CS 315 - Lecture 7 - Sep 16, 2015

Chapter 13: Objected Oriented Analysis

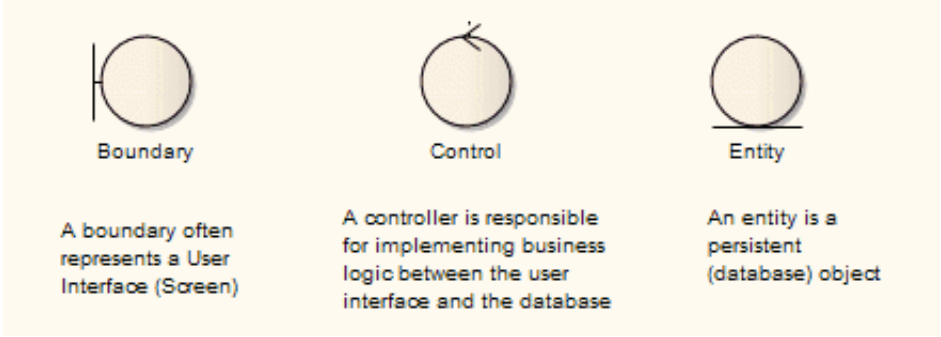
- Lecture Slides
- Analysis
  - Focuses on producing an analysis model of the system which is:
    - Correct
    - Complete
    - Consistent
    - Verifiable
  - Goal: Obtain deeper understanding of the requirements
    - Describe requirements in a way that will result in a maintainable design and implementation
    - Different from requirement elicitation, focus on structuring and formalizing the requirements
    - Not necessarily understandable by the client, moves closer to technical side
  - Forces client and developers to make difficult decisions as early as possible
    - Resolve difficult issues early in development
- Analysis Model
  - Models
    - Data
    - Function
    - Behavior
  - Types
    - Functional model
      - Functionalities of the system
      - Use cases, scenarios
    - Object model
      - Individual concepts manipulated by system and their properties
      - Classes, components
      - Static elements of the system
    - Dynamic Model

        - Behavior of system
        - Data flow, activities

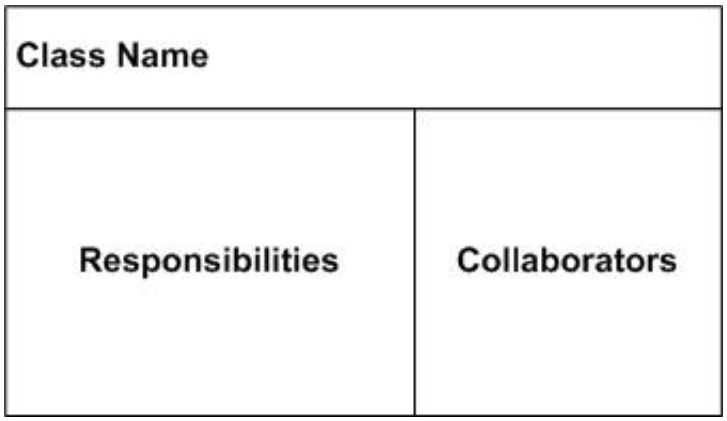
- Object Identification
    - Identifying objects (or object classes) is the **most difficult part** of object oriented design
    - There is no "magic formula" for object identification
    - Iterative process
    - Approaches
        - Scenario-based analysis
            - Identify objects, attributes, and methods per scenario
        - Behavioral approach
        - Grammatical approach
        - Base identification on tangibles in domain

- Class Extraction
    - Three types
        - Entity Classes
            - Concepts and information that live and remain in the software
        - Boundary Classes
            - Interactions between system and environment/actors
            - Generally associated with IO
        - Control Classes
            - Computations and Algorithms
    - Will use UML stereotypes to build a conceptual model
    - Entity Classes
        - Extract the entity classes, determine their relationships and find their attributes
        - Usually the best way to begin step is to use the two stage noun extraction method
            - Stage 1: Describe the information system in a single paragraph
            - Stage 2: Identify the nouns in this paragraph
    - Boundary Classes
        - Usually easy to identify
        - Inputs and outputs
        - System interfaces
    - Control Classes
        - Each computation is usually modeled by a control class

        - Realizes (accomplishes) a use case

    - UML Notation
        - 

- CRC Cards
    - Class, Responsibilities, and Collaborators
    - Scenario based technique used to identify classes, their attributes and their interaction with other classes
    - Group based
        - Domain Experts
        - Object Oriented design experts
        - Facilitator
        - Scribe

    - Literally use index cards
    - Simple example:
        - 

    - One Class per card
    - First pass from description
        - Identify likely classes
            - Recall iterative and incremental

        - Each person should be responsible for a class (card)
        - Identify responsibilities

        - Start with the obvious

    - Identify collaborators
        - Those classes obviously needed to perform a responsibility

- Iterate through a set of scenarios
- Customize as needed
- Hisorical
    - Record general description on back of card
    - Record attributes on back

- Other approaches
    - Lined sticky notes
    - Whiteboard
    - Markers