

Class Based Modeling

Class Modeling



- **Class diagrams** provide a structural view of systems.
- Class diagrams capture static structure of Object-Oriented systems, or how they are structured rather than how they behave.
- Class diagram carry different information depending on the phase of the development process and the level of detail being considered.
- Initially, class diagrams reflect the problem domain, which is familiar to end users
- As development progresses, class diagrams move towards the implementation domain, which is familiar to software engineers

Class Modeling



- Identification of class is the most difficult part of the object-oriented analysis and design.
- Classes are an important mechanism for classifying objects.
- Chief role of class is to define the attributes, methods, and applicability of its instances.
- A class is a specification of structure, behavior, and the description of an object.
- Classes are important because they create conceptual building blocks for designing systems.

Approaches for Identifying Classes



- The noun phrase approach.
- The class responsibilities collaboration (CRC) approach.
- The use-case driven approach.
- The common class patterns approach.

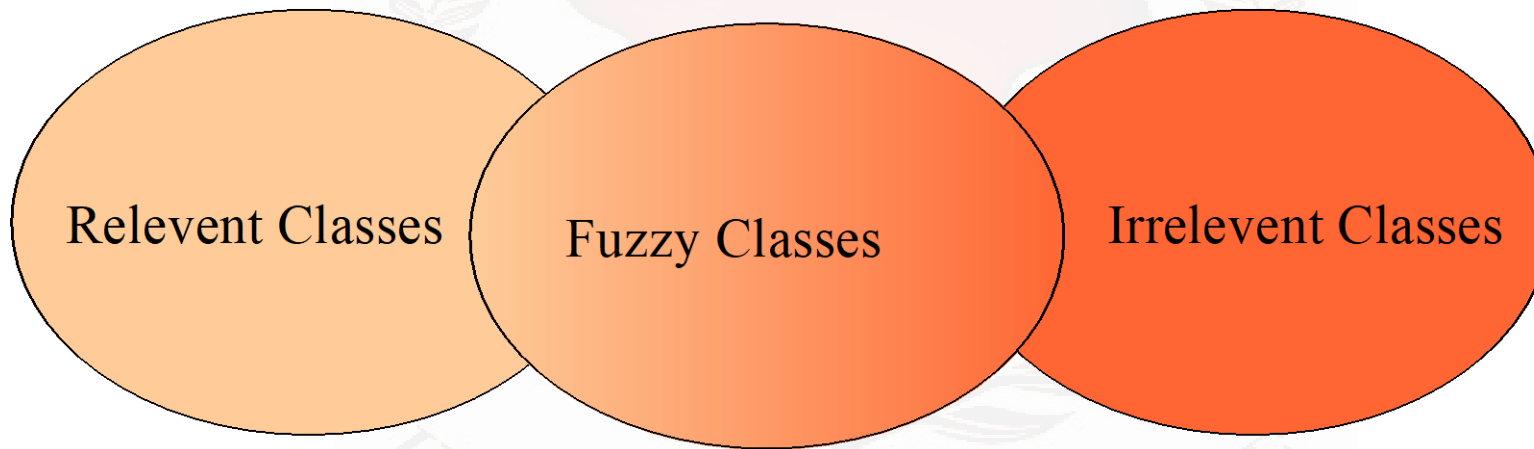
Noun Phrase Approach



- Using this method, you must read through the interviews, requirements specification and Use cases carefully, looking for noun phrases.
- Noun: a word (other than a pronoun) used to identify any of a class of people, places, or things **common noun**, or to name a particular one of these **proper noun**.
- Noun Phrase: a word or group of words containing a noun and functioning in a sentence as subject, object, or prepositional object.
- E.g., I met Raj. (Raj is noun)
- E.g., I met your brother. (your brother is noun phrase)

Noun Phrase Approach

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



1. Relevant classes
2. Fuzzy classes(class that are not sure about)
3. Irrelevant classes

Noun Phrase Approach



- Initial list of noun phrases
- Eliminate irrelevant classes
- Reviewing redundant classes
- Review adjective classes
- Review possible attributes
- Review the class purpose

Guidelines For Identifying Classes

- The followings are guidelines for selecting classes in your application:
 - 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 Identifying Classes



- **External Entities (e.g. other systems, devices, people)**
- **Things (e.g. reports, displays, letters, signals, results)**
- **Occurrences or Events (e.g. property transfer, fund transfer, robot movements)**
- **Roles (e.g. manager, engineer, salesperson)**
- **Organizational Units (e.g. division, group, team)**
- **Places (e.g. manufacturing floor, loading dock, back door)**
- **Structures (e.g. sensors, vehicles, computers)**
- **Not Objects/Attributes (e.g. number, type)**

- 1. Identify potential classes and define attributes based on above mentioned points.**
- 2. Then apply selection characteristics to retain with potential classes.**

Selection characteristics of Potential Classes



- **Retained Information:** information about class must be remembered so that the system should function.
- **Needed Services:** must have set of identifiable operations that can change the value of attributes in some way.
- **Multiple Attributes:** multiple attributes should be represented in the class
- **Common Attributes:** set of common attributes can be defined or applied to all instances of that class.
- **Common Operations:** set of common operations can be defined or applied to all instances of that class.
- **Essential Requirements:** information producing or consuming by external entities should be considered as classes

Guidelines For Refining Classes



Redundant Classes:

- Do not keep two classes that express the same information.
- If more than one word is being used to describe the same idea, select the one that is the most meaningful in the context of the system.

Guidelines For Refining Classes



Adjective Classes:

- Does the object represented by the noun behave differently when the adjective is applied to it?
- If the use of the adjective signals that the behavior of the object is different, then make a new class.
- For example, If *Senior Membership* and *Youth Membership* behave differently, then they should be classified as different classes.

Guidelines For Refining Classes



Attribute Classes:

- Tentative objects which are used only as values should be defined or restated as attributes and not as a class.
- For example, the demographics of Membership are not classes but attributes of the Membership class.

Guidelines For Refining Classes



Irrelevant Classes:

- Each class must have a purpose and every class should be clearly defined and necessary.
- If you cannot produce a statement of purpose, simply eliminate the candidate class.

Identifying a list of candidate classes: Example 1

Books and journals The library contains books and journals. It may have several copies of a given book. Some of the books are for short term loans only. All other books may be borrowed by any library member for three weeks. Members of the library can normally borrow up to six items at a time, but members of staff may borrow up to 12 items at one time. Only members of staff may borrow journals.

Borrowing The system must keep track of when books and journals are borrowed and returned, enforcing the rules described above.

- Take a coherent, concise statement of the requirement of the system
- Underline its noun and noun phrases, that is, identify the words and phrases the denote things
- This gives a list of candidate classes, which we can then whittle down and modify to get an initial class list for the system

In this particular case we discard



- **Library**, because it is outside the scope of our system
- **Short term loan**, because a loan is really an event, which so far as we know is not a useful object in this system
- **Member of the library**, which is redundant
- **Week**, because it is a measure, not a thing
- **Item**, because it is vague (we need to clarify it)
- **Time**, because it is outside the scope of the system
- **System**, because it is part of the meta-language of requirements description, not a part of domain
- **Rule**, for the same reason

This leaves:



- Book
- Journal
- Copy (of book)
- Library member
- Member of staff



Example -2



Example Case study (ATM)

The following section provides a description of the ATM system requirements.

- ☐ The bank client must be able to deposit an amount to and withdraw an amount from his or her accounts using the touch screen at the bank atm. Each transaction must be recorded and the client must be able to review all transactions performed against a given account.
 - ☐ In this bank client can have two types of accounts: a checking account and savings account.
 - ☐ The bank client will interact with the bank system by going through the approval process. After the approval process, the client can perform the transaction. The steps involved in transaction are:
 - a) Insert ATM card
 - b) Perform approval process
 - c) Ask type of transaction
 - d) Enter type
 - e) Perform transaction
 - f) Eject card
 - g) Take card
-
- ☐ Initially the client enters the PIN code that consists of 4 digits. If PIN is valid the clients account will be available. Else an appropriate message is displayed on the screen.
- ☐ Neither a checking nor savings account can have a negative balance. If the balance in saving account is less than the withdrawal amount requested, the transaction will stop and the client will be notified.

- List of nouns identified are:

- Account
- Account balance
- Amount
- Atm card
- Bank
- Bank client
- Card
- Checking account
- Client
- Clients account
- Four digits
- Invalid PIN
- Message
- PIN
- PIN code
- Savings
- Savings account
- Transaction
- Transaction history
- Touch Screen

Eliminate irrelevant classes

Here is the revised list of classes

- Account
- Account balance
- Amount
- Atm card
- Bank
- Bank client
- Card
- Checking account
- Client
- Clients account

- ~~Four digits~~
- Invalid PIN
- ~~Message~~
- PIN
- PIN code
- Savings
- Savings account
- Transaction
- Transaction history
- ~~Touch Screen~~

- Account
- Account balance
- Amount
- Atm card
- Bank
- Bank client
- ~~Card~~
- Checking account
- ~~Client~~
- ~~Clients account~~

- ~~Four digits~~
- Invalid PIN
- ~~Message~~
- PIN
- ~~PIN code~~
- ~~Savings~~
- Savings account
- Transaction
- Transaction history

Reviewing the redundant classes and building a common vocabulary

- Client, Bank client = Bank client
- Account, Client's Account = Account
- PIN, PIN code = PIN
- Checking, Checking Account = Checking Account
- Savings, Savings Account = Savings Account

Reviewing the Possible Attribute

- **Amount** -A value not a class
- **Account Balance** -An attribute of the Account class.
- **Invalid PIN** -only a value, not class
- **Transaction history** - attribute of transaction class
- **PIN** -attribute of Bank client class
- **Password**: An attribute of Bank Client class

Revised list after eliminating the attributes

- Account
 - ~~Account balance~~
 - ~~Amount~~
 - Atm card
 - Bank
 - Bank client
 - ~~Card~~
 - Checking account
 - Client
 - ~~Clients account~~
- ~~Four digits~~
 - ~~Invalid PIN~~
 - ~~Message~~
 - ~~PIN~~
 - ~~PIN code~~
 - ~~Savings~~
 - Savings account
 - Transaction
 - ~~Transaction history~~

Review the class purpose:

- ATM card class
- Bank client class: Details about client
- Account class: abstract class
- Savings and checking account: inherits account class.
- Transaction class: keeps track of the records.
- Bank class

Example -3



- The hospital has several wards divided into male wards & female wards. Each ward has a fixed number of beds. When a patient is admitted they are placed onto an appropriate ward. The Doctors In the hospital are organized into teams such as Orthopedics A, or Pediatrics, and each team is headed by a consultant doctor. There must be at least one grade 1 junior doctor in each team. The Administration department keeps a record of these teams and the doctors allocated to each team. A patient on award is under the care of a single team of doctors, with the consultant being the person who is responsible for patient. A record is maintained of who has treated the patient, and is must be possible to list the patients on award and the patients treated by a articular team. Identify the noun class