

Development of University Ontology for aSPOCMS

Sanjay K. Dwivedi

Babasaheb Bhimrao Ambedkar University, Lucknow, India

Email: skd200@yahoo.com

Anand Kumar

Babasaheb Bhimrao Ambedkar University, Lucknow, India

Email: anand_smsvns@yahoo.co.in

Abstract— In these days, ontology is the most popular and widespread formalism of knowledge representation. With the requirement of our system aSPOCMS (An Agent-based Semantic Web for Paperless Office Content Management System), we present the construction of university ontology to retrieve the information and accomplish the process of files of various sections by using the predefined workflow in university ontology. The aSPOCMS aims to provide the facility to manage and process the files/documents of various departments and sections of higher educational institutions (i.e. universities) in paperless environment. This paper reveals the conceptualization of university knowledge through construction of university ontology. Generalized structure of Indian universities and workflow processes have been taken for ontology development by describing the class hierarchy, and demonstrate the graphical view of ontology. We also demonstrate the ability of university ontology to execute intelligent query to retrieve the information.

Index Terms— Semantic Web, Ontology, SemanticWorks tool, OWL, University Concepts

I. INTRODUCTION

Ontology is most popular technology for knowledge representation in Semantic Web. The major reason is that the applications require more knowledge sharing and reuse. Here, we discuss the essential steps in optimal ontology development process of university domain. These steps may be favorable to construct the ontology of other domain.

A. Semantic Web

The inventor of today's web, Tim Berners-Lee introduced the concept of Semantic Web [1]. In his vision of Semantic Web, content located on web should be available, processible and understood by both people and machines. It is the extension of current web in which information is given well defined meaning [2], which makes it easily processible by machine. The well defined and linked data on web can be used for common understanding, effective discovery and reuse of particular knowledge across various applications. There are three major technologies: Resources Description Framework

(RDF) [3][4][5], Resource Description Framework Schema (RDFS) [6][7] and Ontology Web Language (OWL) [8]. RDF is represented as triples statements which consist of a subject such as the resource, a predicate that is a property associated with resource and its object such as the value of the property. RDF Schema defines valid classes, properties for an unequivocal class, data type's properties, hierarchical relationships between classes or properties. OWL is a semantic markup language for sharing ontologies on the web and is designed for the use of software agents. OWL is used to describe the important concepts in a domain, essential properties of each concept and restrictions on properties such as property cardinality, property value type, domain and range of a property.

B. aSPOCMS System

The aSPOCMS (An Agent-based Semantic Web for Paperless Office Content Management System) [9] has been designed to provide the paperless environment to universities by processing electronic form of files or documents via predefined workflow of processes within university ontology. The varieties of information from different resources such as employees, departments, workflows and files of a typical university are involved in order to process the electronic form of files or documents.

The formalization of information represented in ontology can be easily interpreted by computer and the information resided in ontology can be processed on semantic level efficiently. Therefore, ontology is initiated in university domain to represent the facts and workflow.

The aSPOCMS is an agent-based Semantic Web system. It enables paperless office content management system that uses RDF, RDFS and OWL for metadata declaration and reasoning rules. The architecture of this system aSPOCMS is shown in figure 1. It has four major modules: communicator, access control, knowledge manager and reasoner. Communicator will provide the interface to users to communicate with the system. The access control has the capability to specify the authorizations over concepts defined in ontology. The user can annotate over concepts according to relationships, which are defined in ontology. The

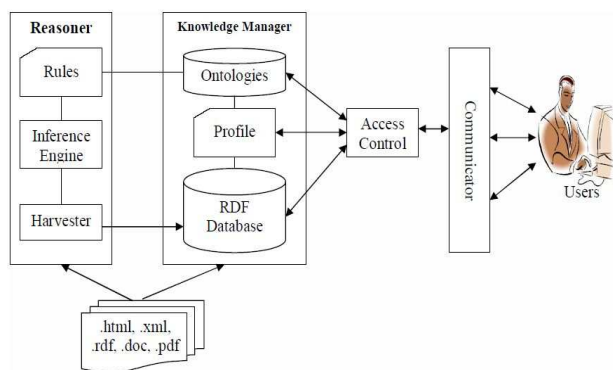


Figure 1. Architecture of aSPOCMS (adopted from [9]).

knowledge manager is the major component of the architecture as it will manage knowledge base of university and picking the knowledge by sorting, and structuring data according to the domain ontology. It has the RDF database, user's profile and ontologies. Ontology is the most prominent sub component of knowledge manager, which describes the conceptual terms and relations between these terms of university. The RDF database defines the metadata of conceptual terms of ontology.

The reasoner has the rules, an inference engine and harvester. The inference engine uses the rules to derive additional realistic knowledge from the university ontology. The harvester will harvest the additional knowledge in RDF triple format.

C. Ontology

The Semantic Web languages like OWL provide the facility to encode the ontology and an approach to integrate ontologies of multiple domains to support ontology sharing and mapping. The OWL language is divided into three syntax classes [10]. They are, OWL Lite, OWL DL and OWL Full in order of their increasing expressiveness. OWL Lite supports the users for a hierarchical classification of concepts and their simple constraint features. The advantage of this language is that it is easier to understand and implement than the other two; however, it restricts expressivity. OWL Lite has the lower formal complexity than OWL DL and OWL Full. OWL DL is the sublanguage of OWL Full which supports the maximum expressiveness without defeating the computational completeness. OWL DL is not fully compatible syntactically and semantically to RDF. The entire OWL languages are called as OWL Full in which all OWL language primitives use and allow combinations of the primitives in arbitrary ways with RDF and RDF Schema. This language comprises the possibility of changing the meaning of predefined primitives of RDF or OWL by applying the language primitives to each other. Therefore, OWL full formalism is used to construct the university ontology in this paper.

We used Altova SemanticWorks[11] to construct the ontology in OWL languages. The approach is depicted through creating university ontology based on student, employees, university structure, workflow of file processes and relationship between them.

II. SEMANTICWORKS: ONTOLOGY EDITOR TOOL

Altova SemanticWorks is a graphical RDF/OWL editor for building Semantic Web applications. It provides powerful, easy-to-use functionality for a visual creation and editing of RDF, RDF Schema (RDFS), OWL Lite, OWL DL, and OWL Full documents. There is no methodology associated to this tool. This editor is capable to manage the following files: N-triples, XML, RDF, RDFS and OWL. A screenshot of SemanticWorks with university ontology are viewed in figure 2. The figure shows some OWL class and their instances on university ontology.

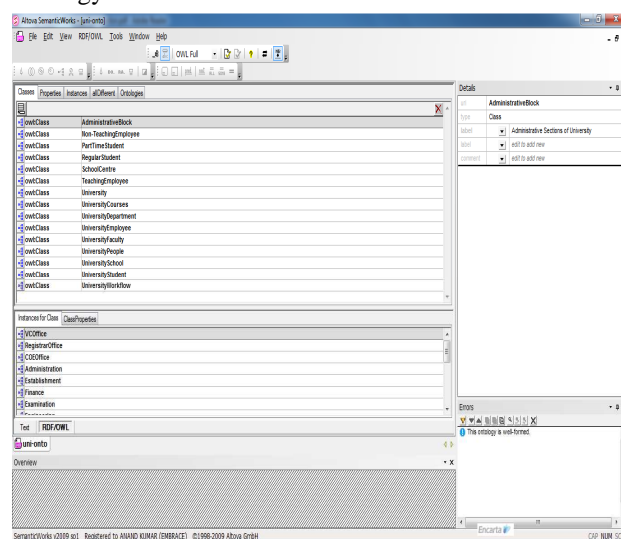


Figure 2. Note how the caption is centered in the column.

III. RELATED WORK

A number of researcher's efforts are carried in development of ontology using various domains and purpose. We found some researches being carried on ontology development of universities. In the following subsections, we focus some of the prominent researches on university ontology:

S. Lovrenčić et al. [12] described university studies ontology in Croatia domain modeling and presented the way of university domain knowledge representation for study purpose and have shown that description logic ALC is enough to fulfill this task. They have developed the ontology of university studies for study content. University studies ontology has super classes, subclasses, their individuals and properties of study content of university. Formal documents related to university studies are identified by this ontology. The study content is represented as hierarchical structure, which is able to show the entire educational content, the sequence of learning and the structure of educational concepts such as super and sub classes. The advantage of description logic representation formalism is to provide direct possibility for further development of Web ontology.

Naveen Malviya et al. [13] aimed to focus on information of university not for human consumption only but also made available to machine consumption. Ontology is used by ontology developers to concentrate on conceptual terms and created university ontology

using Protégé. The authors have taken the example of Rajiv Gandhi Technical University Bhopal (India), for the ontology development with various aspects like super class and subclass hierarchy, creating a subclass instances for class illustration and query retrieval process.

Paul Kogut et al. [14] discussed the recent convergence of UML and ontologies. The modeling of ontology information is described in class diagrams and Object Constraint Language (OCL) constraints.

Boyce, S., & Pahl, C. [15] presented a method for domain experts rather than ontology engineers to develop ontologies for use in the delivery of courseware content and focused in particular on relationship types that allow to model rich domains adequately.

In contrast to the current status of university ontology research, in this paper university ontology and its logic reasoning is designed on the basis of generalized structure of Indian universities and the workflow processes of files/documents.

IV. DEVELOPMENT OF UNIVERSITY ONTOLOGY

The developed university ontology here is in Indian perspective using OWL language and ontology editor tool. This ontology is the integration of four levels [16][17] i.e. top level ontology, domain ontology, task ontology and application ontology. We have taken the organizational structure of more than 10 Indian universities/institutes, 30 schools (or faculties) and 150 departments (or centers) for the purpose of ontology construction. The essential steps for development of university ontology with their implementation are described below:

A. STEP I: Classes and their hierarchy

Classes of the university are defined in this step. Hierarchy of class such as subclass and super class are also identified in this step. The major classes of university are represented in figure 3 and the hierarchical relationships between classes are represented in figure 4.

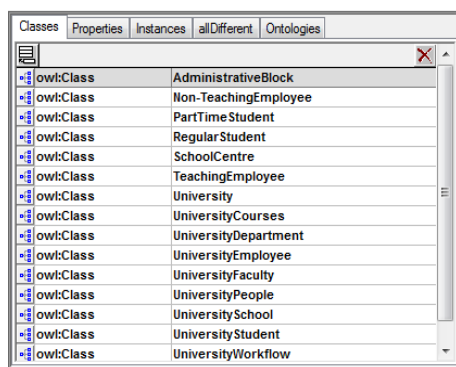


Figure 3. OWL Classes of University Ontology.

In figure 3, OWL class AdministrativeBlock has all sections related to administrative block of a university as resources. Non-TeachingEmployee has all the profile of employees of university. Similarly, remaining OWL classes have their relevant resources of the university. Figure 4 shows the graphical view of OWL classes and relationship among classes as

properties. The OWL class UniversityDepartment (or its equivalent class SchoolCentre) is the subclass of UniversityFaculty (or its equivalent class UniversitySchool) using the syntax `rdfs:subClassOf` and `owl:equivalentClass` properties respectively. Similarly, OWL class UniversityFaculty is the subclass of University and equivalent class of UniversitySchool with `rdfs:subClassOf` and `owl:equivalentClass` properties.

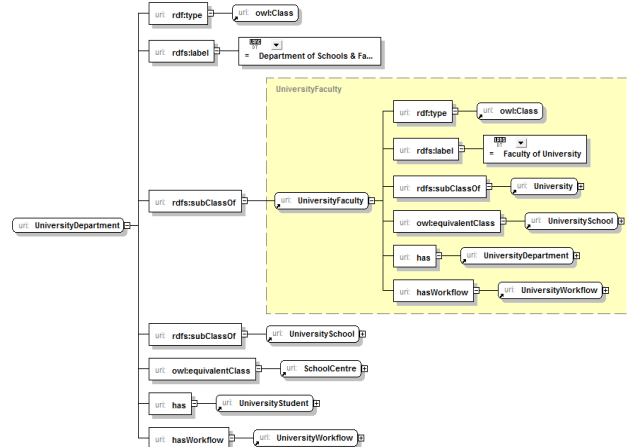


Figure 4. View of Hierarchical Relationships between Classes.

B. STEP II: Object properties of ontology

Object properties define the relationship between classes, through which we want to define among classes. These properties show the relationship between individual to individual. Some user defined object properties are shown in figure 5 which is used to

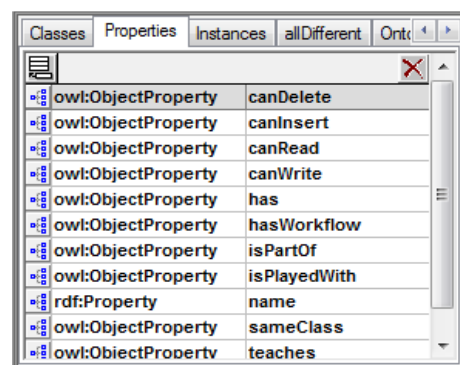


Figure 5. Defined Object Properties of University.

represent the relations between classes of university. The use of these properties is illustrated in figure 4. The OWL class UniversityDepartment is associated with UniversityStudent and UniversityWorkflow from the object properties 'has' and 'hasWorkflow' respectively.

C. STEP III: Metadata properties of ontology

Metadata properties show the relationship between individual and their data literal. In this step, we introduce the metadata properties of the resources of university

ontology. The metadata of a resource of the university is represented in figure 6.

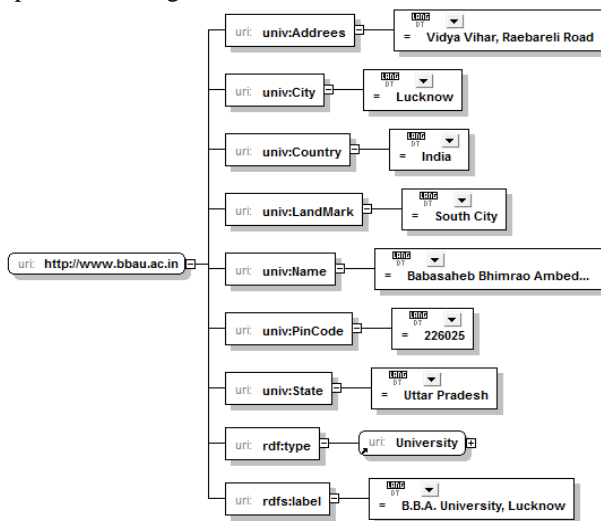


Figure 6. Metadata properties of the University.

The figure 6 represents the address of a university by using metadata properties, and the namespaces 'univ' [18] and 'univwf' [19] are used to define the workflow processes. The URI <http://bbau.ac.in> is represented a university and metadata of this resource and is described by metadata properties, which are shown in table I with literal values:

TABLE II.
METADATA PROPERTIES WITH LITERAL VALUES OF RESOURCE

Metadata Properties	Literal Values
univ:Address	Vidya Vihar, Raebareli Road
univ:City	Lucknow
univ:Country	India
univ:LandMark	South City
univ:Name	Babasaheb Bhimrao Ambedkar University
univ:PinCode	226025
univ:State	Uttar Pradesh

D. STEP IV: Property and Relationship

In order to define the link inside or between the classes, we use property to construct the relationship between individuals. The data properties are used to show the link between individuals to data type literal. The object properties also used to construct the relationship between individuals. The object properties domain and ranges can be defined as following spinet of XML code as examples.

```
<rdf:Description rdf:about="#sameClass">
  <rdf:type>
    <rdf:Description
      rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    </rdf:type>
  </rdfs:domain>
  <rdf:Description
    rdf:about="#SchoolCentre"/>
  </rdfs:domain>
  <rdfs:range>
```

```
<rdf:Description
  rdf:about="#UniversityDepartment"/>
</rdfs:range>
</rdf:Description>
```

In the above list of code, object property sameClass has the domain and range as OWL classes SchoolCentre and UniversityDepartment respectively.

E. STEP V: Axioms of Ontology

The relationship between attributes and individuals of class can be described by axioms. Four axioms of classes are used. These are, the existence of class, subclass, equivalent class and disjointwith, which are constructed using the OWL syntax `rdf:id`, `rdfs:subClassOf`, `owl:equivalentClass` and `owl:disjointwith` respectively. Some axioms of university ontology are shown in figure 7.

ID	Axiom Type & Attribute Type	Arguments
32	SubClassAxiom	LOADED
32	SUB-CLASS	#:SchoolCentre
32	SUPER-CLASS	#:UniversitySchool
33	SubClassAxiom	LOADED
33	SUB-CLASS	#:SchoolCentre
33	SUPER-CLASS	#:UniversityFaculty
34	EquivalentClassesAxiom	LOADED
34	DESCRIPTIONS	(#:SchoolCentre #:UniversityDepartment)
37	SubClassAxiom	LOADED
37	SUB-CLASS	#:UniversityDepartment
37	SUPER-CLASS	#:UniversityFaculty
38	SubClassAxiom	LOADED
38	SUB-CLASS	#:UniversityDepartment
38	SUPER-CLASS	#:UniversitySchool
39	EquivalentClassesAxiom	LOADED
39	DESCRIPTIONS	(#:UniversityDepartment #:SchoolCentre)
40	ObjectPropertyAssertionAxiom	LOADED
40	SUBJECT	#:UniversityDepartment
40	REL-OBJECT-PROPERTY	#:has
40	OBJECT	#:UniversityStudent
41	ObjectPropertyAssertionAxiom	LOADED
41	SUBJECT	#:UniversityDepartment
41	REL-OBJECT-PROPERTY	#:hasWorkflow
41	OBJECT	#:UniversityWorkflow
44	EquivalentClassesAxiom	LOADED
44	DESCRIPTIONS	(#:UniversitySchool #:UniversityFaculty)

Figure 7. Some axioms of university ontology.

Axioms for attributes: The relations between attributes are described by the axioms of attribute. The axioms of attributes are listed in table II.

TABLE I.
AXIOMS WITH THEIR SYNTAX

S. No.	Relations	Syntax
1.	Relation of inclusion	<code>rdfs:subPropertyof</code>
2.	Equivalent	<code>owl:equivalentProperty</code>
3.	Inverse	<code>owl:inverseOf</code>
5.	Limitation of Function	<code>owl:FunctionalProperty</code>
6.	Inverse Function	<code>owl:InverseFunctionalProperty</code>
7.	Relation of Symmetry	<code>owl:SymmetricProperty</code>
8.	Transitive	<code>owl:TransitiveProperty</code>

Axioms of Instances: OWL provides two types of axioms between instances. One of these is the composition of members and value of attributes, in which firstly we classify the information and then describe the composition of each class and the value of its attribute. The other axiom defines whether the two instances are equivalent (`owl:sameAs`) or not (`owl:differentFrom` and `owl:AllDifferent` etc). In our construction of university ontology, functional property is applied in object properties like teachers and students are associated with department with 'has' property. In other way, teachers and students can be associated with department with 'is' property such as:

“Department has teachers and students.”

“Teacher/student is the member of department.”

Here, we see 'is' property as the inverse of 'has' property.

F. STEP VI: The instance of ontology

After defining the efficient classes of ontology, we should select the relevant class to define the instances for the class. The `rdf:type` syntax is used to state the class of instances. we noted that more than one class can be associated with same instances and many instances can be associated with a particular class. The instances of various OWL classes of university ontology are defined in figure 8. Each instance can belong to many classes or same instance can belong to many classes.

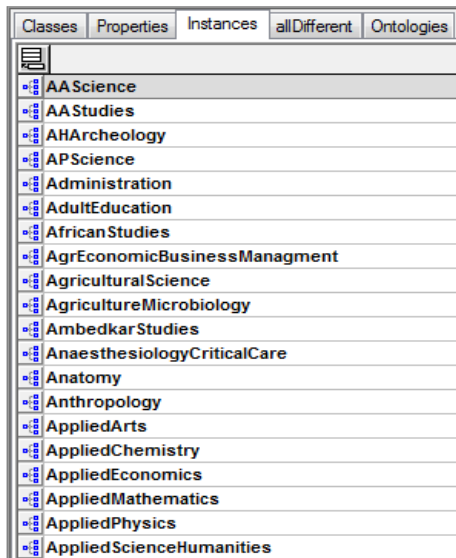


Figure 8. Instances of OWL classes of University.

G. STEP VII: Reasoning of Ontology

The reasoning is the most important part of ontology. Ontology reasoner i.e. RacerPro checks the consistency and correctness of concepts, roles, individuals, assertions, axioms, taxonomy, role hierarchy, query etc. and also find the logical contradictions implicitly described in university ontology, which are useful to cross check the various concepts and relationships of constructed ontology. We put forward the reasoning mechanism of university ontology in section 5.

V. REASONING MECHANISM OF UNIVERSITY ONTOLOGY

In this research, we grouped the reasoning mechanism into two categories. One, the ontology reasoning using description logic and the other is user-defined reasoning using first-order logic. Both categories are used to construct the university ontology by using OWL language.

A. Ontology Reasoning

Description logic (DL) allows us to specify a terminological hierarchy using a restricted set of first-order formulas [20]. To fulfill the important logical requirements, the equivalence of OWL description logic permits to exploit the considerable descriptive logic reasoning. These requirements associated with satisfiable of concept, including of OWL classes, class consistency and checking of instances. A partial set of reasoning rules that support OWL Full has been used to construct the university ontology, which are represented in table III.

TABLE III.
OWL REASONING RULES

Property	Reasoning Rule
TransitiveProperty	$(?P \text{ rdf:type owl:TransitiveProperty}) \wedge (?X ?P ?Y) \wedge (?Y ?P ?Z) \Rightarrow (?X ?P ?Z)$
subClassOf	$(?X \text{ rdfs:subClassOf } ?Y) \wedge (?Y \text{ rdfs:subClassOf } ?Z) \Rightarrow (?X \text{ rdfs:subClassOf } ?Z)$
subPropertyOf	$(?X \text{ rdfs:subPropertyOf } ?Y) \wedge (?Y \text{ rdfs:subPropertyOf } ?Z) \Rightarrow (?X \text{ rdfs:subPropertyOf } ?Z)$
disjointWith	$(?X \text{ owl:disjointWith } ?Y) \wedge (?A \text{ rdf:type } ?X) \wedge (?B \text{ rdf:type } ?Y) \Rightarrow (?A \text{ owl:differentFrom } ?B)$
inverseOf	$(?X \text{ owl:inverseOf } ?Y) \wedge (?A \text{ rdf:type } ?X) \wedge (?B \text{ rdf:type } ?Y) \Rightarrow (?A \text{ owl:disjointWith } ?B)$
unionOf	$(?X \text{ owl:unionOf } ?Y) \Rightarrow (?A \text{ rdf:type } ?X) \vee (?B \text{ rdf:type } ?Y)$
intersectionOf	$(?X \text{ owl:intersectionOf } ?Y) \wedge (?A \text{ rdf:type } ?X) \Rightarrow (?B \text{ rdf:type } ?Y)$
equivalentClass	$(?X \text{ owl:equivalentClass } ?Y) \wedge (?A \text{ rdf:type } ?X) \wedge (?B \text{ rdf:type } ?Y) \Rightarrow (?A \text{ rdfs:subClassOf } ?Y) \wedge (?B \text{ rdfs:subClassOf } ?X)$

X, Y and Z are three OWL classes, A, and B are the instances of classes of any structure of university. We used these reasoning rules for our system. The examples of some properties are described below:

For `rdfs:subClassOf` property:

$?X \rightarrow \text{UniversityDepartment}$

$?Y \rightarrow \text{UniversitySchool}$

$?Z \rightarrow \text{University}$

Reasoning Description: UniversityDepartment is the subclass of UniversitySchool and UniversitySchool is the subclass of University then UniversityDepartment is the subclass of University.

For owl:disjointWith property:

?X \rightarrow MathematicalScience is the instance of UniversitySchool class.

?Y \rightarrow MedicalScience is the instance of UniversitySchool class.

?A \rightarrow OperationalResearch is the instance of MathematicalScience.

?B \rightarrow ForensicMedicine is the instance of MedicalScience.

Reasoning Description: If MathematicalScience is disjoint with MedicalScience and OperationalResearch is the RDF type of MathematicalScience and ForensicMedicine is the RDF type of MedicalScience then OperationalResearch is different form of ForensicMedicine.

For owl:unionOf property:

?X \rightarrow UniversityDepartment

?Y \rightarrow UniversitySchool

?A \rightarrow MedicalScience is the instance of UniversityDepartment.

Reasoning Description: UniversityDepartment is the union of UniversitySchool then MedicalScience is RDF type of UniversityDepartment or UniversitySchool.

B. User-defined Reasoning

User-defined properties provide the more flexible reasoning mechanism, which is an enormous range of high-level reasoning within the entailment of first-order logic. The user defined properties are used to construct the relations between OWL classes of our university ontology, which also established the relational link with instances of OWL classes. Typical example of user defined properties and their reasoning rules are listed in table IV.

TABLE IV.
USER DEFINED REASONING RULES

User Defined Property	Reasoning Rule
has	$(?X \text{ has } ?Y) \wedge (?Y \text{ has } ?Z) \Rightarrow (?X \text{ has } ?Z)$
sameClass	$(?X \text{ sameClass } ?Y) \wedge (?A \text{ rdf:type } ?X) \wedge (?B \text{ rdf:type } ?Y) \Rightarrow (?X \text{ owl:equivalentClass } ?Y)$
teaches	$(?D \text{ hasfaculty } ?X) \wedge (?D \text{ hasStudent } ?Y) \Rightarrow (?X \text{ teaches } ?Y)$

The implementation of these properties with OWL classes is shown in figure 9.

X, Y and Z are three OWL classes, A, and D are the instances of classes of any structure of university. We used these reasoning rules for our system. Some examples of properties are shown below:

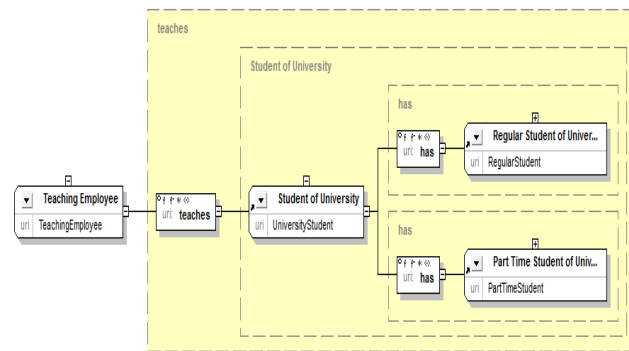


Figure 9. Implementation of User-defined Properties.

For 'has' property:

?X \rightarrow University

?Y \rightarrow UniversitySchool

?Z \rightarrow UniversityDepartment

Reasoning Description: University has UniversitySchool and UniversitySchool has UniversityDepartment then University has UniversityDepartment.

For 'sameClass' property:

?X \rightarrow UniversityFaculty

?Y \rightarrow UniversitySchool

?A \rightarrow MedicalScience is the instance of UniversityFaculty.

Reasoning Description: In this case, if UniversityFaculty is same class as UniversitySchool and MedicalScience is the RDF type of UniversityFaculty and UniversitySchool then both classes are equivalent class.

For 'teaches' property:

?X \rightarrow TeachingEmployee

?Y \rightarrow UniversityStudent

?D \rightarrow UniversityDepartment

Reasoning Description: If UniversityDepartment has faculty TeachingEmployee and has student UniversityStudent then TeachingEmployee teaches the UniversityStudent.

VI. RESULTS

We represent the results of ontology, which can show the correctness and consistency of constructed university ontology. For this, graphical views of various ontologies have been created (SemanticWorks) as shown in this section. Further, some DL rules have also been applied to extract the information from ontology. The hierarchy of OWL classes, subclasses and their relationship is shown in the graphical view of ontology. Metadata of these classes are also depicted in visualization of ontology.

A. Visualisation View

Here, we add some important classes of concepts and some important subclasses of generalized structure of universities. Visual view of some classes and their subclasses results using ontology editor tool is shown in figure 10. Asserted view depicts the classes, subclasses and their relationship, which we defined in the construction of university ontology.

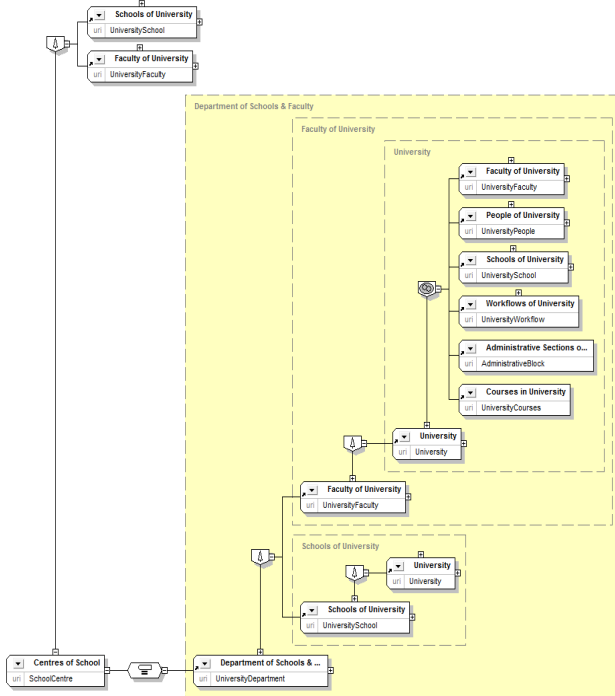


Figure 11. Graphical view of classes, subclasses and relationship.

In the figure, SchoolCentre is the equivalent class of UniversityDepartment and subclass of UniversitySchool and UniversityFaculty. The UniversityDepartment class is the subclass of UniversityFaculty and UniversitySchool classes, which are the subclass of University class. Furthermore, the University class is the union of (or has the members) following classes: UniversityFaculty, UniversityPeople, UniversitySchool, UniversityWorkflow, AdministrativeBlock and UniversityCourses.

Figure 11 presents the metadata of various OWL classes defined in university ontology. The relations of OWL class UniversityPeople is shown in table V.

TABLE V.
RELATIONS AND THEIR VALUES

Relations	Relational Value (Metadata)
rdf:type	Owl:class
rdfs:label	People of university
rdfs:subClassOf	University
has	UniversityEmployee, UniversityStudent class

Similarly, we can see the some other metadata of the

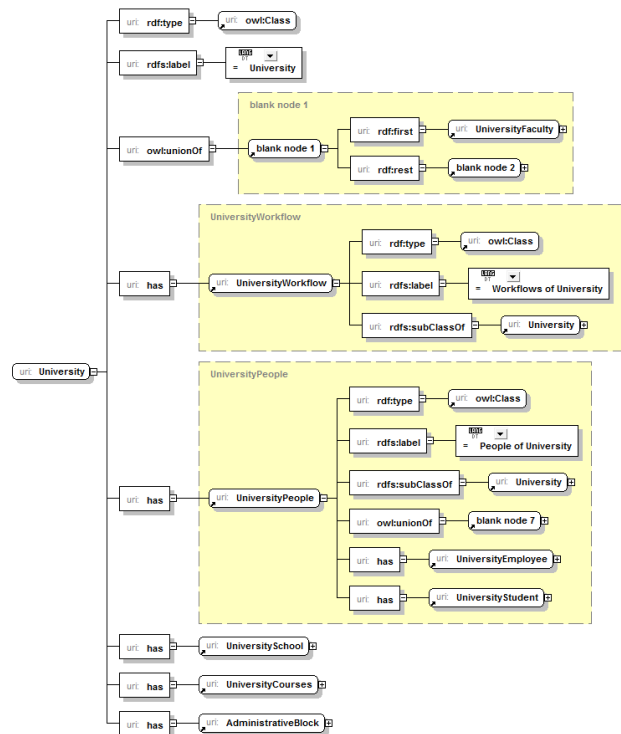


Figure 10. OWL classes and their metadata. concepts in figure 11.

In these graphical views, we show that the classes, subclasses and their relationship, ontology reasoning, user-defined reasoning and metadata of classes etc. of university are similar to predefined organizational structure and information, which are represented in the construction of ontology. We can clearly see various concepts and their metadata of the universities, which can help to rectify the unusual concepts, relations and metadata etc.

B. Results of Query Retrieval

The DL query is used to retrieve any information about the concept of university from designed university ontology. The purpose of these queries is intended to check the correctness of concepts and information, which are designed at the time of construction of ontology.

We can provide class or any property name correctly to retrieve the information and reasoner will display related information about particular class or property. For example, if we want to retrieve various sections of administrative block of a university, then we must enter the class name correctly (upper or lower case) as created in the ontology construction. The results of some of the input queries we tested on university ontology using RacerPro [21] inference engine are listed in table VI with the description of expected results and actual results as verification. The results are as per our expectations (as shown in column 2 & 3 of table VI).

VII. CONCLUSION

The workflow processes of university domain are a challenging task for knowledge representation and

TABLE VI.
QUERY AND THEIR RESULTS BY USING RACERPRO REASONER

Input Query	Description of Expected Result	Actual Result
? (retrieve (?X) (?X #!:AdministrativeBlock) :abox file://E:/University%20Ontology/uni-onto.owl) [retrieve all subclasses of OWL class 'AdministrativeBlock']	InformationBureau, Legal, Estate, Planning, SC-STCell, PPPCell, ProctorialBoard, SportBoard, ComputerCentre, Hospital, Library, Engineering, Examination, Finance, Establishment, Administration, COEOffice, RegistrarOffice, VCOffice. [all the section of Administrative Block]	((?X #!:InformationBureau))((?X #!:Legal))((?X #!:Estate)) ((?X #!:Planning))((?X #!:SC-STCell))((?X #!:PPPCell)) ((?X #!:ProctorialBoard))((?X #!:SportBoard)) ((?X #!:ComputerCentre))((?X #!:Hospital))((?X #!:Library)) ((?X #!:Engineering))((?X #!:Examination))((?X #!:Finance)) ((?X #!:Establishment))((?X #!:Administration)) ((?X #!:COEOffice))((?X #!:RegistrarOffice)) ((?X #!:VCOffice)))
? (individual-types #!:PhysicalEducation file://E:/University%20Ontology/uni-onto.owl) [describe the types of individual 'PhysicalEducation']	UniversityDepartment, SchoolCentre, UniversityFaculty, UniversitySchool, University. [which types of the Physical Education Department]	((#!:UniversityDepartment #!:SchoolCentre) (#!:UniversityFaculty #!:UniversitySchool) (#!:University) (*top* top))
? (describe-individual #!:Anthropology file://E:/University%20Ontology/uni- onto.owl) [describe the individual 'Anthropology' of university ontology]	Anthropology is the individual of university ontology as a department of university. This query describes about Anthropology Department.	(#!:Anthropology :assertions (#!:Anthropology #!:UniversityDepartment)) :role-fillers nil :told-attribute-fillers nil :told-datatype-fillers (#!:rdfs:label ("Department of Anthropology")) :annotation-datatype-property-fillers (#!:rdfs:label ("Department of Anthropology")) :annotation-property-fillers nil :direct-types :to-be-computed
? (describe-concept #!:UniversityDepartment file://E:/University%20Ontology/uni- onto.owl) [describe the concept 'UniversityDepartment' of university]	This query provides the description of UniversityDepartment concept. The synonyms of this concept are SchoolCentre and parents are UniversityFaculty and UniversitySchool.	(#!:UniversityDepartment :told-primitive-definition (and #!:UniversitySchool (and #!:UniversityFaculty (and #!:UniversityFaculty (and #!:UniversitySchool (and #!:UniversityDepartment #!:SchoolCentre)))) :synonyms (#!:UniversityDepartment #!:SchoolCentre) :parents (#!:UniversityFaculty #!:UniversitySchool)) :children (*bottom* bottom)))

ontology development. This paper focused a way of university domain knowledge representation and ontology development, which makes such type of information like machine understandable format. The Semantic Web technologies fulfills the requirement of this work, where the represented information is understandable by machine and cooperate to human users for efficient result on intelligently described information. Some essential steps are described to construct and elaborate the reasoning mechanism of university ontology. In the reasoning mechanism, the reasoning is carried out according to the connotative relationships between concepts and shows the result according to DL rule. Altova SemanticWorks tool is used to create and

edit the university ontology in visualized format. The ontology and user-defined reasoning mechanism of concepts, individuals, axioms and assertions have been elaborated and tested by inference engine.

REFERENCES

- [1] Berners-Lee, T., "Semantic Web Road Map", W3C Design Issues. [Online]. Available: <http://www.w3.org/DesignIssues/Semantic.html>.
- [2] Tim Berners-Lee, James Hendler and Ora Lassila. "The Semantic Web", Scientific American May 2001.
- [3] Resource Description Framework. <http://www.w3.org/RDF/>.

- [4] W3C, "RDF Primer", W3C Recommendation, Retrieved from, <http://www.w3.org/TR/2004/REC-rdf-primer>, February 2004.
- [5] O. Lassila and R. R. Swick (editors). Resource description framework (rdf) model and syntax specification. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>, Feb, 1999.
- [6] D. Brickley and R.V. Guha (editors). Resource description framework (rdf) schema specification 1.0. <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>, March, 2000.
- [7] W3C. RDF Schema Specification. <http://www.w3.org/TR/PR-rdf-schema/>, 1999.
- [8] OWL Web Ontology Language Reference. <http://www.w3.org/TR/owl-ref/>.
- [9] Sanjay K. Dwivedi and Anand Kumar, "An Agent-based Semantic Web for Paperless Office Content Management System", ICDM 2010: Proceedings of the 2010 Third International Conference on Data Management, pp. 352-360, Ghaziabad, India, March 2010.
- [10] OWL: Web Ontology Language Overview. <http://www.w3.org/TR/owl-features>, 10 Feb 2004.
- [11] SemanticWorks Semantic Web tool - Visual RDF and OWL editor, <http://www.altova.com/semanticworks.html>.
- [12] S. Lovrenčić, Mirko Cubrilo. "University Studies Ontology – Domain Modeling", 11th International Conference on Intelligent Engineering Systems 2007. IEEE Xplore, pp. 55-58.
- [13] Naveen Malviya, Nishchol Mishra and Santosh Sahu. "Developing University Ontology using Protégé OWL Tool: Process and Reasoning", International Journal of Scientific & Engineering Research, Volume 2, Issue 9, September 2011, pp. 1-8.
- [14] P. Kogut, S. Cranefield, L. Hart, M. Dutra, K. Baclawski, M. Kokar, J. Smith, UML for Ontology development, Knowledge Engineering Review, 2002.
- [15] Boyce, S., & Pahl, C. (2007). Developing Domain Ontologies for Course Content. Educational Technology & Society, 10 (3), 275-288.
- [16] Kumar, A., Dwivedi S. K. "Ontology Exemplification for aSPOCMS in the Semantic Web", World Congress on Information and Communication Technologies (WICT) 2011. IEEE Xplore, pp. 473-478.
- [17] Sanjay K. Dwivedi, Anand Kumar. "Ontology Exemplification and Modeling for aSPOCMS in the Semantic Web", International Journal of Computer Information Systems and Industrial Management Applications. MIR Labs USA, Volume 5 (2013) pp. 542-549.
- [18] Sharmin Rashid Linta, Md. Maidul Islam & Md. Rakibul Islam (2012). "An Enhanced Model of E-Learning Management System Using Semantic Web Technology and Development of Universal Namespace for University Domain", International Journal of Computer Science Issues, Vol. 9, issue 2, No. 2, March 2012.
- [19] Sanjay K. Dwivedi and Anand Kumar (2013). "Development of Universal Namespace for Workflow of University Domain for aSPOCMS", International Journal of Web & Semantic Technology (IJWesT) Vol.4, No.1, January 2013, pp. 1-17.
- [20] Zhang, D.Q.; Gu, T.; Pung, H.K.; "Ontology Based Context Modeling and Reasoning using OWL", Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, pp. 18-22, Singapore, March 2004.
- [21] RacerPro Version 2.0, Retrieved from, <http://www.racersystems.com/products/download/index.php> tml.