Introduction to Software Testing
(2nd edition)
Chapter 7.6

Graph Coverage for Use Cases

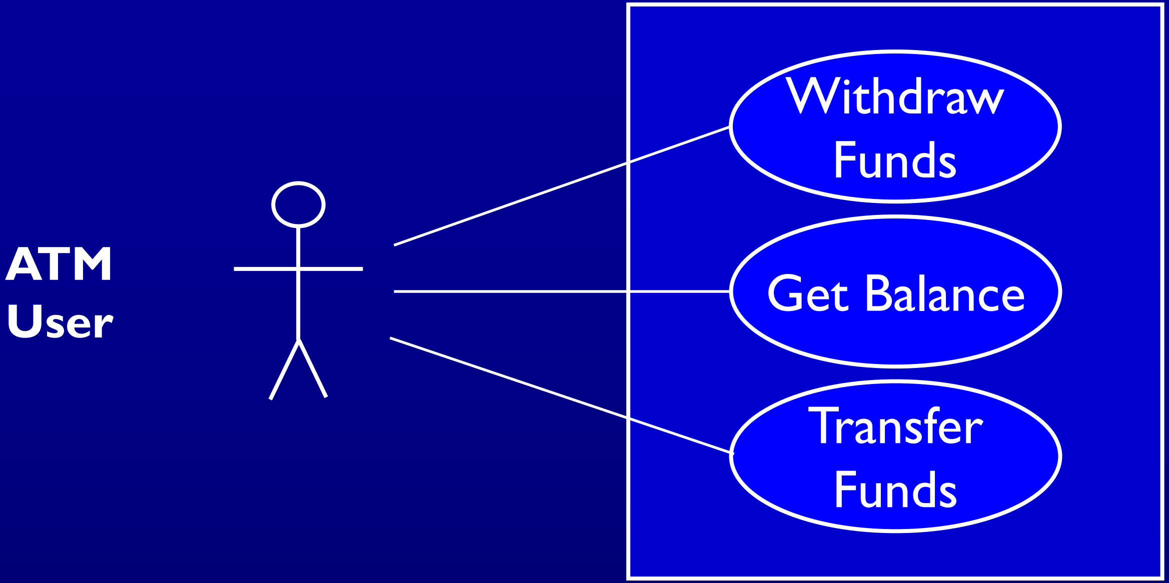
Paul Ammann & Jeff Offutt

http://www.cs.gmu.edu/~offutt/softwaretest/

UML Use Cases

- UML use cases are often used to express software requirements
- They help express computer application workflow
- We won’t teach use cases, but show examples

Simple Use Case Example



- Actors : Humans or software components that use the software being modeled
- Use cases : Shown as circles or ovals
- Node Coverage : Try each use case once …

Use case graphs, by themselves, are not useful for testing

Elaboration

- Use cases are commonly elaborated (or documented)
- Elaboration is first written textually
  - Details of operation
  - Alternatives model choices and conditions during execution

Elaboration of ATM Use Case

- Use Case Name : Withdraw Funds
- Summary : Customer uses a valid card to withdraw funds from a valid bank account.
- Actor : ATM Customer
- Precondition : ATM is displaying the idle welcome message
- Description :
  - Customer inserts an ATM Card into the ATM Card Reader.
  - If the system can recognize the card, it reads the card number.
  - System prompts the customer for a PIN.
  - Customer enters PIN.
  - System checks the card's expiration date and whether the card has been stolen or lost.
  - If the card is valid, the system checks if the entered PIN matches the card PIN.
  - If the PINs match, the system finds out what accounts the card can access.
  - System displays customer accounts and prompts the customer to choose a type of transaction. There are three types of transactions, Withdraw Funds, Get Balance and Transfer Funds. (The previous eight steps are part of all three use cases; the following steps are unique to the Withdraw Funds use case.)

Elaboration of ATM Use Case—(2/3)

- Description (continued) :
  - Customer selects Withdraw Funds, selects the account number, and enters the amount.
  - System checks that the account is valid, makes sure that customer has enough funds in the account, makes sure that the daily limit has not been exceeded, and checks that the ATM has enough funds.
  - If all four checks are successful, the system dispenses the cash.
  - System prints a receipt with a transaction number, the transaction type, the amount withdrawn, and the new account balance.
  - System ejects card.
  - System displays the idle welcome message.

Elaboration of ATM Use Case—(3/3)

- Alternatives :
  - If the system cannot recognize the card, it is ejected and the welcome message is displayed.
  - If the current date is past the card's expiration date, the card is confiscated and the welcome message is displayed.
  - If the card has been reported lost or stolen, it is confiscated and the welcome message is displayed.
  - If the customer entered PIN does not match the PIN for the card, the system prompts for a new PIN.
  - If the customer enters an incorrect PIN three times, the card is confiscated and the welcome message is displayed.
  - If the account number entered by the user is invalid, the system displays an error message, ejects the card and the welcome message is displayed.
  - If the request for withdraw exceeds the maximum allowable daily withdrawal amount, the system displays an apology message, ejects the card and the welcome message is displayed.
  - If the request for withdraw exceeds the amount of funds in the ATM, the system displays an apology message, ejects the card and the welcome message is displayed.
  - If the customer enters Cancel, the system cancels the transaction, ejects the card and the welcome message is displayed.
- Postcondition :
  - Funds have been withdrawn from the customer's account.

Wait A Minute …

- What does this have to do with testing ?
- Specifically, what does this have to do with graphs ???
- Remember our admonition : Find a graph, then cover it!
- Beizer suggested "Transaction Flow Graphs" in his book
- UML has something very similar :

Activity Diagrams

Use Cases to Activity Diagrams

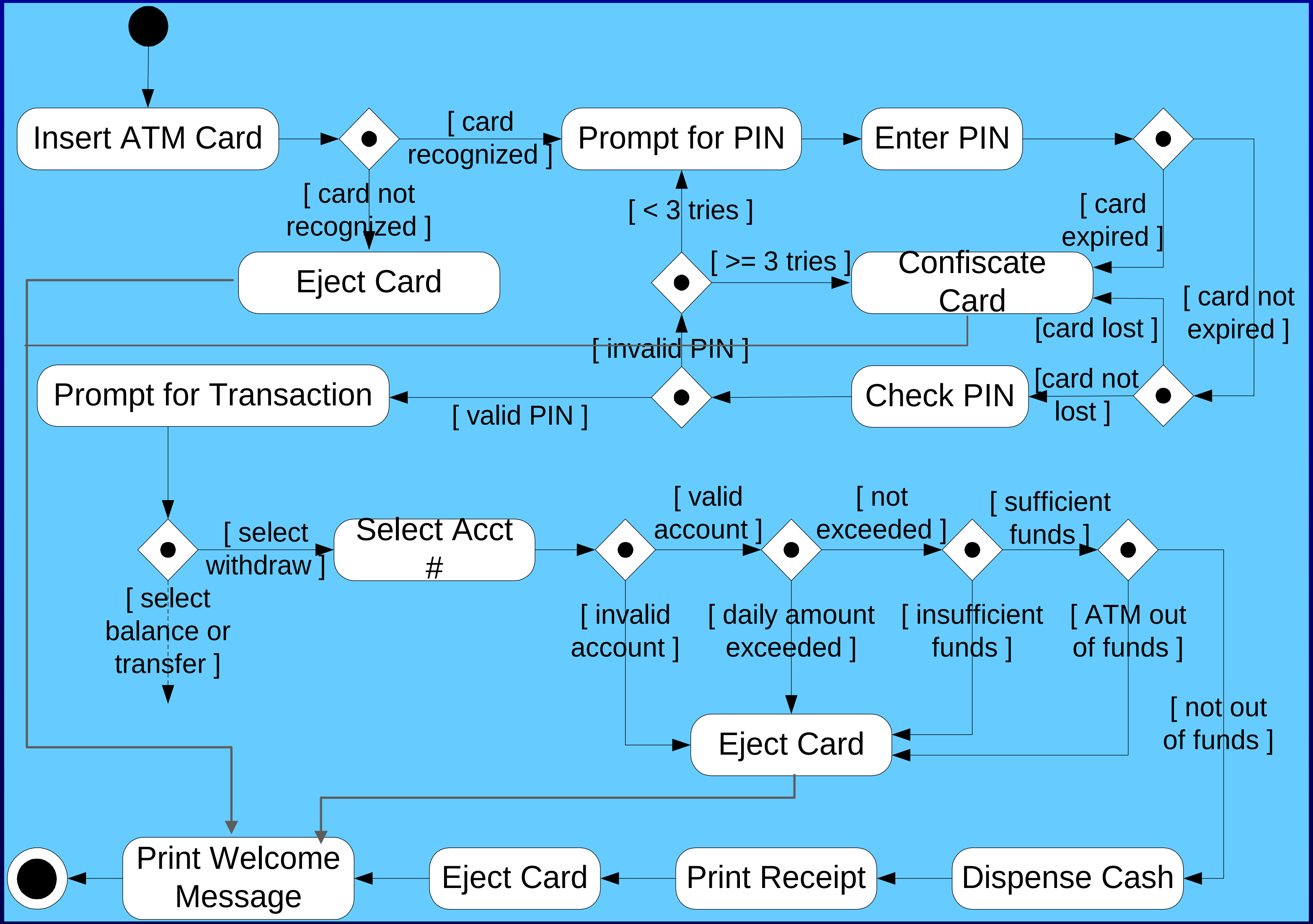
- Activity diagrams indicate flow among activities
- Activities should model user level steps
- Two kinds of nodes:
  - Action states
  - Sequential branches
- Use case descriptions become action state nodes in the activity diagram
- Alternatives are sequential branch nodes
  - Alternatives represent decisions
- Flow among steps are edges

Use Cases to Activity Diagrams

- Activity diagrams usually have some helpful characteristics:

1. Few loops
   - They usually do not have many loops, and most loops they contain are tightly bounded or determinate
   - For example, ATM withdraw activity graph contains a three-iteration loop when the PIN is entered incorrectly
   - So complete path coverage is often feasible and sometimes reasonable
2. Simple predicates
   - This is because the use case is usually expressed in terms that the users can understand
   - This means that the logic coverage criteria are usually not useful
3. No obvious DU pairs
   - This means that data flow coverage criteria are not applicable

ATM Withdraw Activity Graph



Covering Activity Graphs

- Two criteria that are most obviously applicable to use case graphs are:
  - Node Coverage
  - Edge Coverage
- Test case values are derived from labels on nodes and predicates
- Data flow techniques do not apply
- One other criterion for use case graphs is based on the notion of "scenarios"

Covering Activity Graphs

- Scenario Testing
  - Scenario : A complete path through a use case activity graph
  - Should make semantic sense to the users
  - Number of paths often finite
    - Use case graph is usually finite and it is possible to list all possible scenarios
  - If not, scenarios defined based on domain knowledge
    - Domain knowledge can be used to reduce the number of scenarios that are useful or interesting
  - Use "specified path coverage," where the set S of paths is the set of scenarios
    - Specified path coverage is exactly what we want here (instead of complete path coverage)
    - Choose useful scenarios among all possible scenarios (according to domain knowledge)

Covering Activity Graphs

- Scenario Testing (continue)
  - If the tester chooses all possible paths as scenarios, then specified path coverage is equivalent to complete path coverage
  - Scenarios are chosen by people and they depend on domain knowledge
  - Note that specified path coverage does not necessarily subsume edge coverage or node coverage, but scenarios should be defined so that it does

Summary of Use Case Testing

- Use cases are defined at the requirements level
- Can be very high level
- UML Activity Diagrams encode use cases in graphs
  - Graphs usually have a fairly simple structure
- Requirements-based testing can use graph coverage
  - Straightforward to do by hand
  - Specified path coverage makes sense for these graphs