IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

Proposed UML Approach for Ontology Design and Representation: A Banking System Case Study

P. Rathee[1*], S.K. Malik[2]

[1] USIC&T, GGSIP University, New Delhi, India
[2] USIC&T, GGSIP University, New Delhi, India

[*]*Corresponding Author: preetirathee09@gmail.com, Tel.: +91-9582141520*

Available online at: www.ijcseonline.org

Accepted: 20/Jun/2018, Published: 30/Jun/2018

***Abstract*—**For sharing and integrating knowledge on the web semantically, Ontology is one of the most significant technologies of the semantic web stack to provide structured information on the web in machine processable format. Ontology has various research issues where Ontology design and representation is the most fundamental. For Ontology design and representation, there are various methods and tools available. One of the crucial Meta model-based approaches is UML (Unified Modelling Language) which is a graphical notational language for ontology design and representation. The UML has been extensively followed through the software engineering network and its scope is broadening to consist of various extra modeling features. In UML, the flow of control and data through the different stages in a procedure is represented by using structural and behavioral notations like in activity diagrams, Use case and class diagrams. In this paper, first Ontology key issues and the role of UML for Ontology design and representation has been explored and discussed. Second, a related literature review of Ontology and UML for Ontology representation has been presented. Third, Ontology creation activities and building stages have been discussed with the help of diagrams along with UML usage and benefits. Fourth, a case study of Banking System has been chosen for Ontology design using UML which includes Use case diagram, Activity diagrams and Class diagram to represent Banking System Ontology.

Keywords— *Ontology, UML, Association, Generalization Aggregation, Use Case Diagram, Class Diagram, Activity Diagram, Banking System, Ontology Design.*

I. INTRODUCTION

Sir Tim Berners's Lee proposed a layered stack for Semantic web technologies in which Ontology is a key concern to embed semantics and integrate machine-understandable knowledge in a structured form [1] which has been shown in figure 1.Ontology is widely utilized in web semantics and forms the core of semantic web architecture. It is the key technology in semantic web layered stack where OWL (Web Ontology Language) is the language most widely used. It offers with the properties and nature of entities. It also indicates the connection between numerous data and offers with the idea of how information is associated with each other [1].

Ontologies additionally study the subdivision and hierarchal relationships between entities and research the differences and the similarities [2]. They have various issues like Ontology merging, matching, designing and import-export etc. Ontology designing and development is most fundamental and is the foundation for various Ontology Concerns as shown in figure 2.

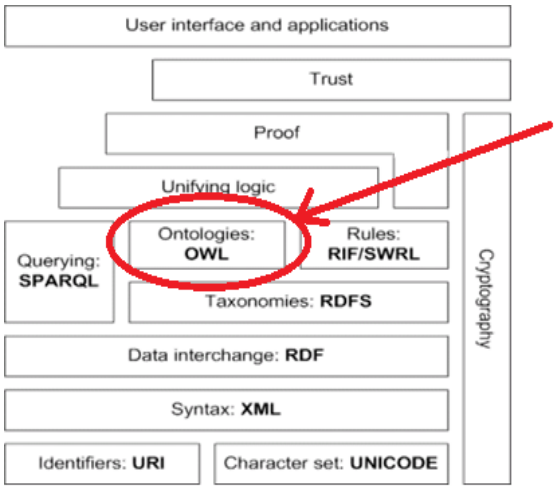


Figure 1. Tim Berners's Lee Layered Stack [Source: Lee 2006]

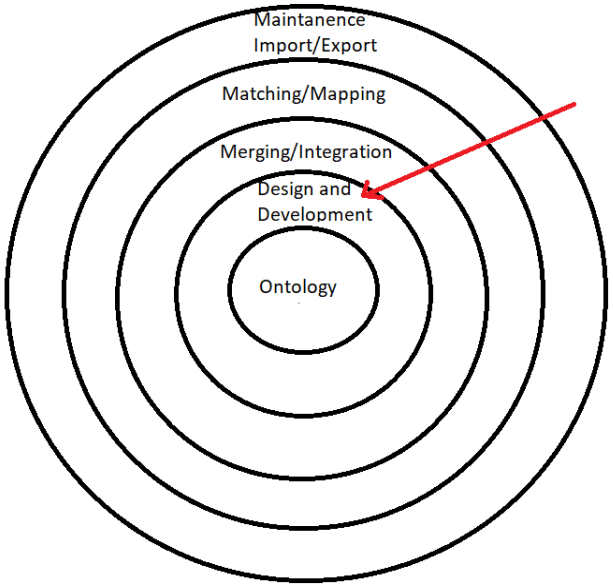


Figure 2.Key Issues of Ontology

Designing an ontology is a complex task and it consists of activities such as selection, description and analysis, which are executed during specification, conceptualization, implementation and evaluation of an ontology [3]. There are several methods and tools for building ontologies like Cyc Methodology [4], Enterprise Ontology (EO) Methodology [5], Enhanced Methodology [6] and Integrated Ontology Development Methodology [7] as shown in figure 3.

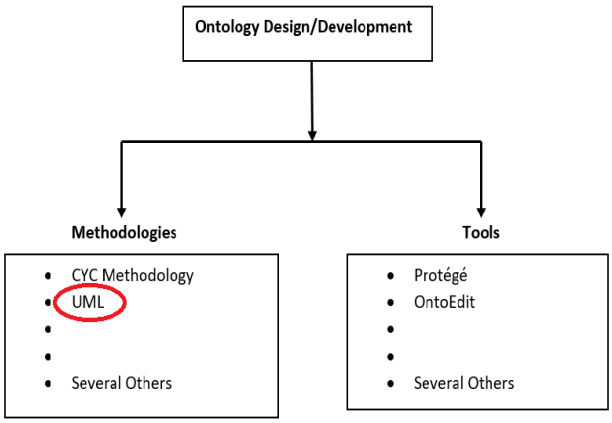


Figure 3. Ontology Design Methodologies and Tools

One of the approaches to design an ontology is by using methods applied in Software Engineering and by following software development life-cycle [8]. In this, specifications are captured inclusive of Unified Modelling Language (UML) [9].UML, encircled in figure 3, is a notational language based on a model called Meta Model and it describes specifications with graphical notations using activity diagrams [10]. It is a methodology independent, and it follows software life-cycle. In UML, the flow of control and data through the different stages in a procedure is represented by using structural and behavioural notations like in activity diagrams, Use case diagrams and by using Class diagrams [11].

The paper is organized as follows. In Section I, Ontology key issues have been defined and role of UML for Ontology design and representation has been explored along with necessary discussions. Section II presents a related literature survey of Ontology and UML for Ontology representation. In section III, Ontology creation activities and building stages for Ontology design and development along with UML usage and benefits are presented. Section IV presents a case study of Banking System for Ontology design using UML which includes use case diagram, activity diagrams and class diagram to represent Banking System. Section V concludes research work with future directions.

II. RELATED WORK

Ontology is defined as "an explicit specification of a conceptualization" [12] and design conditions are proposed for shared conceptualization and to provide knowledge sharing ie. Coherence, transparency etc. Extended definition of Ontologies is as "A formal specification of a shared conceptualization" [13] and proposed that existing ontologies can be reused to add specifications. Aspirez [13] combined Gruber and Borst definitions to define Ontology as, "A formal explicit specification of a shared conceptualization" where word explicit defines concepts, properties, axioms, restrictions and association composing them, formally used to state machine-readable, conceptualization defines the abstract model and a basic interpretation of existing things.

An iterative approach to develop ontology was proposed by Noy and McGuinness at Stanford University [14]. Ontology is designed by understanding the domain knowledge, by reusing existing ontology, defining classes and class hierarchy, determining class properties and slots. A prototyping technique is proposed by Chaware and Rao [7] to design Ontology which basically uses specification, conceptualization, acquisition, integration, implementation, evaluation and documentation. Enterprise Ontology uses four steps to design and developed ontology ie. by identifying scope, knowledge of Ontology, evaluation and documentation. Evaluation is done by using competency questions as in software requirements specifications. Toronto virtual Ontology (TOVE) uses scenario base method to define the functionalities of Ontologies. Firstly, relevant scenarios are identified to develop relevant questions. Ontology terminologies are specified for developing formal questions and finally, ontology evaluation is performed.

All above methods representing knowledge engineering field, the other approach used to develop Ontology by using UML ie. applied in software engineering [8]. A software engineering based technique [15] support ontology engineering for the semantic web they focus on class diagrams and object diagrams for expressing conceptual models of domain Ontologies. UML notations [16] are used to bridge the gap between Ontological models and software development. UML Based Ontology Tool-set project (UBOT) [17] which uses Ontology engineering and natural language based text annotating tool for DAML. In this, UML is used for visualizing and editing DAML Ontologies. CODIP (Components for Ontology-Driven Information Push) project [17] uses DAML Ontology for disseminating DAML messages in which UML is used to build and map DAML Ontologies. Sandpiper software [17] enables frame-based knowledge representation by extending UML to represent Ontological Knowledge. So UML is successfully used for Ontology related tasks.

III. ONTOLOGY DESIGN AND DEVELOPMENT

Ontologies represent domain knowledge by using the hierarchical description of classes, concepts, relations and services of specified domains [10]. Nowadays, Ontology is an essential element for various knowledge management applications. To represent semantic information or to provide machine-understandable data, Ontology languages are required. Ontology Development include steps like [10]

- Defining classes and entities in the domain Ontology.
- Representing classes in a subclass–superclass hierarchical structure.
- Describing relationships among entities and classes.
- Providing values to slots and instances.

A. Ontology Creation Activities

To design or build Ontologies, different creation activities are required like Ontology management development stage and support stage [10] as shown in figure 4 .

- ✓ **Ontology Management:** Includes different Scheduling activities, Quality Control and Quality Assurance activities.
- ✓ **Ontology Development:** Includes predevelopment activities, development and post development activities.
- ✓ **Ontology Support:** Includes Knowledge procurement, estimation, incorporation, integration, alignment, and documentation and formation management.

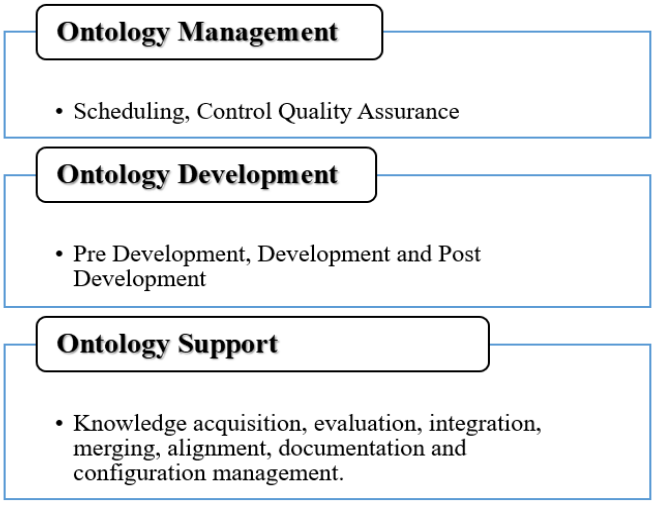


Figure 4. Ontology Creation Activities.

B. Ontology Building Stages

Different automatic and semi-automatic approaches have been designed for Ontology building. Ontology building includes stages like to identify the purpose of building domain Ontology which includes questions like: why to build Ontology? How to build? and for whom to build? Building Ontologies is basically classified under three stages- Ontology capture, which includes identification of key concepts of Ontology like defining entities, classes, axioms and constraints etc. [2], Ontology coding to represent knowledge in some formal language by using Ontology modelling languages and possible integration concepts of Ontology with existing Ontologies as tinted in Figure 5 [2].

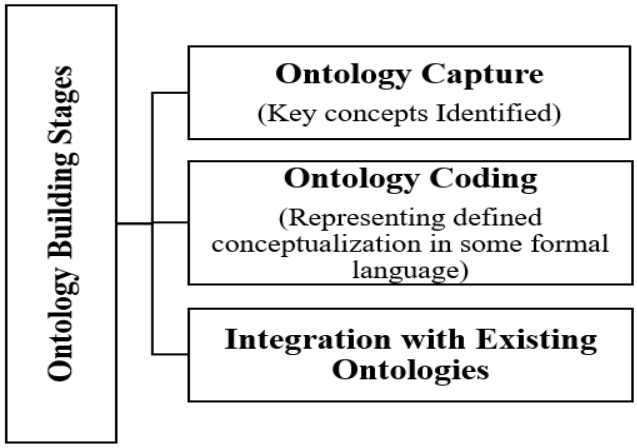


Figure 5. Ontology Building Stages

Basic steps for design and development of Domain Ontology are [2] [10]

- ✓ Domain Knowledge Specification: a primary step in Ontology designing is understanding the domain knowledge for Ontology building.
- ✓ Key Concepts identification: It involves identification of entities, classes and concepts of knowledge.
- ✓ Build Taxonomic Structure: Representing classes and entities relationships in hierarchal form.
- ✓ Identify association between Classes: Roles and properties used to represent the relationship between classes.
- ✓ Ontology Implementation: Involves formalization of Ontology by using modeling language to provide semantics to the information.

C. UML for Ontology Design and Development

UML is a methodology independent, and it follows software life-cycle. In UML, the flow of control and data through the different stages in a procedure is represented by using structural and behavioral notations like in activity diagrams [8]. It has a totally large and summarily increasing user community. Users of expanded information device infrastructure might be much more likely to be familiar with the notations used [11].In UML, there may be a graphical illustration for information expressed in UML. One of this graphical representation is essential to allow customers of dispensed information systems to browse an Ontology and find out ideas which can appear in their queries [9]. It uses various kind of the diagrams to represent the static and dynamic behavior of data. Here, static diagrams are used to model an Ontology which consists of class diagrams to represents classes in a particular domain and their relationships, and object diagrams to define instances of those classes.

Structured Similarities of using UML to develop Ontologies

- Ontologies define classes and subclasses, relationships between them, their attributes and axioms that specify constraints. In UML, classes/subclasses and their relationships are represented by using class diagrams and object diagrams [8].
- Both UML and OWL uses the concept of classes [8].
- In UML and OWL, model structure is used like packages in UML and Ontology in OWL [8].

IV. ONTOLOGY FOR BANKING SYSTEM USING UML: A CASE STUDY

Banks plays a significant role in the economy of a country. They help in national improvement by providing various facilities like loan facility, transactions import-export facility, credit to small-scale business, self-employed etc. [12]. The banking system has a large amount of data like data for finds transfer, a record of branches and employees, a database for customers, risk management, financial record, loan records etc. So to manage this large amount of data, it is helpful to build an ontology for banking services [12].

Ontology is the representation of knowledge in a particular domain by using a set of concepts and by showing the relation among those concepts. Mostly ontology defines individuals, classes, concepts and relations. Table 1 describes different components of an ontology [3].

Table 1. Components of Ontology

S.NO	Component	Description
1.	Individuals	Instances of a particular domain or objects.
2.	Classes	Basically a set. Collection of similar objects/instances.
3.	Attributes	Properties or features of objects.
4.	Functions	Actions performed by individuals.
5.	Relations	A relationship exists among classes and objects
6.	Events	Changing of attributes and relations

As UML describes specifications with graphical notations using activity diagrams, class diagrams [8]. UML follows software life-cycle and methodology independent technique. It uses structural and behavioral notations to represent the flow of control through different stages like in activity diagrams [8]. Here, banking system ontology is built by using UML Class diagrams, Activity diagrams and Object diagrams. XML and Java Code are also represented by using visual paradigms software.

A. Use Case Diagram Representation

Use Case diagram is used to describe functionalities of System. So we can utilize Use Case diagram to understand Ontology functionalities [11]. It basically uses an actor to interact with system ie. we can define individuals to interact with the system and their functionalities. It generally defines who all are using the database and for what purpose. The figure represents the use case diagram representation for banking system Ontology which includes different actors and their functionalities [11].

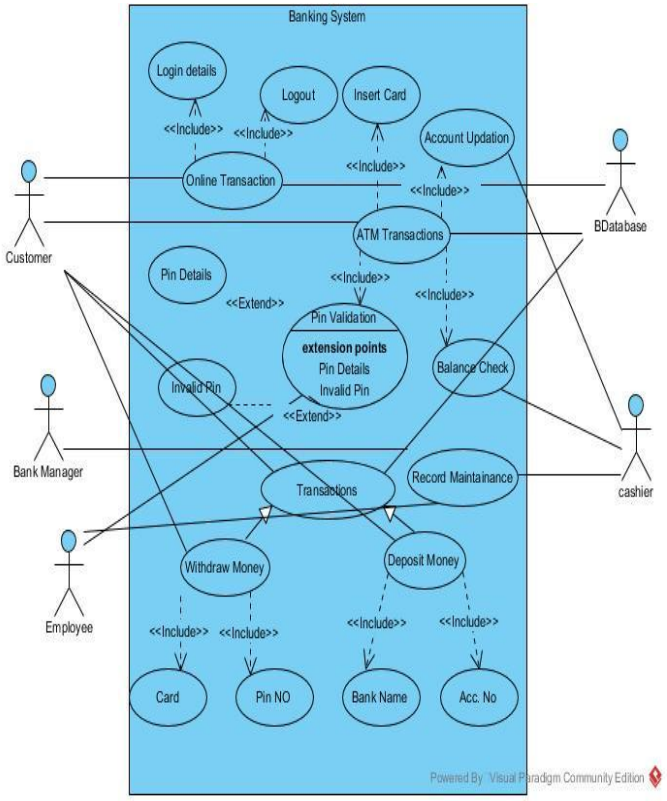


Figure 6. Use Case Diagram for Banking System Ontology

Figure 6 represents Use case representation of banking system ontology. There are different individuals as actors like customers, bank manager, bank database, cashier and employee etc. individuals have a different role in Ontology database as described:

- Customer- Customers can do online transactions, ATM transactions, withdraw money, deposit money etc.
- Employee- Employees perform pin validation, record maintenance, etc.
- Cashier- checks balance, record maintenance, transactions record etc.
- BDatabase- stores all transactions records, employee records, branch records etc.
- Bank Manager- maintains and check bank database, employee records and branch records etc.

B. Activity Diagram Representation

UML activity diagram is a state transition diagram to represent workflow from one state to another state ie. it is a flowchart that defines control flow from one operation to other[12]. These representations are used to model the activity flow of the system, to identify the system requirements.

There are various Benefits of Using Activity Diagrams to represent Ontologies as listed below [8] [11] [18]:

- To identify pre and post conditions of Ontology**.**
- To identify workflow among Operations defined.
- To model Ontological structural elements like class methods, functions etc.
- To represent the continuous behavior of information.
- To define roles performed in activities.
- To define roles of individuals in Ontology.

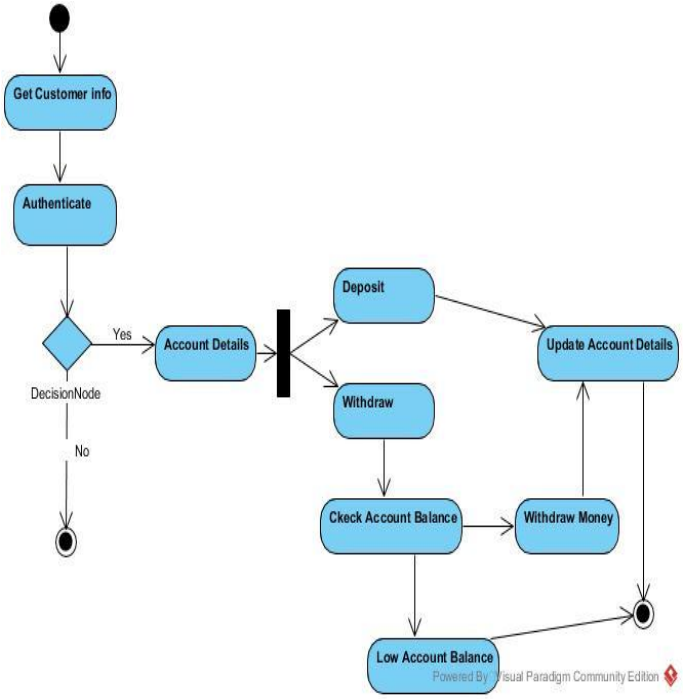


Figure 7. Activity diagram representation of Bank Ontology

The above figure 7 represents activity representation of bank ontology which is helpful in the study of dynamic behavior in an ontological form. It is basically used to gather the requirements of the model. The figure shows activity flow for an individual to perform operations in the banking system.

C. Class Diagram Representation

Class diagrams in UML are used to represent ontology in the form of classes and their relations. In class Diagrams, classes are represented in the box which is divided into 3 parts [11]

- Name of the classes used in Domain Ontology
- Attributes to describe the properties
- Operations that defines the actions performed by the class instances.

Ontology is the shared view of knowledge representation and therefore all the attributes in classes have public visibility mode. Operations used in classes represents queries performed by instances [8].
In UML, to show the relation among classes of a domain, three types of relationships are used [8] [11] as shown in table 2.

Table 2. Relationships Used in Class Diagrams

S.No	Relationships in Class Diagram	Description
1.	Generalisation	Represented by lines with large hollow arrowheads pointing to the super class
2.	Association	Represented by solid lines between two classes with optionally named ends, or roles
3.	Aggregation	An association with a diamond at the aggregate end of the link. UML includes a stronger type of aggregation (composite aggregation, notated by a solid black diamond) which implies ownership of the parts by the aggregate.

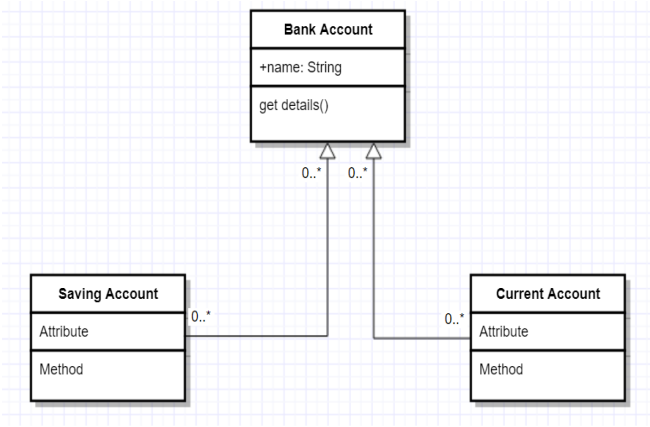


Figure8. Classes showing Generalization Relationship

Above figure 8 represents the generalization concept in which "Saving Account" and "Current Account" are similar classes from bank management domain. They have similar attributes and similar operations so, in generalization, similar classes are combined to form a super class like here we have "Bank Account" class which is a super class and "Saving Account" and "Current Account" are derived classes as shown in the figure.

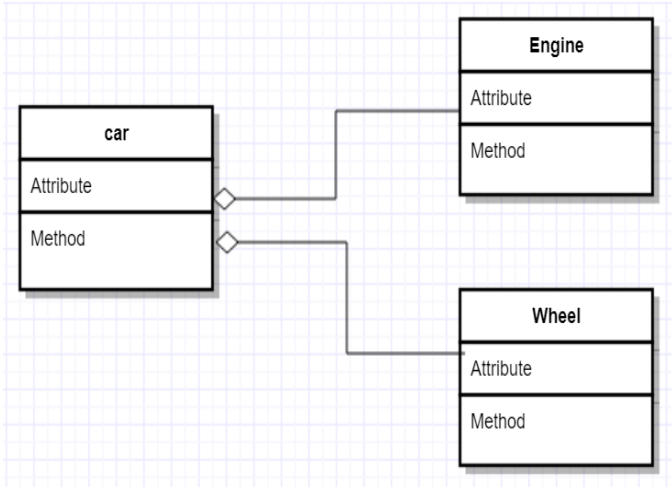


Figure9. Classes representing Aggregation Relationship

Above figure 9 represents aggregation relationship among "Car", "Engine" and "Wheel" classes. As aggregation is a type of association in which instance of one class can be used in another class, it does not represent any parent-child relationship. Here, if Car class is deleted, then engine class and wheel classes still exist. Basically, aggregation represents shared association among classes. As shown above, instances of wheel and engine classes can be used in car class ie. it represents containership concept among classes.

Association relationship concept is basically used to represent a static relation among classes in which there exists some type of relation or dependancy among classes. As shown in figure 10, "Student" and "College" classes have association relationship as students studies in college. These two classes are related to each other by this relation.

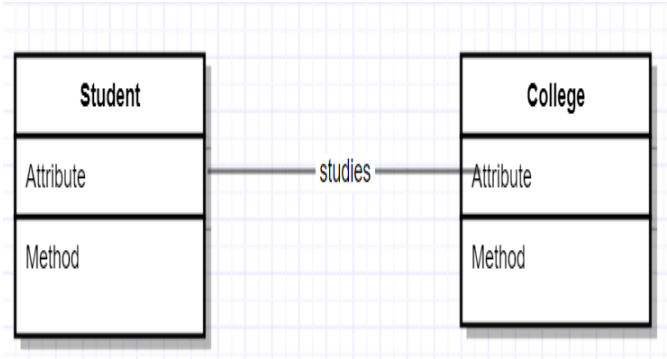
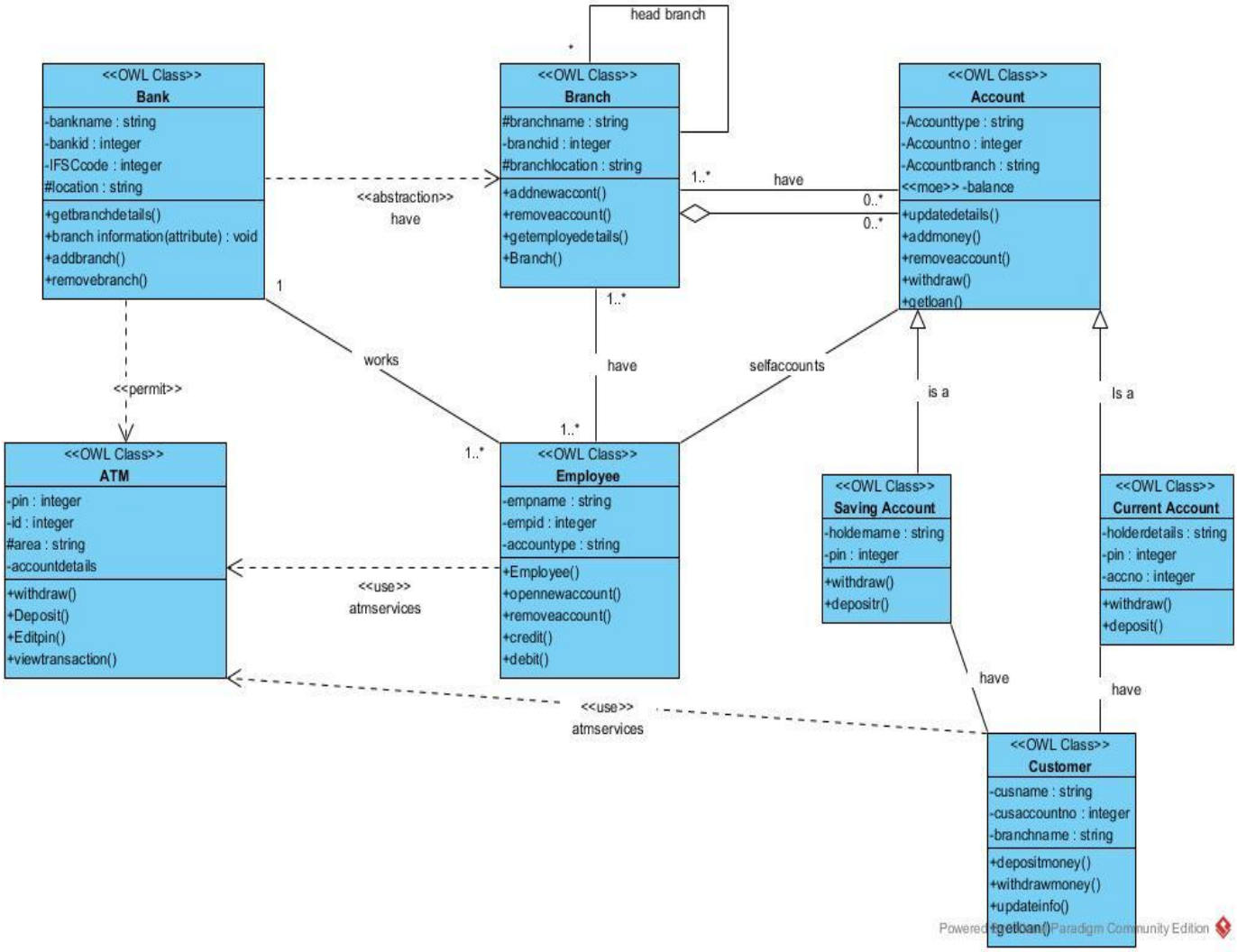


Figure 10. Classes Representing Association Relationship



SNo.	Class	Services
1.	Bank	Bank class represents head office that can have further subclasses as branch classes. This class have details regarding the branches
2.	Branch	Branch class represents a particular branch of bank class with the employee and accounts details
3.	Account	Accounts class provides all information regarding accounts.
4.	Saving Account	It is a subclass of Account class to represent particular account type
5.	Current Account	It is a subclass of Account class to represent particular account type
6.	Employee	Represents information reading all branch employees.
7.	ATM	Represents ATM of a Bank and its connection with a particular branch to withdraw and deposit money.

Figure11 Class Diagram representing Banking System

The figure 11 above represents banking system ontology in class diagram form. It represents banking system class with attributes and operations and also describes the relationship among classes in form of association, aggregation and generalization like the "Account" is a generalized class for "Saving Account" and "Current Account". Following is the description of classes used in the class diagram.

Table 3. Description of classes used in a class diagram

Table 3 describes the class used in the class diagram with their Services shown in figure 11. A Java code snippet for the Bank class is given as below:


```
public class Bank {

 private string bankname;
 private integer bankid;
 private integer IFSCcode;
 protected string location;

 public void getBranchDetails() {
  // TODO - implement Bank.getBranchDetails
  throw new UnsupportedOperationException();
 }

 /**
  *
  * @param attribute
  */
 public void addBranchInformation(int attribute) {
  // TODO - implement Bank.addBranchInformation
  throw new UnsupportedOperationException();
 }

 public void addBranch() {
  // TODO - implement Bank.addBranch
  throw new UnsupportedOperationException();
 }

 public void removeBranch() {
  // TODO - implement Bank.removeBranch
  throw new UnsupportedOperationException();
 }
```

Figure 12. Java code for Bank Class

The above figure 12 represents the java code for Bank class. It represents the attributes and operations of Bank class with their visibility modes like here one can have bank name and bankid with private mode and location with protected mode.

V. RESEARCH INFERENCES

- UML is a useful method to represents Ontologies in the form of classes and subclasses with the aid of graphical notations by using Use Case, Activity and Class diagrams.
- Representing in the form of graphical notations enables better semantics interpretation among Ontologies. UML is best suited for Ontology related tasks like mapping and consistency checking etc.
- Ontology creation, building and representation using UML Use Case, Activity and Class diagram is a better Meta model-based approach.
- May be a useful resource for Ontologists or researchers.

VI. CONCLUSION AND FUTURE SCOPE

Here, one of the crucial Meta model-based approaches which is a graphical notational language is used for Ontology design and representation in the form of UML. UML is extensively followed through the software engineering network and its scope is broadening to consist of extra various modeling features. Here, the flow of control and data through the different stages in a procedure is represented by using structural and behavioral notations like in Activity diagrams, Use case and Class diagrams. Starting from the role of UML for Ontology design and representation, various Ontology building activities has been presented along with UML usage and benefits. The most significant is the case study of a banking system Ontology using UML approach which includes various Activity and Class diagrams etc. They may be quite useful in various Ontology operations which may be a part of Future work. In future, the case study may be extended for more useful inferences or different case studies may be explored and related research concerns.

REFERENCES

[1] A. M. Khattak, R. Batool, Z. Pervez .," Ontology Evolution and Challenges", Journal of Information Science and Engineering 29, 851-871, 2013 .

[2] G. Flouris, D. Manakanatas, H. Kondylakis, D. Plexousakis, and G. Antoniou," Ontology change: classification and survey", The Knowledge Engineering Review, Vol. 00:0, 1–29.c, Cambridge University Press, 2007.

[3] G.Pérez, A. F.Lopez, M. Corcho, "Ontological Engineering with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web", Advanced Information and Knowledge Processing 1st ed. 2004.

[4] D. B. Lenat and R. V. Guha," Building Large Knowledge Based Systems-Representation and Inference in the CYC project", Addison- Wesley Publishing Company, 1990.

[5] Uschold, M., and King, M.," Towards a Methodology for Building Ontologies". IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing (pp. 6.1,6.10), Montreal, Canada, 1995.

[6] Ohgren, A. and Sandkuhl, K. "Towards a methodology for ontology development in small and mediumsized enterprises", In Proceedings of the IADIS International Conference on Applied Computing, pages 369–376, 2005.

[7] Chaware, S. Rao, "Integrated approach to ontology development methodology with case study", International Journal of Database Management Systems, 2:13–19, 2010.

[8] Jalloul, G.," UML by Example", Cambridge University Press, USA, 1st edition, 2004.

[9] F. López, M," Overview Of Methodologies For Building Ontologies", Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5) Stockholm, Sweden, August 2, 1999.

[10] Noy, N.F., McGuinness, D.L.," Ontology Development 101: A Guide to Creating Your First Ontology", Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001.

[11] Lunn, K." Software Development with UML", Palgrave MacMillan, New York, USA, 2003.

[12] T. R. Gruber. A Translation Approach to Portable Ontologies. Knowledge Acquisition, 5(2):199–220, 1993

[13] Borst, W., " Construction of Engineering Ontologies for Knowledge Sharing and Reuse", Ph.D. Dissertation, University of Twente,1997.

[14] C. Roussey, F. Pinet, M. A. Kang, and O. Corcho," An Introduction to Ontologies and Ontology Engineering", Ontologies in Urban Development Projects, Advanced Information and Knowledge Processing, Springer-Verlag London Limited, 2011.

[15] Baclawski, et al. " Extending the Unified Modeling Language for Ontology development ", International Journal Software and System Modeling (SoSyM),1(2),2001.

[16] X. Wang, C. W. Chan, H. J. Hamilton," Design of Knowledge-based systems with the Ontology domain system approach", Proceeding SEKE '02 Proceedings of the 14th international conference on software engineering and knowledge engineering , Pages 233-236, 2002.

[17] P. Kogut et al. "UML for Ontology Development", Knowledge Engineering and Management,2004.

[18] D. Nicola, A., Missikoff, M., and Navigli, R." A software engineering approach to ontology building. Information Systems", 34(2):258–275, 2009.

[19] R. B. S.López , M. S.López, M. Bravo, I. I. Gurrola, and V. G. S. Arias," GODeM: A Graphical Ontology Design Methodology", Research in Computing Science 84 , 2014.

[20] D. Kalibatiene, O. Vasilecas , " Survey On Ontology Languages", Conference Paper in Lecture Notes in Business Information Processing ,January, 2011.

Authors' Profile

Ms. Preeti Rathee completed her Masters in Technology (Computer Science and Engineering) from ITM University, India. She is currently pursuing Ph.D. from University School of Information, Communication and Technology, GGSIP University. She is currently working as Assistant Professor in Maharaja Surajmal Institute of Technology, GGSIP University. She has published few research papers in international conferences/journals of repute. Her main research interest focuses on Ontology and various Semantic Web concerns.

Dr. Sanjay Kumar Malik completed his Ph.D. in the area of "Semantic Web" from USIC&T, GGSIP University, Delhi. He is currently working as Associate Professor in University School of Information, Communication and Technology, GGSIP University. He has more than 18 years of industry and academic experience in India and abroad (Dubai and USA). His areas of research interest are Semantic Web and Web Technolgies. He has several research papers in reputed international conferences (India/abroad) and Journals. He has been session chair in several international IEEE/Springer conferences and honoured with third best researcher award, 2011 by GGSIP University for his contributions in research.

