

Object oriented analysis (CoA - Your Don)

→ OOA model is designed to develop & describe functionality of the system.

→ The idea of this design is to extend this model with respect to:

- processes (task)
- human interfaces
- DBMS issues

• According to CoA OOA is a process of identifying the object and classes that will be used in a system and their relationship and interaction.

Method:

OOA uses structuring principle to join them with object oriented point of view. This method consists of five steps:

① Finding classes and objects:

It specifies how classes and objects found within the application domain and identifying classes and objects within the context of system responsibilities.

② Identifying structures:-

It can be done in two different ways

① generalization specialization structure

Captures the inheritance based hierarchy of identifying class

② Whole part structure: how object is part of another object.

③ Identifying subjects:

It is done by partitioning the class and object model into larger units.

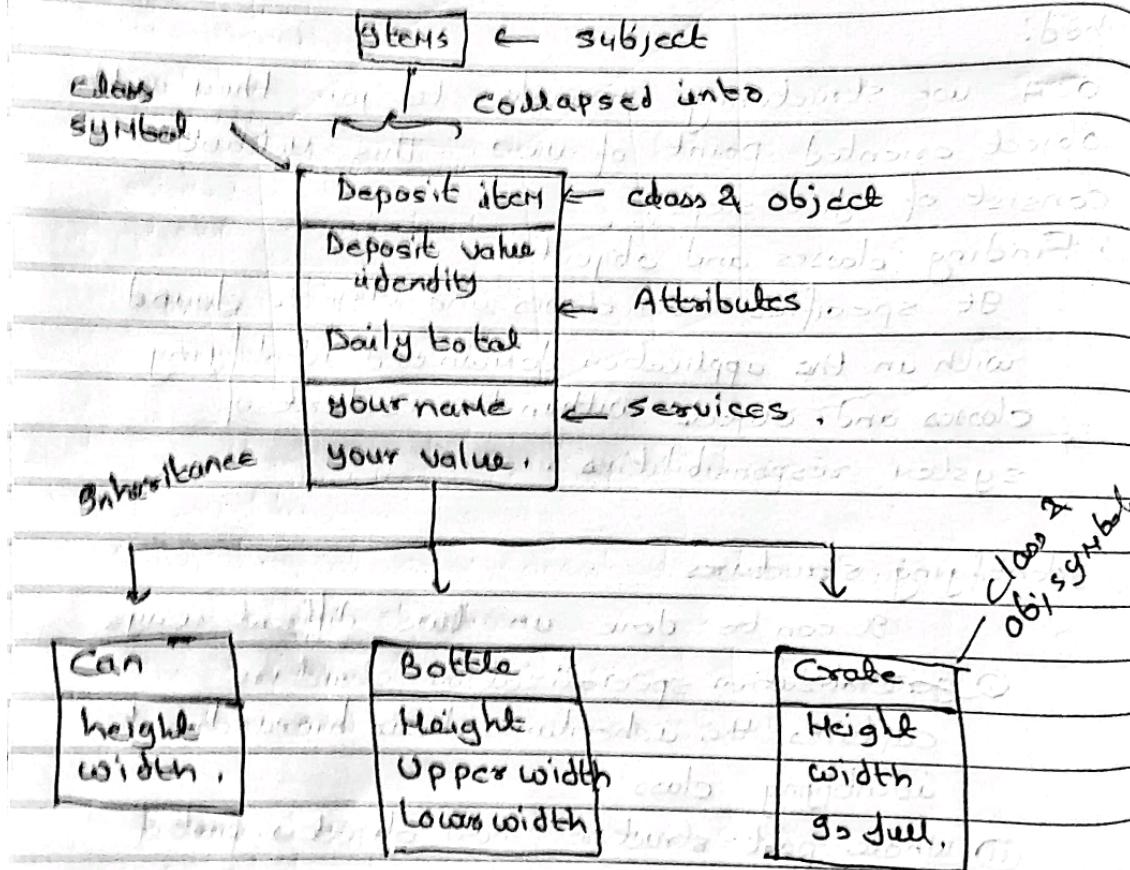
→ It is appropriate to use structure identified earlier to define subjects.

II Defining attributes:

- It can be done by identifying information and associations that should be associated to each and every instance.
- Identified attributes are placed in correct order of inheritance hierarchy.

II Defining services:

- Defining the operation of the class.
- This can be done by identifying object states and defining services as methods.



COA for part of recycling system.

Object oriented design BOOCH

- similar to COAD method in finding objects, its aim is to establish a grounding for implementation.
- OOD according to BOOCH is a process defining structure and behaviour of soft object.
- It covers the analysis and design phase of developing object oriented system.
- It strongly emphasizes the iterative process and creativity of developer as components in object oriented design.
- The BOOCH Method consists of following order of events.

① Identifying class and objects :-

- It involves finding key abstraction in the problem space.
- It is found by learning terminology of problem domain.

② Identifying semantics :-

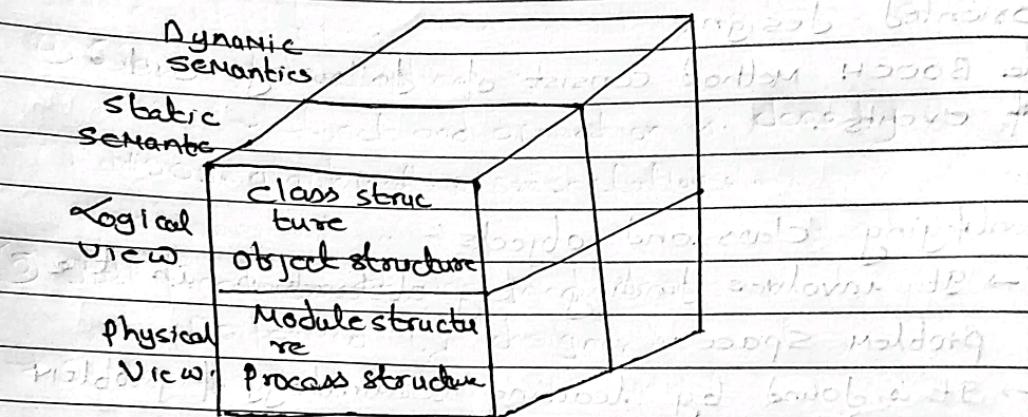
- It involves establishing relationship between classes and objects identified earlier.
- How object may be used by other objects is also important part of identifying its semantics.

③ Identifying relationships :-

- It involves relationship between classes and objects in previous activities.
- Identify how these interact with each other.
- Static and dynamic semantics & mechanism b/w the objects are defined.

Implementing class and object:

- get involved delving into the class and objects and determining its implementation.
- Decision is made how to use programming language to implement classes.
- The class and object are structured in Model.
- This structuring is made clear by Booch in 1991.



Documentation aspect of OOD

Deliverables:-

- The logical view consist of class structure and object structure.
- The physical view consist of Module of system and its processes structure.

→ Booch uses two dynamic diagram.

① state transition diagram.

Booch describes semantics of class

② timing diagram

describes how events happen between the objects

Hierarchical Object Oriented design (HOOD)

- It develops a Model that may implement directly to target language.
- HOOD aims to at supporting an abstract view of design and implementation.
- It also explain object-based approach.
(Object of inheritance and class).
- Objects are classified as
 - ① Active object (execute in parallel)
 - ② Passive object (execute sequentially at certain level)

Method:

The Method starts with identification of one object for entire system is called root object

- root object is divided into internal objects.
- It has four phases:-
 - ① Problem definition:-
 - specifying problem at current level.
 - first statement of problem is made to provide a context for current object level.
 - Requirements receive from parent object are analyze and structured.

② Development of informal strategy:-

- Means that a solution to the problem defined previously is outlined.
- This is done by using natural language employing design of current level.
- It may be of 10 sentences.

③ Formalization of Strategy:-

- extraction of main concept of informal strategy to formalize description of solution

It can be performed in following steps.

① Identification objects

→ Extracting nouns from informal strategy
and select appropriate ones

② Identification of operations

→ Extracting verbs from informal strategy

③ Grouping object & operation

→ Attaching each operation to appropriate objects

④ Graphical description

→ Object and operation is done by
Mood graphical representation

⑤ Justification of designed decision

Performed by designer

Formalization of solution

→ by developing a formal model for
identified objects

→ It is referred as Object Development
Skeleton (ODS)

Mood Design
Root object

↓
problem definition

Solution strategy

formalize the
strategy

objects &
operations

↓
formalize solution
of parent object

↓
terminal objects

↓
formalize solution
to terminal objects

↓
MOOD design steps

→ GT may be characterized as object decom
position

→ GT is very complex and suffers several
flaws

→ Major flaws decision must be taken very
early in development when dividing entire
System into components

Object Modeling Technique:- (OMT)

- It covers analysis, design, and implementation methodology, focuses on creating model of objects.
- It uses that model to develop a software.
- Developed by James Rumbaugh around 1991.
- It is dynamic approach for designing software.
- OMT is fast and easy to understand approach for identifying and modelling all the objects to make a system.
- Features like attributes, methods, inheritance and association can be expressed easily in OMT.
- OMT uses three kinds of Models.

① Object Model:

- It describes structure of object in system.
- It represents graphically with object diagram.
- It shows objects in the system and their relationship.

Object Model of bank.

Customer	Account	transaction
Name	Number	trans. date
Pincode	Amount	time
Ac No	withdraw	Amount
	deposit	Balance
		Ac no.

(i) Dynamic Model: It represents objects and their state changes over time. It represents states, transitions, events and actions of various objects.

Eg: Transaction diagram, swim and stagger etc
Sequence diagram, etc

(ii) Functional Model:
 → Handles the process perspective of the system.
 → Focus on process, dataflow diagram and action.
 → DFA show the flow of data between different process in system

Purpose of Modeling.

- (i) Testing physical entities before designing
- (ii) Communication with customer
- (iii) Visualization
- (iv) Reduction complexity

Responsibility - Driven Design (RDD)

→ A model is designed from requirement specification which is the basis for actual implementation.

→ It supports basic concepts of object-orientation such as class, object and inheritance.

→ For each class different responsibility are defined with specific responsibility.

The Method comprises a number of phases where each phase is described as no of activities.

- ① Exploratory phases
- ② Refining phases

① Exploratory phases

① Classes & objects existing in system
Accomplished by reading the specification and extracting nouns.

→ Essential nouns are found from system.

- physical objects in the domain
- Conceptual entities or object
- External interfaces
- Larger categories of classes

② Responsibility of class

By looking for verbs in requirement specification we can find the actions of the object in system, wherever information is found this is allocated in one class.

→ The purpose of class also specify responsibility.

⑩ Collaboration between class :-

the actual collaboration between the class is found by asking a question like

- ① what objects do ?
- ② what does this class need to collaborate to fulfil responsibility .

Refining phase .

① Hierarchies between class :-

- Inheritance hierarchies between classes are further refined .
- By using UML diagram .
- Common responsibility are placed as high as possible .

⑪ Subsystem :-

- Are the groups of class and are identified to simplify the patterns of collaborations .
- It shows how different class and subsystem collaborate to fulfil their responsibility .
 - All the paths between classes in the system could be captured in graphs .

⑫ Protocols :-

The responsibilities and contract are further refined to pure protocols of specific signatures of each operation .

The output of RPD consisting of .

- A graph of each class hierarchy .
- " " " collaboration of each system
- Specification of each class
- " " " subsystem .
- " " " of contract supported by each class and subsystem .

→ R&D design technique uses informal techniques and design to guidelines to develop appropriate design.

→ standards and basic costs with specific tasks

• 11.02.2009 17:16

→ ready plan

→ good standard solutions
→ needs standard working environment
→ better working conditions
• hospital norm. Brian up
• first as basic no difficulties norm
• addition up

→ table and bin costs to acquire and staff
→ standards to笑着 with planning and
→ specific bin costs traffic cost separate
→ difficulties next. 17:16 and standards
have cost in needs standard setting with HA
• simpler in budget and bins

→ standards and bin costs to acquire and staff
→ costs to笑着 with planning and
• no longer able to笑着

→ simplified 11.3 for the two
• standard costs bins to acquire
• make bins to standards

• costs bins to acquire