Use Case Modeling

SWE 321
Fall 2014

Reference: Gomaa, Chapter 6

Copyright © 2014 Rob Pettit
All rights reserved. No part of this document may be reproduced in any form or by any means, without the prior written permission of the author.
Architectural Views

- **Static**
  - Classes
  - Associations

- **Dynamic**
  - Interactions
  - Behavior over time

- **Use Cases**
  - Requirements
  - Testability
  - Planning

- **Implementation**
  - Code organization

- **Deployment**
  - System topology
  - Installation
Steps in Using COMET/UML

1. Develop Software Requirements Model
   - Develop Use Case Model (Chapter 6)
2. Develop Software Analysis Model
3. Develop Software Design Model

Figure 6.1 COMET object-oriented software life cycle model
Use Case Modeling

• Use Case Model:
  – Identifies black box functionality of system
    • Use case: black box function / goal / feature
    • Actor: External role interacting with system
  – Defines system boundary
  – Shows interaction between actor and black box system
• Use cases refined in Dynamic Model
  – Show objects participating in use case
  – Develop interaction diagrams
• Use cases refined further in Design Model
• Use cases form basis of integration & system test cases
Basic UML Use Case Notation

- Use case
- «actor»
- Non-Human Role
- Actor
- Customer
- Salesperson
- Shipping Clerk
- Supervisor
- Establish credit
- Place order
- Check status
Actors

- Actor models external roles interacting with the system
- Actors interact directly with system
  - Human user
  - External device
  - Timer
  - External system
  - Primary actor initiates use cases
  - May use I/O devices or external system to physically interact with system
- Typically human if interacting through standard I/O devices
- Often non-human for embedded systems where input is through sensors, etc.
Figure 6.1 Example of actor and use case

ATM Customer

Withdraw Funds
Example of input device actor
More on Actors

• Primary Actor
  – Starts the use case by providing input to the system
• Secondary Actor
  – Participates in use case
  – Can be Primary Actor of a different use case
• Actor
  – Represents a role rather than an individual
    • e.g. with Blackboard, Dr. Pettit can act as a Teacher or a Student

2 Actors, 1 physical person

• Conversely, ALL students represented by single Student actor
Use Cases

• Define system functional requirements in terms of Actors and Use Cases
  – Narrative description - most important part of use case
  – Set of scenarios associated with a common goal
• Identifying use cases
  – Consider requirements of each actor who interacts with system
  – Use case is a complete sequence of events initiated by an actor
    • Use case starts with input from an actor
    • Describes interactions between actor and system
      – Captures what the system does in response to inputs, not internals of how it does it
  – Some questions to ask:
    • What functions will a specific actor want from the system?
    • Does the system store and retrieve information? If so, what triggers this behavior?
    • Are any actors notified when the system changes state?
    • Are there any external events that affect the system? Who/what notifies the system and what actions are taken?
  – Document basic path and all alternatives
Figure 6.7 Banking system actor & use cases

ATM

Customer

Withdraw Funds

Query Account

Transfer Funds
Documenting Use Cases

• Name
• Summary
  – Short description of use case
• Dependency (on other use cases)
• Actors
• Preconditions
  – Conditions that are true at start of use case
• Description
  – Narrative description of basic path
• Alternatives
  – Narrative description of alternative paths
• Nonfunctional Requirements
  – Considerations needed for performance, security, etc.
• Postcondition
  – Condition that is true at end of use case
• Outstanding questions
  – Discussion points for stakeholders
Use Case Name: Withdraw Funds
Summary: Customer withdraws a specific amount of funds from a valid bank account.
Actor: ATM Customer
Precondition: ATM is idle, displaying a Welcome message.

Description:
1. Customer inserts the ATM Card into the Card Reader.
2. If the system recognizes the card, it reads the card number.
3. System prompts customer for PIN number.
4. Customer enters PIN.
5. System checks the expiration date and whether the card is lost or stolen.
6. If card is valid, the system then checks whether the user-entered PIN matches the card PIN maintained by the system.
7. If PIN numbers match, the 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.
9. Customer selects Withdrawal, enters the amount, and selects the account number.
10. System checks whether customer has enough funds in the account and whether daily limit has been exceeded.
11. If all checks are successful, system authorizes dispensing of cash.
12. System dispenses the cash amount.
13. System prints a receipt showing transaction number, transaction type, amount withdrawn, and account balance.
15. System displays Welcome message.
Alternatives:
- If the system does not recognize the card, the card is ejected.
- If the system determines that the card date has expired, the card is confiscated.
- If the system determines that the card has been reported lost or stolen, the card is confiscated.
- If the customer entered PIN does not match the PIN number for this card, then the system re-prompts for the PIN.
- If the customer enters the incorrect PIN three times, then the system confiscates the card.
- If the system determines that the account number is invalid, then it displays an error message and ejects the card.
- If the system determines that there are insufficient funds in the customer’s account, then it displays an apology and ejects the card.
- If the system determines that the maximum allowable daily withdrawal amount has been exceeded, then it displays an apology and ejects the card.
- If the ATM is out of funds, then the system displays an apology, ejects the card, and shuts down the ATM.
- If the customer enters Cancel, the system cancels the transaction and ejects the card.

Postcondition: Customer funds have been withdrawn.
Activity Diagrams

- Initial Node
- Control Flow
- Action or Activity
- Object Flow
- Branch
- Merge
- Fork
- Join
- Final Node

Diagram:

- Start
- Event
- Activity
- Data Flow
- Decision
- [Yes]
- [No]
- Return
- Split
- Return
- Finish
Mapping To The Use Case

- Precondition
- Actor input
- System Step
- Alternative or extension flow
- Basic Flow
- Returning alternate flow
- Parallel activities
- Postcondition
Advanced Use Case Modeling

• Most industrial applications contain a large number of use cases and actors
• Advanced use case modeling can help simplify the overall model by:
  – Factoring common use case functionality and actor roles
  – Providing extension mechanisms for use case variations
• Two basic relationships for use cases:
  – «include»
  – «extends»
Use Case Relationships

- **Include** relationship
  - Factor common repetitive patterns (sequences) in several use cases
  - Separate more complicated steps or steps that are more likely to change
  - Use case specification for «include» is similar to a function call
  - Included use case may or may not be complete
  - Base use case is **not** complete without the included use cases

- Example
  - Withdraw Funds use case **includes** Validate PIN use case
Figure 6.9 Example of inclusion use case and include relationships

ATM Customer

- Validate PIN
  - «include» Withdraw Funds
  - «include» Query Account
  - «include» Transfer Funds
Use Case Name: Validate PIN
Summary: System validates customer PIN.
Actor: ATM Customer

Precondition: ATM is idle, displaying a Welcome message.

Description:
1. Customer inserts the ATM Card into the Card Reader.
2. If the system recognizes the card, it reads the card number.
3. System prompts customer for PIN number.
4. Customer enters PIN.
5. System checks the expiration date and whether the card is lost or stolen.
6. If card is valid, the system then checks whether the user-entered PIN matches the card PIN maintained by the system.
7. If PIN numbers match, the 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.

Alternatives:
• If the system does not recognize the card, the card is ejected.
• If the system determines that the card date has expired, the card is confiscated.
• If the system determines that the card has been reported lost or stolen, the card is confiscated.
• If the customer-entered PIN does not match the PIN number for this card, the system re-prompts for the PIN.
• If the customer enters the incorrect PIN three times, the system confiscates the card.
• If the customer enters Cancel, the system cancels the transaction and ejects the card.

Postcondition: Customer PIN has been validated.
Use Case Name: Withdraw Funds

Summary: Customer withdraws a specific amount of funds from a valid bank account.

Actor: ATM Customer

Dependency: Include Validate PIN abstract use case.

Precondition: ATM is idle, displaying a Welcome message.

Description:

1. Include Validate PIN abstract use case.
2. Customer selects Withdrawal, enters the amount, and selects the account number.
3. System checks whether customer has enough funds in the account and whether the daily limit will not be exceeded.
4. If all checks are successful, system authorizes dispensing of cash.
5. System dispenses the cash amount.
6. System prints a receipt showing transaction number, transaction type, amount withdrawn, and account balance.
7. System ejects card.
8. System displays Welcome message.

Alternatives:

• If the system determines that the account number is invalid, it displays an error message and ejects the card.
• If the system determines that there are insufficient funds in the customer’s account, it displays an apology and ejects the card.
• If the system determines that the maximum allowable daily withdrawal amount has been exceeded, it displays an apology and ejects the card.
• If the ATM is out of funds, the system displays an apology, ejects the card, and shuts down the ATM.

Postcondition: Customer funds have been withdrawn.
Use Case Relationships

• **Extend** relationship
  – Provides conditional variation to use cases
  – Extension points identified in the base use case
  – Extension use cases executed at the specified extension points based on identified conditions
  – Extension use cases are generally not complete
  – Base use case *is* complete without the extended use cases
    • Extension points are simply “hooks”
  – Same use case can be extended in different ways

• When to use **extend**
  – Show conditional parts of use case
  – Model complex or alternative paths

• Example
  – Pay by Cash **extends** Checkout Customer
Figure 6.11 Example of extend relationship
Example of Use Case with Extensions

Use Case Name: Checkout Customer
Summary: System checks out customer.
Actor: Customer
Precondition: Checkout station is idle, displaying a “Welcome” message.
Main Sequence:
1. Customer scans selected item.
2. System displays the item name, price, and cumulative total.
3. Customer repeats steps 1 and 2 for each item being purchased.
5. System prompts for payment by cash, credit card, or debit card.
6. «payment»
7. System displays thank-you screen.

Use Case Name: Pay by Cash
Summary: Customer pays by cash for items purchased.
Actor: Customer
Dependency: Extends Checkout Customer.
Precondition: Customer has scanned items but not yet paid for them.
Description of insertion segment:
1. Customer selects payment by cash.
2. System prompts customer to deposit cash in bills and/or coins.
3. Customer enters cash amount.
4. System computes change.
5. System displays total amount due, cash payment, and change.
6. System prints total amount due, cash payment, and change on receipt.
Additional Use Case Options

• Dealing with large number of use cases
  – Can structure use cases in packages
    • Related functionality
    • Related actors
• Can use graphical modeling to document steps within use case
  – Modeled with (subset of) UML activity diagrams
    • Similar to flow chart
Case Study: Banking System

- Multiple Automated Teller Machines (ATM)
  - Customer inserts ATM Card
  - Enters Personal Identification Number (PIN)
  - ATM Transactions
    - PIN Validation
    - Withdraw Funds from Checking or Savings Account
    - Query Account
    - Transfer funds between accounts
- Banking System maintains information about
  - Customers
  - Debit cards
  - Checking and Savings Accounts
Figure 21.1 Banking System use case model
Use Case Summary

- Use case models provide the transition between software/system requirements specifications and software design
  - Actors: identify external roles (human, system, or device)
  - Use cases: identify black-box functionality assigned to the software system
- Must involve stakeholders when constructing use case models
  - In general, stakeholders can understand actors and use cases
- Must have well-written specifications, not just drawings