

UML Overview:

UML is a general purpose modeling language. It was initially started to capture the behavior of complex software and non software system and now it has become an OMG standard.

UML provides elements and components to support the requirement of complex systems. UML follows the object oriented concepts and methodology. So object oriented systems are generally modeled using the pictorial language.

UML diagrams are drawn from different perspectives like design, implementation, deployment etc.

At the conclusion UML can be defined as a modeling language to capture the architectural, behavioral and structural aspects of a system.

Objects are the key to this object oriented world. The basic requirement of object oriented analysis and design is to identify the object efficiently. After that the responsibilities are assigned to the objects. Once this task is complete the design is done using the input from analysis.

The UML has an important role in this OO analysis and design, The UML diagrams are used to model the design. So the UML has an important role to play.

UML notations:

UML notations are the most important elements in modeling. Efficient and appropriate use of notations is very important for making a complete and meaningful model. The model is useless unless its purpose is depicted properly.

So learning notations should be emphasized from the very beginning. Different notations are available for things and relationships. And the UML diagrams are made using the notations of things and relationships. Extensibility is another important feature which makes UML more powerful and flexible.

UML Diagrams:

Diagrams are the heart of UML. These diagrams are broadly categorized as structural and behavioral diagrams.

- Structural diagrams are consists of static diagrams like class diagram, object diagram etc.
- Behavioral diagrams are consists of dynamic diagrams like sequence diagram, collaboration diagram etc.

The static and dynamic nature of a system is visualized by using these diagrams.

Class diagrams:

Class diagrams are the most popular UML diagrams used by the object oriented community. It describes the objects in a system and their relationships. Class diagram consists of attributes and functions.

A single class diagram describes a specific aspect of the system and the collection of class diagrams represents the whole system. Basically the class diagram represents the static view of a system.

Class diagrams are the only UML diagrams which can be mapped directly with object oriented languages. So it is widely used by the developer community.

Object Diagram:

An object diagram is an instance of a class diagram. So the basic elements are similar to a class diagram. Object diagrams are consists of objects and links. It captures the instance of the system

at a particular moment.

Object diagrams are used for prototyping, reverse engineering and modeling practical scenarios.

Component Diagram:

Component diagrams are special kind of UML diagram to describe static implementation view of a system. Component diagrams consist of physical components like libraries, files, folders etc.

This diagram is used from implementation perspective. More than one component diagrams are used to represent the entire system. Forward and reverse engineering techniques are used to make executables from component diagrams.

Deployment Diagram:

Component diagrams are used to describe the static deployment view of a system. These diagrams are mainly used by system engineers.

Deployment diagrams are consists of nodes and their relationships. An efficient deployment diagram is an integral part of software application development.

Use Case Diagram;

Use case diagram is used to capture the dynamic nature of a system. It consists of use cases, actors and their relationships. Use case diagram is used at a high level design to capture the requirements of a system.

So it represents the system functionalities and their flow. Although the use case diagrams are not a good candidate for forward and reverse engineering but still they are used in a slightly differently way to model it.

Interaction Diagram:

Interaction diagrams are used for capturing dynamic nature of a system. Sequence and collaboration diagrams are the interaction diagrams used for this purpose.

Sequence diagrams are used to capture time ordering of message flow and collaboration diagrams are used to understand the structural organization of the system. Generally a set of sequence and collaboration diagrams are used to model an entire system.

Statechart Diagram:

Statechart diagrams are one of the five diagrams used for modeling dynamic nature of a system. These diagrams are used to model the entire life cycle of an object. Activity diagram is a special kind of Statechart diagram.

State of an object is defined as the condition where an object resides for a particular time and the object again moves to other states when some events occur. Statechart diagrams are also used for forward and reverse engineering.

Activity Diagram:

Activity diagram is another important diagram to describe dynamic behaviour. Activity diagram consists of activities, links, relationships etc. It models all types of flows like parallel, single, concurrent etc.

Activity diagram describes the flow control from one activity to another without any messages. These diagrams are used to model high level view of business requirements.