ADAPTIVE GAME-BASED E-LEARNING USING SEMANTIC WEB TECHNOLOGIES

ASAD ALI

ADAPTIVE GAME-BASED
E-LEARNING USING SEMANTIC WEB TECHNOLOGIES

NEU
2017

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

BY
ASAD ALI

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

NICOSIA, 2017

ADAPTIVE GAME-BASED E-LEARNING USING SEMANTIC WEB TECHNOLOGIES

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

By

ASAD ALI

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

NICOSIA 2017

Asad ALI: ADAPTIVE GAME-BASED E-LEARNING USING SEMANTIC WEB TECHNOLOGIES

Approval of Director of Graduate School of Applied Sciences

Assoc Prof. Dr.Nadire Cavus

We certify this thesis is satisfactory for the award of the degree of Master of Science in Computer Engineering

Examining Committee in Charge:

Prof Dr. Rahib Abiyev — Committee Chairman, Department of Computer Engineering, NEU

Assoc. Prof Dr.Melike Şah Direkoglu — Department of Computer Engineering, NEU

Asst. Prof. Dr Duygu Celik Ertugrul — Department of Computer Engineering, EMU

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name:

Signature:

Date:

To my parents…

ACKNOWLEDGEMENTS

First of all, I highly appreciate the unconditional support of my supervisor Assoc. Prof. Dr. Melike Şah Direkoglu throughout the whole research study. For sure, without her support, motivation, and guidance, this thesis would have never been possible. I also pass my gratitude towards my department for allowing me to carry out the research study and all their assistance that each member provided.

ABSTRACT

Recently technology has played a vital role in the education and huge amount of work has been done in the e-learning field to allow users to learn beyond the traditional class rooms. The emerging of Semantic Web technologies allow the experts to develop an intelligent e-learning system easily and also allow the users to access and learn the contents available on the Web. Generally, when the learning materials are presented as a quiz/game, the users engage more with the contents, and thus learn quickly and easily. Significant amount of work have been done in developing quiz-based e-learning systems but few of them integrate Semantic Web technologies. The Semantic Web is an ideal framework for e-learning system because the use of ontologies increases the reusability of the system and the Semantic Web rules are used to adapt the e-learning system and separate the adaptivity from our source code. Also different users have different backgrounds and learning capabilities, so there is always a need of developing a system which adapts itself according to the user's knowledge and preferences. In this work, we developed a novel adaptive quiz/game-based e-learning tool using Semantic Web technologies which re-used knowledge from DBpedia to allow users to learn their learning materials quickly and easily. The questions/answers of the quiz are taken from the domain of Physics, Chemistry, and Geography which are extracted from DBpedia using Protégé editor. Moreover, the adaptive system is compared to a baseline system and finally user evaluations are performed in which different users test both the adaptive and baseline system and their views and suggestions about these systems.

Keywords**:** Semantic web; jena; e-learning; user interface

ÖZET

Son zamanlarda teknoloji eğitimde hayati bir rol oynamıştır ve e-öğrenme alanında kullanıcıların geleneksel sınıf odalarının ötesinde öğrenmelerine olanak tanımak için büyük miktarda çalışma yapılmıştır. Semantik Web (anlamsal ağ) teknolojilerinin ortaya çıkışı, uzmanların akıllı bir e-öğrenme sistemi geliştirmelerine ve kullanıcıların Web'de bulunan içeriğe akıllı bir şekilde erişmesine ve öğrenmesine imkân tanımaktadır. Geneleneksel olarak, öğrenme materyalleri bir sınav/oyun olarak sunulduğunda, kullanıcılar daha fazla içerikle etkileşime girer ve böylece hızlı ve kolay öğrenirler. Yarışma tabanlı e-öğrenme sistemleri geliştirmek için önemli miktarda çalışma yapılmış ancak bunlardan bir kaçı Semantik Web teknolojilerini bütünleştirmiştir. Semantik Web, e-öğrenme sistemi için ideal bir çerçeve sunmaktadır. Çünkü ontolojilerin kullanılması sistemin yeniden kullanılabilirliğini artırır ve Semantic Web kuralları, e-öğrenme sistemini uyarlamak (değişik kullanıcılara adapte etmek) ve aynı zamanda sistem fonksiyonlarını kaynak kodundan ayırmayı sağlar. Ayrıca farklı kullanıcıların farklı geçmişi ve öğrenme yetenekleri vardır, bu nedenle kullanıcının bilgisine ve tercihlerine göre kendisini uyarlayan bir sistem geliştirmeye ihtiyaç vardır. Bu çalışmada, DBpedia'dan gelen bilgiyi kullanarak, kullanıcıların öğrenme materyallerini hızlı ve kolay bir şekilde öğrenmelerini sağlamak için Semantic Web teknolojilerini kullanan yeni bir adaptif yarışma/oyun tabanlı e-öğrenme aracı geliştirdik. Oyunun soruları ve cevapları, Protégé editörünü kullanarak DBpedia'dan çıkarılan Fizik, Kimya ve Coğrafya alanından alındı. Dahası, uyarlamalı sistem bir temel sistemle karşılaştırdı ve nihayet kullanıcıların uyarlamaları ve temel sistemi hem de bu sistemler hakkındaki görüş ve önerilerini farklı kullanıcıların test ettikleri değerlendirmeler yapıldı.

***Anahtar kelimeler*:** Semantic web (Anlamsal ağ); jena; e-öğrenme; Kullanıcı arayüzü

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

IDE	Integrated Development Environment
FOAF	Friend of a Friend
N3	Notation 3 – a format for representing RDF triples
OWL	Ontology Web Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SPARQL	SPARQL Protocol and RDF Query Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WWW	The World Wide Web
W3C	The World Wide Web Consortium
XSD	Xml Schema Definition
XML	eXtensible Markup Language

CHAPTER 1
INTRODUCTION

1.1 Introduction

Electronic learning, also known as e-learning, is a research area which utilizes electronic technologies to provide access to educational curriculum outside traditional classroom (Elena 2010). It is a just-in-time education and learning system that aims to replace the old fashioned learning system.

E-Learning generally referred as making use of Information Communication Technology (ICT) tools and technologies where (1) learners learn to utilize some use cases and (2) teachers use these technologies to teach learning materials. E-learning is also referred as online learning or virtual learning, or distributed learning which enable learners to learn with their own place and their own pace. From past one and half decade, organizations and academics have started to use e-learning technologies to educate their employees and students respectively. While new tools and technologies have been developed, e-learning have adapted itself and thus formed into a new shape.

In the past decade, Semantic Web research area is started to become a new trend of computer science. Semantic Web aims to provide machine processable data which then enables more intelligent user interfaces and applications. Although great of work has been done in the e-learning area, with the involvement of Semantic Web technologies into e-learning tools enable more intelligent, machine readable and reusable e-learning technologies. Ontology development, ontology based annotation of learning materials and semantic querying of data makes Semantic Web a suitable framework for e-learning system implementation (John, 2007).

There are a lot of quiz based e-learning technologies that are available on web. These learning tools motivate users to play the quiz as well as learn. When the quiz/game based e-learning is combined with the integration of Semantic Web tools, it does not only provide fun but also intelligence to that fun. E-learning tools within itself is divided into a number of categories; traditional curriculum e-learning and quiz/game-based e-learning. Very few work has been proposed for creating game based e-learning using Semantic Web tools, and this is the motivation behind our work in this thesis. We introduce a game based e-learning tool that combines the advantages of Semantic Web. By doing so, we use knowledge from well-known Semantic Web source, which is DBpedia. In addition we adapt the game-based e-learning to the preferences and knowledge of learners, which is the key contribution of this thesis.

1.2 Aims and Objectives

In this thesis we developed a novel quiz/game-based e-learning tool using Semantic Web. We re-used knowledge from DBpedia for the game. In particular, we developed three quiz categories, namely Physics, Chemistry, and Geography.

- One of the purpose of this thesis is to apply Semantic Web technologies to quiz/game-based e-learning. To achieve this, we introduce a novel quiz ontology.
- The second purpose is to adapt the game to different learners. For this vision, we apply a number of adaptation rules, as well as, developed a new user ontology for game adaptation.
- Instead of creating RDF data, we aim to use existing Semantic Web knowledge and uses DBpedia knowledge base for this purpose.
- We develop a game where users learn subjects like Physics, Chemistry,and Geography
- Users learn while solving quiz because if some questions are answered incorrectly, the details of that question will be displayed to the user from DBpedia in order to correct the answer in next round. In particular, we utilize rdfs:comment of corresponding DBpedia resource as a hint.
- In order to not force users, users can select their preferred category first before starting to a game.
- According to the capabilities of the users (expert or novice users), the questions are re-ordered so that most easy questions will be displayed first to the novice users and difficult questions will be displayed first to diligent/expert users. This is an adapatation we apply based on knowledge.
- `Users will able to track about the categories in which they are expert and in which category they have high score as well as users able to check their previous scores of the categories.

- Automatically preferred category of a user is predicted using semantic rules and the user is shown with this category at the beginning of the game.
- We achieve adaptation through the use of semantic rules based on existing triples.
- A summary of their scores are displayed to users at the end of the quiz game in order to motivate them. In particular, scores are illustrated graphically.

1.3 Motivation

With the increase of electronic gadgets, people have started working and playing with different kinds of technological devices. This trend have also been observed for students; they are now more interested in learning using computers and other electronic devices in order to interact with learning material and learn them.

In our view, the shift of learning process to technological devices are encouraging students to learn and concentrate on their courses than the traditional class rooms. This is why the concept of e-learning has been introduced which allows students to learn their curriculum via electronic devices. In particular, if the e-learning is through some game-based activities, it can motivate students more to learn (Guo, 2006). Besides this, most students are shy in their class rooms to ask questions if they have problem in some areas.

With the use of Semantic Web technologies, e-learning process have now been much easy and robust than ever before. Both useful for developers to create the application easily and students to learn more intelligently, Semantic Web technologies help students in interactive learning in different learning areas. We really inspired by Semantic Web technologies to develop a game- based and adaptive-based e-learning system in which users will learn their core courses like Physics, Chemistry, and Geography through a game and according to their preferences. First we talk with teachers and develop the game-based quiz based on their feedback. In particular, learners can keep track of their status like which category they have selected more, in which category they score high marks and to which class of students they belong according to their quiz scores. Like traditional class rooms, if some students do not know about a question, our application uses the semantic web concept rdfs:comment and will show a thorough detail about that question from DBpedia, which is the semantic web version of Wikipedia.

CHAPTER 2
RELATED WORK

In this chapter, we briefly describe what is Semantic Web, e-learning, Semantic Web based e-learning and adaptive e-learning.

2.1 Semantic Web Technologies

In this section, we briefly explain what Semantic Web is and how it differs than the current Web. At the end, some of the Semantic Web technologies are explained such as RDF, RDFS/OWL, Semantic Rules, and SPARQL.

2.1.1 Current Web

Current Web is developed by Sir Tim Berners-Lee. It is the combination of interconnected Web pages, called hypertext documents which span over the Internet. These Web pages contain text, images, audios, videos and can be accessed using hyperlinks and viewed by using Web browsers (Wikipedia 2006). All of these data can be accessed and exchanged using HyperText Transfer Protocol (HTTP). The Web pages are written in HTML and can be accessed using URL (Uniform Resource Locator).

A newer version of the current Web, so called Web 2.0, has taken users into new generations as compared to the original Web. With the rise of social media sites like Facebook and Twitter, has enable development of new concepts like blogs, wikis and other social media Websites. Therefore Web 2.0 also allows more easy sharing and rating of part of knowledge on websites without uploading the whole page.

Since there are millions of different web pages connected to each other via hyperlinks, getting the right information at right time is a tedious job in the current Web applications. For example, if we ask computer to “Show me all programming language books written by Wrox authors, whose price is less than $100 and number of pages less than 500”. This is beyond the capability of the current Web applications and to hit this search, we have to give the search engines intelligence and make it smarter. This is why the idea of Semantic web comes in the early 2000.

2.1.2 Semantic Web

Lee and Hendler (1999) define the Semantic Web as: “The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users.”

The idea of Semantic Web is simple: Making the Web machine understandable rather than machine readable. The current Web is only understandable by people, and applications can not understand it. Thus an application cannot communicate with other applications. Semantic Web provides standards to represent data in machine-understandable way. Thus reasoning about the users can be assisted by bringing them the relevant information. In the Semantic Web, it is also possible to re-use the exixting vocabularies. For example, a book written by an author can be described by two vocabularies: book title described by Dublin Core (Dublin Core Initiative) and author described by using the FOAF (Friend-of-a-Friend) vocabulary (Dodds 2004).

The Semantic Web does not replace the current Web, rather it is an extension of the current Web in which information has been given a well-defined meaning and thus information can be understood and processed by machines (Stumme, 2006). According to John Markoff, Semantic Web is a set of technologies that offer efficient new ways to help computers organize and draw conclusions from online data. The core of Semantic Web is the semantic layer cake which provides different components in layers to develop Semantic Web applications (Elena, 2010). One of the important components of the semantic layer cake is the “Rules” which allows computers and applications to reason about the Web content and infer new knowledge based on the existing one.

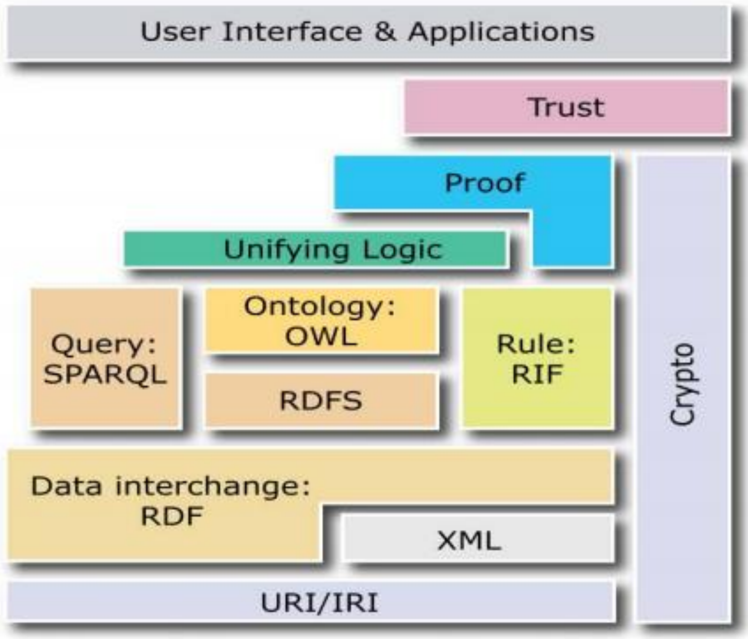


Figure 2.1 A sample XML syntax

2.1.2.1 Unicode and Uniform Resource Identifier (URI)

Unicode is a character encoding scheme which help developers to create software applications working in any language of the world. It provides a unique number for every character regardless of any language and platform.

Uniform Resource Identifier (URIs) are the addresses which uniquely identify a resource on the Web. A resource can be anything like a city, person, file, disease, food, etc. In simple words, a URI is a sequence of characters that uniquely identifies a physical or abstract resource on the Web.

A URI can be a Uniform Resource Locator (URL), a Uniform Resource Name (URN) or both. In addition to identify a resource on the Web, a URL can also be used to locate the resource and describe its primary access mechanism. If the access mechanism or network location is given in a URI, like “http” or “ftp”, then the URI becomes URL.

A typical URL is given below:

http://www.example.com/myfile.txt

ftp://filelocation.com

URN is also a subset of URI and identify the resource by name. URN can be used to refer to book names by identifying its International Standard Book Number (ISBN).

Urn: ISBN: 08764653

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

<book id="book1">

    <author>John Horton</author>

    <title>Introduction to XML</title>

    <genre>Programming</genre>

    <price>30</price>

</book>
```

Figure 2.2 A sample XML syntax

2.1.2.2 Extensible Markup Language (XML)

Extensible Markup Language (XML) is a meta-language for documents markup and allows us to define our own tags. XML gives syntax for documents markup and syntax to the structure of documents.

XML derived from Standard Generalized Markup Language (SGML), which is a language for defining markup languages. XML has a smaller and simple syntax than SGML which help developers in creating, managing and displaying documents.

A simple syntax of XML describing a book is shown in Figure 2

Some of the key advantages of XML are:

- XML provides both human and machine readable format.
- It can be used in a variety of platforms and with a variety of tools and hence provide interoperability.
- It is extensible and new tags can be created with less effort compared to SGML and an XML tag can contain any number of attributes.

- XML is W3C standard.
- The hierarchical structure of XML is suited to most types of documents (though not for all types).
- It supports multilingual documents using Unicode and information in any human language can be easily communicated.

2.1.2.3 The Resource Description Framework (RDF)

The Resource Description Framework describes resources on the Web. A resource can be anything like a person, book, country, disease, moon and which can be assigned a URI by which they can be identified. Standardized by W3C, RDF is used to describe the resources and allows to encode, exchange, and reuse the structured data on the Web.

RDF describes the resources in the form of triples, which is just a simple statement. A triple constitutes of a Subject, Predicate, and Object which forms a statement. When collection of triples combined, it forms a directed graph in which the arrows points from subject to object. The text on the arrows are called predicate or property of the triple.

A simple triple (statement) is:

Messi is a player

• Subject (Resource): Messi

• Predicate (Property): is a

• Object (Value): player

A subject and predicate in a triple must be a resource and must have assigned a unique URI, where as an object may be a resource or a simple value like name, number etc. In the example as shown in Figure 2.3, object is a resource.

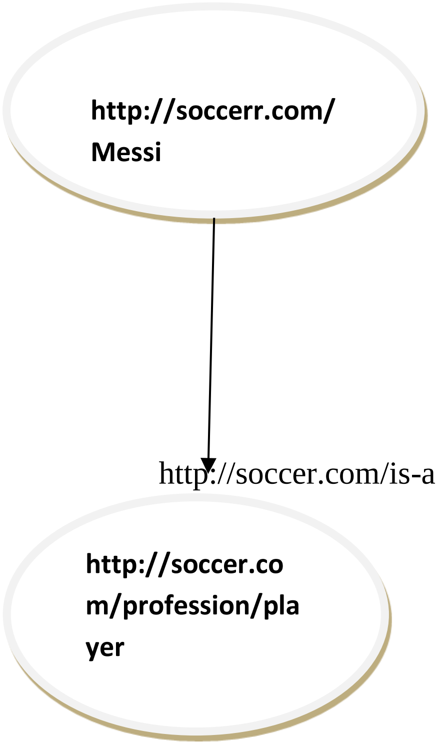


Figure 2.3 RDF Graph

RDF triples can be serialized in several ways:

- RDF/XML: It is most widely used RDF serialization which uses XML syntax. It is W3C recommendation since February 2004.
- N-Triples: It is also W3C recommendation and uses simple, plain text for exchanging and storing RDF data.
- Notation 3 (N3): It is compact and much more human readable than RDF/XML format.
- Turtle (Terse RDF Triple Language): After RDF/XML, it is commonly used and is a W3C candidate recommendation.

The RDF/XML syntax for the above graph is given below:

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

<rdf:RDF

xmlns:soc="http://soccer.com#"

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

<rdf:Description rdf:about="http://soccer.com/messey">

<soc:is-a>player</soc:is-a>

</rdf:Description>

</rdf:RDF>
```

Figure 2.4 RDF/XML Serialization of the RDF Graph

2.1.2.4 The Resource Description Framework Schema (RDFS)

Resource Description Framework (RDF) provides some basic vocabulary to triples like rdf:type and does not go in detail like what is the sub class or sub property of a particular class or property respectively.

Resource Description Framework Schema (RDFS) defined by W3C, provides more rich vocabulary than RDF. The basic constructs RDFS provides is listed below:

- rdfs:Class
- rdfs:Resource
- rdfs:subClassOf
- rdfs:subPropertyOf
- rdfs:domain
- rdfs:range

According to RDFS documentation, (Brickley,2004), rdfs:Class is the super class of everything. In addition, rdfs:Resource is anything which can be assigned to a URI and can be placed as subject or object of the RDF triple.

A property/predicate has a domain and range defined by rdfs:domain and rdfs:range. If the property is data property, the rdfs:domain is a class and rdfs:range is a data type like integer or string. If the property is object property, both the rdfs:domain and rdfs:range should be instances of classes. rdfs:subClassOf and rdfs:subPropertyOf shows a class and a property that is the sub- class and sub-property of a particular class and property respectively.

RDFS provides some basic level of reasoning. For instance, if Vitz is a type of Car and Car is the sub class of Vehicle, then the reasoner can explicitly infer that "Vitz is a Vehicle".

2.1.2.5 Ontology

Ontology is basically, the study of something which "exists" and its categories. In Computer Science, an ontology is a vocabulary which provides detail of some domain. Ontologies are used to provides a formal and shared understanding of the domain of interest.

Several authors have defined ontology in their own words but we will use how Stanford has defined the ontology. An ontology is a formal explicit description of concepts in a domain of classes (Natalya , 2000). It also provides the properties, attributes and relation between the classes of domain. The ontology with all of its associated data is then called knowledge base.

An ontology can be created using the following steps:

- Define the concept (classes) of domain.
- Arrange the classes in sub class/super class hierarchy (this hierarchy is called taxonomy).
- Describe the relation and attributes of the relation.
- Create the real world instances of the classes.

Ontologies have some key advantages which are described by (Natalya , 2000). Ontologies are used to share common understanding of the structure of information, allows reuse of domain knowledge, and make explicit domain knowledge.

2.1.2.6 Web Ontology Language (OWL)

The web ontology language is the ontology specification of W3C. OWL is a standard knowledge representation language formally recommended by W3C in 2004 and is compatible with eXtensible Markup Language (XML) as well as other W3C standards. OWL extends both RDF and RDFS and provides more variety of vocabularies and reasoning capabilities.

The basic ontology which have two classes is shown in Figure 2.5 below.

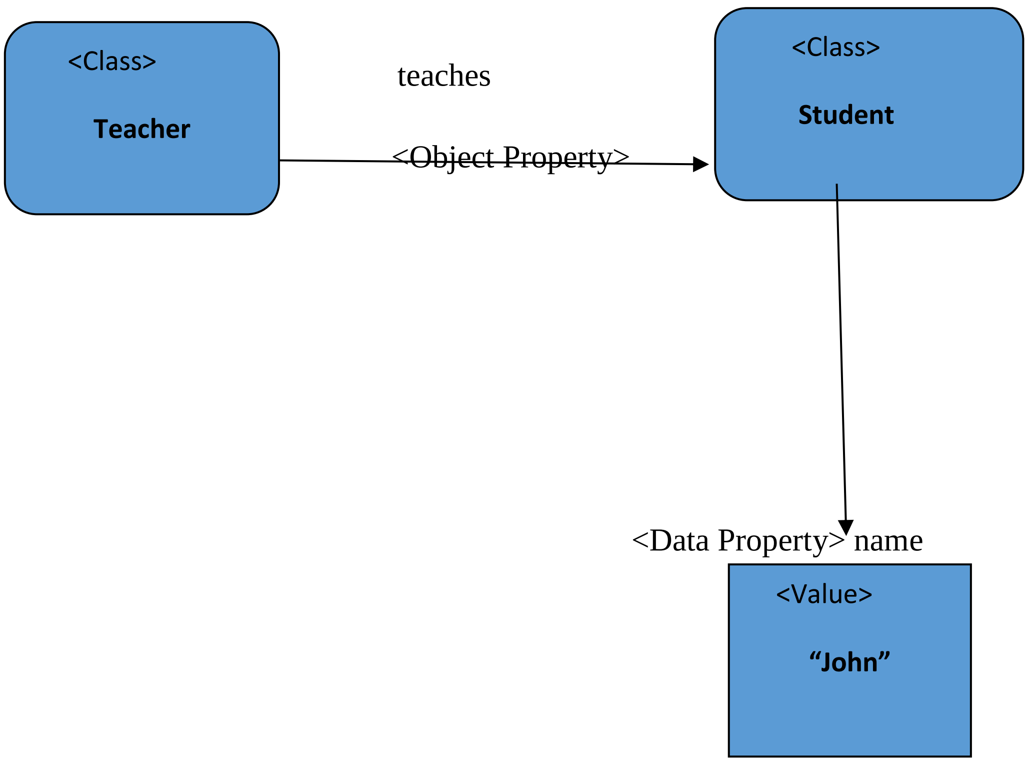


Figure 2.5 An RDF Example

OWL can be used to replace some RDF and RDFS relations such as owl:Class can be used for rdfs:Class. In addition, rdf:Property is replaced with owl:DatatypeProperty when the property is data type and owl:ObjectProperty when it is object property.

OWL has three sub languages OWL Lite, OWL DL and OWL Full, depending on the expressivity.

- OWL Full: It is the union of OWL and RDF syntax. It provides maximum expressiveness but with no computational guarantee.

- OWL DL (Description Logic): OWL DL has the closest correspondence to description logic which is more expressive without losing computational completeness.
- OWL Lite: Users with low requirements and simple modeling need use OWL Lite and include simple constraint features.

An example, owl data in RDF turtle syntax is shown in Figure 2.6

```
@prefix rdf :< http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix owl :< http://www.w3.org/2002/07/owl#>.
@prefix:  http://someuri.org/
:Teacher   rdf:type   owl:Class .
:Student   rdf:type   owl:Class .
:name      rdf:type     owl:DatatypeProperty .
:name      rdfs:domain  :Person .
:name      rdfs:range     :String .
:teaches    rdf:type     owl : ObjectProperty .
:teaches   rdfs:domain  :Teacher.
:teaches   rdfs:range     :Student
:Teacher    :name " John "^^xsd : string.
:Student     :name " Bush "^^xsd : string.
:Teacher      :teaches   :Student .
```

Figure 2.6 OWL data in RDF turtle syntax

2.1.2.7 SPARQL

SPARQL Protocol and RDF Query Language (SPARQL) is the standard RDF query language. Recommended by W3C since 2008, SPARQL resemble the SQL language and have the same SELECT, WHERE, FILTER BY terms. SELECT, CONSTRUCT, DESCRIBE, and ASK queries can be used in SPARQL to query the existing RDF triples or create new ones.

SPARQL is used to query RDF graph which comprise of triples. A typical SPARQL query consists of triples which extract subjects, predicate, and objects from RDF graphs. Variables are used in the query to get the required results from the graph and a question mark (?) should be used before the variable name. Prefixes can be used to define the Uris of the triples used.

The Figure 2.7 shows a basic SPARQL query which gets the capital city of Turkey from DBpedia (Semantic Web version of Wikipedia).

```
PREFIX dbr : <http: //dbpedia.org/resource/>

PREFIX dbo : <http: //dbpedia.org/ontology/>

SELECT ? capital

WHERE {

 dbr:Turkey dbo:capital ?capital. }
```

Figure 2.7 A SPARQL query example

UNION, OPTIONAL, FILTER BY, ORDER BY etc. are used in the complex SPARQL queries to get data from multiple graph, restricts the result by some conditions and sort the results in ascending or descending order. Figure 2.8 shows a SPARQL query which uses OPTIONAL keyword. It will extract the person name from the graph and if there is person’s age in the graph, the SPARQL engine will display it as well. If the OPTIONAL keyword is not used and there is no age information in the graph, nothing can be displayed.

```
PREFIX rdf:<http://www.w3.org /1999/02/22 rdf –syntax-ns#>

PREFIX foaf:http://xmlns.com/foaf /0.1/

  SELECT ?person  ?name

WHERE

 ?person   rdf:type   foaf:Person.

 ?person  foaf:name ?name OPTIONAL {

?person foaf:age ?age .}

                                        }
```

Figure 2.8 A SPARQL query example using OPTIONAL

2.1.2.8 Rule Engine/Reasoning

There are two basic types of reasoning used in Semantic Web: Ontology based reasoning and Rule based reasoning. Ontology based reasoning is useful for classification based reasoning and it is based on RDFS and OWL axioms. It does not require any rule engine.

Rules based reasoning need a rule engine and a language for representing the rules. Semantic Web Rule Language (SWRL), Notation 3 (N3) logic, and Rule Interchange Format (RIF) are basic rule definition languages. Jena rules are another type which needs a rule engine.

- SWRL's basic form is XML and also support human-readable form. It is supported by Protégé ontology editor and also supported by reasoners like Pellet and Hermit. It provides unary predicates to describe classes and binary predicates to describe properties.
- Notation3: It is also called N3 for short, and is considered human readable and support to write formulas inside rules. It supports a reasoning engine CWM, written in Python and is open source.

- Rule Interchange Format, RIF in short, is a collection of dialects which intends to share and exchange rules in semantic web based rules system**.** There are many rule languages available and RIF is used to exchange rules between these languages, RIF supports three dialects: Core Dialect, Basic Logic Dialect and Production Rule Dialect.

- Jena Rules: Jena is a Java API used to build Semantic Web applications. Jena reasoner support various rules which are built in as well as user defined rules. The basic rules supported by Jena reasoner are discussed briefly below:
  Transitive Rule: Transitive rule says if A is brother of B and B is brother of C, then A is brother of C.

OWL Inverse: This rule states if A is a father of B, then B is the son of A. The instance of the property sonOf will be inferred using owl inverse rule.

Jena Generic Rules: These are user defined and widely used rules which users implement according to their needs and requirements. Jena generic rules can be either Forward-chaining or Backward-chaining. The input of forward chaining are rules and data, while its output is “extended data”. The input of backward chaining are rules, data plus statement. While the output is statement true or false.

An example of Jena generic rule can be the following:

If a student1 takes courseA and student scores 80% in that course, then student 1 is a diligent student.

```
(?x  http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.myOntology#Student) "

            + "( ?x  http://www.myOntology#takesCourse  ?course1 )"

            + "( ?course1  http://www.myOntology#Obtainedscore  ?score )"

            + "greaterThan(?score, 80) "

   + " -> (?x    http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.myOntology#DiligentStudent )]"
```

Figure 2.9 Sample Rule using Jena Syntax

2.1.2.9 User Interface

User Interface (UI) is the final layer in semantic web layers cake which provides the users of the system to communicate and interact with the application. Along with Cryptography and Trust, UI is another semantic web technology which is not standardized yet and will be implemented in future.

2.2 E-Learning

Earlier it was called online learning, the e-learning term originated from 1980’s. There is no specific and agreed definition of e-learning and many authors have defined it differently in their work. It is a mood of learning which extensively make use of Information & Communication Technologies (ICT). (Rosenberg, 2001) proposed the following definition of eLearning: “the use of Internet technologies to deliver a wide array of solutions to increase the knowledge and performance.”

According to (Hawkins, 2005), E-Learning has changes from a fully-online course to using technology to bring part or all of a course that is free of permanent time and place. (OECD, 2005) define e-learning as: “the use of information and communication technologies in multiple processes of education to support and improve learning in institutions of higher education, and allow information & communication technology as addition to traditional classrooms, online learning or combination of the two modes".

(European Commission ,2001) also provides the definition of e-learning as the the usage of new electronic technologies to enhaunce the learning quality by allowing access to facilities and services as well as remote exchanges and collaboration.

Unlike face to face learning which is called, c-learning, in e-learning users interact with their course materials and tutors from their preferred location and preferred time via electronic devices. There are three approaches a student can learn in the e-learning process: Independent learning, Facilitated learning and Collaborative learning. In independent learning, a student learn learning materials in his own environment and according to his own schedule and is not dependent on a facilitator. Of course, he will have access to the tutors and other facilitator, but if and when he wants. In facilitated learning, the control of the learning process are with the students. The role of the tutor, more or less, is just to facilitate students and provide learning materials. This method is usually followed in university education where sudents learn from each other when they reach to some solutions. In collaborative learning, the learning process takes place between groups of people. Bases on feedback of one another, students learn their course modules.

According to (Alghatani , 2003), e-learning can be of two types: computer based and internet based e-learning. In computer based e-learning, the ICT devices and other software and hardware are used in the learning process whereas in the internet based e-learning learning contents are available on the Web and provides links to the materials which are placed on internet.

E-Learning is a research area which utilizes electronic technologies to allow access to the learning materials outside of a classroom. It is a just-in-time education and learning system which aims to replace the old fashioned learning system.

The following are some advantages and disadvantages of the e-learning method.

- Compared to classroom based learning, e-learning is cost effective to deliver.
- It is self-paced in that one can learn according to his pace and capability.
- Learners can skip the unnecessary modules and thus it is faster process to learn.

- Learners learn with their preferred location and time.
- While in traditional class rooms, different teachers teaches same subjects with different course materials, e-learning process is consistent.
- Course content can easily be updated and will rapidly be available to all users

On the other hand, eLearning may have some shortcomings and risk as below:

- The development of e-learning system can be very costly.
- While in face to face learning, students deliver presentations, ask questions, collaborate with teacher and staff member, e-learning method does not broadcast your skills and courage.
- In some difficult questions which need extensive explanation of teacher and guidance from other class mates, e-learning method fails to clarify you the problems.
- If there are some exams, quizzes, assignments, it is very difficult to control and monitor cheating in e-learning system.
- E-learning cannot be applied to each and every field and courses which requires huge lab works and practical works with instruments needed, interactive and face to face system of learning is mandatory.

2.3 Semantic Web and E-Learning

E-Learning is just-in-time and a faster delivery time when compared with the traditional face to face learning. It's a cost effective learning method in which carbon footprint can be reduced to a substantial extent. It is more flexible than the class room based learning in that one can learn at their own place and own pace.

According to (Haghshenas, 2013), the current web is a robust service for research and education, but its usage is hindered by the lack of ability of the user to navigate easily the huge information he/she requires. Semantic Web can be the best option to cope with this problem because its architecture support both the Web contents and its associated semantics. Ontologies, ontology based annotation of learning materials and semantic querying of data has enabled Semantic Web an ideal choice for developers to create semantic based e-learning applications.

The Semantic Web has opened new era for E-learning by adding some powerful features to the Web. Key property of the Semantic Web architecture (common-shared-meaning, machine-process able metadata), enabled by a set of suitable agents seems to be powerful enough to satisfy the e-Learning requirements: fast, just-in-time and relevant learning (Guo and Chen 2006). Users can find relevant material very easily and if the system is adaptive, users can find information according to their preferences.

2.3.1 Ontologies in E-Learning

With the increase in e-learning applications, it is obvious that different authors of the application will use different technologies and thus it will be a tidy job to combine different learning materials. Besides, developers, the users of the applications like students and teachers may have different backgrounds so there is a need to establish a mechanism of a shared understanding of vocabularies. Nothing is more suitable and powerful than Semantic Web ontologies to achieve this goal.

Ontologies, which can be used to describe the shared meaning of the vocabulary, are used in different ways inside the e-learning applications. Sharing of domain data, assessment and personalization, reasoning and curriculum modeling can be used through ontologies.

Ontologies can be used in three ways in the e-learning systems.

- Content Ontology: It is used to describe the concepts, relationship between these concepts and domain of learning materials like Physics, Geography etc. Content ontology also provide properties of the concept like "part of". For instance, quantum physics is part of Physics domain.
- Pedagogical Ontology: Also called contextual ontology, it presents the learning materials in various contexts like tutorials, diagrams, assignments and their solutions. For example, if the e-learning is a quiz type, it provides an explanation if user solve a question in correctly.
- Structure Ontology: It describe the structure of the contents of learning materials and the associated navigation through which those contents can be reached.

The basic conceptual e-learning architecture is shown below:

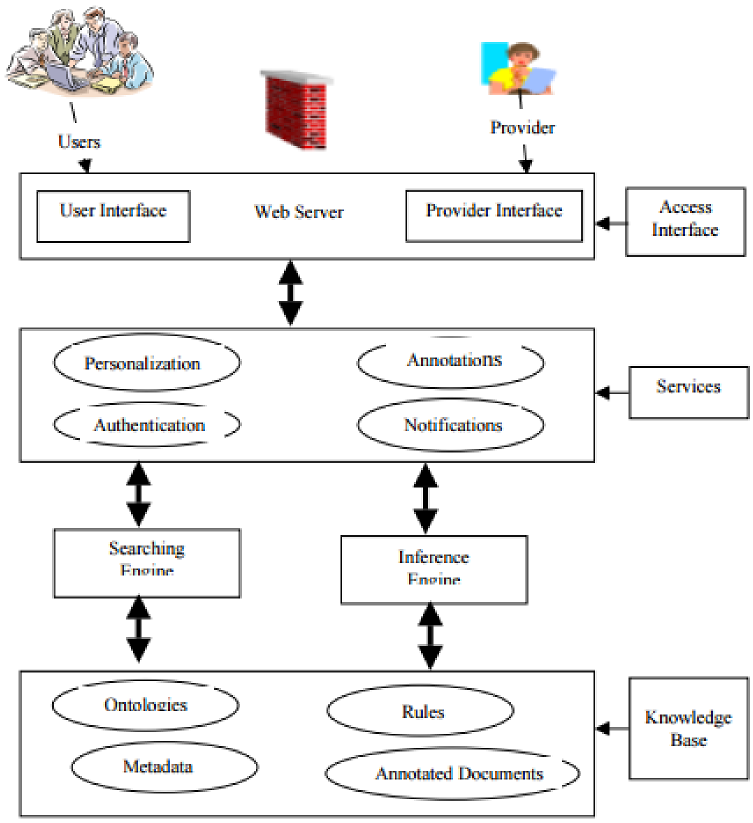


Figure 2.10 An Example Of Conceptual Semantic E-Learning Architecture

2.4 Elements of E-Learning Architecture

- Knowledge Base: This is the main and core component of the e-learning architecture where most of our data are saved. Our ontologies, rules, meta data, and educational resources like course description are stored here.
- Search Engine: It allows to search the required data and provides methods to query the knowledge base.
- Inference Engine: This component executes our inference rules and based on that rules, generate new knowledge based on existed one.
- Services: It provides services like adaptation and personalization, the annotation of course materials etc.
- Access Interface: It provides the user interface through which students, teachers and other administration interact with the system and work with data.

2.5 Adaptive E-Learning

In recent past, information technology has played a vital role in e-learning process and has brought rapid and significant changes in it. Different users/learners have different background, different capabilities and different preferences and learning will be quite bore if it is designed against their interest and preferences. In recent past, work has been done to develop e-learning system which adopts itself according to user requirements, behavior, and preferences and this kind of e-learning is called adaptive e-learning. Adaptive based learning has proved to be very effective because it not based on fixed and same learning content path. Adaptive e-learning is an approach in the e-learning system that allows users or students to learn according to their understanding by adjusting the system's navigation, presentation and contents according to their knowledge and preference (Carmona ,University of Aveiro). This is one of the main advantage of e-learning system that it adapts itself unlike traditional class rooms learning which does not care much about learner's background and behavior. According to (Cronbach ,1977), Adaptive e-learning is based on the thought that different learners have different learning abilities and that different educational settings can be more appropriate for one type of learner than for another, according to ATI, or aptitude x treatment interaction.

The ultimate aim of adaptive e-learning is to provide the users of the system the right contents to the right students and at the right time. Intelligent tutoring system and Adaptive hypermedia system are the two main areas where adaptive e-learning has been used.

Recently, many works have been done in adaptive e-learning using Semantic Web technologies like ontologies. (Sangineto, 2008) proposed an adaptive e-learning system which produced personalized courses by combining learning material via statistical knowledge which is represented using ontologies. (Henze, 2004) also proposed a personalized e-learning framework using Semantic Web which uses ontologies in three types of resources: domain, users and observations. (Chung 2008) worked on a personalized mobile English vocabulary learning system which depends on Item Response Theory and learning memory cycles, and suggests suitable English vocabulary for learning according to individual learner's vocabulary abilities and memory cycles. (Montazer, 2009) proposed a personalized multi agent system which is based on item response theory (IRT) and artificial neural network (ANN) which presents adaptive tests (based on IRT) and personalized recommendations (based on ANN).

2.6 Game-based e-learning

Games provide a unique structure to support traditional teaching and learning strategies (Boyle, 2011). Game-based learning allow learners to freely analyze, plan and experience things without any difficulty. Accoding to (Teed, 2004), any activity can be a game if it has:

- Competition: motivation to win and cross the scores of the opponents allows players/users of the game to improve their performance.
- Engagement: Learners in game are so engrossed that they are not ready to stop till the end of the game.
- Rewards: Learners are more excited throughout the game that they will get a reward/points at the end of game.

Game-based learning are used to improve the performance of learners compared to the learners who uses traditional e-learning methods. Learners play the game several times for mastering it and thus becomes master in the lesson taught in the game. The game-based learning are more effective than traditional learning in that it is highly engaging, easily convey the knowledge to the real world environment, immediate feedback in response to any mistakes, and the speed of learning depends on the individual capabilities.

There are game-based learning approaches, but few of them integrate Semantic Web technologies. For example, in the work of (Bratsas, 2012) presents a web game which uses Greek DBpedia to extract knowledge. It is an educational quiz game which focuses on native Greek speakers. The users of the quiz game are primarily school students. At the end, users evaluation are also conducted.

2.7 Uses of Semantic Web Technologies in Adaptive E-learning

The involvement of Semantic Web technologies in the e-learning systems allow us to develop an intelligent, re-usable e-learning sytems. The usage of otologies, the ontology based annotations of learning materials and the SPARQL queries to get data makes the Semantic Web an ideal framework for e-learning systems. Also, Semantic Web technologies provides a suitable plateform to develop an adaptive based e-learning system which involves the reasoning capabilities of the Semantic Web rules and the highly expresivity of the SPARQL queries. The following are some reasons to use Semantic Web in our adaptive e-learning system.

- Semantic Web provides more intelligent searches, navigations and inferences.
- Jena rules are standard and can be easily used to adapt the e-learning systems.
- The use of Jena rules and user ontology allows us to separate the adaptivity from our (Java) code because the rules only update the ontology and not the code.
- The use of ontologies increases the reusability of our e-learning system. For instance, if we have to create a quiz-game for another domain i-e medical, all we need to do is to change the knowledge base of the domain.

Moreover, the traditional server based adaptive e-learning systems are very expensive, difficult and time consuming to implement while the involvement of Semantic Web technologies make it inexpensive, simple and flexible to implement.

2.8 Related Work

Several work has been done in the e-learning area which has integrated the Semantic Web technologies and provides the adaptivity but all these e-learning systems are content based. There is always a need to develop a system which should be game-based rather than the traditional e-learning system. For example, in the work of (Bratsas, 2012) presents a web game which uses Greek DBpedia to extract knowledge. It is an educational quiz game which focuses on native Greek speakers. It allow users to learn easily and quickly compared to the traditional e-learning system. But it is just a simple baseline system which does not provide any adaptivity.

In our proposed we have developed a game-based e-learning system using Semantic Web technologies which adapts itself according to the users knowledge, capabilities, and preferences. The Table 2.1 summarizes the related work and shows the diffence of our approach from the existing work.

Table 2.1 Comparing features of the related work

Related Work	Adaptive-Based	Game-Based	Integrate Semantic Web Technologies
Adaptive Learning Management System Using Semantic Web Technologies	Yes	No	Yes
A Personalized Adaptive E-Learning Approach Based On Semantic Web Technology	Yes	No	Yes
Ontology-Driven E-Learning System Based On Roles And Activities For Thai Learning Environment	No	No	Yes
A Personalized Adaptive E-Learning Approach Based On Semantic Web Technology	Yes	No	Yes
Game-Based Learning With Ubiquitous Technologies	No	Yes	No
Introduction to the Special Section on Game-based Learning: Design and Applications	No	Yes	No
Semantic Web Game Based Learning: An I18n approach with Greek DBpedia	No	Yes	Yes
Proposed Work	Yes	Yes	Yes

CHAPTER 3
SYSTEM ARCHITECTURE

3.1 System Architecture

In this chapter we will briefly discuss various components of the proposed game-based e-learning tool that uses Semantic Web technologies. Our e-learning system consist of three parts: user interface, adaptation module, and knowledge base.

The Figure 3.1 shows the basic architecture of our quiz system. The tools we have used in our work are Protégé editor, Java langaue, Netbeans IDE (Integrated Development Environment), Jena APIs and SPARQL queries to get data from ontology. All of our quiz data are stored in quiz ontology, created in Protégé 4.3. Data are extracted from DBpedia knowledge base by providing links to DBpedia resources inside Protégé editor. We have used Jena methods to connect to our ontology and read data from the ontology. SPARQL queries are used to load the quiz data in our Java application. All of the quiz users are created using Jena rules. After user complete the quiz, all of his/her data are stored in user ontology.

The quiz system contains questions from physics, chemistry, and geography categories and the data comes from DBpedia. Similarly, the questions have two levels: Easy level and Expert level which depends on the complexity of questions. Besides, the levels, we have divided each and every question from 1-5 both in easy and expert level, with 1 shows the most easy question and 5 shows the most difficult question. So both easy and expert level questions have a degree of complexity. The data is stored in an owl file which is created using Protégé editor and DBpedia links are also given to each resource used in the quiz game. All of our data is stored in two owl files, user.owl and quizontology.owl, which describe user data and quiz domain knowledge respectively.

user.owl also store user information like: percentage of each category, scores, preferred category, high score category and class of user i-e the user belongs to diligent, average, or novice class based on his/her performance in the game. The most recent scores of the users are stored in user.owl files while we also keep track of user scores in previous games in backup.owl.

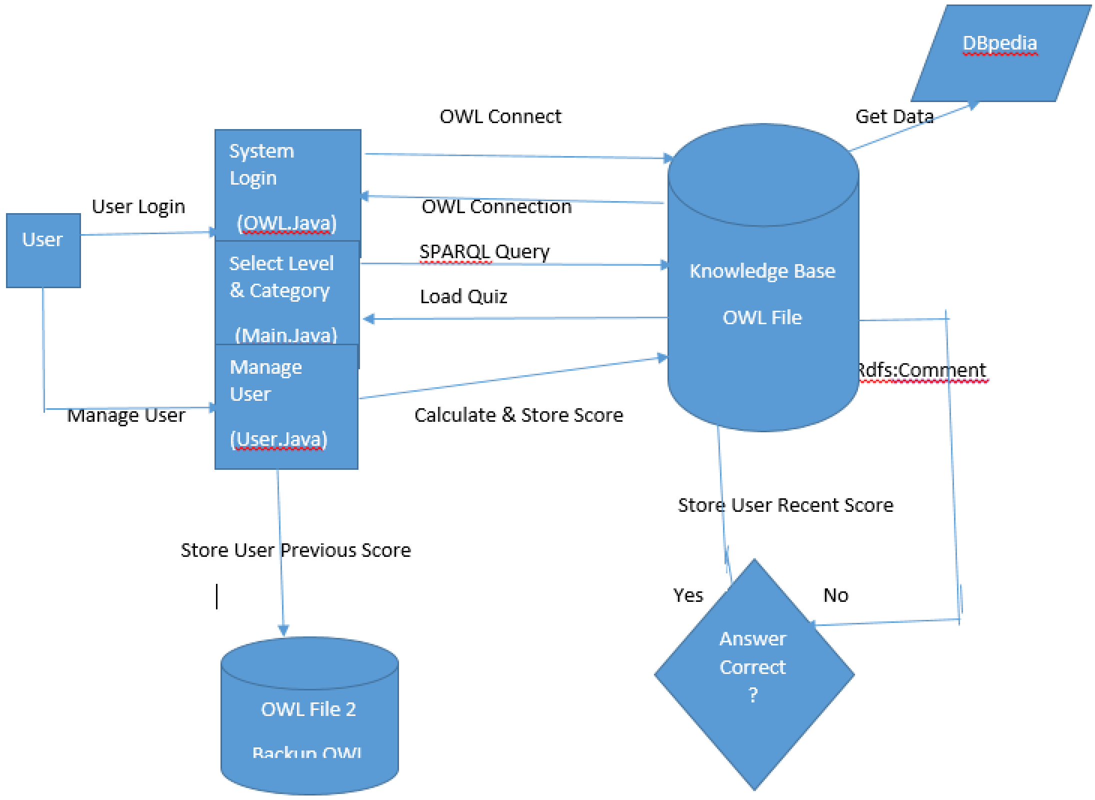


Figure 3.1 Proposed System Architecture

If the user selects a wrong answer, we display an explanation of right/correct answer in a text field so that user learn also while playing the quiz game. The explanation of the answer comes from DBpedia rdfs:comment datatype property.

3.2 Quiz Ontology and Knowledge Base

The data of our quiz game is stored in ontologies, created in Protégé editor. We have two ontologies here, one to load and store quiz related data and the other for adaptation of the game to different users. There are fifteen questions per category in knowledge base, and since there are three categories (Physics, Chemistry, and Geography), two levels of each category (Easy and Expert), we have a total of one ninty questions.

3.2.1 Quiz Ontology

All the data of the proposed quiz system are stored in the ontology quizontology.owl. It has classes, data properties, object properties and instances. The following are classes of quiz ontology.

- Question: It describes the questions of the quiz.
- Answer: It shows the the right answers of the question.
- Category: This class describes the categories (Physics, Chemistry, and Geography) which are the three different classes that are used in the quiz.
- Level: It describes the level of the questions. Two levels used (Easy and Expert) depending on the complexities of the questions.
- Points: It shows the scores for each question. Easy level questions have two points and expert level questions have four points.

An additional class "Complexity" is given in adaptive ontology which shows the complexity level of each and every question, both in easy level and expert level questions. A question has an object property called hasComplexity which has domain Question class and range as Complexity class. For example, we rank each question, both in easy and expert level, between 1-5, with 1 depicts the easiest questions and 5 shows most complex question.

The object properties of the quiz ontology are given below:

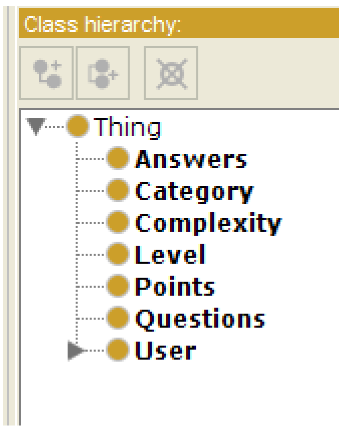
- hasAnswer: It shows the answer of a particular question.
- hasQuestion: It is the reverse property of has Answer and describes the question of related answer.
- hasLevel: It shows the level of the questions and its members are Easy and Expert level.
- hasScore: It shows the points for the questions. Each easy level question have 2 points and expert level have four points.
- hasCategory: It describes the category of questions and have members Physics, Chemistry, and Geography.

Similarly, some of the data properties of the baseline ontology are given below:

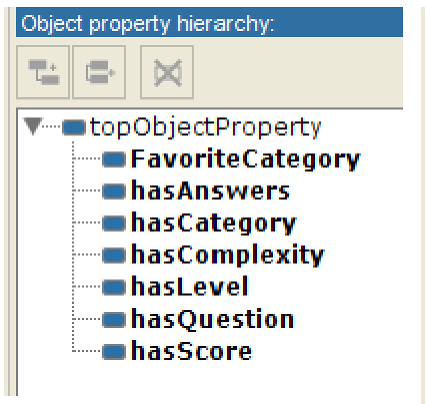
- Score: It shows the user score of a particular category

- ChoiceOne: It shows the first choice among four choices for a question given.
- ChoiceTwo: It shows second choice.
- ChoiceThree: It shows the third choice.
- CorrectChoice: It shows the correct choice/answer of the question.

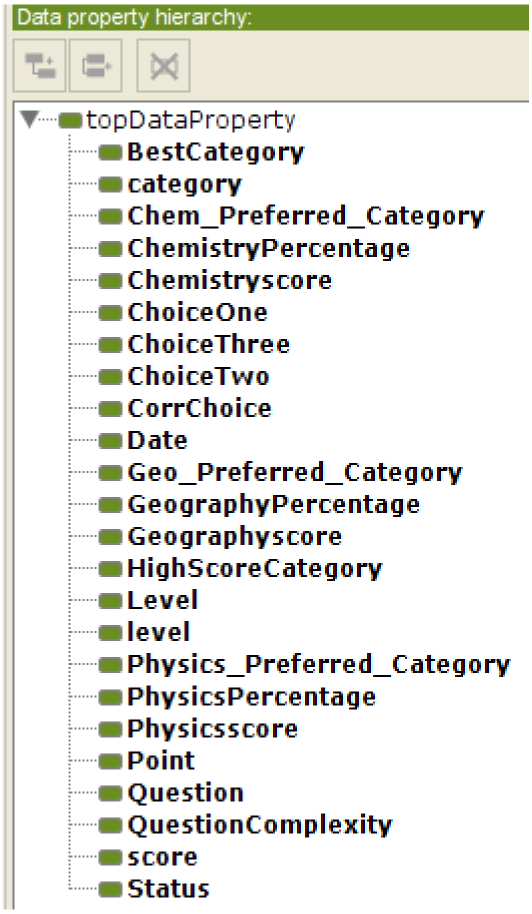
Figure 3.2 shows an overview of the quiz ontology from Protégé ontology editor. 3.2(a) shows quiz ontology classes, (b) shows object properties, and (c) shows quiz ontology data type properties respectively.



(a) Classes of Quiz Ontology



(b) Object Properties of Quiz Ontology



(c) Data Properties of Quiz Ontology

Figure 3.2 Quiz Ontology

3.2.2 DBpedia as a Knowledge Base

Wikipedia is the largest knowledge base of mankind and 7th most popular website in terms of users visit. It is available in more than 280 languages and contain more than 3.5 million articles (Wikipedia 2015) but it has some limitations like its search capability is not robust and just limited to keyword searching. Also there are some inconsistencies due to the duplication of information and different language editions.

DBpedia (DB for database) is the Semantic Web version of Wikipedia and allows us to get structured information from Wikipedia (Bizer, 2009). It allows to semantically query Wikipedia and link other dataset on the web from Wikipedia. DBpedia information is available in more than 120 languages. DBpedia knowledge is extracted from Wikipedia by using an automated extraction algorithm. DBpedia first introduced in 2006, now contains more than 2.46 billion triples, where 470 million triples from English DBpedia. Since it contains triples, it can be queried conceptually and practically using SPARQL. For example, if we want search of all the universities in Cyprus, it can be queried via a simple SPARQL query inside Semantic Web application(s) or online through DBpedia SPARQL endpoint.

One of the disadvantage of DBpedia is that it is not synchronized with Wikipedia on a regular basis. Wikipedia databases are created and edited on daily basis and there is no guarantee that DBpedia information are synchronized with the most recent edition of Wikipedia and thus there is no guarantee that everything available on Wikipedia will be available on DBpedia. To cope with this problem, people from Leipzig University introduced what is called “Live extraction” of DBpedia which works on a continuous stream of updates from Wikipedia and processes that stream on the fly (Mohamed Morsey, et al, 2012). Live extraction of DBpedia allows to be up-to-date with Wikipedia knowledge base with a minimal delay of only few minutes rather than few months delay.

All of our quiz data are taken from DBpedia and the answers of the quiz questions are DBpedia resources. We have provided a link to it inside Protégé ontology editor. Figure 3.3 shows the quizontology.owl which depicts a DBpedia resource Centripetal Force is annotated in Protégé editor. Similarly we have done this for all the data/answers of the quiz game.

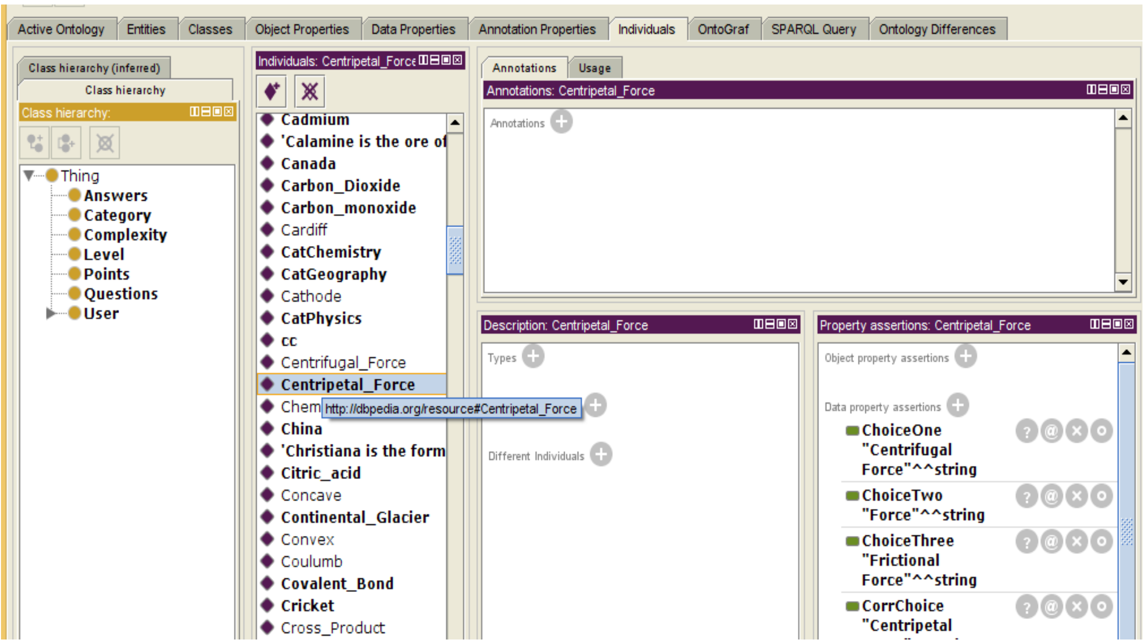


Figure 3.3 DBpedia Link of a Resource

3.3 User Ontology

Adaptation is provided by the use of a user ontology that we have developed. We created data properties like PreferredCategory and HighScoreCategory to store user preferences and achievements.

One important class in our user ontology is User class and all of the adaptivity takes place in this class. We have divided the User class into NoviceUser, AverageUser and DiligentUser based on the user's percentage in the game. These classes are sub divided in terms of the category in which they appear. For instance, a Novice user is divided into PhysicsNovice, ChemistryNovice, and GeographyNovice. If a user appear to Physics quiz and score less than 50% score, he/she will be assigned automatically to PhysicsNovice user and similarly for other categories. For example, the hierarchy of User class is shown in Figure 3.4

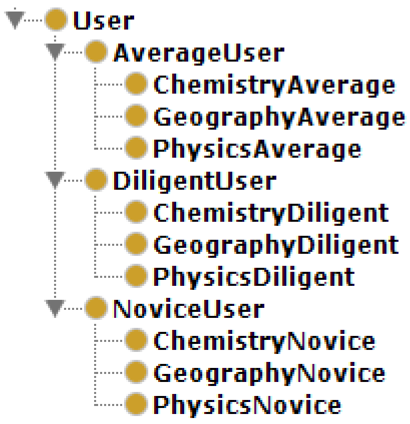


Figure 3.4 User Class from Protégé Ontology Editor

3.4 Adaptivity and Semantic Rules

Adaptation is achieved based on user's background knowledge, capabilities and preferences.

We have developed a number of adaptationJena API provides a rule based inference engine which deduce knowledge using variety of Jena rules. Thus if we have an ontology or knowledge base, we can easily make inference on it using some Jena rules and reasoner. The structure of the Jena reasoner is shown in Figure 3.5.

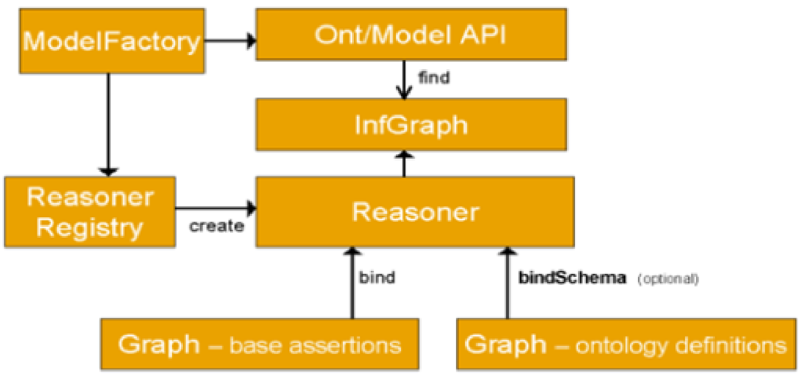


Figure 3.5 Structure of Jena Reasoner

3.4.1 User Categorization

We have used Semantic Web's Jena rules to assign each user to a different class according to their performan4.1.ce in the quiz. Jena generic reasoner are used to execute the rules and categorize users into different classes. Figure 3.6 shows jena rule in order to assign a user to physics average class based on his/her performance in physics questions. This rule says if a user is a type of User class and his/her percentage in physics category for a particular quiz is between 50% and 80%, then the user will be in the "PhysicsAverage" class. Similarly, Figure 3.7 shows a rule which will assign a user to "GeographyDiligent" class if he/she scores more than 80% in geography quiz.

```
[rule1 :(?x http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#User)

( ?x http://www.semanticweb.org/t/ontologies/2016/7/myOWL#PhysicsPercentage  ?marks
)" greaterThan(?marks, 50), lessThan(?marks,81)

     ->  (?x http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#PhysicsAverage )].
```

Figure 3.6 Jena Rule for Average User in Physics Category

```
[rule1 :(?x http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#User)

( ?x http://www.semanticweb.org/t/ontologies/2016/7/myOWL#GeographyPercentage
?marks )" greaterThan(?marks, 80))

     ->  (?x http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#GeographyDiligent )].
```

Figure 3.7 Jena Rule for Diligent User in Geography Category

Jena rules are used to implicitly assign instances to users based on its inference capabilities.

3.4.2 Question Re-ordering

The system display the questions to users in different orders according to the capabilities and performance of users in the quiz game. For instance, if a user is a novice user (From Novice class), the system will display the questions to him/her in ascending order i-e the easiest questions with complexity level of 1 will be displayed first and the most difficult questions with complexity level 5 will be last in the order, and intermediate questions with complexity from 2-4 will be in the middle order. Similarly, for diligent users, the questions will be displayed in descending order, with most difficult questions are displayed first and least difficult questions are displayed at the end. Questions will be in random order for users belongs to Average class. The Figure 3.9 shows the order of questions in Protégé, with the most difficult questions first and the most easy in the last displayed to users of Diligent class. In particular, we use SPARQL queries to perform the adaptation based on different user's class.

3.4.3 Preferred Category

User ontology also keep track of the user's preferred category and store it in userontology. A user's preferred category is one which users have selected the maximum number of time while playing the quiz game. For example, if a user has selected Physics category five times, Chemistry three times, and Geography category six times, then the user's preferred category is Geography. The provided adaptation is, the Geography (Preferred) category will appear the first in the drop down list when he/she login to the system. If a user have another category as preferred category, then that category will appear the first in the list. At the end of the quiz, we show the user's preferred category graphically.

The Jena rule that automatically infers the preferred category of a user for Geography category is shown in Figure 3.8

```
rule5:(?x http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#User)
( ?x  http://www.semanticweb.org/t/ontologies/2016/7/myOWL#Physics_Preferred_Category   ?cat1 )
( ?x  http://www.semanticweb.org/t/ontologies/2016/7/myOWL#Chem_Preferred_Category  ?cat2)
( ?x  http://www.semanticweb.org/t/ontologies/2016/7/myOWL#Geo_Preferred_Category  ?cat3 )
```

```
greaterThan(?cat3,?cat1), greaterThan(?cat3,?cat2)

-> (?x  http://www.semanticweb.org/t/ontologies/2016/7/myOWL#FavoriteCategory

http://www.semanticweb.org/t/ontologies/2016/7/myOWL#CatGeography )
```

Figure 3.9 Jena Rule for Geography Preferred Category

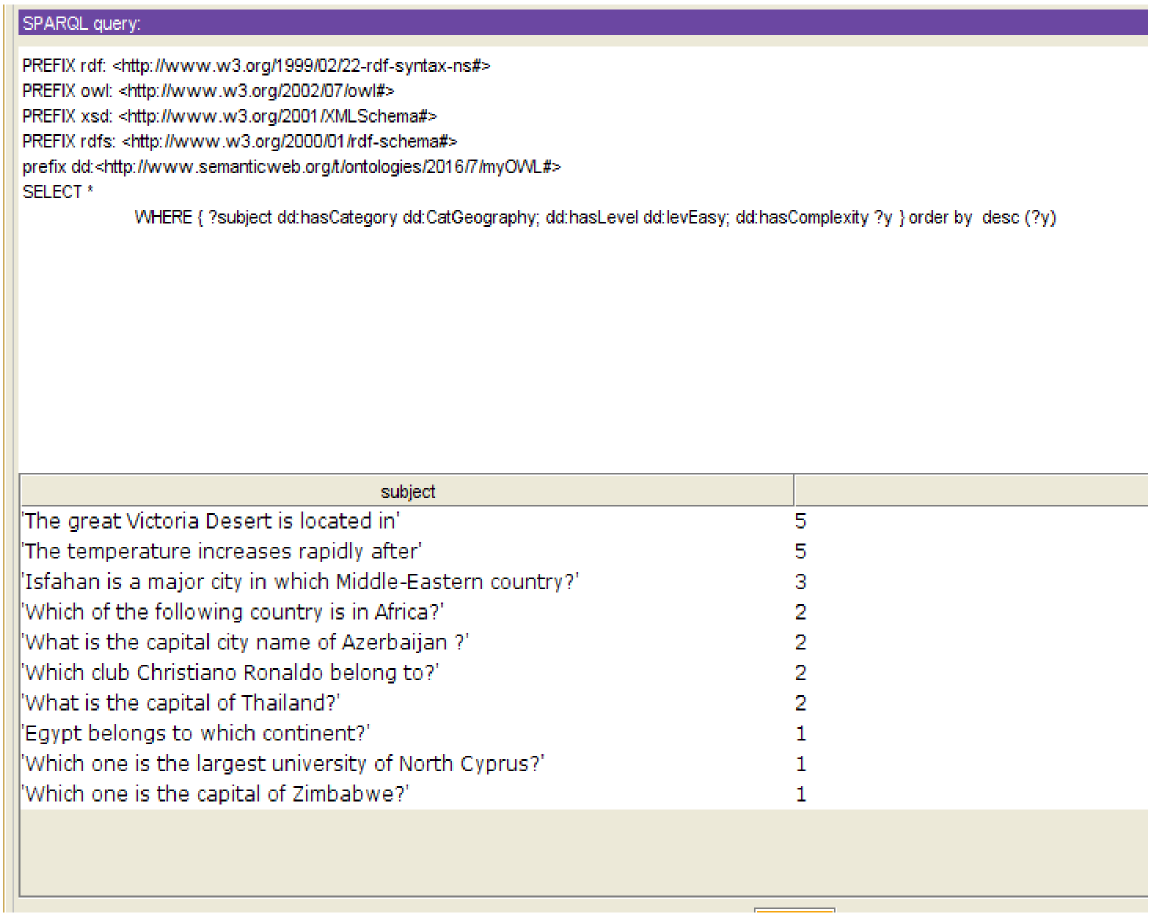


Figure 3.10 Order of Questions in Descending Order

3.4.4 High Score Category

The adaptive quiz system also keep track of user's highest score category i.e the category in which a user has the maximum score among the three categories. If, for example, a user have 50% in Physics category, 65% in Geography, and 80% in Chemistry category, then the highest score category for that user will be Chemistry. When the user login to the system next time, he/she will find the highest score category last in the drop down list because we want them to concentrate on the categories where they have low percentage.

The Jena rule for the highest score category for the Chemistry is shown in Figure 3.10.

```
rule6:(?x http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#User)

(?x http://www.semanticweb.org/t/ontologies/2016/7/myOWL#PhysicsPercentage  ?score1
)

 (?x http://www.semanticweb.org/t/ontologies/2016/7/myOWL#ChemistryPercentage
?score2 )

 (?x http://www.semanticweb.org/t/ontologies/2016/7/myOWL#GeographyPercentage
?score3)

 greaterThan(?score1,?score2), greaterThan(?score1,?score3) -> (?x
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#HighScoreCategory
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#CatChemistry)
```

Figure 3.11 Jena rule for the Highest Score Category for Chemistry

3.4.5 Rdfs Comment

If a user chooses wrong option while solving the quiz, we will assume that he/she does not have enough knowledge about that question and thus we provide two to three line explanation about it. The idea is allowing users to learn while solving the quiz game. This explanation is taken from DBpedia, called rdfs:comment property which is usually a short description about a particular Wikipedia/DBpedia resource. For instance, if a correct answer of a question is Centripetal force and user select another choice, the short explanation about Centripetal force will be displayed to him/her.

3.4.6 Sub/Super Class Relationship

We also have the rdfs sub-class/super-class inference in which if x is a type of class B and B is subclass of C, then x is the also the type of C. For instance, in our case, if a student1 is a type of PhysicsDiligent and PhysicsDiligent is a sub class of DiligentUser then student1 is a type of DiligentUser class as illustrated in Figure 3.11. Similarly, we have Jena rules for novice users.

```
[rule3:(?x  http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#PhysicsDiligent ) + (
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#PhysicsDiligent
http://www.w3.org/2000/01/rdf-schema#subClassOf
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#DiligentUser ) +

 →( ?x  http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#DiligentUser )
```

Figure 3.12 Sub-class Inference Jena Rule for Diligent User

3.5 SPARQL

SPARQL Protocol and RDF Query Language (SPARQL) is used to query RDF graph which comprise of triples and extract data in the form of subjects, predicate, and objects from RDF graph. We have used several SPARQL queries to get the required results such as loading the questions, answers and their choices from owl file, getting the rdfs:comment as explanation, and show the high score of a particular category and the student name who got the high score.

3.5.1 Query to Load the Quiz

This is the most important SPARQL query which loads the whole quiz game to the user when he/she login to the system. The query includes questions, answer, choices, level, category, and points of questions. The query is shown in Figure 3.12. We have shown here the query for adaptive system where we have used questions complexity also for questions re-ordering (i.e ORDER BY desc(?complexity)). This query will execute when the users of the quiz are Diligent users and will display the most difficult questions first and least difficult afterwards.

```
SELECT  *
         WHERE {
        ?Qs mo:Question ?QsDesc.
         ?QS mo:hasAnswers ?AnsQ.
         ?AnsQ mo:hasQuestion ?Qs.
         ?AnsQ mo:ChoiceOne ?ANSONE.
         ?AnsQ mo:ChoiceTwo ?ANSWTWO.
         ?AnsQ mo:ChoiceThree ?ANSWTHREE.
         ?AnsQ mo:CorrChoice ?ANSCORR.
        ?Qs mo:hasCategory ?Cat.
         ?Cat mo:category ?CatName.
         ?Qs mo:hasLevel ?lev.
         ?lev mo:level ?LevName.
        ?Qs mo:hasScore ?point.
        ?point mo:score ?Score.
        ?Qs mo:hasComplexity ?complexity.
         FILTER ( ?CatName ='"+ctg+"' ).
         FILTER ( ?LevName ='"+lvl+"' ).
         } ORDER BY desc (?complexity)
```

Figure 3.13 SPARQL Query to Load the Quiz Game

hasQuestion, hasAnswers, hasCategory, hasScore, and hasLevel are the object properties of the query which shows each question has an answer, a category (Physics etc), a level(Easy level or Expert level), and a score(Easy level questions have 2 points, expert level have 4 points). ChoiceOne, ChoiceTwo, ChoiceThree are three choices of a question and CorrChoice is answer (correct choice) of a question.

We have used Filter keyword of SPARQL to filter the user's choice with the category and level he/she has selected in the quiz game. Finally, we have order the questions in descending order for users that belongs to NoviceUser class.

Similarly another query is used where the questions are displayed in ascending order for novice users. For example, easy questions first and difficult after (i.e ORDER BY asc(?complexity)).

3.5.2 Query for rdfs:comment

When user selects wrong answer in the quiz game, the system will briefly explain the correct answer and the explanation comes from rdfs:comment. We have used SPARQL query to get this detail from the knowledge base. The query is shown in listing 3.13. 's' in the query is our Java variable that holds a URI of a the current questions.

```
 SELECT  *
 WHERE { ?question rdfs:label ?label  .
?question  rdfs:comment ?comment .
 "FILTER regex  (?label , '"+s+"' )
```

Figure 3.14 SPARQL Query for Explanation of Answer

```
SELECT  ?student ?highScore
WHERE {
{ SELECT  (max(?score) as ?highScore)
 { ?student dd:PhysicsPercentage ?score }  }
```

```
?student dd:PhysicsPercentage ?highScore }
```

Figure 3.15 SPARQL query for Highest Score in a Category

3.5.3 Query for Highest Score

In our adaptive quiz system, when a user login and selects a category for the quiz, say Physics, then the system will display the highest score of Physics category ever achieved by any user. For example, when a user select Physics category, a message will be displayed, "Physics quiz game have the highest score 80% scored by user John, can you pass through it". Its main purpose is to motivate users to get the maximum score in the quiz.

We have used three SPARQL queries for this functionality, one for each category. The SPARQL query is shown in Figure 3.14. We have also similar queries for Chemistry and Geography categories.

3.6 System User Interface

We have developed two systems, one without any adaptation which is called baseline system and the other with the described adaptations that is called adaptive system. We will compare the performance of these two systems in order to test if there is any benefit to users coming from the adaptation of the quiz game. The user interface of our baseline and adaptive system have been developed in Java Swing in NetBeans Integrated Development Environment (IDE).

3.6.1 Baseline User Interface

The user interface of the baseline system is very simple and self-explanatory. Users first register his self by entering the name, and optionally a registration (student) number. The users then select the type of quiz i.e which category of questions they would like to see first and also the level of the quiz which can either be easy or expert level. The Figure 3.15 shows the login page and when user selects Physics category and Expert level. A next button will allow users to move to next questions if answer is correct. If the answer is not correct, explanation of the answer will be displayed to him/her. Figure 3.16 shows when user incorrectly answer a question and a detail about the answer is shown to him/her. When the user finishes the quiz, a summary is shown to him including the graphical representation of his high score and favorite category. The Figure 3.17 shows the screen of quiz summary for a user.

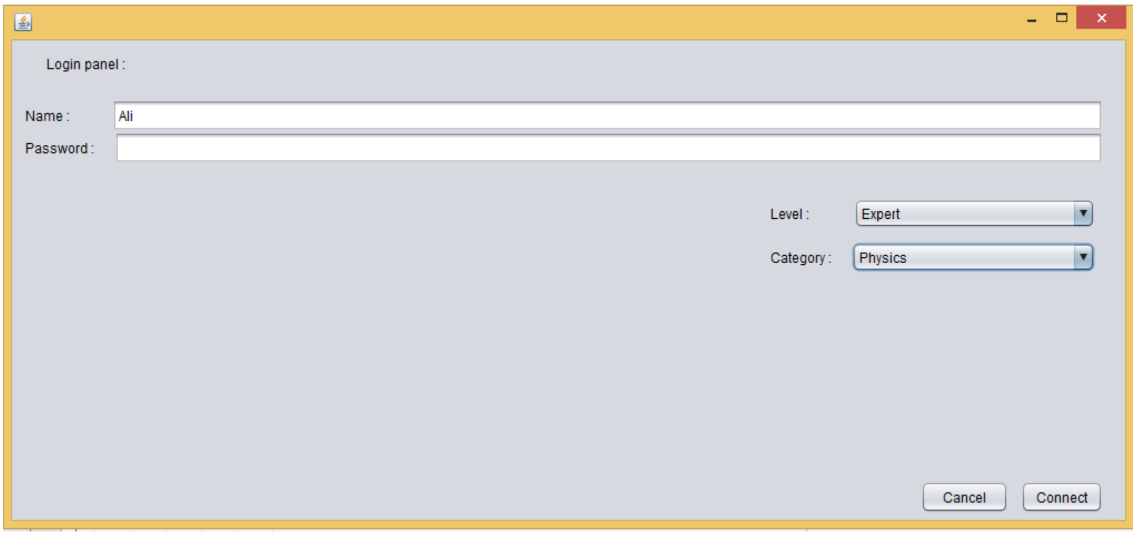


Figure 3.16 Baseline System Login Page

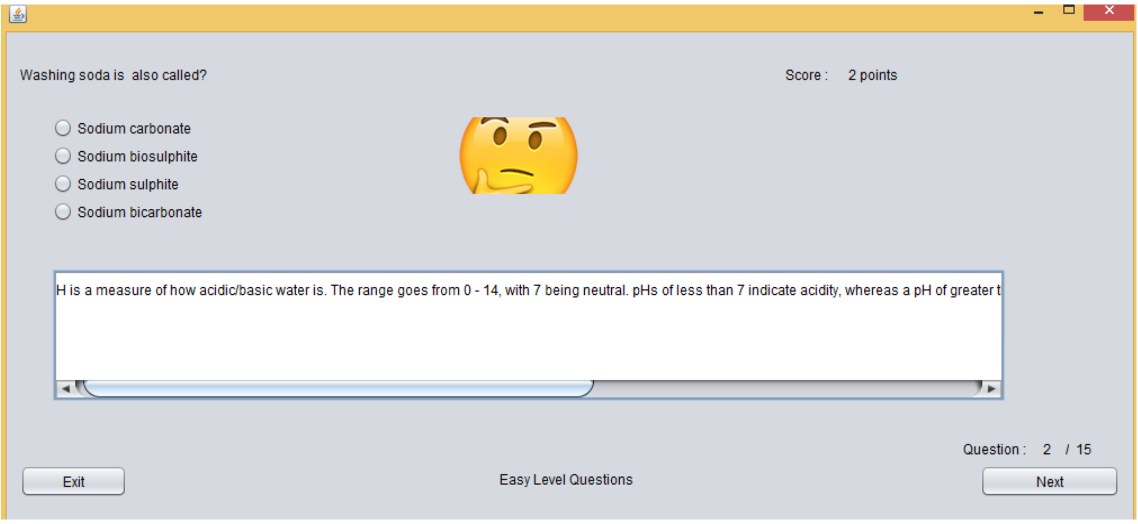


Figure 3.17 Explanation of the Answer from DBpedia

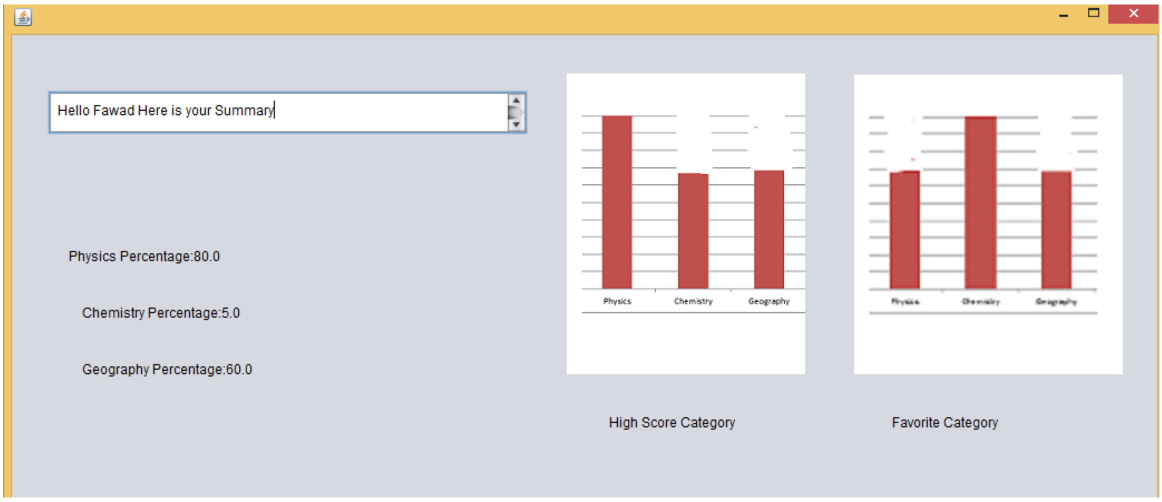


Figure 3.18 Summary of User Performance

3.6.2 Adaptive User Interface

The adaptive user interface adapts the quiz game to different users. When the user selects the category of quiz game i.e Physics, a motivational message is displayed to him/her about the highest score of that category and the user name who have achieved this high score shown in Figure 3.18.

. The category which a particular user has selected more time that will be considered as the preferred category of the user. When user login to the system, his/her preferred category will be appear first in the drop down list. Similarly, the category in which a user got maximum score will be recorded as user’s high score category and will appear last in the list. The default sequence of the category list when a user login for first time is Physics, Chemistry, and Geography and then it will appear according to user preference. Also if a user is diligent user (From Diligent class), expert level questions will appear first in the drop down box and easy level will if users are from Average or Novice class. For example, a user Zulfikar has Geography as preferred category which appears first in the list and Physics as high score category, appeared last in the category shown in Figure 3.19. Another user Saidi has Chemistry preferred category and Geography high score category shown in Figure 3.20.

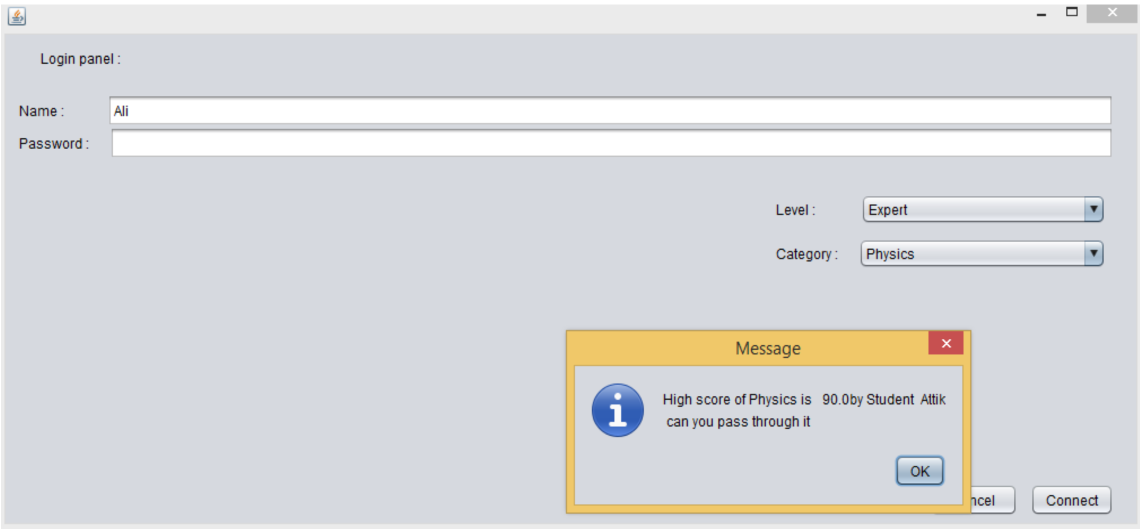


Figure 3.19 Motivational Message about the High Scorer

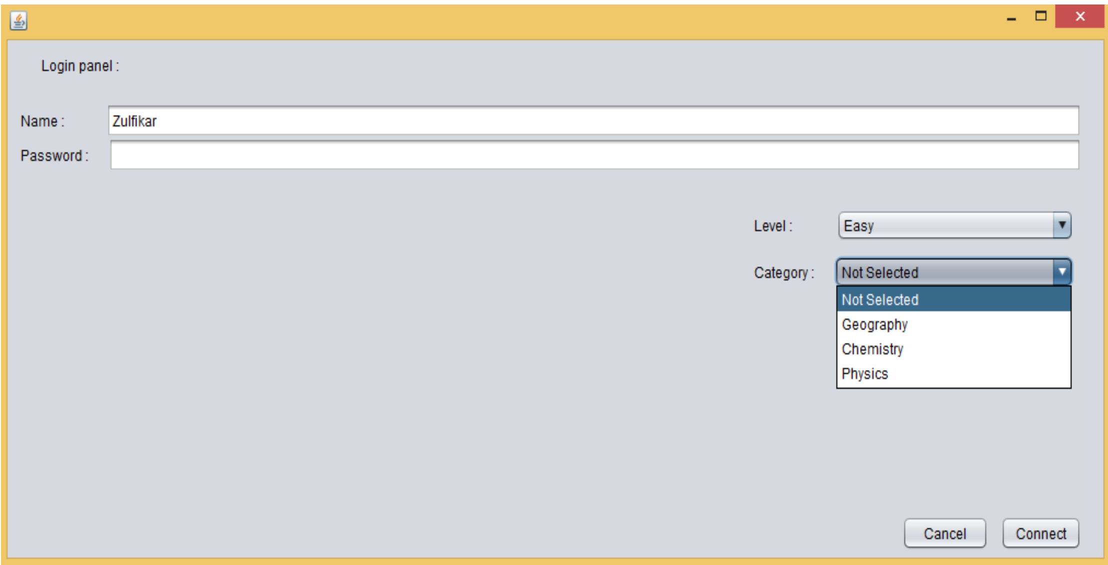


Figure 3.20 User1 Adaptive User Interface Screen

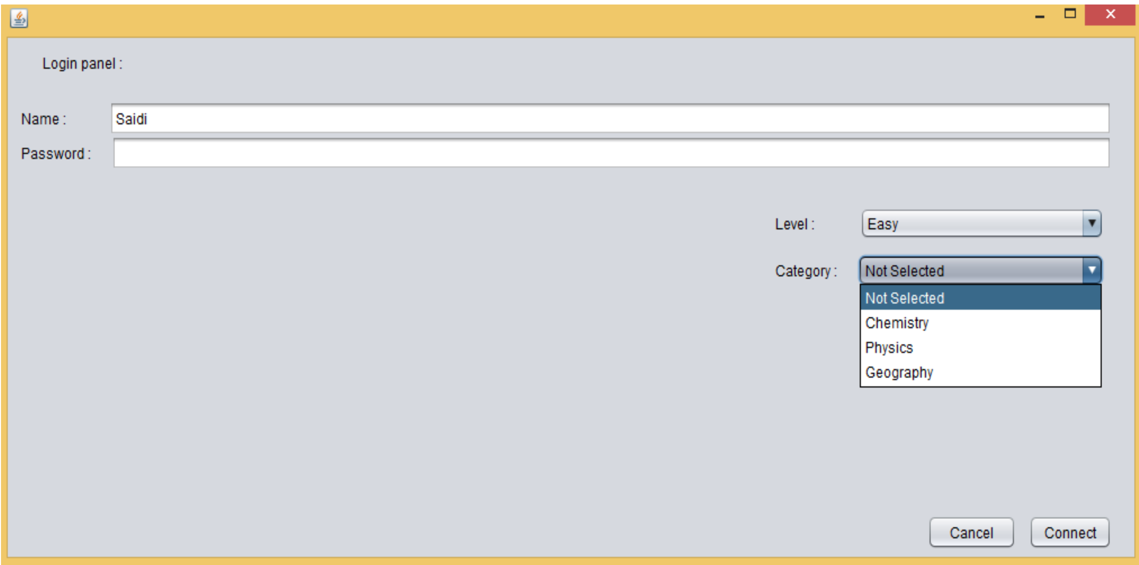


Figure 3.21 User 2 Adaptive User Interface Screen

We also re-order questions for different class of users based on their class and capabilities. For instance, if a user is a novice user we will show him the easiest questions first and difficult ones in the end. Similarly, if a user is diligent user, then we will show the more complex questions first and the least complex in the end.

Zulfikar and Saidi users mentioned above are novice users so easy level questions will be displayed first to them. Another user, Macmillan is a diligent user so he will find most difficult questions first, with the degree of complexity decreasing till the last question.

In Figure 3.9 from Protégé editor, we have seen that the more complex questions are "The great Victoria desert is located in?" and "The temperature increases rapidly after?" from geography category with the highest degree of complexity of 5. Diligent users, Macmillan in this case, will find these two questions first when he login to geography quiz and questions with low degree of complexity afterwards. The Figure 3.12 and 3.13 shows the first two questions with highest complexity i.e 5.

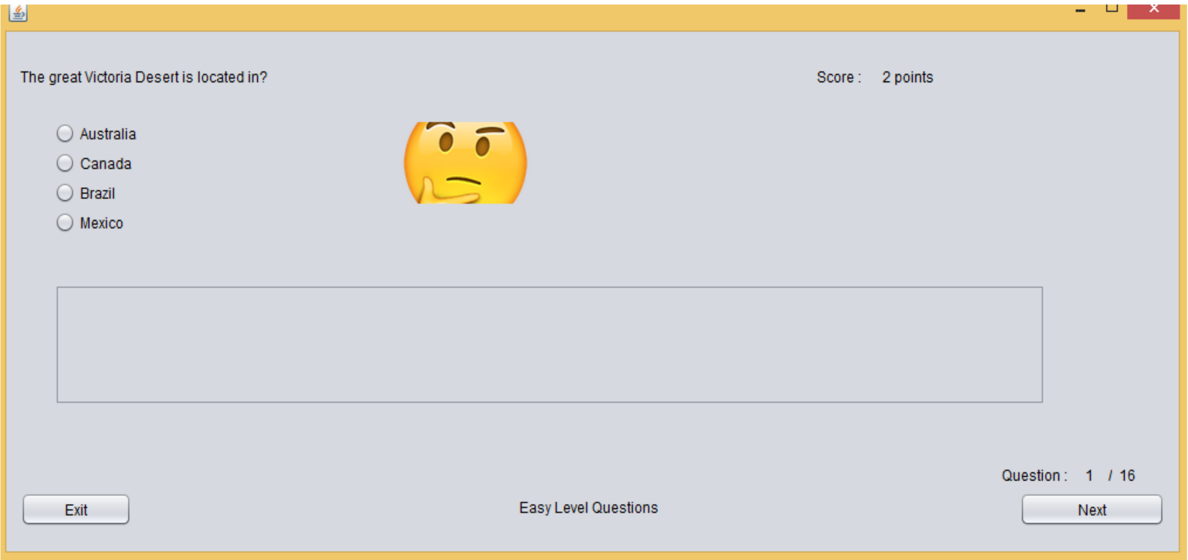


Figure 3.22 Question with Complexity of 5 Appears First in the Quiz

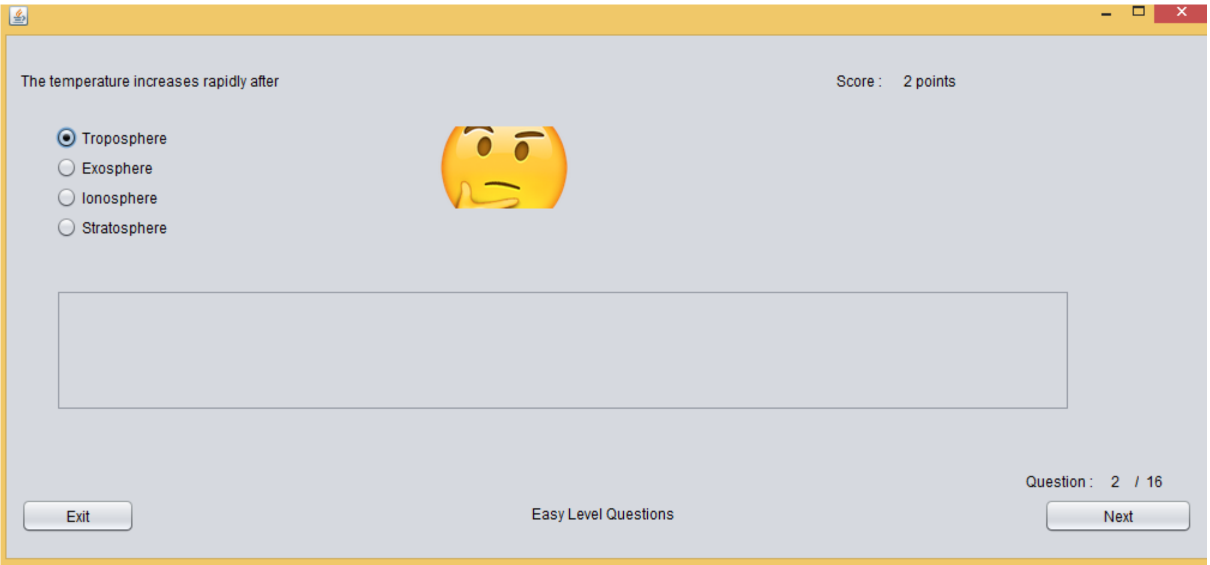


Figure 3.23 Another Question with Complexity of 5 Appears Second in the Quiz

CHAPTER 4
EVALUATIONS

4.1 User Studies

We performed a user study in which different users used both the baseline and adaptive quiz systems and have recorded their views and performance.

In order to evaluate the performance of our adaptive quiz system, we have compared it to a baseline quiz system in a user study. A particular user have to play the quiz both with baseline and adaptive quiz systems, and attempt all questions of the three categories (Physics, Chemistry, and Geography). In addition, users have two rounds for each system so a user tried both the baseline and adaptive system twice. For tests to be un-biased and to remove the learning effect, we swaped both systems for different users. For instance, if a user1 tried the baseline system first and then the adaptive system, then user2 must try first the adaptive and then the baseline system.

The adaptive system contains features like explanation of answers if the user chooses an incorrect answer, a message about the highest score and scorer of the category he/she has selected, the category list according to the user's preferences and re-ordering of questions according to user's class and capabilities. We also record the user's scores for the previous tries for the adaptive quiz game. If a user try ten times for a particular category i.e Physics, then ten instances of Physics scores will be stored in an rdf file (backup.owl).

The baseline system just contain the explanation of answers if he/she did not know about the answer. The users perform the quiz and their score and percentage are recorded. The users are not displayed with any type of adaptivity in the baseline system when solving the quiz questions. For example, in Figure 3.19 and Figure 3.20, we have shown the shuffling of category list according to user's preferred category and user's high score category for the adaptive system but in the baseline system users will not see any adaptivity. For instance, a user Yasir has tried the baseline system quiz and he mostly selected the category Chemistry but still he sees the default categorization of the list when login to the system as shown in Figure 4.1. Similarly, there are no details of user's class based on their quiz performance so there will be no re-ordering of questions for different users and thus all users will have questions with random complexity regardless of their performance in the game.

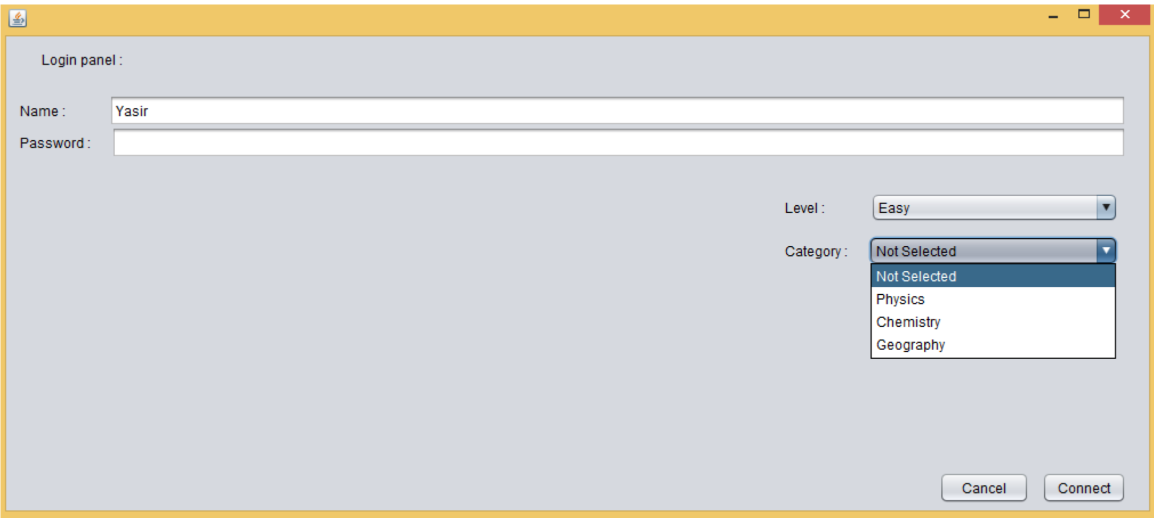


Figure 4.1 No Adaptation in Category List

In the study, both baseline and adaptive systems have different set of questions i-e fifteen questions per category in the baseline and fifteen questions per each category in the adaptive system. Thus in total, there are ninety questions in the baseline system and a different set of ninety questions in the adaptive system, in order to remove the learning effect. All the questions are taken from DBpedia and are stored in two separate rdf files, baseline.owl and adaptive.owl.

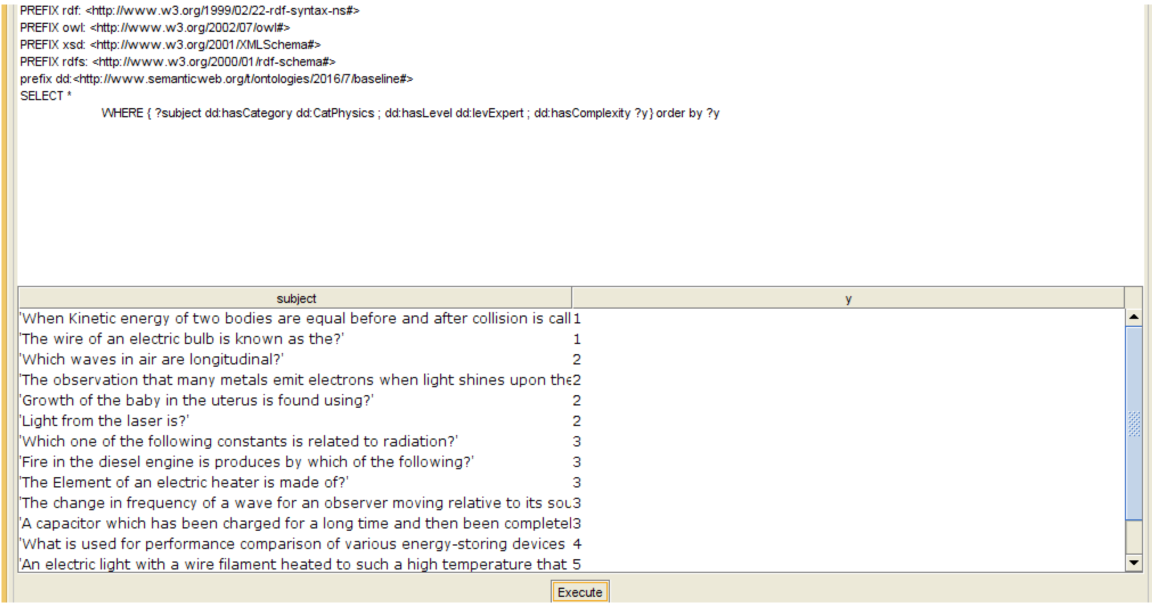


Figure 4.2 Order of Questions in Protégé Editor

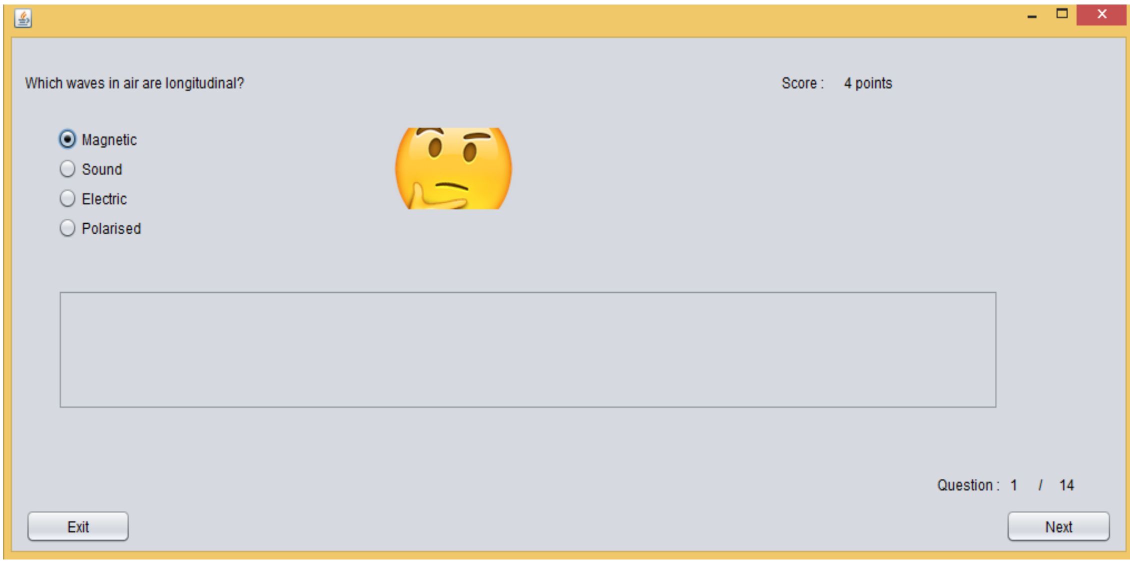


Figure 4.3 Questions Appear Random to Users

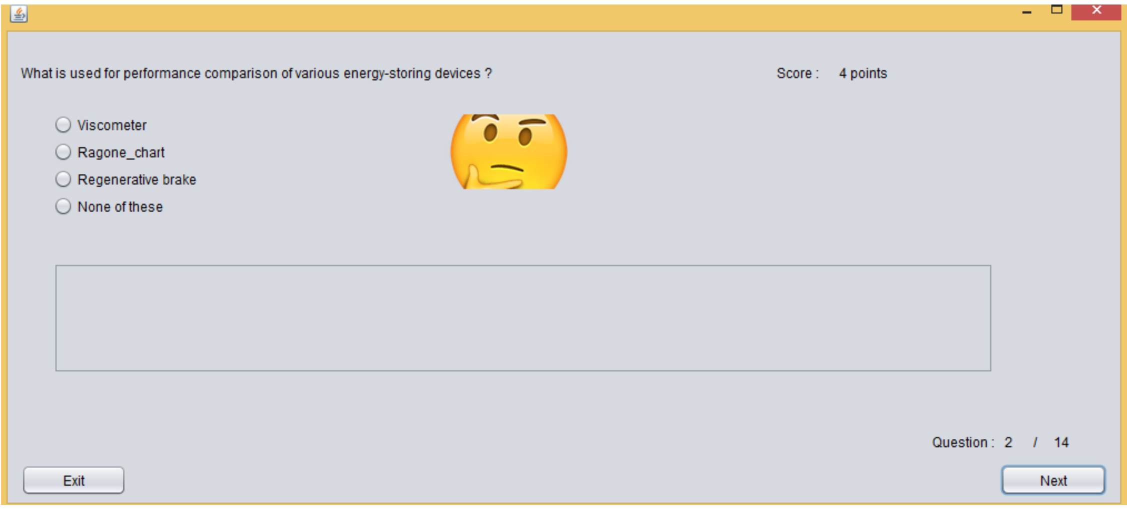


Figure 4.4 Questions Appear Random to Users

4.2 Task

To check the performance of the adaptive quiz system, we asked different users to perform some tasks. First a background questionnaire was given to users to learn their background information. The questionnaire contains basic information about name, age, department name, and country name. Users of different countries, departments, and qualifications were asked to test both systems. In this study we have used twenty participants, of which 60% are undergraduate students, 30% are master level students and 10% are Phd students.

First a user was asked to try the baseline (or adaptive) system first and solve the questions of all three categories. Their performance in the quiz like percentage and time (minutes and seconds) was recorded. The same user then tried a second round with the same system and we recorded their performances as well. When the participant completed two rounds with the baseline system, the user then tried the adaptive system (or baseline system) and perform two rounds of the adaptive quiz system. In adaptive system, apart from user percentage and time, we have recorded user's preferred category, the highest score category and the class of user based on his/her performance in the quiz. When the user tried, we first showed the adaptive system (swapped the system to remove the learning effect). Thus, the user will first appear to the adaptive system, perform two rounds of all three categories and then appear to the baseline system for another two rounds for all categories.

After a user performed the quiz game (either baseline or adaptive systems), post questionnaires were given to him/her to get feedback about the system. We asked users in the post questionnaire to share their views about the systems like what are their opinions about the system, which component of the quiz system they like or dislike the most and their views about which area/component is needed to improve. We performed few tried with two users to see if there are any usability issues that needs to be fixed before the user study. The following were suggestions: When the user solve the questions incorrectly, the explanations are displayed to them in a text area and that explanation was there till the user choose a wrong answer of another question (Until the user solve the questions correctly, the explanation of previously incorrect answer were displayed). This was pointed out by a user and we overcomed it as follows: When the user chooses the right option of the next question, the previous answer explanation becomes hidden. Similarly, a user pointed out that "in the middle of quiz, I do not know which level of questions (Easy/Expert) I am solving". We tried to overcome this, if a user selects Easy level questions, a label will show "Easy Level Questions" till the end of questions and similarly for expert level questions. Some of suggestions could not be fixed at this point because of time limitations. For example, some users said category lists should be extended and it should include Programming and English language questions, etc. Similarly, one suggestion was the quiz system should be designed such that a teacher is able to create questions and which will change from time to time. Inclusion of numerical questions in Physics category were suggested by many users.

4.3 Comparison of Adaptive system and Baseline System

After the user completed two rounds with both the adaptive and the baseline system, we have recorded their scores/percentage and the time they have taken to accomplish the quiz game. We have then compared the users progress in both the adaptive and baseline systems. The comparison measures user progresses (if progresses or not) in second round for both the adaptive and baseline systems. The progress is compared in terms of score and time. We have found out that our adaptive system is more efficient than the baseline system in terms of both users scores and game completion times. A big percentage of users managed to achieve higher score in less time in adaptive system as compared to the baseline system.. Figure 4.7, Figure 4.8, and Figure 4.9 shows the time needed to perform individual tasks of physics, chemistry and geography categories respectively.

All users performed tasks similarly for the adaptive and baseline systems in the first round. However, in the second round, in the adaptive system almost all users achieved higher score in less time. The Figure 4.5 shows the overall time needed to perform adaptive and baseline quiz systems for all users and Figure 4.6 shows the overall average time needed to perform the tasks. Figure 4.7 and Figure 4.8 shows the overall scores of all users and the average user's score in both the systems, respectively.

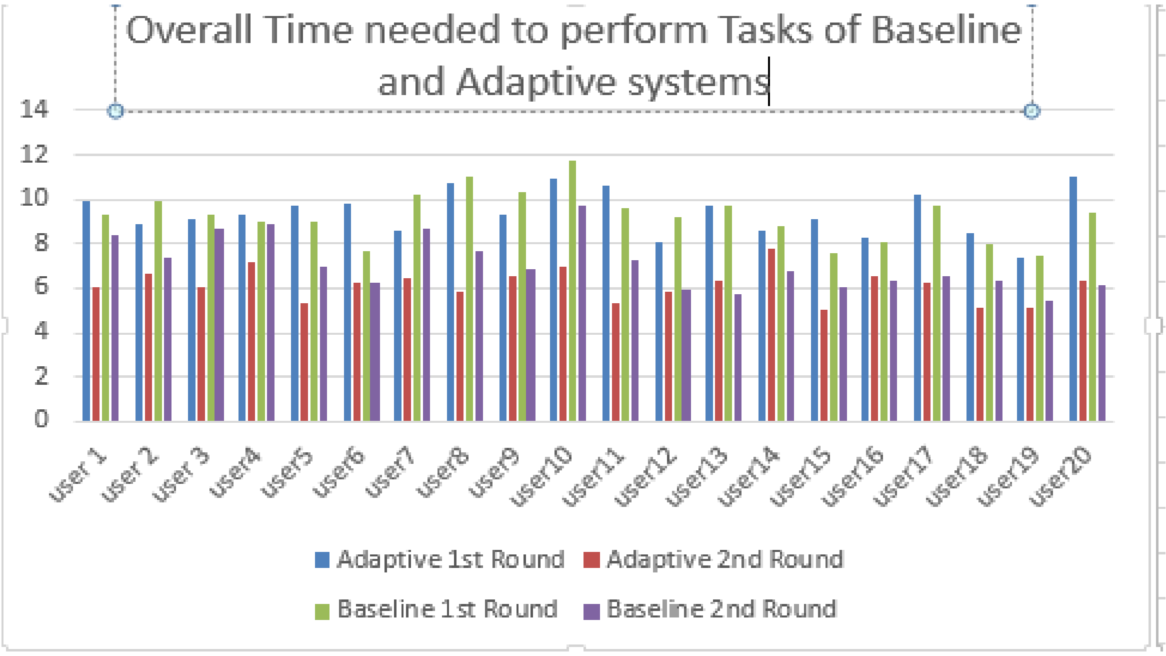


Figure 4.5 Overall Time Needed For Both Systems

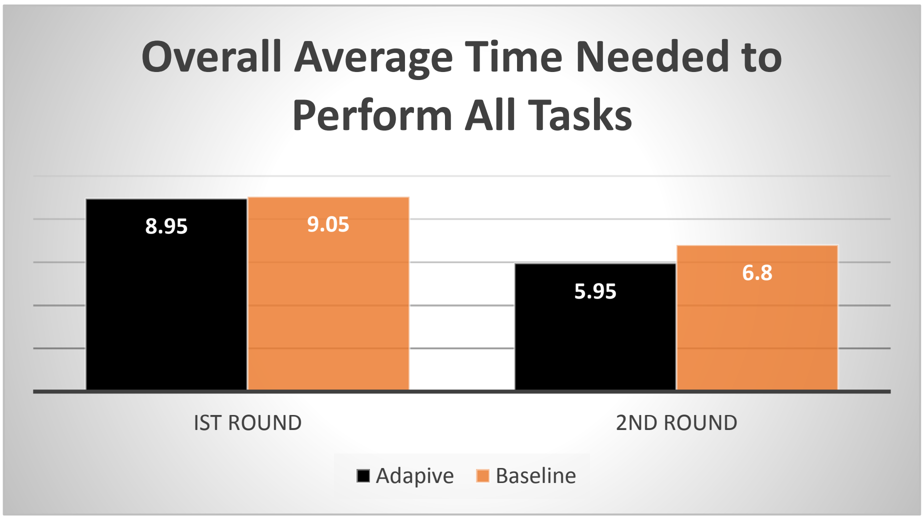


Figure 4.6 Average Time Needed For Both Systems

Both figures clearly depicts adaptive system is better than the baseline system and users took less time than baseline system to achieve the goal. For instance, the average time a user takes to complete the baseline and adaptive tasks in round one are 8.95 and 9.05 minutes respectively. While in second round of both the systems, a user completed the task (all the three categories of the quiz game) of baseline system in 6.80 minutes and adaptive system in 5.95 minutes. Similarly, the average score a user achieved is 142 and 140 for round first of baseline and adaptive systems respectively while in second round user got 185 score in baseline and 204 in adaptive sysem so it means the users perform quite better in the second round of adaptive system.

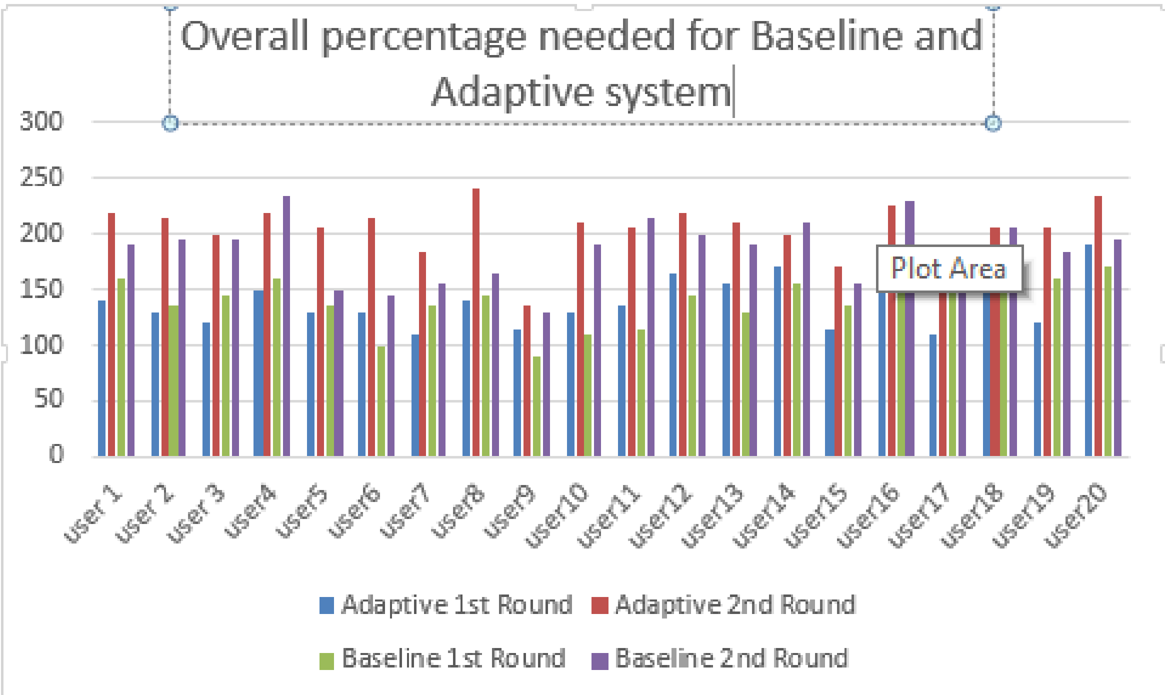


Figure 4.7 Overall User Score of Both Systems

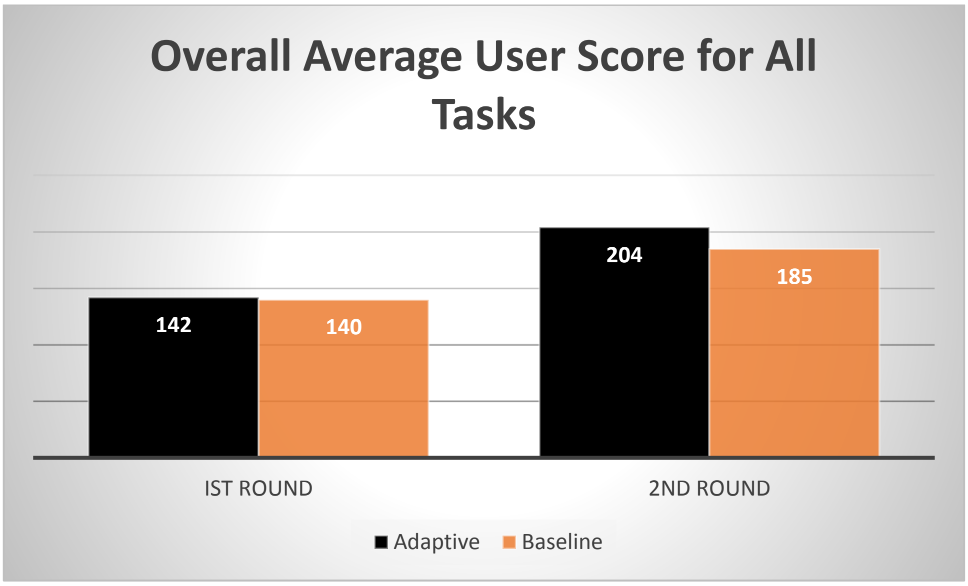


Figure 4.8 Overall Average User Score of Both Systems

For individual category of Physics, the average time taken by user to complete the task for baseline first round is 3.23 minutes and for adaptive first round is 3.21 where as for the second round, it takes 2.45 minutes for baseline system and 2.2 minutes for adaptive system to complete the quiz game as shown in figure 4.9. The average scores recorded by users in first round of Physics category in baseline and adaptive systems are 45.78 and 46.66 while in second round it is 63 and 70.41 respectively shown in figure 4.10.

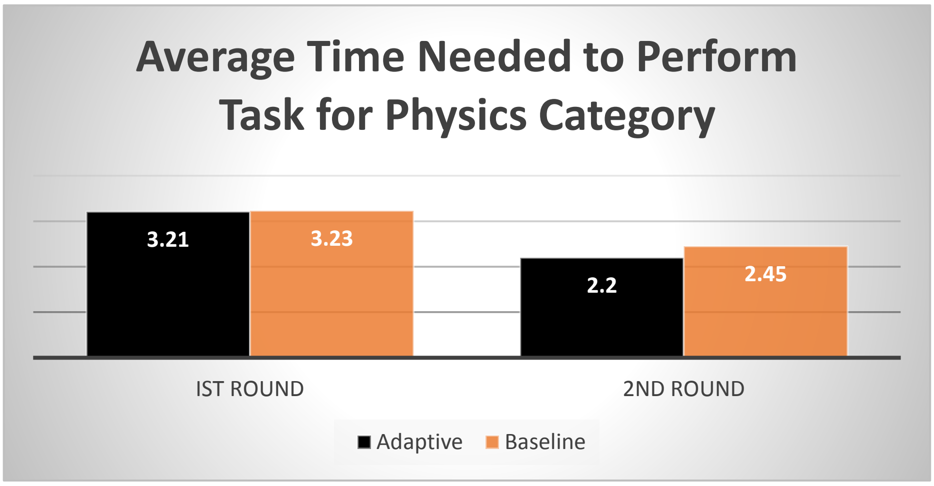


Figure 4.9 Average Time Needed For Completing Physics Task

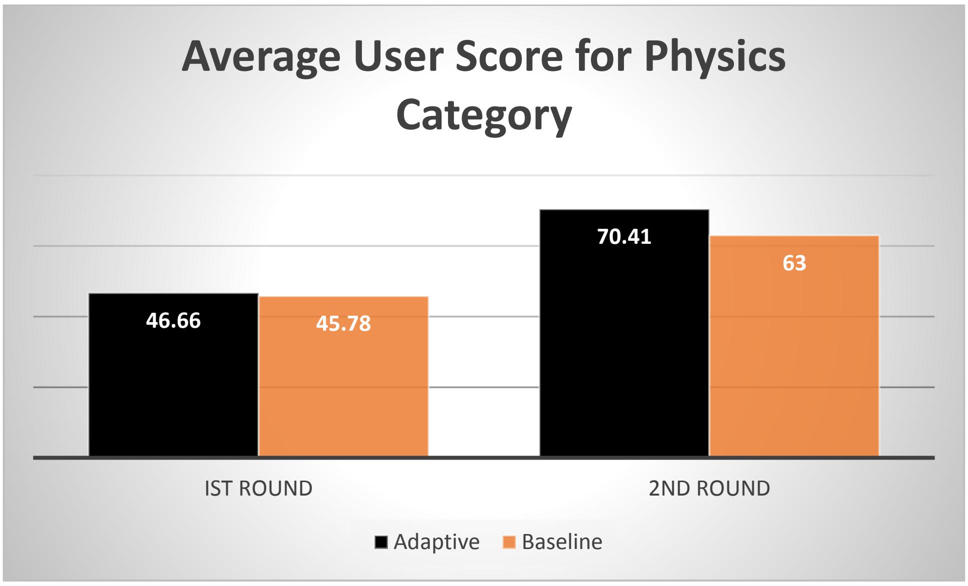


Figure 4.10 Users Score of Physics Task

For Chemistry category, the time taken by the users to complete the task is 3.75 minutes for baseline and 3.35 minutes for adaptive systems in the round first, while in second round the baseline task completion reduces to 3.0 minutes and adaptive system time reduces to 2.4 as shown in figure 4.11. The average Chemistry score in first round of baseline system is 35 and in adaptive system, it is about 41 while in second round, the score achieved by users are 52 and 60 respectively which shows users improved in the second round, particularly in the adaptive system. The figure 4.12 shows the scores users achieved in Chemistry category.

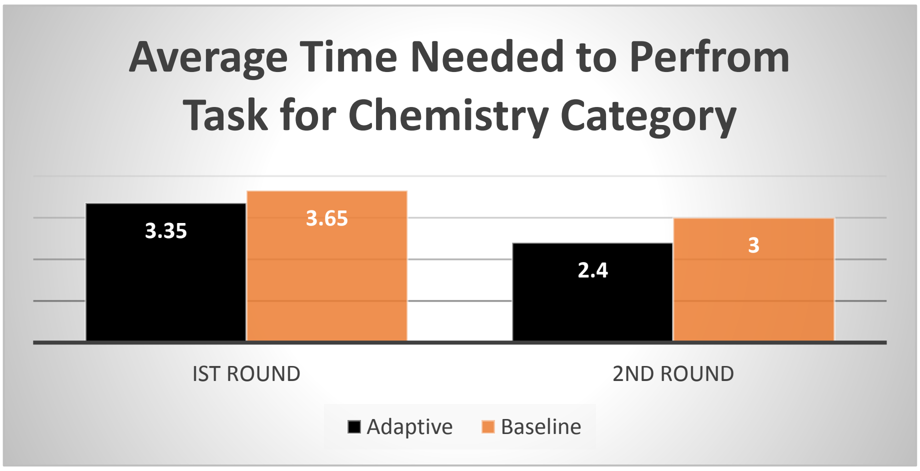


Figure 4.11 Average Time Needed For Completing Chemistry Task

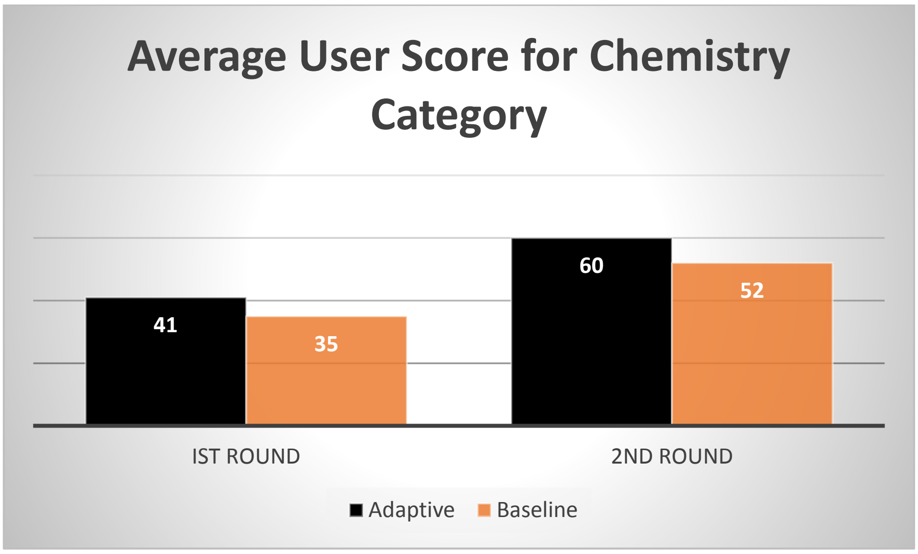


Figure 4.12 Users Score of Chemistry Task

Figure 4.13 shows the user's score in first round of baseline system is 51.75 and in adaptive system, the score is 49.16. The difference between the baseline and adaptive user's scores in round second, however, is very low compared to the other two categories as in the baseline users achieved 75score while in adaptive task, average users score is 78 which shows the difference is not big as shown in figure 4.14. In our view, the reason behind this is most of the users have already a very good knowledge of Geography questions and when users have already knowledge about the related areas, the advantages of adaptivity then becomes limited and that is why the difference between users baseline and adaptive scores in second round are lower than the other two categories. For geography category, Figure 4.14 shows the time needed to perform the task for round one and round two.

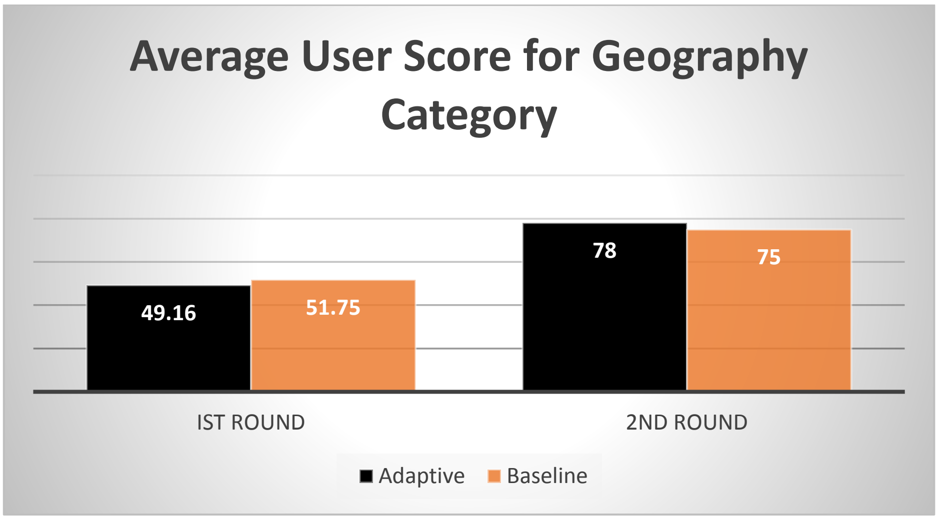


Figure 4.13 Average User Score for Geography Task

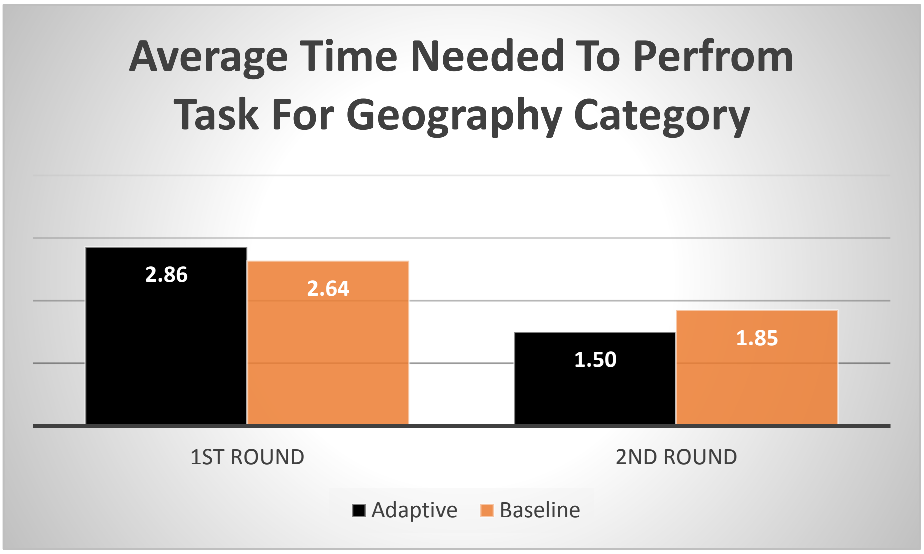
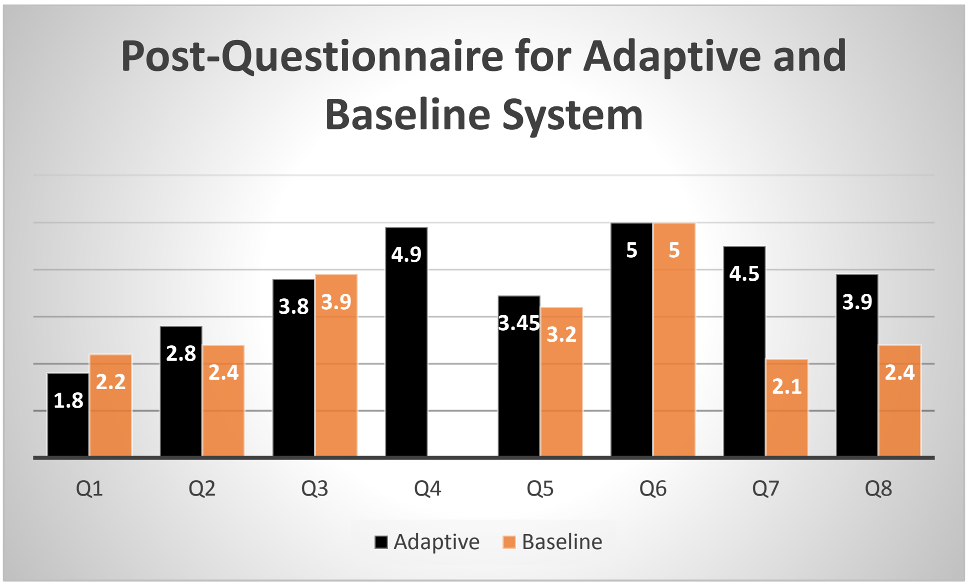


Figure 4.14 Average Time Needed for Geography Task

4.4. Post-Questionnaire

In this section, we summarize the results of post-questionnaires. In particular, we wanted to learn users' opinions about different features of the proposed adaptive quiz-based game. We also compare users' opinions about the baseline system with the adaptive system since we asked the same questions after completing tasks with each system. Users pointed out that the task was complex with an average of 1.80 in adaptive system as compared to 2.2 of the baseline system. They also said that they perform well on tasks with an average of 2.8 in the adaptive system compared to an average of 2.4 of the baseline system. According to users, they found it easy to navigate in the baseline system with an average of 3.9 compared to 3.8 in the adaptive system. In adaptive system, every time the user selects a category to start the quiz game, a message dialog box is displayed to him/her showing the highest score and scorer of the category. Users have to read the motivational message and click the OK button to disappear the dialog box so some of the users found it an extra step and this is why they point slightly less for adaptive system for navigation. Almost all the users, with the average of 4.9, pointed out that they were inspired and motivated by the person who had the highest score in a particular category and wished the highest scorer was he/she. Please note that the highest score was not shown in the baseline system and when displayed in the adaptive system, users found it motivating compared to the baseline system, which we consider 0. About the presentation, result structure and contents of the quiz to solve the task, the users were satisfied with the adaptive task at an average of 3.45 compared to 3.2 of the baseline. On both systems, users pointed out that they learned and improved their score with the help of explanations (e.g comments) with an average of 5 (all the users of both adaptive and baseline systems choose 5, which is a maximum value). 100% of users who tested the system have improved their scores significantly in the second round and this is due to the explanation given to them when users select wrong answer. Finally, participants thought that the adaptive task was more motivating with an average of 4.5 compared to the 2.1 for the baseline system task. Also they have found the adaptive task fun with the average of 3.9 compared to 2.4 of the baseline task.



Q1.The task was complex.

Q2.I did well on tasks.

Q3.Did you find easy to navigate within the quiz?

Q4.Were you inspired by the other student's top scores at the beginning of the quiz?

Q5.I am satisfied with the system performance, guidance and assistance.

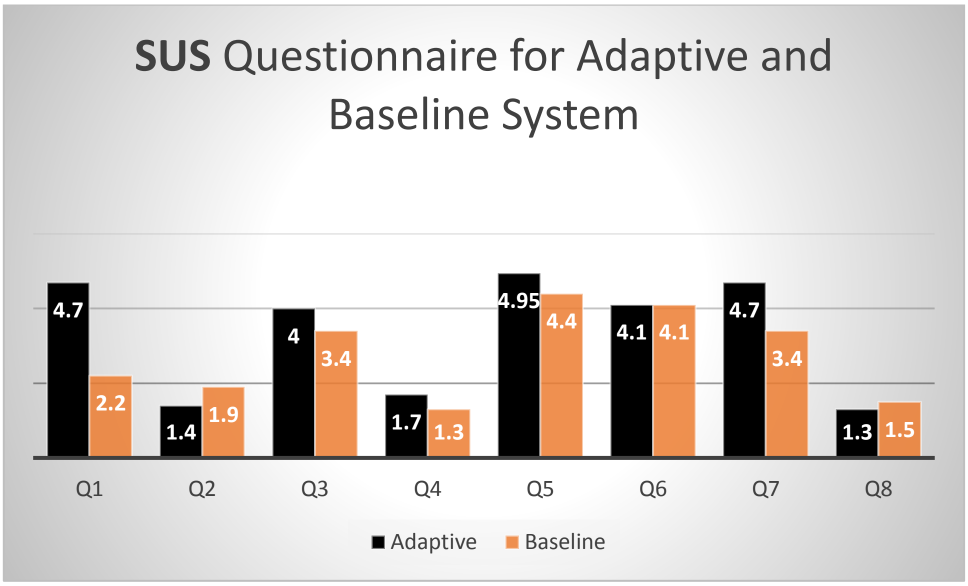
Q6.Did the explanations (e.g comments) help you to learn the incorrect answers and improve your score?

Q7.I found the interaction with the system motivating.

Q8.I found the interaction with the system fun

Figure 4.15 Post-Questionnaire for Adaptive and Baseline System (average of all tasks. 1=strongly disagree, 2=disagree, 3=not sure, 4=agree, 5= strongly agree)

4.5. Standard Usability Scale (SUS) Questionnaire



Q1. I think that I would like to use the system frequently. Q2. I found the system unnecessarily complex. Q3. I thought the system was easy to use. Q4. I think that I would need assistance to be able to use the system. Q5. I found the various functions in the system were well integrated. Q6. I would imagine that most people would learn to use the system very quickly. Q7. I felt very confident using the system. Q8. I needed to learn a lot of things before I could get going with the system.

Figure 4.16 Standard usability scale (SUS) Questionnaire for Adaptive and Baseline System (average of all tasks. 1=strongly disagree, 2=disagree, 3=not sure, 4=agree, 5= strongly agree)

We have used the SUS to determine the overall usability of both the adaptive and baseline systems. With the average of 4.7, users pointed out out that they will like to use the adaptive system frequently compared to the 2.2 for baseline system. The users found the adaptive system unnecessarily complex with the average of 1.4 as compared to the baseline which is 1.9. Similarly, with the average of 4 users have found the adaptive system more easy compared to the average of 3.4 for baseline system. For adaptive system users pointed out that they need assistance to use the system at the average of 1.7 compared to the baseline system which is 1.3. That is because most of the users found the adaptivity of the system slightly strange like the shuffling of questions order, the categories and the level of the questions in the user interface compared to the baseline system which is always static and fixed. Though all the users becomes used to and like the adaptivity later, most of them needed initial guidance. According to the users, the functions in the adaptive system are well integrated with the average of 4.95 compared to the 4.4 for the baseline system. With using the adaptive system, users think they felt confidence at the average of 4.7 compared to the using of baseline system which is at the average of 3.4.

CHAPTER 5
CONCLUSIONS AND FUTURE WORK

In this work, we have developed a game based e-learning system which allows users of different categories to learn with motivation and fun. The users will solve the quiz game and will learn the learning materials taken from the areas of Physics, Chemistry, and Geography, extracted from DBpedia. When a particular user have no knowledge about a question and selects a wrong answer in the quiz, a short explanation of the question will be displayed to him/her. In addition, we have used Semantic Web rules and SPARQL to provide some adaptivity in our e-learning system and the system will adapt itself according to the user's capabilities and preferences. At the end of the quiz game, a detailed user evaluation has been carried out in which most of the users have pointed out that the adaptive system is better than the baseline system in terms of users performance in the game.

In future, we will use fuzzy rules and fuzzy inference system, apart from Semantic Web rules to implement our e-learning system. Fuzzy logic can be used when we have vague information and we can use it in the users categorization. For instance, if we set threshold of 80% marks for user to be in the category of Diligent User, so if he scores 80% or more he will be Diligent otherwise in the category of Average User. But what if some User gets 79%, so putting him in the Average User category will not be fair so for that and some other problems we need to use Fuzzy rules so that we can get results in membership function like a user is 0.8% DiligentUser, 0.2% AverageUser, and 0.0% NoviceUser.

Moreover, we can do our user evaluation also in memeber functions in which we can say a user is agreed to a particular task 0.3% and disgree 0.7% so that we can get more accurate and precise results when we analyze it using graphs.

We will use Fuzzy Control Language and Java's jfuzzy library to implement the system.

REFERENCES

Abdullah, F. A. (2011). Evaluating the Effectiveness of the E-learning Experience in Some Universities in Saudi Arabia from Male Students' Perceptions

Al-Yahya, M., George, R., & Alfaries, A. (2015). Ontologies in e-learning: review of the literature. *International Journal of Software Engineering and Its Applications*, *9*(2), 67-84.

Baylari, A., & Montazer, G. A. (2009). Design a personalized e-learning system based on item response theory and artificial neural network approach. *Expert Systems with Applications, 36(4),8013-8021.*

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., & Hellmann, S. (2009). DBpedia-A crystallization point for the Web of Data. *Web Semantics: science, services and agents on the world wide web*, *7*(3), 154-165.

Bratsas, C., Chrysou, D. E., Eftychiadou, A., Kontokostas, D., Bamidis, P., & Antoniou, I. (2012, April). Semantic web game based learning: An i18n approach with greek dbpedia. In *2nd International Workshop on Learning and Education with the Web of Data* (LiLe-2012 at WWW-2012), Lyon, France.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., & Rosati, R. (2013). Data complexity of query answering in description logics. *Artificial Intelligence*, *195*, 335-360.

Carmona, C., Castillo, G., & Millán, E. (2008, July). Designing a dynamic bayesian network for modeling students' learning styles. In *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on* (pp. 346-350). IEEE.

Chen, C. M. (2008). Intelligent web-based learning system with personalized learning path guidance. *Computers & Education*, *51*(2), 787-814.

Eric Prud'hommeaux,W3C, Andy Seaborne. (2008) SPARQL Query Language for RDF. https://www.w3.org/TR/rdf-sparql-query/

Goebel, R., Zilles, S., Ringlstetter, C., Dengel, A., & Grimnes, G. A. (2008). What is the role of the semantic layer cake for guiding the use of knowledge representation and machine learning in the development of the semantic web?. In *AAAI Spring Symposium: Symbiotic Relationships between Semantic Web and Knowledge Engineering* (pp. 45-50).

Guo, W., & Chen, D. (2006, June). Semantic Approach for e-learning System. In *Computer and Computational Sciences, 2006. IMSCCS'06. First International Multi-Symposiums on* (Vol. 2, pp. 442-446). IEEE.

Henze, N., Dolog, P., & Nejdl, W. (2004). Reasoning and ontologies for personalized e-learning in the Semantic Web. *Educational Technology & Society*, *7*(4), 82-97.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., ... & Bizer, C. (2015). DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, *6*(2), 167-195.

Maha Al-Yahya, et al, "Ontologies in E-Learning: Review of the Literature", International *Journal of Software Engineering and Its Applications*, Vol. 9, No. 2 (2015), pp. 67-84.

Mayer, R. E. (2002). Multimedia learning. *Psychology of learning and motivation*, *41*, 85-139.

M. Haghshenas, "Some Properties of Semantic Web in e-Learning", *International Journal of Innovation, Management and Technology,* Vol. 4, No. 2, April 2013.

Natalya F. Noy and Deborah L. McGuinness. ‘Ontology development 101: A guide to creating your first ontology’. In: (2000). URL: http://protege.stanford.edu/publications/ ontology_development /ontology101-noy-mcguinness.html.

OECD (2005). E-learning in tertiary education [Online]. Available at http://www.cumex.org. (Accessed 27 /02/ 2014).

Sah, M., & Wade, V. (2011, June). Automatic mining of cognitive metadata using fuzzy inference. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia* (pp. 37-46). ACM.

Sangineto, E. (2008). An Adaptive e-Learning platform for personalized course generation. *Architecture Solutions for E-Learning Systems*, 262-281.

Stumme, G., Hotho, A., & Berendt, B. (2006). Semantic web mining: State of the art and future directions. *Web Semantics: Science, Services and Agents on the World Wide Web*, *4*(2), 124-143.

Varlan, S. E. (2010). Advantages of semantic web technologies in the knowledge based society. *Scientific Annals of the'Alexandru Ioan Cuza'University of Iasi: Economic Sciences Series*.

APPENDICES

APPENDIX 1
SPARQL QUERIES

All of our SPARQL queries are given below.

Query to Load the Quiz

```
SELECT  *

          WHERE {

         ?Qs mo:Question ?QsDesc.

          ?QS mo:hasAnswers ?AnsQ.

          ?AnsQ mo:hasQuestion ?Qs.

          ?AnsQ mo:ChoiceOne ?ANSONE.

          ?AnsQ mo:ChoiceTwo ?ANSWTWO.

          ?AnsQ mo:ChoiceThree ?ANSWTHREE.

          ?AnsQ mo:CorrChoice ?ANSCORR.

         ?Qs mo:hasCategory ?Cat.

          ?Cat mo:category ?CatName.

          ?Qs mo:hasLevel ?lev.

         ?lev mo:level ?LevName.

         ?Qs mo:hasScore ?point.

         ?point mo:score ?Score.

         ?Qs mo:hasComplexity ?complexity.

          FILTER ( ?CatName ='"+ctg+"' ).

          FILTER ( ?LevName ='"+lvl+"' ).

          } ORDER BY desc (?complexity)
```

SPARQL Query1 to Load the Quiz Game

```
SELECT  *
        WHERE {
       ?Qs mo:Question ?QsDesc.
        ?QS mo:hasAnswers ?AnsQ.
        ?AnsQ mo:hasQuestion ?Qs.
        ?AnsQ mo:ChoiceOne ?ANSONE.
        ?AnsQ mo:ChoiceTwo ?ANSWTWO.
        ?AnsQ mo:ChoiceThree ?ANSWTHREE.
        ?AnsQ mo:CorrChoice ?ANSCORR.
       ?Qs mo:hasCategory ?Cat.
        ?Cat mo:category ?CatName.
        ?Qs mo:hasLevel ?lev.
      ?lev mo:level ?LevName.
       ?Qs mo:hasScore ?point.
       ?point mo:score ?Score.
       ?Qs mo:hasComplexity ?complexity.
        FILTER ( ?CatName ='"+ctg+"' ).
        FILTER ( ?LevName ='"+lvl+"' ).
        } ORDER BY  (?complexity)
```

SPARQL Query2 to Load the Quiz Game

Query for rdfs:comment

```
SELECT  *



 WHERE { ?question rdfs:label ?label  .

?question  rdfs:comment ?comment .

 "FILTER regex  (?label , '"+s+"' )
```

SPARQL Query for Explanation of Answer

Query for Highest Score

```
SELECT  ?student ?highScore

WHERE

{

{

SELECT  (max(?score) as ?highScore)

 { ?student dd:PhysicsPercentage ?score

} }

  ?student dd:PhysicsPercentage ?highScore }
```

SPARQL query for Highest Score in a Category

APPENDIX 2

SEMANTIC WEB RULES

User Categorization Rule1:

```
[rule1 :(?x http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#User)

( ?x http://www.semanticweb.org/t/ontologies/2016/7/myOWL#PhysicsPercentage  ?marks
)" greaterThan(?marks, 50), lessThan(?marks,81)

     ->  (?x http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#PhysicsAverage )].
```

Jena Rulefor Average User in Physics Category

User Categorization Rule2

```
[rule1 :(?x http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#User)

( ?x http://www.semanticweb.org/t/ontologies/2016/7/myOWL#GeographyPercentage
?marks )" greaterThan(?marks, 80))

     ->  (?x http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#GeographyDiligent )].
```

Jena rule for Diligent user in Geography category

Rule for Preferred Category:

```
Rule2:(?x http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#User)

( ?x  http://www.semanticweb.org/t/ontologies/2016/7/myOWL#Physics_Preferred_Category   ?cat1 )

( ?x  http://www.semanticweb.org/t/ontologies/2016/7/myOWL#Chem_Preferred_Category  ?cat2)

( ?x  http://www.semanticweb.org/t/ontologies/2016/7/myOWL#Geo_Preferred_Category  ?cat3 )

greaterThan(?cat3,?cat1), greaterThan(?cat3,?cat2)

 ->  (?x  http://www.semanticweb.org/t/ontologies/2016/7/myOWL#FavoriteCategory
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#CatGeography )
```

Jena rule for Geography Preferred Category

Rule for Sub-Class/Super-Class Inference:

```
Rule4:(?x  http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#PhysicsDiligent ) + (
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#PhysicsDiligent
http://www.w3.org/2000/01/rdf-schema#subClassOf
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#DiligentUser ) +

 →( ?x  http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#DiligentUser )
```

Sub-class Inference Jena Rule for Diligent User

Rule for High Score Category:

```
Rule3:(?x http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#User)

(?x http://www.semanticweb.org/t/ontologies/2016/7/myOWL#PhysicsPercentage  ?score1
)

 (?x http://www.semanticweb.org/t/ontologies/2016/7/myOWL#ChemistryPercentage
?score2 )

 (?x http://www.semanticweb.org/t/ontologies/2016/7/myOWL#GeographyPercentage
?score3)

 greaterThan(?score1,?score2), greaterThan(?score1,?score3) -> (?x
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#HighScoreCategory
http://www.semanticweb.org/t/ontologies/2016/7/myOWL#CatChemistry)
```

Jena rule for the Highest Score Category for Chemistry

APPENDIX 3

JAVA CODE

Connect to OWL:

```
import com.hp.hpl.jena.ontology.Individual;
import com.hp.hpl.jena.ontology.OntClass;
import com.hp.hpl.jena.ontology.OntModel;
import com.hp.hpl.jena.ontology.OntModelSpec;
import com.hp.hpl.jena.ontology.OntProperty;
import com.hp.hpl.jena.query.Query;
import com.hp.hpl.jena.query.QueryExecution;
import com.hp.hpl.jena.query.QueryExecutionFactory;
import com.hp.hpl.jena.query.QueryFactory;
import java.util.HashMap;
import java.util.Map;
import static baseline.OpenOWL1.OpenConnectOWL;


import java.util.ArrayList;
import java.util.List;


import com.hp.hpl.jena.query.QuerySolution;
import com.hp.hpl.jena.query.ResultSet;
import com.hp.hpl.jena.query.ResultSetFormatter;
import com.hp.hpl.jena.rdf.model.Literal;
import com.hp.hpl.jena.rdf.model.ModelFactory;
import com.hp.hpl.jena.rdf.model.RDFNode;
import com.hp.hpl.jena.rdf.model.Resource;
```

```
import com.hp.hpl.jena.util.FileManager;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.Arrays;
import java.util.Collections;
//Class Start here
class OpenOWL {
  static  String  s, OntModel OpenConnectOWL(){
       OntModel mode = null;
   mode = ModelFactory.createOntologyModel( OntModelSpec.OWL_MEM_RULE_INF );
   java.io.InputStream in = FileManager.get().open( "F://20-8.owl" );
   if (in == null) {
      throw new IllegalArgumentException("File not found");
   }   return  (OntModel) mode.read(in, "");
   } static  com.hp.hpl.jena.query.ResultSet ExecSparQl(String Query){
        com.hp.hpl.jena.query.Query query = QueryFactory.create(Query);
            QueryExecution qe = QueryExecutionFactory.create(query, OpenConnectOWL());
            com.hp.hpl.jena.query.ResultSet results = qe.execSelect();
           return results; }
      static  String GetResultAsString(String Query){
      try {
         com.hp.hpl.jena.query.Query query = QueryFactory.create(Query);
             QueryExecution qe = QueryExecutionFactory.create(query, OpenConnectOWL());
             com.hp.hpl.jena.query.ResultSet results = qe.execSelect();
             if(results.hasNext()){
               ByteArrayOutputStream go = new ByteArrayOutputStream ();
```

```
ResultSetFormatter.out((OutputStream)go ,results, query);
    s = new String(go.toByteArray(), "UTF-8");
}
else{                    s = "rien";
}
} catch (UnsupportedEncodingException ex) {
   Logger.getLogger(OpenOWL.class.getName()).log(Level.SEVERE, null, ex);
} return s;    }}
```

Code When User Select Category/Level and Connect to Game:

```
Level = levelCb.getSelectedItem().toString();
Category = CategoryCb.getSelectedItem().toString();
name=jTextField1.getText();
doOWLWork(myQuery3(Level, Category));
// Adjust the Jpanels when user move to next question
mainpanel.removeAll();
mainpanel.add(qcmpanel1);
mainpanel.repaint();
mainpanel.revalidate();
if(!Question.isEmpty()){   // if list is not empty then do the work
  QsL.setText(Question.get(0).getQs().toString());
   one = Question.get(0).getChoiceOne();
   two = Question.get(0).getChoiceTwo();
   four= Question.get(0).getChoiceThree();
   three = Question.get(0).getCorrectChoice() ;
   jTextArea1.setVisible(false);
  // Shuffle Questions Orders
  List<String> choices = Arrays.asList(one, two, three, four);
  Collections.shuffle(choices);
```

```
c1.setText(choices.get(0));
c2.setText(choices.get(1));
c3.setText(choices.get(2));
c4.setText(choices.get(3));
```

Questions/Choices Initialization Code:

```
public class QuestionsInfo {

    protected String Qs;
    protected String ChoiceOne;
     protected String ChoiceTwo;
     protected String ChoiceThree;
    protected String CorrectChoice;
    protected String level;
    protected String category;
    protected String Score;

// ?Qs   ?CorrAns    ?ChoiceOne  ?ChoiceTwo ?level ?category  ?Score

 public QuestionsInfo(String Qs, String ChoiceOne, String ChoiceTwo,String ChoiceThree, String CorrectChoice, String level, String category, String Score) {
        this.Qs = Qs;
        this.ChoiceOne = ChoiceOne;
        this.ChoiceTwo = ChoiceTwo;
         this.ChoiceThree = ChoiceThree;
        this.CorrectChoice = CorrectChoice;
        this.level = level;
        this.category = category;
```

```
    this.Score = Score;
}

public String getLevel() {
    return level;
}

public String getCategory() {
    return category;
}

public String getScore() {
    return Score;
}



public String getChoiceTwo() {
    return ChoiceTwo;
}

public String getQs() {
    return Qs;
}





public String getChoiceOne() {
```

```
        return ChoiceOne;
    }

    public void setChoiceOne(String ChoiceOne) {
        this.ChoiceOne = ChoiceOne;
    }
     public String getChoiceThree() {
        return ChoiceThree;
    }

    public void setChoiceThree(String ChoiceThree) {
        this.ChoiceThree = ChoiceThree;
    }

    public String getCorrectChoice() {
        return CorrectChoice;
    }

    public void setCorrectChoice(String CorrectChoice) {
        this.CorrectChoice = CorrectChoice;
    }




    @Override
   public String toString() {
    return Qs + " : " + ChoiceOne + " : " + CorrectChoice;
}
```

}

Setting All Questions in User Defined Array:

```
public List<QuestionsInfo1> listQuestions(String Query) {

    ResultSet rs =  OpenOWL1.ExecSparQl(Query);
    List<QuestionsInfo1> results = new ArrayList<>();
    while (rs.hasNext()) {

   QuerySolution sol = rs.nextSolution();
    String Qs = sol.get("QsDesc").asLiteral().getLexicalForm();
    String CorrectChoice = sol.get("ANSCORR").asLiteral().getLexicalForm();
    String ChoiceOne = sol.get("ANSONE").asLiteral().getLexicalForm();
    String ChoiceTwo = sol.get("ANSWTWO").asLiteral().getLexicalForm();
    String ChoiceThree = sol.get("ANSWTHREE").asLiteral().getLexicalForm();
    String level = sol.get("LevName").asLiteral().getLexicalForm();
    String category = sol.get("CatName").asLiteral().getLexicalForm();
    String Score = sol.get("Score").asLiteral().getLexicalForm();

QuestionsInfo1 vi = new QuestionsInfo1(Qs,  ChoiceOne, ChoiceTwo, ChoiceThree, CorrectChoice, level, category, Score);
        results.add(vi);
    }
    return results;  }
```

Code For Pointing Out the Right Answer:

```
boolean testChoicesCorrect(String CorrectAnsw){
        boolean moveTonext =false;


        if(c1.isSelected()){
            if(c1.getText().equals(CorrectAnsw)){
                moveTonext =true;
            }
        }else if(c2.isSelected()){
            if(c2.getText().equals(CorrectAnsw)){
                moveTonext =true;
            }
        }else if(c3.isSelected()){
            if(c3.getText().equals(CorrectAnsw)){
                moveTonext =true;
            }


        }
        else if(c4.isSelected()){
            if(c4.getText().equals(CorrectAnsw)){
                moveTonext =true;}}


        return moveTonext;}
```

Code When Answer is Correct:

```
if(testChoicesCorrect(three)){
        RDFSComment1 rd=new RDFSComment1();
        jTextArea1.setVisible(false);
```

```
// add score
String Qspoint = scorelbl.getText();
switch(Qspoint) {
    case "2 points":
    UserScore+=2;
    break;
    case "4 points":
        UserScore+=4;
         break;          }
```

Code for Displaying RDFS Comment When Answer is Wrong:

```
else if(!testChoicesCorrect(three)){   //if answer wrong, SPARQL query for Comment

        String queryString=
             "prefix dd:<http://www.semanticweb.org/t/ontologies/2016/7/baseline#>"+
      "PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#> "  +
             "prefix dbr: <http://dbpedia.org/page/>"+
       "PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> "+
               "SELECT  * " +

             " WHERE { ?x rdfs:label ?y  . ?x rdfs:comment ?z .  " + "FILTER regex  (?y ,
'"+s+"' ) " +

     System.out.println(queryString);
        Query query = QueryFactory.create(queryString);
       QueryExecution qexec = QueryExecutionFactory.create(query, model) ;
       ResultSet results = qexec.execSelect() ;
```

```
Literal l=results.next().get("z").asLiteral();
String mystr=l.toString();
RDFSComment1 rd=new RDFSComment1();
jTextArea1.setVisible(true);
jTextArea1.setText(mystr);
```

Code For Motivational Message When User Selects Physics Category:

```
JComboBox comboBox = (JComboBox) evt.getSource();

        Object selected = comboBox.getSelectedItem();
        if(selected.toString().equals("Physics")){



          String queryString2=
         "prefix dd:<http://www.semanticweb.org/t/ontologies/2016/7/baseline#>"+

          "SELECT  ?student ?highScore " +

      " WHERE {" + " {"+ "select  (max(?score) as ?highScore)" + " { ?student dd:Physicsscore
?score" +
" }" +
"  } " +
    "?student dd:Physicsscore ?highScore"   +

     "}";

      Query query2 = QueryFactory.create(queryString2);
       QueryExecution qexec = QueryExecutionFactory.create(query2, model) ;
       ResultSet results2 = qexec.execSelect() ;
```

```
while (results2.hasNext())
   {
            QuerySolution binding = results2.nextSolution();
            Resource subj = (Resource) binding.get("student");
      String s=subj.getLocalName().toString();
      Literal r=binding.getLiteral("highScore");
     String ss= r.getLexicalForm();
          qexec.close(); }
```

Load Quiz Game Code Based on User Class:

```
public class myframe extends javax.swing.JFrame {
   List<QuestionsInfo> Question;
       String prefix1 ="PREFIX mo:<http://www.semanticweb.org/t/ontologies/2016/7/myOWL#> ";
       String prefix2 = "PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#> ";
       String prefix3 = "PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> ";
       String prefix4 = "PREFIX owl:<http://www.w3.org/2002/07/owl#> ";
       String prefix5= "PREFIX dbped:<http://dbpedia.org/resource/>";
       String prefix = prefix1+prefix2+prefix3+prefix4+prefix5;
       myI myii = new myI();
private String myQuery(String lvl, String ctg){

       OntModel model=ModelFactory.createOntologyModel();
        InputStream in =FileManager.get().open("F://myontology.owl");
          if (in==null) {
              throw new IllegalArgumentException( "File: " +  " not found");
          }          model.read(in,"");
           String ns="http://www.semanticweb.org/t/ontologies/2016/7/myOWL#";
```

```
Query_ListQuestions =prefix  +

        "SELECT  * " +
         " WHERE { "
        + "?Qs mo:Question ?QsDesc. "
      //  + "?QsDesc rdfs:label ?label ."
         + "?QS mo:hasAnswers ?AnsQ. "
         + "?AnsQ mo:hasQuestion ?Qs. "
         + "?AnsQ mo:ChoiceOne ?ANSONE."
         + "?AnsQ mo:ChoiceTwo ?ANSWTWO."
         + "?AnsQ mo:ChoiceThree ?ANSWTHREE."
         + "?AnsQ mo:CorrChoice ?ANSCORR. "
        + "?Qs mo:hasCategory ?Cat. "
         + "?Cat mo:category ?CatName. "
         + "?Qs mo:hasLevel ?lev. "

        + "?lev mo:level ?LevName. "
        + "?Qs mo:hasScore ?point."
        + "?point mo:score ?Score. "

         + "?Qs mo:hasComplexity ?l. "
       //+ "Qs mo:QuestionComplexity ?l"
         + "FILTER ( ?CatName ='"+ctg+"' ). "
         + "FILTER ( ?LevName ='"+lvl+"' ). "

    + "}"
```

```
            + "ORDER BY  (?l) "
          + "";
System.out.println(Query_ListQuestions);
    return Query_ListQuestions;
  }
  private String myQuery2(String lvl, String ctg){
      Query_ListQuestions =prefix  +

          "SELECT  * " +
           " WHERE { "
          + "?Qs mo:Question ?QsDesc. "
        //  + "?QsDesc rdfs:label ?label ."
           + "?QS mo:hasAnswers ?AnsQ. "
           + "?AnsQ mo:hasQuestion ?Qs. "
           + "?AnsQ mo:ChoiceOne ?ANSONE."
           + "?AnsQ mo:ChoiceTwo ?ANSWTWO."
           + "?AnsQ mo:ChoiceThree ?ANSWTHREE."
           + "?AnsQ mo:CorrChoice ?ANSCORR. "
          + "?Qs mo:hasCategory ?Cat. "
           + "?Cat mo:category ?CatName. "
           + "?Qs mo:hasLevel ?lev. "

          + "?lev mo:level ?LevName. "
          + "?Qs mo:hasScore ?point."
          + "?point mo:score ?Score. "

           + "?Qs mo:hasComplexity ?l. "
         //+ "Qs mo:QuestionComplexity ?l"
           + "FILTER ( ?CatName ='"+ctg+"' ). "
```

```
            + "FILTER ( ?LevName ='"+lvl+"' ). "



        + "}"
                + "ORDER BY desc (?l) "
            + "";
  System.out.println(Query_ListQuestions);
        return Query_ListQuestions;
    }
```

Setter/Getter:

```
public class PreferredCategory {


    private int phCounter;

    private int chCounter;

    private int geoCounter;

   // private String level;

   // private String Category;



   //.... anything else

public PreferredCategory() {

   // public PreferredCategory(int phCounter, int chCounter, int geoCounter) {

        this.phCounter = phCounter;
```

```
    this.chCounter = chCounter;
    this.geoCounter = geoCounter;
    //this.level = level;
    //this.Category = Category;
}
 public int getphCounter() {
    return phCounter+1;
}

public void setphCounter(int phCounter) {
    this.phCounter = phCounter;
}

public int getchCounter() {
    return chCounter+1;
}

public void setchCounter(int chCounter) {
    this.chCounter = chCounter;
}
```

Code for User Creation:

```
OntClass user1 = model.getOntClass(ns + "User");
OntClass phdiligent = model.getOntClass(ns + "PhysicsDiligent");
OntClass chemdiligent = model.getOntClass(ns + "ChemistryDiligent");
OntClass geodiligent = model.getOntClass(ns + "GeographyDiligent");
OntClass phavg = model.getOntClass(ns + "PhysicsAverage");
OntClass chemavg = model.getOntClass(ns + "ChemistryAverage");
OntClass geoavg = model.getOntClass(ns + "GeographyAverage");

OntClass phnovice = model.getOntClass(ns + "PhysicsNovice");
OntClass chemnovice = model.getOntClass(ns + "ChemistryNovice");
OntClass geonovice = model.getOntClass(ns + "GeographyNovice");
```

Code to Calculate Percentage:

```
if (Category.equalsIgnoreCase("Physics")){
Literal scoreLiteral1=scoreValue.asLiteral();
physics=  scoreLiteral1.getInt();
Literal phSumliteral = model.createTypedLiteral(sum);
int i=phSumliteral.getInt();
 indiv.setPropertyValue(phypref, phSumliteral);
 indiv.setPropertyValue(phypercentage,perc);


        }
if (Category.equalsIgnoreCase("Chemistry")){
Literal scoreLiteral1=scoreValue.asLiteral();
```

```
chem=  scoreLiteral1.getInt();
Literal chSumliteral = model.createTypedLiteral(sum);
int i=chSumliteral.getInt();
 indiv.setPropertyValue(chempref, phSumliteral);
 indiv.setPropertyValue(chempercentage,perc);

        }
if (Category.equalsIgnoreCase("Geography")){
Literal scoreLiteral1=scoreValue.asLiteral();
geography=  scoreLiteral1.getInt();
Literal geoSumliteral = model.createTypedLiteral(sum);
int i=geoSumliteral.getInt();
 indiv.setPropertyValue(geopref, geoSumliteral);
 indiv.setPropertyValue(geopercentage,perc);}
```

Code to Save Data to File on Disk:

```
try (FileOutputStream writer = new FileOutputStream("E://userOntology.owl")) {
           inf.write(writer, "RDF/XML");  //inf is inference model

        } catch (IOException ex) {
            Logger.getLogger(myframe.class.getName()).log(Level.SEVERE, null, ex);
        }
```

```
// inf.write(System.out, "N3"); }
```

Code to Save User Previous Score:

```
OntClass user = mymodel.getOntClass(ns + "User");
  OntProperty mydate=mymodel.getOntProperty( ns+ "Date");
  OntProperty score1= mymodel.getOntProperty(ns+"Physicsscore");
  OntProperty score2= mymodel.getOntProperty(ns+"Chemistryscore");
  OntProperty score3= mymodel.getOntProperty(ns+"Geographyscore");
Individual ind = user.createIndividual(ns + myname);
Calendar cal = GregorianCalendar.getInstance();
Literal Date = mymodel.createTypedLiteral(cal);
ind.addLiteral(mydate, Date);
Literal score = mymodel.createTypedLiteral(myscore);
if (mycategory=="Physics"){

    ind.addLiteral(scorePhysics, score); //addLiteral() does not replace previous values
    ind.addLiteral(mydate, Date);
   }
if (mycategory=="Chemistry"){

    ind.addLiteral(scoreChem, score);
   ind.addLiteral(mydate, Date);}
```

```
if (mycategory=="Geography"){

    ind.addLiteral(scoreGeo, score);
   ind.addLiteral(mydate, Date);
   }
      try (FileOutputStream writer2 = new FileOutputStream("E://backup.owl")) {
       mymodel.write(writer2, "RDF/XML");


    } catch (IOException ex) {
        Logger.getLogger(myframe.class.getName()).log(Level.SEVERE, null, ex);
    }
  }
```