International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

# Design Ontology for Open Learning Domain

**Dr. Mohammed Awad Mohammed Ata Elfadiel[1], Eiman Alsiddig Altayeb Ibrahim[2]**

[1]Open University of Sudan, E-learning Center, Khartoum – Sudan

[2]Sudan University of Science and Technology, College of Computer Science and Information Technology, Khartoum – Sudan

**Abstract:** ***Ontologies have become core components of many large applications. One of the more common goals in developing the ontologies is to Share common understanding of the structure of information among people or software agents. This paper addresses the issues of constructing ontology for Open Learning Domain and presents a methodology for creating the ontology. This paper listed the steps of ontology development process, and addressed the complex issues of defining class hierarchies, enumerated the terms that can possibly found in the domain, and then organized them in a hierarchy depending on which class subsumes another. And then defined properties for each class and the relationships linked these classes. Final step in designing the ontology was defining some instances to be able to make queries. After The Ontology has been designed, it sent to the Reasoner to check classes' consistency and to compute subsumption relationships. An important result of this research is designing ontology for Open Learning Domain, explicit representations and full definitions for MOODLE objects, properties and their relationships. Also can make any query to retrieve information about MOODLE. Furthermore, the designed ontology can be shared and reused in applications related to Open Learning Domain.***

**Keywords:** Ontology, MOODLE, eLearning, reasoned

## 1. Introduction

Ontologies are typically specified in languages that allow abstraction away from data structures and implementation strategies; in practice, the languages of ontologies are closer in expressive power to first-order logic than languages used to model databases. For this reason, ontologies are said to be at the "semantic" level, whereas database schema are models of data at the "logical" or "physical" level. Due to their independence from lower level data models, ontologies are used for integrating heterogeneous databases, enabling interoperability among disparate systems, and specifying interfaces to independent, knowledge-based services. In the technology stack of the Semantic Web standards, ontologies are called out as an explicit layer. There are now standard languages and a variety of commercial and open source tools for creating and working with ontologies [1].

MOODLE (Modular Object-Oriented Dynamic Learning Environment) is a free open-source learning management system or e-Learning platform, that serves educators and learners across the globe.

MOODLE was developed in 2002 by Martin Dougiamas to help educators create online courses with a focus on interaction and collaborative construction of content. Since then, the main development of MOODLE is led by Martin and the core team at MOODLE Headquarters, as well as hundreds of other developers around the world who have helped fuel the growth of MOODLE through contributing and testing code, and being active participants in community forums [4].

MOODLE is mostly useful to programmers and education theorists. As such it applies both to the way MOODLE was developed, and to the way a student or teacher might approach studying or teaching an online course[2].

An Open Learning Domain becomes more important due to the continuous increase in student number joined open learning systems and increase in systems work with this domain. Thus, these systems need to be ontologically committed. Problem is that not agreeing on the meaning of terms may result in misunderstanding. So can we develop an ontology for Open Learning Environment Domain?

## 2. Method

Ontology engineering (or ontology building) is a subfield of knowledge engineering that studies the methods and methodologies for building ontologies. It studies the ontology development process, the ontology life cycle, the methods and methodologies for building ontologies, and the tool suites and languages that support them.

**Protégé-OWL**

The Protégé-OWL is an extension of Protégé that supports the Web Ontology Language (OWL). OWL is the most recent development in standard ontology languages, endorsed by the World Wide Web Consortium (W3C) to promote the Semantic Web vision. "An OWL ontology may include descriptions of classes, properties and their instances. Given such an ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailments may be based on a single document or multiple distributed documents that have been combined using defined OWL mechanisms.

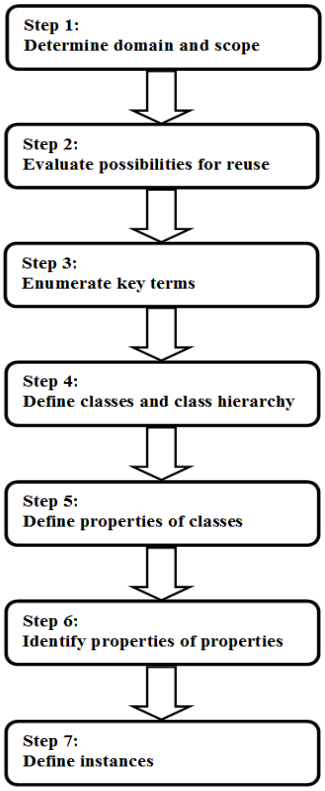


**Figure1:** Ontology Development **[3].**

Ontology engineering is a set of tasks related to the development of ontologies for a particular domain. The main steps in the methodology are as follows (Figure 1).

### 2.1 Determine the domain and scope of the ontology

One of the ways to determine the scope of the ontology is to sketch a list of questions that a knowledge base based on the ontology should be able to answer, competency questions. These questions will serve as the litmus test later: Does the ontology contain enough information to answer these types of questions? Do the answers require a particular level of detail or representation of a particular area? These competency questions are just a sketch and do not need to be exhaustive [7].

In the Open Learning domain, the following are the possible competency questions:

- How users can login to the system (MOODLE)?
- How to add new users?
- How to search for, edit, delete or perform bulk actions on users?
- Who are MOODLE users?
- How to add or remove permissions from students, teachers and other users on MOODLE, and who can do this?
- Who has rights to create and manage the courses?
- Who can enroll users into courses and how?
- To which extent allowed to MOODLE user to deal with the resources, courses and activates?

Judging from this list of questions, the ontology will include the information on the process of allowing a user to login to a MOODLE site based on their username and password, types of MOODLE users and process of assigning users to roles (usually student) in a courses, and soon.

### 2.2 Consider reusing existing ontologies

Reusing existing ontologies may be a requirement if a system needs to interact with other applications that have already committed to particular ontologies or controlled vocabularies [7]. In this research, the ontology will be built from scratch.

### 2.3 Enumerate important terms in the ontology

It is useful to write down a list of all terms we would like either to make statements about or to explain to a user. What are the terms we would like to talk about? What properties do those terms have? What would we like to say about those terms? For example, important Open Learning-related terms will include:

***MOODLE site, MOODLE courses, Authentication, Accounts, New User, Enrolments, Roles, permissions, Courses, Activities, Resources, Administrator, Course Creator, Teachers, Non Editing Teacher, Students, Guest, Questions, Activities, Resources, Blocks, User name, Password, Course Name, Course ID, Course Duration, Student No, Student Name,hasAuthenticates, hasCreates, isCreatedBy, hasEnroles, hasManages, helps, views*** and so on.

Initially, it is important to get a comprehensive list of terms without worrying properties that the concepts may have, or whether the concepts are classes or slots.

The next two steps, developing the class hierarchy and defining properties of concepts (slots) are closely intertwined. It is hard to do one of them first and then do the other. These two steps are also the most important steps in the ontology design process.

### 2.4 Define the Classes and the Class Hierarchy

There are several possible approaches in developing a class hierarchy:

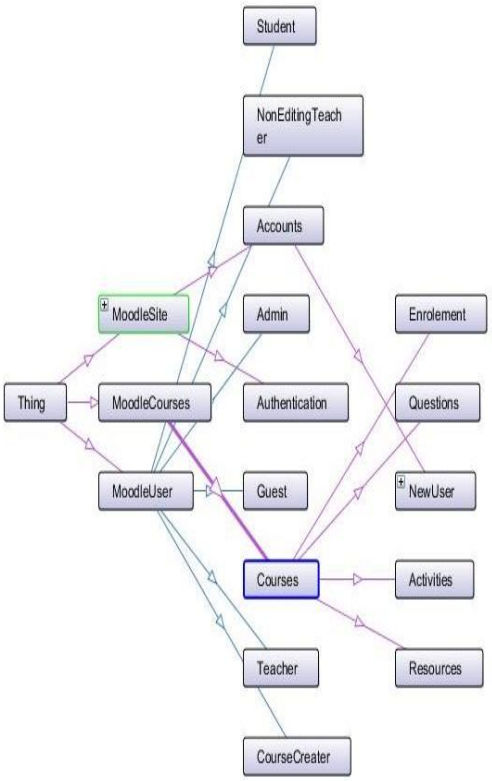


**Figure 2:** MOODLE Classes Hierarchy

### 2.5 Define the properties of classes

The classes alone will not provide enough information to answer the competency questions from Figure 1 Step 1. Once we have defined some of the classes, we must describe the internal structure of concepts.

We have already selected classes from the list of terms we created in Figure 1 Step 3. Most of the remaining terms are likely to be properties of these classes. These terms include**:**

- ***has Authenticates and its sub properties:hasFullAuthenticatesOn and hasPartialAuthenticatesOn.***
- ***hasCreates.***
- ***isCreatedBy***which is inverse property to***hasCreates.***
- ***hasEnroles***and its sub properties: ***hasManualEnrolesIn and hasSelfEnrolesIn.***
- ***hasManages***and its sub properties:***hasFullManageOn and hasPartialManageOn,User name, Password, Course Name, Course ID, Course Duration, Student No*** and***Student Name.*** For each property in the list, we must determine which class its describe. These properties become slots attached to classes. Thus, **MoodleUser** class will have the following properties:

***has Authenticates*** **and its** ***sub properties:hasFullAuthenticatesOn***belong to the **Admin as** in Figure3**.**
***hasPartialAuthenticatesOn,*** belong to the **CourseCreater as** in Figure3**.**
***helps***belong to the **NonEditingTeacher**, ***Student No*** and***Student Name*** belong to **Student,** ***views*** belong to the **Guest**.
**MOODLECourses** class will have a ***Course Name, Course ID, Course Duration, hasCreates and hasEnroles***with its sub properties:***hasManualEnrolesIn***and***hasSelfEnrolesIn*** properties.

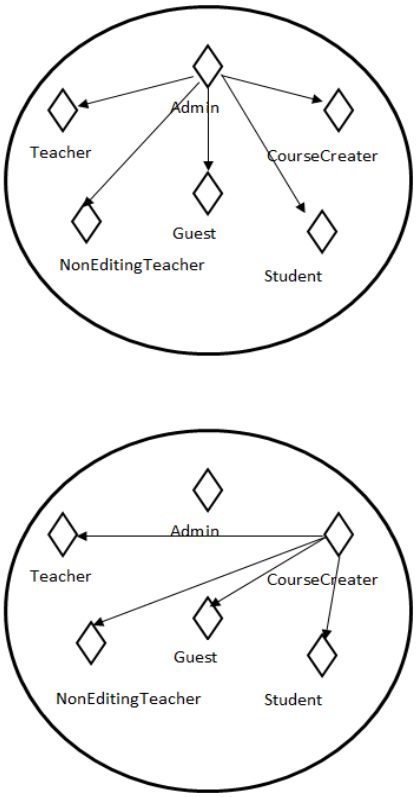


**Figure 3:** has Full Authenticates On, has Partial Authenticates Onproperties.

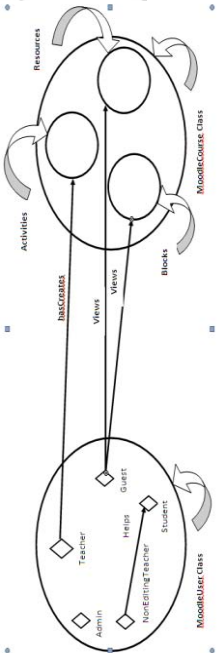


**Figure 4:** hasCreates and Views Relationship

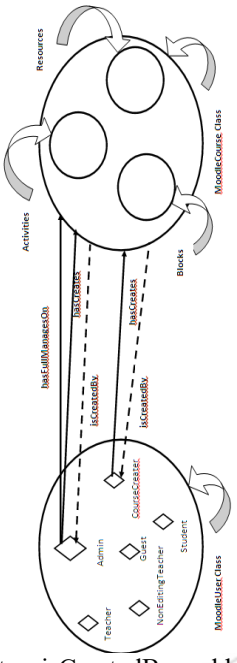


**Figure 5:** hasCreates, isCreatedBy and hasFullManagesOn relationship

Relationships to other individuals, these are the relationships between individual members of the class and other as shown in (Figure 4and Figure 5).

### 2.6 Define properties of properties

This step is to identify the properties that each property has.

Properties may have a domain and a range specified. Properties link individuals from the domain to individuals from the range. For example, in this ontology, the property hasCreates would probably link individuals belonging to the class **MoodleUser** to individuals belonging to the class of **MoodleCourse**. In this case the domain of the hasCreates property is **MoodleUser** and the range is **MoodleCourse** this is depicted in (Figure 6).

It is important to realize that in OWL domains and ranges should not be viewed as constraints to be checked. They are used as `axioms' in reasoning.

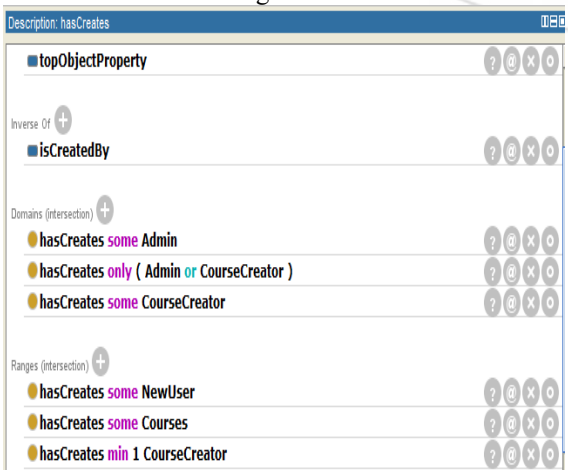


**Figure 6:** Object Restriction

### 2.7 Create instances

At this stage the ontology populate with instances of classes. For example, class Courses may have instances such as DataBase and SoftwareEngineering.

After finished designing the ontology and ensure that all classes are consistent and well defined, then can make queries using SPARQL. For example,

**Select ***
**Where**
**?subject ?hasCreates ?object**

And the result shown in Figure7

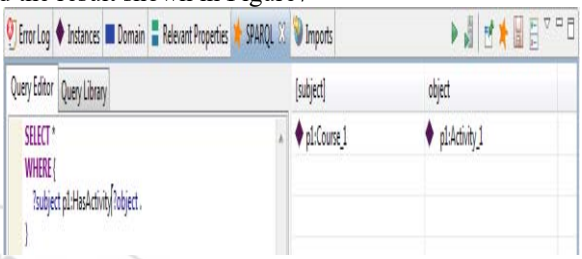


**Figure 7:** A query using SPARQL

## 3. Results

After developing the ontology has been finished, it sent to the reasoner to check class consistency and to compute subsumption relationships. The reasoner generates inferred class hierarchy and the researcher ensures that all the classes are consistent with its own definitions.

As result, have explicit representation and full definitions for MOODLE objects, properties and their relationships. And also can make any query to retrieve information. For example, we can make a query to know who can create courses or who has full authentication right or who can helps students and so on.

### 3.1 Results Discussion

After the Reasoner has been invoked - via the „start Reasoner" button in the Reasoner drop down menu- to automatically compute the classification hierarchy, and also to check the logical consistency of the ontology ,inconsistent classes appeared and Reasoner stopped working. After several times to fix this mistake , the researcher found out that it was due to logical mistake, "hasSelfEnrole" property defined to be functional and transitive at same time( Figure8). The inconsistency fixed and the Reasoner invoked once again.

The inferred hierarchy has been computed, and an inferred hierarchy window popped on top the existing asserted hierarchy window. (Figure9) and (Figure10) display the difference before and after invoking the reseaoner.

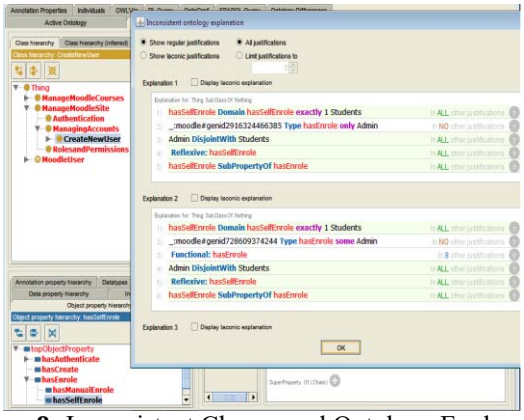

**Figure 8:** Inconsistent Classes and Ontology Explanation

## 4. Conclusion

In this research Protégé Framework and OWL used to design the ontology for Open Learning Environment, enumerated the terms that can possibly found in the domain, then organized them in a hierarchy depending on which class subsumes another, and defined properties for each class. After that the steps went deeply to define constraints on the properties such as: cardinality constraints, domain and range constraints. Final step in designing the ontology was defining some instances to be able to make queries. After that, the Reasoner used to check the consistency of the classes and to generate inferred class hierarchy.

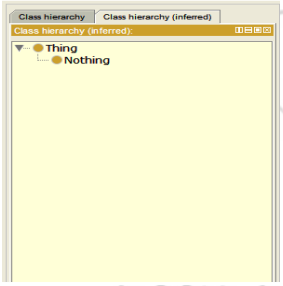


**Figure 9:** Inferred Class Hierarchy before Invoking the Reasoner.

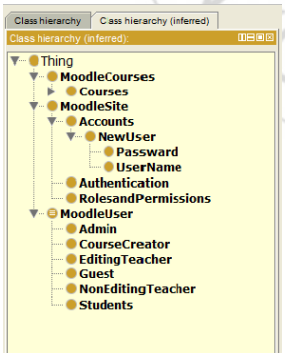


**Figure10**: Inferred Class Hierarchy After Invoking the Reasoner.

If a class has been found to be inconsistent its icon will be highlighted in red color as in (Figure 8).

## References

[1] Tom Gruber "Ontology" , Stanford University, 2008.
[2] http://docs.MOODLE.org/23/en/Authentication14/2, 5:25 pm
[3] Michael Wooldridge "An Introduction to MultiAgent Systems" ,University of Liverpool,2008.
[4] http://www.moodlerooms.com/community-initiatives/what-is-moodle
[5] Adina Magda Florea "Introduction to Multi-Agent Systems", University of Bucharest,1998.
[6] Fernando Batista, Joana Paulo, Paula Vaz, Ricardo Ribeiro "Ontology construction" , February 9, 2006.
[7] Floriano Z0ini , Leon Sterling "Designing Ontologies for Agents", Department of Computer Science Software Engineering,University of Melbourne, Parkville 30502,Australia,1999.
[8] Matthew Horridge " A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools" , Edition 1.3.
[9] http://protege.stanford.edu/overview/protege-owl.html ,14/2, 3:11 pm

## Author Profile

**Mohammed Awad Mohammed Ata Elfadiel**received the B.S. degree in Computer science form Omdurman Islamic University in 2007, M.S.C degree in Computer Programming from Sudan Academy of Science in 2011, PhD in computer science from Alribat National University in 2015. Now works for Open University of Sudan.

**EimanAlsiddigAltayeb Ibrahim** received the B.S. degree in Computer science form Omdurman Islamic University in 2007, M.S.C degree in Computer Science from Sudan University of Science and Technology in 2014, PhD researcher in computer science at Sudan University of Science and Technology. Now works for Omdurman Islamic University.