Object-Oriented Analysis (OOA)

**Analysis** – process of identifying *what* the system must do, not *how* the system must be implemented.

Starting point is usually a requirements document.

+ conversations with domain experts

Challenges:

- modern systems can be very large and complex, must understand problem in depth in short time
  - need to communicate with domain experts and to other analysts
    - read about the domain
    - need good notation, be concise
  - system requirements change!
    - model must be general enough to be resilient to change
    - capture commonality between parts of system
    - re-use previously defined components

Success Criteria

Model - clear, precise specification of the system’s functions.

Must be easy for the user (domain expert) to read.

Must be sufficiently detailed for the developer to design and implement the system from the specification.
OOA – is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain.

Different methodologies and notations exist (e.g. Booch, Coad&Yourdon)

OOA is focused around the entities (objects) of the problem domain. Iterative process!

From Booch[OOA&D] with modification.

From Coad-Yourdon

Activities (not sequential steps!) involved in OOA

Find Class-&-Objects
Identifying Structures (inheritance, a.k.a. gen-spec, whole-part)
Identifying Subjects (subject-areas)
Defining Attributes
Defining Services
1. Identifying classes and objects.

1. Where to look?
Model the following components (“Object Models”, pp. 5,6):

- Problem Domain (PB)
  - models underlying business
- Human Interaction (HI)
  - models user input and reporting
- Data Management (DM)
  - models interface between database and/or file systems
- System Interaction (SI)
  - interface to other systems
- Not This Time
  - interesting but outside the scope of current project

2. What to consider?
How to identify and classify objects?

An **object** represents an individual, identifiable item, unit, or entity, either real or abstract, with a well-defined role in the problem domain. [Smith and Tokey]

A **class** is a set of objects that share a common structure and common behavior.

Look for candidate classes and objects among:

[Shlaer and Mellor]

<table>
<thead>
<tr>
<th>Category</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Tangible things</td>
<td>Car, table, pressure sensor</td>
</tr>
<tr>
<td>- Roles</td>
<td>Mother, teacher, politician</td>
</tr>
<tr>
<td>- Events</td>
<td>Landing, sale, request</td>
</tr>
<tr>
<td>- Interactions</td>
<td>Meeting</td>
</tr>
</tbody>
</table>
[Ross]

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td>humans who carry out some function (a.k.a. roles)</td>
</tr>
<tr>
<td>Places</td>
<td>areas set up for people or things</td>
</tr>
<tr>
<td>Things</td>
<td>physical objects</td>
</tr>
<tr>
<td>Organizations</td>
<td>organized collections of people, resources, facilities or capabilities with a defined mission.</td>
</tr>
<tr>
<td>Concepts</td>
<td>principles or ideas used to organize and keep track of business activities</td>
</tr>
<tr>
<td>Events</td>
<td>things that happen, or steps in an ordered sequence.</td>
</tr>
</tbody>
</table>

[Coad and Yourdon]

Other systems (external systems with which application interacts)
Devices
Events remembered
Roles played (by users while interacting with the system)
Locations (sites important for the application)
Organizational units (groups to which users belong)

How to classify?

**Recall:** A **class** is a set of objects that share a common structure and common behavior.

Given set of objects can be classified in many equally proper ways!

Class description includes **Name**, **Attributes** and **Services**.
Attributes represent data-properties (structure) of each object in the class.
Services define the behavior of the objects.
Checklist for including classes and classifying objects (p. 66 of OOA).

- Does the system need to know and remember anything about the Objects in the Class?
- Does an Object need to provide some service to the system?

- All of class attributes are applicable to all objects within that class (and it’s specializations, a.k.a. subclasses)
- All of services of the class are provided by all objects within that class
- Class usually has multiple distinguishing attributes.
- There usually is more than one object in a class.

- Avoid merely derived results

How to name classes?

- use the standard vocabulary of the domain – do not invent new names
- use singular noun or adjective-noun for class names

2. Identifying Structures

Gen-spec - describes the class hierarchy
Whole-part – describes containment
Instance connections – describe association between classes

Notation: