

Tutorial No - 2

Page No.	5
Date	

Q. Explain conceptual model of unified modeling language (UML) in detail.

UML is a standard language for specifying, visualising, constructing and documenting artifacts of software systems. UML stands for unified modeling language. UML can be described as a general purpose visual modeling language to visualize, specify, construct and document software system.

• Conceptual model of UML-

A conceptual model can be defined as a model which is made of concepts and their relationships. It helps to understand the entities in the real world and how they interact with each other. The conceptual model consists of three parts:

① Building blocks of UML:-

These are fundamental elements in UML. Building block of UML contains three types of elements.

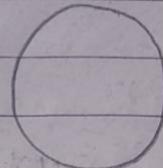
(i) Things :- A diagram can be viewed as graph containing vertices and edges. In UML, vertices are replaced by things and edges by relationship. There are four types of things in UML.

a) structural things : represents static aspects of software system. These are

seven structural things in UML.

- class :- It is collection of similar objects have similar attributes, behaviour } relationship.

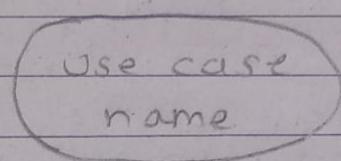
	ClassName	
class	attributeslist	
	operations list	



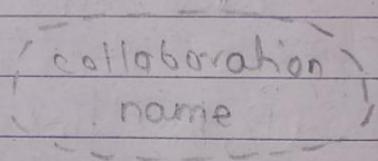
interface

- Interface:- It is collection of operation signature & or attribute definitions that ideally define cohesive set of behaviours.

- Use case :- A use case is collection of actions defining interactions bet" a role and system.



use case

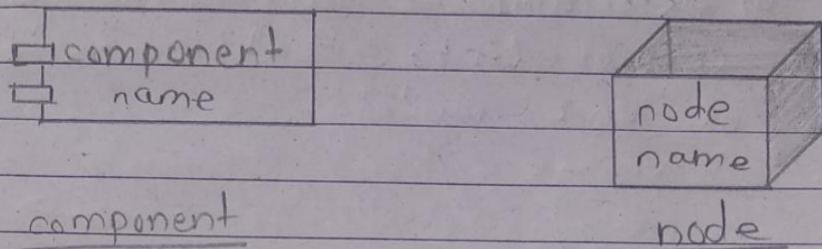


collaboration

- collaboration - A component It is collection of interactions among objects to achieve a goal.

- components - A component is physical & replacement part of a system.

- node - A node is physical element that exists at run time and represents computational resource.
- Active class - A class whose objects can initiate its own flow of control and work in parallel with other objects.



b) Behavioural things -- Represents dynamic aspect of software system. Behaviour of software system can be modelled as interactions or as sequence of state change.

c) Grouping things:- Elements which are used for organizing related things and relationships in models.

d) Annotational things - A symbol to display comments represented as rectangle with dog ear at top right corner.

ii] Relationships:- Things in a diagram are connected through relationships so relationship is connection between two or more things.

a) Dependency - It is relationship between two things in which change in one element also affects other.

dependency



b) Association - Association is basically set of links that connects elements of UML model.

--- Association ---

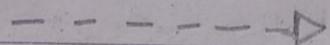
c) Generalization - It can be defined as relationship which connects specialised element with generalization element.

Generalization



d) Realization - It can be defined as relationship in which two elements are connected. One element describes some responsibility and other one implements them.

realization



Hello
World

Graphics

ex - Dependency.

company

Department

ex Association

Savings

account

Account

ex Generalization

HTML

parser

parser

ex Realization

Diagrams:- A diagram is collection of element often represented as graph consisting of vertices & edges joining these vertices. These vertices in UML are things & edges are relationships. There are 9 UML diagrams:

- ① class diagram.
- ② object diagram
- ③ use case diagram
- ④ component diagram
- ⑤ deployment diagram
- ⑥ sequence diagram
- ⑦ collaboration diagram
- ⑧ statechart diagram
- ⑨ activity diagram.

2] Rules (Semantics) :-

The rules of UML specify how UML's building blocks come together to develop diagram. The rules enables user to create well-formed models.

A well-formed model is self-consistent & also consistent with each other - UML has rules.

(i) Names - What element can be called as things, relationships and diagram.

(ii) scope - The context that gives specific meaning to a name.

(iii) visibility - How these names are seen & can be used by other names.

(iv) Integrity - How things properly relate to one another.

(v) Execution - what it means to simulate model.

3] Common mechanisms in UML -

There are four common mechanisms that apply throughout UML.

i) specifications - Behind every graphical notation in UML there is precise specification of details that element represents. ex- class icon is rectangle } it specifies name, attributes of class.

ii) Adoptions - The mechanism is UML which allows users to specify extra information with basic notation of an element is adoption.

In ex, the access specifiers
`+ (public)`, `# (protected)`
`- (private)` represent visibility of attributes which is extra representation.

(iii) Common to

Transaction
<code>+ execute()</code>
<code>+ rollback()</code>
<code># priority()</code>
<code>- timestamp()</code>

iii) common divisions :- In UML there is clear division between semantically related elements like :
 separation between class & object
 separation between an interface and its implementation.

iv) Extensibility Mechanisms - UML extensibility mechanisms allow to extend lang. in controlled way

- ① stereotypes : - extends vocabulary of uml.
- ② Tagged values - extends properties of uml building block.
- ③ constraint - extends semantics of uml like specifying new rules or modifying existing rules.