MODULE 1: Perform Object-Oriented Analysis and Design in Java Technology

Week 3

Unit 1: Transition Analysis to Design using Interaction Diagrams

Most Essential Learning Competencies: *At the end of the course, you must be able to:*

1. Discussand define unified modeling language
2. Explain the characteristics of UML
3. Enumerate the types of unified modeling language

Reading Activity

The Unified Modeling Language

The Unified Modeling Language (UML) is a graphical language for OOAD that gives a standard way to write a software system's blueprint. It helps to visualize, specify, construct, and document the artifacts of an object-oriented system. It is used to depict the structures and the relationships in a complex system.

Brief History

It was developed in 1990s as an amalgamation of several techniques, prominently OOAD technique by Grady Booch, OMT (Object Modeling Technique) by James Rumbaugh, and OOSE (Object Oriented Software Engineering) by Ivar Jacobson. UML attempted to standardize semantic models, syntactic notations, and diagrams of OOAD.

Systems and Models in UML

System − A set of elements organized to achieve certain objectives form a system. Systems are often divided into subsystems and described by a set of models.

Model − Model is a simplified, complete, and consistent abstraction of a system, created for better understanding of the system.

View − A view is a projection of a system's model from a specific perspective.

Conceptual Model of UML

The Conceptual Model of UML encompasses three major elements −

- Basic building blocks
- Rules
- Common mechanisms

Characteristics of UML

1. It is a generalized modeling language.
2. It is different from software programming languages such as Python, C, C++, etc.
3. It is a pictorial language which can be used to generate powerful modeling elements.

4. It is related to object-oriented designs and analysis.
5. It has unlimited applications even outside the software industry. It can be used to visualize the workflow of a factory.

Basic Building Blocks

The three building blocks of UML are −

- Things
- Relationships
- Diagrams

Things

There are four kinds of things in UML, namely −

- Structural Things − These are the nouns of the UML models representing the static elements that may be either physical or conceptual. The structural things are class, interface, collaboration, use case, active class, components, and nodes.
- Behavioral Things − These are the verbs of the UML models representing the dynamic behavior over time and space. The two types of behavioral things are interaction and state machine.
- Grouping Things − They comprise the organizational parts of the UML models. There is only one kind of grouping thing, i.e., package.
- Annotational Things − These are the explanations in the UML models representing the comments applied to describe elements.

Relationships

Relationships are the connection between things. The four types of relationships that can be represented in UML are −

- Dependency − This is a semantic relationship between two things such that a change in one thing brings a change in the other. The former is the independent thing, while the latter is the dependent thing.
- Association − This is a structural relationship that represents a group of links having common structure and common behavior.
- Generalization − This represents a generalization/specialization relationship in which subclasses inherit structure and behavior from super-classes.
- Realization − This is a semantic relationship between two or more classifiers such that one classifier lays down a contract that the other classifiers ensure to abide by.

Diagrams

A diagram is a graphical representation of a system. It comprises of a group of elements generally in the form of a graph. UML includes nine diagrams in all, namely −

- Class Diagram

- Object Diagram
- Use Case Diagram
- Sequence Diagram
- Collaboration Diagram
- State Chart Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram

Rules

UML has a number of rules so that the models are semantically self-consistent and related to other models in the system harmoniously. UML has semantic rules for the following −

- Names
- Scope
- Visibility
- Integrity
- Execution

Common Mechanisms

UML has four common mechanisms −

- Specifications
- Adornments
- Common Divisions
- Extensibility Mechanisms

Specifications

In UML, behind each graphical notation, there is a textual statement denoting the syntax and semantics. These are the specifications. The specifications provide a semantic backplane that contains all the parts of a system and the relationship among the different paths.

Adornments

Each element in UML has a unique graphical notation. Besides, there are notations to represent the important aspects of an element like name, scope, visibility, etc.

Common Divisions

Object-oriented systems can be divided in many ways. The two common ways of division are −

- Division of classes and objects − A class is an abstraction of a group of similar objects. An object is the concrete instance that has actual existence in the system.
- Division of Interface and Implementation − An interface defines the rules for interaction. Implementation is the concrete realization of the rules defined in the interface.

Extensibility Mechanisms

UML is an open-ended language. It is possible to extend the capabilities of UML in a controlled manner to suit the requirements of a system. The extensibility mechanisms are −

- Stereotypes − It extends the vocabulary of the UML, through which new building blocks can be created out of existing ones.
- Tagged Values − It extends the properties of UML building blocks.
- Constraints − It extends the semantics of UML building blocks.

UML defines specific notations for each of the building blocks.

Class

A class is represented by a rectangle having three sections −

- the top section containing the name of the class
- the middle section containing class attributes
- the bottom section representing operations of the class

The visibility of the attributes and operations can be represented in the following ways −

- Public − A public member is visible from anywhere in the system. In class diagram, it is prefixed by the symbol '+'.
- Private − A private member is visible only from within the class. It cannot be accessed from outside the class. A private member is prefixed by the symbol '–'.
- Protected − A protected member is visible from within the class and from the subclasses inherited from this class, but not from outside. It is prefixed by the symbol '#'.

An abstract class has the class name written in italics.

Example − Let us consider the Circle class introduced earlier. The attributes of Circle are x-coord, y-coord, and radius. The operations are findArea(), findCircumference(), and scale(). Let us assume that x-coord and y-coord are private data members, radius is a protected data member, and the member functions are public.

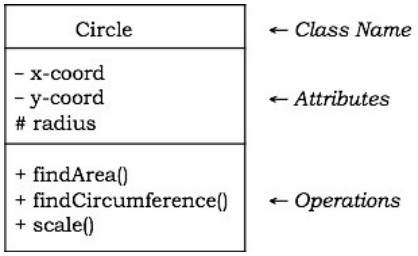

Figure 3.1 The figure gives the diagrammatic representation of the class.

Object

An object is represented as a rectangle with two sections −

- The top section contains the name of the object with the name of the class or package of which it is an instance of. The name takes the following forms −
  - object-name − class-name
  - object-name − class-name :: package-name
  - class-name − in case of anonymous objects
- The bottom section represents the values of the attributes. It takes the form attribute-name = value.
- Sometimes objects are represented using rounded rectangles.

Example − Let us consider an object of the class Circle named c1. We assume that the center of c1 is at (2, 3) and the radius of c1 is 5.

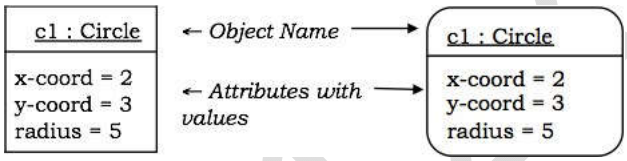

Figure 3.2: The figure depicts the object.

Component

A component is a physical and replaceable part of the system that conforms to and provides the realization of a set of interfaces. It represents the physical packaging of elements like classes and interfaces.

Notation − In UML diagrams, a component is represented by a rectangle with tabs as shown in the figure below.

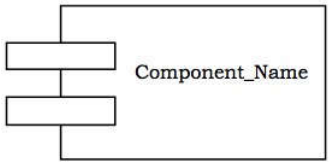

Figure 3.3: The figure a component is represented by a rectangle with tabs.

Interface

Interface is a collection of methods of a class or component. It specifies the set of services that may be provided by the class or component.

Notation − Generally, an interface is drawn as a circle together with its name. An interface is almost always attached to the class or component that realizes it.

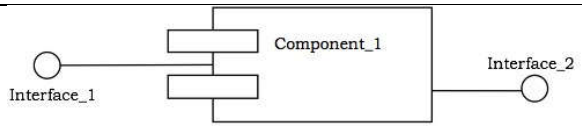

Figure 3.4: The figure gives the notation of an interface.

Package

A package is an organized group of elements. A package may contain structural things like classes, components, and other packages in it.

Notation – Graphically, a package is represented by a tabbed folder. A package is generally drawn with only its name. However, it may have additional details about the contents of the package.

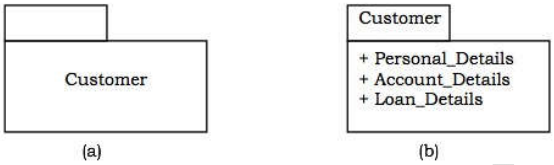

Figure 3.5: The figure illustrates notation.

Relationship

The notations for the different types of relationships are as follows –

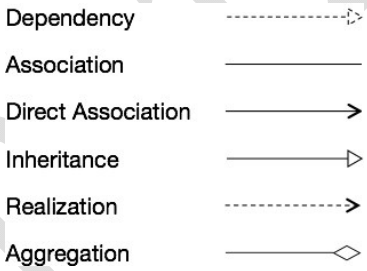

Figure 3.6: The different types of relationships.

Usually, elements in a relationship play specific roles in the relationship. A role name signifies the behavior of an element participating in a certain context.

Example – The first figure shows an association between two classes, Department and Employee, wherein a department may have a number of employees working in it. Worker is the role name. The '1' alongside Department and '*' alongside Employee depict that the cardinality ratio is one–to–many. The second figure portrays the aggregation relationship, a University is the "whole–of" many Departments.

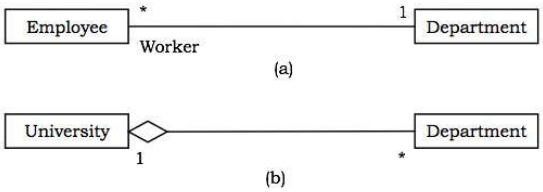

Figure 3.7: The figures show examples of different relationships between classes.

Read Additional Resources

1. **Unified Modeling Language**
   Week3_Unified Modeling Language.pdf
2. **Unified Modeling Language**
   Week3_Unified Modeling Language UML.pdf

Watch Video Resources

1. **Unified Modeling Language**
   Week3_Unified Modeling Language.mp4
2. **Types of Unified Modeling Language**
   Week3_Types of Unified Modeling Language.mp4

Self-Check

Quiz 3.1

Instructions: Write your answer on the Answer Sheet (AS) provided in this module.

1. Enumerate the characteristics of UML (5-points).
2. Different types of UML diagrams and give example. (15-points).

Internet References

1. https://www.youtube.com
2. https://www.tutorialspoint.com
3. https://www.youtube.com/watch?v=bb6FJWt7ZE4
4. https://www.youtube.com/watch?v=8CBnAmYnwk0