

# Development Of Protégé-Based Ontology For E-Learning Semantic Web Applications

Udoiongodwin Ezekiel<sup>1</sup>

Department Of Electrical/Electronic And Computer Engineering,  
University of Uyo, Akwa Ibom State Nigeria

Ozuomba Simeon<sup>2</sup>

Department Of Electrical/Electronic And Computer Engineering,  
University of Uyo, Akwa Ibom State Nigeria  
simeonozuomba@uniuyo.edu.ng

Constance Kalu<sup>3</sup>

Department Of Electrical/Electronic and Computer Engineering  
University of Uyo, Akwa Ibom State Nigeria

**Abstract—** An approach for creating ontology for application for use in semantic e-learning web applications is presented. The approach is based on the use of the command prompt to upload the e-learning content to the Protégé Web Ontology server via the Protégé web API. The ontology is then extracted in Resource Description Format (RDF) format by the Protégé Web Ontology server. Subsequently, the extracted ontology can be access using a web client which is a third party user web application developed to connect and communicate with the Protégé Web Ontology server via the Protégé web API. The detail steps of the procedure for the extraction and visualization of the ontology and access the e-learning contents are presented along with the requisite screenshots and brief discussions of the results. In all, the paper clearly demonstrates the applicability of Protégé Web Ontology server in facilitating creation of ontology using remote access medium. This approach is very suitable for collaborative creating and maintenance of ontologies.

**Keywords—** Ontology, Protégé, Semantic Web, E-Learning, Protégé Web Ontology Server, Application Programming Interface (API)

## 1. Introduction

Across the globe, e-learning tools and applications have continued to evolve especially after the COVID-19 incidence [1,2,3,4]. This has resulted in the rapid growth in the development and adoption of different e-learning contents. As such researchers are working on strategies for more effective presentation and distribution of e-learning contents. As usual, web application and mobile applications have been the most widely used means of presenting e-learning contents to the end users [5,6,7,8]. More especially, semantic web applications [9,10,11,12] are most suitable designed to support e-learning contents. As such in this paper, development of Protégé-based ontology for e-

learning semantic web application is presented. A comprehensive overview of Protégé software is presented along with the methodology for the development of the semantic e-learning web application [13,14,15]. The function decomposition of the application is presented as well as the procedure for the creation of e-learning content and ontology.

Specifically, the approach in this paper is based on the use of the command prompt to upload the e-learning content to the Protégé Web Ontology server via the Protégé web API [16,17]. The ontology is then extracted in Resource Description Format (RDF) format [18,18] by the Protégé Web Ontology server. Subsequently, the extracted ontology can be access using a web client which is a third party user web application developed to connect and communicate with the Protégé Web Ontology server via the Protégé web API. The detail steps of the procedure for the extraction and visualization of the ontology and access the e-learning contents are presented along with the requisite screenshots and brief discussions of the results.

## 2.0 About Protégé Software

Basically, Protégé is a Java-based software that is designed to run on a number of different operating systems such as MacOS, Windows, Unix and Linux. Protégé software consists of a number of modules, namely ontology storage module, knowledge model module, application programming interface (API) module and the graphical user interface (GUI) module. When running Protégé on a desktop, the user will be interacting with the stored ontologies via the Protégé GUI module while application programs, as well as plugins that work with Protégé will interact with the stored ontologies via the Protégé API. In either case, both the GUI and API access the knowledge model module of the Protégé. Through the use of different plugins, Protégé is enabled to import and export ontologies in different formats such as XML, RDF, OWL, as well as the Protégé native format. RDF is the acronym for Resource Description Framework which is and XML-based semantics of data created by W3C. Another popular format for visualizing ontology in Protégé is OntoGraf. OntoGraf is protégé plug-in for ontology graph visualization.

Specifically, it enable classes and individuals in an ontology to be visualized as nodes in a graph while the relationships between the classes and individuals are represented as edges of the graph.

When Protégé is running with an ontology saved as a file, the whole ontology is stored in the memory. However, when the ontology is too large to fit into the memory, then Protégé do have provision for a relational database to hold the ontology. In addition, user programs and plugins.

Finally, apart from using the desktop Protégé to create and view ontology, there is also Protégé Web Ontology server which can be accessed by web application (called web client) Protégé API calls. In this way, one can create a web application and upload the contents via the web application to the Protégé Web Ontology server where the ontology can be created based on the uploaded content. The Protégé Web Ontology server approach is adopted in this study.

### 3.0 Methodology

1) *The major concern of this paper is the development of Protégé-based ontology for semantic web e-learning applications. In this work, the extraction of the ontology is done using Protégé API Server accessed via a web client which is a web application. In this paper, the web client is defined as a web application that is used to connect with Protégé Server through the Protégé API and enable the*

*user to view and create ontologies on the Protégé Server through the client web application.*

The web application is used to present the e-learning framework that provides hosted learning contents and features, whereby a user can receive lecture notes and create assignments on any area of the subject or course. The e-learning framework with the help of the ontology extraction and mapping tools in Protégé software enables the learning contents to be analyzed, the key terms in the course or subject considered are extracted and graphically mapped to show the relationships, and flow of the ideas contained the contents.

### 3.1 System Functional Decomposition

The system consists of three modules that includes Web App, Protégé API Server, and Protégé, as shown in Figure 1. The web app is where the e-learning content is created, store and accessed. The Protégé API Server is where the raw e-learning content is converted to Resource Description Format (rdf) and it is also used to view the rdf content. The Protégé module is used to view the ontograph and rdf content. Also, the Protégé software is used to validate the information using the reasoned plug-in. Particularly, the reasoned pellet 1.5.2 is used to provide a method to test the consistency of the properties, characteristics and constraint of the ontology developed in this paper .

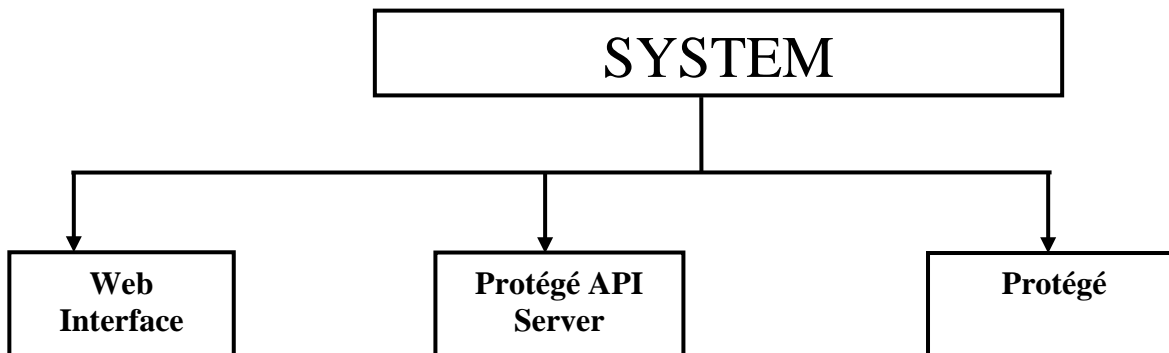


Figure 1 – System Functional Decomposition

### 3.2 Method / Procedures in the Creation of E-Learning Content and Ontology

In order to create the ontology for e-learning content, the requisite contents must first be developed. Again, in this work, the ontology is created by using a web client to access the Protégé Web Ontology server through the Protégé API calls. Essentially, the web application that will be used as the web client is also created

**Step 1:** Create the contents of the e-learning material. In this paper, the subject of the e-learning material is Computer Programming in QBasic Language Laboratory Manual

**Step 2:** Create the web application that will be used as the web client

**Step 3:** The contents of the e-learning are entered or uploaded via the command prompt to the Protégé Web Ontology server

**Step 4:** The Protégé Web Ontology server is started and the Protégé API calls are made to create and extract the ontology from the contents of the e-learning and express the ontology in rdf format

**Step 5:** The extracted rdf file are uploaded to the Protégé desktop application to enable viewing of the graph of the ontology.

The details of the procedures for extraction of the ontology in RDF format using the Protégé web API (Step 3 and Step 5 above).

Uploading the e-learning content to the Protégé Web Ontology server via the Protégé web API is done through the command prompt. The detail steps of the procedure for uploading the e-learning content to the

Protégé Web Ontology server via the Protégé web API and the extraction of the ontology from the uploaded content are presented as follow:

1. Open command prompt
2. Change directory to the protégé web API folder
3. Generate the ontology in rdf format using the command:  

```
generate-mapping -o name-of-rdf-file.ttl -u  
apiusername-p apipassword  
jdbc:apiconnector://127.0.0.1/name-of-rdf-file
```
4. Extract the ontology using the command:  

```
dump-rdf -f RDF/XML -b http://localhost:2020/  
name-of-rdf-file.ttl >name-of-rdf-file.rdf
```
5. Start the ontology server using the command:  

```
d2r-server name-of-rdf-file
```
6. Open a browser and type <http://localhost:2020> to open the protégé web interface
7. Import the rdf file into protégé desktop application to view the graph.

### 3.3 Tools Used for the System Implementation

The system implementation was achieved with the following web application development and deployment tools and technologies:

- i. OPERATING SYSTEM: Windows 7 or higher
- ii. WEBSERVER: Apache
- iii. Protégé software
- iv. SCRIPTING LANGUAGE: PHP
- v. WAMP SERVER: This is a combination of Windows, Apache, MYSQL and PHP tools
- vi. ADOBE DREAMWEAVER CS6: Adobe macromedia dream weaver for webpage design and interface controls.

### 3.4 System Implementation and Tools Used

#### a) Setup of the Application Development Environment – The WAMP Server.

Before the coding of the system, the web application development environment and WAMP server local hosting platform were set up. The WAMP server setup file was downloaded from <http://www.WAMPServer.com/en/download.php>. After downloading, the WAMP server was set up on the computer system that was used to develop and implement the system.

- b) **Creation of the Database and the Tables:** The WAMP server's phyMyAdmin was used to create the database and database tables
- c) **Coding for the Web pages – HTML and PHP:** Adobe Dreamweaver was used in typing and editing the HTML codes and PHP scripts.

- d) **Deployment of the Web Application:** Apache web server, already embedded in WAMP Server, was used for the PHP web application deployment. The complete web application folder named "index" was copied and pasted into the wamp www folder. Next, the web application site address (in this case, <http://localhost/ontology>) was typed into the address bar of a web browser like the Google Chrome. The homepage of the web application opens in the browser. By clicking on any link on the web application, the corresponding page to the link is displayed on the browser.
- e) **Carryout the Procedures for the Creation of Ontology via the Protégé web API**

Once the web application is ready and locally hosted on a computer with internet access, the uploading of the e-learning content to the Protégé Web Ontology server via the Protégé web API is done through the command prompt, as state in section 3.2.

## 4. Results and Discussion

In this section, the detail screenshot of the procedure for extraction of the ontology in RDF format using the Protégé Web Ontology server accessed by the web application via the Protégé web API is presented along with the web application visualization of the e-learning contents, as well as screenshots of some key forms used in uploading the e-learning contents. The steps and the accompanying screenshots are presented as follows;

**Step 1:** Open command prompt (shown in Figure 2)

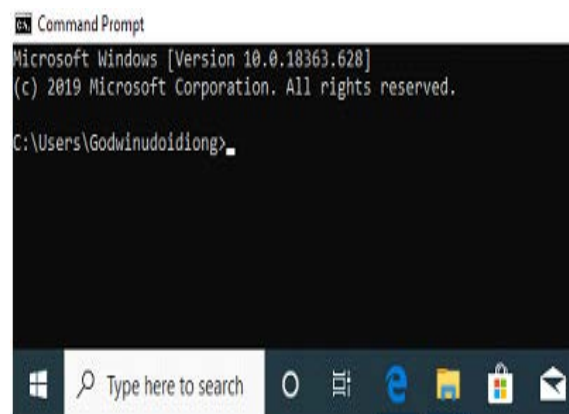


Figure 2 Screenshot showing the loading of command prompt

**Step 2:** Change directory to the protégé Web API folder (shown in Figure 3)

Navigate to command prompt by typing ( cd\ ) and press enter on the computer keyboard, (shown in Figure 3).

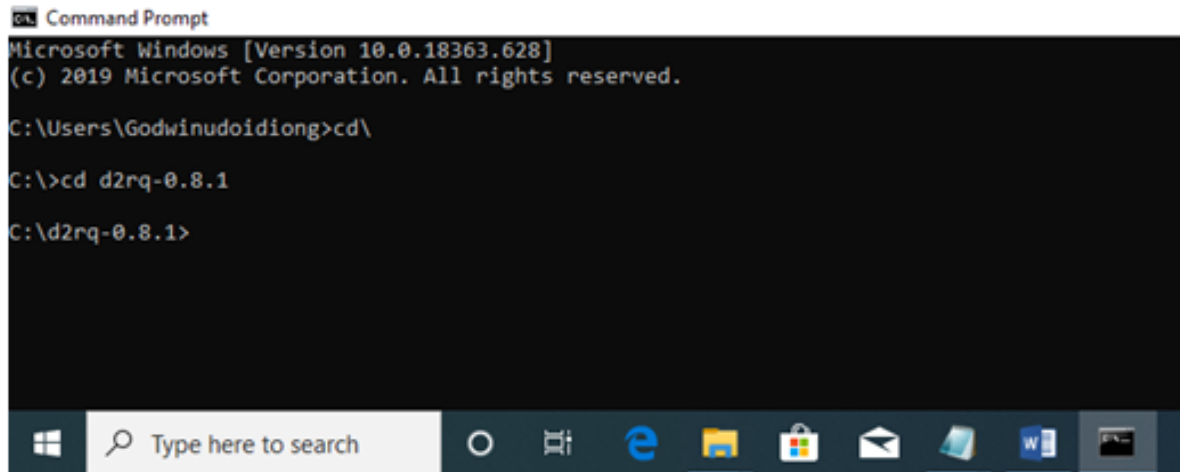


Figure 3 Screenshot showing change directory to the protégé Web API folder

**Step 3:** Generate the Ontology in rdf format by using the command prompt (shown in Figure 4)

`generate-mapping -o rdf.ttl -u root jdbc:mysql://127.0.0.1/rdf`  
and press enter on the computer keyboard (shown in Figure 4).

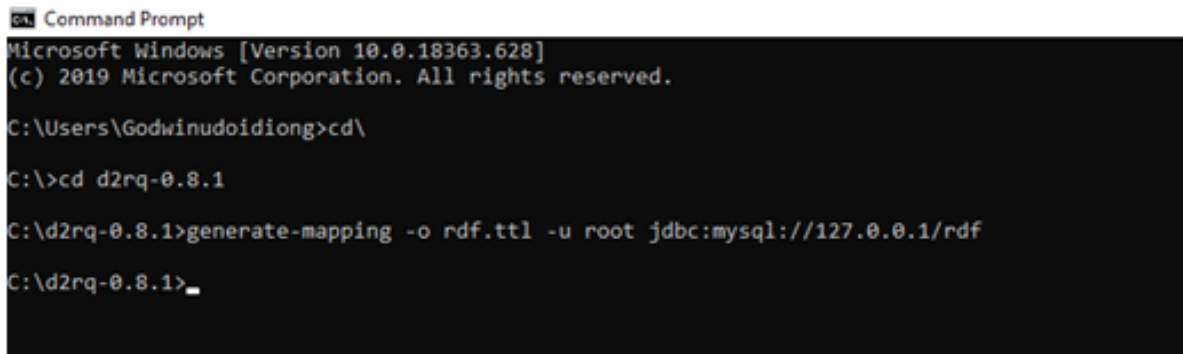


Figure 4 Screenshot showing how to generate the ontology using the prompt

**Step 4:** The command for Extracting the Ontology (shown in Figure 5).

`dump-rdf -f RDF/XML -b http://localhost:2020/ rdf.ttl > myresearch.rdf` (into command prompt) then press enter on the keyboard (shown in Figure 5).

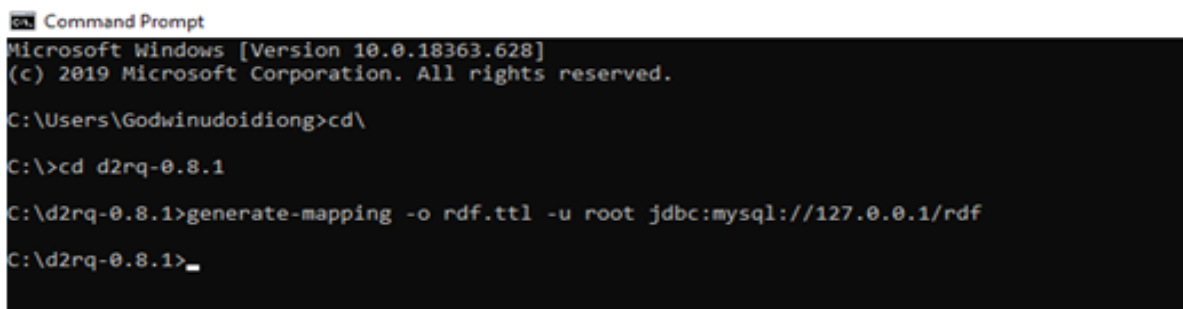


Figure 5 Screenshot showing the command for Extracting the Ontology

**Step 5:** Start the Ontology server using the command prompt (shown in Figure 6)

To start the Ontology server using the command prompt so as to view the rdf file in the server enter this `-d2r-server rdf.ttl` and press on the computer keyboard, (shown in Figure 6).

```
Command Prompt
Microsoft Windows [Version 10.0.18363.628]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Godwinudoidiong>cd\
C:\>cd d2rq-0.8.1
C:\d2rq-0.8.1>generate-mapping -o rdf.ttl -u root jdbc:mysql://127.0.0.1/rdf
C:\d2rq-0.8.1>dump-rdf -f RDF/XML -b http://localhost:2020/ rdf.ttl > myproject.rdf
C:\d2rq-0.8.1>
```

Figure 6, Screenshot showing the command to start the server, showing the code in the Ontology using the command prompt

Step 6: Open a browser and type <http://localhost:2020> and press enter on the computer keyboard to open the protégé Web interface (shown in Figure 7)

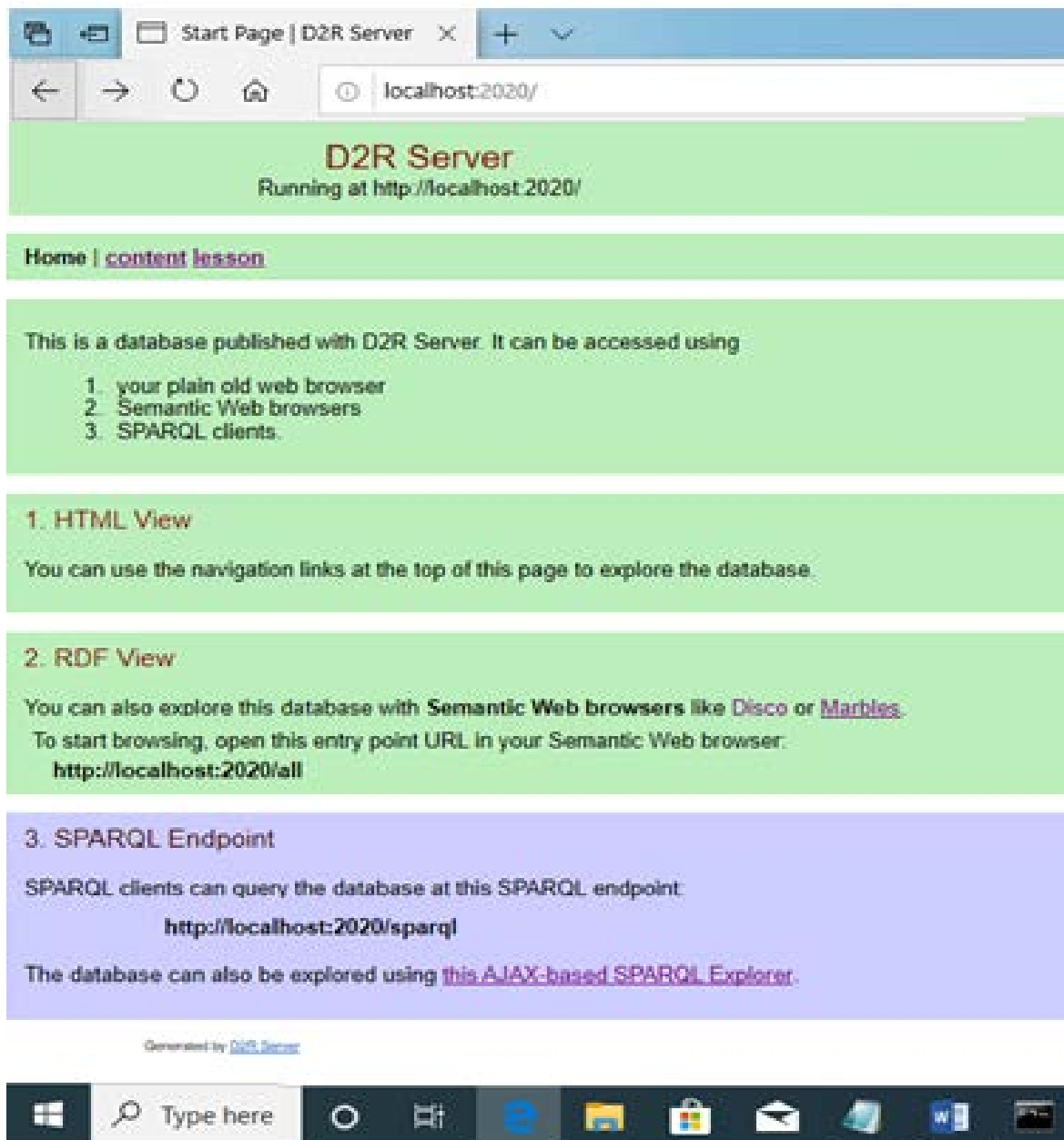


Figure 7 Screenshot showing how to open the protégé server in a web application

**Step 7:** To load the Ontology in a web application (shown in Figure 8)

To load the Ontology in a web application type:  
“http://localhost/Ontology” on the browser (shown in

Figure 8).

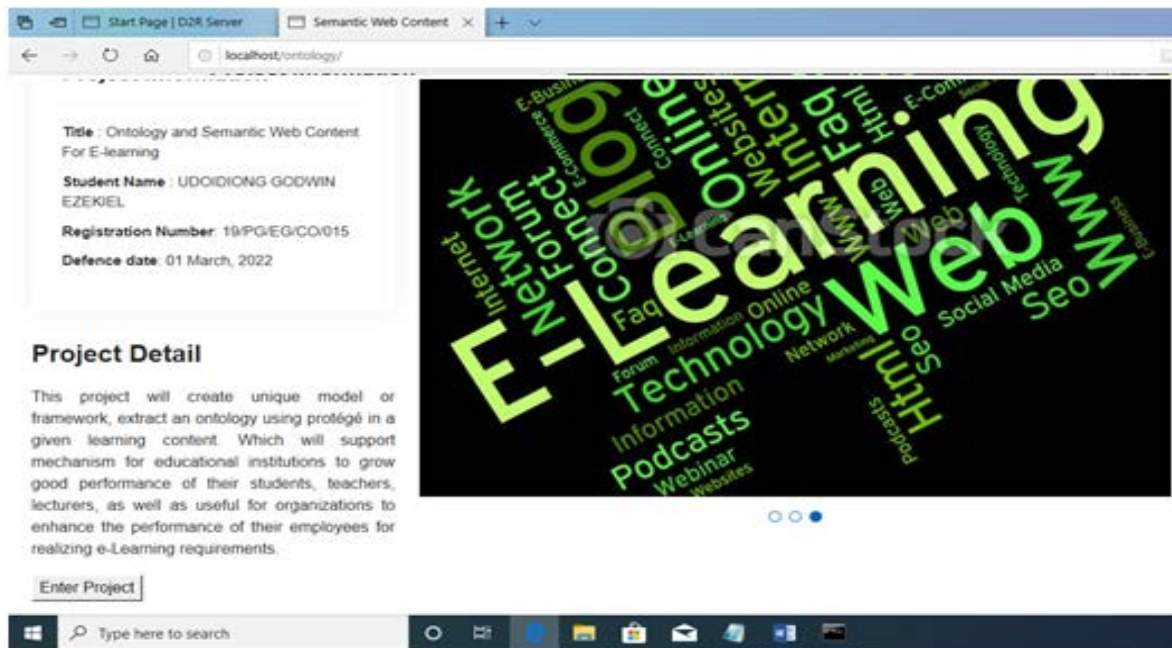


Figure 8 Screenshot showing how to load the Ontology in a web application

This contains the students' information, the login (Enter research) to load the form for accessing the lecture note, assignment, etc. Sample screenshot of two e-learning contents web pages are shown in Figure 9 and Figure 10.

The screenshot showing the input form for registering the course information is presented in Figure 11 while the

screenshot showing the input form for uploading the course lessons and assignments is presented in Figure 12.

Furthermore, screenshot showing the lesson rdf description of the web protégé interface is presented in Figure 13. To view the ontograf in the protégé, import the rdf into protégé desktop application to view the ontograf, lessons, etc. (shown in Figure 14).



Figure 9 First sample screenshot of e-learning contents web page

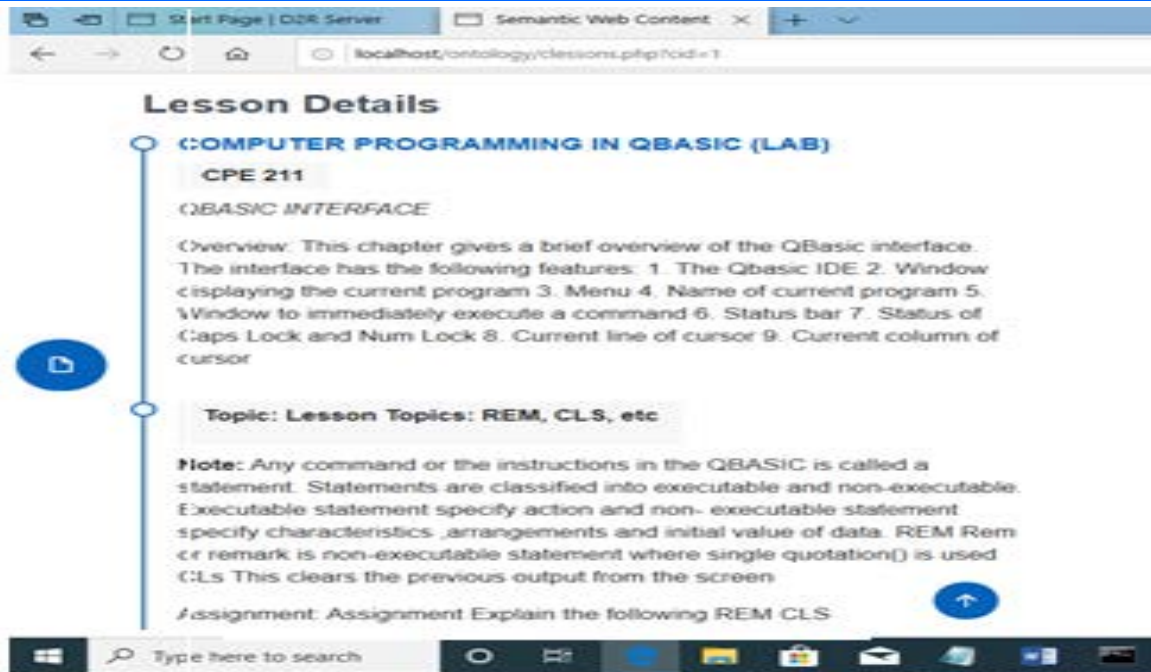


Figure 10 Second sample screenshot of e-learning contents web page

A screenshot of a web form titled 'E-LEARNING'. The form contains four input fields: 'Course Title', 'Course Code', 'Course Description', and 'Overview of the chapter'. At the bottom of the form is a button labeled 'Save Course Information'. The entire form is enclosed in an orange border.

Figure 11 The screenshot showing the input form for registering the course information

## ENTER LESSON DEATILS

Lesson Topic

Assignment

Lesson Note

Save Lesson Details

Figure 12 The screenshot showing the input form for uploading the course lessons and assignments

### lesson #1

Resource URI: <http://localhost:2020/resource/lesson/1>

[Home](#) | [All lesson](#)

Property	Value
<a href="#">rdfs:label</a>	lesson #1
<a href="#">vocab:lesson_cid</a>	1 (xsd:integer)
<a href="#">vocab:lesson_id</a>	1 (xsd:integer)
<a href="#">vocab:lesson_lassignment</a>	Write a brief history of QBASIC
<a href="#">vocab:lesson_lnote</a>	QBASIC is similar to fortran
<a href="#">vocab:lesson_ltopic</a>	Introduction to QBASIC
<a href="#">rdf:type</a>	<a href="#">vocab:lesson</a>

*The server is configured to display only a limited number of values (limit per property bridge: 50).*

#### Metadata

```
<http://localhost:2020/data/lesson/1>  
dc:date 2021-12-18T20:48:45.195Z  
prv:containedBy <http://localhost:2020/dataset>  
void:inDataset <http://localhost:2020/dataset>  
rdf:type prv:DataItem  
rdf:type foaf:Document
```

Figure 13: Screenshot showing the lesson rdf description of the web protégé interface



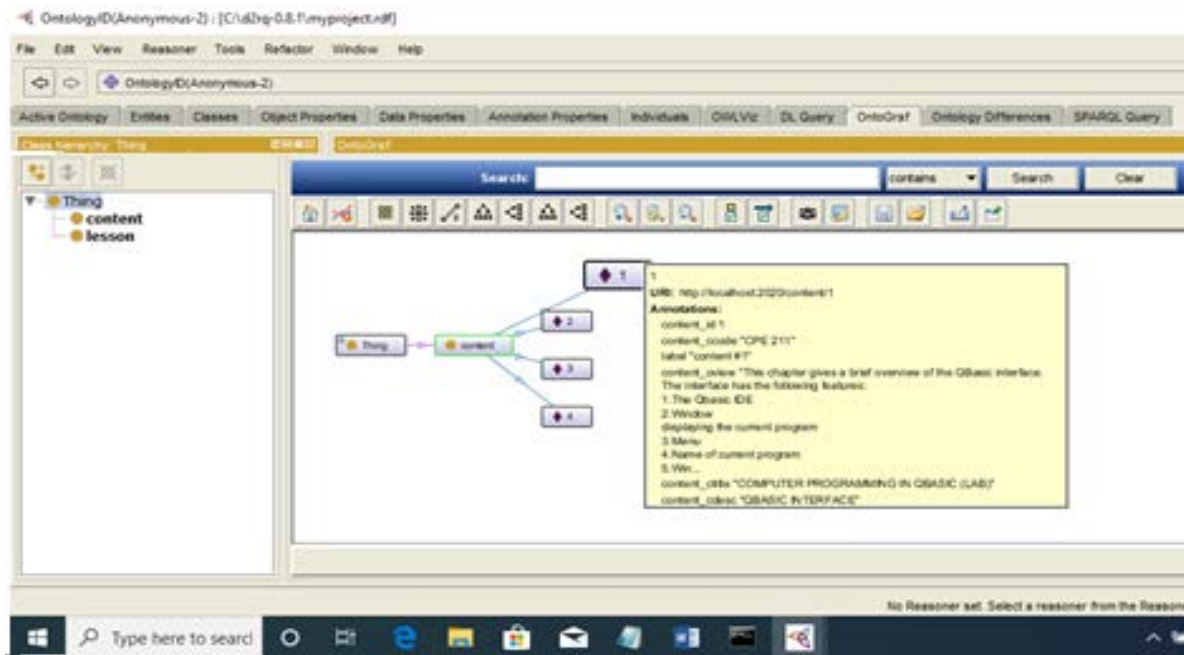


Figure 14 Screenshot showing the ontograp in Protégé

#### 4. Conclusion

An approach for creating ontology for application in use in semantic e-learning web applications is presented. The approach is based on the use of the command prompt to upload the e-learning content to the Protégé Web Ontology server via the Protégé web API. The ontology is then extracted in rdf format by the Protégé Web Ontology server. Subsequently, the extracted ontology can be access using a web client which is a user third party web application developed to connect and communicate with the Protégé Web Ontology server via the Protégé web API.

The detail steps of the procedure for the extraction and visualization of the ontology and access are presented along with the requisite screenshots and brief discussions of the results.

#### References

1. Qiao, P., Zhu, X., Guo, Y., Sun, Y., & Qin, C. (2021). The development and adoption of online learning in pre-and post-COVID-19: Combination of technological system evolution theory and unified theory of acceptance and use of technology. *Journal of Risk and Financial Management*, 14(4), 162.
2. Zalat, M. M., Hamed, M. S., & Bolbol, S. A. (2021). The experiences, challenges, and acceptance of e-learning as a tool for teaching during the COVID-19 pandemic among university medical staff. *PLoS one*, 16(3), e0248758.
3. Chavarría-Bolaños, D., Gómez-Fernández, A., Dittel-Jiménez, C., & Montero-Aguilar, M. (2020). E-Learning in Dental Schools in the Times of COVID-19: A Review and Analysis of an Educational Resource in Times of the COVID-19 Pandemic. *Odovtos International Journal of Dental Sciences*, 22(3), 69-86.
4. Habes, M., Ali, S., Khalid, A., Haykal, H. A., Elareshi, M., Khan, T., & Ziani, A. (2021). E-Learning acceptance during the covid-19 outbreak: a cross-sectional study. In *Artificial Intelligence Systems and the Internet of Things in the Digital Era: Proceedings of EAMMIS 2021* (pp. 65-77). Cham: Springer International Publishing.
5. Nedungadi, P., & Raman, R. (2012). A new approach to personalization: integrating e-learning and m-learning. *Educational Technology Research and Development*, 60, 659-678.
6. Castiblanco Jimenez, I. A., Cepeda García, L. C., Violante, M. G., Marcolin, F., & Vezzetti, E. (2020). Commonly used external TAM variables in e-learning, agriculture and virtual reality applications. *Future Internet*, 13(1), 7.
7. Frehywot, S., Vovides, Y., Talib, Z., Mikhail, N., Ross, H., Wohltjen, H., ... & Scott, J. (2013). E-learning in medical education in resource constrained low-and middle-income countries. *Human resources for health*, 11, 1-15.
8. Asarbaksh, M., & Sandars, J. (2013). E-learning: the essential usability perspective. *The clinical teacher*, 10(1), 47-50.
9. Pauwels, P., Zhang, S., & Lee, Y. C. (2017). Semantic web technologies in AEC industry: A literature overview. *Automation in construction*, 73, 145-165.
10. Hendler, J., & Berners-Lee, T. (2010). From the Semantic Web to social machines: A research challenge for AI on the World Wide Web. *Artificial intelligence*, 174(2), 156-161.
11. Fensel, D., Facca, F. M., Simperl, E., & Toma, I. (2011). *Semantic web services* (Vol. 357). Heidelberg: Springer.

12. Fisher, M., Blace, R., Hebler, J., & Perez-Lopez, A. (2011). *Semantic web programming*. John Wiley & Sons.
13. Musen, M. A. (2015). The protégé project: a look back and a look forward. *AI matters*, 1(4), 4-12.
14. Giri, K., & Gokhale, P. (2015). Developing a banking service ontology using Protégé, an open source software.
15. Thakor, P., & Sasi, S. (2015). Ontology-based sentiment analysis process for social media content. *Procedia Computer Science*, 53, 199-207.
16. Redmond, T. (2010). An Open Source Database Backend for the OWL API and Protege 4. In *OWLED* (Vol. 614).
17. Zeebaree, A. Z. S. R., Adel, A. Z., Jacksi, K., & Selamat, A. (2019). Designing an ontology of E-learning system for duhok polytechnic university using protégé OWL tool. *J Adv Res Dyn Control Syst Vol*, 11, 24-37.
18. Kinjo, A. R., Suzuki, H., Yamashita, R., Ikegawa, Y., Kudou, T., Igarashi, R., ... & Nakamura, H. (2012). Protein Data Bank Japan (PDBj): maintaining a structural data archive and resource description framework format. *Nucleic acids research*, 40(D1), D453-D460.
19. Jevsikova, T., Berniukevičius, A., & Kurilovas, E. (2017). Application of resource description framework to personalise learning: Systematic review and methodology. *Informatics in Education*, 16(1), 61-82.