

1. What is UML? Explain about the importance of modeling?

UML:

The unified modeling language 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.

Importance of modeling:

- * A model is a simplification of reality.
- * A model provides the blueprints of a system.
- * Models may encompass detailed plans, as well as more general plans.
- * We build models so that we can better understand the system we are developing.
- * Through modeling, we achieve four aims:

1. Models help us to visualize a system as it is (or) as we want it to be.

2. Models permit us to specify the structure (or) behaviour of a system.

3. Models give us a template that gives (or) guides us in constructing a system.

4. Models document the decisions we have made.

"We build models of complex system because we cannot comprehend such a system in its entirety."



2) What are the things available in UML?

things:

In things we have following.

* Structural

* Behavioural

* Grouping

* Annotation.

Structural things:

The nouns of UML models; usually the static parts of the System in question.

Class:

An abstraction of a set of things in the problem domain that have similar properties and/or functionality.

Interface:

A collection of operations that specify the services needed by a class component.

collaboration:

A collection of UML building blocks that work together to provide some functionality within the System.

Use case:

An abstraction of a set of functions that the System performs; a use case is 'realised' by a collaboration.

Active class:

A class whose instance is an active object; an active object is an object that owns a process (or thread).

Component:

A physical part of the System.

Node:

A physical element that exists at run-time and represents a Computational resource.

Behavioural things:

the verbs of UML models; usually the dynamic parts of the system in question.

Interaction:

Some behaviour constituted by messages exchanged among objects.

→ state machine:

A behaviour that specifies the sequence of states of an object goes through during its lifetime.

Grouping things:

the organisational part of the UML model; provides a higher level of abstraction.

Package:

A general purpose element that comprises UML-elements-structural, behavioural (or) even grouping things.

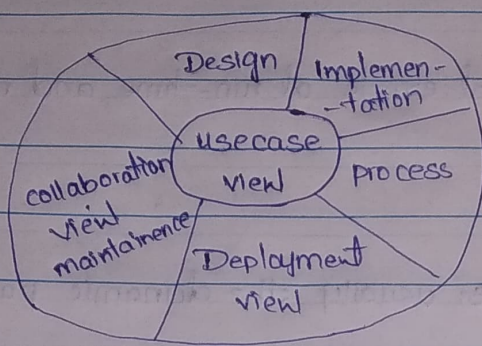
Annotational things:

The explanatory part of the UML model; adds information/meaning to the model elements.

Note:

A graphical notation, for attaching constraints and/or comments to elements of the model.

3) Explain about the architecture of UML?



* UML place an important role to defining different perspectives of a System.

* These perspectives are design implementation process and deployment.

* the centre is the use case view which connects all these four perspectives.

* A usecase represents the functionality of the System hence other perspectives are connected with use case.

Design:

* Design of the System consists of classes, interfaces and collaboration.

* UML provides class diagram, object diagram to support this.

Implementation:

* It defines the components assembled together to make a complete physical System.

* UML Component diagram is used to support the implementation perspectives.

process:

* process defines the flow of the System.

* Hence, the same elements as used in design are also used to support these perspectives.

Deployment View:

- * Deployment represents the physical nodes of the system that forms the hardware.
- * UML deployment diagram is used to support these perspectives.

4) Explain about the relationships available in UML?

Relationships in UML:

Articulates the meaning of the links between things.

Dependency:

A semantic relationship where a change in one thing causes a change in the semantics of the other thing.

Notation: ----->

(arrow-head points to the independent thing)

Association:

A structural relationship that describes the connection between two things.

Notation: _____

Generalization:

A relationship between a general thing and a more specific kind of that thing (called the "child" or "subclass"), such that the latter can substitute the former.

Notation: ———>

Realization:

A semantic relationship between two things where one specifies the behaviour to be carried out, and the other carries 'out' the behaviour.

"a collaboration realizes a usecase"



Notation: ----->

carrowl-head points to the thing being realized)