



## Automation of Requirement Model in Object Oriented Software Analysis

Upasana Bhardwaj\*, Divya Nasa  
Computer Science Department

Maharaja Agrasen Institute of Technology, India

---

**Abstract**— This paper is about automation of usecase which (main tool in requirement modelling in object oriented software analysis). Usecases are the output of requirement model in objects oriented software development process. Using requirement elicitation template which help us to generate event table which will map into usecases. this process of automation will reduce our efforts to make usecases manually which will consume time. Usually we make usecase using drag and drop technology in any related software but with this approach on having requirement template we can easily generate usecase with better understanding of events occurring in the system as this approach also deals with the generation of event table. Relationships in usecases can also be produced with this. Thus the advantage of using such approach is to reduce the time consumed and also to better understand the system.

---

### I. INTRODUCTION

One of the components of automation of object oriented software development process is automation of Requirement model of object oriented software development as in [1]. Object oriented software development has various models. In this process we have a analysis phase which has two models, these models are requirement model and analysis, our automation is limited to requirement model process of generating usecases as in [2]. Generally in many organizations for building software in requirement analysis phase usecase is used as an analysis tool. Usecases are made manually by picking up the graphical image and manually labeling them according to the requirement (functionalities) of the software. In our automation we are eradicating the need of making the usecases manually, as our approach will automatically generate usecase according to the requirement. What we all want is an requirement elicitation template data compiled through the requirement elicitation template which gives the detail of what requirement are there of software, taking Requirement elicitation data as input we will automate the usecases with series of steps following our approach as in [4]. Thus automation of usecases will reduce the manual effort and also reduce the time require for the software development lifecycle.

### II. MANUAL WAY TO MAKE USECASES

Usecases are made manually with various tools, one of such tool is Rational Rose which provides a platform to make usecases manually, in tools like Rational Rose we have some features like graphical shapes which are required to make usecases. A user of such tool have to study complete requirements of the system, functionalities invoked as the result of meeting such requirement then by drag and drop option usecases are made, which require labeling of each usecase, labeling of each actor, making corresponding actions between actor and usecases by drawing arrows between them, study the include functionality that will be invoked as a result of prime functionality, than making them with the help of that tool. Such a making of usecases require lots of time in studying about the system, system behavior, system requirement and than making usecases as in [5].

### III. PURPOSE

Our aim is to reduce the manual effort require to make usecases and fasten the process of software development as in [3], and to simplify the understanding of the user by reading usecases. We will provide an automation approach to generate usecases in which we will reduce the effort require making usecases with the tools like Rational Rose, here in our automation we will able to make usecases with just clicking on a button, only the requirement for our automation is an requirement elicitation data.

### IV. OUR AREA OF CONCERN IN REQUIREMENT MODEL

We will automate an approach for generating use case model in requirement model of OOSE . Usecase model consists of usecases as in [11] .Thus we will focus on applying an automation approach for generating usecases.

#### A. Usecase model

- Represents actively interchange information with the system.
- It represents a human, a machine or another system.
- Use case models a dialogue between actors and the system.
- A use case is initiated by an actor to invoke certain functionality in the system.

- It models a dialogue between one or more actors and the system that requirement elicitation data a result of measurable value to atleast one actor is a complete and meaningful flow of events, represents a major system usage goal for one or more of the actors that interact with the use case.
- A use case is a generic description of an entire transaction of events involving the system and objects external to it. A use case can therefore be seen as a description of different states and the events that make the system transit from one state to another as in [9].
- Together the uses cases represent all the defined ways of using the system and the behaviour it exhibits whilst doing so.
- Each usecase is a specific type of using the system. A scenario (in UML) denotes an instance of a use case as in [12]. When a user (an actor instance) inputs a stimulus, the use case instance (a UML scenario) executes and starts a transaction belonging to the usecase, consisting of actions to perform as in [9].
- In OOSE the system model, as a whole, is use case driven. So if you want to change the system's behaviour, you should remodel the appropriate actor(s) and use case(s).
- The diagram notation used for expressing the use case model is defined in the UML as in [8]. Actors are classes, notated in their simplest form as stick figures with an instance name (or class box). Ellipses represent the different use cases and have an identifier naming them. Also the whole name is given a name. Lines identify the associations between actors and use cases.

The identification of each use case requires a detailed consideration of the system's requirements. A systematic approach representing the different use cases.

- **Usecase example:**

Example1: A simple usecase of customer as actor with login and register as usecases

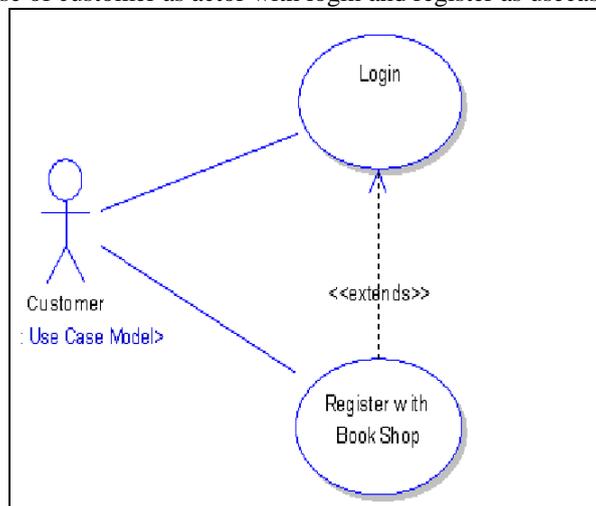


Fig 1: Book Shop Usecase

**B. What do we need to implement in requirement model?**

To provide automation approach for the very first model in object oriented software development – Requirement model. We will provide an approach which will automate the generation of usecases.

Our approach is basically concentrate on automatically generating usecases .such that if we have an input consisting of all requirements documented in some form so this can serve as an output to our approach from which the process of making usecases can be done automatically . Such an approach will involve understanding of complete requirements of the system and then helping the tool which is m

ade by this approach to automatically map those requirements to usecases.

## V. APPROACH USED

Our approach is to use event table as the main basis for extracting the usecases ad in [5], to start for automation we must have data compiled in requirement elicitation template which has all its requirements for the software, than provide event analysis, event analysis involves all event occurring in the system, requirements itself in the Requirement elicitation data state event that is occurring in the system. This analysis will help to generate an event table which will list all events in graphical form, so users can access all events occurring in the system. Finally we can map information on event table to usecase.

**A. Stepwise Approach-**

- Requirement specifications: To start for this process of automation, we should have listing all requirements that is Requirement elicitation data (stated above ) which will be represented in the form of event table listing all the requirements as events occurring in the system, as we have to generate usecases so we will consider only functional requirements here.
- Event analysis: from functional requirement template we will analyse events that invoke a particular functionality in the system.

- Event table: generate event table.
- Usecases: generate usecases from event table.

1) *Flow graph of Approach*

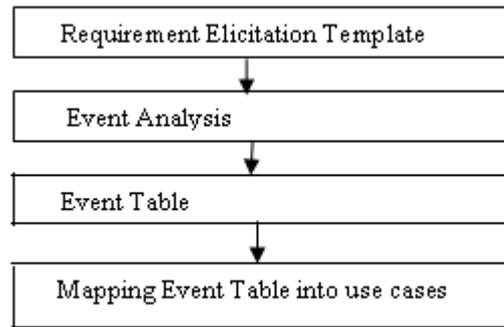


Fig-2: Flow Graph Of Usecase

2) *Event Analysis*

- An event is an occurrence at specific time and place, that can be described and should be remembered by the system or can be said that an event which causes the system to do something.
- For external events ,the message is data entering the system and for internal events the message is reaching point in time that makes the system process
- All requirements in Requirement elicitation document corresponds an event occurring in the system, that means for each and every requirement to be fulfilled some event has to be occurred.
- Such that these events occurring in the system must map into the requirements, that means they should be traceable.

3) *Types of events in a system*

- External Events: An event that occurs outside of the system, usually initiated by external actor or user.
- Temporal Internal Events : An event that happens when the system reaches a specific point in time.
- Conditional Internal Events: An event occurs when something happens inside the system and the system must initiate some process to response for this event.
- All these types of events occurring in the system are mapped into requirements stated in Requirement elicitation data.

4) *Event table*

- Event table is a table listing all actions as in [4].
- The result of event analysis is documented in event table.
- An event table represents the functional requirements of the system, thus for making an event table we have to prepare a functional requirement template which will be represented in event table.
- Functional requirements capture the intended behaviour of the system. This behaviour may be expressed as services, tasks or functions the system is required to perform.
- Each event in the system corresponds to an essential usecase.
- It is important to analyse each event of the system, this is because if we miss to state any of the events occurring in the system than it will affect our output as usecases .thus to have each and every requirement to be mapped into usecases each and every event.

5) *Attributes of event table*

1. Functional requirements stated in the Requirement elicitation data can be mapped into various attributes of event table.
2. Attributes of event table helps to explain the requirements in terms of attributes.
3. Attributes of the event table are :-
  - Event - An event is result of any invoked functionality of the system.
  - Input Message - For external events, the message is data entering the system and for internal events, the message is reaching point in time that makes system process.
  - General Source- It is to define from which user event has been generated, and to determine whether the source has any generalization relationship
  - Output Message- It is the output which is produces by the system as a result of the event generated
  - Includes- This is to determine the includes relationship between actions, if an event causes a system to perform some action, and that action leads to series of action than this actions are included in the primary action.
  - Extends –As includes relationship, extends is use to describe the extends relationship between the actions.

- Destination- Destination is the actor which receives the result of an action executed on behalf of an event.

6) *Mapping event table to usecases*

- Now when event table is prepared from Requirement elicitation data, all attributes of event table are listed. Our next step is to map the event table to usecases.
- Attributes of event table will help to map event table to usecases.
- In mapping the source listed in the event table is labelled as an actor in usecase diagram, and destination as another actor in a usecase diagram.
- Action in the event table is mapped as usecases , this leads to the simple hierarchy as Requirement elicitation data contains all requirement which will lead to many events in event table , as a result of which actions are involved for each event , this actions are mapped into usecases .
- In our Requirement elicitation data we have requirements which are mapped into usecases, we also have nested requirement which is mapped as included usecase in usecase diagram.
- Each usecase thus represents a functionality that is stated in the requirement which is a requirement for the system as in [10].
- Each actor states that he is the source from which functionality invoked refers to.
- Given below is diagram to explain how the event table attributes are mapped into the usecases in which source mapped as an actor and corresponding action is mapped to usecase which belongs to that actor.

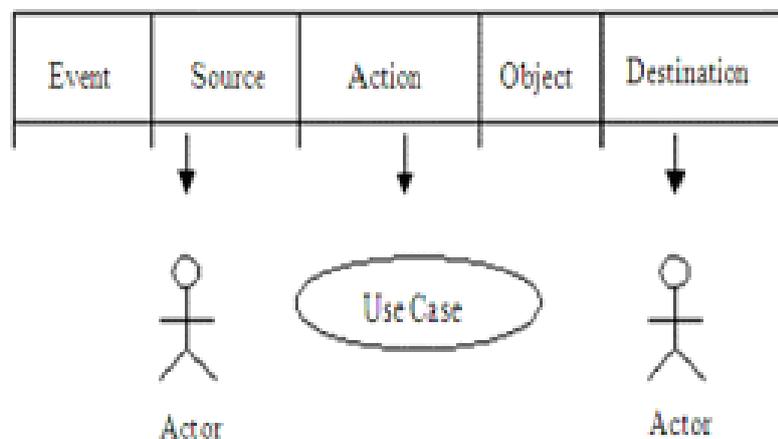


Fig3: Mapping

## VI. DESIGNING

For designing such a tool which provide an automation of usecases in requirement model as in[5] we have used Netbeans platform and java as a programming language in which we use swing classes for using Requirement elicitation data as a input interface in which user enters it requirement and is used to collect data of requirements in an Ms access database, from which an event table is made in a form of jframe which is linked with this database to fetch all requirements as different events.

- Our Requirement elicitation data is a input for our tool, so it should be of some generic format so that it can be processed for any application. It should have following fields:
  - i. Requirement id- it is a unique id given to each requirement.
  - ii. Requirements- This field states all requirement for the software.
  - iii. Nested Requirement- for a particular requirement there can be many nested requirements ,for example for login is a requirement and for login user must enter his username and password which is a nested requirement.
  - iv. Primary User- This field contains information about the user which has initiated the action.
  - v. Secondary User-This field contain information about the user to which the action is destined to, that means destination of the user.
  - vi. User type- This field tells about the types of primary user, for example a user can be a member or non member this field will help about the generalization of user.
- Although given above is the prescribed format for the requirement elicitation data but as for making this tool for generic purposes many features can be included in requirement elicitation data in order to have more detailed usecase. Its just more the information in requirement elicitation data, higher the level of usecase obtained as a result.
- In order to start for the automation, our tool must require this kind of an requirement elicitation data, as all the data from which automation has to take place is populated from requirement elicitation data itself.

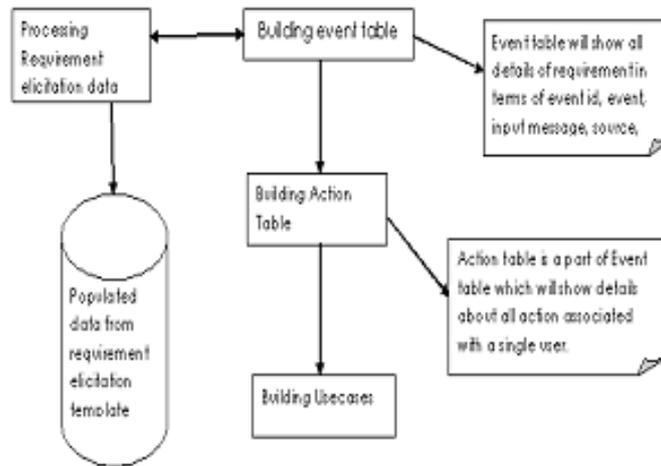


Fig-4: Flow Graph Of Usecase

All this phases of design architecture are explained as follows-

- **Preprocessing of requirement elicitation data-** Obtaining an input as requirement elicitation data, verifying that it contains all information to obtaining a required level usecase. Requirement elicitation data should contain all requirements so that each of them can be obtained as functionality in terms of usecase.
- **Populating Data-** Requirement elicitation data information about requirement of software is populated in terms of Ms Access database. This database will provide further basis on which event table and action table is obtained.
- **Building Event Table-** from the data populated by requirement elicitation data event table is made in event table requirement id is mapped into event id, requirements is mapped into action in the event table, primary user is mapped into initiator ,secondary user is mapped into destination and
- **Building Action Table-** Action table is part of vent table or you can say an abstraction view of event table in which you can have all actions listed in various fields for particular user. If you want all action of a particular user than you can simply select that user, actions corresponding to that user can be seen listed in action table.
- **Building Usecases-** Once event table is prepared and action table is prepared we can link it to obtain usecases which will map all actions of action table into usecases and initiator from event table into actor. In this way many usecases can be made for different user.

## VII. IMPLEMENTATION

We have implemented the approach in the series of steps ,each step will explain the particular phase of implementation of the approach and give the snapshot along with it.

### A.. Step 1

The first step is to have an application interface so that we can enter as many applications to the, all applications can be viewed with this interface. Thus provides information to the user usecase of which application user wants to look for, user can simply select the application id given to the application, selecting the application id will give the application name and application description so that user can get some description about the application.

#### Application Interface for Global Personal Marketplace.

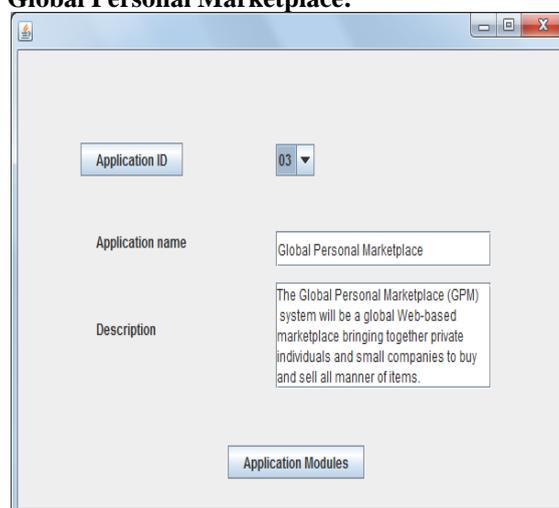


Fig-5: Application Interface for Global Personal Marketplace

**B. Step2**

For each applications we have modules, each module has a usecase, for this we have to select the particular module of an application for which a usecase will be generated. for every module there is a module id which is prefixed by application id .for example first module of application id 01 will be assigned module id 0101,second module will be assigned 0102. As module id is selected corresponding module name and its description is displayed.

**Module Interface of Global Personal Marketplace.**

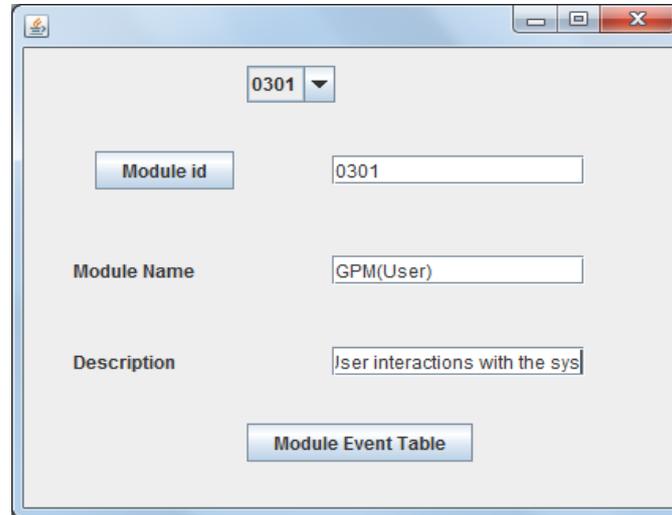


Fig-6: Module Interface for Global Personal Marketplace (User).

This interface provides-

1. Module id- unique id given to each application,here it is 0301.
2. Module Name- Name of the applicationof the given id, here it is GPM(user).
3. Module Description- gives a brief description about application and its modules.

**C. Step3**

In this OCR Requirement elicitation data is used to populate the data which is to be served as an input for our automation. Data is populated thus from this data event table can be generated.

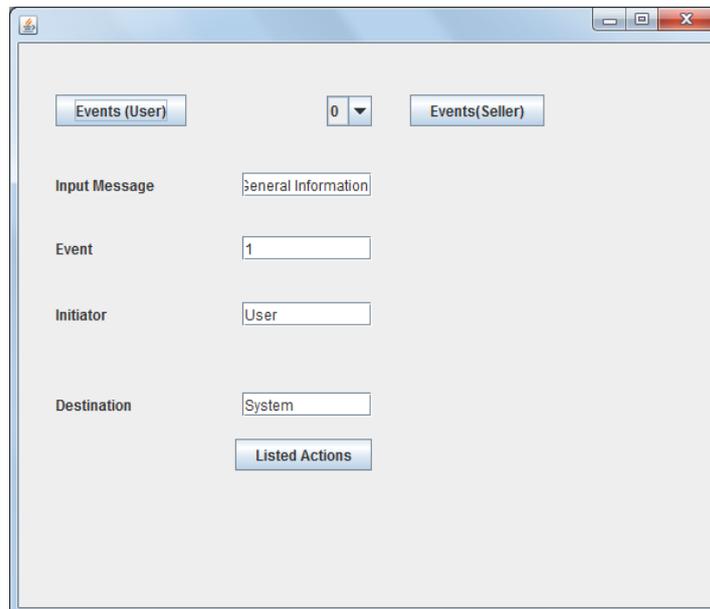


Fig7:Event Table of Global Personal Marketplace (User)

**D. Step4**

Action Table from event table – When an entry from event table is fetched than we can list all are actions in event table to action table. This make our work easier as to display event table for each action will be cumbersome for this we had made action table in which for a particular user we will list all actions. User can simply select the user in action table, as soon as the user select the user action table consisting list of actions are displayed.

➤ **Action table of Module1 of Application3(Global Personal Marketplace)-**

Action1	Display General Information	User
Action2	Register	USECASE
Action3	log on	
Action4	log off	
Action5	Review Account Status	
Action6	Inquiries	
Action7	Obtain Email Address	
Action8	Maintain User Account	

Fig:8. Action table of Module1 of Application3 (Global Personal Marketplace)-

**VII. RESULTS**

On supplying the requirement elicitation data, the tool is able to generate relevant and there by the requirement model. Now in this chapter we will give the results of such implementation in terms of usecases. Usecases can be displayed when button named “Usecases”.

**Usecase of Application3- Global Personal Marketplace, Module-1(User)**

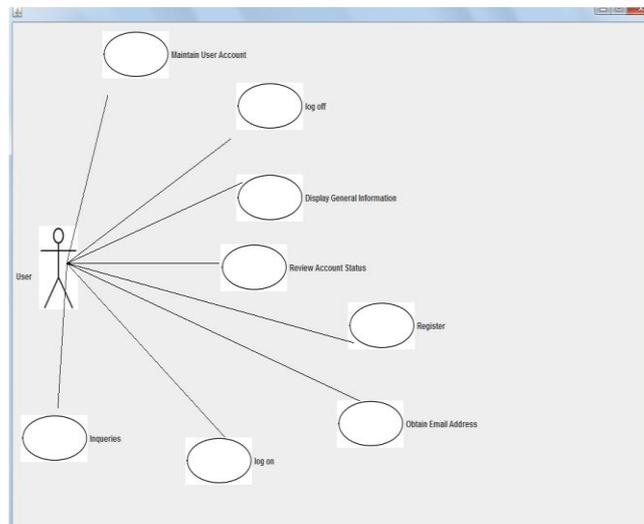


Fig-9: Usecase of Global Personal Marketplace(User).

**IX. CONCLUSION**

We have implemented the automation of Requirement Modeling. We have used requirement elicitation data for populating the data , this data is used as a basis for making event table, which will list every event in the system as a consequence invoking functionality of the system which is a result of requirement stated in requirement elicitation data, this event table can be abstracted in the form of action table which will display the action of any particular user, it reduces the effort of generating event table for each action, making each action listed in action table for a particular user, these actions are mapped into usecases wherever we have action in action table that is mapped into oval shaped usecases and wherever we have user it is mapped into actor shape on usecase frame . Thus this will ease the process of making usecases. Our automation has reduces the effort used for making usecases in tools like Rational Rose, which require lots of time for making usecases. Thus effort and time to make usecases by our automation has reduced. This tool for automation of usecases can work on any application, application module to make usecases for them. Tools like Rational Rose used for such modeling increase the effortas in [10] as you have to drag and drop usecases than label each actor and usecases, after that relationships are made with this tool but in our automation tool you have to provide requirement

elicitation data than the process of obtaining outcome as usecases is automated with just a button click. Thus this automation can be used to make usecases with less time and effort than other tools.

## References

- [1]. Distributed concurrent development System: An Object-Oriented Process Model, Mikio Aoyama, Business Switching Software Division, Fujitsu limited.
- [2]. The Construction Of An Object-Oriented Specification Model, Sangbum Lee and Doris L., Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803-4020.
- [3]. Enabling Automated Analysis Through the Formalization Object-Oriented Approach, Betty H.C. Chengy and Laura A. Campbell, *Michigan State University*
- [4]. A Systematic Approach to Object Development, Richard H. Dinger, Principal Engineer Boeing Computer Service.
- [5]. Quantitative Approaches in Object-Oriented Software Engineering, Mario Piattini<sup>1</sup>, Fernando Brito e Abreu<sup>2</sup>, Geert Poels<sup>3</sup>, and Houari A. Sahraoui<sup>4</sup> Universidad de Castilla-La Mancha, Ronda de Calatrava, 5, 13071, Ciudad Real, Spain.
- [6] Formal Verification of UML Diagrams: A First Step Towards Code Generation Jeffrey Smith<sup>1</sup>, Mieczyslaw Kokar<sup>2</sup> and Kenneth Baclawski<sup>2</sup> Mercury Computer Systems, Inc. 2 Northeastern University.
- [7] Transformation of Uml Class Diagram to Uml Sequence Diagram, Lavleen Kambow Department of Computer Science G.N.D.E.C, Ludhiana, India.
- [8] UML Diagrams: an aid to Database Design Specification: A Review Mohammad Ibrar Department of Computer Science, Metas Adventist College, Ranchi, India.
- [9] UML Activity Diagrams as a Workflow Specification Language Marlon Dumas and Arthur H.M. ter Hofstede Cooperative Information Systems Research Centre Queensland University of Technology GPO Box 2434, Brisbane QLD 4001, Australia
- [10] On the Contribution of UML Diagrams to Software System Comprehension Irit Hadar and Orit Hazzan, Technion – Israel Institute of Technology, Israel.
- [11] On the Contribution of UML Diagrams to Software System Comprehension, Irit Hadar and Orit Hazzan, Technion – Israel Institute of Technology, Israel
- [12] Object-Oriented Database Representation Through UML Dr. Vipin Saxena Associate Professor, Department of Computer Science Babasaheb Bhimrao Ambedkar University (A Central University), Vidya Vihar, Rae Bareilly Road.
- [13] Unified Modeling Language (UML) for Database Systems and Computer Applications Sunguk Lee\* Research Institute of Industrial Science and Technology, Pohang, Korea