Experiment No. 6

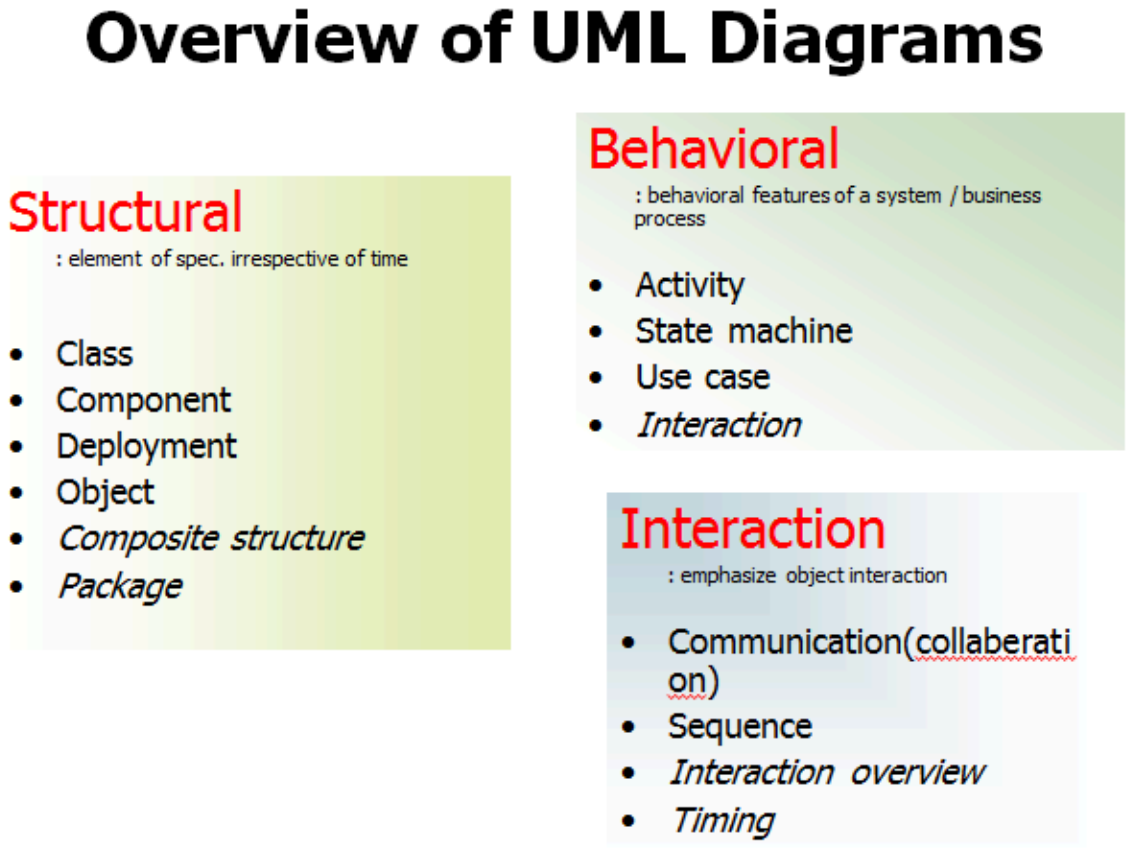
Title: Creation of Structural UML Diagrams

Batch: B–1 **Roll No.: 16010422234** **Experiment No.: 06**

Aim: Creation of UML Diagrams

Resources needed: IBM Rational Rose/Open Source UML Tool-star UML

Theory:



UML specification defines two major kinds of UML diagrams: **structure diagrams and behaviour diagrams.**

Structural diagrams show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts.

Behavioral diagrams show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time.

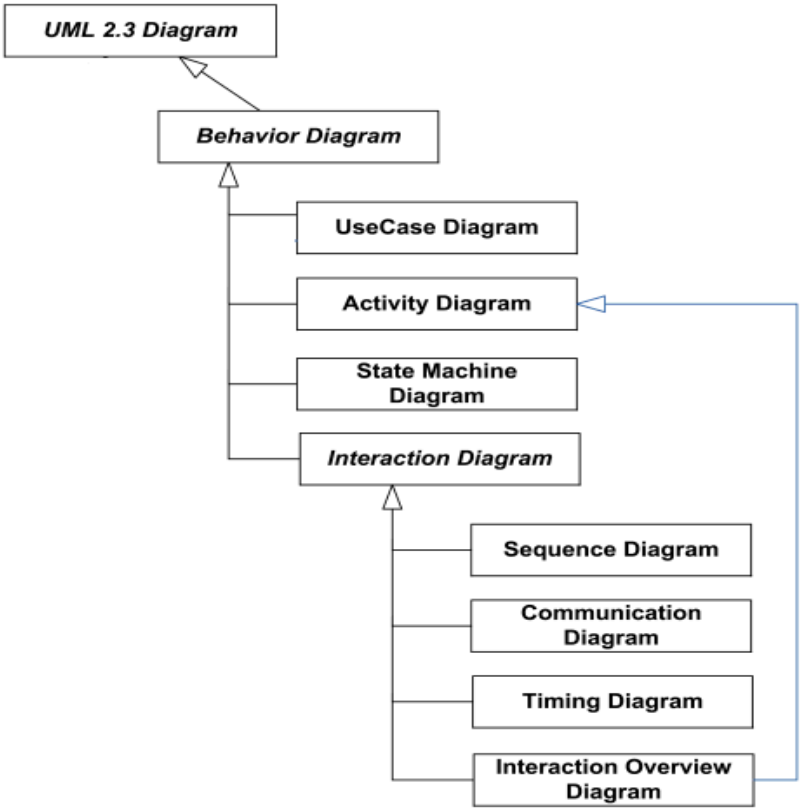


Figure 4.1 UML Behaviour Diagram

I. Structure Diagrams:

Structure diagram shows static structure of the system and its parts on different abstraction and implementation levels and how those parts are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts.
Structure diagrams are not utilizing time related concepts; do not show the details of dynamic behavior. However, they may show relationships to the behaviors of the classifiers exhibited in the structure diagrams.

1. Class diagram is a static structure diagram describing structure of a system on the (lowest) level of classifiers (classes, interfaces, etc.). It shows the system's classifiers, their attributes, and the relationships between classifiers.

2. Object diagram shows instances of classifiers and links (instances of associations) between them.

3. Package Diagram shows packages and dependencies between the packages. Models allow to show different views of a system, for example, as multi-layered (aka multi-tiered) application

4. Component Diagram shows components and dependencies between them. This type of diagram is used for Component-Based Development (CBD), to describe systems with Service-Oriented Architecture (SOA).

Composite Structure diagram could be used to show:

- Internal structure of a classifier
- Internal Structure diagrams show internal structure of a classifier - a decomposition of the classifier into its properties, parts and relationships.
- Behaviour of collaboration
- Collaboration use diagram shows objects in a system cooperating with each other to produce some behavior of the system.

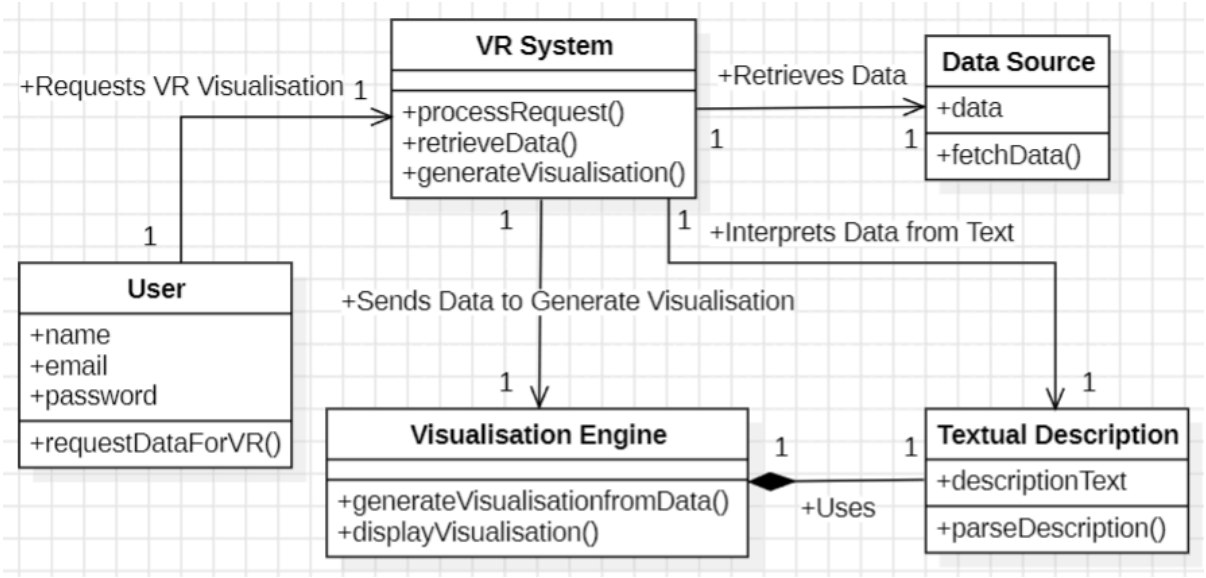
Deployment diagram shows execution architecture of a system that represents the assignment (deployment) of software artifacts to deployment targets (usually nodes).

Profile diagram is an auxiliary UML diagram which allows defining custom stereotypes, tagged values, and constraints. The Profile mechanism has been defined in UML for providing a lightweight extension mechanism to the UML standard. Profiles allow to adapt the UML meta model for different platforms (such as J2EE or .NET), or domains (such as real-time or business process modeling).
Profile diagrams were first introduced in UML 2.0.

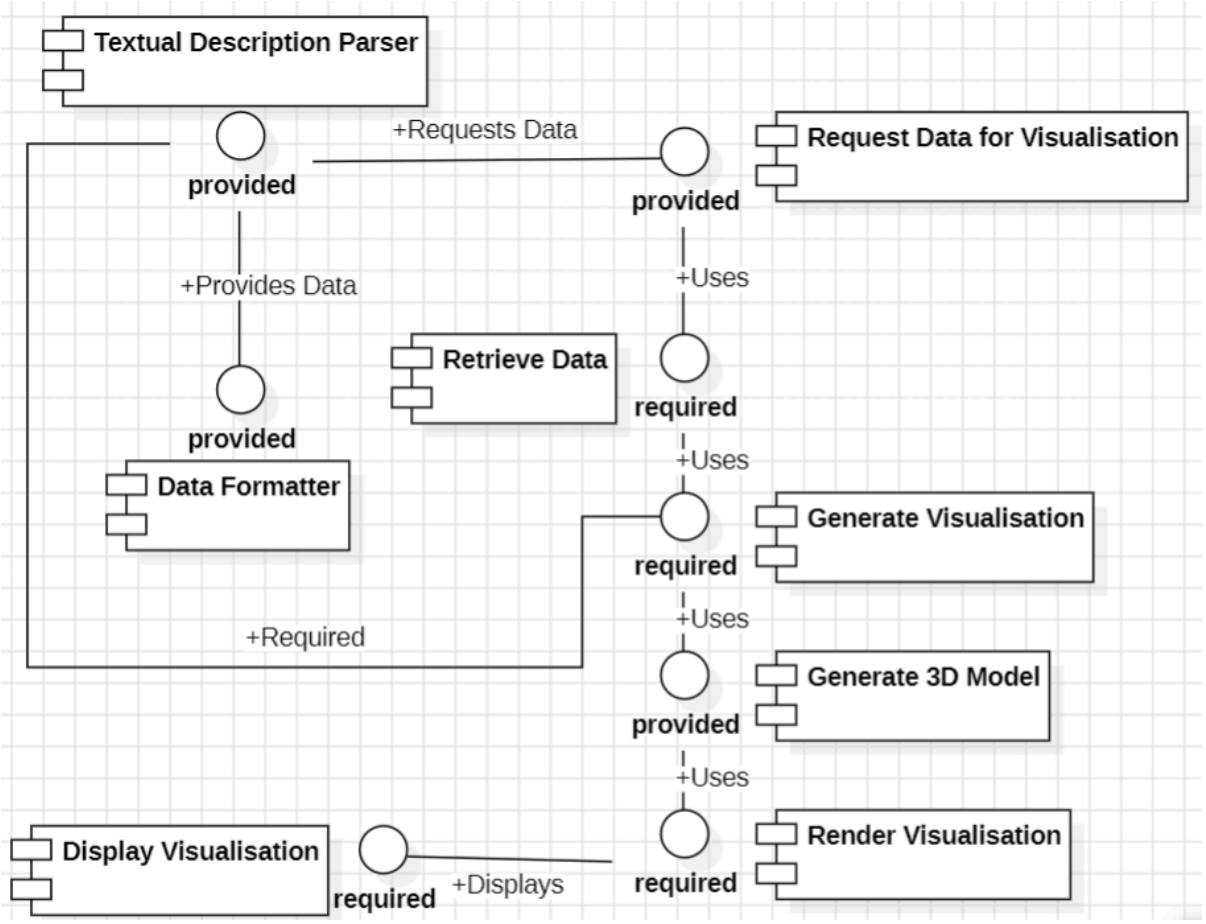
Procedure:
Prepare mentioned diagrams for chosen problems using Rational Rose/ any other Open Source UML tool.

Results: Printout of mentioned behavior diagrams

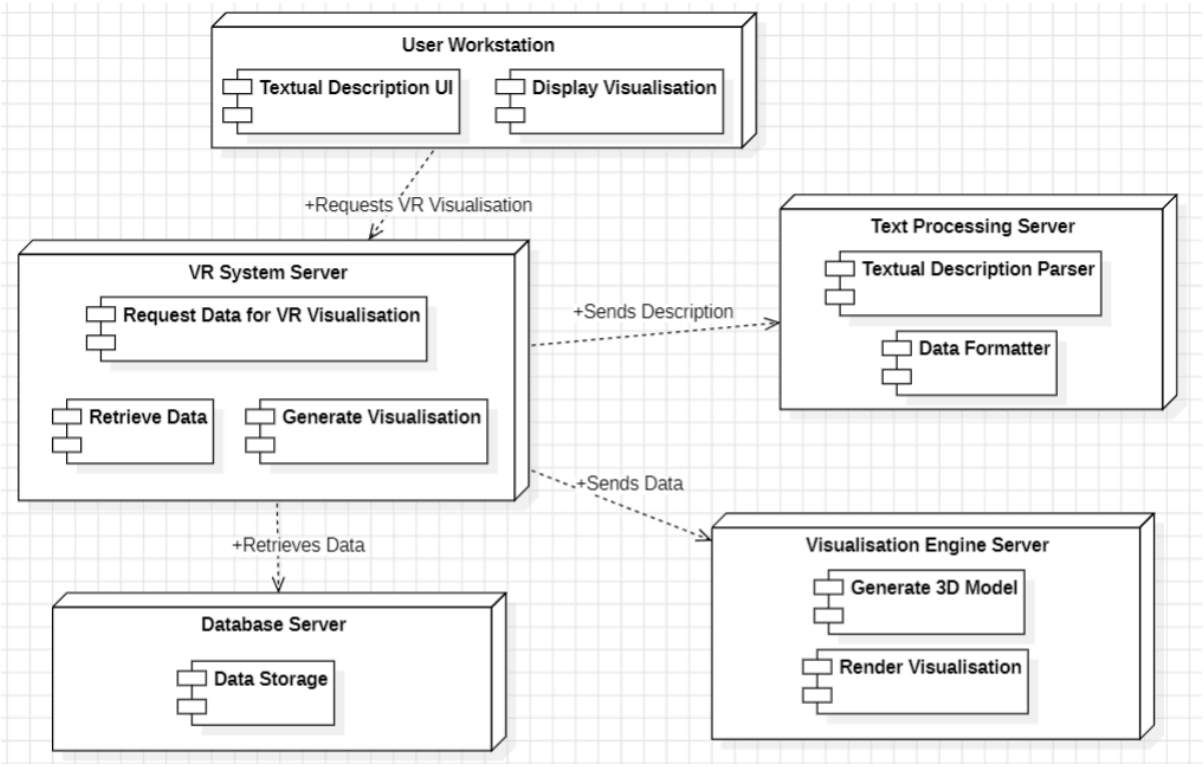
Class Diagram



Component Diagram



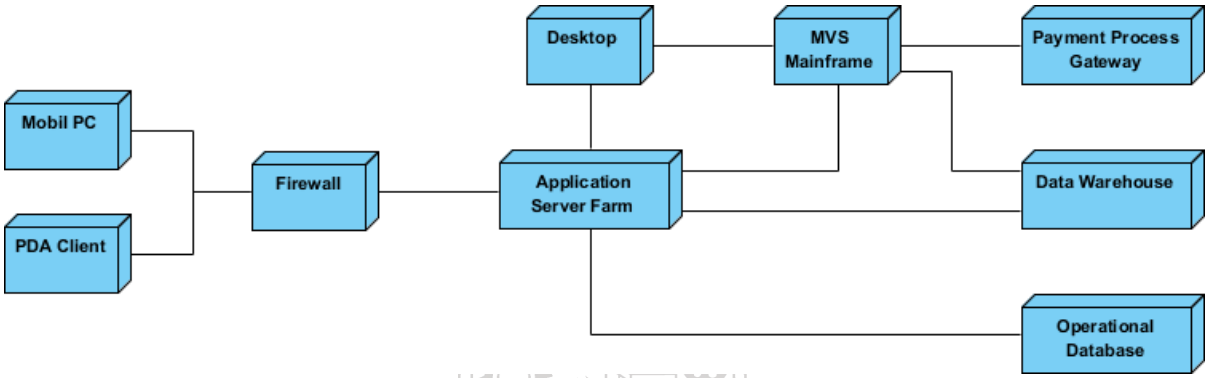
Deployment Diagram



Questions:

1. **Explain the difference between component diagram and deployment diagram.**
   - **Component Diagram:** Represents the organization and dependencies of software components in a system. It is used to model high-level structures like modules, libraries, and executables and how they interact. It focuses on the logical aspect of the system.
   - **Deployment Diagram:** Represents the physical deployment of software components on hardware nodes. It shows how different software elements are distributed across various devices, servers, or cloud environments. It focuses on the execution environment and system topology.

2. **Give an example of a deployment diagram for modelling a fully distributed system.**



This **Corporate Distributed System Deployment Diagram** represents a multi-tier architecture, where:

- Clients interact with the system through mobile or desktop devices.
- A secure **firewall** protects the internal infrastructure.
- An **Application Server Farm** processes requests.
- A **Mainframe** and **Payment Gateway** handle core business transactions.
- A **Data Warehouse** and **Operational Database** store and manage corporate data.

This architecture ensures scalability, security, and efficient data processing in a corporate environment.

Outcomes: CO3 – Demonstrate requirements, modeling and design of a system

---

Conclusion:

Through this experiment, we successfully created UML diagrams, including structure and behavioral diagrams. These diagrams help in visualizing the static and dynamic aspects of a system, providing a clear representation of the system's components, their relationships, and their interactions over time. By understanding and implementing UML diagrams, we gained insights into system modeling, which is crucial for software design and development.

Grade: AA / AB / BB / BC / CC / CD / DD

Signature of faculty in-charge with date

References:

Books/ Website

1.Michael Blaha, James Rumbaugh, "Object-Oriented Modeling and Design with UML", Prentice-Hall of India, 2nd Edition

2. Mahesh P. Matha, "Object-Oriented Analysis and Design using UML", Prentice-Hall of India

3. Timothy C Lethbridge, Robert Laganiere, "Object-Oriented Software Engineering – A practical software development using UML and Java", Tata McGraw-Hill, New Delhi.

4. http://www.uml-diagrams.org/uml-23-diagrams.html