Object Analysis: Classification

Credits: Material for the slides is drawn from a variety of sources including Object Oriented Analysis and Design using UML by Ali Bahrami.

Outline

- The concept of classification.
- How to identify classes
  - Noun phrase approach.
  - Common class patterns approach.
  - Use case driven approach

... Intelligent classification is intellectually hard work, and it best comes about through an incremental and iterative process

Booch

..There is no such thing as the perfect class structure, nor the right set of objects. As in any engineering discipline, our design choice is compromisingly shaped by many competing factors.

Booch

The Concept of Classification

- **Classification is the process of checking to see if an object belongs to a category or a class and it is regarded as a basic attribute of human nature.**
- **A class is a specification of structure, behavior, and the description of an object.**

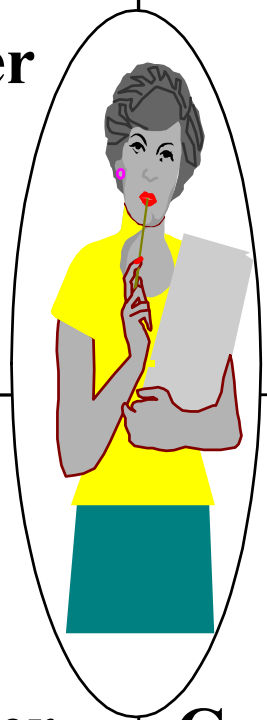
The Challenge of Classification

- **Intelligent classification is intellectually hard work and may seem rather arbitrary.**
- **Martin and Odell have observed in object-oriented analysis and design, that**

  "In fact, an object can be categorized in more than one way."

Employer

Employee



Pet Owner

Good Credit Risk

Approaches for Identifying Classes

- **The noun phrase approach.**
- **The common class patterns approach.**
- **The use-case driven approach.**

Identifying Classes

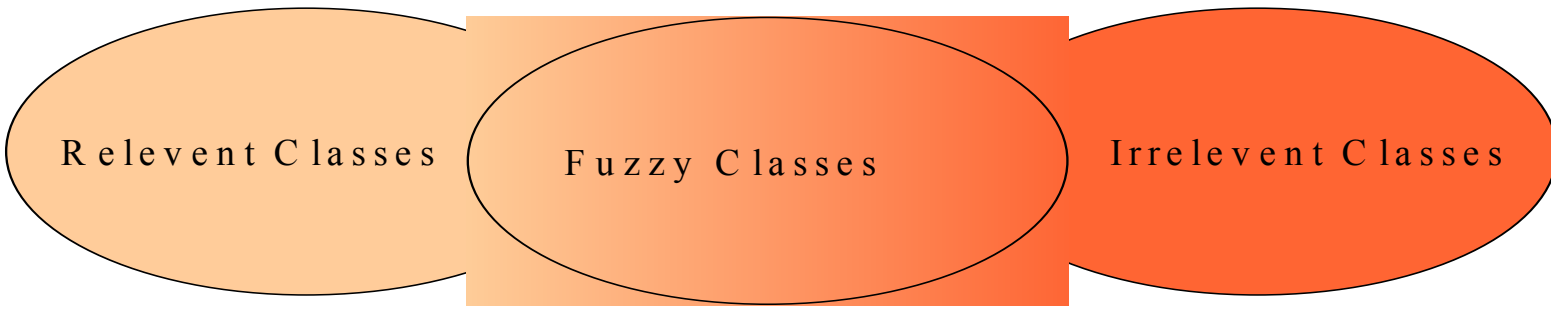
Noun Phrase Approach

Noun Phrase Approach

- **Using this method, you have to read through the Use cases, interviews, and requirements specification carefully, looking for noun phrases.**

Noun Phrase Strategy (Cont'd)

- **Change all plurals to singular and make a list, which can then be divided into three categories.**



Noun Phrase Strategy (Cont'd)

- It is safe to scrap the Irrelevant Classes.
- You must be able to formulate a statement of purpose for each candidate class; if not, simply eliminate it.
- You must then select candidate classes from the other two categories.

Guidelines For Identifying Classes

- Look for nouns and noun phrases in the problem statement.
- Some classes are implicit or taken from general knowledge.
- All classes must make sense in the application domain.
- Avoid computer implementation classes, defer it to the design stage.
- Carefully choose and define class names.

Guidelines For Refining Classes

Redundant Classes:

- **Do not keep two classes that express the same information.**

Adjective Classes:

- **Does the object represented by the noun behave differently when the adjective is applied to it?**

Guidelines For Refining Classes (Cont'd)

Attribute Classes:

- **Tentative objects which are used only as values should be defined or restated as attributes and not as a class.**

Irrelevant Classes:

- **Each class must have a purpose and every class should be clearly defined and necessary.**

Guidelines For Refining Classes (Cont'd)

- **For example, If *Adult Membership* and *Youth Membership* behave differently, than they should be classified as different classes.**
- **For example the demographics of Membership are not classes but attributes of the Membership class.**

Guidelines For Refining Classes (Cont'd)

- If you cannot come up with a statement of purpose, simply eliminate the candidate class.

Identifying Classes

Common Class Pattern Approach

Common Class Patterns Approach

- **This approach is based on the knowledge-base of the common classes that have been proposed by various researchers.**
  - **Events**
  - **Organization**
  - **People**
  - **Places**
  - **Tangible things**
  - **Concepts**

Candidate Classes - Events

- These are points in time that must be recorded and remembered.
- Things happen, usually to something else, at a given date and time, or as a step in an ordered sequence.
- For example order which is an event that must be remembered.

Candidate Classes - Organization

- The organizational units that people belong to.
- For example, accounting department might be considered as a potential class.

Candidate Classes - People and Person (Roles and Roles Played)

- The different roles users play in interacting with the application.
- It can be divided into two types (Coad & Yourdon):
  - Those representing users of the system, such as an operator, or a clerk;
  - Those people who do not use the system but about whom information is kept.

Candidate Classes - Places

- **These are physical locations, such as buildings, stores, sites or offices that the system must keep information about.**

Candidate Classes - Tangible Things and Devices

- Physical objects, or group of objects, that are tangible, and devices with which the application interacts.
- For example, cars, pressure sensors, ATM machines

Candidate Classes - Concepts

- **Concepts are principles or ideas not tangible but used to organize or keep track of business activities and/or communications.**
- **Can you name a few?**

Identifying Classes

Use case driven approach

Use-case Driven Approach

- **Once the system has been described in terms of its scenarios, we can examine the textual description or steps of each scenario to determine what objects are needed for the scenario to occur.**

Use-case Driven Approach

- To identify objects of a system and their behaviors, the lowest level of executable use cases is further analyzed with a sequence diagram.
- By walking through the steps, you can determine what objects are necessary for the steps to take place.

Guidelines for Naming Classes

- **The class should describe a single object, so it should be the singular form of noun.**
- **Use names that the users are comfortable with.**
- **The name of a class should reflect its intrinsic nature.**
- **By the convention, the class name must begin with an upper case letter.**
- **For compound words, capitalize the first letter of each word - for example, LoanWindow.**