ISSN 2581 – 3781

AJCSE ASIAN JOURNAL OF COMPUTER SCIENCE ENGINEERING

Available Online at www.ajcse.info
Asian Journal of Computer Science Engineering 2018;3(2):19-25

RESEARCH ARTICLE

Constructing an Office Domain Ontology using Knowledge Engineering Process

Sameia Suha, S. R. Rashmi, Akshaj Jain, Krishnan Rangarajan, D. R. Ramesh Babu

Department of CSE, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India

Received: 15/01/2018; Revised: 28/02/2018; Accepted: 16/03/2018

ABSTRACT

Knowledge-based identification of human activities in systems depends primarily on rich contextual domain knowledge casing all of the information about the human, objects around human and also relations among them. Knowledge engineering plays an important role in building knowledge-based expert systems, to solve complex problems such as human activity recognition. This requires formal representation of the knowledge which is based on the conceptualization of the domain. Ontology is a widely chosen representational model that depicts knowledge as a set of concepts. In this work, we have applied knowledge engineering process for constructing the domain ontology of the office environment in agreement with the ontology development life cycle.

Key words: Domain knowledge, expert systems, ontology, reasoning

INTRODUCTION

An expert system is a software that uses experts' knowledge which is stored in the knowledge base to take decisions.[1] There are various domains in which expert systems can be applied. The first expert system was named Dendral which was used to analyze the chemical compounds. Expert systems are widely applied in the medical domain for improving the diagnosis. Other domains in which experts systems can be used are monitoring systems, process control systems, knowledge domain, and finance or commerce domain to name a few. The facts stored in the knowledge base of an expert system are the understanding of the domain expert in divergence with the facts of a book.[1]

Rule-based intelligent systems with human-curated rule sets have a huge disadvantage over the ontological-based intelligent systems as the rule-based intelligent systems have a tendency to omit the implicit meaning in statements or given rules. Whereas, the ontological-based intelligent systems always contain the implicit meaning in statements too, thus progressing way beyond the rule-based intelligent systems toward achieving full consciousness within the given system. Important control information can get lost while using rule-based intelligent systems while this is not the case with ontological intelligent systems, thus ontological intelligent systems are far superior to rule-based intelligent systems. For example, suppose we have two meanings of the same sentence, but one is explicit and another is implicit, the rule-based intelligent system will most probably take into account only the explicit meaning given by the sentence, whereas the ontological system will take into account the explicit meaning in the sentence.

Let's take, for example, a statement which says if x and y are > 0, so will its product. The rule-based system would just represent this statement and it would not add the formulation that if the product is >0, x > 0, so will y. The ontological type system would take into consideration both the first statement and the second statement and thus there is no loss of implicit information.[15]

Figure 1 shows the basic building blocks of an expert system and are discussed further down. The knowledge engineer consults the domain expert to build the knowledge base. The knowledge base and the inference engine are the cruxes of the expert system, where inference engine applies reasoning on the knowledge base to realize the inferences and provides it to the user.

Domain expert

A trained person who has expertise in solving problems of a specific domain. This person can

Address for correspondence:
Sameia Suha,
E-mail: sameia.suha@gmail.com

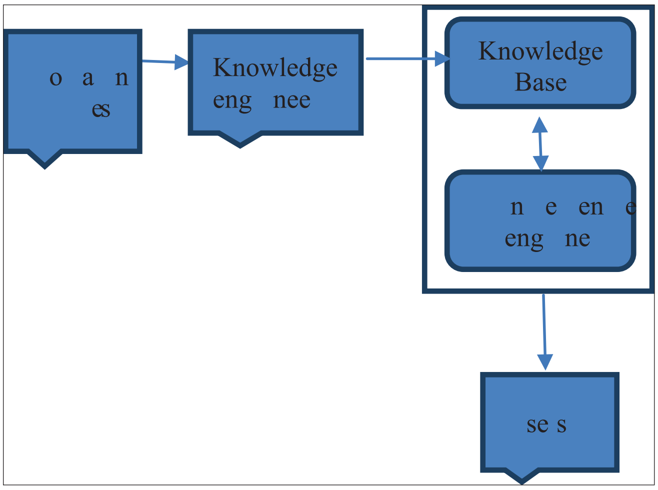

Figure 1: Block diagram of expert system

give his best results in the domain for the problem stated. The knowledge engineer takes input from the domain experts to build the system.[12] Although the experts do not need to understand the algorithms of the system, they play an important role in the development process by communicating their understanding and facts of the subject matter to the engineer, hence, giving an ample amount of time to the project. Domain expert is also involved in testing the specifications of the expert system.

Knowledge engineer

A knowledge engineer is someone who designs and builds the expert system. The main tasks of knowledge engineer are as follows:

- Discussion of the problem statement with the domain expert as to how the problem can be solved.
- To develop the reasoning methods, find relation between the objects and represent the facts in the knowledge base.
- Choose some development software or tool, for implementation of the knowledge represented.
- Testing, maintenance, and integrating the components of the expert system.

Knowledge base

Knowledge is a collection of facts and understanding cultured through experience and learning.[8] The knowledge base contains the knowledge of the domain as given by the expert. The expert systems are also known as knowledge-based system. Knowledge is something which is known or may be the result of understanding the subject through experience. Knowledge base is one which stores the knowledge of a particular domain or the complete system. Facts and rules are constructed into a knowledge base and used by expert systems to extract the conclusions.

There are various forms of knowledge representation. One such custom is to use ontology which provides a common understanding of the domain knowledge. Ontology is a way to explicitly represent the knowledge using classes and the relationships among classes. In ontology, new facts can be inferred from IF-THEN rules. If part lists the set of conditions. If the IF part is satisfied, then part is concluded.

For example: "If you are hungry, then eat."

Inference Engine

Inference engine fetches new facts from the knowledge base. The conclusions can be drawn through forward chaining which uses the rules or backward chaining inference method that works backward from the goals.

End user

End user is one who is in need of an expert system and uses the system for the desired operation. For example, a doctor may need an expert system which can assist the doctor in diagnosis.

The rest of the paper is divided into different sections as follows: Section 2 is about knowledge engineering process and sections 3 and 4 explain the steps involved in constructing the ontology and an example of office ontology, respectively, followed by the conclusion.

KNOWLEDGE ENGINEERING APPROACH: ONTOLOGY DEVELOPMENT LIFE CYCLE

Figure 2 shows the knowledge engineering process used for the construction of ontology. However, there is no unique ontology development life cycle for the construction of the ontology. The subsequent iterative steps can be considered as a conventional way of applying the knowledge engineering process to develop an ontology depending on the application.

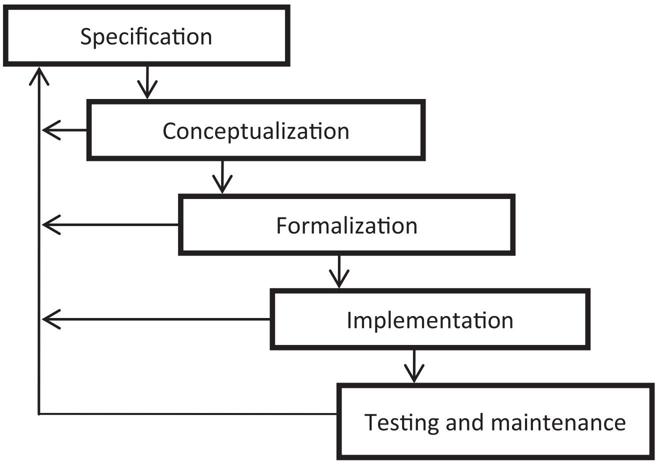

Figure 2: Ontology development life cycle

Specification

The initialization steps involve identification and specification of the problem statement for the ontology to be constructed. It also involves identifying the important characteristics of the problems, the resources used to solve the problem and the reasoning strategies to be used. This step is carried out by the domain expert in collaboration with the knowledge engineer.

Conceptualization

Concepts are the entity that exists in real world. Ontology is an explicit way of specifying the concepts. It structures the concepts and the relationship between the concepts in the domain of interest. Conceptualization is explicitly done by the knowledge engineer with the consultation of the domain expert.

Formalization

The knowledge engineer coverts the concepts into formal language. Web ontology language abbreviated as OWL is a language that provides axioms to define the concepts as a hierarchy of class and subclasses. This formal conceptualization also includes properties as relationship between the concepts and set of individuals belonging to the class along with the property of individual.[13]

Implementation

Implementation part refers to the realization of the actual specification listed by knowledge engineer and domain expert. Actual implementation tools are used here for the construction of ontology where the formalization is fed into the expert system shell by the engineer.

Testing and maintenance

The system is subjected to the process of testing by the domain expert to check for the specifications. The maintenance refers to the taking care of the product after the delivery. It includes correction of bugs and improves performance of the system along with other attributes. This is done by knowledge engineer.

PROCEDURE FOR IMPLEMENTATION OF DOMAIN ONTOLOGY

Ontology construction is conversion of knowledge into software representation. It provides a common way of understanding the knowledge and domain reusability. Once the concepts and the essential specification are obtained from the expert of the domain, analysis of domain knowledge is possible. Underneath traditional conditions, ontology creation itself is not the goal. Ontology can be analogous to the delineation of knowledge and its structure.

Simple ontology development method consists of:

1. Determine the aim, domain, and scope of the ontology
2. Reuse prevailing ontologies if appropriate
3. Input the domain expert's knowledge into the system
4. Label important terms in the ontology
5. Define classes of ontology along with their hierarchy
6. Define relationships or properties along with its domain and range
7. Creating instances

Step 1

The first step consists of determining the desired domain on which the ontology must be constructed. In the given example of the ontology below, the aim, domain, and scope are to recognize activities that take place in the office. The domain is selected by the knowledge engineer who is interested in creating the ontology. The necessary information required to construct the ontology is collected from the domain expert. This phase must decide the application that will use the ontology.

Step 2

One of the main advantages of ontology is reusability. Any existing ontology of the interested domain if possible can be reused. The existing ontologies for specific domains can be utilized by filtering and expanding the prevailing properties and classes. This reduces the time required to construct the ontology.

Step 3

The knowledge engineer should take the information from the trained expert and he should input this same information into the ontology. The domain expert here was the office manager and the knowledge engineer took the information of day-to-day activities from the manager and inserted the information into the ontology.

Step 4

List all significant terminologies and statements with the help of domain expert, ought not to consider the idea of overlapping relationships of concepts or terms, and even concept of class or property.

For example, the important terms for office domain include actions, events, person, and objects, among others.

Step 5

The next step is to determine and define ontology classes of the domain for the ontology construction. In OWL, we can expressly declare the resources to be a class by stating that it's of RDF:type owl:class. Syntactically, this amounts to exploitation of an owl:class component. It is accustomed OWL convention to name classes with singular nouns. It is additionally customary for the names to start out with uppercase and to use camel case for multiword names. The employment of uniform resource identifier to name classes and properties is a vital facet of the Semantic Web. The classes in the office ontology are as follows:

- Action
- Event
- Person
- Objects

Step 6

OWL outlines two varieties of properties:

i. Object property
ii. Data type property

Object properties express relationships between a pairs of class, whereas data type properties describe relationships between a class and a data value. Similar to classes, we can describe properties by including subcomponents. RDFs:subproperty of tells that a resource or object related with one property can also be related with another property. The rdfs: domain and rdfs:range state the domain and range of a property, which tells how a property links a subject with an object. The rdfs: domain of a property applies to the subject of any statement. Whereas, the rdfs: range of a property applies to the object of the statement. Although these properties could appear simple, they will result in variety of misunderstandings and should be used fastidiously.

Step 7

The individuals of a class are created in the instance tab by selecting the class and naming the individuals. Defining an individual instance of a class requires choosing a class, creating an individual instance of that class, and naming the instance.[14]

Once an ontology is created its consistency can be checked and reasoning can be applied. It is important to note that there can be different ontologies for the same domain, the design of ontology is dependent on the application it will be used for.[14] AQ2

Consistency Check

The knowledge of a domain consisting of the concepts and their roles covered in the ontology can be checked for consistency. A consistent ontology is the one which holds a set of conditions. An ontology can be checked for:

1. Structural consistency
2. Logical consistency
3. User-defined consistency.

An ontology is said to be consistent if it adheres to the syntax and the logics of the language used for the construction of the ontology and is built according to the user requirements.

Reasoning

A reasoner is the tool that infers logical consequences from a group of declared facts. One of the prominent ways to express the rules is through Semantic Web Rule Language which sits above the ontology web language. The JESS is used as the inference engine in the forthcoming example.

IMPLEMENTATION OF AN OFFICE DOMAIN ONTOLOGY

This section shows construction of office ontology and inferences of the events using the constructed ontology and SWRL rules.
First, the problem statement is to recognize the events carried out by person in an office environment. Hence, the domain in which the ontology is constructed is the office domain. New ontology is built; therefore, the reusability step mentioned above is omitted. After the domain is decided, the important terms related to the ontology are listed down and the classes and the subclasses are decided. Four classes and their subclasses along with an object property are shown in Figure 3.
The next phase in ontology construction is the design and implementation phase which can be carried out according to the following steps.
List the events which are the result of a sequence of actions that are carried out by a person. The events picked are as follows:

1. Enter.
2. Exit.
3. Print.
4. Sit.

The next step is to create the OWL classes defining actions, events, objects, and person [Figure 4].
The next step consists of defining the property along with domain and range. In this example, an object property named action is assigned its domain as person and events, its range as actions and objects. The domain and range tells in what way the property links the subject with the object [Figure 5].
Create instances for each class in individual tab. The instances are the individual of the classes.

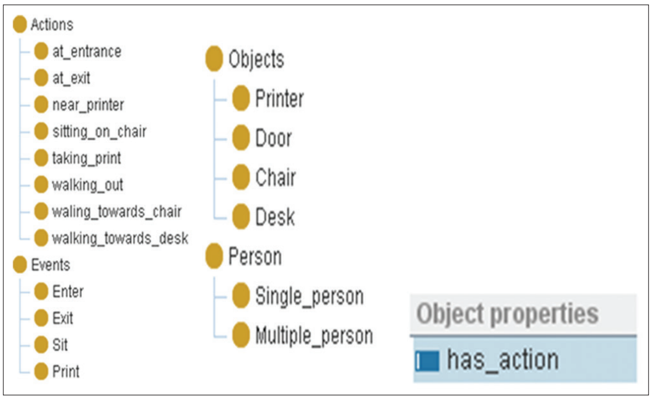

Figure 3: Classes and property of office ontology

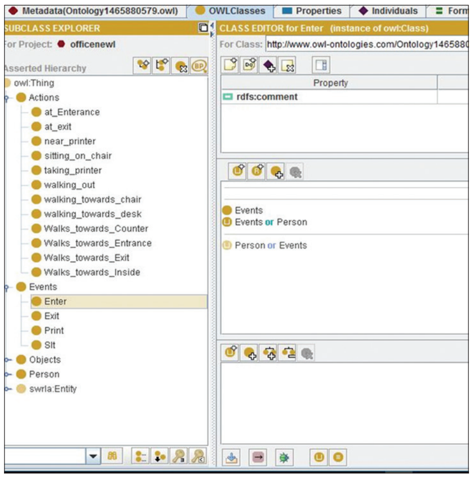

Figure 4: Creation of classes in protégé 3.4

The class for which an instance must be created is selected from the individual class and an instance is added by naming it [Figure 6].
The constructed ontology can be viewed in various forms including the nested view, tree view, nested composite view, domain-range view, and class tree view. Nested composite view of the office ontology using the jambalaya tool as a plugin of protégé is shown in Figure 7.
In Figure 8, the box represents the classes and the pink line is the relationship defined through the domain and range of the object property.
The consistency of the ontology is checked using the pellet direct reasoner. The results show that the ontology is consistent with respect to the concepts and inferred hierarchy.
Once the ontology is consistent, inference can be done to recognize the activities performed by the person in the office environment. This is done using the SWRL rules. SWRL tab must be activated and the rule is written in the multiline rule editor. The rules are fired using the JESS inference engine which converts the OWL and SWRL to JESS and back to OWL after the inference [Figure 9].

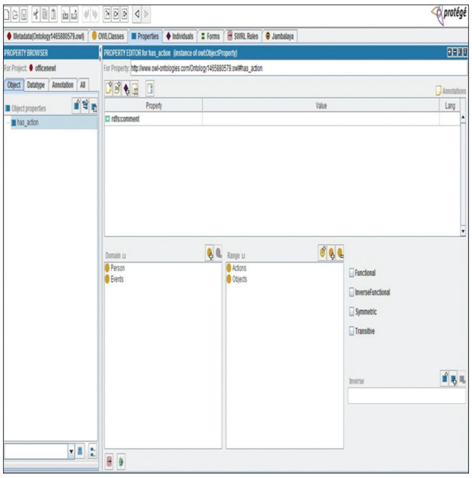

Figure 5: Property along with domain and range

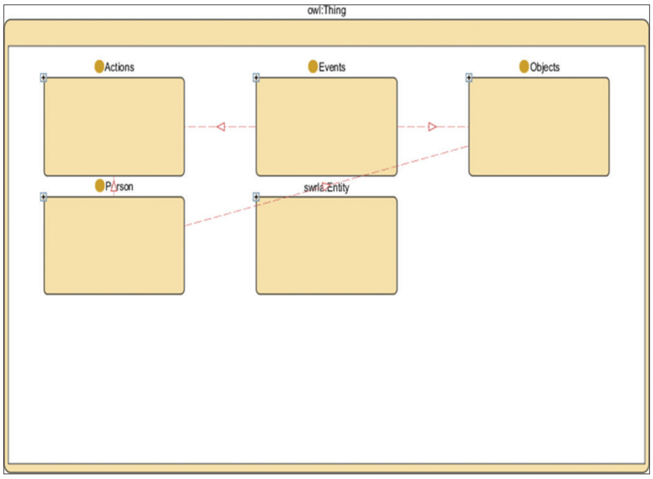

Figure 7: Nested composite view

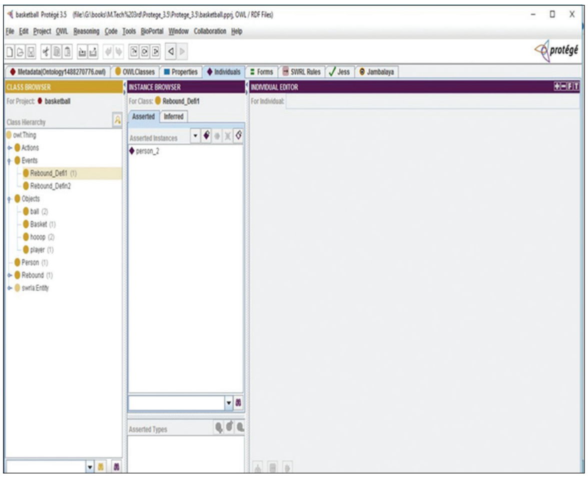

Figure 6: Creation of instance for class

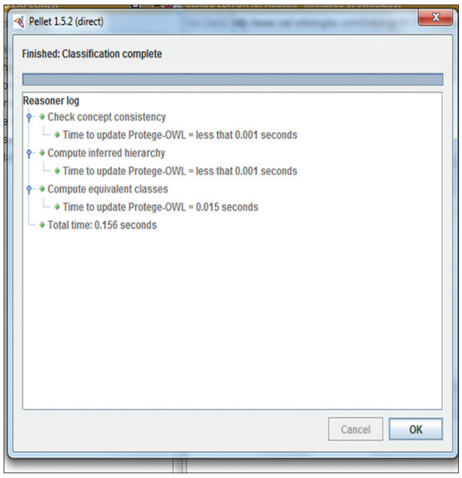

Figure 8: Consistency check results

To see the working:

1. Create instances of single person that have properties mentioned in rules.
2. Run the JESS tab.
3. New instances of the class single person will be added in the events classes [Figures 10 and 11].

The next phase is the testing and maintenance phase where the ontology is tested with various inputs to check the correct inference, further improvements to the ontology can be done during the maintenance phase. More events and activities along with the rules can be added as desired.

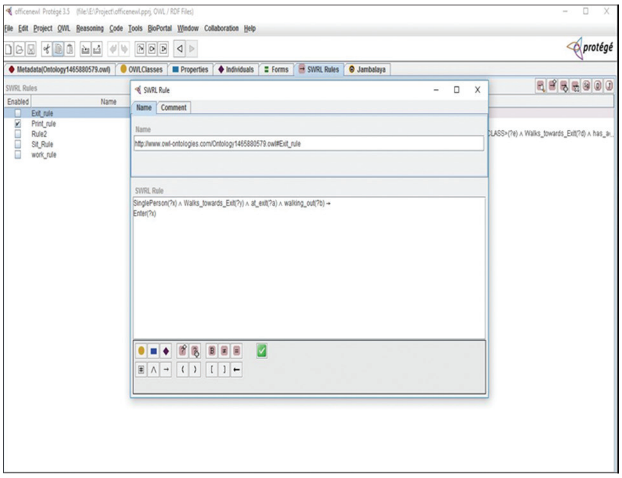

Figure 9: Multiline rule editor

CONCLUSION

In this paper, we discussed about the procedure for modeling ontologies for domain knowledge representation. We built the ontology for office domain using the tool Protégé 3.4 and used JESS engine for reasoning on the ontology. Knowledge engineering approach is thoroughly followed including the ontology life cycle for building this ontology, and it was tested for consistency. This ontology is reusable and expandable. This work will be helpful for the beginners in the ontology research.

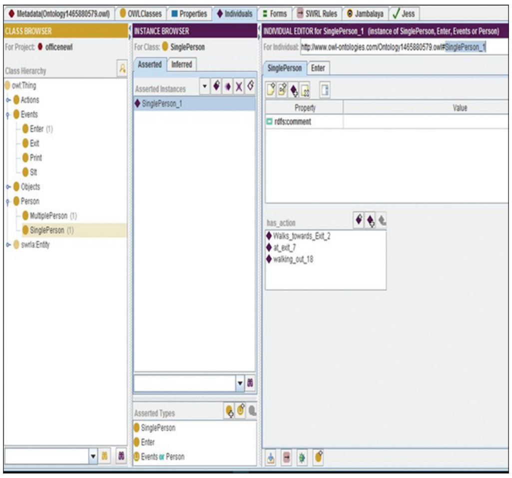

Figure 10: Input instances

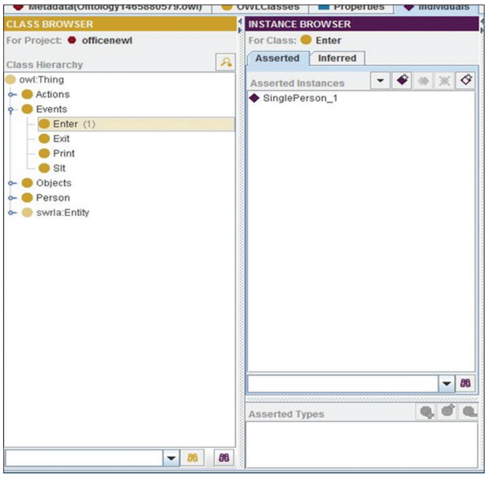

Figure 11: Inferred instance in the events: Enter subclass

AQ5 **Table 1:** Comparison of expert system and human expert

Resource parameters	Human expert	Expert system
Availability	On working days/retire	24*7
Geographical location	Specific location	Across globe
Replacement	Not replaceable	Can be replaced
Performance	Variable	Constant
Speed	Variable	Constant
Updation	Inconvenient	Easy to update

REFERENCES

AQ3

1. Rani S. Expert System of AI, 01 October 2014, Vol. 4, No. 5.
2. Xi FE, Jihua S. A Study in Knowledge Ontology Building in the Area of Knowledge Engineering. Third International Conference on Semantics, Knowledge and Grid; 2007.
3. Kureychik VM. Overview and Problem State of Ontology Models Development. Russia: Southern Federal University; 2017.
4. Ying-ying Y, Zong-yong L, Zhi-xue W. Domain Knowledge Consistency Checking for Ontology-based Requirement Engineering. China: ICCSSE; 2008.
5. Jin Z. Ontology-based requirements elicitation. J Computers 2000;23:486-92.
6. Li Z, Wang Z, Yang Y. Towards a Multiple Ontology Framework for Requirements Elicitation and Reuse. In: Proceedings of 31st IEEE International Computer Software and Application Conference, Beijing; 2007. p. 189-95.
7. Vel´asquez JD, Jain LC, editor. Advanced Techniques in Web Intelligence-1. Verlag: Springer; 2010.
8. Hyeon J, Oh K, Chung YJ, Kang BH, Choi HJ. Constructing an Initial Knowledge Base for Medical Domain Expert System using Induct RDR. Daejeon, Republic of Korea: School of Computing, KAIST; 2016.
9. Xia FE, Jihua S. Specifying Ontologies: Linguistic AspectsEngineering, Third International Conference on Semantics, Knowledge and Grid; 2017.
10. Source: Portal.surrey.ac.uk. AQ4
11. Wei Q. Development and Application of Knowledge Engineering Based on Ontology, 2010 Third International Conference on Knowledge Discovery and Data Mining; 01/2010.
12. Staab S, Studer R, Schnurr HP, Sure Y. Knowledge processes and ontologies. IEEE Intell Syst 2001;16:26-34.
13. Andrea M, Francesco R, Maurizio T, Salvatore M. Formalizing Knowledge by Ontologies: OWL and KIF. Pisa, Italy: CNR, IIT Department, Via Moruzzi 1, I-56124; 2015.
14. Noy NF, McGuinness DL. Ontology Development 101: A Guide to Creating YourFirst Ontology. Stanford, CA: Stanford University, 94305; 2001.
15. Nilsson NJ. Principles of Artificial Intelligence. Los Altos, CA: Morgan Kaufmann; 1980.

Author Queries???
AQ1:Kindly provide running title
AQ2:Kindly review the sentence.
AQ3:Kindly cite references 2-7, 9,10 and 11 in the text part
AQ4:Kindly provide complete reference details
AQ5:Kindly cite table 1 in the text part