UML:
Fundamental UML Diagrams for High-Level Conceptual Modeling

The Use Case Diagram,
The Sequence Diagram,
The Class/Object Diagram, and
The Activity Diagram
Lecture Objectives

• Overview of UML
• Fundamental UML Models for High-level (Coarse Grained) Conceptualizations
• Rational Rose as a CASE tool for UML Modeling
• Exemplar Problem and its representation in UML
Fundamental UML Models

- UML provides 5 views of (i.e. five different ways of perceiving) a System:
  - The Use-Case view
  - The Logical view
  - The Process View
  - The Implementation View
  - The Deployment View

  These 3 views represent the detailed (software and hardware level) views of a system
  These two views represent the high-level views of a system
Fundamental UML Models

• For this class, we shall focus on (and learn) only the high-level UML View, and their associated diagrams.

• The Use Case view models the end-users perspective of a system and employs the following diagrams:
  – Use-case diagram
  – Sequence diagram

• The Logical View models the “details” of a system without factoring in the technology being used to perform or support these details. This view provides a “blue-print” of the system. It employs the following diagrams:
  – Sequence diagram
  – Class diagram
  – Activity Diagram
Use Case Diagram: Definition

- Use cases are widely used to decompose the problem or its solution into concrete functional units

- Use cases facilitate the:
  - Scoping of the solution: determine the size, complexity, and constituent parts of the proposed system
  - Decomposing of the proposed system into manageable work-units (jobs) that can be delegated to different development teams/personnel
Use Case Diagram: Key Terms

Actor
• Anyone or anything that interacts with the system
Use Case Diagram: Key Terms

**Use Case**

- An encapsulation of some major function fulfilled or performed by the system to be designed.
- Represents a function, a workflow, a major process etc.
- Use Cases drive the design/conceptualization process
Use Case Diagram: Key Terms

USES RELATIONSHIP
• Allows one use case to use the functionality provided by another use case.
• Primary use case call secondary use-case (arrow from calling use case to called use case)

EXTENDS RELATIONSHIP
• Allows one use case to optionally extend the functionality provided by another use case (arrow from called to calling use case)
  [ arrow direction inverse of the USES relationship]
Use Case Diagram: Example of a Use-Case Diagram
Use Case Diagram: Another Example of A Use Case Diagram
Sequence Diagram: Definition

- The sequence Diagram provides a chronological depiction of the ACTIVITIES that take place in a task, and WHO (the individual or thing or resource) that is responsible for performing each event/action.
  - ACTIVITIES represent the jobs or “value-adding” mechanisms within the task.
  - ACTIVITIES are represented as arrows that are labeled with some “action-phrase” or verb.
  - ‘WHO’ (the individual or thing or resource) that performs a specific set of ACTIVITIES is represented as an object.
- Note: Activities are performed by objects. The set of activities that a specific object performs is called the “methods” of that object.
SEQUENCE DIAGRAMS: KEY TERMS

OBJECT

- A representation of an some real-life phenomenon within the work system
- Each object can have attributes and behaviors
  - ATTRIBUTES:
    - Data about the object.
    - Object’s Characteristics that are of interest to the systems designer
  - BEHAVIORS
    - The activities that can be carried out by the object
SEQUENCE DIAGRAMS: KEY TERMS

OBJECT INTERACTION (MESSAGE)

• Objects are able to interact with other objects.
• These interactions reveal the actions/events of the task or work-system being modeled.
• The interaction between two objects is also called a message (i.e. a message being sent from one object to another object).
• Object interaction is articulated as an ARROW linking two objects.
  – Each such arrow is labeled with a verb phrase.
  – The verb phrase represents the action/event that is to be performed by the “receiving” object.
  – The “receiving” object is the object to which an arrow is pointing.
SEQUENCE DIAGRAM: KEY TERMS

SCENARIO

• A scenario is a complete depiction of a specific task that shows all the activities that must be performed in order for the task to be fulfilled.
  – It also elaborates all the OBJECTS that are involved in completing a given task
  – Additionally, it elaborates the chronology or order in which the objects interact in order to fulfill a task.

• Viewed in another way, a scenario is one instance of the possible flow of activity (workflow) through a use-case.
  – Therefore a scenario represents one possible instantiation of a use-case.
  – We develop sequence diagrams to articulate ALL possible scenarios of each of the use-cases in a work system
SEQUENCE DIAGRAM SYMBOLS

• **Actor**: someone or something outside the system that interacts with the system, either by giving or receiving information or both.

• **Object**: an instance of a class; that encapsulates state and behavior.

• **Lifeline**: represents the existence of the object at a particular time.

• **Focus of Control**: shows the period of time during which an object is performing an action, either directly or through a subordinate procedure.

• **Message**: a specification of communication between objects that conveys information with the expectation that activity will ensue.

• **Link**: is a pathway for communication between objects on a sequence diagram.
1. Each sequence diagrams captures ONE SCENARIO.

2. Each Scenario begins with an ACTOR: to designate the point of entry into the system.

3. The Scenario continues through a series of messages (interactions) until it is terminated by:
   1. Having a message passed back to the ACTOR (or to another actor)
   2. Having the message passed to a PERSISTENT state (e.g. storage of data)
CLASS DIAGRAM: DEFINITION

• The class diagram provides a “snap shot” of an entire system (work system)
• It is a condensed representation of “everything that makes up” a system
• The class diagram thus aggregates all the various sequence-diagrams into one single model
• In doing so, the class diagram achieves a number of key objectives:
  – It displays, in one diagram, all the STAKEHOLDERS (customers, …, etc) of the work system. Each stakeholder is depicted as a single class in the class diagram
  – It displays in one single diagram, what each stakeholder’s RESPONSIBILITIES are. Responsibilities are depicted as the set of methods in each class.
  – It displays the relationships (associations) among the stakeholders. Relationships are depicted as lines (or arrows) connecting a pair of classes
CLASS DIAGRAM: KEY TERMS

• CLASS:
  – For purposes of this course, a class is the same thing as an object
  – A class represents a THING – one of the three key players in a work system (customer, participant, ….)
  – A class is modeled as a rectangle
CLASS DIAGRAM: KEY TERMS

• RELATIONSHIP (ASSOCIATION)
  – A relationship is the interaction between any two classes.
  – A relationship indicates that one class sends a message to another class.
  – A relationship is modeled as a line or arrow with special “arrow-heads”.
  – The “arrow-head” used depicts the TYPE of the relationship.
  – In essence there are three fundamental types of relationships:
    • Dependency
    • Associative relationship
    • Inheritance, and
    • Aggregation, which is further sub-divided into
      – Composition
      – Generic Aggregation
Class Diagram: Identifying Relationships

- Since a class interacts with other classes, its relationships are also established
  - A relationship is an association between any two classes.
- We use Interaction Diagrams to identify Relationships
  - **Messages** passed between objects of two different classes, or objects of the same class, indicate relationships
CLASS DIAGRAM: KEY TERMS

Dependencies

• A dependency is a relationship between two objects:
  – Represented with a dotted arrow
  – Used to indicate NON-STRUCTURAL relationships between classes (e.g.: what causes the supplier to be VISIBLE to the client?)

• Dependencies are TRANSCIENT (temporary) links, therefore:
  • have a limited duration
  • Are context independent relationships
  • Are summary relationships (does not expose detail of relationship. One must peruse behavioral model for details)
CLASS DIAGRAM: KEY TERMS
Modeling Dependencies
CLASS DIAGRAM: KEY TERMS

Associations

- Associations are structural links between classes

- They are represented as SOLID ARROWS, or SOLID LINES
CLASS DIAGRAM: KEY TERMS
Associations and Dependency
CLASS DIAGRAM: KEY TERMS

Association Navigability

- DEF: Ability to navigate from an associating class to the target class using the association that links these classes
  - Indicated by SOLID arrow for unidirectional, and NO arrow for bidirectional
CLASS DIAGRAM: KEY TERMS
Composition

- **DEF:** A form of aggregation with strong ownership and coincidental lifetimes
  - The parts cannot survive the whole (e.g. order cannot exist without items being ordered [order lines])

- Indicated by a SOLID diamond
- By definition, composition is a NON-SHARED aggregation (multiplicity of 1 at the diamond)
CLASS DIAGRAM: KEY TERMS

Generic Aggregation

- DEF: Whole-part relationship
- Generic aggregation refers to non-composite aggregation (i.e. sub classes can survive the super-class, can exist independent of the super-class)
  - e.g. wheel, engine etc can exist independent of vehicle
- Indicated by a TRANSPARENT DIAMOND
- Where shared, the multiplicity (at the diamond) is greater than 1, where not, it is 1
CLASS DIAGRAM: KEY TERMS

Generalization

- **DEF:** “is a” relationship
- Concept gives rise to ABSTRACT and CONCRETE classes
- Abstract classes cannot have any objects
- Concrete classes can have objects

Diagram:

```
    abstract
       ↓
    concrete

Animal
abstractAttribute
abstractOperation()

1
0..*

Lion
ConcreteAttribute
concreteOperation()

Fish
```

```
Comprehensive Class Diagram