

Case Study: Visualization of Interaction Modeling in Object Oriented Unified Modeling Language (UML)

Mr. Kamaldeep Singh, Er. Daljeet Singh

Guru Nanak Dev Engineering College, Ludhiana 141006, Punjab, India.

Abstract— Interaction particularly means the communication between Various types of Objects or Classes that how two or more elements interact with each other to model a particular Design. Modeling basically refers to create a Simple Prototype to provide a Model for Implementing various types of Structures in any field of Engineering for Development of Projects at a Higher level of Accuracy and to provide more and more Precise Output Product. Interaction Modeling do the job by providing us various means to achieve our goals in Proper Modeling.

Keywords—UML; Sequence diagram; Use case diagram; Abstraction Levels; Activity diagram; Interaction Modeling; Object Oriented UML Modeling; Scenarios; Sequence Diagram;

I. INTRODUCTION TO INTERACTION MODELING

Interaction modeling is a technique to describe the way of modeling of various objects or elements that communicate or interact with Each other via modeling. Although. Interaction particularly means the set of messages that are exchanged within a collaboration of various kinds of objects/entities. In Real world problems objects communicate with each other by sending or receiving messages, therefore whenever a group of objects collaborate to perform a specific task there will be always Interaction takes place among those objects.

II. INTERACTION DIAGRAMS

An interaction diagram includes various diagrams with which the concepts of Interaction modeling are defined and clarified these diagrams are generally created with a UML Compiler like Agro UML, Plant UML etc. Interaction diagrams are very helpful in reasoning about the flow of execution of a Process of Modeling, Model, and Block Diagram of a System. These Interaction Diagrams focus on that How a message is to be dispatched across the time of its Implementation. Also These Interaction Diagrams Describes the Structural Relationships between Various Objects that how these Relationships takes place between them.

A. Static Behaviour of a System

Static Behavior of a system is the Response of the System or we can say the Outer Static view point of a System when it is NOT in its Working or Operating State.

B. Dynamic Behaviour of a System

In order to model a particular the most important aspect to keep in mind is the dynamic behavior of the system. Dynamic

Behavior is the behavior is the behavior or we can say the how the system responds according to the given Instructions of the user while the system is in its Operating Environments or Operable State. Along with the sufficiency of the Static behavior is far more important aspect in the modeling of a system

III. ABSTRACTION LEVELS IN UML MODELING

Interaction modeling takes place at the various levels of Abstraction these levels have their own Different way to describe the Abstraction of the UML elements and Objects. In UML we have various types of Diagrams to model the Dynamic Nature of a System. These diagrams include:-

1. Use Case Model
2. Sequence Diagrams
3. Activity Diagrams

A. Use Case Model

A Use Case is an External Interface of a System. Use Case Models Provide us Information to understand the behavior of a System. These use cases can be created by Interacting and having Discussions with Intensive users who have deled with Such Working conditions and Patterns Earlier. A Use case Specifies the types and forms of a system's Functioning .mostly these use cases have some type of Relationships with other Use Cases and these Relationships can of the following types:-

(i) <<include>>:- this type of relations takes place when a use case takes place inside another use case.

(ii) <<extends>>:- this type of relations takes place When in a certain situation we have to extend one user case to make another use case from it at that time.

(iii) Generalization:- Generalization deals particularly with Inheritance among various Use Cases where one user Case acts as a Parent Class and another one is Derived from it and Termed as Child Class.

Here in the Use Case Diagram Figure1. A Simple example is shown about how a person goes for an online movie ticket booking program and what its options available to that user are. Although this diagrams shows the simplicity and conceptual views of use cases in the Unified Modeling Language.

To Clear the Concept of Use case look at the fig.below:-

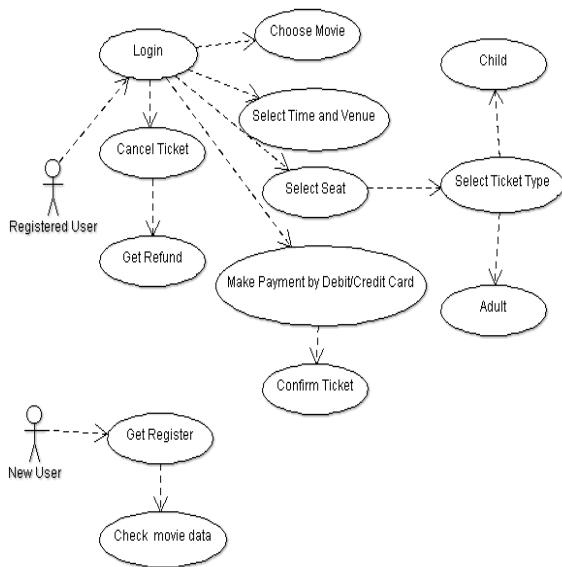


Fig. 1: Use Case Model for Online Ticket Booking

B. Sequence Models

A Sequence Model Describes the order and pattern of how various objects communicate with each other. Sequence models include a sequence of interactions between various objects. In basics Sequence Models includes two major portions:-

Scenarios:- A Scenario is a Sequence which takes place in the execution pattern in a System. For example- When a Customer check there mobile’s Sim Card balance by Dialing a Code like *444# and following the onscreen instructions sequence wise according to their execution order.

Sequence Diagrams:- Sequence Diagrams are those diagrams that describes the scenarios in a Sequence Diagram Structure i.e. via a block Diagram. a Sequence Diagram includes:-

- (i) Object:- Object is a entity that participates in a Process.
- (ii) Lifeline:-It is a Dotted line dropped from Each Object in a Diagram.
- (iii) Active:- Active is a Stage which shows the execution stage of a Object.
- (iv) Message:- A Message is basically a Command which is Given By a Object on any other Object.

Messages are of two types:-

- a. *Synchronous Message*- A Synchronous Message is used or takes place whenever the Sender who sends the message waits until the message is get received by the receiver and the receiver has finished the processing of the message only than the activation can takes place.
- b. *Instantaneous Message*- Instantaneous Messages are those messages where the sender don not have to wait for the receiver’s activation the time taken to Send and Receive message is Negligible or very less we can say that the activation takes place in Fractions of a Second. For

example: in a process method call the Horizontal lines between processes are working through Instantaneous Messages.

- (v) Return:- whenever there is a condition that one object commands another object at such circumstances Return is used and Return is generally denoted by Dotted Lines.
- (vi) Conditional: A Conditional is very Similar to a conditional Operator it is generally Enclosed inside the Square Brackets mostly Boolean or Logical Expressions are Sent or defined via Conditional.
- (vii) Iteration:- Iteration means sending a same message multiple times for sending iteration messages they are put down into a Square Brackets and are preceded by an asterisk (*) and the statement or expression written within the Brackets defines the rule of iteration.
- (viii) Deletion:-for deleting a particular portion in a sequence Diagram an “X” is used which actually indicates the termination of an Object or an Element.

Important points to create an efficient Sequence Diagram

- (i) One must prepare Scenario for Every Particular Use Case.
- (ii) Always Abstract the Scenarios into Sequence Diagrams.
- (iii) Try to Break the Complex or Tough Interactions into smaller and Easier Interactions.
- (iv) Always create own different diagrams for error conditions According to the Requirements.

To clearly understand the concepts of Sequence Diagram a general Block Diagram is shown below:-

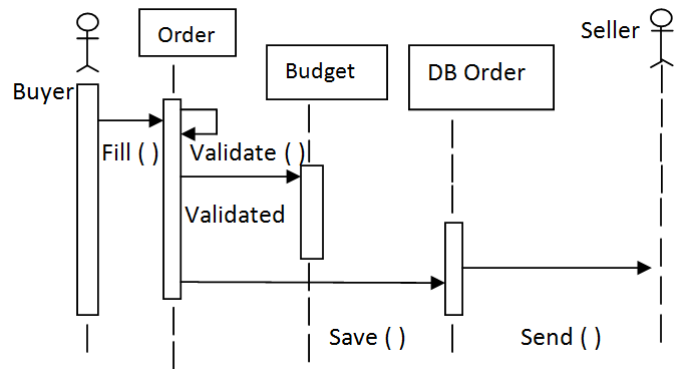


Fig.2: Sequence Diagram of Online Shopping

In the above Figure2 there is sequence wise dealing is shown between a Seller and a Buyer that how A Seller Sales a Set of Goods to a Buyer and How the validation of the order and the Payment is Done Step by Step Sequence Wise.

C. Activity Models

Activity models include the representation of the flow of execution in a particular activity manner. It includes its various conditional entities. To clearly understand the Concept of Activity Models a General block Diagram of an

activity model of on online movie ticket booking is shown below:-

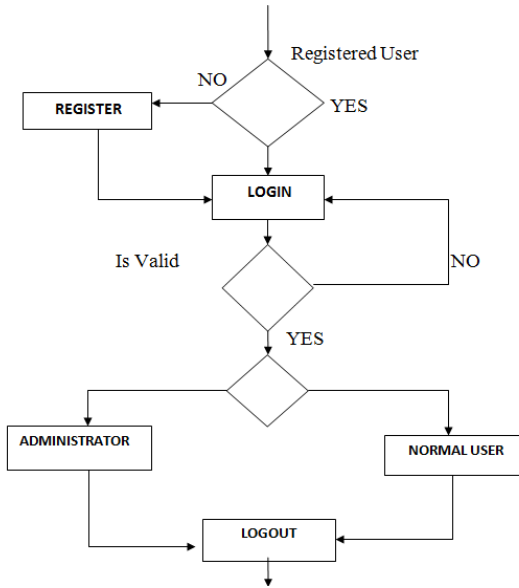


Fig.3: Activity Diagram User Login

In above Figure3 there is an activity diagram is shown which describes about the implementation and execution of a User account or we can say a user portal works for a website and how it has 2 types of users one are the administrators and other are the Regular or Normal users and how their accounts are proceeded accordingly.

Important points to create an efficient Activity Diagram

- Always make to clear to NOT mix-up the Activity Diagrams with Flowcharts.
- Always carry consistency at each level of your Activity Diagram
- Create Conditions very Carefully
- Always points about the Concurrent Activities
- Make your Activity Diagram as Executable As Possible

IV. REFERENCES

- [1]. Frederick Eddy, James Rumbaugh, Michael Blaha, William Premerlani, William Lorenson: Object-Oriented Modeling and Design, Pearson Education.
- [2]. James Rumbaugh, Michael R. Blaha: Object-Oriented Modeling and Design with UML, Pearson Education.
- [3]. Timothy C. Lethbridge, Robert Laganieri: Object Oriented Software Engineering, Practical Software Development using UML and Java, Tata McGraw-Hill edition.
- [4]. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY-Dreamtech India Pvt. Ltd.
- [5]. Preeti Verma, "Effect of different UML diagrams to evaluate the size metrics for different software projects", Global journal of computer science and technology software and engineering, vol. 15, issue. 8, version 1.0, February. 2015.

- [6]. Object oriented analysis design using UML Thakur publications,by Prof.Vikas Verma, Gania Goyal,Poonam Sharma.
- [7]. D. D'Souza, A. Sane, and A. Birchenough, "First-class Extensibility for UML – Packaging of Profiles, Stereotypes, Patterns", UML'99, LNCS 1723, Springer-Verlag, 265- 277, 1999.
- [8]. D. D'Souza and A. Wills, Objects, Components, and Frameworks with UML: The Catalysis Approach, Addison-Wesley, 1997.
- [9]. A. Eden, J. Gil, and A. Yehudai, "Precise Specification and Automatic Application of Design Patterns", ASE'97, IEEE Press, 1997.
- [10].G. Florijn, M. Meijers, P. van Winsen, "Tool Support for Object-Oriented Patterns", ECOOP'97, LNCS 1241, Springer-Verlag, 472-495, 1997.
- [11].M. Fontoura, "A Systematic Approach for Framework Development", Ph.D. Thesis, Computer Science Department, Pontifical Catholic University of Rio de Janeiro, Brazil (PUC-Rio), 1999.
- [12].M. Fontoura, L. Moura, S. Crespo, and C. Lucena, "ALADIN: An Architecture for Learningware Applications Design and Instantiation", Technical Report MCC34/98, Computer Science Department, Computer Science Department, Pontifical Catholic University of Rio de Janeiro, Brazil (PUC-Rio), 1998.
- [13].G. Froehlich, H. Hoover, L. Liu, and P. Sorenson, "Hooking into Object-Oriented Application Frameworks", ICSE'97, IEEE Press, 491-501, 1997.



Kamadeep Singh Received his Bachelor of Computer Applications Degree from Punjabi University Paitala in 2016 and currently Pursing his Master of Computer Applications from Guru Nanak Dev Engineering College Ludhiana his fields of Interest are Operational Research, UML, CSOA and C++.



Singh Daljeet received his B-Tech degree in Computer Science and Engineering from Punjab Technical University, Jalandhar College, B.C.E.T, Ludhiana, Punjab, India, in 2008, the M.Tech, degree in Computer Science and Engineering from Punjab Technical University, Jalandhar, College, Guru Nanak Dev Engineering College, Ludhiana, Punjab, India in year 2012, currently he is doing PhD in computer science and engineering. He is a Assistant Professor at present, with Department of Computer Science and Engineering, in Guru Nanak Dev engineering College. His research interests include Software Engineering, Software Metrics, UML, Object Oriented Paradigm, and Object Oriented Metrics. At present, He is engaged in Research of UML diagrams simplifications.