

Unit - 5:

## # Object Modeling Technique (OMT) in details:

- ⇒ OMT is an object oriented analysis, design and implementation methodology focuses on creating a model of objects from the real world and use that model to develop object-oriented software.
- developed by James Rumbaugh in around 1991.
- OMT is fast and approach for identifying and modeling all the objects making up a system.
- Details such as class, object, attributes, methods, inheritance, etc. can also be expressed easily.

## # OMT uses three kind of Models:

- Object Model : - describe the objects in the system and their relationship.
- Dynamic Model : - describe the interaction among objects in the system.
- Functional Model : - describe the data transformation of the system.

# Object Model : - object, - class, - attribute, - operation, - method, - link, - association, - multiplicity, - aggregation, - generalization, - inheritance, - module, - sheet, - metadata, - instantiation, - homomorphism.

# Dynamic Model : - events, - states, - state diagrams,

# Functional Model : - Action, - Activity, - Actor, - client, - constraints, - control flow, DFD, - data, - operation,

# OMT consist of four phases ; which can be performed iteratively :

1. Analysis: This phase involves the preparation of precise and correct model for the real world problems. performed by analyst showing important properties.
2. System Design: This phase determines all system architecture, concurrent tables and data storage. In simple word, high level architecture of the system is designed by the designer.
3. Object Design: In this phase the object designer build a design model containing data structure and algorithm for objects and class.
4. Implementation: In this phase, prepared design is converted into fully developed Software.

# OMT Analysis - consist of iterating following steps:

- generating a problem statement.
- building an object Model.
- building an dynamic model.
- building an functional Model.
- finding operations.

## # Responsibility Driven Design (RDD)!

RDD is an object oriented design technique that focuses on the responsibility of objects. It is a way of thinking about object oriented design that emphasizes the behaviour of objects rather than their data.

There are following four steps in RDD:

### 1. Identifying Responsibilities:

first step is to identify relation of each objects which can be done by asking question like :

- \* What does this object do?
- \* What actions the object must perform?
- \* What information the object needs to store?

### 2. Assigning Responsibilities:

can be done by analyzing:

- \* Objects expertise.
- \* Objects access to information.
- \* Objects relationship with other objects.

### 3. Encapsulating responsibilities:

Once the responsibilities are assigned, then those responsibilities needs to be encapsulated in the objects so that these responsibilities should be hidden from the rest of the system which improve modularity and maintainability.

4. Collaborating responsibility : each object should collaborate with each other to fulfill the responsibility assigned to them. So, those objects must communicate and share their information.

## # RDD vs. OMT:

### Responsibility Driven Design

	Object Modeling Technique.
<u>Focus:</u>	<ul style="list-style-type: none"> <li>1. Primarily focus on identifying and allocating responsibility to objects.</li> </ul>
	<ul style="list-style-type: none"> <li>① focuses on creating a visual model of the system capturing structure and relationships.</li> <li>② Provides a set of graphical notation to represent object, class and association.</li> </ul>

### Emphasis:

	<ul style="list-style-type: none"> <li>2. Emphasizes the behavioral aspect of objects, concentrating on what the objects do &amp; how they collaborate to achieve the system goal.</li> </ul>
	<ul style="list-style-type: none"> <li>③ emphasizes both structural &amp; behavioral aspects of object.</li> <li>It covers representation of object &amp; class as well as their interactions.</li> </ul>

### Design process:

	<ul style="list-style-type: none"> <li>4. Involves identifying responsibilities, assigning responsibilities, Encapsulating responsibilities and Collaborating responsibilities between objects.</li> </ul>
	<ul style="list-style-type: none"> <li>④ Involves creating models like class diagrams, State transition diagram and interaction diagram.</li> </ul>

Documentation

Application scope.

Notations:

## RDD

⑤ May not rely heavily on formal documentation rather focuses on capturing responsibility.

⑥ Detailed level of design focuses on designing methods, responsibilities, and collaboration of individual objects.

⑦ doesn't have specific graphical notations, often involves textual description and/or pseudo-code to define responsibility and methods.

## OMT.

⑤ It places strong emphasis on documentation through diagrams that represents various aspects of the system.

⑥ Higher level of design, where focuses on visualizing the overall system structure, class objects & their relationship.

⑦ Provide set of graphical notation such as class diagram, state transition diagram and object interaction diagram.

## # What is Object Oriented Analysis? Explain with eg.

- \* It is the process of identifying, understanding and modeling of the objects and their interaction in the system. The goal
- \* The goal of OOA is - clear and comprehensive understanding of problem domain and its requirement.
- \* Analysis phase will analyze, specify and define the system which is to be built
- \* Two different models are developed
  - Requirement model.
  - Analysis Model.

### Main Purpose:

- \* How to characterize a system.
- \* To know what are different relevant objects.
- \* How do they relate to each other.
- \* How to specify or model a problem to create effective design.
- \* Examine requirement; analyse their implication.

### Example:

#### Problem Statement:-

Design a library management system in which user can search, borrow, return books.

## 1. Identify Objects:

- \* Users: That uses the system with attributes.
- \* Books: represents books with attributes.
- \* library: represents library with attributes.
- \* Transaction: represents borrow, return.

## 2. Define Relationships:

- \* a user can borrow multiple books.
- \* a Book can be borrowed by multiple users.
- \* library manages the inventory of books.
- \* transaction involves user borrowing, returning books.

## 3. Define attributes and methods:

### \* Book:

Attributes: ISBN, author, title, etc.

methods:

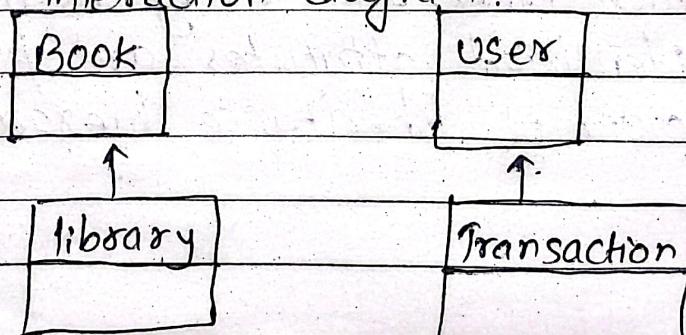
### \* User:

Attributes: Name, contact, email, address, etc.

method :

\*

## 4. Create object interaction diagram:



## Object Oriented Analysis / COAD- Yourdon (OOA):

- In OOA, an Analysis Model is developed to describe the functionality of the system.
- The idea in COAD-Yourdon design is to extend this Model with respect to processes (tasks)
- OOA according to COAD is a process of identifying the objects and classes that will be used in a System and describing their relationship and interaction.
- This method consists of 5 steps:
  1. finding classes and objects:
    - It specifies how classes and objects should be found within a problem domain and within context of system's responsibilities.

### 2. Identifying Structure:

It is done in two different ways

- generalization: Specializing general spec.
- Whole Part Structure.

### 3. Defining subjects.

### 4. Defining attributes:

It is done by identifying information and the association that should be associated with each other and every instance.

Identified attributes are placed in correct order of inheritance hierarchy.

### 3. Defining subjects:

- It is done by partitioning the class and objects model into larger units.
- Subjects are group of class and objects.

### 5. Defining Services:

- It means defining operation of classes.

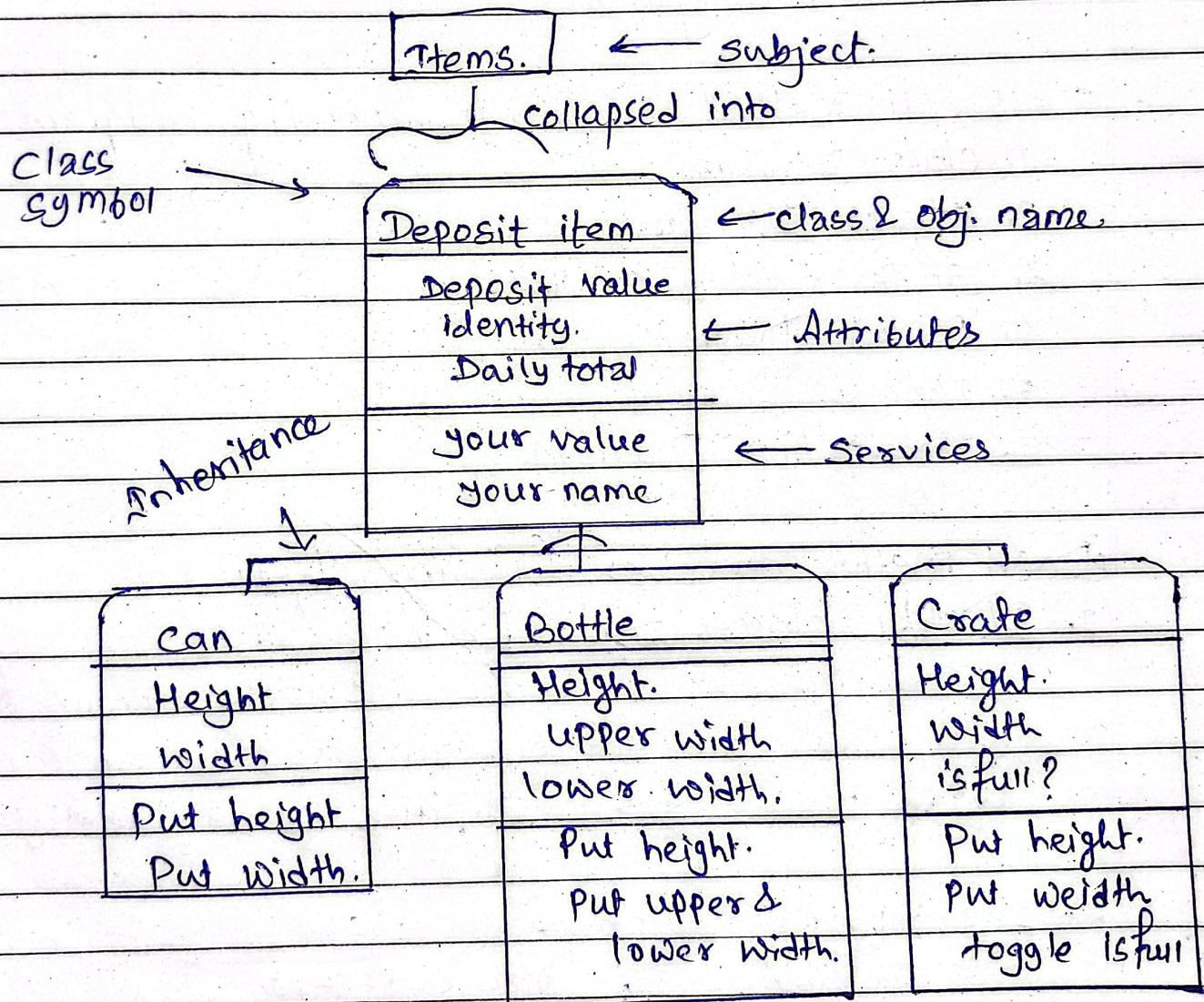


fig: OOA for a part of recycling system.

## Object Oriented Design | Booch (OOD/Booch)

- OOD according to Booch is a process of defining structure and behavior of soft objects.
- It is widely used object oriented method that helps us design our system using the object paradigm.
- It covers the analysis and design phase of developing object oriented system.
- The Booch Method consist of following diagram:
  1. Class Diagram.
  2. Object Diagram
  3. State transition Diagram.
  4. Module Diagram.
  5. Process Diagram.
  6. Interaction Diagram.

Booch method involves following steps:

Identify class and objects:

- involves finding key abstraction in problem space and important mechanism that offer the dynamic behaviour over several objects.

Identify class and object semantics:

- involves establishing the relationship between classes and objects identified earlier.

Identifying class and objects relationship:

- involves extending the previous activities to include the relationship between classes and objects and to identify how these interact with each other.

#### 4. Implementing class and objects:

- involves delivering into class and objects and determining of how to implement them.
- A decision is made on how to use particular programming language to implement this usage classes.

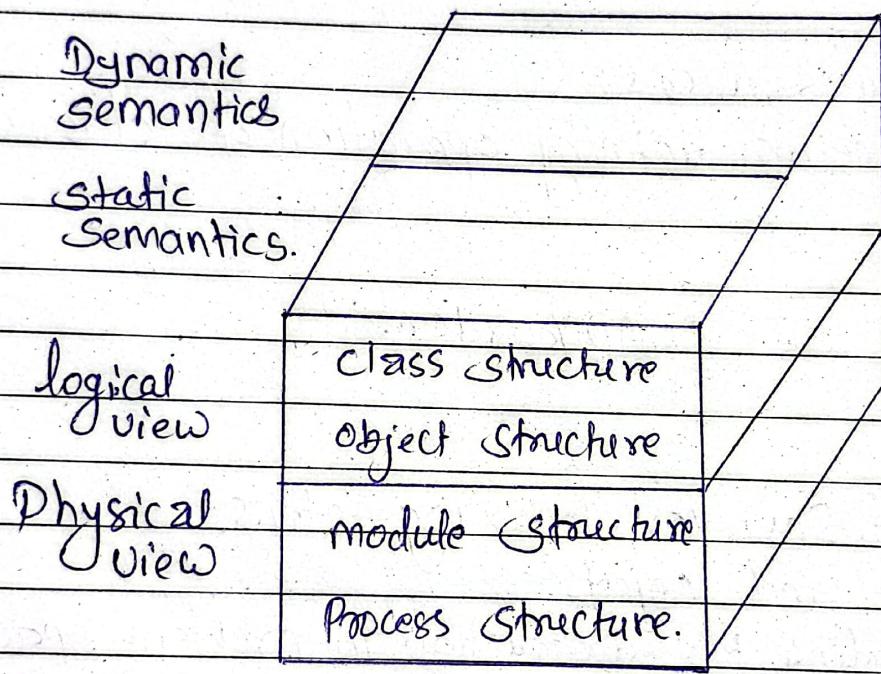


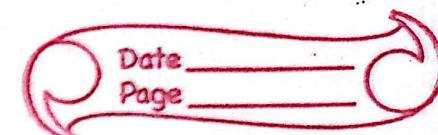
figure : Documentation aspect in OOD.

Booch consist of macro & micro development process.

1. macro development Process : It serve as controlling framework for micro process and its main concern is technical management of system. It consist of following steps:

- ① conceptualization.
- ② Analysis and Development of model.
- ③ Design or Create System Architecture
- ④ Evolution or Implementation.
- ⑤ Maintenance.

- Hood is a hierarchy of software units, based on identification of problem domain



## 2. Micro Development Process:

It is description of day to day activities operation by Software developer.

It consists of following steps:

- ① Identify objects and classes.
- ② Identify objects and classes Semantics.
- ③ Identify objects and classes relationship.
- ④ Implementing classes and objects.

## Hierarchical Object Oriented Design (HOOD)

- HOOD is a method of hierarchical decomposition of the design into software units, based on identification of objects, classes and operations reflecting problem domain entities.
- It is a detailed design method which starts after analysis and extends down to coding and testing.
- It has object oriented Paradigm.
  1. unit of decomposition is no more than action, but the Object.
  2. An object is an abstraction of a real world entities
- The hierarchy described in HOOD takes two phases:
  1. uses :- dependency of one object on another service.
  2. functional Decomposition:- objects split into child object to give functionality of parent object.

HOOD has four phases:

1. Problem definition:

- A statement of the problem is made to provide a content for the current object level.

2. Development of informal solution strategy:

- Solution to the Problem defined previously is outlined

3. formalization of the strategy:

- major concept of the informal solution strategy is extracted to formalize the sol.

#### 4. formalize the solution:

- It is done by developing a formal model of each identified objects.
- This is performed in five steps:
  - a. Identification of objects: it is done by extracting the nouns from the informal solution strategy and selecting appropriate one
  - b. Identification of operation: is done by extracting the verbs from the informal solution strategy.
  - c. Grouping objects and operations: involving attaching each operation to an appropriate object.
  - d. Graphical description is done by using HOOD graphical formalism.
  - e. Justification of design decision are performed by designer, who explains the reason for his decision.

(P.T.O)

## HOOD design:

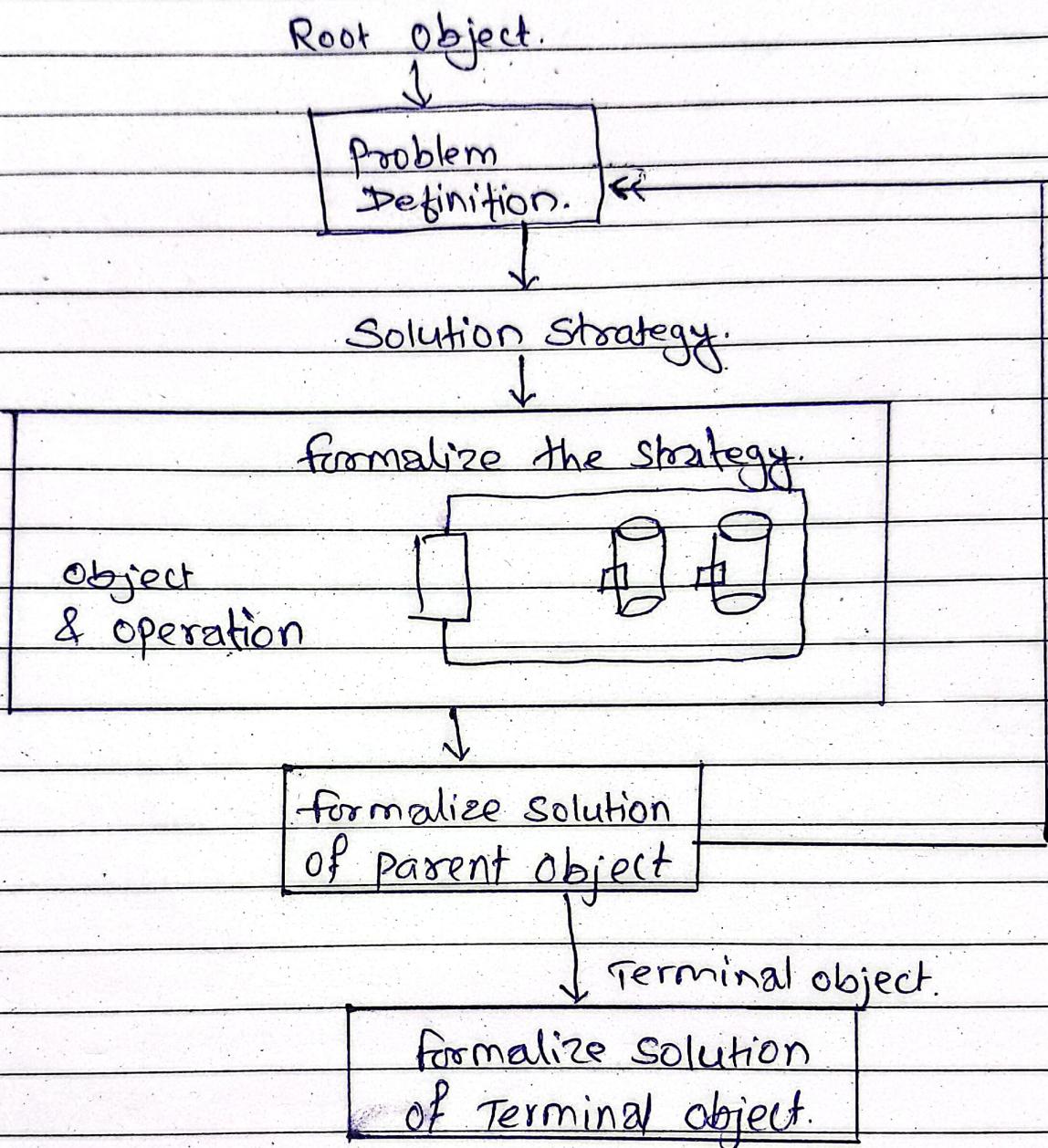


Figure: Basic design steps in HOOD.

## # Requirement Engineering from User requirement prospective

Developing a requirement model from user requirement involves the process of systematically capturing, analyzing and documenting the needs of the users and stakeholders. It helps to create clear and comprehensive understanding of the system or product.

### Step 1: Requirement Elicitation:

- gather info from stakeholder, their needs and expectation.
- involves interview, survey, workshop and other forms of communication.

### Step 2: Requirement Analysis:

- Analyse requirement for any inconsistencies, ambiguity, priority or gap.
- makes clear and accurate user's need.

### Step 3: Requirement Specification:

- Requirement is documented in structured way using templates such as use-cases, user stories, or functional requirement documentation.

### Step 4: Requirement validation:

- The documented requirement is validated with stakeholders including end-user, clients and the

development teams to ensure its accuracy, completeness and alignment with user needs.

#### (Step: 5: Requirement Verification:

→ Validated requirements are now verified to check whether they are complete, consistent and unambiguous or not.

#### (Step: 6: Requirement Communication:

The document is shared among developer teams, testers and other relevant stakeholders to ensure common understanding.

#### (Step: 7: Requirement Management :

→ Track changes and updates to requirement throughout the project life cycle. every changes must be documented.