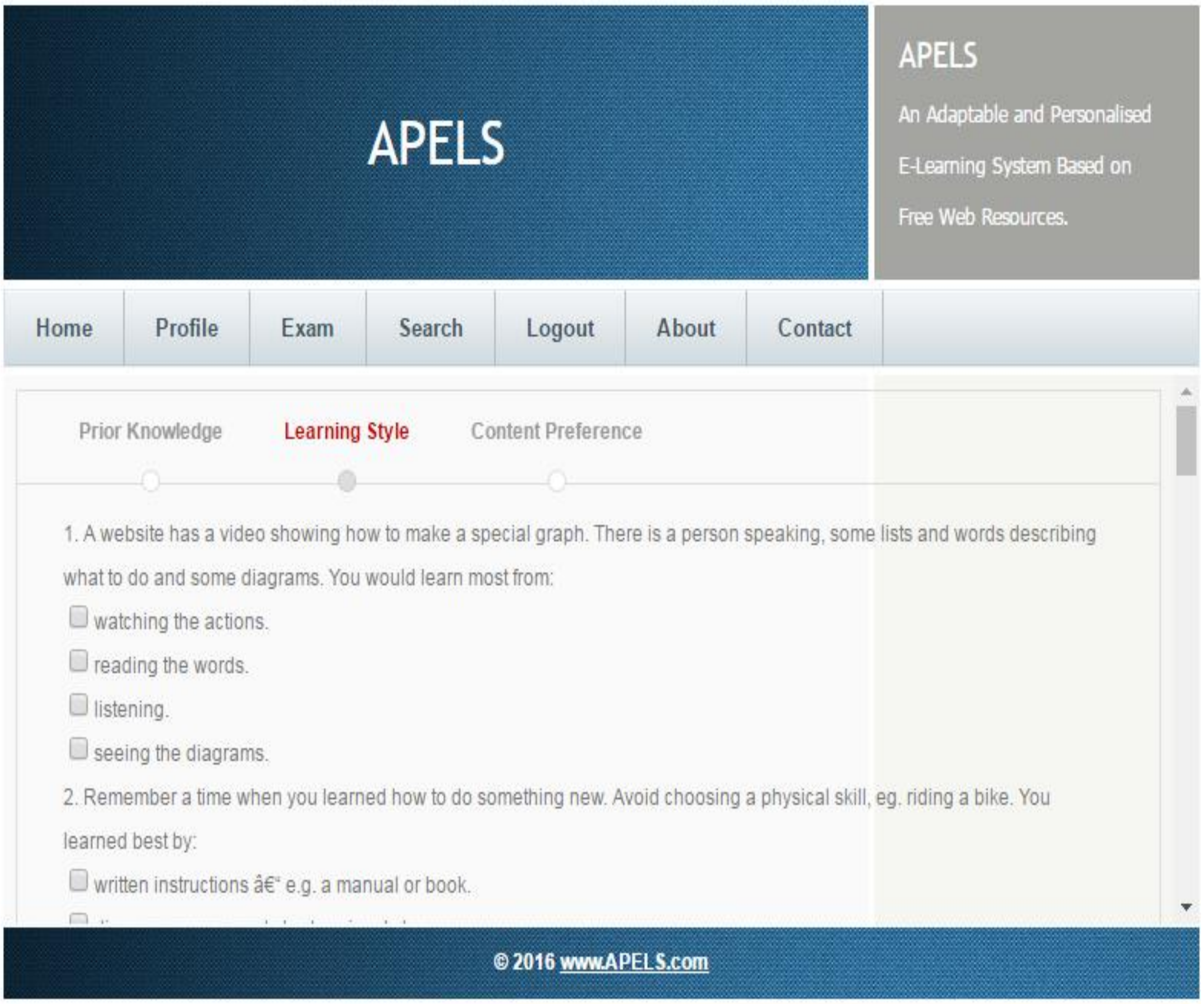

Figure 4-3 Sample questions of the VARK learning model implemented in APELS

After completing the questionnaire, the learners will be informed with their initial learning style preferences as retuned by the VARK score. The scores given by VARK model are a mixture of Visual, Aural, Read/Write and Kinesthetic and the highest score is assigned as the learning style of the user.

4.3.4 Content Preference

This component aims to provide two versions of the content, with each one specifying the same concepts of learning, although they are to be detailed in a different sequence and manner that will be based on the preferences of the learners. For example, one version can potentially begin with fewer definitions that will lead into distinct examples, while a contrasting example might begin with definitions prior to leading on to a detailed concept explanation and a reduction in examples. Therefore, the following algorithm defines how to adapt to the learner based on his/her content preference and learning style as given in **Figure 4-4**.

IF User (Z) select Module(X) and Learning style (Y) and Content preference('More_Definitaions_And_Less_Examples') THEN get URL based on LEARNINGSTYLESCORE, LEARNINGOUTCOMESCORE.

IF User (Z) select Module(X) and Learning style (Y) and Content preference('Less_Definitaions_And_More_Examples') THEN get URL based on LEARNINGSTYLESCORE, LEARNINGOUTCOMESCORE.

Y: V:"Visual" OR A:"Aural" OR R " Read/Write" OR K " Kinaesthetic "

LEARNINGSTYLESCORE : predicted by learning style model (VARK)

LEARNINGOUTCOMESCORE : given by rules (as developed in section 5.5.2)

Figure 4-4 Content preference selection algorithm

4.4 The Knowledge Extraction Model

Once the details of the learner and his/her chosen area are known, these are saved and submitted to be processed by the knowledge extraction model which is at the heart of the APELS architecture and is responsible for the extraction of the learning resources from the Web that would satisfy the learner’s need, learning style and learning outcomes. The model is divided into two phases: the relevance phase and the ranking phase. Figure 4-5 shows all the components and processes that support the knowledge extraction model.

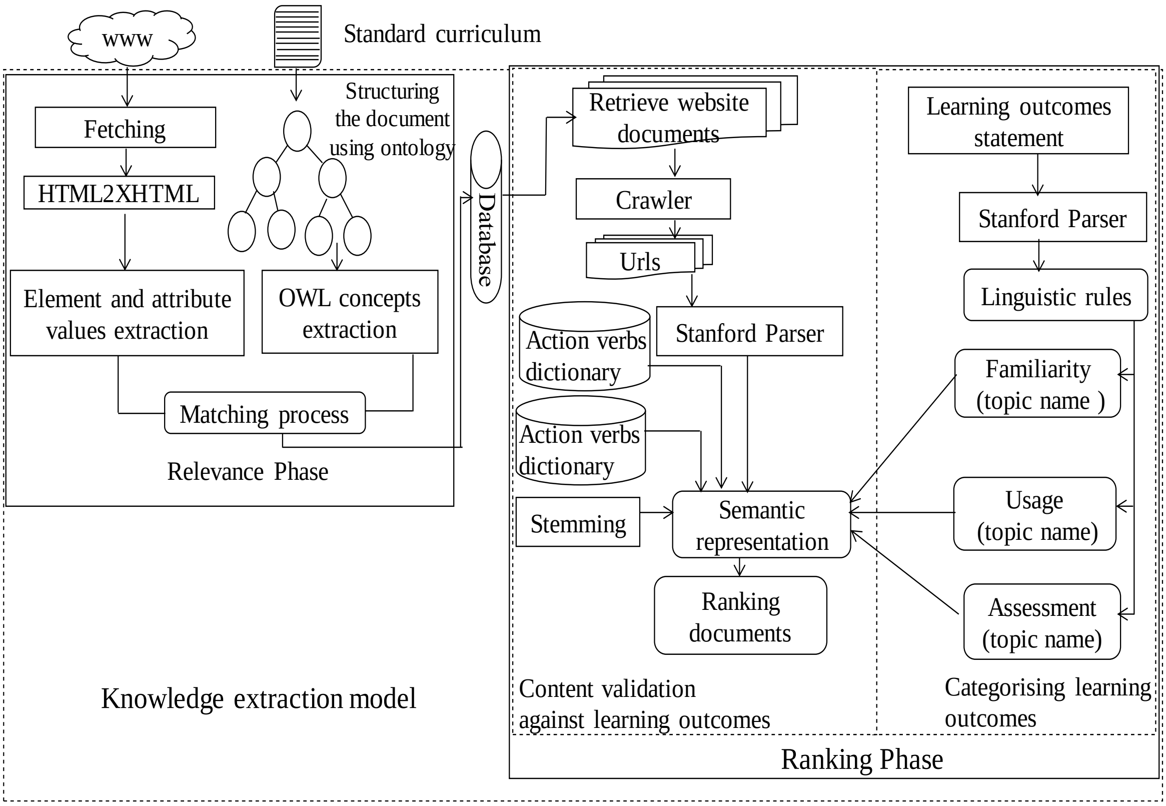

Figure 4-5 The knowledge extraction model

4.4.1 Relevance Phase

The Relevance phase uses an ontology (see section 5.5.1.1) to help in extracting the required domain knowledge from the Web in order to retrieve relevant information as per users 'requests. A number of models are developed to support this phase, which include: Fetching, HTML2XHML, Element and attribute values extraction, Ontology and OWL concepts extraction and Matching process. These processes and models are described in the following sections.

4.4.1.1 Fetching

The relevance phase starts first by fetching a list of websites that deal with the specific module (learning area). The search engine providers like Google, Yahoo and Bing, use a free open source package policy, so it was decided to use Google as an open source code, which was implemented in PHP (Github, 2016), to be integrated with the APELS system when fetching specific websites from the Internet based on learners' requests. These websites are first transformed into XHTML to provide the information in a friendly accessible format and easier for extraction and comparison as needed by the other processes and detailed in section 4.4.1.2.

4.4.1.2 HTML2XHML

Documents can be structured or unstructured. Unstructured documents have no (or very little) fixed pre-defined format, whereas structured documents are usually organized according to a fixed pre-defined structure. A structured document, for instance, comes in the form of an organised book chapter, with each individual section is divided into paragraphs. The most common way to structure the contents of documents in recent years was to use the W3C standard for information repositories and exchanges, and Extensible Hypertext Markup Language (XHTML) (Pemberton, 2000) is part of the family of Extensible Mark-up Language (XML) (Bray et al., 2008). XHTML is more strict than HTML because it has to respect all the XML rules (Closing tag for each opening tag, attributes have to have a value etc.). Therefore, HTML2XHTML approach was developed which will automatically create XHTML data sources from the collected HTML webpages. This tool includes some steps on how to transform HTML documents to XHTML documents and are summarised as follows:

1. Download HTML content from the WWW.
2. Save HTML content into file.
3. Apply HTML2 XHTML function which is able to clean and fix up a wide range of problems in HTML sources such as:
    - Missing or mismatched end tags are detected and corrected.

      For example: <h1> heading
      <h2> subheading </h3>
      Is corrected to

      <h1> heading </h1>
      <h2> subheading </h2>

    - End tags in the wrong order are corrected.

<p> here is a para bold <i> bold italic bold? </i> normal?

Is corrected to

<p>here is a para bold <i>bold italic</i> bold? normal?

- Fixes problems with heading emphasis.

For example: <h1><i> italic heading </h1>

Is corrected to

<h1><i>italic heading</i></h1>

- Recovers from mixed up tags, getting the <hr> in the right place.

For example: <i><h1>heading</h1></i>
<p>new paragraph bold text
<p>some more bold text

is corrected to

<h1><i>heading</i></h1>
<p>new paragraph bold text
<p>some more bold text

- Adding the missing "/" in end tags for anchors.

For example: <a href="#refs">References<a>

Is corrected to

References

- Perfecting lists by putting in tags missed out.

4. After processing cleaning and fixing up by the HTML2 XHTML function, it produces a clean and valid XHTML file.

4.4.1.3 Elements and Attribute Values Extraction

The specified text is enclosed in both the start and end tags within XHTML documents, defined as elements. For instance, <greeting>Hello, world! </greeting> are specific examples of these elements, while they may also be defined with attributes, as provided in the start tag (<note date="2008-01-10"></note>). This gives the attribute name as “date”, while the attribute value is stated as “2008-01-10”. An example of an XHTML document with elements, attributes and values is illustrated in Figure 4-6 together with its associated XHTML tree structure.

```
<? XHTML version="1.0" encoding="iso-8859-1"?>
<bookstore>
        <book category="cooking">
            <title lang="en">Everyday Italian</title>
           <author>De Laurentis</author>
           <year>2005</year>
           <price>30.00</price>
        </book>
        <book category="children">
                    <title lang="en">Harry Potter </title>
                    <author>J. K. Rowling</author>
                    <year>2005</year>
                    <price>29.99</price>
        </book>
</bookstore>
```

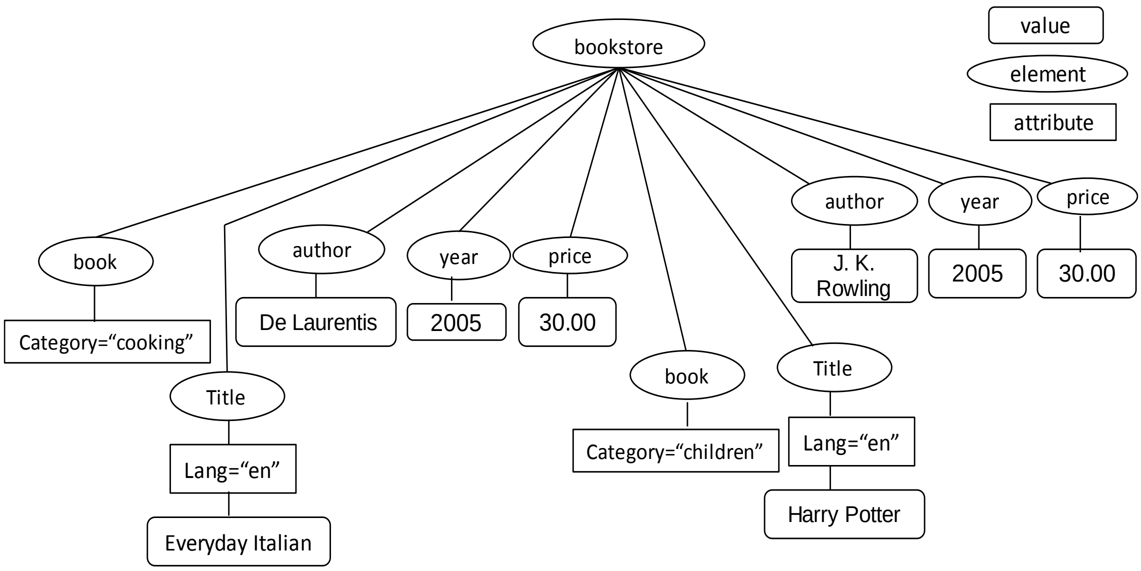

Figure 4-6 XHTML document and its associated tree structure

Xpath (XML language) is a query language for selecting or extracting useful elements and attribute values (i.e., subtrees) from XHTML documents. It has been defined by the W3C to navigate the components of an XHTML document (Clark and Derose, 1999). It uses path expressions to select nodes in an XHTML document.

For example, the Xpath expression (//a/@href [contains (.,topic)]), means that it will return any XHTML elements which have an Attribute with the value "topic". Thus, XPath is utilised in our work to extract XHTML elements and attributes values and these are saved in a vector denoted as V = [V_1, V_2, V_3,…, V_m]. The algorithm used for the extraction of XHTML elements and attributes values is given in Figure 4-7.

Algorithm : Elements and attribute values extraction
Input: XHTML Document Output: Vector of XHTML Values (V) BEGIN 1. Declare Vector (V) ; XHTML Document, Xpath; 3. Define Xpath to get the XHTML elements and attributes values from the input XHTML Document. 4. Pass XHTML Values and store into (V). 5. Return Vector of XHTML Values (V). END

Figure 4-7 XHTML elements and attribute values extraction algorithm

4.4.1.4 Ontology

An ontology is defined as a representation of a phenomenon's abstract model in the world through the use of conceptualisation, which assists in identifying the appropriation of domain concepts, through the use of formal definitions in terms of axioms and the concepts' semantic relationships (Chi, 2009). Knowledge representation using ontologies facilitates organising the metadata of complex information resources. These metadata provide syntactic and semantic information about information resources which are encoded as instances in the ontology (Sheth et al., 2002). A subset of the ontology for an educational domain is shown in Figure 4-8, where User, Student, Lecture, Module, Physics module, Math module, Basic Physics, Biological physics, Calculus, Linear algebra, Differential Equations are defined as concepts or classes. Different relations between concepts are defined in this example such as the Student class is defined as a subclass of User class and Physics Module and Math Module as subclass of Module. Furthermore, Figure 4-8 shows that Student has another relation with the Module which is “studies”. The relation "ispartof" or "has part" is used to represent part of whole class for example calculus is part of Math module.

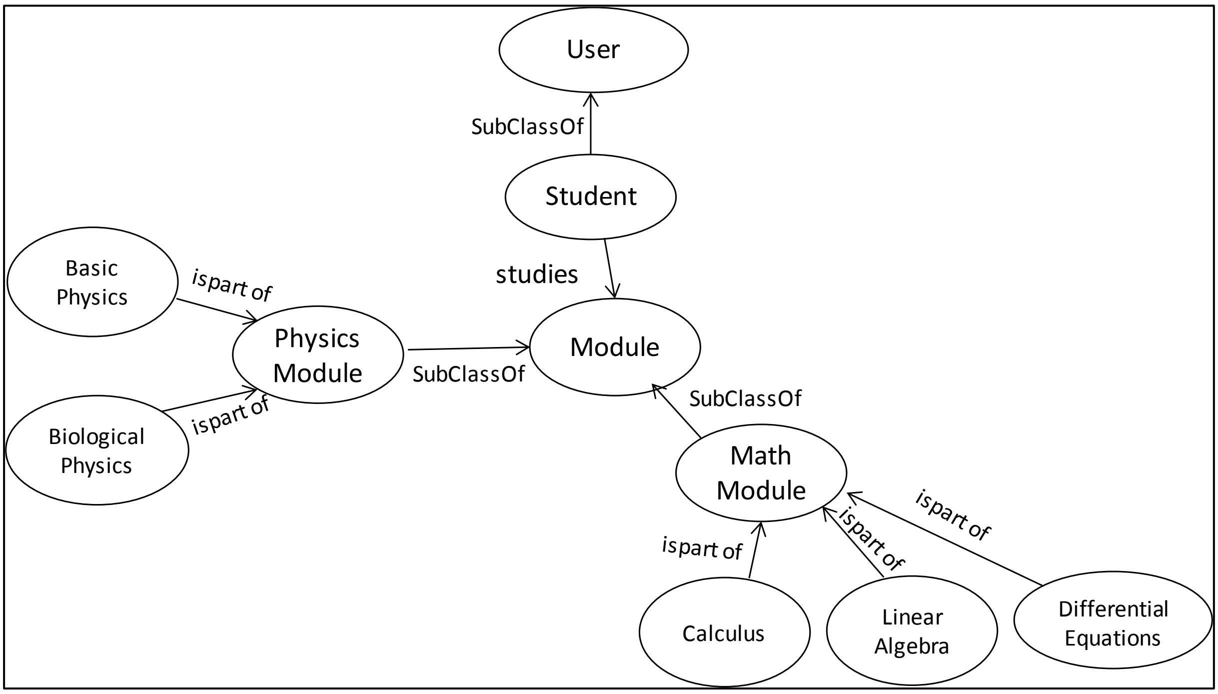

Figure 4-8 Example of subset of ontology

Ontology Web Language (OWL)

The Ontology Web Language (OWL) is a commonly utilised popular language for ontology representation, as defined by the World Wide Web Consortium (W3C) Bechhofer et al. (2004).

As there were already several ontology languages designed for use in the Web, OWL had to maintain as much compatibility as possible with these existing languages, including SHOE, which is a Knowledge representation language that allows ontologies to be designed and used directly on the World Wide Web (Berners-Lee et al., 2001), OIL (Fensel et al., 2000), which is a Web-based representation language that combines the widely used modeling primitives from frame-based languages with the formal semantics and reasoning services provided by description logics, and DAML+OIL (Horrocks, 2002), which are ontology languages developed for the Semantic Web. As such, they support its aim of increasing the amount of information on the Web that is computationally accessible (i.e., that can be unambiguously interpreted and processed by software as well as humans), and is integrated with RDF. The integration of OWL and RDF results in OWL being based on RDF's syntax, thus the web-based applications can directly access OWL ontologies.

The Protégé Editor

Protégé is an ontology and knowledge base editor produced by Stanford University (Noy et al., 2003) . Protégé is a tool that enables developers to create and edit domain ontologies. It allows the definition of classes, class hierarchies, variables, variable-value restrictions, and the relationships between classes and the properties of these relationships (Noy and Musen, 2000). Furthermore, protégé comes with visualization packages such as OntoViz; all of these help the user to visualize ontologies with the help of diagrams.

In our research, the APELS system uses an ontology to help in extracting the required domain knowledge from the Web in order to improve the information retrieval process and organize and update the learning resources specific to the user. Without the use of this semantic knowledge, the extraction process would be inefficient as simple keyword or metadata based searches supported by the current search engines and standards are not semantically rich and may not extract the needed information. We focus on concepts or classes and the relationships between these concepts when building the Ontology. The role of the ontology in the APELS system can be summarized as follows:

- Organizing the concepts or learning material using relationships in specific domains.
- Improving the information retrieval process.
- Updating the learning resources to specific user requirements.
- Defining synonyms of concepts via corresponding relations.
- Adaptability and extendibility as it facilitates adding/changing domains without the need to change the whole system.

4.4.1.5 OWL Concepts Extraction

The OWL file obtained from the protégé tool is used to extract the concepts or classes that are represented in a specific domain through the domain ontology. These concepts will be saved in a vector denoted as C = [c_1, c_2, c_3,…, c_m] to determine similarities with the XHTML files produced from HTML files. The algorithm used for the extraction of OWL concepts is given in Figure 4-9.

Algorithm: Ontology concept extraction
Input: OWLOntologyDocument Output: Vector of Ontology Concepts (C) BEGIN 1. Declare Vector (C) , OWLOntologyDocument, Xpath; 2. Define XPATH to get the Ontology concepts from the input OWLOntologyDocument 4. Pass ontology concepts and store into (C) 5. Return Vector of Ontology Concepts (C) END

Figure 4-9 OWL concepts extraction algorithm

4.4.1.6 The Matching Process

The matching process computes the similarity measure between the ontology concepts that represent the learning domain, saved in the vector C, and the values extracted from the websites, saved in the vector V. Given a set of relevant websites and their associated value vectors, the website with the highest similarity is selected as the best matching website for the learner's request. The Dice Coefficient (Dice, 1945) was utilised in this process, as it has been used extensively in many Information Retrieval (IR) applications due to its good performance and easy to use. Moreover, Normalisation (stemming) is used in the matching process to improve the performance of the similarity measure. Once the system extracts the concepts or terms from the web, these terms are analysed to get the root of the word that will be matched with the ontology concepts. As explained in Chapter 2, removing the common endings from words, such as 'ing', or 'es' increases the performance of IR systems and in this research it has increased the number of matches between the ontology concepts and terms in the documents. Therefore, the Porter stemming algorithm (Porter, 1980) was used as it is the widest applied stemming technique for removing iteratively suffixes from a given word, reducing it to its stem. Nevertheless, there are some limitations of the current approach when running our experiments. Sometimes parts of the website only are relevant and appropriate for the learner and a combination of two or more websites contents will provide a better learning material. In addition, some concepts or terms may be given different names, although they have the same meaning. For instance, the equivalent terms for the concept "Calculus" includes arithmetic, mathematics etc. This issue was solved by defining corresponding relations such as synonyms in the domain ontology.

Given two vectors C and V defined as follows:

$C = [c_1, c_2, c_3, \ldots, c_m]$ where C_i represent an ontology Concept and

V = $[V_1, V_2, V_3, \ldots, V_m]$ where V_j represent XHTML Elements and Attribute Values Extraction

The similarity measure between vectors *C* and *V* using the Dice coefficient is calculated by equation (4.1).

$$J\ (C_i, V_j) = \frac{2|c_i \cap v_j|}{|c_i| + |v_j|} \tag{4.1}$$

Where $c_i \cap v_j$ is the number concepts in *C* that are also present in *V*, the algorithm used for similarity process is given in Figure 4-10.

Algorithm: Similarity Finding
Input: $C=\{C_1,C_2,...C_n\}$, $V=\{V_1,V_2,...V_m\}$ Output: Similarity S(C,V) 1. Begin 2. V=array_unique(*V*) 3. S1=sizeof (c) , S2=sizeof(v) 3. For each Ontology Concept C_i Compare C_i with all values V_j in V If $C_i = V_i$ then sc = sc +1 4. END For 5. Similarity S(C,V)= $\frac{2*(sc)}{(s1)+(s2)}$ 6. Return S(C,V)

Figure 4-10 The similarity measure algorithm

Figure 4-11 illustrates an example where the concepts that model the learning domain encircled in the OWL ontology concepts and the elements attribute values encircled in XHTML document were extracted for computing their similarities.

Figure 4-11 OWL concepts and XHTML values extraction

4.4.2 The Ranking Phase

After performing the matching process, the learning outcome validation approach was added to ensure the selection of the most relevant websites to enable learning according to the learning outcomes set by standard curricula. The validation of learning outcomes includes two components: Categorising learning outcomes statements and Content validation against learning outcomes.

4.4.2.1 Categorising Learning Outcomes Statements

The suitability of the contents of the selected website should be evaluated to ensure that they fit the learner's needs. Matching the content to learning outcomes of curricula is very important when assessing the validity of the selected websites. Basically, learning outcomes are statements of what a student is expected to know, understand and/or be able to demonstrate after the completion of the learning process (Kennedy, 2006). Likewise, Mclean and Looker (2006) described learning outcomes as explicit statements of what we want our students to know, understand or be able to do as a result of completing our courses. Bloom's Taxonomy (Bloom, 1956), is one of the most important and popular frameworks for developing learning outcomes in order to help students understand what is expected of them. The following sections explain how to identify learning outcomes, which include Bloom's Taxonomy and Nouns and Verbs Extractor.

4.4.2.1.1 Bloom's Taxonomy

The Bloom's taxonomy can be used to identify a set of learning outcomes. Typically, a learning outcome contains a verb and a noun. In one hand, the verb describes the intended cognitive level of the Bloom's taxonomy and includes Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation. In the other hand, the noun describes specific subject that student wants to learn. For example: basic structure of the genetic material; nature of chromosomes and the organisation.

Thus, a set of action verbs was used based on the Bloom's taxonomy to analyse the learning outcomes. Furthermore, the Bloom taxonomy identified a list of suitable action verbs into six levels representing the following cognitive skills: Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation. For example, action verbs such as define, describe and identify are used to measure basic levels of cognitive skills in comprehension, while action verbs such as carry out, demonstrate, solve, illustrate, use, classify and execute are used to measure basic levels of the application cognitive skills. Table 4-1illustrates a set of action verbs associated with the intended cognitive level of the Bloom's original taxonomy.

Table 4-1 Set of action verbs associated with the intended cognitive levels of the Bloom's original taxonomy (Bloom, 1956)

Bloom's Original Taxonomy of the Cognitive Domain	
Cognitive Level	Sample verbs to use in Writing Intended Student learning outcomes
knowledge	Define, Identify, Name, Recognize, Retrieve, Duplicate, List, Recall, Reproduce, Tell.
Comprehension	Calculate, Conclude, Predict, Discuss, Explain, Classify, Clarify, Translate, Reproduce, Exemplify.
Application	Carry out, Demonstrate, Solve , Illustrate, Use, Classify, Execute, Implement, Practice, Utilize.
Analysis	Discriminate, Compare, Differentiate, Examine, Infer, Attribute, Contrast, Distinguish, Select, Formulate.
Synthesis	Check, Judge, Monitor, Critique, Reconstruct, Defend, Verify, Detect, Coordinate, Dispute.
Evaluation	Construct, Design, Compose, Produce, Improve, Create, Invent, Generate, Plan, Combine.

In 2001, a former student of Bloom's, Lorin Anderson, and a group of cognitive psychologists, curriculum theorists and instructional researchers, and testing and assessment specialists published a revision of the Bloom's Taxonomy entitled A Taxonomy for Teaching, Learning, and Assessment. The revision updates included significant changes in terminology and structure. In the revised framework, "action words" or "verbs", instead of nouns, are used to label the six cognitive levels, and three of the cognitive levels are renamed. Figure 4-12 illustrates the differences between Bloom's original taxonomy and the 2001 revised one.

Original Taxonomy (1956)	**Revised Taxonomy (2001)**
Knowledge	Remembering
Comprehension	Understanding
Application	Applying
Analysis	Analyzing
Synthesis	Evaluating
Evaluation	Creating

Noun Form ⟶ **Verb Form**

Figure 4-12 Differences between Bloom's original taxonomy and the 2001 revised one

Table 4-2 A set of action verbs associated with the intended cognitive level of the revised Bloom's taxonomy (Anderson et al., 2001)

Bloom's Original Taxonomy of the Cognitive Domain	
Cognitive Level	Sample verbs to use in Writing Intended Student learning outcomes
Remembering	Define, Identify, Name, Recognize, Retrieve, Duplicate, List, Recall, Reproduce, Tell.
Understanding	Calculate, Conclude, Predict, Discuss, Explain, Classify, Clarify, Translate, Reproduce, Exemplify.
Applying	Carry out, Demonstrate, Solve , Illustrate, Use, Classify, Execute, Implement, Practice, Utilize.
Analysing	Discriminate, Compare, Differentiate, Examine, Infer, Attribute, Contrast, Distinguish, Select, Formulate.
Evaluation	Check, Judge, Monitor, Critique, Reconstruct, Defend, Verify, Detect, Coordinate, Dispute.
Creating	Construct, Design, Compose, Produce, Improve, Create, Invent, Generate, Plan, Combine.

4.4.2.1.2 Nouns and Verbs Extractor

In order to identify learning outcomes statements, first, two types of dictionaries are used. The action verbs dictionary that contains the action verbs that have been manually defined based on the Revised Bloom's taxonomy (Anderson et al., 2001) and the topic name synonym dictionary whose terms are retrieved from the ontology. Second, as previously mentioned in chapter 2, the Stanford CoreNLP (Manning et al., 2014) is used in this research as a tool to parse learning outcome statements and generate semantic representation in XML format, including PoS tagging text and a typed dependency representation (see Figure 4-13).

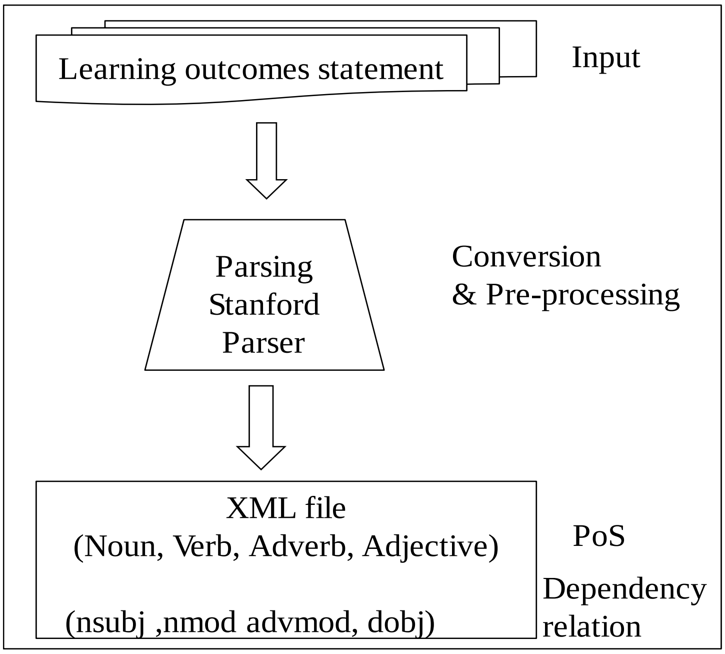

Figure 4-13 Stanford CoreNLP

In addition, the XML format shows data in a tree structure format where tags are assigned for each word. Theses tags include <root> (governor of sentences), <lemma> (word form), < sentence id > (sentence number), tokens, word, PoS, parser, dependencies etc. as shown in Figure 4-14. The Parser tag is very important as it represents the context-free phrase structure grammar representation, which is used to describe the structure of sentences and words in natural language that provide the structure derivations of the grammar.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="CoreNLP-to-HTML.xsl" type="text/xsl"?>
<root>
 <document>
  <sentences>
   <sentence id="1">
    <tokens>
     <token id="1">
      <word>define</word>
      <lemma>define</lemma>
      <CharacterOffsetBegin>0</CharacterOffsetBegin>
      <CharacterOffsetEnd>6</CharacterOffsetEnd>
      <POS>VB</POS>
      <NER>O</NER>
      <Speaker>PER0</Speaker>
     </token>
    </tokens>
    <parse>(ROOT (S (VP (VB define)))) </parse>
    <dependencies type="basic-dependencies">
     <dep type="root">
      <governor idx="0">ROOT</governor>
      <dependent idx="1">define</dependent>
     </dep>
    </dependencies>
    <dependencies type="collapsed-dependencies">
     <dep type="root">
      <governor idx="0">ROOT</governor>
      <dependent idx="1">define</dependent>
     </dep>
    </dependencies>
    <dependencies type="collapsed-ccprocessed-dependencies">
     <dep type="root">
      <governor idx="0">ROOT</governor>
      <dependent idx="1">define</dependent>
     </dep>
    </dependencies>
   </sentence>
  </sentences>
 </document>
</root>
```


Figure 4-14 XML format generated by Stanford CoreNLP

Consequently, from the product of the Stanford parser, all the tokens with the verb tag were extracted and then check if it matches with a verb from the action verbs dictionary that contains the action verbs that have been manually defined based on the Revised Bloom's taxonomy (Anderson et al., 2001).

A set of rules is used to identify learning outcomes statements by searching the pattern token in the tagged verb in the action verbs dictionary that have been manually defined based on the Revised Bloom's taxonomy in order to identify learning outcomes statement. Table 4-3 illustrates the rules that are used to assign learning outcomes based on action verbs in the Bloom's taxonomy.

Table 4-3 Rules to assess learning outcomes statement using Bloom's taxonomy

Rule for Remembering Level If pattern token in tagged verb =Remembering (action verbs) then learning outcomes =" Remembering" Example action verbs for remembering level Define ,Identify ,Name, Recognize, Retrieve, Duplicate, List, Recall, Reproduce, Tell.
Rule for Understanding Level If pattern token in tagged verb =Understand (action verbs) then learning outcomes ="Understand" Example action verbs for Understand level Calculate, conclude, Predict, Discuss, Explain, Clarify, Translate, Reproduce, Exemplify.
Rule for Applying Level If pattern token in tagged verb =Applying (action verbs) then learning outcomes ="Applying" Example action verbs for Applying level Carry out, Demonstrate, Solve, Illustrate, Use, Execute, Implement, Practice, Utilize.
Rule for Analyzing Level If pattern token in tagged verb =Analyzing (action verbs) then learning outcomes ="Analyzing" Example action verbs for Analyzing level Discriminate, Compare, Differentiate, Examine, Infer, Attribute, Contrast, Distinguish, Select.
Rule for Evaluating Level If pattern token in tagged verb =Evaluating (action verbs) then learning outcomes ="Evaluating" Example action verbs for Evaluating level Check, Judge, Monitor, Critique, Reconstruct, Defend, Verify, Detect, Coordinate, Dispute.
Rule for Creating Level If pattern token in tagged verb =Creating (action verbs) then learning outcomes ="Creating" Example action verbs for Creating level Construct, Design, Compose, Produce, Improve, Create, Invent, Generate, Plan, Combine.

4.4.2.2 Content Validation Against Learning Outcomes

The evaluation of the topic's content will be against the identified learning outcomes statements. Each learning outcome contains an action verb followed by usually a noun phrase that acts as the object of the verb. Together, the action verbs and noun phrases are referred to as Keywords or key phrases. These are used in academic publications for example to give an idea about the content of the article to the reader as they are a set of representative words, which express the meaning of an entire document. In addition, when significant phrases or (key phrases) in the document are provided, the relevance of a document can be determined quickly by a prospective reader, as well as initiating fast information retrieval (Siddiqi and Sharan, 2015).

In this research, NLP techniques were used to validate the contents against learning outcomes. Linguistic knowledge / features of the words were used to extract significant key phrases and keywords that represent each document, in order to decide which website satisfies the learning outcomes. Eight linguistic rules have been designed to capture key phrases and keywords based on determining linguistic patterns in dependency relation and part of speech using the Stanford English Parser. For example, a linguistic rule was employed to extract the syntactic structure of sentences that include a noun phrase followed by the verb "to be” as in the phrases "variable is" and "algorithms are". These expressions indicate that the document has definitions of the concept. Moreover, a number of components are developed to validate the content against learning outcomes. This include a Crawler, a Dependency relation, and a PoS tagger. These components are described in the following subsections.

4.4.2.2.1 The Crawler

The goal of this step is to fetch webpages from the Web using keywords or topic names. These extracted webpages will be evaluated later to validate the content against a set of learning outcomes. An algorithm was developed to check whether the keyword or topic name is included in the URL of the webpage (Meziane and Kasiran, 2003). For example, in the website “http://www.cplusplus.com”, the system will extract all URLs appearing on this website, then the system checks if the keywords or topic name is included in the URL of the webpage, it will save that page in the database to evaluate the content against the identified learning outcomes statements, otherwise it will ignore it, and checks the following webpage. However, some target keywords are not included in the URLs. This issue was solved by extracting the title tag or title element of the webpage, which is a crucial element in identifying the content of the webpage. Then the system checks if the keyword or topic name matches with the text value of the title tag of the webpage. Figure 4-15 illustrates the crawler process in the APELS system.

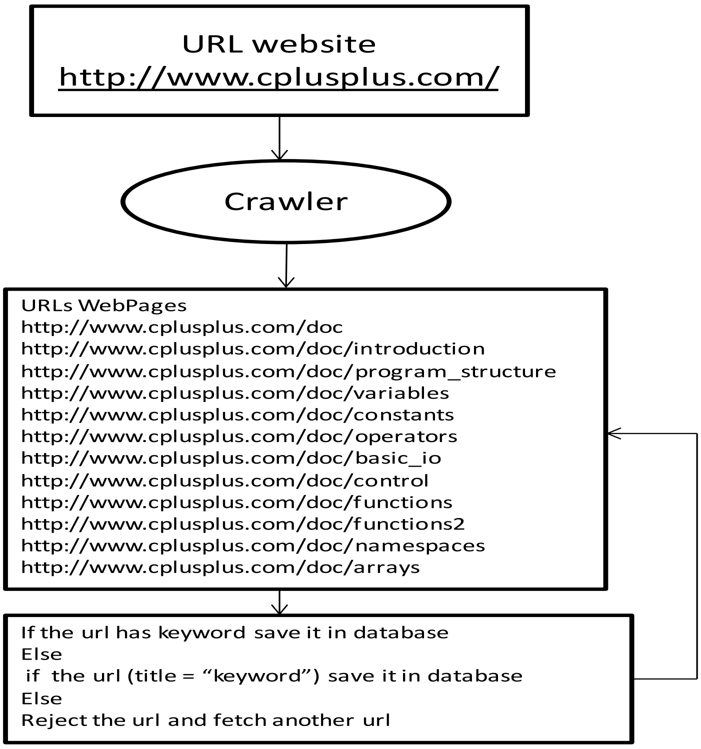

Figure 4-15 Crawler process in APELS

4.4.2.2.2 Dependency Structure

Dependency Grammar (Tesnière, 1959) is a syntactic tradition that determines sentence structure on the basis of word-to-word connections, or dependencies. It names a family of approaches to syntactic analysis that all share a commitment to word-to-word connections. In addition, the document's words are connected to each other by directed links, and called one of them, the head and the other the dependent. As in the example given in Figure 4-16, the dependency link is an arrow pointing from the head (hit) to dependents (Mark, ball) and the arrow pointing from head (ball) to dependents (the).

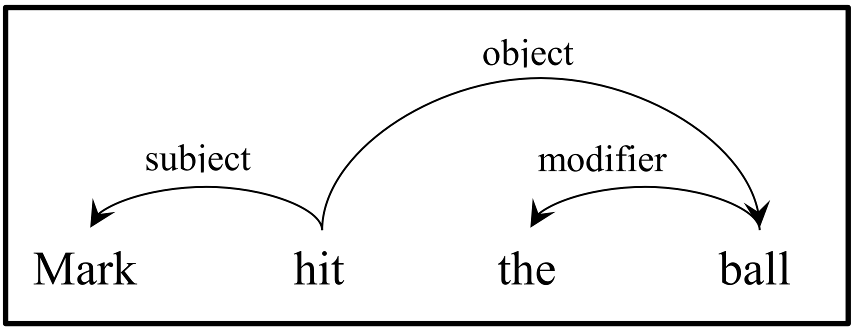

Figure 4-16 An example of the dependency structure

Dependency Features

The Stanford parser was applied to build the dependency tree, which can determine the dependency relationships between the words of a sentence (Manning et al., 2014). Figure 4-17 shows the dependency tree for the example sentence (Bell, based in Los Angeles, makes and distributes electronic, computer and building products). From the resulting dependency relationships between words, the dependency relation between (distributes), and (Bell) is shown through (nsubj: norman subject); while the type of relation between the two words (distributes), and (products) is shown by the labels in the edges between the nodes (dobj: direct object).

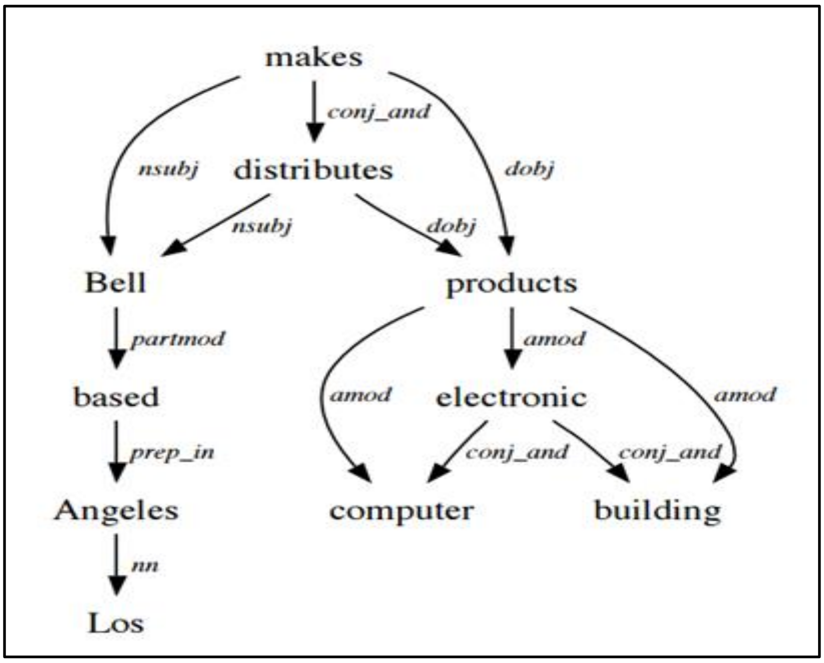

Figure 4-17 The dependency structure as a tree for the sentence (Bell, based in Los Angeles, makes and distributes electronic, computer and building products)

The Stanford Lexicalized Dependency Parser (De Marneffe and Manning, 2008) provides a simple description of the grammatical relationships in a sentence, establishing relationships between "head" words and words which modify those heads ("refer"). For the "Bell, based in Los Angeles, makes and distributes electronic, computer and building products." the Stanford Dependencies (SD) representation would be:

nsubj(makes-8, Bell-1)	nsubj(distributes-10, Bell-1)
vmod(Bell-1, based-3)	nn(Angeles-6, Los-5)
prep in(based-3, Angeles-6)	root(ROOT-0, makes-8)
conj and(makes-8, distributes-10)	amod(products-16, electronic-11)
conj and(electronic-11, computer-13)	amod(products-16, computer-13)
conj and(electronic-11, building-15)	amod(products-16, building-15)
dobj(makes-8, products-16)	dobj(distributes-10, products-16)

Each typed dependency is structured in the following way:

dependency_name (governing word – index, subordinate word – index)

For instance, if we consider the typed dependency for *dobj (makes-8, products-16)* in the sentence above, we can see the relationship between word 8 (makes) and word 16 (products) is of a direct object (dobj). Moreover, because (makes) is the governing word and (products) is the subordinate, (products) is therefore the direct object of (makes).

In our work, the APELS system extracts the key phrases using the dependency parsing approach by checking the output of a typed dependency pattern of Stanford parser whether the governor of the dependency is an action verb and its subordinate a topic name or whether the governor of the dependency is a topic name and its subordinate an action verb. In addition, the other advantage of the dependency parsing approach is to extract the relationship information between governing word and subordinate word even if they are separated by many words. For example, in the sentence above, the output of the typed dependency for nsubj (makes-8, Bell-1) showed a dependency relation between the words “makes” and “Bell” although there are many words separating them.

4.4.2.2.3 Normalisation (Stemming)

The normalisation (stemming) process is necessary for our key phrases extraction process using the dependency parsing approach. Once the Stanford parser produces the typed dependency between a pair of words, these are analysed to get the root of the word that will be looked up in the action verb dictionary and the topic name synonyms from the ontology. Therefore, the Porter stemming Algorithm (Porter, 1980) was used as it is the widest applied stemming technique for removing iteratively suffixes from a given word, reducing it on its stem. Moreover, the other distinctive feature of the Normalisation (stemming) process is to reduce the size of the action verbs dictionary and topic name synonym because they contain all the different forms of the word. The Algorithm for extracting the action verb, topic name and their relationship is given in Figure 4-18.

Algorithm: Action verb, Topic name and relationship Extractor
1: Parse the sentences in the XML document and produce the typed dependencies for them using the Stanford Parser. 2: for each typed dependency pattern do 3: Extract governor of dependency and subordinate of the dependency; 4: Stem governor of dependency and subordinate of the dependency; 5: if (dep-governor =action verbs dictionary[i]) && (dep-subordinat =topic name synonym [j]) then sc++ 6: if (dep-governor =topic name synonym [i]) && (dep-subordinat =action verbs dictionary synonym [j]) then sc++ 7: end if 8: end for. sc :number of the relationship between action verb and topic name

Figure 4-18 Action verb, topic name and their relationship extraction algorithm

4.4.2.2.4 PoS Tagging

Three kind of linguistic rules have been designed in order to extract the key phrases and keywords from the text. For example, in order to investigate whether the content provided clarification of the topic via examples, a rule was used to search for terms such as” for example” or” for instance” in the text. Thus this rule is based on searching the patterns in PoS, which is one of the main tasks in syntactic text analysis in order to signify contrasting lexical word categories that include: adjectives, adverbs, verbs, nouns, which are present within texts.

In the next chapter, an explanation of how the APELS system employs linguistic knowledge / features of the words including part of speech and dependency relation to extract relevant key phrases and keywords that represent each document will be illustrated. Then, apply our linguistic rules to decide which website satisfies the learning outcomes. Finally, APELS system ranks the relevant documents based on the number of occurrences of keywords and key phrases in the document.

4.5 Content Delivery Model

Once the APELS system extracted information, taking into consideration the learner’s requirements, learning style and learning outcomes, then it will structure and generate a learning plan in a similar way as academic staff would do for their module specification including the contents.

4.5.1 The Planner

Vital issues in research for learning systems stem from 'adaptation' or 'personalisation', as systems which have the ability to change specific parameters and allow behaviour adaptation are referred to as “adaptable”. Separately, “Adaptive” is the term given to the systems that provide an assumption of the needs required and are able to adapt to users automatically (Santally and Senteni, 2005). The APELS system includes a planner, which contains adaptation techniques to support the adaptive functionality of the system in order to update the content based on the learner’s content preference. The adaptation process takes also advantage and uses the learner’s feedback and learning style. The functionality of the planner will be described in details in this section. The Planner plays a vital role in structuring the extracted information into module specifications for particular learner including the following key components:

- Module code and title
- A summary of the programme aims
- The intended learning outcomes
- The program structure is divided into five categories: Topic name, Recommendation link, where links provide personalised content of learning material that automatically was extracted from freely available resources on the Web to an individual learner according to his/her prior knowledge (see section 4.3.2), content preference (see section 4.3.4) and learning style (see section 4.3.3), Learning hours as suggested by ACM/IEEE, which was subdivided evenly to cover all the topics, for example 2 hours for each topic as shown in Figure 4-19. The programme structure also includes Exercise and Evaluation. Figure 4-19 shows a snapshot of the module specifications page for particular user in the APELS system for the fundamental programming module including three layered formats consisting of module code and title, summary of programme aims, intended learning outcomes and program structure.

Module Title :Fundamental Programming

Module Code: CF201

First Name : **Samir**
Last Name : **Arohma**
Learning Style : **Visual**

Aims of the Module

The aim of this module is to introduce the student of fundamental programming concepts and enhance their problem-solving. Students will learn the basics of scalar types (Integers, Strings, Booleans) and fundamental control structures in procedural programming (loops, assignment statements, conditional expressions). The module uses the C++ programming language as the implementation environment. This course also will allow the students how to implement file I/O, functions and recursion for solving a problem.

Learning Outcomes

1.Identify and describe uses of primitive data types. [Familiarity]
2.Write programs that use primitive data types. [Usage]
3.Write programs that use standard conditional [Usage]
4.Write program that use iterative control [Usage]
5.Write program that use functions. [Usage]
6.Write a program that uses file I/O. [Usage]
7.Choose appropriate conditional and iteration constructs for a given programming task. [Assessment]
8.Describe the concept of recursion and give examples of its use. [Familiarity]

Program Structure

Topic name	Recommended Link	Learning Hours	Exercise	Evaluation
Conditional	Link	2	Link	Is this useful ?
Loops	Link	2	Link	Is this useful ?
Variables	Link	2	Link	Is this useful ?
Functions	Link	2	Link	Is this useful ?
Recursion	Link	2	Link	Is this useful ?

Figure 4-19 Module specifications for the fundamental programming module

4.5.2 The Adaptation Process

The planner contains adaptation rules used to modify the learning content based on learners' feedback, and thus, this would be advantageous for the next generation of E-learning systems. A strong feedback from users is a good opportunity to rank and evaluate the content.

Accordingly, four questions were devised and implemented in the evaluation section of the module specification page. These are:

1- How satisfied are you with the content?
2- How satisfied are you with completeness of the content?
3- How satisfied are you with academic quality of the content?
4- How satisfied are you with the learning experience?

The answer of these questions can be rated from 5 to 1 where “5” Strongly satisfied, “4” Satisfied,”3” Neutral, “2” Not satisfied, and “1” very dissatisfied.

Questions 1, 2 and 3 were designed in order to investigate the learners’ opinion about the quality content delivered, whether it is relevant and clear which helps learners to fully comprehend concepts. Whereas, question 4 is associated with the learning style of the learner, which was used to update the learning style based on the learner’s feedback, which will be explained in this section. Moreover, it was used to know the extent to which the learners are satisfied with the learning experience.

To evaluate the produced content, the system calculates the average score of the first three learner’s answers using a simple equation (4.2), which helps devise decisions in order to update the content and re-rank the webpages in the system.

$$\text{User rating} = \frac{\text{answer for question 1+ answer for question 2+ answer for question 3}}{4} \qquad (4.2)$$

Where 4 represents the total number of questions.

For example, if “Satisfied” is selected in the first question by the users, the second question is “Neutral”, and the third question shows “Very dissatisfied”, then the final score is computed using a simple equation and illustrated as:

4 (Satisfied) + 3 (Neutral) + 1 (very dissatisfied) = 8 then calculate average = 8 / 4 = 2.00

This average score will be stored in the user’s rating database and this plays a vital role in ranking the webpages in the system based on the learner’s feedback. The system updates the content of the URL in Recommendation link by finding the higher score in user rating which will be recommended to other users. For example, if the learner would like to study Function topic as part of Fundamental programming concepts, and he was not satisfied with the content, as a result he gave a low average score. Then the system exchanges the learning content presentation with one that has a higher score. Over time the system keeps evaluating the presented content based on learner's feedback, in order to assist them in learning in a better and more effective and efficient manner.

A vital instrument that assists individuals and improves learning experiences is achieved through the utilisation of learning styles within the remit of education as this enables personalised design of the content of the course according to the way they learn (Sadler-Smith, 1996). Moreover, adapting the learning content to the learner based on his/her learning style will provide an enjoyable learning environment, which will facilitate making a good learning experience (Graf and Kinshuk, 2007). The system first identified the specific learning style of the learner through the VARK questionnaire (Fleming, 2016). This type of learning style can be updated based on the answers of the fourth question of the learner's feedback. Moreover, the following equation (4.2) was introduced to calculate the score of learning style based on the answer of the fourth question.

$$\text{LSS} = \text{y-} \left(\sum_{i=1}^{3}(5 - scoreQuestion[i])/3\right) \quad (4.2)$$

Where:

LSS is the learning style score.

y is the answer to question 4.

i is the answer to question 1-3

5 is The number of points on the scale.

For example, if "Neutral" is selected in the first question by the users, the second question is "Not satisfied", the third question shows "Neutral", and fourth question is "satisfied "then the calculation score is shown as: LSS = 4 – (((5-3) + (5-1) + (5-3)) /3) = 1.33

This score of a particular learning style will be stored and then the planner automatically update the present content of the learner based on the learning style. For example, if the learner would like to study Stack topic as part of Algorithms and data structures, and s/he was not satisfied with the learning experience, and the score given for that was very low then, the system will search for a better content which has a higher score for the user rating and learning style. The algorithm used to adapt the content and learning style based on learner's feedback is given in Figure 4-20.

Algorithm : Update the content based on learner's feedback
1. Begin 2. Score1= answer for question 1, Score2= answer for question 3 Score3= answer for question 3, Score4= answer for question 4 3. User rating = $\frac{(\boldsymbol{Score\ 1+Score\ 2+Score\ 3})}{\boldsymbol{4}}$ 4. Update user rating in Database 5. Calculate learning style based on the formula **Learning** $\boldsymbol{Style = scoreQuestion}[\boldsymbol{4}] - \sum_{\boldsymbol{i=1}}^{\boldsymbol{3}}(\boldsymbol{5 - scoreQuestion}[\boldsymbol{i}])/\boldsymbol{3})$ 6. Update learning style score in Database 7. Order user rating and learning style by descending 8.end

Figure 4-20 Content and learning style adaption algorithm

4.6 Summary

APELS design was illustrated by including its three main models, which were employed for extracting the information from the Web in order to satisfy learner's requirements. We also illustrated the components and processes in the learner model which is very crucial to support the adaptability and personalisation processes of the E-learning system. In addition, this chapter introduced the knowledge extraction model which is at the heart of the architecture of APELS as it is responsible for the extraction of the learning resources from the Web. With respect to the content validation, our proposed learning outcomes validation approach was presented in order to evaluate the topic content against a set of learning outcomes as defined by standard curricula. Finally, the content delivery model was presented in the form of a planner, which is responsible for generating and structuring the learning plan for the module including the content. In addition, adaptation rules were described in this model for content adaptation based on the learner's content preference. The adaptation process takes also advantage and uses the learner's feedback and learning style.

Chapter 5 : Case Study and Implementation

5.1 Introduction

To illustrate the working mechanisms of the APELS system and how it would be used in practice, an implementation using the ACM/IEEE (Sahami et al., 2012) Computer Society Computer Science Curriculum, which is an internationally recognised and commonly adopted in the design of computer science and software engineering curricula across the world is used. At the start of this chapter, the body of knowledge of the ACM/IEEE Curriculum, which is developed by the world leading professional societies in the field of computing, will be described in details. This include the principle elements that revolve around the areas of knowledge and the subsequent units, required learning hours, and learning outcomes.

The second part of this chapter will present the technical aspects of the implementation of our rules and approaches for the APELS system, which was described in Chapter 4. Section 5.5.1.1, and will illustrate the use of an ontology to structure the knowledge domain by organising the topics of the ACM/IEEE Computer Science Curriculum and semantic relation between domain topics. Section 5.5.1.2, will present the implementation of the matching process and will include a number of models and rules to support the relevance phase. In section 5.5.1.3, how the relevance phase is implemented and used through the provision of examples from the ACM/IEEE curriculum will be discussed. In section 5.5.2.4, the implementation of the learning outcome validation process, which uses linguistic rules based on grammatical dependencies relation and PoS tagging, will be presented in order to validate the content against a set of learning outcomes as defined in standard curricula. Following this, a brief summary of the chapter is also given.

5.2 Background and History

Over the last forty years, many major organisations have developed computing curriculum guidelines for colleges and universities (Atchison et al., 1968). As a result, there are now curricular volumes for computer science, as the field of computing has begun to advance, develop and diversify. An example of such organisation is the Association for Computing Machinery (ACM), a scientific and professional organisation that was founded in the year 1947 and is currently the largest scientific and educational computing society in the world. It is concerned with the development and sharing of new knowledge in relation to all aspects of computing. These developments included all computing areas such as computer engineering, systems of information technology, software engineering and computer science. The advancements in the guidelines of curricula with regards to computer science is a big challenge given the quick developments in the field. Furthermore, the ever expanding diversity within computer science creates new challenges to the understanding and modernisation of the curriculum within the field coupled with the increasing integration of computing with other disciplines.

The ACM has published many curriculum recommendations that were relevant for the computer society community since 1968. In addition, the Computer Society of the Institute for Electrical and Electronic Engineers (IEEE-CS), which originated in 1946, also started to contribute to the development and enhancement of the computer science (CS) discipline (Wood, 1995). This institute has focused on computing from the engineering perspective. Furthermore, volunteer boards are maintained within 6 areas of the CS, which are: education, membership, professional activities, publications, standards, and technical and conference activities. Additionally, the CS participates in the perpetual enhancement of relevant computing curricula for colleges, which is undertaken together with the ACM. Another organisation, the Association for Information Systems (AIS), a global organisation that serves academics specialising in Information Systems, advances knowledge and the improvement in the application of understanding the use of information systems was founded in 1994.

In the modern era, and with their combined efforts, the ACM and IEEE have adopted a strategy to sponsor new defined strategies to implement and improve curriculum guidance in relation to computing, which is conducted at approximately 10 years intervals. As the field of computing has grown and diversified, it is important to have the curricular recommendations, therefore, there are now curricular volumes for Computer Engineering, Information Systems, Information Technology, and Software Engineering in addition to Computer Science (Shackelford et al., 2006). These volumes are updated regularly with the aim of keeping computing curricula relevant. A complete volume of CS was originally released back in the year 2001 (Curricula, 2001), although in 2008, a new interim review was produced that extended the findings and brought new conclusions (Cassel et al., 2008). Through these two documents the outline of topics that need to appear in the CS curricula were shown together with the most commonly utilised aspects. Meanwhile, new topical areas were shown to be required within the body of knowledge. Finally, the Computer Science Curricula 2013 (Sahami et al., 2013) was produced, which is currently the latest development and it is this version that is used in this case study. However, prior to implementing this version, it is necessary to show a clear understanding of the body of knowledge that has been used for the implementation of the architecture of the APELS system and its evaluation.

5.3 Overview of the Body of Knowledge

When a new curriculum is developed in any field, it is vital to identify the correct body of knowledge that underlies its development and implementation. With regard to studying computer science curriculum for example, the ACM/IEEE CS 2013 identified a set of specific areas that when combined produced a representation of the computer science’s body of knowledge, which in CS 2013 includes Information Assurance and Security, alongside Parallel and Distributed Computing along the more traditional computer science topics. Furthermore, the principles that were demonstrated for CS 2013 have a significant connection with those from prior curricula, such as those developed in 2001 and 2008.

This body of knowledge enhances and creates an understanding in defining the entire curriculum for an institution that offers computer science programmes. In addition, curricula recommendations are provided in relation to the requirements of an institution, as well as to provide the continuation in the evolution of the field. Overall, CS 2013 highlighted that realistic and adoptable recommendations must be instilled, which enable guidance together with flexibility that permits the development of curricula that are designed to respond to rapid changes. Therefore, the following sections explain in more detail, the structure of the body of knowledge, and these include defined knowledge areas and preferred units, hours of curricula as well as the learning outcomes.

5.4 Structure of the Body of Knowledge

CS2013 presents a body of knowledge (BoK) for the computer science curriculum, which is organised hierarchically into three levels, in order to present a set structure of the development within the curriculum. The highest level of the hierarchy is the Knowledge Areas (KAs), which represents a particular disciplinary subfield that can be used at specific moments within the development of the system. Each KA is identified by a two-letter abbreviation, such as OS for operating systems or PL for programming languages. Additionally, each KA is broken down into smaller divisions that are called Knowledge Units (KUs), which represent individual thematic modules within an area, and so the evaluation and understanding of the entire curriculum can be analysed and comprehended more easily. For example, Software Development Fundamentals is a KA, which requires 43 hours of study and includes a set of KUs, such as Algorithms and Design (11hr), Fundamental Programming Concepts (10hr), Fundamental Data Structures (12hr), and Development Methods (10hr) as illustrated in Figure

5-1. Finally, each KU is further subdivided into a set of topics and learning outcomes, which are the lowest level of the hierarchy.

AL-Algorithms and Complexity (28)
AL/Basic Analysis (4)
AL/Algorithmic Strategies (6)
AL/Fundamental Data Structures and Algorithms (12)
AL/Basic Automata, Computability and Complexity (6)
AL/Advanced Computational Complexity
AL/Advanced Automata Theory and Computability
AL/Advanced Data Structures, Algorithms, and Analysis

CN-Computational Science (1)
CN/Modeling and Simulation
CN/Processing
CN/Interactive Visualization
CN/Data, Information, and Knowledge

IAS. Information Assurance and Security (8)
IAS/Fundamental Concepts (3)
IAS/Network Security (5)
IAS/Cryptography
IAS/Risk Management
IAS/Security Policy and Governance
IAS / Digital Forensics
IAS / Security Architecture and Systems Administration
IAS/Secure Software Design and Engineering

IS. Intelligent Systems (10)
IS/Fundamental Issues (1)
IS/Basic Search Strategies (4)
IS/Basic Knowledge Representation and Reasoning (3)
IS/Basic Machine Learning (2)
IS/Advanced Search
IS/Advanced Representation and Reasoning
IS/Reasoning Under Uncertainty
IS/Agents
IS/Natural Language Processing
IS/Advanced Machine Learning
IS/Robotics
IS/Perception and Computer Vision

NC. Networking and Communication (10)
NC/Introduction (1.5)
NC/Networked Applications (1.5)
NC/Reliable Data Delivery (2)
NC/Routing And Forwarding (1.5)
NC/Local Area Networks (1.5)
NC/Resource Allocation (1)
NC/Mobility (1)
NC/Social Networking

SP. Social Issues and Professional Practice (16)
SP/Social Context (3)
SP/Analytical Tools (2)
SP/Professional Ethics (4)
SP/Intellectual Property (2)
SP/Privacy and Civil Liberties (2)
SP/Professional Communication (1)
SP/Sustainability (2)
SP/History
SP/Economies of Computing
SP/Security Policies, Laws and Computer Crimes

AR. Architecture and Organization (16)
AR/Machine level representation of data (3)
AR/Assembly level machine organization (6)
AR/Memory system organization and architecture (3)
AR/Interfacing and communication (1)
AR/Functional organization
AR/Multiprocessing and alternative architectures
AR/Performance enhancements

GV. Graphics and Visualization (3)
GV/Fundamental Concepts (3)
GV/Basic Rendering
GV/Geometric Modeling
GV/Advanced Rendering
GV/Computer Animation
GV/Visualization

IM. Information Management (10)
IM/Information Management Concepts (3)
IM/Database Systems (3)
IM/Data Modeling (4)
IM/Indexing
IM/Relational Databases
IM/Query Languages
IM/Transaction Processing
IM/Distributed Databases
IM/Physical Database Design
IM/Data Mining
IM/Information Storage And Retrieval
IM/MultiMedia Systems

OS. Operating Systems (10)
OS/Overview of Operating Systems (2)
OS/Operating System Principles (2)
OS/Concurrency (3)
OS/Scheduling and Dispatch (3)
OS/Memory Management (3)
OS/Security and Protection (2)
OS/Virtual Machines
OS/Device Management
OS/File Systems
OS/Real Time and Embedded Systems
OS/Fault Tolerance
OS/System Performance Evaluation

SF. Systems Fundamentals (27)
SF/Computational Paradigms (3)
SF/Cross-Layer Communications (3)
SF/State-State Transition-State Machines (6)
SF/Parallelism (3)
SF/Evaluation (3)
SF/Resource Allocation and Scheduling (2)
SF/Proximity (3)
SF/Virtualization and Isolation (2)
SF/Reliability through Redundancy (2)
SF/Quantitative Evaluation

DS. Discrete Structures (41)
DS/Sets, Relations, and Functions (4)
DS/Basic Logic (9)
DS/Proof Techniques (11)
DS/Basics of Counting (5)
DS/Graphs and Trees (4)
DS/Discrete Probability (8)

HCI: Human Computer Interaction (8)
HCI/Foundations (4)
HCI/Designing Interaction (4)
HCI/Programming Interactive Systems
HCI/User-Centered Design & Testing
HCI/Design for Non-Mouse Interfaces
HCI/Collaboration & Communication
HCI/Statistical Methods for HCI
HCI/Human Factors & Security
HCI/Design-Oriented HCI
HCI/Mixed, Augmented and Virtual Reality

PD. Parallel and Distributed Computing (14)
PD/Parallelism Fundamentals (2)
PD/Parallel Decomposition (4)
PD/Communication and Coordination (4)
PD/Parallel Algorithms, Analysis, and Programming (3)
PD/Parallel Architecture (2)
PD/Parallel Performance
PD/Distributed Systems
PD/Cloud Computing
PD/Formal Models and Semantics

SDF. Software Development Fundamentals (43)
SDF/Algorithms and Design (11)
SDF/Fundamental Programming Concepts (10)
SDF/Fundamental Data Structures (12)
SDF/Development Methods (10)

SE. Software Engineering (27)
SE/Software Processes (3)
SE/Software Project Management (2)
SE/Tools and Environments (2)
SE/Requirements Engineering (4)
SE/Software Design (8)
SE/Software Construction (2)
SE/Software Verification and Validation (3)
SE/Software Evolution (2)
SE/Formal Methods
SE/Software Reliability

Figure 5-1 Body of knowledge for computer science curriculum (Sahami et al., 2013)

5.4.1 Knowledge Areas

New and modern advancements in computing technology, together with pedagogy have resulted in certain concepts within the core of the curriculum to evolve over time, as particular elements of prior structures and organisations could well have become inappropriate in the process of defining and analysing the discipline. Consequently, the details from the study in 2013 updated and modified the curriculum's organisation in a variety of manners, as it added fresh KAs while also restructuring other ones. The KA represents topical areas of study in computing and are:

- AL - Algorithms and Complexity
- AR - Architecture and Organization
- CN - Computational Science
- DS - Discrete Structures
- GV - Graphics and Visual Computing
- HCI - Human-Computer Interaction
- IAS - Information Assurance and Security
- IM - Information Management
- IS - Intelligent Systems
- NC - Networking and Communications
- OS - Operating Systems
- PBD - Platform-based Development
- PD - Parallel and Distributed Computing
- PL - Programming Languages
- SDF - Software Development Fundamentals
- SE - Software Engineering
- SF - Systems Fundamentals
- SP - Social Issues and Professional Issues

5.4.2 Knowledge Units

The knowledge units (KUs) were defined in CS 2013 in relation to Computer Science, and detailed what is vital in the implementation of CS curricula. This also aims to identify examples of courses and programs, in order to create the provision of correct guidance in relation to the structure of the curricula and development. Each KU functions with a KA that details the set of topics required and the intended learning outcomes. In defining different levels, it is possible to create other curriculum approaches, which can be explored within CS.

5.4.3 Required Learning Hours

CS 2013 identified the measurements for the units in regards to the body of knowledge that referred to hours of learning in the curricula. This "hour" relates as a directive of the time that is required to present the relevant material in a lecture-oriented format of a traditional classroom approach. Nevertheless, this does not include preparation time or the time that is spent outside of the class by the students (e.g., in self-study, lab classes, assessments, etc.). Indeed, students generally spend a large amount of additional time with the development of their materials that are presented in the class. Following CC 2001 and CS 2008, the unit of coverage in the body of knowledge in relation to lecture hours is understandable and transferrable in cross-cultural contexts.

5.4.4 Learning Outcomes

Learning outcomes are central components to any body of knowledge. Basically, they are intended to capture what students are able to do after they have acquired the knowledge. CS 2013 has developed a set of learning outcomes designed to promote assessment of student achievement. These learning outcomes have an associated task of mastery in the Bloom's taxonomy, which has been well explored within the Computer Science domain based on the ACM/ IEEE Computing curriculum (Sahami et al., 2013). The task of mastery is defined in the Familiarity, Usage and Assessment tasks.

5.4.4.1 Familiarity Task

This task of mastery concerns the basic awareness of a concept. It provides an answer to the question "What do you know about this?". The initial level of understanding of any topic is answering the question "what the concept is or what it means?". For instance, if we consider the notion of iteration in software development, this would include for-loops, while-loops and iterators. At the "Familiarity task," a student would be expected to understand the definition of the concept of iteration in software development and know why it is a useful technique.

5.4.4.2 Usage Task

After introducing the concept to the learner, it would be essential to apply the knowledge in a more practical way, such as using a specific concept in a program, use of a particular proof technique, or performing a particular analysis. It provides an answer to the question "How to use it?". For instance, if we consider the concept of arrays in programming languages, a student at the "Usage" task, should be able to write or execute a program properly using a form of array.

5.4.4.3 Assessment Task

This task of mastery implies more than using a concept; it involves the ability to select an appropriate approach from different alternatives. It provides an answer to the question "Why would you do that?" Furthermore, the student is able to consider a concept from multiple viewpoints and/or justify the selection of a particular approach to solve a problem. For instance, understanding iteration in software development, at the "Assessment" task would require a student to understand several methods for iteration and be able to appropriately select among them for different applications

5.5 System Implementation

The computer science field using the ACM/IEEE Computing curriculum as the standard curriculum was described in detail at the beginning of this chapter including its main components Knowledge areas, Knowledge units, curricula hours, and learning outcomes. In the second part of this chapter, the working mechanisms of the APELS system will be introduced and how it would be used in practice using the ACM/IEEE Curriculum as an example. Moreover, structuring the knowledge domain using ontology and implementation of relevance phase through the provision of examples will be given. An implementation of the learning outcome validation process to validate the content against a set of learning outcomes is also provided.

5.5.1 Relevance Phase

The ontology was used in the relevance phase to structure the knowledge domain of a pre-selected area of the Computer Science Curriculum, in order to extract relevant learning resources from the Web. Moreover, a number of models and rules are developed to support this phase, which include: Fetching, HTML2XHTML, Element and Attribute Values Extraction, OWL Ontology Concepts Extraction, and Matching Process. The implementation of these processes is explained in the following subsections.

5.5.1.1 Domain Knowledge Construction

The APELS system uses an ontology to help extract the required domain knowledge from the Web in order to improve the information retrieval process, organize and update learning resources specific to the user. Therefore, concepts are organised into a set hierarchy, together with the semantic relations that relate them. Moreover, in order to edit and develop the ontology for a specific domain, the Protégé editor was utilised, as the Graph User Interface (GUI) within it, allows developing ontology to focus on terms of concept without contemplating output of the ontology language syntax. The Protégé Editor also defines different classes and hierarchies, together with variables and potential restrictions, as well as the connections between the classes and how these relationships are structured (Noy et al., 2003).

As mentioned in section 5.2, certain elements of the ACM/IEEE Computer Society Computer Science Curriculum (Sahami et al., 2013) were used to illustrate the implementation of the APELS system.

For example, the knowledge area “Software Development Fundamentals” can be defined as a class and its knowledge unites, such as Algorithms and Design, Fundamental Programming Concepts, and Fundamental Data Structures can be defined as it’s a subclasses. Finally, the lowest level of the hierarchy, which includes a set of topics, can be defined as a subclasses of the KU. For example, a set of topics such as Structure of a Program, Variables, Expressions, Conditional, Control Structures, Functions, File Input and Output, and Concept of Recursion can be defined as subclasses of the KU (Fundamental Programming Concepts). Figure 5-2 shows a snapshot of the Protégé editor which illustrates the hierarchy of the relevant domain concepts and relations between these concepts for ACM/IEEE Computer Science Curriculum.

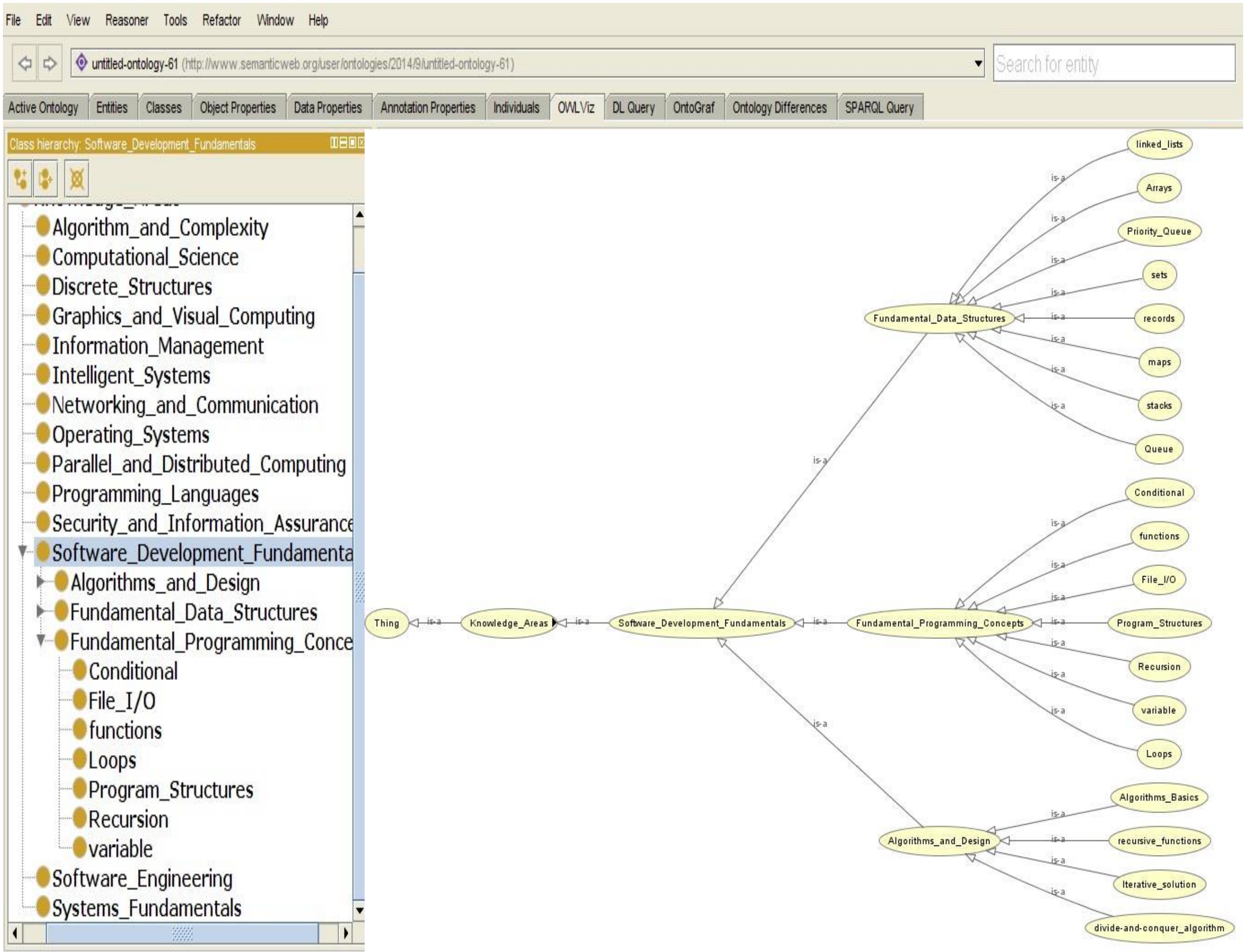

Figure 5-2 Classes and subclasses in the Protégé editor

In addition, the ontology web languages (OWL) will structure the output of the ontology editor, as this is utilised in order to produce domain structure's ontological modelling. Furthermore, class and subclass definition are provided through OWL, which was used in the current study to describe the hierarchy of the topics. The following OWL code illustrates the overall class-subclass relationship for knowledge area "Software Development Fundamentals":

```
<SubClassOf>
    <Class IRI="#Fundamental_Programming_Concepts"/>
    <Class IRI="#Software_Development_Fundamentals"/>
</SubClassOf>

<SubClassOf>
    <Class IRI="#Structure of Program"/>
    <Class IRI="#Fundamental_Programming_Concepts"/>
</SubClassOf>

<SubClassOf>
    <Class IRI="#Variables"/>
    <Class IRI="#Fundamental_Programming_Concepts"/>
</SubClassOf>

<SubClassOf>
    <Class IRI="#Expressions"/>
    <Class IRI="#Fundamental_Programming_Concepts"/>
</SubClassOf>

<SubClassOf>
    <Class IRI="#Simple I/O"/>
    <Class IRI="#Fundamental_Programming_Concepts"/>
</SubClassOf>

<SubClassOf>
    <Class IRI="#Conditional"/>
    <Class IRI="#Fundamental_Programming_Concepts"/>
</SubClassOf>

<SubClassOf>
    <Class IRI="#Contril structure"/>
    <Class IRI="#Fundamental_Programming_Concepts"/>
</SubClassOf>

<SubClassOf>
    <Class IRI="#Functions"/>
    <Class IRI="#Fundamental_Programming_Concepts"/>
</SubClassOf>

<SubClassOf>
    <Class IRI="#Concept of recursion"/>
    <Class IRI="#Fundamental_Programming_Concepts"/>
</SubClassOf>
```

In addition, the ontology will be used not only for improving the information retrieval process, but also to provide semantically identical concepts. For example, concepts may be given different names although they have the same meaning. For instance, the equivalent terms for the concept “IF Statement” includes Conditional or Selection Statement. This issue was solved by defining corresponding relations, such as synonyms in the domain ontology. A semantic relationship named “EquivalenceClass”, which is defined as two classes that can be interpreted as being equivalent or sharing the same instances, as equality may be utilised to devise synonymous classes (McGuinness, 2004). Thus, each topic (class) is assigned one or more alternative topic names (classes) using semantic relationship “Equivalence To” in OWL ontology. The following OWL code shows the definition of this example via the EquivalentClasses relation.

```
 <EquivalentClasses>
    <Class IRI="#IF Statement"/>
    <Class IRI="#Conditional"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class IRI="# Selection statements "/>
    <Class IRI="# Conditional "/>
</EquivalentClasses>
```

5.5.1.2 Ontology Validation

Once the ontology for APELS system was developed using the protégé editor, it is important that the ontology be evaluated to ensure that it has a large coverage of the terms used in the chosen domain. The first process for ontology validation is to select appropriate computer science ontologies to be compared to the APELS ontology. Once the ontology is chosen for comparison purposes, the next step would be to select the concepts from ACM/IEEE 2013 report that would be compared to the concepts in APELS ontology and the other Computer Science ontologies. After reviewing the literature for the most appropriate computer science ontology to be used for comparison with APELS ontology, the Computer Science ontology (CS) (Otto, 2008) and Association for Computing Machinery-Computer Classification System ontology (ACM-CCS) (Gasevic et al., 2011) were selected based on two important criteria, which are: the ontologies covers the concepts in the computer science domain and they are written in OWL.

5.5.1.2.1 Term Extraction

The Stanford parser tool (Manning et al., 2014) is used only for ontology validation to extract all terms form the ACM/IEEE 2013 report in order to generate a list of terms that would be compared with the concepts in APELS, CS, and ACM-CCS. The Stanford parser tool can read various forms of plain text input and generate semantic representation in XML format, including all tokens with the PoS tagged. Moreover, a list of terms in APELS, CS, and ACM-CCS were extracted using OWL concepts extraction algorithm (see Section 4.4.1.5). The algorithm was developed to extract the concepts or classes that are represented in a specific domain throughout the ontology domain.

5.5.1.2.1 Terms to Concepts or Classes Matching

Two-hundred and six terms have been extracted from ACM/IEEE (2013) report using Stanford parser tool that covers the computer science curriculum. These terms will be mapped to concepts in APELS, CS, and ACM-CCS in order to verify the ontologies that offer better term coverage of the computer science domain. Therefore, an algorithm was developed to compute the percentage of a number of matches among the number of terms extracted from ACM/IEEE (2013) report and number of concepts extracted from APELS, CS, and ACM-CCS. The percentage of matching is calculated based on the following formula:

$$S(C,V) = \frac{sc}{n} * 100$$

Where *sc* is the number of concepts extracted from APELS, CS and ACM-CCS that are matching to terms in ACM/IEEE (2013) report and *n* is the number of terms extracted from ACM/IEEE (2013) report. The algorithm used for the number of matches of terms in the ACM/IEEE report and the concepts in domain ontology is given in Figure 5-3

Algorithm: Percentage of total match

Input: total terms of ACM/IEEE report, total Classes of domain ontology

Output: Percentage of the total match between terms in ACM/IEEE report and concepts in domain ontology

1. Begin

2. Declare no of match sc; sc = 0;

3. For each Ontology classes:

   Compare classes with all concepts in ACM/IEEE reports

   If classes of domain ontology = concepts in ACM/IEEE reports then sc = sc +1

4. END For

$$\text{Percentage of total match}(\text{Classes}, \text{Concepts}) = \frac{\text{sc}}{\text{n}} * 100$$

Figure 5-3 The percentage of total match algorithm

APELS has a total of 85 classes while CS has 28 classes and ACM-CSS has 96 classes. Figure 5-4 displays the percentage of the number of matches in APELS ontology which is (41.26 %); this means that the APELS covers 85 of 206 terms in the ACM/IEEE (2013) report, while the percentage of the number of matches in CS is (10.68 %) which means that CS covers 22 of 206 terms in ACM/IEEE 2013 report, and the percentage of the number of matches in ACM-CSS is (33.50 %) which means that ACM-CSS covers 69 of 206 terms in ACM/IEEE 2013 report

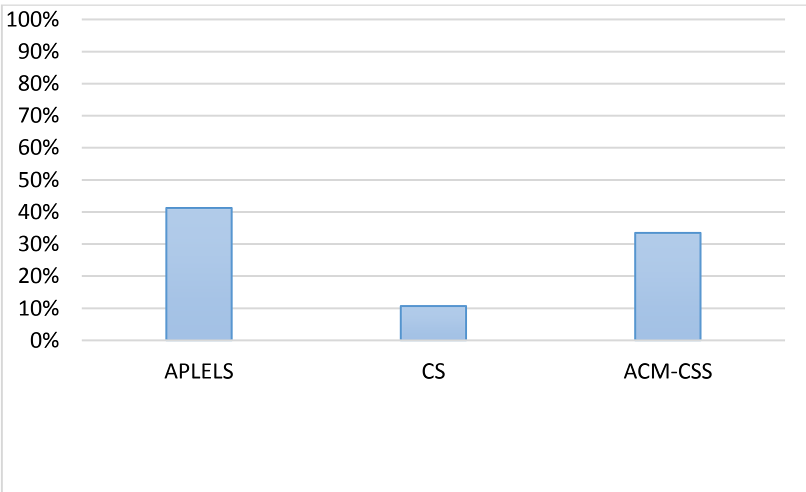

Figure 5-4 The percentage of number of matches in APELS, CS, and ACM-CSS

Comparing CS with ACM-CCS, it becomes evident that APELS returns the highest total match which has a better coverage of terminologies used in the ACM/IEEE computer curriculum (2013) report.

5.5.1.3 The Matching Process

After completing the development of the ontology and exporting the OWL file from the Protégé editor, an approach named "the matching process" was proposed that uses Dice coefficient as similarity measure to computes the similarities between the ontology concepts that are represented in the ACM/IEEE Computer Science Curriculum and the values of the elements extracted from the websites (as described in section 4.4.1.3), in order to retrieve relevant websites that satisfy the learners' needs.

A set of methods and functions have been developed in the matching process, in order to extract the number of websites that are relevant and contain the appropriate information of the specific domain as needed by the learner. For example, the method "WebsiteToXhtmlConversion" is used to transform HTML documents to XHTML to provide the information in a friendly accessible format and easier for extraction and comparison. Moreover, the method "ExtractWebsiteElements" is utilised to extract XHTML elements and attribute values and saved in a vector denoted as $V = [V_1, V_2,...,V_n]$. In addition, the method "OwlFileConcepts" is utilised to extract the OWL concepts that are represented in a Computer science curriculum through the domain ontology and saved in a vector denoted as $C = [C_1, C_2,...,C_m]$. These concepts will be used to determine similarities with the XHTML elements and attribute values extracted.

Furthermore, several rules have been developed to support the matching process using Dice coefficient. First, the system will check whether the ontology concepts in C that are represented in the ACM/IEEE Computer Science Curriculum are matching with values in V that are extracted from the website. Otherwise, the system will check whether the values extracted from the website is one of the topic name synonyms in the ontology, by retrieving all the synonyms of the topic name that have been defined in the ontology. The method "getAllSynonyms" is used to return all the synonyms of a topic name that is defined in the ontology. For example, if the topic name defined in the website is the "IF statement", while the topic name identified in ontology is "Conditional", the "getAllSynonyms" method will return "Selection statements" and "IF statement" as the synonyms of the topic name "Conditional". Then, the system will match "IF Statement" which appears in the synonyms of topic name "Conditional". Since the topic name "IF Statement" in the website has been matched with one of synonyms of "Conditional", then the system returns true.

5.5.1.4 Implementation

This section presents the way the various technologies are applied to support the implementation of the APELS architecture. The Components used to design APELS are summarized in Figure 5-5.

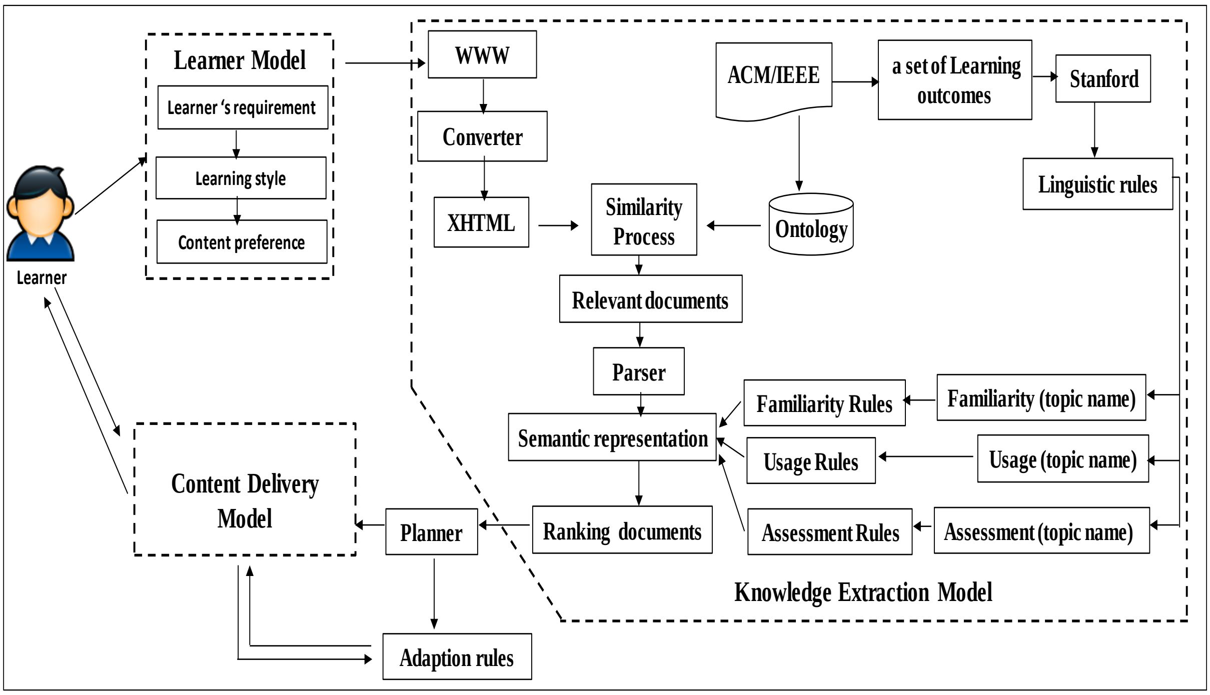

Figure 5-5 Components used in the APELS architecture

As explained before, the ontology was used in the relevance phase to structure the knowledge domain of a pre-selected area of the Computer Science Curriculum (Sahami et al., 2013), in order to extract relevant learning resources from the Web. This section illustrates how the relevance phase is applied to automatically extract the relevant learning material from the freely available resources on the Web. Two learning modules have been selected to be tested in our system. The first module is “Fundamental Programming Concepts”, which is defined as a Knowledge Unit (KU) in CS 2013, with a minimum learning time of 10 hours, whereas the second module is “Algorithms and Data Structure” with a minimum learning time of 12 hours. The KU Fundamental Programming Concepts include the set of topics: Structure of Program, Variables, Expressions, Conditional, Control Structures, Functions, File Input and Output, and Concept of Recursion. The KU of Algorithms and Data Structure include a set of the following topics: Concept of algorithm, a Divide-and-Conquer Algorithm, Iterative Algorithm, Recursion, String, Tree, Stack, Queue, Graph, Array, and Linked List.

The APELS system returns a list of websites for the Fundamental Programming Concepts module ranking them according to the highest similarity score as shown in Table 5-1. In addition, Table 5-1identifies that the website (*www.cal-linux.com/tutorials/)* is the most similar website to the ontology concepts in the OWL.

Table 5-1 The matching similarity for fundamental programming concepts to OWL file

N	www	OWL Concepts Extracted	No of Elements and attribute values Extracted	$\mathbf{C \cap V}$	$\mathbf{C + V}$	$\frac{2\,|\mathbf{C} \cap \mathbf{V}|}{|\mathbf{C}| + |\mathbf{V}|}$
1	www.cal-linux.com/tutorials/	8	20	4	28	0.29
2	www.learn-cpp.org/	8	38	5	46	0.22
3	www.penguinprogrammer.co.uk/	8	41	5	49	0.20
4	www.tenouk.com/cncplusplustutorials.html	8	52	6	60	0.20
5	www.tutorialcup.com/cplusplus/index.htm	8	55	6	63	0.19
6	www.cplusplus.com/doc/tutorial/	8	57	6	65	0.18
7	www.studytonight.com/cpp/	8	77	6	85	0.14
8	www.w3schools.in/cplusplus/	8	83	6	91	0.13
9	www.cprogramming.pickatutorial.com/	8	107	7	115	0.12
10	www.c4learn.com/cplusplus/cpp-history/	8	99	6	107	0.11
11	www.exforsys.com/tutorials/c-plus-plus.html	8	97	6	105	0.11
12	www.noobtuts.com/cpp	8	29	2	37	0.11
13	www.cprogramming.com/tutorial	8	116	7	124	0.11
14	www.programiz.com/cpp-programming	8	73	4	81	0.10
15	www.functionx.com/cpp/	8	104	5	112	0.09
16	www.tutorialspoint.com/listtutorials/c-and-c++/1	8	156	5	164	0.06
17	www.deitel.com/Tutorials/Freetutorialsandarticles/tabid/1575/Default.aspx#CPLUSPLUS	8	102	2	110	0.04
18	www.en.cppreference.com/w/cpp/language	8	227	4	235	0.03

The results from Table 5-1 indicate that the websites (www.*cal-linux.com/tutorials/*) and (*www.learn-cpp.org/)* have the highest similarities to OWL file than the other websites. The website (*www.cal-linux.com/tutorials/*) has a similarity score of (0.29), which was computed using the Dice coefficient as described in section 4.4.1.6. The ontology concepts of the Fundamental Programming Concepts module saved in vector C is 8, and the values extracted from this website saved in vector V is 20. The number of matches between the two vectors (4) is divided by size of two vectors (C+V=28). This is also true for the website (*www.learn-cpp.org/*) where the similarity scores (0.22). Therefore, according to Dice coefficient these two websites are the most relevant websites to ontology concept in the OWL. On the other hand, the other websites have poor ranking indicating that the similarity measures with the required topics are low. For example, the website (*www.en.cppreference.com/w/cpp/language*) obtained a low score of (0.03), which is not good enough to be proposed as good learning material.

The key concepts of the module are enriched in the OWL file as shown in the right-hand side of Figure 5-6, and links extracted from the website (*www.cplusplus.com/doc/tutorial)* on the left-hand side to compute their similarities to the ontology concepts. The relevant links that match to OWL depicted in the red boxes while the un-matched links are shown in black boxes.

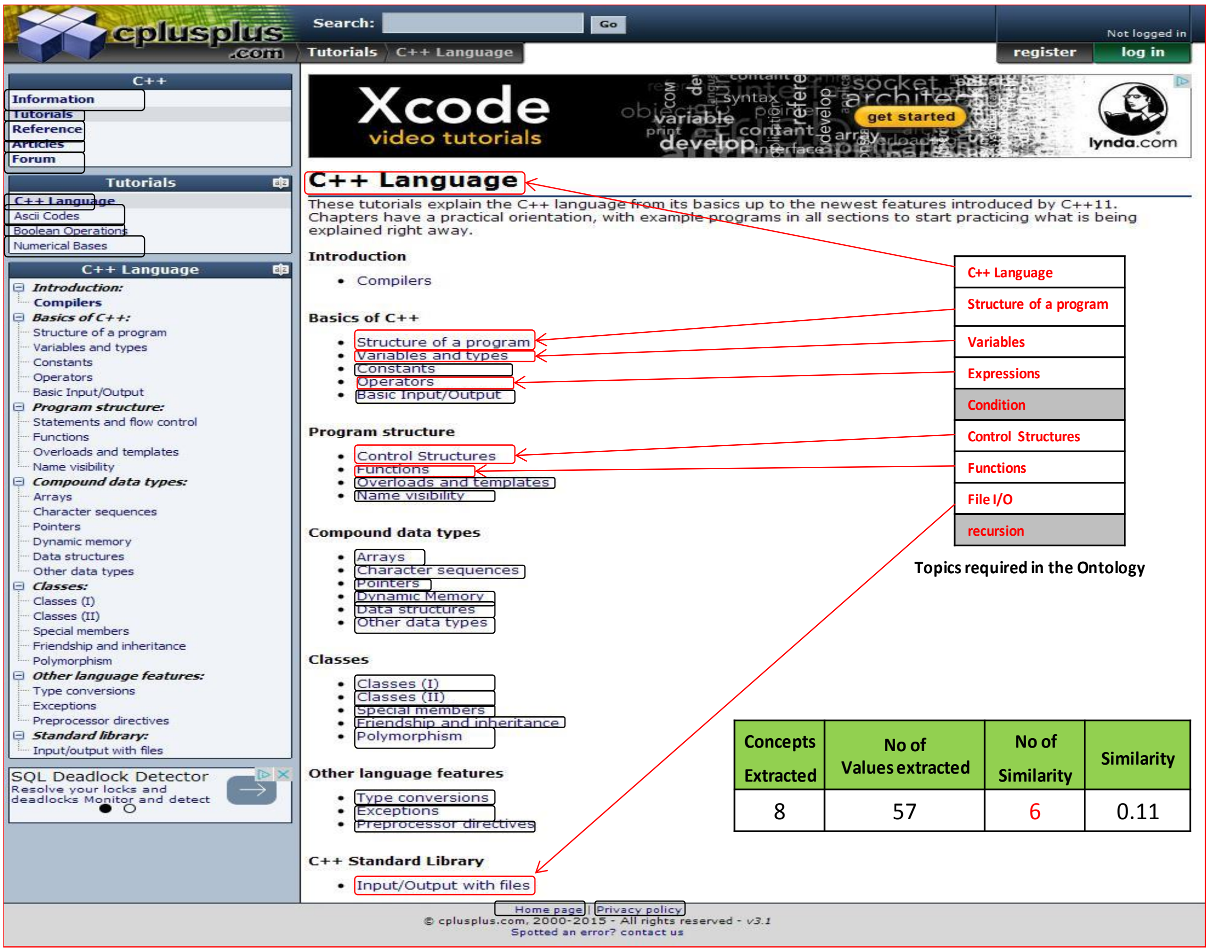

Figure 5-6 The similarity of OWL concepts to www.cplusplus.com/doc/tutorial

As mentioned previously, Algorithms and Data Structure is the second module used in this case study. Similarly, websites retuned for this module by the APELS system were ranked according to the best similarity as given in Table 5-2. The following websites (*www.algolist.net/Data_structures/*) and (*www.cpp.edu/~ftang/courses/CS240/notes.htm*) are the most similar websites to the ontology concepts in the OWL.

Table 5-2 The matching similarities for algorithms & data structure to OWL file

N	WWW	OWL Concepts Extracted	No of Elements and attribute values Extracted	$\mathbf{C} \cap \mathbf{V}$	$\mathbf{C} + \mathbf{V}$	$\frac{2\,\lvert \mathbf{C} \cap \mathbf{V} \rvert}{\lvert \mathbf{C} \rvert + \lvert \mathbf{V} \rvert}$
1	www.algolist.net/Data_structures/	11	19	8	30	0.53
2	www.cpp.edu/~ftang/courses/CS240/notes.htm	11	14	6	25	0.49
3	www.eecs.qmul.ac.uk/~mmh/DCS128/notes/learningOutcomes.html	11	12	3	23	0.26
4	www.cs-fundamentals.com/data-structures/introduction-to-data-structures.php	11	31	7	42	0.33
5	www.learn-cpp.org/	11	34	5	45	0.22
6	www.tutorialspoint.com/data_structures_algorithms/	11	75	9	86	0.21
7	www.radford.edu/~nokie/classes/360/	11	55	6	66	0.18
8	http://tekslate.com/tutorials/datastructures-tutorials/#tutorials	11	46	5	57	0.18
9	www.studytonight.com/data-structures/	11	52	5	63	0.16
10	www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_5/ data_structures/miniweb/pg2.htm	11	39	4	50	0.16
11	www.cprogramming.com/algorithms-and-data-structures.html	11	54	5	65	0.15
12	www.staff.ustc.edu.cn/~csli/graduate/algorithms/book6/toc.htm	11	46	4	57	0.14
13	www.people.cs.aau.dk/~normark/prog3-03/html/notes/theme-index.html	11	34	3	45	0.13
14	www.seas.gwu.edu/~csci133/fall04/	11	29	2	40	0.10
15	www.personal.kent.edu/~rmuhamma/Algorithms/algorithm.html	11	155	7	166	0.08
16	www.en.wikibooks.org/wiki/Algorithms	11	119	4	130	0.03
17	www.en.wikiversity.org/wiki/Introduction_to_Algorithms	11	90	1	101	0.02

The website (*www.algolist.net/Data_structures/*) has a similarity score of (0.53), where the ontology concepts of the Algorithms & Data Structure module saved in vector C is 11, and the values extracted from this website saved in vector V is 19. The number of matches between the two vectors as determined 8, divided by size of two vectors (C+V=30). Likewise, for the website (*www.cpp.edu/~ftang/courses/CS240/notes.htm*) where the similarity scores (0.49). Nonetheless, the website (*www.en.wikiversity.org/wiki/Introduction_to_Algorithms*) has the lowest similarity as it obtained a score of (0.02) indicating that it is of less relevance to ontology concepts in the OWL.

5.5.2 The Ranking Phase

This section describes the implementation of the learning outcomes validation approach to ensure that the selection of the previous relevant websites (as shown in Table 5-1 and Table 5-2) will enable learning according to the learning outcomes set by standard curricula.

The approach will use a linguistic knowledge / features of the words using NLP tools to extract significant key phrases and keywords that represent each document, in order to decide which website satisfies the learning outcomes. Additionally, eight linguistic rules have been designed to capture key phrases and keywords based on finding linguistic patterns in dependency relation and part of speech using the Stanford English Parser. Also, a set of keyword based rules is used to seek whether the content provides instance keywords belonging to a specific programming languages. These rules are employed to identify Familiarity, Usage, and Assessment tasks which have been well explored within the Computer Science domain. The following sections explain how these rules have been used for extracting significant key phrases and keywords from content that would satisfy learning outcomes.

5.5.2.1 Familiarity Rules

Several rules were defined to extract significant key phrases and keywords from the Web that would satisfy the Familiarity task. Two rules are employed to extract syntactic structure of sentences include noun phrase followed by verb "to be" expressed as "is" and "are" such as in the phrases "variable is" and "algorithms are". These expressions will help a student to understand what a concept is or what it means. As we mentioned before, the Stanford parser is used to generate semantic representation which includes PoS tagged and a typed dependency representation for each sentences in XML format. The system extracts parse tag that identifies the syntactic structure or grammatical relationship for each sentence.

Figure **5-7** illustrates syntactic structure of the sentence “algorithm is a list of steps to follow in order to solve a problem.” as an example of applying PoS tagger, which classify each word it into verb, noun, adjective, adverb etc.

<parse> (ROOT (S (NP (NN algorithm)) (VP (VBZ is) (NP (NP (DT a) (NN list)) (PP (IN of) (NP (NNS steps) (S (VP (TO to) (VP (VB follow) (SBAR (IN in) (NN order) (S (VP (TO to) (VP (VB solve) (NP (DT a) (NN problem)))))))))))) (. .))) </parse>

Figure 5-7 Example of key phrases extraction

The system will extract the pattern of the token with the noun tag (NN) in the topic name (algorithm) from the ontology and then check if it is followed by the pattern of token with the verb tag (VB (VBZ) ("is”). The following linguistic rules (Rule 1, Rule2) are applied to find number of syntactic structure of sentences (sc) including a noun phrase followed by the verb "to be".

Rule 1

IF "(NN topic name) (VP (vbz is)) " THEN sc=sc+1 ;

Rule 2

IF "(NNS topic name) (VP (vbp are))" THEN sc=sc+1;

Two rules are designed to extract other kinds of key phrases that also satisfy the Familiarity task. In ACM/IEEE Computing curriculum, the Bloom’s taxonomy uses a set of special action verbs to create learning outcomes (Sahami et al., 2013). These action verbs are classified into three tasks Familiarity, Usage, and Assessment as shown in Figure 5-8. For example, action verbs such as define, describe, discuss, explain, and identify are used for the Familiarity task.

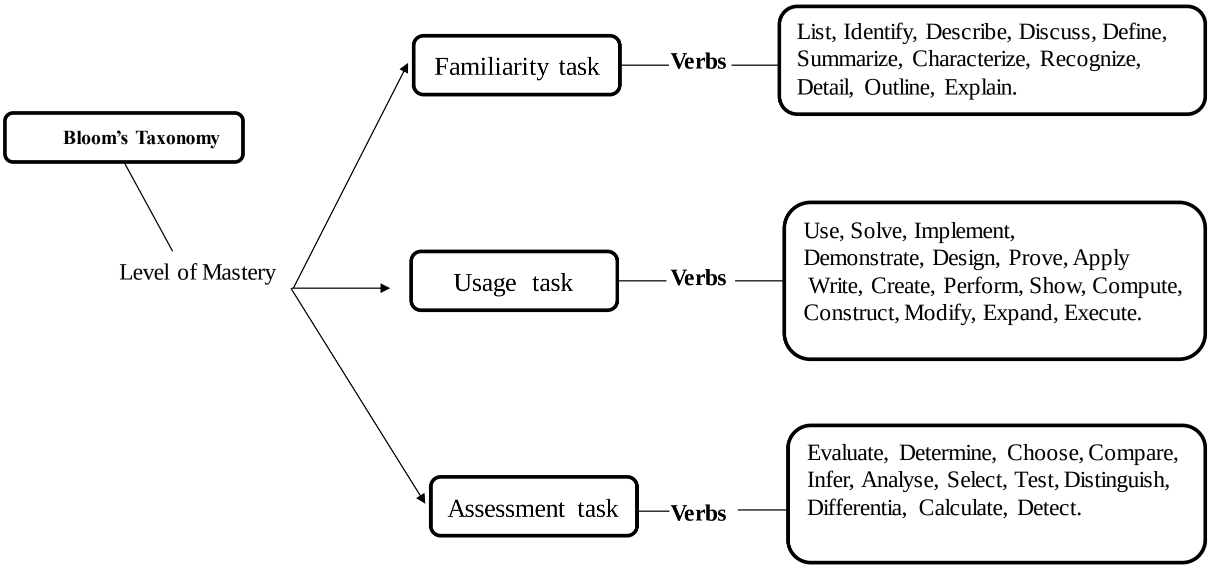

Figure 5-8 The Bloom's taxonomy action verbs in ACM/IEEE Computing curriculum

Thus, a rule was designed to extract the potential relationship between the action verbs associated with Familiarity task and topic name using dependency relations. The following is an example for illustrating the significant key phrases, which comprises of the relationship between the action verb and a topic name as shown in Figure 5-9.

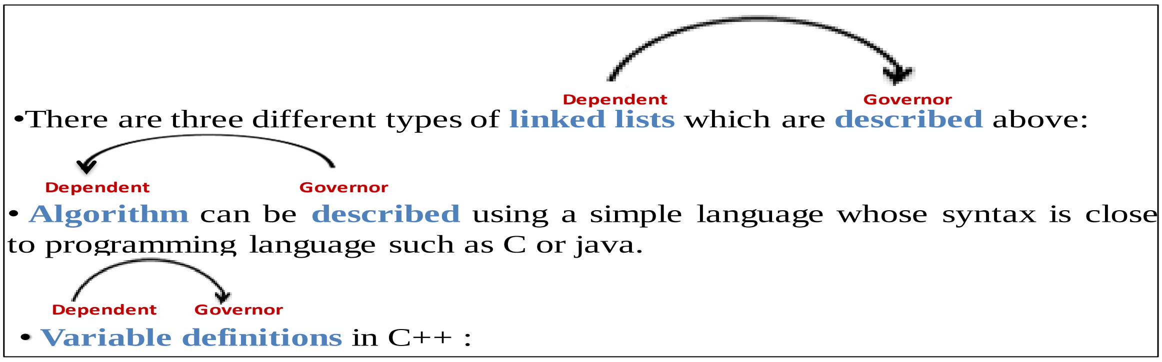

Figure 5-9 Example of key phrases extraction using dependency relation (Familiarity)

To extract key phrases from the text, two types of dictionaries are used. The action verbs dictionary that contains the action verbs associated with the Familiarity task and the topic name synonym dictionary whose terms are retrieved from the ontology. Furthermore, after parsing each sentence in the document using the Stanford parser, the system extracts the key phrases where the word defined between governor dependency tags, is an action verb associated to familiarity and the word defined between dependent tags is the topic name. Key phrases also can be found in opposite arrangement where the word defined between governor dependency tags, is topic name and the word defined between dependent tags is an action verb. The following two linguistic rules (Rule 3 and Rule 4) are applied to find number key phrases (fdr), which include the potential relationships between the action verbs associated with Familiarity task and the topic name.

Rule 3

IF "/dep(/governor = actionVerbs[FamiliarityActionverbs] / dependent = topic name[Ontology concepts])" THEN fdr=fdr+1;

Rule 4

IF "/dep(/governor = topic name[Ontology concepts] / dependent = actionVerbs[FamiliarityActionverbs])"; THEN fdr=fdr+1;

Moreover, the Familiarity task concerns the basic understanding of a concept. It provides an answer to the question “What do you know about this?”. Accordingly, using the expressions such as "For example" or "For instance" in the content will help the reader to understand the content more clearly, instead of providing ambiguous overviews. Thus, the fifth and sixth rules are used to seek whether the content has terms such as” for example” or” for instance”. In addition, a PoS tagger is used to tag each word in the text. The system will extract the pattern of token with the noun tag (NN) and check if token is “example”. The fifth and sixth linguistic rules are applied to find number terms such as “example” or “instance” (ex) in the text.

Rule 5

IF ("NN example") or ("NN instance") THEN ex=ex+1;

Rule 6

If ("NNS examples") THEN ex=ex+1;

5.5.2.2 Usage Rules

A student with a Usage task should be able to write a program properly using specific concepts. It provides an answer to the question "How to use it?". Thus three rules were designed to extract significant key phrases and keywords from the Web that would satisfy the Usage task. two rules are utilised to extract the potential relationship between the action verbs associated with the Usage task and the topic name using dependency relations. The following is an example for illustrating the significance of the key phrases, which comprise the relationship between the action verb and the topic name as shown in Figure 5-10.

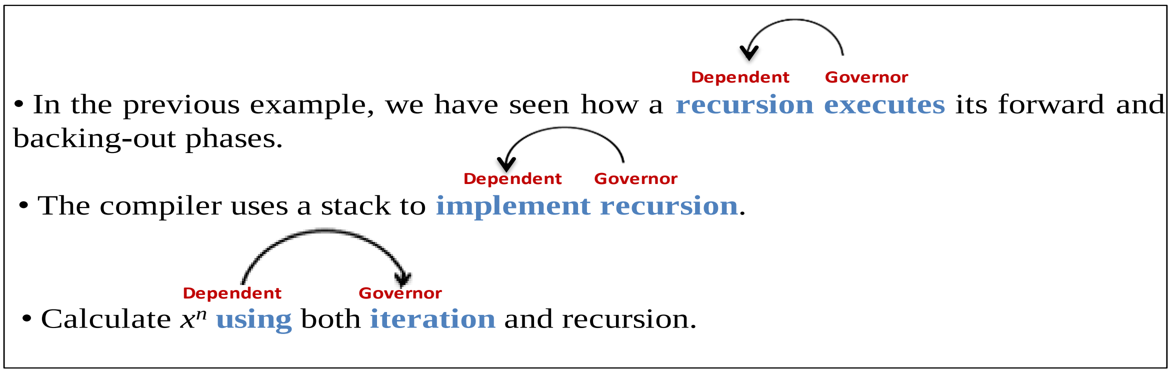

Figure 5-10 Example of key phrases extraction using dependency relation (Usage)

Rules used to find the relationship between the action verb and the topic name in the Familiarity task were applied to extract usage-related key phrases. The word defined between governor dependency tags, is an action verb associated to usage and the word defined between dependent tags is the topic name. Key phrases also can be found in an opposite arrangement where the word defined between governor dependency tags, is a topic name and the word defined between dependent tags is an action verb. The following two linguistic rules (Rule 7 and Rule 8) are applied to find a number key phrases (udr), which include the potential relationships between the action verbs associated with the Usage task and the topic name.

<u>Rule 7</u>

IF "/dep(/governor= actionVerbs[UsageActionverbs] / dependent = topic name [Ontology concepts])" THEN udr=udr+1;

Rule 8

IF "/dep(/governor= topic name Ontology concepts / dependent = actionVerbs[UsageActionverbs])" THEN udr=udr+1;

Extract instance keyword pattern

A set of keyword based rules is used to seek whether the content provides instance keywords belonging to a specific programming languages. These instances keywords will teach the learner how write programs properly using specific concepts. For instance, a block of code using the "while loop" statement in C++ program as shown in Figure 5-11, can help the learner to understand how to use the "while" concept in programming.

```
int k = 1;

while (k < 10)
   {
   k = 2 * k;
   cout << k << endl;
   }
```

Look at the grade3.cpp program as a more practical example. It prompts the user to repeatedly
the user enters the nonsense grade of -1. The following is an outline of the structure of the WHI

```
cin >> NumberGrade;

while (NumberGrade != -1)
   {
   LetterGrade = ConvertGrade(NumberGrade);
   cout << "Letter grade is: " << LetterGrade << endl;

   cin >> NumberGrade;
   }
```

Before we reach the loop we read in the first numeric grade. As long as it is not the special endi
screen. Before looping around to try the condition again, we read in a new numeric grade so tha
a -1.

This special -1 value is called a **sentinel**. A sentinel is any special value used to mark the end o
This pattern is used so often that it should be memorized so that you do not have to reinvent it.

```
Get the first data item;

while (item != sentinel)
   {
   Process the item;
   Get another data item;
   }
```

Figure 5-11 Example of Usage task

To extract instance keywords, it is necessary to initially create a file that contains all the pattern syntax of the instance keywords for a specific programming language. For example, the pattern syntax for a While statement in C++ language is '/while[\\ \\s\\;]*\\([^\\;\\:\\,\\)]+\\)'. The system will extract instance keywords from the content by selecting the pattern syntax of the instance keywords. Subsequently, the system checks whether the pattern syntax extracted matches with the pattern syntax instances of the keywords in the file. The set of pattern syntax of instance keywords for C++ language are given in Appendix A.

5.5.2.3 Assessment Rules

For the Assessment task, the student should be able to select the appropriate concept or method among different methods. It provides an answer to the question "Why would you do that?". Furthermore, the student is able to justify the selection of a particular approach to solve a problem. In this case, the system applies Familiarity rules as described in section 5.5.2.1 and Usage rules as described in section 5.5.2.2 for each method or concept. The content produced after applying these rules will help the learner to select the appropriate method or concepts among different methods. Table 5-3 shows the rules that are used in the APELS system for extracting significant key phrases and keywords from documents, in order to decide which website satisfies the Familiarity, Usage, or Assessment tasks.

Table 5-3 Rules used to identify Familiarity, Usage, and Assessment tasks

	Tasks	Examples of learning outcomes	Rules
Familiarity	Learner understands what a concept is or what it means?	Identify and describe uses of iteration	1. POS (NN+VB) 2. POS (NNS+VB) 3. Dependency relation (VB, NN) 4. Dependency relation (NN, VB) 5. POS (NN) 6. POS (NNS)
Usage	Learner is able to use or apply a concept in a concrete way	Write program that use iteration	7. Dependency relation (VB, NN) 8. Dependency relation (NN, VB) In addition, Extract instance keyword pattern.
Assessment	Learner is able to justify the selection of particular approach to solve a problem	Determine which methods of iteration is best for given problem	Same as above.

5.5.2.4 Implementation

The learning outcome validation process was added to ensure that the selected websites will enable learning according to the learning outcomes set by standard curricula. The APELS system produced a list of websites for “Fundament Programming Concepts” and “Algorithms and Data Structure” and those with the highest similarity were chosen (see section 5.5.1.3). Now, the appropriateness of the content of these selected websites will be assessed against learning outcome as described by ACM/IEEE Computing Curricula.

As a first starting point to the topic, it is important for the learner to gain basic knowledge about the topic to be able to answer the question: what do you know about it? This is usually achieved by providing enough definitions, descriptions and/or examples, which is named as Familiarity tasks in the APELS system. A score was given to the task based on the number of occurrence of key phrases related to Familiarity in the content. The key phrases include the number of definitions of topic; action verbs related to Familiarity tasks; topic names and their relationships; as well as important terminology which helps the learners to fully comprehend different concepts like the word “for example”.

As an example, the learning outcomes “define and describe the variable” for the “Fundamental Programming Concepts” module was assessed with Familiarity. In this case, the APELS system, will select a set of URLs from different websites with a topic named “variable” being included in these URLs. Then the system applies three rules that were described in Familiarity rules in section 5.5.2.1 to extract a number of significant key phrases and keywords from the content. As some learners might prefer the topic to be explained practically with more examples rather than being stated simply as definitions, it was important to use two rules that were described in the Usage rules in section 5.5.2.2 to extract usage-related key phrases from the content.

Table 5-4 Results of keywords and key phrases extraction that satisfy Familiarity for the learning outcome “Define and describe variable”

WWW	abstract concepts and explanation	Concrete, practical
	-POS (NN+VB) -Dependency relation (VB, NN) -POS (NN)	Dependency relation (VB, NN) Extract Keyword Pattern
	Occurrences	
www.cplusplus.com/doc/tutorial/variables/	**25**	**21**
en.wikibooks.org/wiki/C_Programming/Variables	**19**	**23**
www.tenouk.com/Module2.html	**12**	**3**
www.tutorialspoint.com/cplusplus/cpp_variable_types.htm	**11**	**8**
www.penguinprogrammer.co.uk/c-beginners-tutorial/variables	**8**	**29**
www.c4learn.com/cplusplus/cpp-variable-naming/	**7**	**2**
cprogramming.pickatutorial.com/variables_datatypes.htm	**6**	**4**
www.tutorialcup.com/cplusplus/variable-types.htm	**5**	**4**
www.w3schools.in/cplusplus/variables/	**3**	**3**

Table 5-4 shows that the system ranks the websites according to the Familiarity score, which was calculated as a number of occurrences of keywords and key phrases in the content. The results shown in Table 5-4 indicate that the website (*www.cplusplus.com/doc/tutorial/variables*) occupies the first rank (25) because it has the highest number of keywords and key phrases, extracted in the content that satisfies the Familiarity learning outcome. Despite the importance of abstract concepts and explanation for understanding the topic (variable), practical example is another key element, which was included in the system evaluation. In this regard, the same website sustains a good score (21) for usage task. On the other hand, the website (*www.penguinprogrammer.co.uk/c-beginners-tutorial/variables*) occupies the fifth rank according to the Familiarity because it contains a low score (8) of concept-related key phrases. However, it contains high score (29) according to concrete/practical aspect, which is reflected by a high number of keywords and key phrases related to usage and implementation of the knowledge around the variable topic. This particular website might be of good quality content for learners, who prefer fewer definitions and more practical examples.

On the other hand, the other websites show poor rankings, as the number of keywords and key phrases extracted is low. For example, the website (*www.w3schools.in/cplusplus/variables*) has only 3 keywords and key phrases, which is not sufficient to select it based on the learning outcomes which is referred to as “Familiarity”.

The second example of a Familiarity learning outcome used with the Algorithms and Data Structure module is: “Discuss the importance of algorithms in the problem-solving process”. A learner often advances when the learning content provides key phrases that help him/her to be aware with concept. These key phrases are defined and evaluated in our APELS system as a Familiarity task. A score was assigned to the task based on the frequency of these key phrases in the content of each website. Based on the learning outcome mentioned earlier in this example. The same methodology described in the first example was applied to extract the key words and key phrases related to Familiarity.

Table 5-5 Results of keywords and key phrases extraction that satisfy Familiarity learning outcome “Discuss the importance of algorithms in the problem-solving process”

WWW	abstract concepts and explanation	Concrete, practical
	-POS (NN+VB) -Dependency relation (VB, NN) -POS (NN)	Dependency relation (VB, NN) Extract Keyword Pattern
	Occurrences	
staff.ustc.edu.cn/~csli/graduate/algorithms/book6/chap01.htm	29	4
www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/intro.htm	15	2
en.wikibooks.org/wiki/Algorithms/Introduction	14	2
http://www.tutorialspoint.com/data_structures_algorithms/algorithms_basics.htm	**12**	**3**
www.macs.hw.ac.uk/~pjbk/pathways/cpp1/node32.html	5	1

The results (Table 5-5) showed that according to the Familiarity score the website (*staff.ustc.edu.cn/~csli/graduate/algorithms/book6/chap01.htm*) occupies the first rank (29) because it provided the highest amount of keywords and key phrases, extracted in the content that satisfies the Familiarity learning outcome “Discuss the importance of algorithms in the problem-solving process”. Contrastingly, a low score was obtained with only 5 keywords and key phrases from the website (*www.macs.hw.ac.uk/~pjbk/pathways/cpp1/node32.html*). As a result, no understanding of algorithm concepts will be acquired by the students, as insufficient data is provided to identify and comprehend the algorithms. Nonetheless, it should be highlighted here that none of the websites give a good score regarding the concrete/practical aspect of the content, however, that might be suitable for learners, who are trying to familiarise themselves with the topic without going into the practical aspect of it.

After the learners have acquired the knowledge, they should be able to use or implement it during the learning process. Using or applying knowledge in a concrete way is identified as Usage tasks in our APELS system evaluation. The number of occurrences of the usage-related key phrases such as “using a code to” or “implement variable” was found to get a score in order to rank the websites based on the highest frequency of these key phrases. The key phrases include the number action verbs related to Usage tasks; topic names and their relationships; as well as the pattern syntax of instance keywords that relate directly to the language of the programme. For example, a learning outcome “Write program that uses and implements s Function” for the “Fundamental Programming Concepts” module was assessed with Usage.

First, the APELS system will select a set of URLs, which contain a topic named “Function”. Then the system applies two rules that were described in the Usage task (section 5.5.2.2) to extract a number of key phrases from the content. As some learners might prefer to be provided with more background information about the topic as well, it was important to use the three rules that were described in the Familiarity task (section 5.5.2.1).

Table 5-6 Results of keywords and key phrases extraction that satisfy Usage learning outcome “Write program that use and implement function”

WWW	Concrete, practical	abstract concepts and explanation
	Dependency relation (VB, NN) Extract Keyword Pattern	-POS (NN+VB) -Dependency relation (VB, NN) -POS (NN)
	Occurrences	
www.tenouk.com/Module4.html	**29**	**31**
www.penguinprogrammer.co.uk/c-beginners-tutorial/functions/	**23**	**12**
www.programiz.com/cpp-programming/function	**20**	**14**
www.cprogramming.com/tutorial/lesson4.html	**14**	**10**
http://cprogramming.pickatutorial.com/functions.htm	**12**	**13**
en.wikibooks.org/wiki/C_Programming/Procedures_and_functions	**10**	**18**
www.tutorialspoint.com/cplusplus/cpp_functions.htm	**7**	**19**
www.learn-c.org/en/Functions	**4**	**3**
www.c4learn.com/cplusplus/cpp-functions-introduction/	**3**	**6**
http://en.cppreference.com/w/cpp/language/functions	2	**6**

The system ranks the website according to the Usage score. The results (Table 5-6) indicate that the website (*www.tenouk.com/Module4.html*) occupies the first rank (29) because it has the highest number of key phrases extracted in the content that satisfies the Usage learning outcome. Meanwhile, same website has the highest number (31) according to abstract concepts and explanation aspect, which is reflected by the high number of keywords and key phrases related to understanding and defining the knowledge around the function concept.

On the other hand, the other websites (*www.c4learn.com/cplusplus/cpp-functions-introduction*) and (*http://en.cppreference.com/w/cpp/language/functions*) show poor ranking, as the number of key phrases extracted is low. For example, the website (*http://en.cppreference.com/w/cpp/language/functions*) has only 2 key phrases. As a result, no understanding of function concepts will be acquired by the students, as insufficient information is provided to use and implement the functions.

The second example of Usage learning outcome used with the Algorithms and Data Structure module is: “Write a program that uses and implements an Array”.

Table 5-7 Results of keywords and key phrases extraction that satisfy Usage learning outcome “Write program that use and implement array”

WWW	Concrete, practical	abstract concepts and explanation
	Dependency relation (VB, NN) Extract Keyword Pattern	-POS (NN+VB) -Dependency relation (VB, NN) -POS (NN)
	Occurrences	
http://www.cplusplus.com/doc/tutorial/arrays/	**26**	**22**
cis.stvincent.edu/html/tutorials/swd/basic/arrays/index.html	**17**	**20**
www.tutorialspoint.com/data_structures_algorithms/array_data_structure.htm	**10**	**11**
www.tutorialspoint.com/cplusplus/cpp_arrays.htm	**9**	**7**
https://ece.uwaterloo.ca/~dwharder/aads/Tutorial/1o/	**4**	**25**

The results (Table 5-7) show that the website (*www.cplusplus.com/doc/tutorial/arrays*) was ranked first with a score of 26 because it has the highest number of key phrases, extracted in the content that satisfies the Usage learning outcome. This high score indicates that this website provides good material for a learner who aims to know how use the concept in a concrete, practical way. In contrast, the website (*https://ece.uwaterloo.ca/~dwharder/aads/Tutorial/1o/*) occupies the lowest rank according to the Usage task with a score of 4. However, it has a high score at Familiarity, that might not be the best for students who had a good background about the topic and looking for tools to apply their understanding in a more practical way.

The next level of mastery involves the ability of the leaner not only to use the concept, but also to look at it from different viewpoints and explain why its approach to solve a certain problem is ideal for him/her. This type of learning outcomes are defined as Assessment. As explained before, a student in an Assessment task should be able to select an appropriate approach from different alternatives. It provides an answer to the question “Why would you do that?” An example of an Assessment task that address the learning outcome in the programming concepts module was implemented in the system, which was “Choose appropriate methods of the iteration for a given programming task”. In this case, Familiarity and Usage rules were applied to assess the leaner knowledge around the topic. Then all the methods were identified for iteration concept: for-loops, while-loops, Do-while and the system applied Familiarity and Usage rules on each method. As a result, the student should understand multiple methods for iteration and be able to appropriately select among them.

Table 5-8 Results of keywords and key phrases extraction that satisfy Assessment learning outcome “Choose appropriate methods of iteration for a given programming task “

www	Concrete, practical	abstract concepts and explanation
	Dependency relation (VB, NN) Extract Keyword Pattern	-POS (NN+VB) -Dependency relation (VB, NN) -POS (NN)
	Occurrences	
http://www.tenouk.com/Module3.html	31	14
http://www.cplusplus.com/doc/tutorial/Control/	30	33
www.penguinprogrammer.co.uk/c-beginners-tutorial/loops/	27	4
http://cis.stvincent.edu/html/tutorials/swd/basic/control/repetition/index.html	22	31
http://www.cprogramming.com/tutorial/lesson3.html	18	11
https://cal-linux.com/tutorials/loops.html	16	16
https://turboc.codeplex.com/wikipage?title=LoopsCPP	9	27

In each website shown in the Table 5-8, all three type of iterations have been discussed, the system ranked these websites as described before based on the highest score obtained with the Familiarity and Usage tasks. Interestingly, the website (*www.cplusplus.com/doc/tutorial/Control*) scored very high at both concrete/practical aspect (30) and abstract concepts and explanation aspect (33) tasks suggesting it as a good content for Assessment outcome to understand many aspects of iteration to be able to choose among all different application available.

However, other websites obtained low score on either the practical or abstract concepts aspect. For example, the website (*www.penguinprogrammer.co.uk/c-beginners-tutorial/loops/*) has a low score of 4 at abstract concepts and explanation aspect, whereas, the website (*https://turboc.codeplex.com/wikipage?title=LoopsCPP*) has 9 at concrete/practical aspect. These low scores will lead to a failure in understanding what the topic is or how to use it. Consequently, it will not be suggested to the learner with regards to the assessment task.

5.6 Summary

This chapter reviewed the ACM/IEEE Computing curriculum, which is an internationally recognised and adopted standard in designing computer science related programs. In addition, it described in details the body of knowledge of ACM/IEEE Curriculum which includes as main components; the areas of knowledge, the subsequent units, required learning hours, and learning outcomes. The second part of this chapter introduced the full implementation of the APELS architecture using the field of computer science and the ACM/IEEE curriculum to design two modules (Algorithms and Data Structure and Fundamental Programming Concepts). All the components identified in the architecture were implemented and examples were given to illustrate how they were applied in this specific case study. Finally, to evaluate the learning outcome validation approach, the content of extracted learning material was evaluated against a set of learning outcomes for the two modules were selected using this novel learning outcome validation approach.

Chapter 6 : Evaluation Methodology

6.1 Introduction

The APELS was developed to select learning resources from an enormous number of freely available resources on the Web according to student's constraints, which are the student's need, learning style and learning outcomes. Initially, the system provides an interface where the learner starts filling his/her details in order to create a Username and Password to login into the system. Likewise, the system provides an interface where the learner has the option to select a specific domain. The system asks the learner some questions such as the learner's background, need, and his/her level of ability in a specific area to ensure that any function performed by the system is personalised to this user. Then, the system views the learning style questionnaire (VARK) in order to identify the initial learning style for the learner. The learner then finally chooses his/her content preference for studying a module. Once the details of the learner and his/her chosen area are known, these are saved and submitted to be processed by the knowledge extraction model.

The knowledge extraction model uses a standard search engine to provide a list of websites that are dealing with the specific domain. The extraction model transforms each website which is usually written in HTML to XHTML in order to provide the information in a friendly accessible format and easier for extraction and comparison. Then, the system computes the similarity between the ontology concepts and the XHTML values, in order to filter the websites that are relevant of specific domain as needed by the learner. Then, the system evaluates the topics' contents extracted from the Web against a set of learning outcomes as defined by the standard curricula.

Finally, the system will return a learning plan where it generates a specific code for each module and provides its details including the module title, summary of the programme aim, intended learning outcomes, program structure and the time required to accomplish each component of the module. Figure 4-2 presents a use case diagram of the APELS system to illustrate its functional requirements.

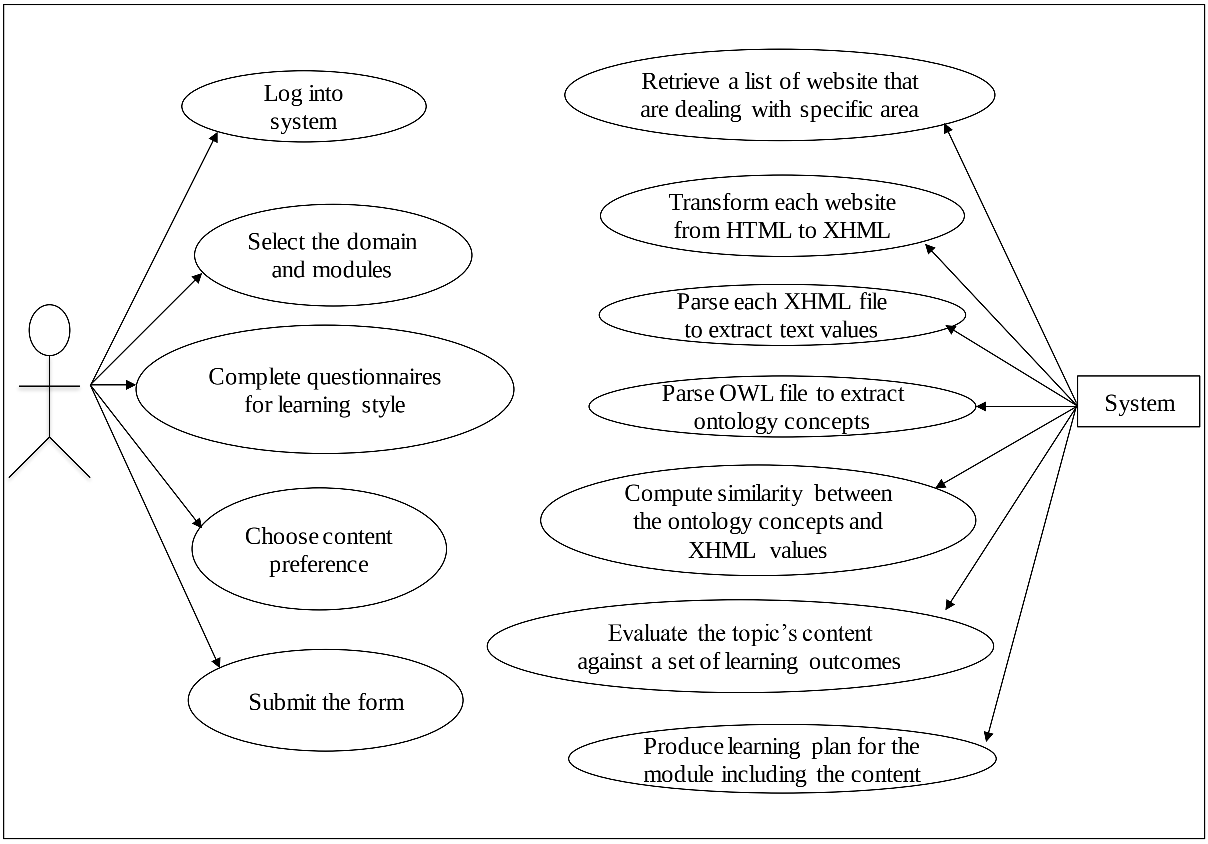

Figure 4-2 Use case diagram of the APELS system

In order to answer the research question "Is it possible to provide personalised learning material to each individual learner using learning content automatically extracted from freely available resources on the Web?", it was essential to carry out an evaluation process/experiment, which specifically tests our hypothesis "APELS can produce the right learning material that suits the learning needs of a learner as teachers would do". To assess the degree to which APELS is successful in achieving this educational objective requires the testing of the following two sub-hypotheses:

1- APELS is usable by the learners and will allow them to provide the right information to determine their backgrounds and needs.

2- APELS can return suitable learning material based on the background information of the learner.

6.2 APELS System Evaluation

To evaluate APELS we have initially aimed at developing a research study/experiment to test the research hypothesis “APELS can produce the right learning material that suits the learning needs of a learner as teachers would do?”. However, it was apparent towards the end of the research, that within the timeframe of the PhD program it will be difficult to get the evaluation results as it would take more than three months for students to participate in a course and a researcher to analyse the results of the study. An attempt to recruit a group of undergraduate students to participate in the experiment has also proved to be very difficult. Hence, the researcher has opted to evaluate the APELS system using experts’ opinions. Ten experts, who are primarily university academic staff members from various disciplines, i.e. computing, mathematics and education, participated in evaluating the system including the following experiments to verify the two sub-hypotheses:

H1. APELS is usable by the learners and will allow them to provide the right information to determine their backgrounds and needs.

The experiment performed to test this hypothesis involves asking the experts to create an account on the APELS system as if they were learners. This will assess the usability aspect of the system specifically the learner’s model.

H2. APELS can return suitable learning material based on the background information of the learner.

The main purpose of the system evaluation was to assess the quality of the produced material. Therefore, qualitative methods were applied to gather and analyse the data required for the evaluation system. A questionnaire was designed to elicit information necessary to evaluate the degree to which the experts were satisfied with the content produced by the system, whether it is of good quality and whether or not it satisfies the learning outcomes for teaching purposes. The questionnaire incorporates an open comment section whereby the experts can state their opinions concerning the content produced by the system.

In addition, at the end of the experiment, a general discussion was held with the experts to subjectively assess the effectiveness of the system by giving their views and making their comments.

6.3 Participants Background and Experience

Ten experts were invited to evaluate the content produced by the system, to assess whether or not it meets the learning outcomes and to express their opinions on the effectiveness of the system. The experts were selected on the basis of their background and valuable experiences. The following gives a summary of the backgrounds of the participants.

Expert 1 is a senior lecturer in Computer Science and Software Engineering with over 15 years experience. He teaches many IT modules including networks, operating systems, and software engineering. He is a member of the higher education academy and won various teaching awards. He is also in charge of developing and teaching the learning skills module. He has used and is an advocate of the use of technology in teaching.

Expert 2 is a senior lecturer in Computer Science and Software Engineering with over twenty years teaching experience. He teaches Programming and Data Structures & Algorithms to undergraduate students in Computer Science and Software Engineering. He has also taught mathematics and Software Quality to Computer Science students.

Expert 3 is a lecturer in Computer Science department with over 10 years experience in teaching in Higher education. He teaches a wide range of core modules in computer science at the university such as Data structures and algorithms**,** Formal development of software systems, and Programming. He also has a good experience in area of Machine learning, pattern recognition, and neural networks**.**

Expert 4 is a senior lecturer in Computer Network Systems with over 15 years experience. He teaches a wide range of core modules in computing science and Computer Network such as Agile Software Development, Software Projects with Agile Techniques, Network Programming and Simulation, Agile Software Project Management, and advanced Programming.

Expert 5 is a senior lecturer in Computer Science and Software Engineering. He is an experienced researcher and developer of virtual environments and their associated systems, with a background in both commercial development and academic research and development. He also teaches a wide range of core modules in C++ programming, Virtual Reality/Virtual Environments, and 3D Computer Graphics.

Expert 6 is senior lecturer in data mining and bio-informatics with over twenty years experience. He teaches a wide range of core modules includes Advanced Databases, Business Intelligence, Web Semantic & Information Retrieval, and Advanced Database Systems.

Expert 7 is a research assistant in computer science department. He has a good experience in designing and development web-based software applications. He also teaches many subjects, such as System analysis, Information Security, Java programming and C++.

Expert 8 is a software programmer in the IT department of a company. He has a good experience and knowledge in software development life cycle, website design & development.

Expert 9 is a teaching assistant in Computer Science in the computer science department. She has research experience in using ontologies for data extraction. She also has developed Database systems for industrial projects.

Expert 10 is a professor of computer science with over 25 years experience in teaching and research. He has taught a large number of Computer Science modules over his career and is a well-respected researcher in the field of data and knowledge Engineering.

6.4 APELS System Training Overview and Background

The evaluation of our system involves holding sessions with domain experts from different academic disciplines; each session lasted for about 30-40 minutes. During the meeting, the experts were given a laptop provided with the system and were asked to use and test the system without assistance from the researcher. The meetings were recorded and observed by the researcher after obtaining the permission from the experts before performing the evaluation and the analysis of the results. To familiarise the experts with our experiment, a full overview and background of the system as well as details of the concepts and system implementation were provided and explained to the experts. The system overview and background given to the expert were as follows:

"This research project is about developing an adaptable and a personalised E-Learning system (APELS) based on freely available resources on the web.

First, *the learner will choose the area s/he wishes to study (a Module) and then through a set of menus the system will take him/her to the detailed area of study.*

Second*, the system captures the learning style of the learner through a questionnaire. We adopted Fleming's VARK learning Model that identifies 4 types of learning style namely, visual, Auditory, R/W and Kinaesthetic. These styles were introduced to the experts. It was also explained that the purpose of capturing the learning style is to help students learn effectively and better. We have also explained that the VARK system is chosen because its questionnaire is shorter, takes less time to fill and the questions it contains are more relevant and concise.*

__Third__, the learner will choose his/her content preference, there are two types of content preference. One version provides some definitions of the concept and fewer examples. While another may start with the fewer definitions and followed some distinct examples.

***Finally,** the system will return a learning plan in a similar way as an academic staff would do for their module specification including their contents.*

The modules are designed based on standard curricula. We have used computer science and the ACM/IEEE curriculum as a case study, which is an international curriculum guideline for computer science domain.

There are various phases that you will be asked to undertake, with Phase 1 being the testing of the system usability, where an account will be created by yourself by using few steps to complete the system's tasks. Following this, Phase 2 is the evaluation of the quality of the produced content, where you are required to answer questions in relation to whether the content of the designed module satisfies the module's learning outcomes. Subsequently, a more general discussion will take place in order to gain more qualitative feedback.

6.5 Methodology and Data Collection

6.5.1 Experiment 1 to Test Hypothesis H1

In order to assess the usability aspect of the system specifically the learner's model, the experts were asked to create an account on the APELS system as if they were learners. Next, the experts were asked to answer the set of questions as used by the VARK learning style to determine the initial learning style of the experts playing the role of a learner in this case. This was followed by the choice of the content preference by the experts (more examples for instance).

6.5.1.1 Create an Account

The first step of the experiment was meant to test the usability of the system. The domain expert has first to create an account (Learner's model) by providing some personal information. This is followed by completing the prior knowledge section and answering the set of the VARK questions to assign the early learning style of the learner. Finally, the learner will choose his/her content delivery preference by selecting either "More definitions and explanation" or "Fewer definitions and more practical examples" options. A screenshot of the completion of the first step is given in Figure 6-1.

Figure 6-1 Create an account in APELS

After creating an account, the domain experts first chose a specific domain, and then they select a module and their expertise level in this module based on their prior knowledge of the area as shown in Figure 6-2.

Figure 6-2 Example for domain selection in prior knowledge page

6.5.1.2 Identify Initial Learning Style

To define the learner's initial learning style, s/he has to answer a set of questions of the VARK learning style system that are provided as part of the APELS system. After completing the questionnaire, the experts were provided with their VARK learning style scores. The scores given by the VARK questionnaire are a mixture of Visual, Aural, Read/Write and Kinesthetic and the highest score was assigned as the initial learning style of the user (see Figure 6-3).

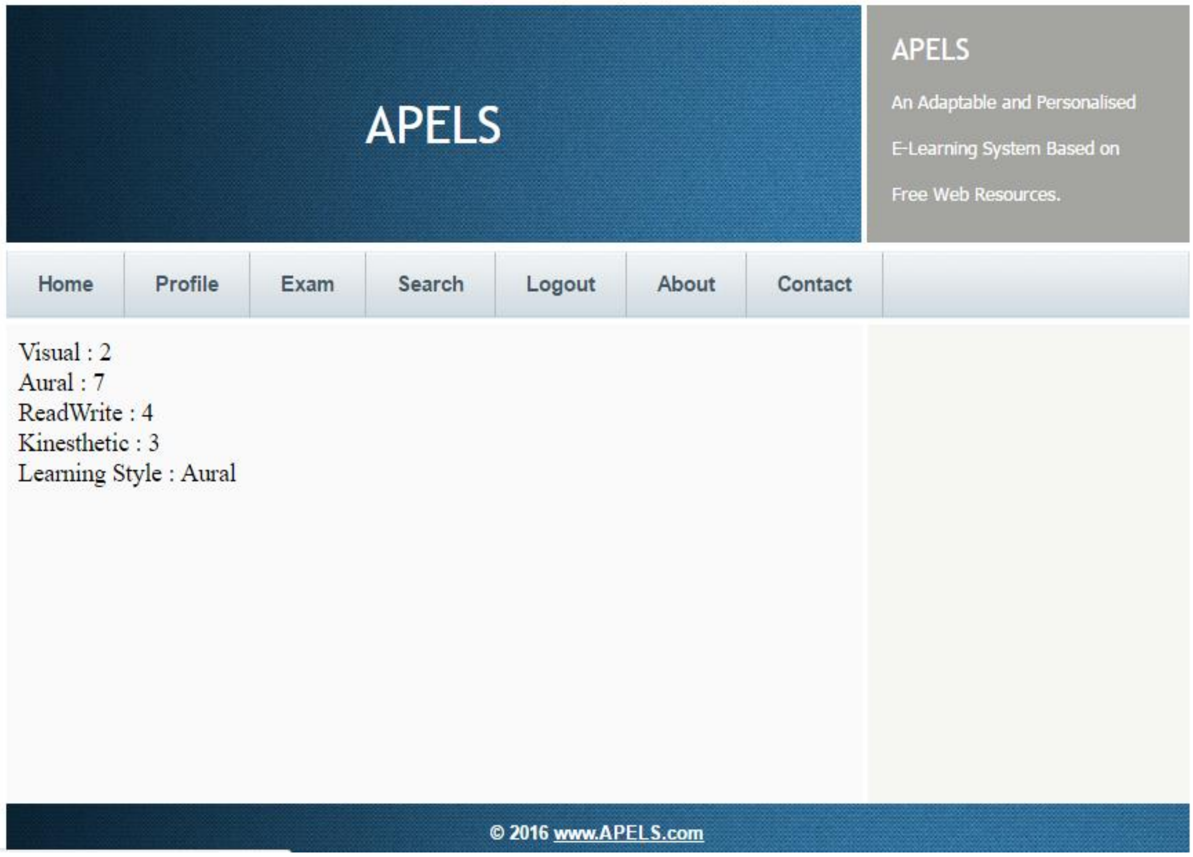

Figure 6-3 Screen results for a user with aural learning style type

6.5.1.3 Create a Sample Learning Module

Once all the required information regarding the learner's chosen area is taken into consideration along with his/her requirements, learning style and learning outcomes, the system produces the module's content and learning plan including aims of the module, learning outcomes and program structure as shown in Figure 6-4.

Module Title :Fundamental Programming

Module Code: CF201

Aims of the Module

The aim of this module is to introduce the student of fundamental programming concepts and enhance their problem-solving. Students will learn the basics of scalar types (Integers, Strings, Booleans) and fundamental control structures in procedural programming (loops, assignment statements, conditional expressions). The module uses the C++ programming language as the implementation environment. This course also will allow the students how to implement file I/O, functions and recursion for solving a problem.

Learning Outcomes

1.Identify and describe uses of primitive data types. [Familiarity]
2.Write programs that use primitive data types. [Usage]
3.Write programs that use standard conditional [Usage]
4.Write program that use iterative control [Usage]
5.Write program that use functions. [Usage]
6.Write a program that uses file I/O. [Usage]
7.Choose appropriate conditional and iteration constructs for a given programming task. [Assessment]
8.Describe the concept of recursion and give examples of its use. [Familiarity]

Program Structure

Topic name	Recommended Link	Learning Hours	Exercise	Evaluation
Conditional	Link	2	Link	Is this useful ?
Loops	Link	2	Link	Is this useful ?
Variables	Link	2	Link	Is this useful ?
Functions	Link	2	Link	Is this useful ?
Recursion	Link	2	Link	Is this useful ?

Figure 6-4 Module specifications page for fundamental programming module

6.5.2 Experiment 2 to Test Hypothesis H2

The main purpose of the system evaluation was to assess the quality of the produced material. Choosing the best teaching material that could suit the learning purposes is always a challenging task for teachers (Ellis, 1997). Predictive and retrospective evaluation can be conducted by teachers to evaluate available learning material. Predictive evaluation is carried out by expert reviewers prior to delivering the course based on specific criteria, represented by a checklist on how to achieve the course outcome (Ellis, 1997). On the other hand, retrospective evaluation is carried out after the material has been used in a teaching context. After that, a decision is made on whether or not the material has worked for learners. Despite the limitations of predictive evaluation represented by the lack of well-defined formula and a subjective nature (Sheldon, 1988), this type of evaluation was employed in this research due to time constrains. Predictive evaluation was performed in this research by involving ten instructors experienced in the field of computer science; they evaluated the quality of the material extracted by APELS on whether it would satisfy the targeted learning outcome as defined by standard curricula. Before this phase took place, a greater level of understanding was provided to the experts in relation to the learning outcomes, as were defined in the ACM/IEEE Computing curriculum (Sahami et al., 2013). In current research, the learning outcomes in ACM/IEEE are defined in terms of three tasks: Familiarity, Usage and Assessment as depicted in Figure 6-6 and explained in more details in Appendix B. The experts are requested to respond to the question "would the content produced by APELS form a good learning material for the learners; a material that could meet one of the targeted learning outcomes, namely familiarity, usage or assessment".

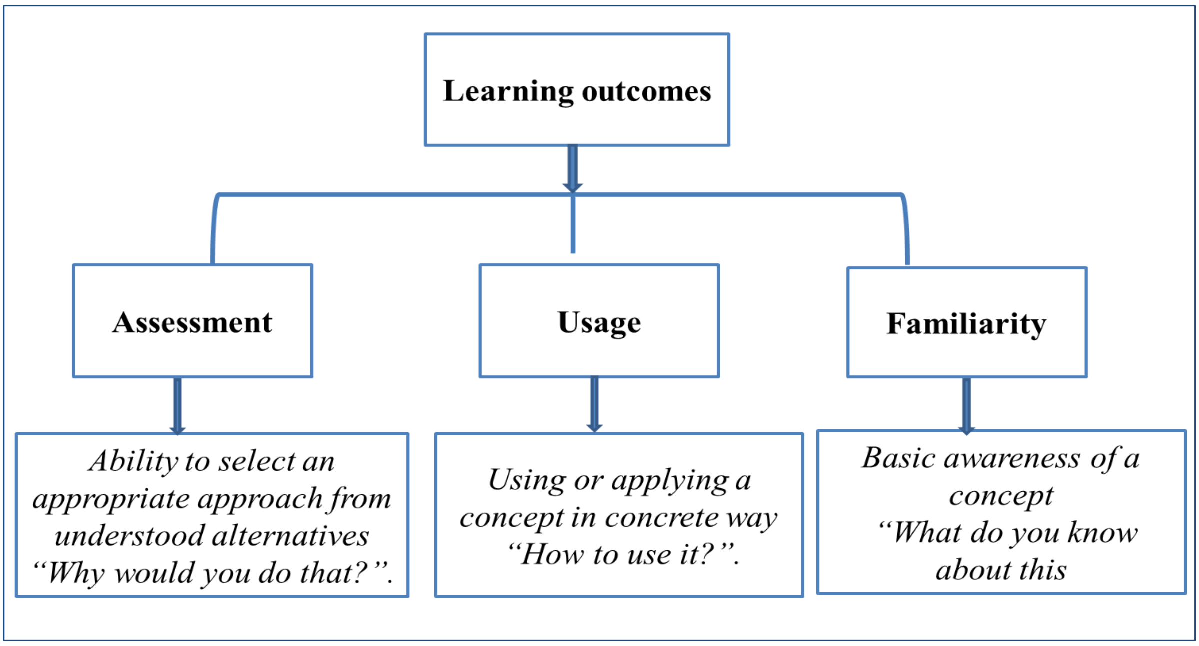

Figure 6-6 Learning outcome classification

Although the learning module was designed and returned to the expert, it was not possible to evaluate all the information provided in the module specification page as it would take very long time, therefore a controlled experiment was conducted to test three selected topics (Recursion, Variable and Loop). This also helped in allowing a consistent view on these three topics.

Therefore, two websites were used that the system returned for a specific module for a particular learner; one with a high score (very suitable) and the other with a low score (not suitable) according to the APELS ranking system. The experts were then asked to state whether or not these websites satisfy the learning outcomes and whether or not the teaching material is good for the learners.

The participating experts took their time to read the produced learning material and to evaluate the content before answering the questions. They were asked to answer the following three questions with simple “yes” and “no” answers.

Q1: Would you agree that this content satisfies the learning outcome, Familiarity?

This question is designed to elicit the experts’ opinion on the quality of the content delivered; whether or not it is associated with the Familiarity task. The experts would verify whether the content provided definitions of the topic as well as important terminology and explanation that help learners to fully comprehend the concept.

Q2: Would you agree that this content satisfies the learning outcome, Usage?

The second question is associated with the Usage task; it attempts to evaluate the content produced by APELS. The domain experts assess whether the content provides example, block of code, or flowchart that assist the learner to understand how to use or apply the concept practically.

Q3: Would you agree that this content satisfies the learning outcome, Assessment?

The third question is designed to explore the experts' opinion regarding the content quality delivered: whether or not it is related to the Assessment task or not. The experts check whether the content includes a simple introduction and some examples that clarify each concept in order to enable the learner to select the method appropriate for a specific problem.

6.6 Analysis of the Experiments Results

6.6.1 Experiment 1

Some domain experts while testing the system usability commented on the system's interface. Their comments were taken into account to improve the overall usability of the system making it easier and simpler to use for future versions. For instance, in the module specification page, one expert stated that "The navigation on the module specification page lacks enough instructions or explanations as shown in the representative Figure 6-5. As a lecturer I will go to the links because they are the only words which are underlined, but students might not know unless it is clearly directed". He recommended adding certain instructions or explanations to clarify the purpose of each link. For example, for each topic there should be a clearly labelled link for the recommended material and exercises to help the users understand the information on the page. This expert is a computer scientist and he has extensive knowledge on software engineering; this was a good feedback to receive.

Program Structure

Topic name	Recommended Link	Learning Hours	Exercise	Evaluation
Conditional	Link	2	Link	Is this useful ?
Loops	Link	2	Link	Is this useful ?

Figure 6-5 Program structure with links to each task in the module specification page

Moreover, another comment was received from the same expert while the testing of the APELS usability; he was filling out the learning style questionnaire. He stated that the learning style page was quite small and that certain learning style questions were very long; thus it was not really obvious to read the full question as it was necessary to scroll to the side to read it. Another expert suggested more improvements to the system interface. For example, these are two questions on the type of content, which were presented in the content preference page as follows:

"more definitions and explanations and fewer practical examples".

Or "fewer definitions and explanations and fewer practical examples"

The expert suggested highlighting "**more**", "**fewer**" in bold type to help learners to understand these questions.

"**More** definitions and explanations and **fewer** examples"

Or "**Fewer** definitions and explanations and **more** practical examples"

A comment from another expert on the same phase concerns the demographic information (Gender) collected by the system. While he was creating an account, he thought it could be an issue. He asked," what is the purpose of female or male section? I think it is not useful information for gender quality purpose". He explained that this was not necessary and could cause legal problems. This expert has a good experience on security management projects for Technology Appraisals; therefore, this information was taken into consideration by the board and the gender field was removed from the APELS interface.

Finally, a further feedback was raised up by one of the experts about the evaluation section of the module specification page while he was testing the adaptability of the system. After the expert was provided with the learning material based on his needs and learning style, he then tested the adaptability by responding to the following questions in the evaluation section:

1- How satisfied are you with the content?
2- How satisfied are you with completeness of the content?
3- How satisfied are you with academic quality of the content?

4- How satisfied are you with the learning experience?

The expert answered “No” to some of them; therefore, the content was changed. He was satisfied with the new content provided by the system, but he thinks it was important to highlight the new link clearly when the window was refreshed to make it clear to the users that his feedback was taken into consideration and that the content was updated accordingly. This issue was addressed in our system based on this feedback.

Overall, the experts were satisfied with the system interface apart from the weaknesses which were addressed to provide a better interface and experience for future users.

6.6.2 Experiment 2

After the experts worked through the system interface, they were asked to assess the quality of the produced content and to indicate whether it satisfied the learning outcome as defined by the ACM/IEEE curriculum. While evaluating of the quality of the produced content phase, variety of positive and negative comments were made by the experts. They were specifically about the produced content as related to Familiarity, Usage, or Assessments learning outcomes as described in section 6.2.4.

Familiarity task

Question 1: The experts are asked whether or not the content is good enough to satisfy the Familiarity aspect of the learning outcomes. The researcher presented the experts with a learning outcome: “Describe the concept of recursion and give examples of its use” along with two websites that they system referred to. The first website recorded a high score (*http://cis.stvincent.edu/html/tutorials/swd/recur/recur.html*) for Familiarity outcome stratification and the second (*http://www.learn-c.org/en/Recursion*), a low score according to the APELS ranking system. Then, the experts were asked, “Do you agree that this content satisfies the learning outcome Familiarity?” 80% of the experts responded positively to “the learning outcome satisfaction was met by the produced content” statement, whilst 20 %

responded negatively: the content did not satisfy the learning outcome. The results are summarised in Figure 6-7.

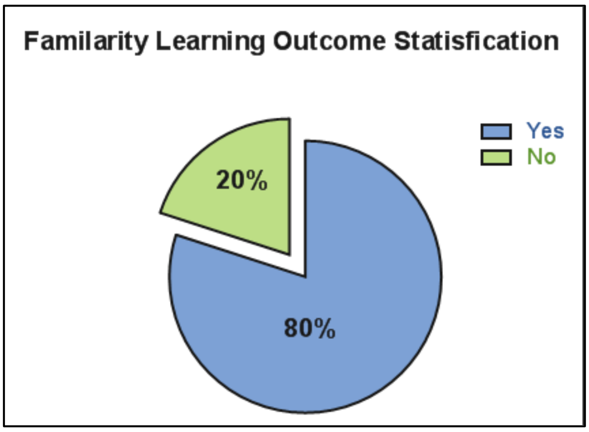

Figure 6-7 Results for question 1 about Familiarity satisfaction by the content

The majority of the experts commented positively on the quality of the content and they stressed the fact that they would choose it as a teaching material when they deliver a lecture on the basic knowledge of recursion. For example, the experts with most experience in software engineering were highly satisfied with the website, which got a high score "this strongly satisfies the Familiarity parameter because it gives clear simple definitions of what recursion is, how the definition is segmented into understandable parts and what the origin of the concept is. It also serves as a very good example of Familiarity in terms on the factorial examples". Moreover, seven other domain experts also found out that the website (the one with the high score) provided more information, more explanations and details about recursion, which reflected the good quality of this content compared to the other website (the one with a low score), which provides very limited information.

On the other hand, 20 % of experts disagreed with the outcomes with respect to satisfaction for different reasons. Although they were satisfied with both websites as teaching material resources for the Recursion concept, they preferred the website with the low score as it was more concise. For instance, one of them, who was familiar with the topic in the given example (recursion), thought that the website of the low score is better for him as fewer details and explanation were provided; “if I was a student with a background on the subject in the given example, I would prefer the second website as I did not provide a thorough explanation”. Nonetheless, the researcher thinks that the user who looks for this website is not well acquainted with the term; thus, s/he finds these details beneficial for them. Otherwise, the user can go to the next level quickly if s/he would find this level very basic and unnecessary. Another criticism was raised by the same expert about how the learning material was presented; he thinks it was very detailed and lengthy, which did not comply with his learning style: “When I need to read a book with complex details, first of all, I look for something that briefly explains the concept and/or the principle to make me aware of the topic”. The researcher thinks that what was provided about recursion was simple and to the point, but if the user prefers more concise presentation with a short text, this can be solved by the system based on his/her learning style as the content can be provided with more figures and visual aids rather than with texts only. However, if s/he prefers a summarized and/or a concise text, it would be difficult for the system to figure it out unless a summarization function was introduced to the system, which was outside the scope of the current research. Another expert thinks that choosing the amount of comprehensive information in the website depends on the requirement of the learner. If the learner just wants to know what recursion is, s/he may look for the concise information, but if s/he needs more information about recursion, s/he will choose this website (high score).

Usage task

Question 2 was raised to assess whether or not the content provided by the system satisfies the Usage parameter of the learning outcomes. The learning outcome, which is used as an example was “Write a program that uses the variable concept” together with the two websites that system made use of; the variable concept was used when making comparison. Again, one website got a high score (*http://www.cplusplus.com/doc/tutorial/Control/*)for Usage satisfaction and the other, a low score (*http://www.c4learn.com/cplusplus/cpp-variable-*

naming/) according to the APELS ranking system. Then the question to be asked was: "Would you agree that this content satisfies the Usage parameter of the learning outcome?" 90 % of the experts said, ''Yes, the content satisfied the Usage learning outcome, whilst 10 % said, "No, the content did not satisfy the Usage learning outcome as illustrated in the Figure 6-8 below.

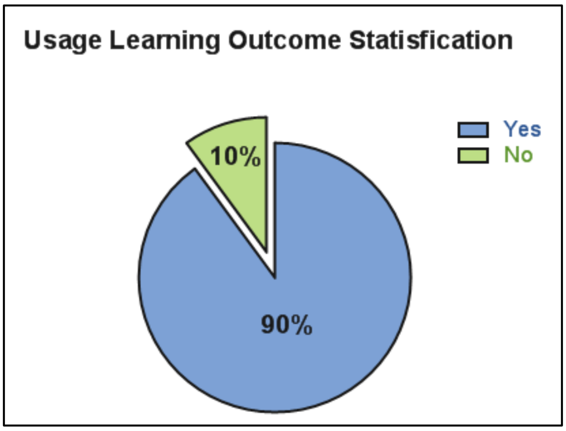

Figure 6-8 Results for question 2 about Usage satisfaction by the content

Many positive comments were made by the experts in relation to the content satisfaction of the Usage learning outcomes. The experts stated that high score website delivered valuable information that matched the learning outcomes very well because "it provided more examples on how to define and use variables in a C++ program. On the other hand, the other low score website did not clarify how to use variables in the programs". Likewise, another expert stated that "the website with a high score is an example on how to use the variables to store information to be referenced and manipulated with a code. It also displayed the different types of variables and how to use them. Therefore, it provided a better learning material for the Usage task, whereas, the website with a low score did not provide any hint on how to use variable, so this website did not satisfy the usage element of the variable topic".

Although a high percentage stated that the content satisfied the Usage learning outcomes, only one expert did not agree that the website with a high score was better to teach students how to use variable as a concept because he thinks the explanation given by the website with a low score provided enough guidance for the students on how to use variable without the need for the examples provided by the other website.

Assessment task

Question 3 asked the experts whether or not the content satisfied the Assessment learning outcomes. The learning outcome which was used as an example was “Determine which type of loop is best for a given problem “. The system consulted two websites for “Loop”, one with a high score (*http://www.cplusplus.com/doc/tutorial/Control/*) and the other with a low score (*https://www.appgamekit.com/documentation/principles/6_loopsync.htm*) using the APELS ranking system. Then the question asked to the expert was, “Do you agree that this content satisfies the learning outcome assessment?

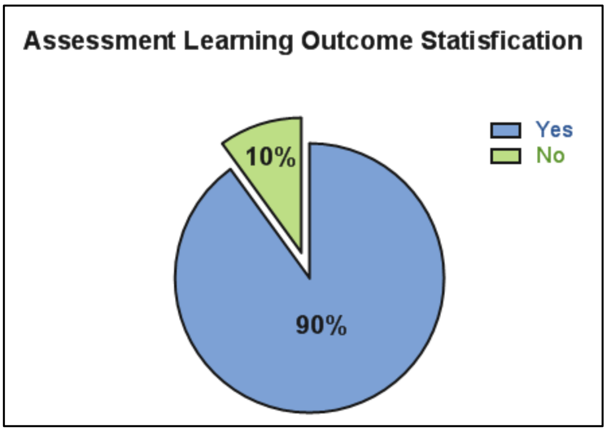

Figure 6-9 Results for question 3 about Assessment satisfaction by the content

90 % of the experts answered, ‘'Yes, the content is of a good quality when delivering a lecture that aims to meet the Assessment learning outcome, whilst 10 % answered, “No”; the results are exhibited in Figure 6-9. Many positive remarks were put forward in relation to the evaluation of the content against the assessment learning outcomes by the experts. The experts found out that the website which the system referred to with the highest score; it combines the three types of loops and provides a simple introduction and fewer examples on each type. That would form a good learning material enabling the students to compare the three types of loops and to decide why they choose one of them. The experts also agreed that the website with low

score did not include explanations neither details that help the learner to select the appropriate method for a specific problem. For instance, one expert stated that “the website with a high score clearly explained how each loop worked and demonstrated how a program could be executed; besides, it presented the results of an executable program. Also, the idea of using a flowchart is very interesting”. The evaluator thinks it is a good example of a learning material for the Assessment task while the website with a low score does not clarify the concepts or the principles of the loops.

On the other hand, only one evaluator did not think that one of the websites was superior to other since they both delivered explanations and examples, which might improve the students` understanding of the topic. Furthermore, he preferred to choose the website with the low score as it is more concise and offers sufficient explanations on the topic; thereby, it can meet the Assessment learning outcome more concretely.

6.6.3 General Discussion about the APELS System

As pointed out by Rogers et al. (2011), an unstructured interview has the advantage of highlighting issues by interviewers that have not been considered by the researcher. Therefore, at the end of the experiment, a general discussion was presented; it aimed to obtain feedback from the experts to identify weaknesses and make suggestions for tackling them. Hence, additional information would be required from the concerned experts that may have otherwise been overlooked. Their insights would contribute to enriching the system. Indeed, it was expected that the experts would have varied views on the system since they came from different backgrounds and have different views on education and they ways it should be delivered. That would untimely benefit the system overall design and its outcomes. This discussion was recorded and transcribed before the evaluation and the analysis of the results.

Various experts during the open-discussion phase made some remarks on the methodology applied for selecting the appropriate learning resources and the content from the web based on the learners' needs, learning styles, and learning outcomes. For instance, one of the experts thinks that is a remarkable work because it saves the students' time and brings them the best material from the freely available online resources. They think that the relevant information, which suits the purpose, has been extracted from a large number of resources that overload the Web; they also appreciate the personalisation feature of the system as it considers the learner's requirement and fetches the learning material they need in a cost and time-efficient manner. One of the experts pointed out that "the learners experience difficulty in finding the information that best suits their needs; they must read and waste time going through links instead of utilising APELS which recommends the required information and material based on the user's input". Moreover, a positive comment was made by one expert in relation to the system performance. He thinks that the system has the ability to perform automatically as it can receive and store information, adapt them to the new situations and learn from experience. However, some experts highlighted some weaknesses in the system. For instance, one expert stated that "the search engine, which superimposes ranking as Google does, is vulnerable to someone who knows how the system works. They might upload lots of keywords and key phrases to give them a higher score, which was the issue with the old days search engine".

Moreover, some aspects of APELS, which require improvement, were raised during the interview; for instance, one of the experts thinks it is crucial for the era of multimedia to provide interactive material with more video, audio and animation. He stated that "it is most important in the education filed these days to make the learning process more enjoyable and fun instead of seeing boring numbers and plain texts in order to enhance the learning effectiveness". Despite the fact that it was not the main focus of the evaluation, the expert came up with some interesting comments about the learning styles. The assessor suggested blending many learning styles because a learner would probably have a mix of learning styles for each topic s/he is going to study rather than learning all topics in the same learning style. For instance, in the given example where VARK obtained his visual learning style, he suggested to have a random mix of all learning styles provided in VARK for each topic instead of having the entire learning material of this topic as visual.

Another negative comment came from one of the experts who thinks that it is not adequate to give a remark of good or satisfactory for a website based on a score given by the system; human evaluation of the system is necessary. “You gave a number or assigned a score for each website; you determined that this is the best website because it has got an algorithm in your system; however, the system needs to be evaluated by a human being to make sure that it is good; a process which is time-consuming”. That is why we implemented this predictive evaluation of the learning material by experienced reviewer to assess the quality of the produced material, Additionally, this feedback was taken into account where the system can be updated and improved based on previous user’s feedback, which was tested by the subsequent evaluators. For examples: the four following questions were devised and implemented in the evaluation section of the module specification page to be filled by the future users.

1- How satisfied are you with the content?
2- How satisfied are you with completeness of the content?
3- How satisfied are you with academic quality of the content?
4- How satisfied are you with the learning experience?

Questions 1, 2 and 3 were designed to investigate the learners’ opinion about the quality content delivered whether it is relevant and clear which helps learners to fully comprehend the concepts. Whereas, question 4 is associated to the learning style of the learner, and it is used to update the learning style based on the learner’s feedback. Moreover, it is used to know the extent to which the learners are satisfied with the learning experience.

6.7 Summary

This chapter presented the experts` evaluation of the system to test the research hypothesis "APELS can produce the right learning material that suits the learning needs of a learner as teachers would do". This included performing two experiments testing two sub-hypotheses H1 and H2 as well as general discussion carried out during the unstructured interviews. The first experiment was to test the usability of the system and the performance of the learning model, which involves the experts creating account as a user and been designed with initial learning style using VARK questionnaire and provided with learning material per their input. The results of this experiment carry promising results as they found it user-friendly and they added some comments that have been addressed in the system.

The second experiment was performed to predictively evaluate of the produced learning material against a set of learning outcomes. In general, the APELS has received positive comments regarding its overall performance since it has met the main objective of providing personalised adaptive learning material to E-learners selected from the freely available resources, which successfully meet the pre-defined learning outcomes. From the questionnaires, the learning material received a positive feedback as 80%, 90% and 90% of the experts think that the produced content is of a good quality and that it successfully meets the pre-defined learning outcomes: Familiarity, Usage and Assessment respectively. That clearly reflects the success of the novel learning outcome validation approach and of the ontology tools used for information extraction from the Web. Similarly, the domain experts praised the adaptability of the system which can change the content based on the users' evaluation. In addition, APELS learns from experience; it updates based on the users' feedback.

On the other hand, certain issues and problems with the system were highlighted by the domain experts; they were related to the interface of the system, which could be easily updated and rectified. These issues included the font size of some information on the page and the inadequate labelling of the navigation. Furthermore, some experts in the process of the open-feedback phase following the close-questions pinpointed certain weaknesses, such as the search engine that is superimposed by our own ranking system based on keywords and key phrases, so it is vulnerable to misconduct by people who know how the system works. Additionally, the lack of the multimedia tools such as the video and retrospectives evaluation of the content by the students and experienced review are main drawbacks of the system, which should be addressed by future researchers.

Chapter 7 Conclusions and Future Work

7.1 Introduction

With the swift advances in the development of E-learning systems, personalisation and adaptability are considered to be the major challenges encountering the E-learning development. In the current study, a personalised and adaptive E-learning system architecture is introduced to provide a personalised and adaptable learning environment to each user from the freely available resources on the Web. In this approach, ontology was employed to model a specific learning subject and to extract the relevant learning resources from the Web based on the learner model (the learner's background, needs and learning styles). Thereafter, the extracted material was validated against the defined learning outcomes using NLP tools and techniques. The contents of the designed models were delivered using the planner component. Moreover, the APELS system provides adaptability based on the learner's feedback. Taken together, this novel system could lay the foundation for the future development of personalized, flexible and adaptable E-learning environment to the learner in a cost-efficient manner.

In this chapter, the objectives of the thesis will be revised and the means of realizing them will be illustrated. In section 7.3, the ideas and suggestions for the future development will be presented.

7.2 Review of the Research Objectives

This section introduces the research objectives and reviews the means of achieving them.

A review of the existing work on automatic knowledge extraction from the Web, E-learning platforms and E-learning styles

The main objective of this study is to develop the E-learning system that can provide personalised adaptable learning material to leaners; the system can be utilised by the educational institutes with great flexibility. First, it is essential to review the existing work in order to identify the tools that are appropriate to develop the proposed novel system. Among the available NLP tools and resources discussed in Chapters 2 and 3, an ontology was selected as the main tool to be utilised to extract the relevant learning resources from the Web. It will take in account the learner`s pedagogical needs, background and learning styles before commencing any search. Among the available learning styles, VARK was chosen due to its ease of use and its free availability in order to identify the preferred learning styles of the learner (Chapter 3), which will be considered in designing a course. Commonly used similarity measures were reviewed in order to select the appropriate one to be employed for matching the extracted material with the ontology concept as applied to a specific domain. The Dice coefficient was chosen in the present study because of its ease of use and its superiority to others in finding the best fit as a result of the intersection between the ontology domain concepts and the entities on the Web.

Designing the architecture of the Adaptable and Personalised E-learning System (APELS)

After reviewing the literature for the most appropriate tools to be used, APELS was designed as discussed in Chapter 4. APELS consists of three main models: the Learner model, Information extraction model and delivery model. Each model is represented as a separate entity in the architecture. The learner model includes information about the learner's background, pedagogical needs, learning styles and content preferences that help the system determine the appropriate teaching strategies. The information extraction model includes the relevance phase and the ranking phase. The relevance phase aims to extract the most relevant websites from the freely available resources rapidly and cost-efficiently. This is performed first by fetching a list of websites that deal with specific areas according to the learner's request and transforming them from HTML to XHTML to be more structured in order to facilitate the matching and knowledge extraction processes. Thereafter, the elements in the XHTML files are extracted using Xpath; finally, these elements were stored in a vector. Meanwhile, the ontology of the specific domain of knowledge was constructed using the protégé tool to obtain the OWL file forming the second vector. The two vectors were then matched by the dice coefficient to find out the best similarity between them in order to extract only the relevant websites.

In the ranking phase, a linguistic analysis of the extracted content was performed using the Stanford CoreNLP tool to semantically annotate the target words. A novel learning outcome validation approach is proposed in this research; it utilises the linguistic feature of NLP to extract significant key phrases and keywords related to the pre-defined learning outcome, which is sub-classified into Familiarity, Usage and Assessment as defined by Bloom's taxonomy. To perform this, eight linguistic rules and keyword based rule were developed in this study to extract key phrases and keywords, which meet the learning outcomes, based on defining the patterns of the parts of speech of the lexical items and their dependency relations. To address the adaptability aspect of the system, a third model was added to APELS, which is the delivery model. This model has a planner that structures the produced content into the module title, a summary of the programme and the intended learning outcomes. Interestingly, the planner is also able to update the content according to the learner's feedback and learning style to ensure the adaptability of the system. The learner model, knowledge extraction model and delivery model of APELS were assessed separately and then they were integrated to formulate a novel E-learning system APELS that was then implemented using a specific field of learning with a well-defined curriculum content, making use of computer science.

Implementing the proposed tools using a specific field of learning with well-defined curriculum content and integrating them into APELS to develop a computer based APELS system

The core of the thesis is to design the appropriate tools for APELS and to assess their functionality. The functionality of the generated tools for each model was first assessed individually through the performance of the necessary tasks (Chapter 4). It is the researcher`s conviction that separating the three models, in particular the knowledge extraction model, enhances the system's flexibility and extensibility and allows for the reusability of all of its components in any educational domain. After designing the APELS architecture, it is important to assess the functionality of this novel system as a whole by an experimental implementation using the computer science domain (Chapter 5). First, the knowledge domain was structured by organising the topics of the ACM/IEEE Computer Science Curriculum and establishing the semantic relations between domain topics using ontology. The produced OWL files were implemented in the matching process to extract the relevant learning resources. The results of the matching process were presented in Chapter 5, where the system consulted a list of websites for the specific module, "Algorithms and Data Structure", with the best match is 53% using the dice coefficient. The websites with the least ranking were excluded from further analysis as they were considered to be irrelevant to the learner`s request. The content of the relevant websites was then validated against the pre-defined learning outcomes. A proposed novel learning outcome validation approach was also applied; it utilised the linguistic features of NLP to extract the significant key phrases and keywords related to the pre-defined learning outcome, which is sub-classified into Familiarity, Usage and Assessment. For example, the highest Familiarity score in the "Fundamental Programming Concepts" module was obtained by the *(www.cplusplus.com/doc/ tutorial/variables*) website, which reflects the higher frequency of the familiarity-related key phrases and keywords in the content; therefore, this website is considered to be the best for a learner who endeavours to gain a basic understanding of a topic containing lots of definitions and illustrations of the fundamental programming concept. To ensure that the system could deliver the right content that satisfy the pre-defined learning outcomes, the system was further evaluated by experts.

Evaluating the proposed system using experts from the field of education

The system evaluation process was described in Chapter 6; it was centred on performing predictive evaluation of the learning material, which was produced by the system to meet the pre-defined learning outcomes drawn by experienced reviewers. The evaluation also included assessing the system usability and the unstructured interviews. Ten experts (university academic staff from various disciplines, i.e. computing, mathematics and education) were invited to perform this task. Overall, the feedback with regard to matching the content to the learning outcomes was positive. 80% of the experts agree that the provided material was of good quality and that it could be used for preparing and delivering a lecture in order to familiarise the students with a given topic, and even more promising, 90% of them think that the content provided by the system in the experiment was so high in quality that it could be used as teaching material to achieve the Usage task. They agree that the content was informative and comprehensive and that it clearly reflects the success of the novel learning outcome validation approach and the NLP tool used to perform this function as well the ontology tools used for information extraction from the Web. These results were promising because extracting suitable learning materials is situated at the heart of APELS. Moreover, one of the key goals of this system was achieved; it was using freely available resources that added the cost-efficiency advantage to the system. Although the main goal of the developed system was achieved successfully, it was also important to assess how convenient and satisfied the user would be while navigating throughout the system. For this aspect, most of the experts except one were satisfied with the system interface; however, few comments were raised by the expert such as the lack of proper instruction while navigating the page, and the small font size of some information items on the page, which should be addressed in the future. The experts also praised the fact that the system was flexible and personalised meeting the student requirements and queries; besides, it could update itself based on the learner feedback, which ensures its adaptability.

7.3 Future Work

There are several directions towards which this research can be further extended and improved. The following areas are potentially worthwhile pursuing in the future:

1. The performed evaluation of the system in this thesis was very useful; however, if time allowed, it could be followed up with a learner evaluation. A learner evaluation can be undertaken by enabling a set of students in a university to test the efficiency of the system and to investigate its usefulness themselves. The idea here is to ask these learners to use the system to create an account in order to get their comments and opinions regarding the capability of the system to provide content that satisfies their learner's requirements. In this type of evaluation, survey questionnaire templates could be utilised and distributed to be filled by these learners, who have already used the system for studying a course, in order to realize the extent to which learners are satisfied with the content produced and to judge the effectiveness of this system.
2. Expanding the system to include other media learning resources such as video and audio files as the current version of the system considers only the text based resources. Video and Audio files should be added in the future to the system as probably some learners might prefer these learning styles which were not included in this system. In order to implement these learning styles, validation of the title of the video or audio is required, besides considering their description. By validating the title and description together, one can get a better chance for retrieving the relevant video from the Web for the user. For example, when uploading videos on YouTube, there are certain requirements by the search engine to have the video listed. These include a title stating what the video is about together

with a clear description of the content of that video. These two learning styles will be most useful additions to the system in the future.

3. Extracting structured data from the websites is not a trivial task. Much of the content available on the Web is formatted in HTML form, which is transformed into XHTML to provide the information in a friendly accessible format that is easier for extraction and comparison. However, the content of few websites are not extracted by our current process because the content of these websites is published in various formats, such as PDF, PPT or word file. In the future, a developed approach for converting these formats to XHTML format is indispensable in order to parse and evaluate the information.
4. One of the significant contributions of the APELS system is to extract relevant concepts from the Web by using the ontology domain. During the development of the system, it was noted that some concepts were not extracted because the synonyms of these concepts were not identified. For example, the synonyms for the concept "IF Statement" includes decision making, conditional, selection statement etc. Therefore, in the future developments of the system, one needs to work with large data in order to define more synonyms for the concepts.
5. The current research has not investigated the possibility of adapting the system to other domains since that may cause problem in practise although the system is designed to be easily adaptable,
6. The system is implemented on a PC and the users these days have many other devices such as tablets and mobile phones; currently, the functionality of the system on other platforms has not been assessed.

In conclusion, this research has proposed a framework for an adaptable and personalised E-learning system (APELS) architecture that is based on the use of ontology and NLP tools to provide a personalised and adaptable learning environment to each user from the freely available resources on the Web. The APELS system provides adaptability based on the learner’s feedback and assessment once the learning process is initiated by the learner. The author hopes that the APELS system is expected to develop over time with more users, which would add more suggestions and solutions if any problem encountered by the users. The author also expects APELS system to be used as a learning tool for other domains in the future.

Reference

Abuleil, S. 2004. Extracting Names From Arabic Text for Question-Answering Systems. *RIAO'04, Proceeding of the 7th International Conference on Coupling Approaches, Coupling Media, and Coupling Languages For Information Retrieval*, University of Avignon (Vaucluse), France ,26-28 April 2004, 638-647.

Ahmed, S. T., Nair, R., Patel, C. & Davulcu, H. 2009. BioEve: bio-molecular event extraction from text using semantic classification and dependency parsing. *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task.* Association for Computational Linguistics.

Anderberg, M. R. 1973. Cluster analysis for applications. Monographs and textbooks on probability and mathematical statistics. Academic Press, Inc., New York.

Anderson, L. W., Krathwohl, D. R. & Bloom, B. S. 2001. A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. *Allyn & Bacon. Boston, MA (Pearson Education Group).*

Appelt, D. E. 1999. Introduction to information extraction technology. *Artificial Intelligence Communications. Tutorial. In Processdings of the International Join Conference on Artificial Intelligence Communications*, August 2, 1999, Stocklhom, Sweden.12 (3), 161-172.

Anuar, N. and Sultan, A.B.M., 2010. Validate conference paper using dice coefficient. *Computer and Information Science*, *3*(3), p.139.

Atchison, W. F., Conte, S. D., Hamblen, J. W., Hull, T. E., Keenan, T. A., Kehl, W. B., Mccluskey, E. J., Navarro, S. O., Rheinboldt, W. C. & Schweppe, E. J. 1968. Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science. *Communications of the ACM,* 11**,** 151-197.

Ausubel, D., Novak, J. & Hanesian, H. 1978. Educational Psychology: a Cognitive View; Holt, Rinehart and Winston: London, 1978. 6. *Entwistle, N.*

Ausubel, D. P. 1960. The use of advance organizers in the learning and retention of meaningful verbal material. *Journal of educational psychology,* 51**,** 267.

Baeza-Yates, R. & Ribeiro-Neto, B. 1999. Modern Information Retrieval Addison-Wesley Longman. *Reading MA.*

Bajraktarevic, N., Hall, W. & Fullick, P. 2003. Incorporating learning styles in hypermedia environment: Empirical evaluation. *In P. de Bra, H. C. Davis, J. Kay & M. Schraefel (Eds.), Proceedings of the Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems. Nottingham, UK, Eindhoven University***,** 41-52.

Baldridge, J. 2005. The opennlp project. *URL: http://opennlp. apache. org/index. html,(accessed 2 February 2012).*

Benajiba, Y., Diab, M. & Rosso, P. 2009. Arabic named entity recognition: A feature-driven study. *IEEE Transactions on Audio, Speech, and Language Processing,* 17 (5)**,** 926-934.

Berend, G. & Farkas, R. 2010. SZTERGAK: Feature engineering for keyphrase extraction. *Proceedings of the 5th international workshop on semantic evaluation.Association for Computational Linguistics.SemEval 2010, Uppsala, Sweden, 15–16 July 2010. Stroudsburg, PA: ACL***,** 186-189.

Berners-Lee, T., Hendler, J. & Lassila, O. 2001. The semantic web. *Scientific american. May 2001,* 284**,** 28-37.

Bittencourt, I. I., Isotani, S., Costa, E. & Mizoguchi, R. 2008. Research directions on Semantic Web and education. *Interdisciplinary Studies in Computer Science,* 19**,** 60-67.

Bleimann, U. 2004. Atlantis University: a new pedagogical approach beyond e-learning. *Campus-Wide information systems,* 21**,** 191-195.

Bloom, B. S. 1956. Taxonomy of educational objectives. Vol. 1: Cognitive domain. *New York: McKay*, 20-24.

Bontcheva, K., Tablan, V., Maynard, D. & Cunningham, H. 2004. Evolving GATE to meet new challenges in language engineering. *Natural Language Engineering,* 10, Cambridge University Press, 349-373.

Boyce, S. & Pahl, C. 2007. Developing domain ontologies for course content. *Educational Technology & Society,* 10, 275-288.

Bracewell, D. B., Ren, F. & Kuriowa, S. 2005. Multilingual single document keyword extraction for information retrieval. *IEEE International Conference on Natural Language Processing and Knowledge Engineering. Wuhan, China*, 517-522.

Bray, T., Paoli, J., Sperberg-Mcqueen, C. M., Maler, E. & Yergeau, F. 2008. Extensible markup language (XML) 1.0. W3C recommendation.

Brill, E. 1992. A simple rule-based part of speech tagger. *In Proceedings of the Third Conference on Applied Computational Linguistics. Trento, Italy: Association for Computational Linguistics*, 112-116.

Brusilovskiy, P. 1994. The construction and application of student models in intelligent tutoring systems. *Journal of computer and systems sciences international,* 32 (1), 70-89.

Brusilovsky, P. 1998. Adaptive educational systems on the world-wide-web: A review of available technologies. *Proceedings of Workshop" WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX*.

Brusilovsky, P. 2001. Adaptive hypermedia, User Modeling and User Adapted Interaction. *Kluwer Academic Publishers. Printed in the Netherlands*, 87-110.

Brusilovsky, P. 2004. KnowledgeTree: A distributed architecture for adaptive e-learning. *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters. May 19 - 21 2004* New York, NY, USA, 104-113.

Brusilovsky, P. 2007. Adaptive navigation support. *The adaptive web.* Springer Berlin Heidelberg. 263-290.

Brusilovsky, P. & Maybury, M. T. 2002. From adaptive hypermedia to the adaptive web. *Communications of the ACM Communications of the ACM - The Adaptive Web CACM Homepage archive. 5, May 2002* 45**,** New York, NY, USA,30-33.

Cai, Z. Graesser A.C., Hu, X. 2015. ASAT: AutoTutor script authoring tool. In R. Sottilare, A.C. Graesser, X. Hu., & K. Brawner(Eds.), Design Recommendations for Intelligent Tutoring Systems: Authoring Tools & Expert Modeling Techniques (Vol.3)(pp. 199-210). Orlando, FL: Army Research Laboratory.

Cantoni, V., Cellario, M. & Porta, M. 2004. Perspectives and challenges in e-learning: towards natural interaction paradigms. *Journal of Visual Languages & Computing,* 15**,** 333-345.

Carro, R. M., Pulido, E. & Rodriguez, P. 2001. TANGOW: a model for internet-based learning. *International Journal of Continuing Engineering Education and Life Long Learning,* 11**,** 25-34.

Cassel, L., Clements, A., Davies, G., Guzdial, M. & Mccauley, R. M. 2008. Computer science curriculum 2008. *Association for Computing Machinery (ACM)/IEEE Computer Society. Retrieved December 2009, from https://www.acm.org/education/curricula/ComputerScience2008.pdf.*

Cassin, P., Eliot, C., Lesser, V., Rawlins, K. & Woolf, B. 2004. Ontology extraction for educational knowledge bases. *Agent-Mediated Knowledge Management. Springer Berlin Heidelberg***,** 297-309.

Chen, I. 2009. Behaviorism and developments in instructional design and technology. *Rogers, P., Berg, G., Boettcher, J., Howard, C., Justice, L. Shenck, K.(Eds). Encyclopedia of Distance Learning. United States of America: Idea Group Incorporated***,** Second Edition.153-172.

Chi, Y.-L. 2009. Ontology-based curriculum content sequencing system with semantic rules. *Expert Systems with Applications. May 2009.* , 36 (4)**,** 7838-7847,ISSN 0957-4174.

Clark, J. & Derose, S. 1999. XML path language (XPath) version 1.0, w3c recommendation. In http://www.w3.org/TR/xpath.html.

Cohen, J. D. 1995. Highlights: language-and domain-independent automatic indexing terms for abstracting. *Journal of the American society for information science,* 46**,** 162.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. & Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research,* 12**,** 2493-2537.

Conlan, O. 2005. *The multi-model, metadata driven approach to personalised eLearning services.* PhD, Department of Computer Science, Trinity College, Dublin.

Conlan, O., Dagger, D. & Wade, V. 2002. Towards a standards-based approach to e-Learning personalization using reusable learning objects. *Proc. of World Conference on E-Learning, E-Learn***,** 15-19.

Craik, F. I. & Lockhart, R. S. 1972. Levels of processing: A framework for memory research. *Journal of verbal learning and verbal behavior,* 11**,** 671-684.

Croft, W. B. 1993. Knowledge-based and statistical approaches to text retrieval. *IEEE Expert: Intelligent Systems and Their Applications,* 8**,** 8-12.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Ursu, C., Dimitrov, M., Dowman, M., Aswani, N. & Roberts, I. 2002. Developing language processing components with

GATE (a user guide). *Techical Report, University of Sheffield, UK, available in https://gate.ac.uk/*.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Ursu, C., Dimitrov, M., Dowman, M., Aswani, N., Roberts, I. & Li, Y. 2009. *Developing Language Processing Components with GATE Version 5:(a User Guide)*, Citeseer.

Curricula, C. 2001. Final Report, December 15, 2001, The Joint Task Force on Computing Curricula IEEE Computer Society Association for Computing Machinery (ACM). https://www.acm.org/education/curric_vols/cc2001.pdf.

De Marneffe, M.-C. & Manning, C. D. 2008. Stanford typed dependencies manual. Technical report, Stanford University.

Dice, L.R., 1945. Measures of the amount of ecologic association between species. Ecology, 26(3), pp.297-302. doi:10.2307/1932409

Dougiamas, M. 2002. Modular object-oriented dynamic learning environment (Moodle). http:// www.moodle.org.

Duarte, J. M., Santos, J. B. D. & Melo, L. C. 1999. Comparison of similarity coefficients based on RAPD markers in the common bean. *Genetics and Molecular Biology,* 22**,** 427-432.

Dunn, R. S. & Dunn, K. J. 1978. *Teaching students through their individual learning styles: A practical approach*, Prentice Hall.

Dwyer, K. K. 1998. Communication apprehension and learning style preference: Correlations and implications for teaching. *Communication Education,* 47**,** 137-150.

Ek, T., Kirkegaard, C., Jonsson, H. & Nugues, P. 2011. Named entity recognition for short text messages. *Procedia-Social and Behavioral Sciences. 1 Jan 2011,* 27**,** 178-187.

Eklund, J. & Brusilovsky, P. 1999. Interbook: an adaptive tutoring system. *UniServe Science News. March 1999*, 12**,** 8-13.

Ellis, R. 1997. The empirical evaluation of language teaching materials. *ELT journal,* 51**,** 36-42.

Ercan, G. & Cicekli, I. 2007. Using lexical chains for keyword extraction. *Information Processing & Management,* 43**,** 1705-1714.

Erk, K. & Padó, S. 2008. A structured vector space model for word meaning in context. *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics***,** 897-906.

Farmer, J. & Dolphin, I. 2005. Sakai: eLearning and more. *EUNIS 2005-Leadership and Strategy in a Cyber-Infrastructure World. https://www.sakaiproject.org/*.

Feilmayr, C., Parzer, S. & Pröll, B. 2009. Ontology-based information extraction from tourism websites. *Information technology & tourism,* 11**,** 183-196.

Felder, R. & Soloman, B. 1997. Index of Learning Styles Questionnaire. Retrieved 6 February, 2006.

Felder, R. M. & Silverman, L. K. 1988. Learning and teaching styles in engineering education. *Engineering education,* 78**,** 674-681.

Felder, R. M. & Soloman, B. A. 1991. Index of learning styles.

Fellbaum, C. 1998. A semantic network of English verbs. *WordNet: An electronic lexical database,* 3**,** 153-178.

Fensel, D., Horrocks, I., Van Harmelen, F., Decker, S., Erdmann, M. & Klein, M. 2000. OIL in a nutshell. *International Conference on Knowledge Engineering and Knowledge Management. Springer Berlin Heidelberg.***,** 1-16.

Fink, J., Kobsa, A. & Nill, A. 1996. User-oriented adaptivity and adaptability in the AVANTI project. *Designing for the Web: empirical studies. Citeseer*.

Fleming, N. D. 2001. *Teaching and learning styles: VARK strategies,* IGI Global. Christchurch, New Zealand: Author.

Fleming, N. D. 2016. A Guide to Learning Styles. The VARK Questionnaire. *Avaliable at : http://www.vark-learn.com/english/page.asp?p=questionnaire on 4 June 2016.*

Fukuda, K.-I., Tsunoda, T., Tamura, A. & Takagi, T. 1998. Toward information extraction: identifying protein names from biological papers. *Pac symp biocomput,* 707**,** 707-718.

Gasevic, Dragan, Amal Zouaq, Carlo Torniai, Jelena Jovanović, and Marek Hatala. 2011. "An approach to folksonomy-based ontology maintenance for learning environments." *IEEE Transactions on Learning Technologies* 4, no. 4 (2011): 301-314.

Gelfand, B., Wulfekuler, M. & Punch, W. 1998. Automated concept extraction from plain text. *AAAI 1998 Workshop on Text Categorization***,** 13-17.

Gilbert, J. E. & Han, C. Y. 1999. Adapting instruction in search of 'a significant difference'. *Journal of Network and Computer applications,* 22**,** 149-160.

Github 2016. Google APIs Client Library for PHP. *Avaliable at : https://github.com/google/google-api-php-client*.

Glasersfeld, E. V. 1995. A constructivist approach to teaching. *Constructivism in education.Erlbaum, Hillsdale. Available at http://www.vonglasersfeld.com/172***,** 3-15.

Goker, A. & Davies, J. 2009. *Information retrieval: Searching in the 21st century*, John Wiley & Sons. Chichester, U.K.

Gomaa, W. H. & Fahmy, A. A. 2013. A survey of text similarity approaches. *International Journal of Computer Applications,* 68.

Gosling, J., Joy, B., Steele, G. & Bracha, G. 2005. Java (TM) Language Specification (The 3rd Edition). Addison-Wesley Professional.

Grabowski, B. & Jonassen, D. 1993. Handbook of individual differences. *Learning and Instruction, Lawrence Erlbaum Associates Publishers*.

Graesser, A.C., 2016. Conversations with AutoTutor help students learn. International Journal of Artificial Intelligence in Education, March 2016 ,26(1), pp.124-132.

Graham, G. & Bechtel, W. 1998. A companion to cognitive science. ISBN: 978-1-55786-542-7. August 1998, Wiley-Blackwell. 1-810.

Gruber, T. R. 1995. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies,* 43**,** 907-928.

Gütl, C., Garcia-Barrios, V. M. & Mödritscher, F. 2004. Adaptation in e-learning environments through the service-based framework and its application for AdeLE. *Proceedings of the E-Learn***,** 1891-1898.

Hahn, U., Romacker, M. & Schulz, S. 2002. MEDSYNDIKATE—a natural language system for the extraction of medical information from findings reports. *International journal of medical informatics,* 67**,** 63-74.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter,* 11**,** 10-18.

Harman, D. 1991. How effective is suffixing? *Journal of the American Society for Information Science,* 42**,** 7-15.

Henderson, M., Shurville, S., Fernstrom, K. & Zajac, M. 2009. Using learning styles to personalize online learning. *Campus-wide information systems,* 26**,** 256-265.

Henze, N., Nejdl, W. & Wolpers, M. 1999. Modeling constructivist teaching functionality and structure in the KBS hyperbook system. *Proceedings of the 1999 conference on Computer support for collaborative learning. International Society of the Learning Sciences***,** 28.

Hepple, M. 2000. Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers. *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics***,** 278-277.

Herrera, J. P. & Pury, P. A. 2008. Statistical keyword detection in literary corpora. *The European Physical Journal B,* 63**,** 135-146.

Honey, P. & Mumford, A. 1992. The manual of learning styles.

Hong, B. & Zhen, D. 2012. An extended keyword extraction method. *Physics Procedia,* 24**,** 1120-1127.

Horrocks, I. 2002. DAML+OIL: A Description Logic for the Semantic Web. *IEEE Data Eng. Bull.,* 25**,** 4-9.

Hu, X. & Wu, B. 2006. Automatic keyword extraction using linguistic features. Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06), 2006. IEEE, 19-23.

Hull, D. A. 1996. Stemming algorithms: A case study for detailed evaluation. *JASIS,* 47**,** 70-84.

Hulth, A. 2003. Improved automatic keyword extraction given more linguistic knowledge. *Proceedings of the 2003 conference on Empirical methods in natural language processing***,** 216-223.

Hutchins, J. 2007. Machine translation: A concise history. *Computer aided translation: Theory and practice*.

Illeris, K. 2009. *Contemporary theories of learning: learning theorists... in their own words,* Routledge.

Jaccard, P. 1901. *Distribution de la Flore Alpine: dans le Bassin des dranses et dans quelques régions voisines*, Rouge.

Jackson, P. & Moulinier, I. 2007. *Natural language processing for online applications: Text retrieval, extraction and categorization*, John Benjamins Publishing.

Jiang, H., Wang, X. & Tian, J. 2010. Second-order HMM for event extraction from short message. *International Conference on Application of Natural Language to Information Systems***,** 149-156.

Jiawei, H. & Kamber, M. 2001. Data mining: concepts and techniques. *San Francisco, CA, itd: Morgan Kaufmann*, 5.

Joshi, P., Chaudhary, S. & Kumar, V. 2012. Information extraction from social network for agro-produce marketing. *Communication Systems and Network Technologies (CSNT), 2012 International Conference on***,** 941-944.

Jurafsky, D. & Martin, J. H. 2009. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. MIT Press.

Kang, N., Van Mulligen, E. M. & Kors, J. A. 2011. Comparing and combining chunkers of biomedical text. *Journal of biomedical informatics,* 44**,** 354-360.

Keefe, J. W. 1979. Learning style: An overview. *Student learning styles: Diagnosing and prescribing programs,* 1**,** 1-17.

Kennedy, D. 2006. *Writing and using learning outcomes: a practical guide*, University College Cork.

Kern, R., Muhr, M. & Granitzer, M. Kcdc. 2010. Word sense induction by using grammatical dependencies and sentence phrase structure. Proceedings of the 5th international workshop on semantic evaluation, 2010. Association for Computational Linguistics, 351-354.

Kilgarriff, A. & Yallop, C. 2000. What's in a Thesaurus? *IN PROCEEDINGS OF THE SECOND CONFERENCE ON LANGUAGE RESOURCE AN EVALUATION.LREC***,** 1371-1379.

Kirsh, D. 2002. E-learning, metacognition and visual design. *International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medicine on the internet, L'Aquila, Italy,* 29.

Kolb, D. 1984. Experiential learning as the science of learning and development. *Englewood Cliffs NPH, editor1984*.

Krovetz, R. 1993. Viewing morphology as an inference process. *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval***,** 191-202.

Krulwich, B. & Burkey, C. 1996. Learning user information interests through extraction of semantically significant phrases. *Proceedings of the AAAI spring symposium on machine learning in information access,* 25**,** 110.

Larsen, B. & Aone, C. 1999. Fast and effective text mining using linear-time document clustering. *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining***,** 16-22.

Latham, A., Crockett, K. and McLean, D. 2014. An adaptation algorithm for an intelligent natural language tutoring system. Computers & Education, February 2014, 71, 97-110. ISSN 0360-1315.

Lee, J. H., Kim, M. H. & Lee, Y. J. 1993. Information retrieval based on conceptual distance in IS-A hierarchies. *Journal of documentation,* 49**,** 188-207.

Li, X. & Chang, S.-K. 2005. A personalized e-learning system based on user profile constructed using information fusion. *DMS,* 2005**,** 109-114.

Liddy, E. D. 2001. Natural language processing. *In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.*

Lin, D. 1998. Automatic retrieval and clustering of similar words. *Proceedings of the 17th international conference on Computational linguistics-Volume 2***,** 768-774.

Locke, W. N. & Booth, A. D. 1955. *Machine translation of languages: fourteen essays*, Published jointly by Technology Press of the Massachusetts Institute of Technology and Wiley, New York.

Loper, E. & Bird, S. 2002. NLTK: The natural language toolkit. *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1***,** 63-70.

Madnani, N. 2007. Getting started on natural language processing with Python. *Crossroads,* 13**,** 5-5.

Madnani, N. & Dorr, B. J. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36**,** 341-387.

Makris, C., Panagis, Y., Sakkopoulos, E. & Tsakalidis, A. 2009. IT Application Development with Web Services. *In M. Khosrow-Pour (Ed.), Encyclopedia of Information Science and Technology, Second Edition (2nd ed.)***,** 2278-2284. Hershey, PA: IGI Global. doi:10.4018/978-1-60566-026-4.ch361.

Manning, C. D., Raghavan, P. & Schütze, H. 2008. Flat clustering. *Introduction to information retrieval.Manning***,** 1(1), 350-374.Cambridge: Cambridge university press.

Manning, C. D. & Schütze, H. 1999. *Foundations of statistical natural language processing,* MIT Press.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S. & Mcclosky, D. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. *ACL (System Demonstrations)***,** 55-60.

Mayor, M. 2009. Longman Dictionary of Contemporary English. *Pearson Longman. 1408215322, 9781408215326. 1-2081*.

Mckenzie, A., Matthews, M., Goodman, N. & Bayoumi, A. 2010. Information extraction from helicopter maintenance records as a springboard for the future of maintenance text analysis. *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems***,** 590-600.

Mclean, J. & Looker, P. 2006. University of New South Wales Learning and Teaching Unit. Available at :*www.ltu.unsw.edu.au/content/course_prog_support/outcomes.cfm?ss=0.*

Melis, E., Andres, E., Budenbender, J., Frischauf, A., Goduadze, G., Libbrecht, P., Pollet, M. & Ullrich, C. 2001. ActiveMath: A generic and adaptive web-based learning environment. *International Journal of Artificial Intelligence in Education (IJAIED),* 12**,** 385-407.

Meziane, F. & Kasiran, M. K. 2003. Extracting Unstructured Information From the WWW to Support Merchant Existence in E-Commerce. *in Antje Dusterhoft and Bernhard Thalheim (Eds), Lecture Notes in Informatics, Natural Language Processing and Information Systems***,** 175-185, GI-Edition, Bonn 2003, Germany. ISBN3-88579-358-X.

Mirko Otto. 2008. Ontologies for semantic search in a collection of documents. Master's thesis, Dept. of Computer Science, Otto von Guericke University Magdeburg

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D. & Miller, K. J. 1990. Introduction to WordNet: An on-line lexical database. *International journal of lexicography,* 3**,** 235-244.

Modritscher, F. 2004. Victor Manuel Garcia-Barrios, and Christian Gutl. *The Past, the Present and the future of adaptive E-Learning. In Proceedings of the International Conference Interactive Computer Aided Learning(ICL2004).Villach, Austria.*

Moens, M.-F. 2006. *Information extraction: algorithms and prospects in a retrieval context*, Springer Netherlands. Springer Science & Business Media. Edition Number 1, 1-246. ISBN 978-1-4020-4993-4.

Mooney, R. 1999. Relational learning of pattern-match rules for information extraction. *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the eleventh Innovation applications of artificial intelligence conference innovative applications of artificial intelligence***,** July 1999, 328-334.

Müller, H.-M., Kenny, E. E. & Sternberg, P. W. 2004. Textpresso: an ontology-based information retrieval and extraction system for biological literature. *PLoS Biol,* 2**,** e309.

Mustafa, Y. E. A. & Sharif, S. M. 2011. An approach to adaptive e-learning hypermedia system based on learning styles (AEHS-LS): Implementation and evaluation. *International Journal of Library and Information Science,* 3**,** 15-28.

Necib, C. B. & Freytag, J.-C. 2003. Ontology based query processing in database management systems. *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"***,** 839-857.Springer Berlin Heidelberg.

Noy, N. F., Crubézy, M., Fergerson, R. W., Knublauch, H., Tu, S. W., Vendetti, J. & Musen, M. A. 2003. Protege-2000: an open-source ontology-development and knowledge-acquisition environment. *AMIA Annu Symp Proc,* 953**,** 953.

Noy, N. F. & Musen, M. A. 2000. Algorithm and tool for automated ontology merging and alignment. *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00). Available as SMI technical report SMI-2000-0831*.

Oblinger, D. G. & Hawkins, B. L. 2005. IT Myths The Myth about E-Learning. *Educause review,* 40**,** 14-15. August 2005.

Oppermann, R. & Rasher, R. 1997. Adaptability and adaptivity in learning systems. *Knowledge transfer,* 2**,** 173-179.

Ouksel, A. M. & Sheth, A. 1999. Semantic interoperability in global information systems. *ACM Sigmod Record,* 28**,** 5-12.

Pal, S., Pakray, P., Das, D. & Bandyopadhyay, S. 2010. JU: a supervised approach to identify semantic relations from paired nominals. *Proceedings of the 5th International Workshop on Semantic Evaluation***,** 206-209.

Papanikolaou, K. A., Grigoriadou, M., Kornilakis, H. & Magoulas, G. D. 2003. Personalizing the Interaction in a Web-based Educational Hypermedia System: the case of INSPIRE. *User modeling and user-adapted interaction,* 13**,** 213-267.

Pemberton, S., 2000. XHTML™ 1.0 the extensible hypertext markup language. W3C Recommendations, pp.1-11.

Peter, S., Dastbaz, M. & Bacon, E. 2008. Personalised e-learning, semantic web and learning ontologies. *Proceedings of ED-MEDIA.*

Piaget, J. 1985. *The equilibration of cognitive structures: The central problem of intellectual development*, University of Chicago Press.

Poesio, M. & Kabadjov, M. A. 2004. A General-Purpose, Off-the-shelf Anaphora Resolution Module: Implementation and Preliminary Evaluation. In Proceedings of the 4th International Conference on Language Resources and Evaluation. *LREC***,** 663–668.

Polifroni, J., Kiss, I. & Adler, M. 2010. Bootstrapping Named Entity Extraction for the Creation of Mobile Services. *LREC*.

Poria, S., Agarwal, B., Gelbukh, A., Hussain, A. & Howard, N. 2014. Dependency-based semantic parsing for concept-level text analysis. *In International Conference on Intelligent Text Processing and Computational Linguistics.Springer Berlin Heidelberg***,** 113-127.

Porter, M. F. 1980. An algorithm for suffix stripping. *Program,* 14**,** 130-137.

Qiu, L., Kan, M.-Y. & Chua, T.-S. 2004. A public reference implementation of the rap anaphora resolution algorithm. In Proceedings of the Fourth International Conference on Language Resources and Evaluation. *arXiv preprint cs/0406031***,** 291–294.

Rawashdeh, A. & Ralescu, A. L. 2015. Similarity Measure for Social Networks-A Brief Survey. *MAICS***,** 153-159.

Reeves, T. C. 1998. Answering Critics of Media and Technology in Education. *ED-MEDIA/ED-TELECOM 98 World Conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecommunications. Proceedings. AACE*.

Reynolds, D. & Muijs, D. 2005. Effective Teaching–Evidence & Practice. SAGE Publications.

Riloff, E. 1996. Automatically generating extraction patterns from untagged text. *Proceedings of the national conference on artificial intelligence***,** 1044-1049.

Rogers, Y., Sharp, H. & Preece, J. 2011. Interaction design: beyond human-computer interaction.

Roget, P. M. 1911. *Roget's Thesaurus of English Words and Phrases,* TY Crowell Company.

Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M. & Edwards, D. D. 2003. *Artificial intelligence: a modern approach*, Prentice hall Upper Saddle River.

Sadler-Smith, E. 1996. Learning styles: a holistic approach. *Journal of European Industrial Training,* 20**,** 29-36.

Sætre, R. 2006. *GeneTUC: Natural Language Understanding in Medical Text.* Doctoral thesis for the degree of doktor ingeniør Trondheim, Norwegian University of Science and Technology (NTNU). Trondheim, Norway

Saggion, H., Funk, A., Maynard, D. & Bontcheva, K. 2007. Ontology-based information extraction for business intelligence. *The Semantic Web.Springer Berlin Heidelberg.***,** 843-856.

Sahami, M., Roach, S., Cuadros-Vargas, E. & Leblanc, R. 2013. Acm/ieee-cs computer science curriculum 2013: Reviewing the ironman report. *Proceeding of the 44th ACM technical symposium on Computer science education.* ACM.

Sahami, M., Roach, S., Cuadros-Vargas, E. & Reed, D. 2013. Computer science curriculum 2013: reviewing the strawman report from the ACM/IEEE-CS task force. Proceedings of the 43rd ACM technical symposium on Computer Science Education, 2012. ACM, 3-4.

Salton, G., Wong, A. & Yang, C.-S. 1975. A vector space model for automatic indexing. *Communications of the ACM,* 18**,** 613-620.

Sampson, D. & Karagiannidis, C. 2002. Accommodating learning styles in adaptation logics for personalised learning systems. *Proceedings of the 14th World Conference on Educational Multimedia, Hypermedia and Telecommunications***,** 24-29.

Sampson, D. & Karagiannidis, C. 2010. Personalised learning: Educational, technological and standardisation perspective. *Interactive educational multimedia***,** 24-39.

Santally, M. I. & Senteni, A. 2005. A learning object approach to personalized web-based instruction. *European Journal of Open, Distance and E-Learning. from http://www.eurodl.org/materials/contrib/2005/Santally.htm*.

Saruladha, K. 2012. *Semantic similarity measures for information retrieval systems using ontology.* School of Engineering, Depat. of Computer Science.

Seng, J.-L. & Kong, I. 2009. A schema and ontology-aided intelligent information integration. *Expert Systems with Applications,* 36**,** 10538-10550.

Shackelford, R., Mcgettrick, A., Sloan, R., Topi, H., Davies, G., Kamali, R., Cross, J., Impagliazzo, J., Leblanc, R. & Lunt, B. 2006. Computing curricula 2005: The overview report. *ACM SIGCSE Bulletin,* 38**,** 456-457.

Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K. & Warke, Y. 2002. Managing semantic content for the Web. *IEEE Internet computing,* 6**,** 80-87.

Shute, V. J. & Psotka, J. 1996. Intelligent Tutoring Systems: Past, Present and Future. D. Jonassen (ed.) Handbook of Research on Educational Communications and Technology. Scholastic Publications.

Sicilia, M.-Á., Lytras, M. D., Sánchez-Alonso, S., García-Barriocanal, E. & Zapata-Ros, M. 2011. Modeling instructional-design theories with ontologies: Using methods to check, generate and search learning designs. *Computers in Human Behavior,* 27**,** 1389-1398.

Siddharthan, A. 2011. Text simplification using typed dependencies: a comparison of the robustness of different generation strategies. *Proceedings of the 13th European Workshop on Natural Language Generation***,** 2-11.

Siddiqi, S. & Sharan, A. 2015. Keyword and keyphrase extraction techniques: A literature review. *International Journal of Computer Applications,* 109.

Sloman, M. 2001. *The e-learning revolution: From propositions to reality,* CIPD Publishing.

Snae, C. & Brueckner, M. 2007. Ontology-driven e-learning system based on roles and activities for Thai learning environment. *Interdisciplinary Journal of Knowledge and learning objects,* 3**,** 1-17.

Soderland, S., Fisher, D., Aseltine, J. & Lehnert, W. 1995. CRYSTAL: Inducing a conceptual dictionary. *arXiv preprint cmp-lg/9505020*.

Stephen Gilfus, M. P., Michael Chasen, Daniel Cane. 1997. Blackboard, Educational technology. *http://uki.blackboard.com/.*

Stotz, S., Lee, J.S., Rong, H. and Murray, D. 2017. The Feasibility of an eLearning Nutrition Education Program for Low-Income Individuals. Health Promotion Practice, 18(1), pp.150-157.

Stoyanchev, S., Song, Y. C. & Lahti, W. 2008. Exact phrases in information retrieval for question answering. *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering***,** 9-16.

Stoyanov, V., Cardie, C., Gilbert, N., Riloff, E., Buttler, D. & Hysom, D. 2010. Reconcile: A coreference resolution research platform. Cornell University.

Stoyanov, V. & Eisner, J. 2012. Easy-first Coreference Resolution. *In Proceedings of COLING, Mumbai. Citeseer***,** 2519-2534.

Sudhana, K. M., Raj, V. C. & Suresh, R. 2013. An ontology-based framework for context-aware adaptive e-learning system. *Computer Communication and Informatics (ICCCI), 2013 International Conference on.* IEEE.

Sung, N. H. & Chang, Y. S. 2004. Business information extraction from semi-structured webpages. *Expert Systems with Applications,* 26**,** 575-582.

Tesnière, L. 1959. *Eléments de syntaxe structurale,* Librairie C. Klincksieck.

Thada, V. & Jaglan, D. V. 2013. Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm. *International Journal of Innovations in Engineering and Technology (IJIET),* 2**,** 202.

Trupti & Deshmukh 2013. Building of domain ontology using NLP. *Published in Elsevier International Conference on Recent Trends in Communication and Computer Networks -ComNet 2013-ISBN978-81-910691-6-3 ,pages 220-225*.

Uhomoibhi, J. O. 2006. Implementing e-learning in Northern Ireland: prospects and challenges. *Published in Emerald Insight Campus-Wide Information Systems,* 23**,** 4-14.

Uryupina, O. 2010. Corry: A system for coreference resolution. *Proceedings of the 5th international workshop on semantic evaluation***,** 100-103.

Uzun, Y. 2005. Keyword Extraction Using Naïve Bayes. *Bilkent University, Department of Computer Science, Turkey www. cs. bilkent. edu. tr/~ guvenir/courses/CS550/Workshop/Yasin_Uzun. pdf.*

Versley, Y., Ponzetto, S. P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X. & Moschitti, A. 2008. BART: A modular toolkit for coreference resolution. *Proceedings* *of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Demo Session.Association for Computational Linguistics***,** 9-12.

Wang, T.-I., Wang, K.-T. & Huang, Y.-M. 2008. Using a style-based ant colony system for adaptive learning. *Expert Systems with applications,* 34**,** 2449-2464.

Weber, G. & Brusilovsky, P. 2001. ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education (IJAIED),* 12**,** 351-384.

Webster, J. J. & Kit, C. 1992. Tokenization as the initial phase in NLP. *Proceedings of the 14th conference on Computational linguistics-Volume 4. Association for Computational Linguistics***,** 1106-1110.

Wentling, T., Waight, C., Gallager, J., La Fleur, J., Wang, C. & Kanfer, A. 2000. E-learning: A review of literature. Urbana-Champaign, IL: University of Illinois. Retrieved September 2001.

Wood, H. M. 1995. Computer society celebrates 50 years. *IEEE Annals of the History of Computing,* 17**,** 6.

Yang, Y., Bansal, N., Dakka, W., Ipeirotis, P., Koudas, N. & Papadias, D. 2009. Query by document. *Proceedings of the Second ACM International Conference on Web Search and Data Mining***,** 34-43.

Yarandi, M., Jahankhani, H. & Tawil, A.-R. 2012. An ontology-based adaptive mobile learning system based on learners' abilities. *Global Engineering Education Conference (EDUCON), 2012 IEEE***,** 1-3.

Yarkova, T.A., Cherkasova, I.I., Timofeeva, A.M., Cherkasov, V.V. and Yarkov, V.G. 2017. Preparing Teachers to Use New Media Visual Communications in Education. International Journal of Emerging Technologies in Learning, 12(2).

Yieke, F. A. 2005. Towards alternatives in higher education: the benefits and challenges of e-learning in Africa. *CODESRIA Bulletin,* 3**,** 73-5.

Zayed, O. H. & El-Beltagy, S. R. 2015. Named Entity Recognition of Persons' Names in Arabic Tweets. *RECENT ADVANCES IN***,** 731.

Appendix

Appendix A: The set of pattern syntax of instance keywords in C++ language

Key instance	Pattern of Key instance
While	('/while[\\ \\s\\;]*\\([^\\;\\:\\,\\)]+\\)/i',
For	('/for[\\ \\s\\;]*\\([^\\:\\)]+\\)/i',
If	('/if[\\ \\s\\;]*\\([^\\;\\)]+\\)/i',
int	('/int[\\ \\s\\;]+[^<>\\:\\.\\(\\)\\;]+\\;/i',
Double	('/double[\\ \\s\\;]+[^<>\\:\\.\\(\\)\\;]+\\;/i',
Float	('/float[\\ \\s\\;]+[^<>\\:\\.\\(\\)\\;]+\\;/i',;
Char	('/char[\\ \\s\\;]+[^<>\\:\\.\\(\\)\\;]+\\;/i',;
#include	('/\\#include[\\ \\s\\;]+[^\\;\\:\\(\\)]+\\;/i';
Void main	'/void[\\ \\s\\;]+\\main\\([^\\)]+\\)/i'
Cin	('/cin[\\ \\s\\;]/i'
Cout	('/cout[\\ \\s\\;]/i'
space	[\\ \\s\\;]/i',);
Function	('/function[\\ \\s\\;],\\([^\\,\\)]+\\)/i',

Appendix B: Learning outcomes in ACM/IEEE 2013 are defined into three tasks: Familiarity, Usage and Assessment

"for clarification, the learning outcomes in ACM/ IEEE are defined into three tasks: Familiarity, Usage and Assessment.

1. Familiarity task: *This task of mastery concerns the basic awareness of a concept. It provides an answer to the question "What do you know about this. The initial level of understanding of any topic is answering the question what the concept is or what it means. For instance, if we consider the notion of iteration in software development, this would include for-loops, while-loops and iterators. At the "Familiarity task," a student would be expected to have a definition of the concept of iteration in software development and know why it is a useful technique.*

2. Usage Task: *This task of mastery implies using or applying a concept in concrete way, which uses a specific concept in a program. After introducing the concept to the learner, it would be essential to apply the knowledge in more practical way. It provides an answer to the question "How to use it?". For instance, if we consider the concept of arrays in programming languages, a student at the "Usage" task, should be able to write or execute a program properly using a form of array.*

3. Assessment Task: This *task of mastery implies more than using a concept; it involves the ability to select an appropriate approach from understood alternatives. It provides an answer to the question "Why would you do that?". Furthermore, the student is able to consider a concept from multiple viewpoints and/or justify the selection of a particular approach to solve a problem. ". For instance, understanding iteration in software development, at the "Assessment" task would require a student to understand several methods for iteration and be able to appropriately select among them for different applications.*

Appendix C: The VARK questionnaire (version 7.8)

1. You are helping someone who wants to go to your airport, the centre of town or railway station. You would:

a. go with her.
b. tell her the directions.
c. write down the directions.
d. draw, or show her a map, or give her a map.

2. A website has a video showing how to make a special graph. There is a person speaking, some lists and words describing what to do and some diagrams. You would learn most from:
a. seeing the diagrams.
b. listening.
c. reading the words.
d. watching the actions.

3. You are planning a vacation for a group. You want some feedback from them about the plan. You would:
a. describe some of the highlights they will experience.
b. use a map to show them the places.
c. give them a copy of the printed itinerary.
d. phone, text or email them.

4. You are going to cook something as a special treat. You would:
a. cook something you know without the need for instructions.
b. ask friends for suggestions.
c. look on the Internet or in some cookbooks for ideas from the pictures.
d. use a good recipe.

5. A group of tourists want to learn about the parks or wildlife reserves in your area. You would:
a. talk about, or arrange a talk for them about parks or wildlife reserves.
b. show them maps and internet pictures.
c. take them to a park or wildlife reserve and walk with them.
d. give them a book or pamphlets about the parks or wildlife reserves.

6. You are about to purchase a digital camera or mobile phone. Other than price, what would most influence your decision?
a. Trying or testing it.
b. Reading the details or checking its features online.
c. It is a modern design and looks good.
d. The salesperson telling me about its features.

7. Remember a time when you learned how to do something new. Avoid choosing a physical skill, eg. riding a bike. You learned best by:
a. watching a demonstration.
b. listening to somebody explaining it and asking questions.
c. diagrams, maps, and charts - visual clues.
d. written instructions – e.g. a manual or book.

8. You have a problem with your heart. You would prefer that the doctor:
a. gave you a something to read to explain what was wrong.
b. used a plastic model to show what was wrong.
c. described what was wrong.
d. showed you a diagram of what was wrong.

9. You want to learn a new program, skill or game on a computer. You would:
a. read the written instructions that came with the program.
b. talk with people who know about the program.
c. use the controls or keyboard.
d. follow the diagrams in the book that came with it.

10. I like websites that have:
a. things I can click on, shift or try.
b. interesting design and visual features.
c. interesting written descriptions, lists and explanations.
d. audio channels where I can hear music, radio programs or interviews.

11. Other than price, what would most influence your decision to buy a new non-fiction book?
a. The way it looks is appealing.
b. Quickly reading parts of it.
c. A friend talks about it and recommends it.
d. It has real-life stories, experiences and examples.

12. You are using a book, CD or website to learn how to take photos with your new digital camera. You would like to have:
a. a chance to ask questions and talk about the camera and its features.
b. clear written instructions with lists and bullet points about what to do.
c. diagrams showing the camera and what each part does.
d. many examples of good and poor photos and how to improve them.

13. Do you prefer a teacher or a presenter who uses:
a. demonstrations, models or practical sessions.
b. question and answer, talk, group discussion, or guest speakers.
c. handouts, books, or readings.
d. diagrams, charts or graphs.

14. You have finished a competition or test and would like some feedback. You would like to have feedback:
a. using examples from what you have done.
b. using a written description of your results.
c. from somebody who talks it through with you.
d. using graphs showing what you had achieved.

15. You are going to choose food at a restaurant or cafe. You would:
a. choose something that you have had there before.
b. listen to the waiter or ask friends to recommend choices.
c. choose from the descriptions in the menu.
d. look at what others are eating or look at pictures of each dish.

16. You have to make an important speech at a conference or special occasion. You would:
a. make diagrams or get graphs to help explain things.
b. write a few key words and practice saying your speech over and over.
c. write out your speech and learn from reading it over several times.
d. gather many examples and stories to make the talk real and practical.