Lecture_10

Identifying object class in architecture design

- Approaches use a grammatical approach to identify objects and their behaviors. For example, nouns often represent potential objects while verbs indicate potential methods or behaviors.
- Base the identification on tangible things in the application domain
- Look for hierarchies and relationships between components in the architecture. Hierarchies may indicate inheritance relationships between object classes.

Design Model

- Design models show the objects and object classes and relationships between entities
- Structural models
    - Subsystem models show how the design is organized into logically related groups
    - Sequence models show the sequence of object interactions that take place
- State diagrams are used to show how objects respond to different service requests and the state transitions triggered by these requests
    - Useful high-level models of a system or an object runtime behavior
    - Relatively simple and a state model adds unnecessary detail

Examples of design models

- Class Models
- Subsystem
- Sequence models
- State machine models

Unified modeling language

- A standardized specification for analysis and design
- A language, not a model
- Allows for visualized design and architecture
- More UML Relationships

Class

- The class represents a concept
- Encapsulates state and behavior
    - Has a name and a type, and can be the initial value

    - Each method has a signature: name, parameter, type, return

Access level	Meaning	Explanation
+	Public	The member is visible to all code in the application
-	Private	The member is only visible to code inside the class
#	Protected	The member is visible only to code inside the class and derived classes
-	Package	The member is visible only to code inside the same package

For more info on class protections and I/O read CPP_Review

Interface

- Specifies a contract
- Any other class that implements the interface must fulfill the contract

Generalization

- Informally called "Inheritance" or "is A" relationship
- Generalization is a directed relationship between a more general class and a more specific class
- Multiple inheritance is allowed in UML
- The children classes inherit the attributes and operations of the parent class and can have additional ones

Association

- Describes the presence of a relationship between classes
- The name of the association and multiplicity may be placed on the line
- The association end name is commonly referred to as role
- The professor is the author of a book; a book is used as a textbook by a professor
- The multiplicity of an association denotes how many objects the instance of a class can legitimately reference
    - every book has at least one author, a professor can write as may books, including none

Multiplicity of Associations

Indication	Meaning
0..1	Zero or one
1	One
0..or *	Zero or more

Indication	Meaning
1..*	One or more
n	Only n (n > 1)
0..n	Zero to n (n > 1)
1..n	One to n (n > 1)

- Captured in source code by means of reference properties
- A class can have an association with itself

Aggregation

- A special type of association denoting a part-whole relationship
- Parts can exist without the whole; when the whole is destroyed the parts aren't
    - The exhaust system "consists of"
    - Muffler "is part of " the exhaust system
    - The tailpipe "is part of " the exhaust system

Dependency

- Represents a "using" relationship
- If a change in the specification in one class affects another (but vice versa) there is a dependency