Ontology Formation and Comparison for Syllabus Structure Using NLP

Masoom Raza

Aditee Patil

Mangesh Bedekar

Rashmi Phalnikar

Bhavana Tiple

MIT World Peace University, Pune, India

Corresponding author: Masoom Raza, Email: masoomrz100@gmail.com

Ontologies are largely responsible for the creation of a framework or taxonomy for a particular domain which represents the shared knowledge, concepts and how these concepts are related with each other. This paper shows the usage of ontology for the comparison of a syllabus structure of universities. This is done with the extraction of the syllabus, creation of ontology for the representing syllabus, then parsing the ontology and applying Natural language processing to remove unwanted information. After getting the appropriate ontologies, a comparative study is made on them. Restrictions are made over the extracted syllabus to the subject "Software Engineering" for convenience. This depicts the collection and management of ontology knowledge and processing it in the right manner to get the desired insights.

Keywords: Ontology, Knowledge management, Natural language processing, RDF, OWL

2021. In Raju Pal & Praveen K. Shukla (eds.), *SCRS Conference Proceedings on Intelligent Systems*, 71–78. Computing & Intelligent Systems, SCRS, India.
https://doi.org/10.52458/978-93-91842-08-6-6

1 Introduction

The management of knowledge and its effectiveness are today considered a powerful asset to stay competitive in almost any marketplace. There are two types of knowledge, i.e., Explicit knowledge, tacit knowledge. Ontology works on explicit knowledge. Building ontology is a great way to populate knowledge management. Ontology represents several concepts under the target domain in a hierarchical way with the relations in between them. The core pillars of knowledge management are people, process, and technology. These support the architecture for knowledge to flow and get utilized effectively. Ontology is gaining popularity for the knowledge management and knowledge representation and how to proceed with the sharing of this knowledge effectively among the actively connected communities.

2 Background

Traditional ontology is concerned with the fundamental questions related to the existence of entities and how they can be meaningfully grouped together and described based on, for example, their similarities and differences. Gruber defines an ontology as "the specification of conceptualizations, used to help programs and humans share knowledge" [2] is also defined as "explicit, formal specification of a shared conceptualization" [4].

In [1], it is well stated the creation of a natural language interface to ontologies so that normal users can acquire the required information easily. There are several major obstacles in the usage of NLP about how ambiguity and complexity make it difficult for machines to understand the language [3].

There is a great need for good representation of information and knowledge in a machine understandable format. There are a huge number of universities and syllabus structures and one often faces difficulty in selecting a syllabus structure which suits them as per their requirement. This paper gives a method for the comparison study and analysis of ontologies based on universities syllabus structure. This not only saves time but also provides an easier method to know which syllabus for a particular course will be better and at what level.

3 Ontology Model

This paper focuses on the formation of ontology for the subject "Software Engineering". The model for representing ontology is formed by defining several components. Mainly these components are Classes (concepts), Relation and Instances. In ontology several classes are related to other classes in the domain. The classes are defined and then described by their attributes. These attributes have something in common (shared) that's why they are forming a class. Attributes describe the class and attributes can themselves be given by name and value pair

Classes are used by individuals (instances or Objects). Instances populates the classes and classes populates the ontology. Let M bet the set of classes (m_1, m_2, m_3....... m_n). The relation R is a subset of the cross product of these classes (m_1 x m_2 x m_3 x.......x m_n). Relations are the special attribute, whose values are objects of other classes.

Below is a simple example of ontology on "Prescriptive Process Model", one of the topics under Software Engineering. The information regarding Software Engineering has been taken from the book “Software Engineering - A Practitioner's Approach” by Roger S. Pressman [5].

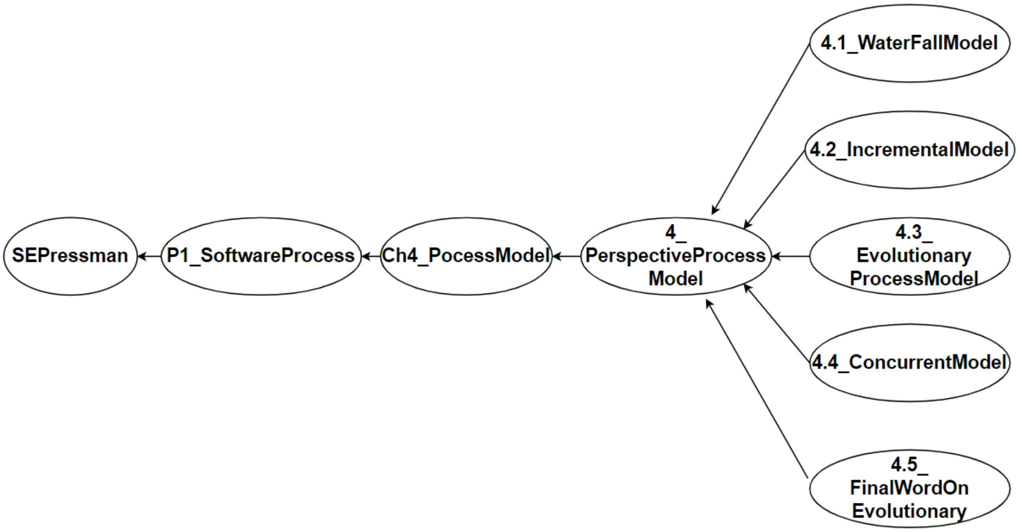


Figure 1. Ontology on Prescriptive Process Model

4 Tool Used

The formation of ontologies, proposed in this paper is done using Protege [6]. Protege enables the fluent formation of taxonomy structure by creation of individuals, properties, classes etc. Figure 2 depicts the steps followed for building ontology on Protégé.

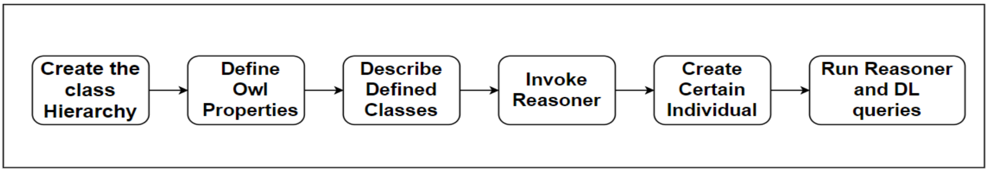

Create the class Hierarchy
Define Owl Properties
Describe Defined Classes
Invoke Reasoner
Create Certain Individual
Run Reasoner and DL queries

Figure 2. Ontology Building Process

5 Preprocessing the OWL (Web Ontology Language) file

Web Ontology Language (OWL) is a knowledge representation language used to describe ontology. OWL corresponds to descriptive logic. Most description logic are subset of First order logic but in difference to FOL, most description logic are decidable. Therefore, it’s possible to make logical deduction based on description logic i.e., to create new knowledge from existing knowledge. In OWL there is an open world assumption, absence of information must not be valued as negative information. Also in OWL, there is no unique name assumption, the difference between entities and classes must be expressed explicitly.

OWL files generated from Protege are parsed by using the inbuilt library of python called RDFLib. RDFLib is a pure python package for working with RDF (Resource Description Framework), which contains most of the things needed to work with RDF. The RDF framework is used to describe resources on the web. RDF allows us to make statements about resources. A statement has the following structure: " <Subject> <Predicate> <Object>" and is known as triples. These triples are the area of interest as they consist of all the information and knowledge of the domain.

6 Processing and comparing Ontologies

After getting the parsed OWL file, several NLP techniques are applied to get important entities for easier comparison. NLP techniques are used to extract concepts and relationships among these components. The syntactical analyzer for text, is used for identifying nouns, verbs, adjectives and adverbs and syntactical dependencies among them (subject of verb, object of verb, etc.). Several other NLP techniques are applied like removing stop words, stemming, word sense disambiguation.

In general, the concepts with the same name in ontologies, can vary in their actual meaning due to the different structure and arbitrariness. The ontologies comparison for similarity and dissimilarity can be based on the Super set concept (parent nodes), Sub set concept (child nodes), Intention and the instances set. Set of rules are defined to measure the semantic distance between the two entities. This comprises how many concepts are similar at what level of the tree structure and how the properties share common interest of the two entities. Considering another ontology on a similar topic: Figure 3. If the comparison is made with the previous ontology (Figure 1), Both the ontologies represent almost the same concepts but the concept names and the hierarchy are different. In this ontology, the waterfall model, it is at level 4 as compared to previous level 5. It has six more siblings while in previous ontology, it has only four siblings. Proceeding deeper to the definition and seeing how the definition and description of these two entities match. There is a need to look at the properties of the concepts and perform the comparison thoroughly. Table 1 gives description of above comparison:

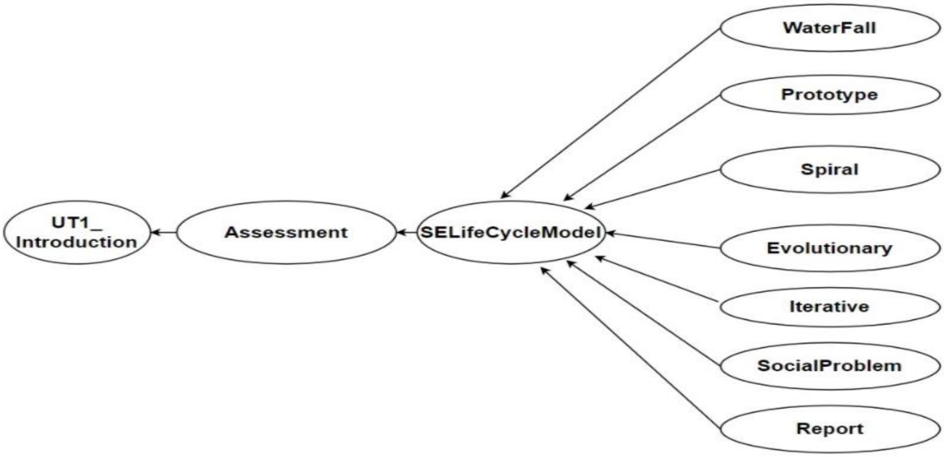


Figure 3. Ontology example for comparison

Table 1. Ontology Comparison

Parameter	Base Ontology	Example Ontology	Comments
Tree depth	5	4	Base ontology have more depth
Siblings	4	6	Example ontology focuses on specification
Parentnode	Prescriptive process model	SELifecycleModel	Base ontology covers the life cycle in several parts whereas example ontology covers all points under one topic.
Definition	Node definition matches but parent node definition not matches completely	Node definition matches but parent node definition not matches completely	Specific level nodes matches whereas at general level there is some dissimilarity.

7 System Architecture

The Figure 4 illustrates the system architecture of the ontology comparison system with the input and output of every step.

- In step 1 the gathered information is used to build ontology (OWL file) and is visualized using OWLViz.

- The OWL File is preprocessed and parsed by using the RDFLib library.
- NLP techniques are applied to get reduced and concise information
- Comparison and analysis is performed to get insights.

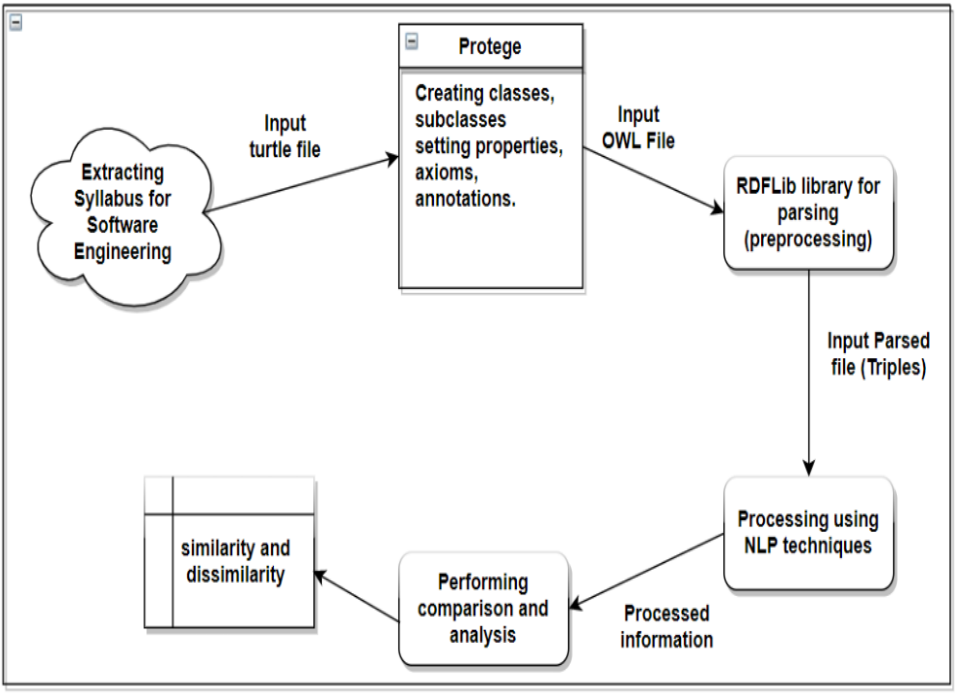
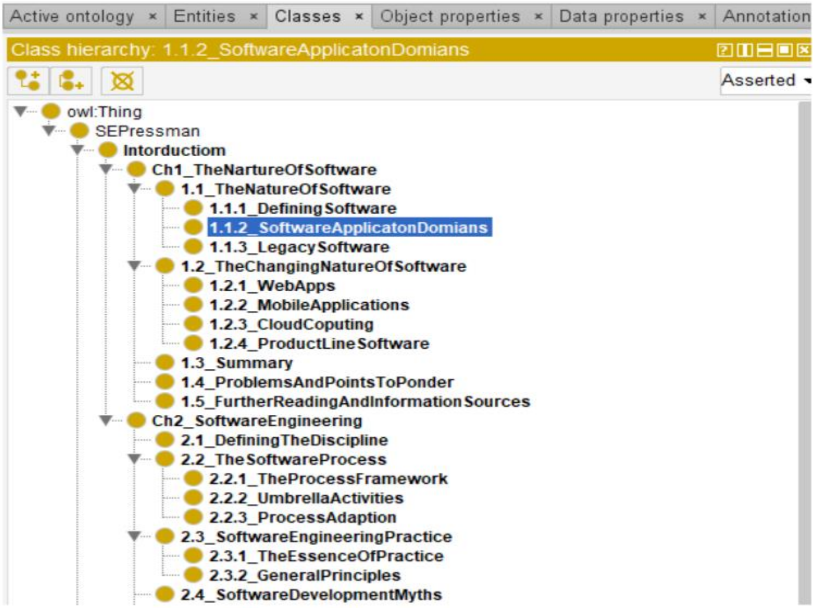

Extracting Syllabus for Software Engineering
Input turtle file
Protege
Creating classes, subclasses setting properties, axioms, annotations.
Input OWL File
RDFLib library for parsing (preprocessing)
Input Parsed file (Triples)
Processing using NLP techniques
Processed information
Performing comparison and analysis
similarity and dissimilarity

Figure 4. System Architecture

8 Results and Discussions

A hierarchical structure is prepared using the Protege tool for the structured representation of the syllabus along with the addition of several properties. The Figure 5 depicts the representation of the syllabus for the subject Software Engineering. The OWLViz module of the Protege is used to visualize the taxonomy structure. Once the owl file is ready, it is parsed, processed and finally comparison is performed.

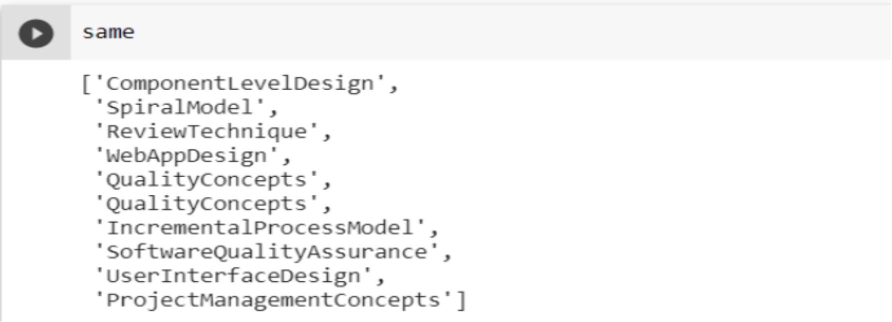
Figure 6 shows the glimpse of the comparison part. This comparison is performed on the basis of the subject and object (RDF statement). Further comparison of the classes along with their functional and data properties will result in deeper and through comparative study and analysis.



Class hierarchy: 1.1.2_SoftwareApplicatonDomians

```
owl:Thing
  SEPressman
    Intorductiom
      Ch1_TheNartureOfSoftware
        1.1_TheNatureOfSoftware
          1.1.1_Defining Software
          1.1.2_SoftwareApplicatonDomians
          1.1.3_Legacy Software
        1.2_TheChangingNatureOfSoftware
          1.2.1_WebApps
          1.2.2_MobileApplications
          1.2.3_CloudCoputing
          1.2.4_ProductLine Software
        1.3_Summary
        1.4_ProblemsAndPointsToPonder
        1.5_FurtherReadingAndInformation Sources
      Ch2_SoftwareEngineering
        2.1_DefiningTheDiscipline
        2.2_The SoftwareProcess
          2.2.1_TheProcessFramework
          2.2.2_UmbrellaActivities
          2.2.3_ProcessAdaption
        2.3_SoftwareEngineeringPractice
          2.3.1_TheEssenceOfPractice
          2.3.2_GeneralPrinciples
        2.4_SoftwareDevelopmentMyths
```

Figure 5. Taxonomy Structure



```
same
```

```
['ComponentLevelDesign',
 'SpiralModel',
 'ReviewTechnique',
 'WebAppDesign',
 'QualityConcepts',
 'QualityConcepts',
 'IncrementalProcessModel',
 'SoftwareQualityAssurance',
 'UserInterfaceDesign',
 'ProjectManagementConcepts']
```

Figure 6. Comparison output

9 Conclusions and Future Work

This paper shows an approach for building the ontology for a given domain and then comparing them to gain insights. The taxonomy structure is created by using OWL based tool Protege and further parsed for the comparison and analysis. The common concepts between the two ontologies are compared and analyzed but deeper comparison on the basis of predicates still needs to be done. Further work is also required to apply more NLP techniques to get concise and accurate text as well as towards the more thorough comparison.

References

[1] Wang C., Xiong M., Zhou Q., Yu Y. (2007) "PANTO: A Portable Natural Language Interface to Ontologies". In: Franconi E., Kifer M., May W. (eds) The Semantic Web: Research and Applications. ESWC 2007. Lecture Notes in Computer Science, vol 4519. Springer, Berlin, Heidelberg.

[2] Thomas R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing". In International Journal of Human-Computer Studies, Volume 43, Issues 5–6, 1995, pp. 907-928, ISSN 1071-5819.

[3] Philip Resnik, "Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language", Journal of Artificial Intelligence Research (JAIR), 11, pp. 95-130, 1999.

[4] Guarino, Nicola Oberle, Daniel Staab, Steffen. (2009). "What Is Ontology?". DOI:10.1007/978-3-540-92673-3_0.

[5] Harvard-18th ed., PRESSMAN, R. S. (1987). "Software engineering: a practitioner’s approach". New York, McGraw-Hill.

[6] John H Gennari, Mark A Musen, Ray W Fergerson, William E Grosso, Monica Crubézy, Henrik Eriksson, Natalya F Noy, Samson W Tu, "The evolution of Protégé: an environment for knowledge-based systems development", International Journal of Human-Computer Studies, Volume 58, Issue 1, 2003, pp.89-123, ISSN 1071-5819.