Lecture 3
Requirements Modeling

1

1

Use Case Modeling

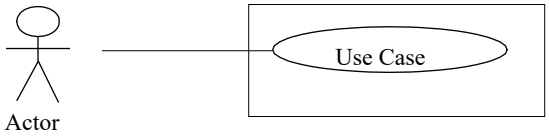
Chapter 6 - *Software Modeling and Design*,
Cambridge University Press

2

2

Use Case Modeling

- Use Case modeling
  - Specify user's requirements
  - Define system functional requirements in terms of Actors and Use cases



3

3

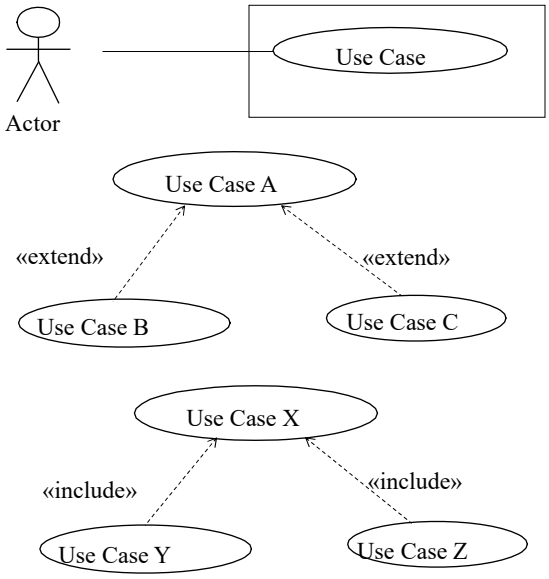
Use Case Modeling

- Initially developed in Use Case model
  - Functional requirements are defined in terms of actors and use cases
  - Shows interaction between actor and black box system
- Use cases refined in Dynamic Model
  - Show objects participating in use case
  - Develop communication diagrams or sequence diagrams
- Use cases refined further in Design Model
- Use cases form basis of integration & system test cases

4

4

Figure 2.1 UML notation for use case diagram



5

5

Actors

- Actors are external to system
- Actors interact directly with system
- An actor could be:
  - Human user
  - External system
  - Input device
  - Timer
- System - System makes decisions and not Actor

6

6

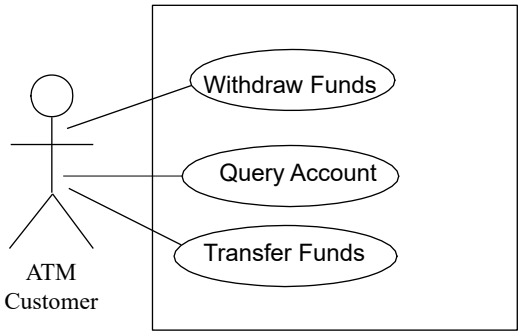
Actors

- Actor initiates interactions with use case
- Primary Actor
  - Starts the use case by providing input to the system
- Secondary Actor(s)
  - Participates in use case
  - Can be Primary Actor of a different use case

Use Cases

- Use case
  - Describes a complete sequence of interactions between actor and system
  - Starts with input from an actor
  - Use case name: Should describe an action, e.g., *Validate PIN*
- How to identify use cases?
  - Identify each actor first
  - Consider requirements of each actor who interacts with system

Use Case Model



Documenting Use Cases

- Name
- Summary - Short description of use case
- Dependency (on other use cases)
- Actors – primary, secondary
- Precondition(s)
  - Condition that exists before use case executes
    - e.g., *ATM is Idle, displaying a welcome message*
- Description of main sequence
  - Most common sequence
  - Sentences start with System or Actor, e.g.,
    - *Customer selects payment by credit card*
    - *System prompts Customer to swipe card*

Documenting Use Cases

- Description of alternative sequences
  - Deviations from main sequence
    - E.g., for error handling
  - Identify step # where deviation starts
  - Rejoin main sequence?
    - Application dependent
- Nonfunctional requirements (optional)
  - Can be specified in a section separated from the use case
  - Can be specified with use case
- Postcondition
  - Condition that is true at end of use case

11

Example of Use Case

Use case name: Validate PIN.

Summary: System validates customer PIN.

Actor: ATM Customer.

Precondition: ATM is idle, displaying a "Welcome" message.

Description of main sequence:

1. Customer inserts the ATM card into the card reader.
2. If system recognizes the card, it reads the card number.
3. System prompts customer for PIN.
4. Customer enters PIN.
5. System checks the card's expiration date and whether the card has been reported as lost or stolen.
6. If card is valid, system then checks whether the user-entered PIN matches the card PIN maintained by the system.
7. If PIN numbers match, system checks what accounts are accessible with the ATM card.
8. System displays customer accounts and prompts customer for transaction type: withdrawal, query, or transfer.

12

Example of Use Case

Alternatives

Step 2: If the system does not recognize the card, the system ejects the card.

Step 5: If the system determines that the card date has expired, the system confiscates the card.

Step 5: If the system determines that the card has been reported lost or stolen, the system confiscates the card.

Step 7: If the customer entered PIN does not match the PIN number for this card, the system reprompts for the PIN.

Step 7: If the customer enters the incorrect PIN three times, the system confiscates the card

Steps 4-8: If the customer enters Cancel, the system cancels the transaction and ejects the card.

Nonfunctional requirements:
a)Security requirement: System shall encrypt ATM card number and PIN.
b)Performance requirement: System shall respond to actor inputs within 5 seconds.

Postcondition: Customer PIN has been validated.

13

13

Static Modeling

Reference: H. Gomaa, Chapter 7 - *Software Modeling and Design*, Cambridge University Press,

14

14

Static Modeling

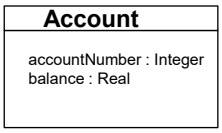
- Define structural relationships between classes
  - Depict classes and their relationships on class diagrams
- Static Modeling
  - Define classes and attributes of classes
  - Defines operations of each class
- Relationships between classes
  - Associations
  - Composition / Aggregation
  - Generalization / Specialization

15

15

Figure 4.2 Example of class with attributes

Class with attributes



Objects with values

16

16

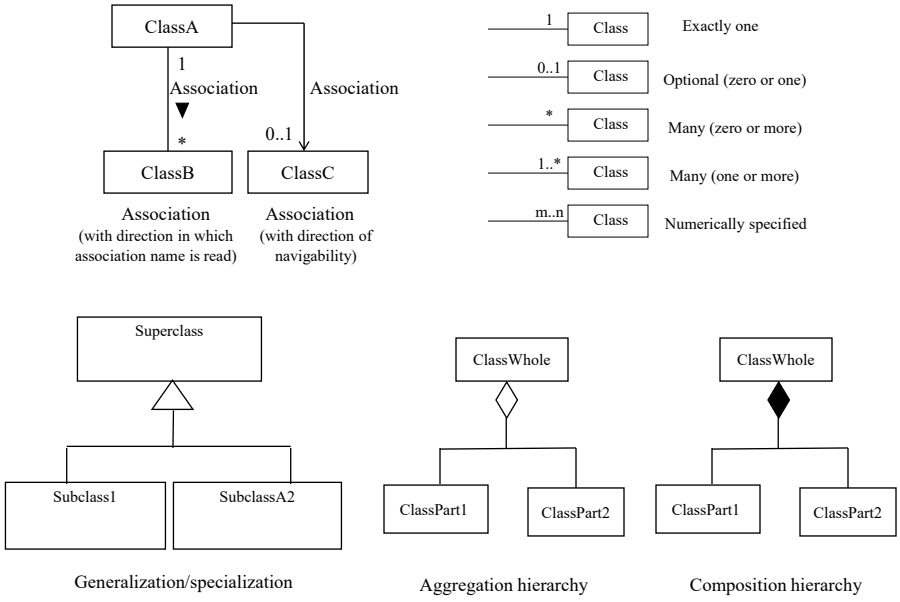
Associations

- Association is
  - Static, structural relationship between classes
  - e.g., Employee works in Department
- Multiplicity of Associations
  - Specifies how many instances of one class may relate to a single instance of another class
  - Examples
    - 1-to-1 association
    - 1-to-many association
    - Optional association (0 or 1)
    - Optional association (0, 1, or many)

17

17

Figure 2.3 UML notation for relationship on class diagram

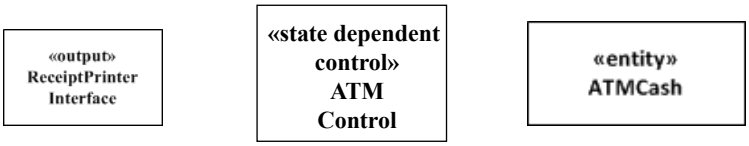


18

18

Object and Class Structuring Criteria

- Objective
  - Determine all software objects and classes in system
    - Use Object and Class Structuring Criteria
    - Guidelines for identifying objects and classes
- Structuring criteria depicted using stereotypes
- **Stereotype**
  - Defines role of class or object in application
  - Depicted using angle brackets, e.g.,
    - «entity», «output», «control»
- Class has same stereotype as objects instantiated from it



19

19

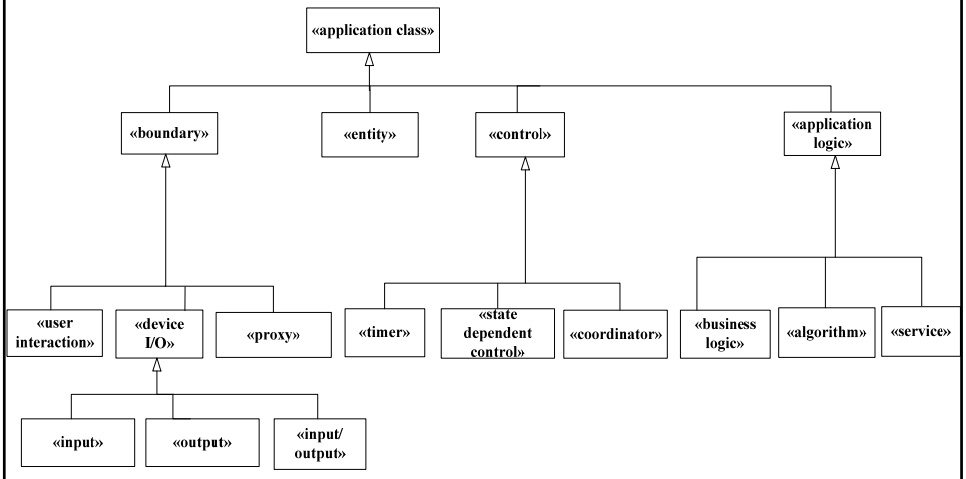
System of Classification

- Objects and classes are categorized
  - A **category** is a specifically defined division in a system of classification
  - Group together classes with similar characteristics
  - Categorization of application classes by stereotype (Fig. 8.1)

os-20

20

Classification of application classes using stereotypes



21

21

Object Structuring Criteria

- Boundary object
  - Software object that interfaces to and communicates with external environment
  - Each software boundary object interfaces to an external (real-world) object
    - User interaction object
    - Device I/O object
    - Proxy object (to communicate with external systems)
- For each software boundary object
  - There is a corresponding external object

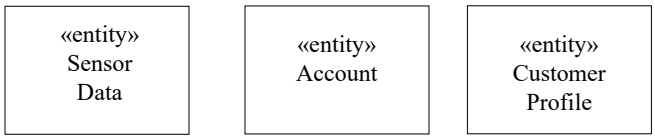
«user interaction» :Operator Interaction

22

22

Entity Classes and Objects

- Entity objects
  - Software object that stores information
  - Determined during static modeling
    - Entity classes and relationships shown on class diagram



23

23

Object Structuring Criteria

- Control objects
  - Provides overall coordination for execution of a group of objects
  - Makes overall decisions
  - Decides when, and in what order, other objects participate in interaction sequence
    - Entity objects
    - Boundary objects

24

24

Application Logic Objects

- Application Logic Object
  - Software object that contains details specific to application
- Application Logic Objects
  - Business Logic Object
  - Algorithm Object
  - Service Object

25

25

Dynamic Interaction Modeling

Reference: H. Gomaa, Chapters 9,11, 21 - *Software Modeling and Design*, Cambridge University Press,

26

26

Dynamic Interaction Modeling

- Use cases realized in Dynamic Interaction Model
  - Show objects participating in each use case
- Determine how objects participate in use case
  - Shows sequence of object interactions in use case
    - Depict on
      - **communication diagram** or
      - **sequence diagram**
- Dynamic Interaction Modeling
  - Approach to determine how objects interact with each other to support use case

27

Communication Diagram

- Graphically depicts objects interacting with each other
  - Show objects as boxes
  - Show their message interactions as arrows
  - Number sequence of messages

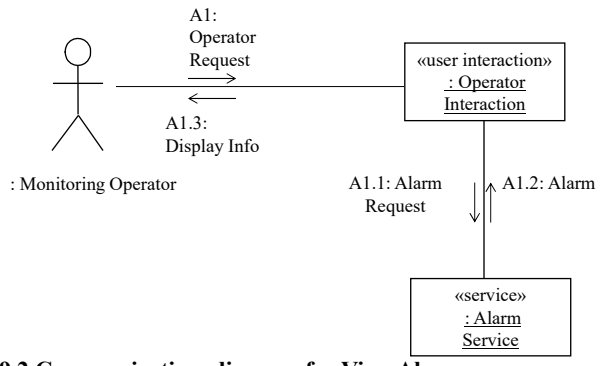


Figure 9.2 Communication diagram for View Alarms use case

28

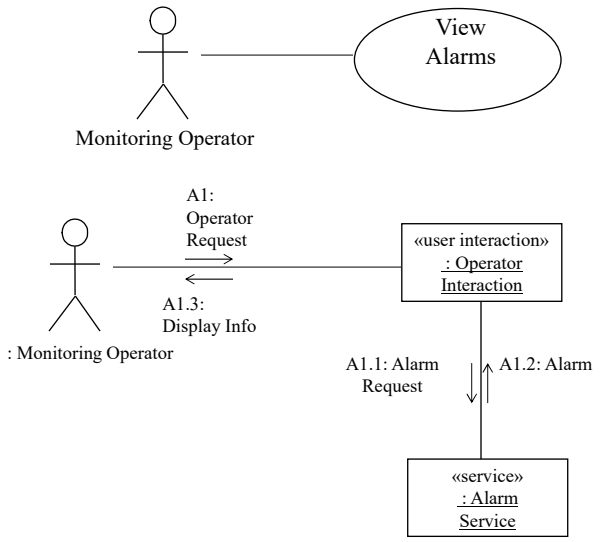
Dynamic Interaction Modeling

- Determine how objects interact with each other to support use case
  - Start with external event from actor
  - Determine objects needed to support use case
  - Determine sequence of internal events following external event
  - Depict on communication diagram or sequence diagram
- Stateless (non state-dependent) Dynamic Interaction Modeling
- State dependent Dynamic Interaction Modeling

29

29

Example of Stateless Dynamic Interaction Modeling
Figure 9.2 Communication diagram for View Alarms use case



30

30

Message Sequence Numbering

- Form of message sequence number
  - [first optional letter sequence][numeric sequence] [second optional letter sequence]
- First optional letter sequence - use case id
- Numeric sequence
  - Message sequence starting with external event
  - A1, A2, A3
- Interactive System
  - Whole number for external event
    - A1, A2
  - Decimal number for subsequent internal events
    - A1.1, A1.2, A1.3, …, A2, A2.1, A2.2,…
- Second optional letter sequence
  - Concurrent event sequences
    - A3, A3a
  - Alternative message sequences
    - D1[Normal], D1A[Error]

31

31

Make Order Request use case description

Use case name: Make Order Request

Summary: Customer enters an order request to purchase catalog items. The customer's credit card is checked for validity and sufficient credit to pay for the requested catalog items.

Actor: Customer, Bank

Precondition: Customer has selected one or more catalog items

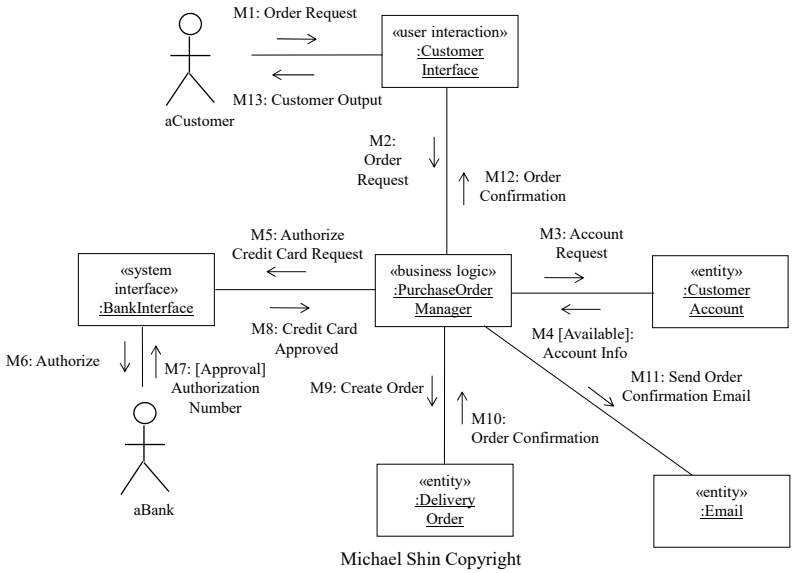
Main sequence:

1. Customer provides order request and customer account Id to pay for purchase.
2. System retrieves customer account information, including the customer's credit card details.
3. System requests to a bank checking the customer's credit card for the purchase amount and, if approved, creates a credit card purchase authorization number.
4. System creates a delivery order containing order details, customer Id, and credit card authorization number.
5. System confirms approval of purchase and displays order information to customer.
6. System sends email confirmation to customer.

32

32

Main Sequence for Make Order Request use case



33