UML Modelling: The Rational E-Development

Er. Meenu Gupta 1, Dr. Rajeev Yadav 2, Minakshi Memoria
Assistant Professor in DCE Gurgaon , Associate Professor in GCEW Gurgaon, Associate Professor in DCE Gurgaon
1gupta.meenu5@gmail.com, 2rajeevhodcse@gmail.com, 3minakshimemoria@gmail.com

Abstract— Unified Modeling language (UML) is one of the important modeling languages used for the visual representation of the research problem. In this paper we used Rational Software architecture to design UML diagrams.

Keywords— Keywords: UML, Use case diagram, Class Diagram, Interaction diagram,

I. INTRODUCTION

UML is a language used to specify, visualize, and document the artifacts of an object-oriented system under development. “Language expresses the idea not methodology” and Modeling: Describing a software system at a high level of abstraction.

It is a powerful modeling language used to represent the research problems visually. Many research has been done on modelling problems by the use of UML, but limited research papers are reported in literature on applications of UML for the computer architecture research problems. By the use of UML, software and hardware architecture problems can be solved and performance can be judged after modelling of the problem. Real time system based on UML model is described by Selic and Rumbaugh[1]. The first represented of UML in the field of telecommunication sector is proposed by Holz [2]. The concept of UML was first invented by the Greddy Booch et al. [3]. The UML application is also proposed by the researchers for the web based application & one of the important papers on this is [4]. Various computer architecture models which can be used for the further research work are available in [5]. The latest research in 2007 on distributed computing is reported by the researchers [6]. OMG is an important active group for inventing the different versions of the UML.

II. ITERATIVE DEVELOPMENT PRODUCES AN EXECUTABLE

In Iterative and Incremental life cycle development precedes a series of iteration that evolve into the final system. Each iteration consists of one or more of the following components: business modeling, requirements, analysis, design, implementation, test and deployment [7]. The developer do not assume that all requirements are known at the beginning of the life cycle: indeed change in anticipated throughout all phases. This life cycle is a risk-mitigating process.

The releases are scheduled to ensure that the highest risks.

The result of this life cycle approach is less coupled with minimal investment. Each iteration result in an executable release.

Fig 1 Iterative and Incremental Development

I. Map of the Territory
I have often used the triangle for success as shown in fig 2 to explain the components need for a successful project.[11]. We need three fact: need, features and requirement.

2. **How many types of documents we have to prepare**

In unified modeling language we have six types of documents.

- a. Stake holder request document (Needs)
- b. Vision document /SRS/problem statement (feature)
- c. Use case specification document (functional)
- d. Supplementary specification document (Non-functional)
- e. Glossary document (Term-def)
- f. Requirement management plan

**Fig 3. Visual Modeling diagram**

3. **View Modeling with UML**

Why model visually? This question arises when we try to design our model. In this we capture structure and behaviour. Shows how system element fit together and promote unambiguous communication. UML is a one language for all practitioners. There are five views describe in UML modeling are: Use case, logical, component, deployment, process view.

III. **USE CASE DIAGRAM**
A use-case diagram is a set of use cases. A use case is a model of the interaction between External users of a software product (actors) and the software product itself more precisely, as an actor is a user playing a specific role describing a set of user scenarios capturing user requirements contract between end user and software developers[10]. A use case is a sequence of transactions performed by a system that yields a measurable result of values for a particular actor.

- **Actor**: A role that a user plays with respect to the system, including human users and other systems. e.g., inanimate physical objects (e.g. robot); an external system that needs some information from the current system.
- **Use case**: A set of scenarios that describing an interaction between a user and a system, including alternatives.
- **System boundary**: rectangle diagram representing the boundary between the actors and the system.

In association, it is a communication between an actor and a use case; Represented by a solid line. In Generalization, it is a relationship between one general use case and a special use case (used for defining special alternatives) Represented by a line with a triangular arrow head toward the parent use case.

Fig 4. Use case Diagram of Doctor appointment for patient
IV. CLASS DIAGRAM

A class is a description of a group of objects with common properties (attribute), common behavior (operations), common relationships to other objects and common semantics. Each class is represented by a rectangle subdivided into three compartments

- Name
- Attributes
- Operations

![Class Diagram](image)

Fig 5. Class diagram

V. ASSOCIATION MODEL TO IMPLEMENTATION

```java
Class Student {
    Course enroll[] = 4;
}

Class Course {
    Student have[] = 4;
}
```
Association that models the part-whole relationship Composition. Also models the part-whole relationship but, in addition, every part may belong to only one whole, and if the whole is deleted, so are the parts.

**Example**: A number of different chess boards: Each square belongs to only one board. If a chess board is thrown away, all 64 squares on that board go as well.

**V. INTERACTION DIAGRAM**

Interaction diagram show how objects interact with one another. UML supports two type of interaction:

1. Sequence Diagram
2. Collaboration Diagram

1. **Sequence Diagram**
   A sequence diagram shows object interaction arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams typically are associated with use case realization in logical view of the system development.

   ![Sequence Diagram](image)

   **Fig7. Sequence diagram**

2. **Collaboration Diagram**
Collaboration diagrams are equivalent to sequence diagrams. All the features of sequence diagrams are equally applicable to collaboration diagrams. Use a sequence diagram when the transfer of information is the focus of attention. Also use collaboration diagram when concentrating on the classes.

**Fig8. COLLABORATION DIAGRAM**

3. **State Diagram**

State Diagrams show the sequences of states an object goes through during its life cycle in response to stimuli, together with its responses and actions; an abstraction of all possible behaviors.

**Fig9. STATE DIAGRAM**

VI. **CONCLUSION**

From the above, it is concluded that the UML Class model is a powerful model used to depict the software development problems and the hardware problems. UML is a standardized specification language for object modeling. Several UML diagrams:
- use-case diagram: a number of use cases (use case models the interaction between actors and software)
- Class diagram: a model of classes showing the static relationships among them including association and generalization.
- Sequence diagram: shows the way objects interact with one another as messages are passed between them. Dynamic model
- State diagram: shows states, events that cause transitions between states. Another dynamic model reflecting the behavior of objects and how they react to specific event

VII. **REFERENCES**

UML Modelling: The Rational E-Development