



Experiment No. 5

Title: Creation of Behavioral UML Diagrams



Aim: Creation of Behavioral UML Diagrams

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

Theory

Overview of UML Diagrams

Structural

: element of spec. irrespective of time

- Class
- Component
- Deployment
- Object
- *Composite structure*
- *Package*

Behavioral

: behavioral features of a system / business process

- Activity
- State machine
- Use case
- *Interaction*

Interaction

: emphasize object interaction

- Communication(collaberation)
- Sequence
- *Interaction overview*
- *Timing*

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

Structure 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.

Behavior 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.

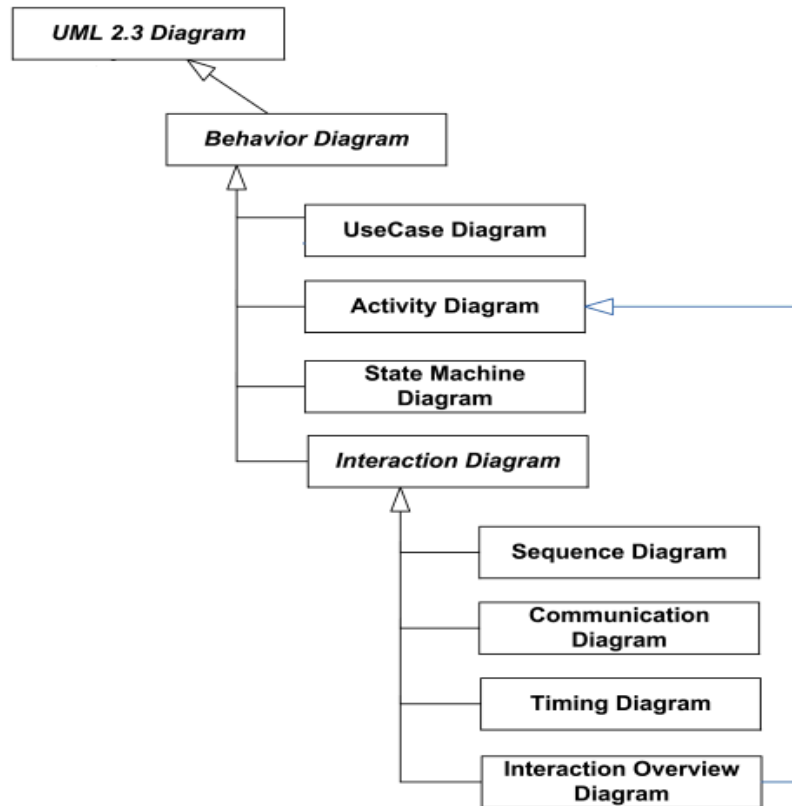


Figure 4.1 UML Behaviour Diagram

I. Behavior Diagram:

Behavior 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.

1. **Use case diagrams** are behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors) to provide some observable and valuable results to the actors or other stakeholders of the system(s).

2. **Interaction diagrams** include several different types of diagrams:

- **Sequence diagram** is the most common kind of interaction diagram, which focuses on the message interchange between lifelines (objects).
- **Interaction overview diagram** defines interactions through a variant of activity diagrams in a way that promotes overview of the control flow. Interaction overview diagrams focus on the overview of the flow of control where the nodes are interactions or interaction uses. The lifelines and the messages do not appear at this overview level.
- **Communication diagram (previously known as Collaboration Diagram)** is a kind of interaction diagram, which focuses on the interaction between lifelines where the architecture of the internal structure and how this corresponds with the message passing is central. The sequencing of messages is given through a sequence numbering scheme.

- **Timing diagrams** are used to show interactions when a primary purpose of the diagram is to reason about time. Timing diagrams focus on conditions changing within and among Lifelines along a linear time axis.

3. Collaboration diagram

A collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages.

We form a collaboration diagram by first placing the objects that participate in the interaction as they exist in a graph, and then add the links that connect these objects as arcs of this graph. Finally adorn these links with the messages that objects send and receive with sequence numbers.

Need of collaboration Diagram:

We use a collaboration diagram to describe a specific scenario. Numbered arrows show the movement of messages during the course of the scenario. A distinguishing feature of a collaboration diagram is that it shows the objects and their association with other objects in the system apart from how they interact with each other. The association between objects is not represented in a sequence diagram.

Elements of collaboration Diagram:

A sophisticated modelling tool can easily convert a collaboration diagram into a sequence diagram and the vice versa. Hence, the elements of a Collaboration diagram are essentially the same as that of a sequence diagram. UML State machine diagram and activity diagram are both behavioural diagrams but have different emphases.

4. Activity diagram shows sequence and conditions for coordinating lower-level behaviors, rather than which classifiers own those behaviors. These are commonly called control flow and object flow models.

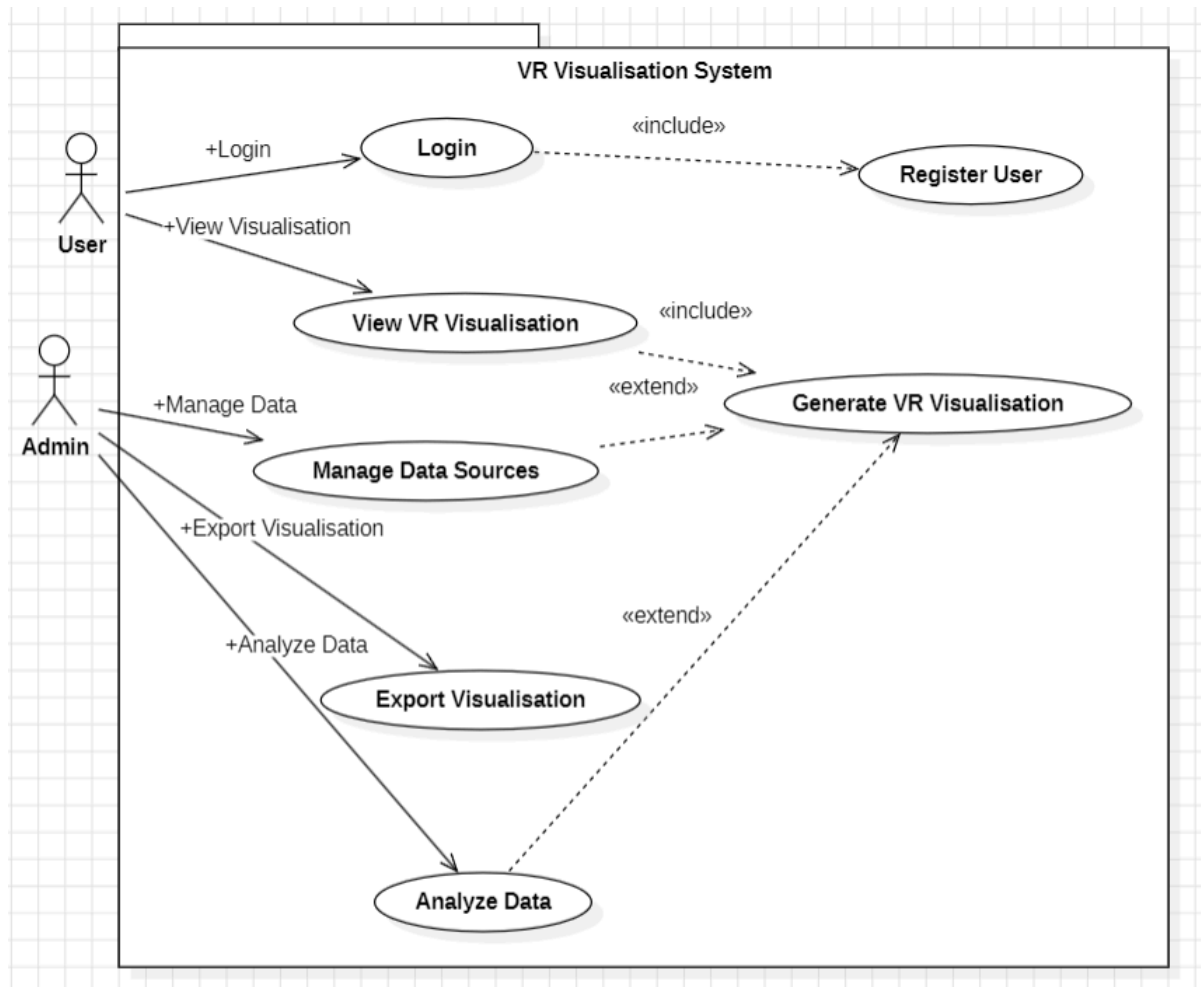
5. State machine diagram is used for modelling discrete behavior through finite state transitions. In addition to expressing the behavior of a part of the system, state machines can also be used to express the usage protocol of part of a system. These two kinds of state machines are referred to as behavioral state machines and protocol state machines.

Procedure:

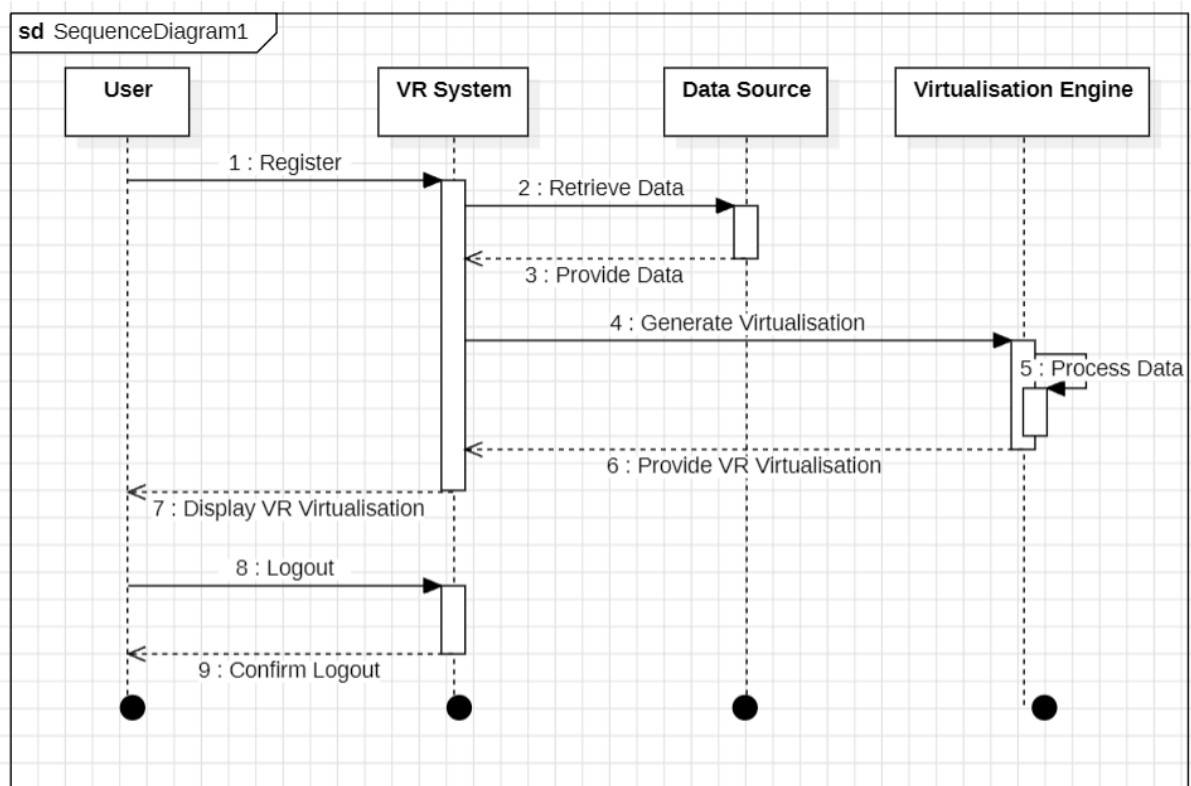
Prepare mentioned diagrams for chosen problems using Rational Rose/any other Open Source UML tool.

Results: Printout of mentioned behavior diagrams

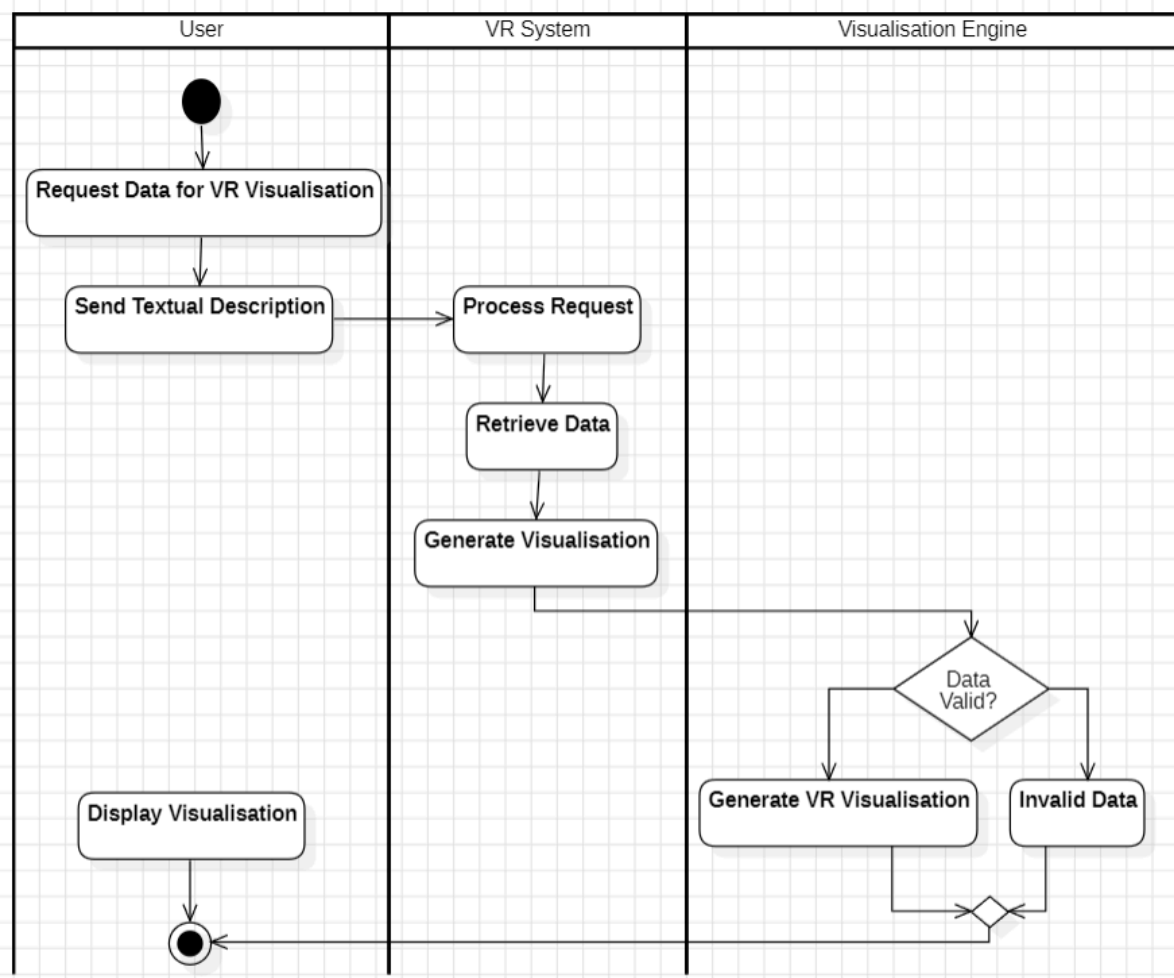
Use Case Diagram



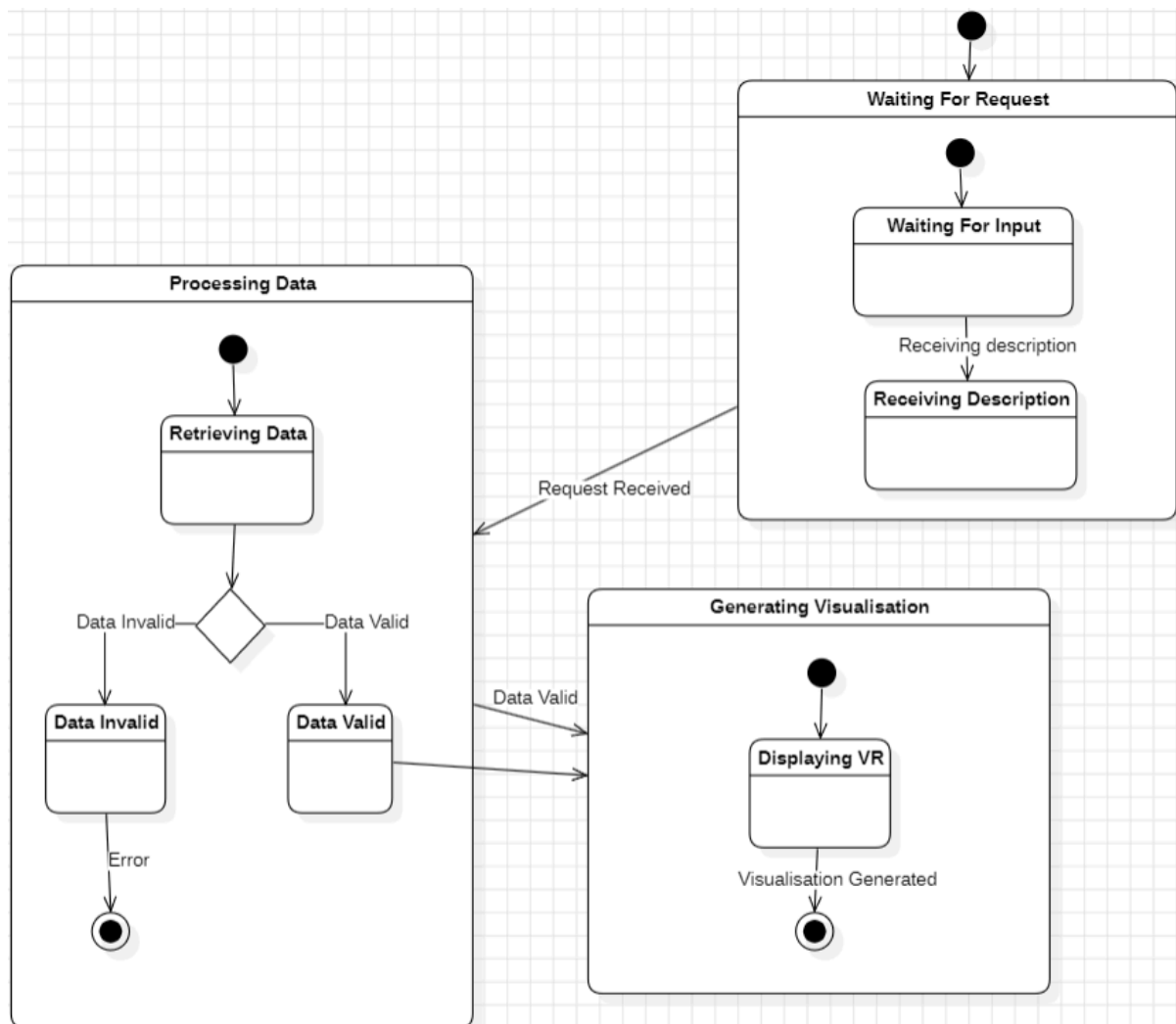
Sequence Diagram



Activity Diagram



State Machine Diagram



Questions:

1. In a use case diagram, relationships between different actors are normally shown.

True (Ans.) False

2. A Communication diagram specifies a scenario.

True (Ans.) False

3. Explain the difference between activity diagram and sequence diagram.

- An activity diagram focuses on the flow of activities and represents the sequence of processes, decision points, and conditions. It is useful for modeling workflows and business processes.
- A sequence diagram focuses on message exchange between objects over time, showing how interactions take place and the order of events in a particular scenario.

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

Conclusion:

In this experiment, we explored different behavioral UML diagrams, including use case diagrams, interaction diagrams (sequence, communication, and timing diagrams), collaboration diagrams, activity diagrams, and state machine diagrams. These diagrams help model the dynamic behavior of a system, illustrating how objects interact over time. By using UML tools like Rational Rose or StarUML, we created behavior diagrams for selected problems, demonstrating an understanding of system modeling and design.

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 Laganier, "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>
-