Object Oriented Processes

R.K.Joshi
Dept of Computer Science and Engg.
IIT Bombay
Life Cycle Models

- Waterfall
- Spiral
- Fountain
- Extreme
- Model Driven
Phases and their relations with object orientation

requirements modeling
analysis of requirements and the domain
domain modeling
design of solution space
implementation
develop-test-build
refactoring
deploying the solution
Requirements modeling

- functional requirements
  - core business logic,
  - business protocols etc.

- non-functional requirements
  - performance, distribution, security etc.
OOSE

Transforming requirements into OO implementation applying object orientation in modeling

There is also a process called OOSE, which is use case driven (Jacobson)
We will look into 3 aspects of OOSE

- early requirements
- part-whole analysis
- CRC
Methods of organization

- Differentiation of experience into objects and their attributes
- Distinction between whole objects and their parts
- Formation of classes of objects and distinction/similarities between the classes
  - [Classification theory]
  - OOA builds upon these three organizational methods
Primitives for organization

- Abstraction
- Encapsulation
- Inheritance
- Association
- Communication with messages
Major approaches to analysis

- Functional decomposition
  - Subfunctions + function interfaces

- Dataflow Analysis
  - Data and control flows
  - Data transformations, transformers
  - Data and control stores

- Information modeling: Entity Relationships
  - Entities, relations, subtypes, associations

- Object Oriented
  - Classes and objects, inheritance, associations, messages
Coad and Yourdon’s OOA Process
Major activities (Layers) in an OOA process

- Finding classes and objects
- Identifying structures
- Identifying subjects
- Defining attributes
- Defining services
Layer 1: Finding classes and objects

- Objects are abstractions in problem domain

- Classes describe one or more objects with uniform set of attributes and behavior
Where to look for?

- Observe first hand – go to the site of use
- Listen to problem domain experts
- Reuse previous OOA results
- Reuse results from similar systems
- Read the requirements document
- Ask the client for a concise summary
What to look for?

- Roles
  - E.g. supervisor, clerk, student, faculty, manager, account holder, member

- Things and Structures in the problem domain
  - E.g. Types of vehicles, types of operators for the system

- Devices
  - E.g. sensor, port, modem

- Operational procedures
  - E.g. registration, drop a course, pay fees, issue book

- Events remembered
  - E.g. successful Registration, update, login session

- Places, locations
  - Machine a.b.c.d, branch of a bank
How to name?

- Use
  - Nouns in singular
  - Nouns with adjectives
  - Activity names
  - Standard vocabulary in problem domain
Layer 2: Identifying Structures

- What’s structure?
- Why structure?
- What to structure?
  - Objects, activities
- How to structure?
  - Inheritance, part-whole, client-server, peer-peer
Layer 3: Identifying Subjects

- What are they?
  - e.g. organization, persons, vehicles, sensors

- Why?

- How
  - select
  - refine
  - construct
  - add
Layer 4: Attribute Layer

- what?
- how?
  - attributes, instance connections etc.
- Visibility
Layer 5: Service Layer

- what?
  - req/response, state machines, protocols
- Why?
- How?
The Part-Whole Relationship
(Partalogy Analysis)
Odell’s Classification Criteria

- **Configuration**
  - Whether parts have functional/structural relationship with the whole

- **Homeomericous**
  - Whether parts are same kind as that of whole

- **Invariance**
  - Whether parts can be separated from whole
Kinds of Aggregation Relations

- Component-Integral Object
- Material-Object
- Portion-Object
- Place-Area
- Member-Bunch
- Member-Partnership
Component-Integral Object

- Defines configuration of parts within a whole
- Wheels part of cart
- Bristles part of tooth brush
- Scenes part of film
Material-Object (made of)

- Describes invariant configuration of parts within a whole
  - Car is partly iron
  - Bread is partly flour
  - Kheer is partly milk
Portion-Object

- Defines a homeomorphicic configuration of parts within whole
  - Slice of bread
  - Meter part of kilometer
Place-Area

- Homeomeric and invariant configuration of parts within a whole
  - Powai part of Bombay
  - Everest part of Himalayas
  - Boundary line part of cricket field
Member-Bunch

- Defines a collection of parts as a whole

- Tree is a part of forest
- Employee is part of forum
- Ship part of fleet
Member-Partnership

- Defines an invariant collection of parts as a whole
  - Laurel part of laurel and hardy
  - Maruti is a partner in maruti-suzuki
## Classification criteria

<table>
<thead>
<tr>
<th></th>
<th>Configurational</th>
<th>homeromeric</th>
<th>Invariant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component-Object</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Material-Object</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portion-Object</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Place-Area</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Member-bunch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Member-partnership</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Classification criteria

<table>
<thead>
<tr>
<th></th>
<th>Configurational</th>
<th>homeromeric</th>
<th>Invariant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component-Object</td>
<td>Yes</td>
<td>No</td>
<td>no</td>
</tr>
<tr>
<td>Material-Object</td>
<td>Yes</td>
<td>No</td>
<td>no</td>
</tr>
<tr>
<td>Portion-Object</td>
<td>Yes</td>
<td>Yes</td>
<td>no</td>
</tr>
<tr>
<td>Place-Area</td>
<td>Yes</td>
<td>Yes</td>
<td>yes</td>
</tr>
<tr>
<td>Member-bunch</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Member-partnership</td>
<td>No</td>
<td>No</td>
<td>yes</td>
</tr>
</tbody>
</table>
Non-aggregational relations

- Topological inclusion
  - Customer is in the store
  - Meeting is in the noon
- Classification inclusion
  - Ramayana is a book
  - UML is a modeling notation
- Attribution
  - Weight of the box is 50 kg
- Attachment
  - Earrings are attached to ears
- Ownership
  - Bicycle is owned by subhash
Transitivity in Part-whole

- A relates to B, B relates to C
- Does A relate to C?
  - Car-engine-piston
  - Maths-bob-bob’s arm

If it’s the same kind of relation, transitivity applies.
E.g. apply this to compositional objects
- rotate, move, delete
Beck and Cunningham’s CRC Method of Object Identification
CRC A Method for object oriented thinking

Classes Responsibilities and Collaborators

- Class name
- Its responsibilities
- Its collaborators
CRC: our recommendation

Collaborators are other CRC cards
Collaborations will be their responsibilities
A system decomposed in CRCs

<table>
<thead>
<tr>
<th>Class name</th>
<th>Its responsibilities</th>
<th>Its collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class name</th>
<th>Its responsibilities</th>
<th>Its collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class name</th>
<th>Its responsibilities</th>
<th>Its collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class name</th>
<th>Its responsibilities</th>
<th>Its collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class name</th>
<th>Its responsibilities</th>
<th>Its collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Contributions of CRC

- The authors found that CRC gives a perspective of object-ness in the problem domain
- It’s a good technique to teach object decomposition to novices
- A card is 4X6 inch size (palmtop)
Classroom Case study

- IIT Lan Academic system for courses, registrations, grades
References

- Peter Coad/ Edward yourdon: Object-oriented Analysis, Yourdon press computing series, Pearson education, 1991