International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 5 (2013) pp. 542-549

Ontology Exemplification and Modeling for aSPOCMS in the Semantic Web

Sanjay K. Dwivedi[1] and Anand Kumar[2]

[1] Department of Computer Science, BBA University,
Vidya Vihar, Raebareli Road, Lucknow-226025, India
skd200@yahoo.com

[2] Department of Computer Science, BBA University,
Vidya Vihar, Raebareli Road, Lucknow-226025, India
anand_smsvns@yahoo.co.in

***Abstract*: Ontology can not only describe the workflow data of the any university more easily understood by computers in semantic encoding scenario, but can also integrate users data from different sources and in different forms for reasoning. Workflow technology promises to increase the efficiency in the execution of workflow based processes. The administrative workflow data of a typical higher learning institution (such as university) can be described by web ontology to increase the efficiency of the execution of workflow based processes in paperless environment. The ontology modeling for administrative workflow processes can be very useful in increasing the efficiency of workflow in the organizations. In this paper, we describe the possible model of university ontology having different levels e.g. top-level ontology, domain ontology, task ontology and application ontology with their suitable exemplification of our system called aSPOCMS (Agent-based Semantic Web for Paperless Office Content Management System) in paperless environment with Semantic Web perspective. We also discussed that how the Semantic Web resource description framework and language can be utilized for paperless office content management system of linked structures from distributed metadata.**

Keywords: Workflow of Administration, Semantic Web, Reasoning Rules, Ontology**.**

I. Introduction

Paperless environment [1] [2] involves electronic documentation as a data processing form, a word processing document, a digital image, and so forth, and submitting or uploading it directly as the claim file either in its original form or in the printed form. However, such paperless office content management systems not enabled with Semantic Web [3] [4] may increase the time for scanning the document and mailing to the particular department [5]. The use of agent based Semantic Web aims to improve the complexity of paperless office content management system and speeding up file retrieval and reduces clutter of the file.

The aSPOCMS (Agent-based Semantic Web for Paperless Office Content Management System) [6] is proposed for managing the files and documents of a typical administrative office of higher educational institutions. It has four sub-sections: communicator, access control, knowledge manager, and reasoner. The knowledge manager has the ontologies and RDF of various resources. In the paperless environment of Office Content Management System, various types of information from different sources are involved in respect to workflow of administrative processes of university with Semantic Web perspectives.

There are several Semantic Web Technologies [7] which provide us with supported tools for describing and annotating resources on the web in standardized ways. These technologies are mainly: Resource Description Framework (RDF) [8] [9], RDF Schema (RDFS) [10] [11] and Ontology Web Language (OWL) [12] [13] and its binding to XML (eXtensible Markup Language) [14]. The RDF provides a framework to encode semantically annotated information and it is a list of triples with resource-property-value. RDFS defines valid classes, properties for a specific class, data types of a property and hierarchical relationship between classes or properties. Finally, OWL is a logical language that define constrains and possible interpretation of terms used to annotate the information namely ontology. The ontology plays immense role as a dictionary for defining vocabulary of Terms for creating RDF documents in a specific domain and organizing hierarchical relationship between Terms. Basically, the university ontology will be made by OWL rules based on description logics. The description logic is a family of knowledge representation language which can be used to represent the terminological knowledge of an application domain in a structure and formally well understandable way.

We describe distinguish levels of university ontology: First, we can describe highly general ontologies of the university based on traditional philosophy (conceptualization) and used to analyze the information systems, which is called as top-level ontology. Second, we can describe the ontologies which restricted to special concept and special domain such as school, department and employee etc. This ontology is known as domain ontology. Third, we can describe the concept related to special task and action, which is called task ontology. Finally, the application ontology is depicting the concept that depends on special task and domain.

In this context, assume a situation that a user is writing an electronic complaint file (e.g. document related to financial support to a department/section of any higher educational institute and university) for the particular department in paperless environment. The claim file is send to particular section with authorized user. The aSPOCMS can automatically transmit the electronic document among participants under predefined workflow. Especially in the paperless environment, it is very important to overcome the one-size-fits-some approach and grant to competent authority with their individual experience. In one-size-fits-some approach the electronic complain file and document must be reliable for all users, which involve in the predefined workflow of the files and documents.

Now a question is raised that how workflow in paperless environment can be described in task ontology. In order to find out appropriate solution with respect to workflow in paperless environment, the agent can predict workflow services capable of interpreting metadata of annotated claim files and documents resources understanding their annotations with respect to ontology of workflow, top-level ontology, specific domain ontology, task-level ontology and application-level ontology. To deliver the claim files and documents resources, ontologies will describe the metadata of files, documents and observations about the performance of staff with the user's current profile. Each aSPOCMS service follows the reasoning rules for some specific purpose. These guidelines (reasoning rules) rules the query for resources and their metadata. The reasoning rules reasons over linked structure of data and metadata descriptions.

II. Resource Representation of aSPOCMS

Semantic Web technologies e.g. Uniform Resource Identifier (URI), Resource Description Framework (RDF), RDF Schema (RDFS) and Ontology Web Language (OWL) make available with attractive potential to interpret and process the information by machine. URI is a global naming scheme to identify the resources as Web identifier by describing its primary access mechanism or by name in a particular namespace. RDF describes the concept of resources and their properties (metadata). The XML is the standard interchange format for RDF in the Semantic Web, and RDF Schemas serve to define the relations between resources of the RDF documents. OWL represents to define the concepts of resources and relations of RDF and RDFS documents. There is no restriction on the use of different schemas together in one RDF document. The schema identification comes with attributes being used from that schema so backward dereferencing is again easily probable. We can rationalize the expectations of Semantic Web [15] over paperless office content management system in higher educational institutions using Semantic Web technologies.

To represent the files and documents/sections of any educational institutions as resiurces for aSPOCMS, efforts are required. These resources can be used to accomplish the shared and reusability of knowledge within aSPOCMS. The shared and reusability of knowledge makes the global linked knowledgebase for agent-based applications such as aSPOCMS. As an example, we depicted the resources of Department of Computer Science and their metadata in figure 1 by using Altova SemanticWorks2009 [16]. We have represented the office file of Department of Computer Science with resource `#DCS_Office_File`.

Figure 1, shows list of URIs of files of

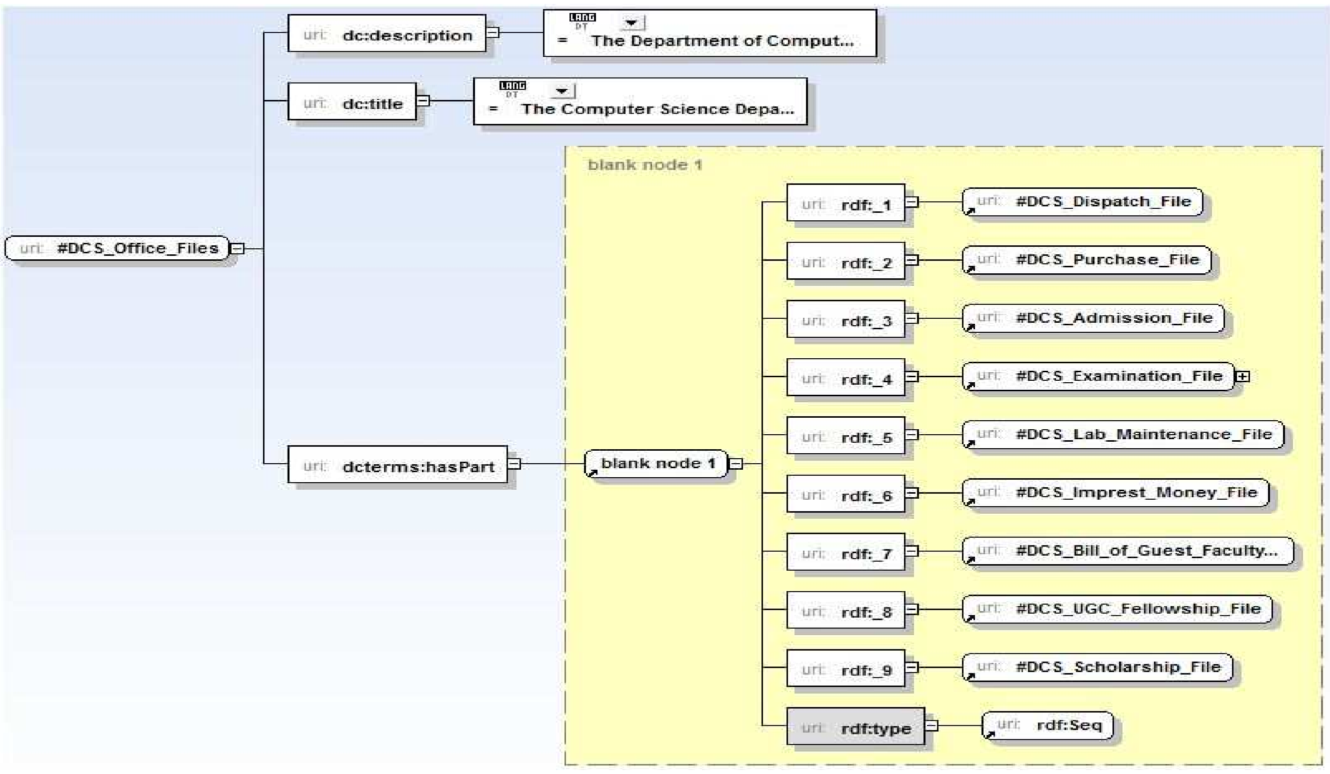

Figure 1: Graphical representation of the resources of Department related files as RDF.

`#DCS_Office_File` as `#DCS_Dispatch_File`,`#DCS_Purchase_File`,`#DCS_Admission_File`,`#DCS_Examination_File`,`#DCS_Lab_Maintenance_File`,`#DCS_Imperst_Money_File`,`#DCS_Bill_Guest_Faculty_File`,`#DCS_UGC_Fellowship_File` and `#DCS_Scholarship_File` etc. with `dcterms:hasPart` from Dublin Core metadata terms etc. The `dc:description`, `dc:title` and `dcterms:hasPart` describe the metadata of `#DCS_Office_Files`.

The graphical RDF representation can be examined in triples of subject-predicate-object. For example:

- <#DCS_Office_File><dc:title><The Computer Science Office files>
- <#DCS_Office_File><dc:description><The Department of Computer Science has various files related to various works>

Hence, the RDF statement in XML is generated as in figure 2, which is depending on RDF graph of related file.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/
terms#" xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:rdf="http://www.w3.org/1999/
02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
    <rdf:Description rdf:about="#DCS_Office_Files">
        <dc:title>The Computer Science Department Office Files</dc:title>
        <dc:description>The Department of Computer Science has various files
        related to various works.</dc:description>
        <dcterms:hasPart>
            <rdf:Seq>
                <rdf:li rdf:resource="#DCS_Dispatch_File"/>
                <rdf:li rdf:resource="#DCS_Purchase_File"/>
                <rdf:li rdf:resource="#DCS_Admission_File"/>
                <rdf:li rdf:resource="#DCS_Examination_File"/>
                <rdf:li rdf:resource="#DCS_Lab_Maintenance_File"/>
                <rdf:li rdf:resource="#DCS_Imprest_Money_File"/>
                <rdf:li rdf:resource="#DCS_Bill_of_Guest_Faculty_File"/>
                <rdf:li rdf:resource="#DCS_UGC_Fellowship_File"/>
                <rdf:li rdf:resource="#DCS_Scholarship_File"/>
                ………..
            </rdf:Seq>
        </dcterms:hasPart>
        ……………
    </rdf:Description>
    <rdf:Description rdf:about="#DCS_Examination_File">
        <dc:title>The Computer Science Semester Examination File</dc:title>
        <dcterms:isPartOf rdf:resource="#CS_Office_Files"/>
        <dcterms:hasPart>
            <rdf:Seq>
                <rdf:li rdf:resource="#Amit_Kumar"/>
                <rdf:li rdf:resource="#Arihant_Singh"/>
                <rdf:li rdf:resource="#Manoj_Bajpai"/>
                <rdf:li rdf:resource="#Abhijeet"/>
                ……………
            </rdf:Seq>
        </dcterms:hasPart>
    </rdf:Description>
    ………………
</rdf:RDF>
```

Figure 2: Representation of RDF statement in XML.

In figure 2, the resource `#DCS_Examination_File` is the semester examination file of Department of Computer Science, which has the list of name of students (e.g. `#Amit_Kumar`, `#Arihant_Singh`, `#Manoj_Bajpai` and `#Abhijeet` etc.) for semester examination.

The metadata of the resource can be described in RDF statement by using the object properties `dc:title`, `dc:description` from Doblin Core Metadata Initiative Standard [17] together and '`dcterms:hasPart`' from Dublin Core Metadata Terms. The agent of aSPOCMS will use the metadata of the resource to annotate the electronic version of files and documents over the workflow of university.

The RDF Schema provides a simple ontology language to express the vocabulary of resources of RDF statements. However, more powerful ontology language is also available, which reside on the top of RDF and RDF Schema. In order to combine reasoning mechanisms on the basis of metadata and resources, the link structures will bring the interoperability ideas with an agent-based application.

III. Reasoning and Restriction of the Resource of University Ontology

The layer of rules and logic framework [18] are found on the top of RDF and ontology layer in the architecture of Semantic Web tower. In our approach, the communication between reasoning rules and the agent-based office content management system in paperless environment will take place by exchanging RDF annotations. The reasoning function is intended to discover the potential relationship with the known relationship and acquire connotative knowledge from given knowledge by using certain logic and rule. An effective reasoning mechanism based on ontology can help to discover more extract and valuable knowledge. The key issues of university ontology and reasoning of aSPOCMS insure that computer can interpret the information described by university ontology according to formal description of ontology and discover the unknown potential relationship and connotative information from known relationship according to the relation property of university ontology. These reasoning rules are encoded in the TRIPLE rule language. Rule language is especially designed for querying and transforming RDF models in TRIPLE [19] [20] [21]. TRIPLE supports namespaces, set of RDF statements reification and rules with syntax close to that of first-order logic. A namespace is a collection of names, which identified by a URIref and URIref denotes the common usage of a URI. The expression of a namespace declaration is in the form of *namespaceabbreviation:=namespace*, and resources can use this namespaces abbreviation. For example, DCS_Office := "#DCS_Office_Files".

An RDF statement (which is a TRIPLE) are similar to F-Logic syntax, which is written as $Subject[\Pr edicate \rightarrow Object]$. Several statements with the same subject can be abbreviated in the following way:

- DCS_Office_File [dc:title->The Computer Science Department office files].
- DCS_Office_File [dc:description->The Department of Computer Science has various files related to various works].
- DCS_Office_File [dcterms:hasPart->DCS_Dispatch_File].

Now, we can add OWL restriction and relation properties in various classes and acquire that the value of the related property. We could visualize the extension of an ontology language such as OWL in figure 3.

There are some classes of university ontology in figure 3 OWL file e.g. Dean, Admission Cell, Department, Head are the owl classes. Further, the relations between OWL classes would be as below:

- Dean is a subclass of Admission Cell. (1)
- Department is a subclass of Dean. (2)
- Dean is an equivalent class of School. (3)
- Head is an equivalent class of Department. (4)

So, unknown information between (1) and (2) axioms is the "Department is a subclass of Admission Cell".e.g. Department $\subseteq$ Dean $\subseteq$ Admission Cell.

Similarly from (1) to (4) axioms, we can represent the relationship e.g.

- School is a subclass of Admission Cell (5)
- Department is a subclass of School. (6)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
    <rdf:Description rdf:about="#AdmissionCell">
        <rdf:type>
            <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
        </rdf:type>
        <rdfs:subClassOf>
            <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Nothing"/>
        </rdfs:subClassOf>
    </rdf:Description>
    <rdf:Description rdf:about="#Schools">
        <rdf:type>
            <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
        </rdf:type>
    </rdf:Description>
    <rdf:Description rdf:about="#Dean">
        <rdf:type>
            <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
        </rdf:type>
        <rdfs:subClassOf>
            <rdf:Description rdf:about="#AdmissionCell"/>
        </rdfs:subClassOf>
        <owl:equivalentClass>
            <rdf:Description rdf:about="#Schools"/>
        </owl:equivalentClass>
    </rdf:Description>
    <rdf:Description rdf:about="#Department">
        <rdf:type>
            <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
        </rdf:type>
        <rdfs:subClassOf>
            <rdf:Description rdf:about="#Dean"/>
        </rdfs:subClassOf>
    </rdf:Description>
    <rdf:Description rdf:about="#Head">
        <rdf:type>
            <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
        </rdf:type>
        <owl:equivalentClass>
            <rdf:Description rdf:about="#Department"/>
        </owl:equivalentClass>
    </rdf:Description>
</rdf:RDF>
```

Figure 3: Representation of restriction among resources by using OWL.

IV. Modeling of University Ontology

Presently, a huge endevours of effort has been devoted to surveying the ontology-related research studies from various aspects including that the ontology representation language [22], ontology development [23] and ontology learning approaches [24]. Ontologies provide a shared and common understanding concept of the domain that people can communicat with agent-based systems. The university ontology is an ontology of aSPOCMS that offers various functions for managing, adopting and standardizing the groups of ontologies. It should accomplish the requirement of reuse of ontologies.

In the sense of this context, the university ontology should be easily accessible and provide the efficient support for re-use of existing relevant ontologies and standardizing them based on various type of ontologies and ontology representation language. The reasoning mechanism is based on university ontology exemplification [25] that we have designed to include three layers in figure 4: presentation layer, semantic layer and application layer.

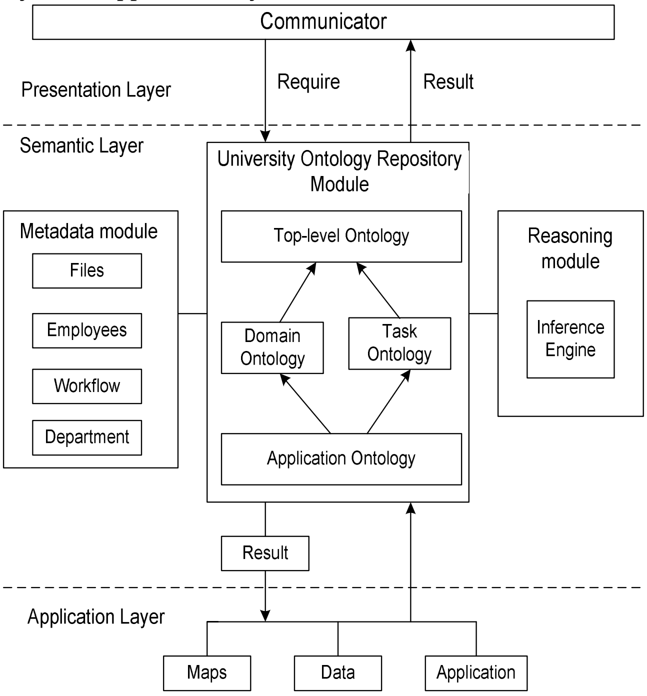

Figure 4: Different Levels of Ontology and their Relationship.

The formalization of university ontology is the conceptualization of concepts, which are sharing among workflow processes. These concepts can use by the agent to transmit the electronic file and document for administrative process of higher educational university and institute. Sharing concepts refer to the concept models of university information related to offices and workflow. We can design the workflow based University ontology to build up a paperless environment for enhanced the office content management system [5]. The presentation layer will provide the interface to the users shows the information which is designed in ontology.

In semantic layer, the university ontology repository module will be divided into four sub-modules: top-level ontology, domain-level ontology, task-level ontology and application ontology. The issues of university ontology repository module insure that the computer can interpret the information described by ontology engineers according to ontology formal description and intended to investigate unknown potential relationship and connotative information from known relationship and relational property of various level of university ontology. Reasoning module will extract the connotative logic relationship from university ontology repository with the help of inference engine. Presently, there are multiple inference engines that can use for reasoning. For example, Jess, F-OWL, Jana and RACER etc. can be used to do parsing and reasoning based on given rule in OWL file. The metadata module will provide the information regarding the facts of the university. The facts of university are the information of files and documents, profile of employee, metadata of department and the information of workflow for administrative process.

The application layer manages the information to be published. As the result, the mapping of the information can be done to utilize the data for the requirement of the users. We can abstract the concepts in heterogeneity information and reason out the connotative knowledge. Therefore, knowledge can be shared and interoperated on semantic layer.

A. Model of Top-level Ontology

Top-level ontologies are used to provide the theoretical underpinnings for representation and modeling in information systems in ways designed to bring benefits in the form of more reliable applications, better quality data-creation, and also help in error-detection [26]. It is generic ontology, which describes very general concepts independent of domain such that general concepts are metadata of Department, School and University etc. This ontology expresses universal concepts and the relationship among these concepts. The general concepts of Departments and various Sections can be used by other higher educational Institute and University also. For example, the defined information for a university resource (such as school, department and section etc.) can be used by the other university.

Top-level ontology involves different resources, which present in a higher educational university and institute. We can see a resource structure defined in figure 5, which includes various sections and related resources of the university.

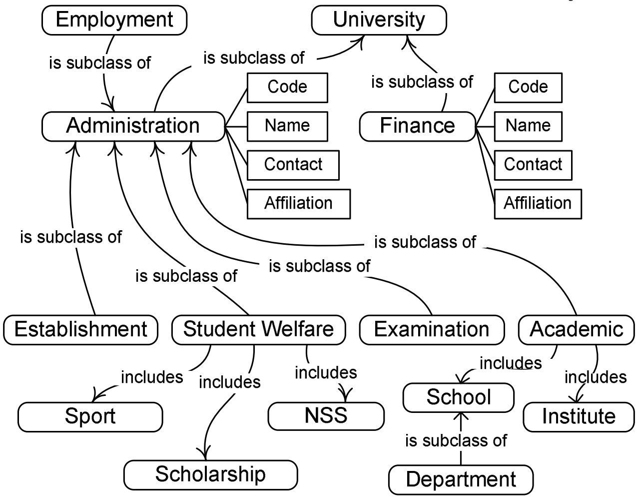

Figure 5: Partial view of the Top-level Ontology of University.

The `Administration` and `Finance` resources are the subclass of `University`. The `Administration` resource has five subclasses: `Establishment`, `Employment`, `Examination`, `Academic` and `Student Welfare` of the university. The `Student Welfare` resource includes `Sport`, `Scholarship` and `NSS`. The `Academic` includes `School` and `Institute`. `Department` is a subclass of `School`.

B. Model of Domain Ontology

Domain ontologies are intended to facilitate the automated data-sharing between complex information systems in specific fields and also at sustaining the automatic construction and population of ontologies developed in this field. Domain Ontology describes the vocabulary related to a generic domain e.g. the vocabulary of generic domain is courses, employees and student's profile of department etc. of higher educational institution. This ontology will provide concepts and relationship among concepts of special domain. The ontology model of the employee's activity profile and relationship among them is depicted in figure 6.

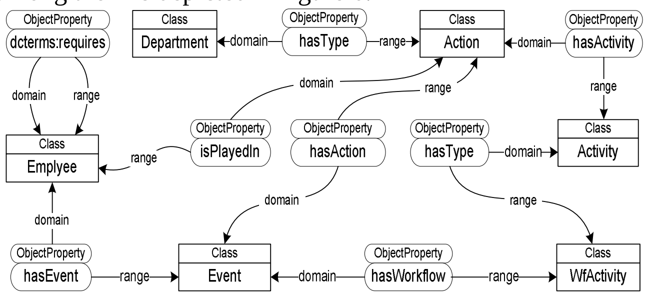

Figure 6: Ontology of employee with their action of event.

In order to create the activity, we need to capture some "Event" about the "Action" in university. By "Action" we mean something preformed by an actor (an employee, a computer system and other entity). The "Event" is something that we can capture by computer system as well as can be done by employee. Events are the "responsibility of a particular employee among the workflow of activity", "the document used", "performed the activity of workflow" and "activity received by the employee" etc. In this way, we can say that the event is modeled by the employee for "performing an action" on particular resource in particular workflow activity (WfActivity) instance of the ontology of aSPOCMS. The users of aSPOCMS and their relationships to other components are shown in Figure 7.

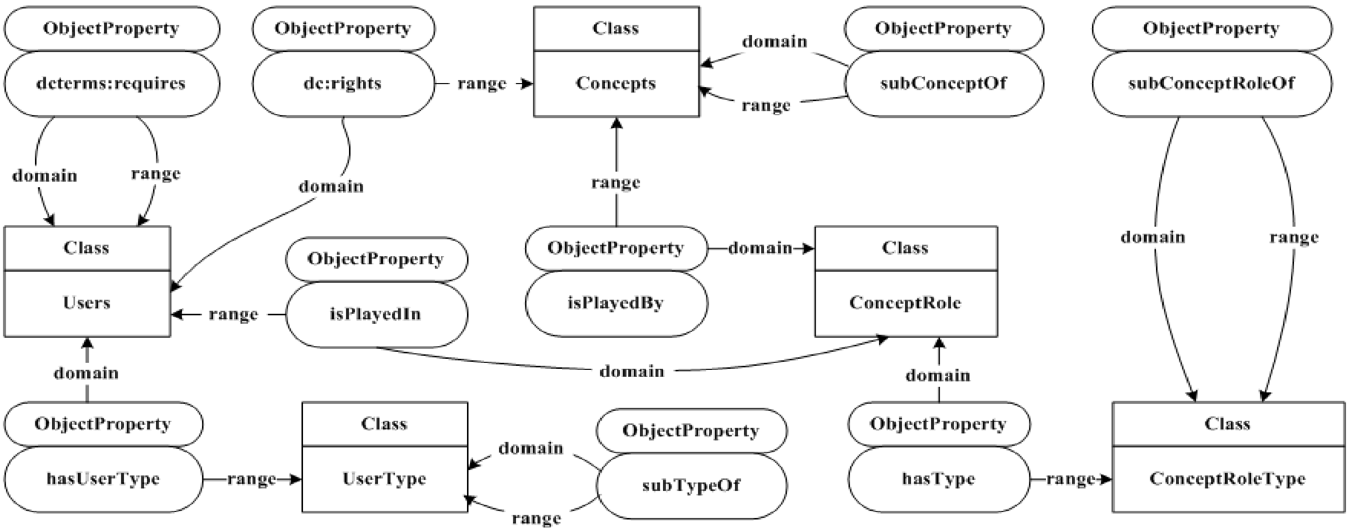

Figure 7: Ontology of Users and their Relationship

The class `Users` is used to annotate a resource. The users will engage on some concept roles e.g. some users are sending the electronic documents to other users for particular task and as well as getting the electronic documents for further processing. These concept role is played by some special concept (actions of workflow) i.e. payment of guest faculty, new admission of student and payment of empress bill etc. We use class `Concept` to annotate the concepts.

Concepts and Users are related through *dc:right* object property which means that users have some rights of concept to play the role in higher educational university and institutes. The users can be ordered by *dcterms:requires* relationship. Users and concepts have a certain role in their collaboration with certain concepts. We represent these facts by instances of `ConceptRole` class and its two properties such that *isPlayedIn* and *isPlayedBy*. Users, concepts and concepts role can form hierarchical struucture. We define *subConceptsOf* and *subConceptRoleOf* vocabularies for these properties.

C. Model of Task Ontology

Task ontologies express the concepts and the relationships among concepts for special task and action (workflow). These ontologies describe the vocabulary required to perform generic tasks and activities, again by specializing the concepts provided by the top-level ontology. It enables to reuse of services. If we have knowledge about processes and tasks within a specific domain then there is a higher potential for reuse of the knowledge and of the processes.

In the task ontology, we can describe various workflows of files and documents and their relationship among other workflow of files and documents of higher educational institutions. It provides the services to the users of aSPOCMS with the competent authorization. The specification of task ontology is depicted in figure 8.

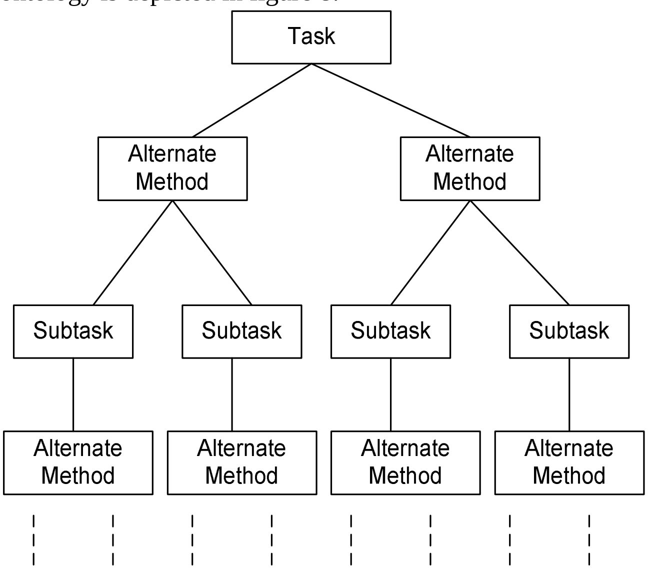

Figure 8: Specification of Task Ontology

In any administrative task (process or workflow) of higher educational institute, there may be alternative method to process it. The workflow can be divided into two categories. One category is fixed pattern workflow and other is non-fixed pattern workflow. In fixed pattern, the workflow of a process has dedicated method to process the document for particular purpose and non-fixed pattern workflow has the alternative method to resolve the process of document/file. When a document regarding a task is submitted in workflow circulation, then it handed over to workflow service, which selects next performer according to pre-set workflow circulation logic and metadata of particular files. Alternative method has various subtasks and further, these subtasks may have various alternative methods and so on. As an example, the graphical workflow of a financial process of document or file in any higher educational institution is shown in figure 9.

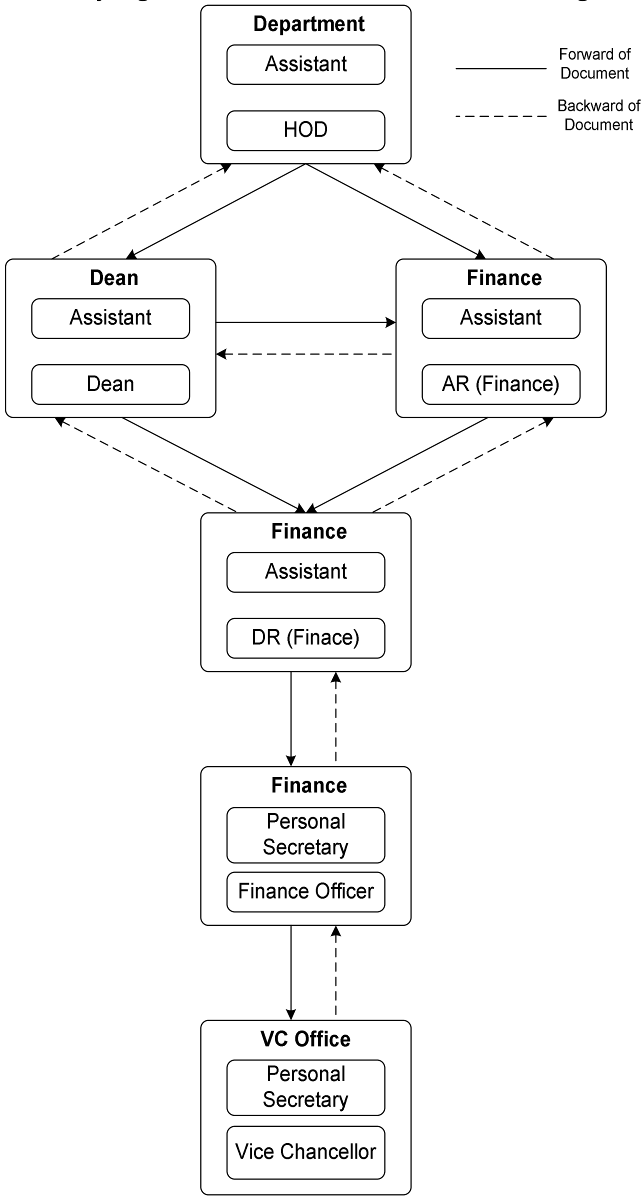

Figure 9: Workflow ontology model of a particular process.

In figure 9, if a file related to financial process has been send from any department of university for financial processing/approval. Normally, the department has two ways to forward the file for processing. The file may be sent directly to `AR(Finance)` or via `Dean` of the particular school to `AR(Finance)` or `DR(Finance)` to proceed the file for processing. When the file is processed then it will be back from the same route.

D. Model of Application Ontology

The application ontology describes the concepts and relationships among concepts, which depends on special domain and task. In this level of ontology, we will describe the relationship between users and their association with various activities of aSPOCMS with authorization. In order to present an activity of an employee, we must need to identify a work context of user. An example of application ontology is depicted in figure 10.

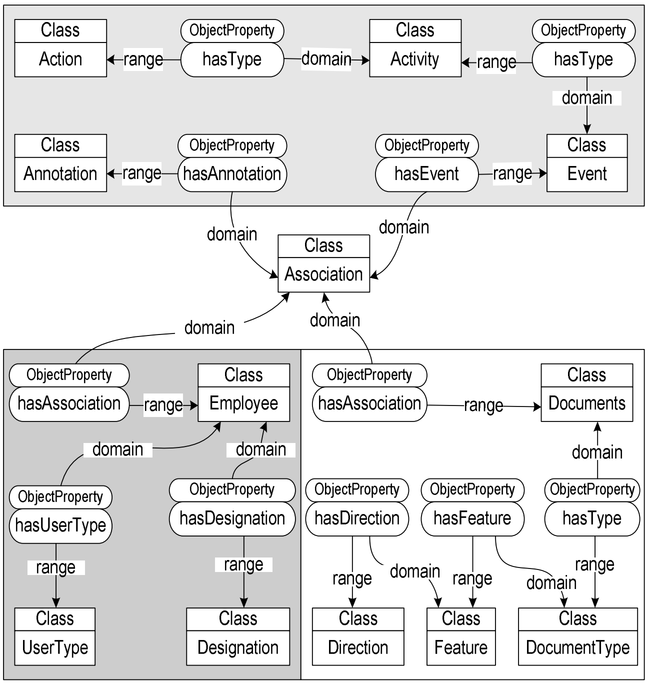

Figure 10: Application ontology.

In figure 10, the main element of the ontology is the `Association`, which is built from following components: `Annotation`, `Event`, `Employee` and `Documents`. `Annotation` and `Event` components are related to `Association` through `hasAnnotation` and `hasEvent` properties respectively. `Employee` and `Documents` are related to `Association` with `hasAssociation` property.

The `Event` refers a particular `Activity` with the workflow of `Action`. The `Employee` class further categorized in `UserType` and related to `hasUserType`, and also point to `Designation` through `hasDesignation` property. `Documents` are related to `DocumentType` through `hasType` and their `Feature`. A `Feature` can be a `Direction`.

V. Observations of User's Performance and their Dedication

The users can interact with a linked structure of the resources of aSPOCMS during runtime. The user's interactions with the resources can be used to draw conclusions about possible performance of users and their dedication with responsibility upon the task. Ontology of observations should provide a structure of information about possible user observations. The performance and dedication of the users can be measured on the basis of interaction time taken by competent users to response their experience regarding the particular workflow process of a file and document i.e.

`Interaction Time=endTime – beginTime`

For example, when the user takes the less interaction time to respond to a file or document, the performance of that user with regard to its responsibility would be measured as good. While a user who takes more interaction time to complete his/her task, the performance of that user with regard to its responsibility would be measured as poor. An ontology model for observing the performance of user and their dedication is depicted in figure 11.

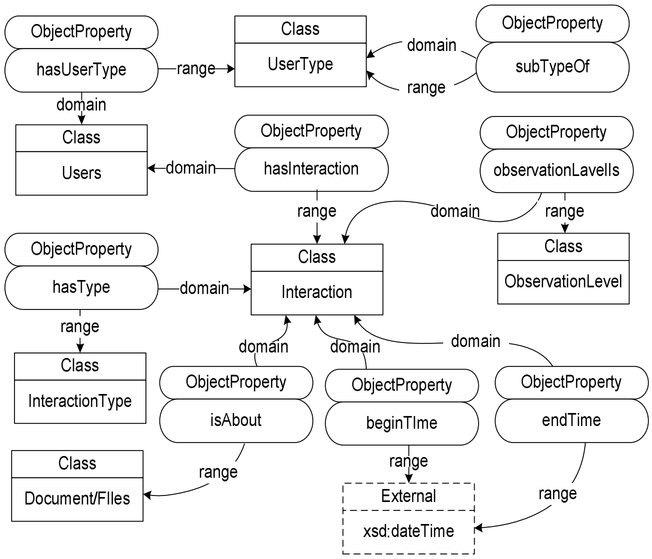

Figure 11: Ontology for User's Performance and their Dedication.

This ontology allows us to instantiate facts that a `Users` of our system has interacted with `hasInetraction` (Object Property) with a particular Document with `isAbout` (Object Property) via an interaction of a specific `InteractionType`. The `Interaction` has taken place in a time interval between `beginTime` and `endTime` and has a certain level with `ObservationLevel`. The users can contribute to an interaction with several workflows of files and documents. There are different kinds of `InteractionType` such as `access, bookmark, annotate` etc. and there are different `OveservationLevels` that a user has visited a page, and worked on a process etc.

VI. Conclusion

Because of its significant meaning in modeling of workflow information, sharing and interoperating workflow data and university taxonomy and so on, ontology plays an important role in higher educational institutions. In this paper, we elaborated the concept of university ontology of aSPOCMS to develop various levels of ontologies e.g. Top-level Ontology, Domain Ontology, Task Ontology and Application Ontology. Top level ontology is the vocabulary of all the resources of the organization such as various departments and sections of higher educational institutions. The domain ontology is depicting the vocabulary of activities of departments and sections and the relations between them. Task ontology is the concepts of special task or actions of various departments and sections. It provided the workflow of the process of file and documents. Application ontology generates the concepts and relationships among concepts, which depends on special domain and task. The integrated mechanism of these various levels of ontologies is a university ontology repository, which will play the significant role in aSPOCMS services. Finally, user's performance and their dedication have been measured according to their interaction time on action of workflows.

References

[1] Blum, B.L. (1989, October). Toward a paperless development environment. *Proceedings of the IEEE International Workshop on Architectures, Languages and Algorithms* (pp. 495-498), Fairfax, VA.

[2] Breaux, P.J. (1994, September). Using computer technology to support cost-effective paperless interactive technical data. *Proceedings of the IEEE Systems Readiness Technology Conference on Cost Effective Support* In to the Next Century (pp. 83-87), Anaheim, CA.

[3] Tim Berners-Lee, James Hendler and Ora Lassila. 'The Semantic Web'. *Scientific American* May 2001.

[4] Tim Berners-Lee, Semantic Web-XML2000. *http://www.w3.org/2000/Talks/1206-xml2ktbl/slide10-0.html. 2000*.

[5] Dr. Sanjay K. Dwivedi, Anand Kumar (2011). "Administrative Office Content Management via Semantic Web – A Case Study", ICICES 2011: *Proceedings of the 2011 IEEE International Conference on Information, Communication & Embedded Systems*, pp. 60, Chennai, India, February 2011.

[6] Sanjay K. Dwivedi and Anand Kumar, "An Agent-based Semantic Web for Paperless Office Content Management System", ICDM 2010: *Proceedings of 2010 Third International Conference on Data management*, pp. 352-360, Ghaziabad, India, March 2010.

[7] Fensel, D., Bussler, C., Ding, Y., Kartseva, A, Klein, V., Korotkiy, M., Omelayenko, M., and Siebes R, "Semantic Web Application Areas", In *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems*, Stockholm, Sweden, June 2002, pp27-28.

[8] O. Lassila and R. R. Swick (editors). Resource description framework (rdf) model and syntax specification. *http://www.w3.org/TR/1999/REC-rdf-syntax-1999022/*, Feb, 1999.

[9] W3C, "RDF Primer", W3C Recommendation, Retrieved from, *http://www.w3.org/TR/2004/REC-rdf-primer*, February 2004.

[10] Brickley, D. & Guha, R. eds. (2002), Resource description framework (RDF) schema specification. *W3C Working Draft* 30 April 2002 Cambridge, MA: W3C.

[11] Resource Description Framework (RDF) Schema Specification 1.0, retrieved October 10, 2004 from *http:/www.w3.org/TR/rdf-schema*.

[12] OWL Web Ontology Language Reference. *http://www.w3.org/TR/owl-ref/*.

[13] Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D. L., & Patel-Schneider, P. F. (2001). OIL: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16(2):38-44.

[14] XML (2003). XML: extensible Markup Language, retrieved October 10, 2004 from *http://www.w3.org/XML/*.

[15] Anand Kumar, Sanjay K. Dwivedi (2011). "Semantic Web Expectations and Challenges in aSPOCMS", *International Journal on Recent Trends in Engineering & Technology* [IJRTET], Volume 5, Issue 1, March 2011, pp. 60-65.

[16] SemanticWorks Semantic Web tool - Visual RDF and OWL editor, *http://www.altova.com/semanticworks.html*.

[17] Weibel, S.; Kunze, J.; Lagoze, C.; Wolf, M. 1998. Dublin Core Metadata for Resource Discovery. IETF #2413. *The Internet Society*, September 1998.

[18] DAML+OIL, retrieved October 10, 2004 from *http://www.daml.org/2001/03/daml+oil-index.html*.

[19] Sintek, M., & Decker, S. (2002). Triple - an rdf query, inference, and transformation language. In Horrocks, I. and Hendler, J. (Eds.) *Proceedings of the International Semantic Web Conference* (ISWC), Berlin: Springer Verlag, 364-378.

[20] Stefan Decker et al. TRIPLE - an RDF Rule Language with Context and Use Cases, *W3C Workshop on Rule Languages for Interoperability*, 27-28 April 2005, Washington, D.C., USA.

[21] Michael Sintek, Stefan Decker: TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web. *International Semantic Web Conference* (ISWC), Sardinia, June 2002.

[22] O. Corcho, & A. Gomez-Perez, A road map on ontology specification languages. In *Workshop on Applications of Ontologies and Problem solving methods*, 14th European Conference on Artificial Intelligence (ECAI'00), Berlin, Germany, August 20-25, 2000.

[23] D.Jones, T. Bench-Capon & P. Visser, Methodology for ontology development. In *Proceedings of IT&KNOWS Conference of the 15th IFIP World Computer Congress* (Chapman-Hall), pp, 20-35. 1998.

[24] A. Maedche, & S. Staab, Ontology Learning for the Semantic Web. To appear in: *IEEE Intelligent Systems*. 16(2), March/April 2001.

[25] 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.

[26] S. K. Milton. Top-Level Ontology: The Problem with Naturalism. In *Proc. FOIS*, 2004.

Author Biographies

Dr. Sanjay K. Dwivedi, Associate Professor, Department of computer science at Babasaheb Bhimrao Ambedkar University, Lucknow -226025 (U.P.) India (A Central University). His research interest is in Artificial Intelligence, Data Mining, Search Engines and e-Governance. He has published number of research papers in journals and conferences both. He is approachable at skd200@yahoo.com.

Anand Kumar, Research Scholar in Department of computer Science at Babasaheb Bhimrao Ambedkar University, Lucknow - 226025 (U.P.) India (A Central University). His research interest is in Semantic Web and Ontology design. He is approachable at anand_smsvns@yahoo.co.in.