Graph Coverage for Use Cases

Idaho State University | Computer Science

Isaac Griffith

CS 4422 and CS 5599
Department of Computer Science
Idaho State University

ROAR

Idaho State University
Computer Science

Outcomes

At the end of Today's Lecture you will be able to:

- Understand how to apply graph testing concepts to UML Use Cases
- Understand how to apply graph testing concepts to UML Activity Diagrams

ROAR

Inspiration

“Pretty good testing is easy to do (that’s partly why some people like to say ‘testing is dead’– they think testing isn’t needed as a special focus because they note that anyone can find at least some bugs some of the time). Excellent testing is quite hard to do.” – James Bach

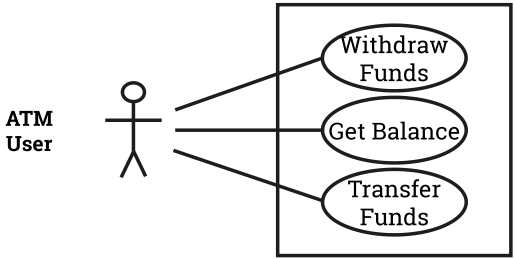
Idaho State University
Computer Science

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**

ROAR

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

Idaho State University
Computer Science

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**:
  1. Customer inserts an ATM Card into the ATM Card Reader.
  2. If the system can recognize the card, it reads the card number.
  3. System prompts the customer for a PIN.
  4. Customer enters a PIN.
  5. System checks the card's expiration date and whether the card has been stolen or lost
  6. If the card is valid, the system checks if the entered PIN matches the card PIN.
  7. IF the PINs match, the system finds out what accounts the card can access
  8. System displays customer accounts and prompts the customer to choose a type

Elaboration of ATM Use Case

- **Description** (continued):
  9. Customer selects Withdraw Funds, selects the account number, and enters the amount.
  10. 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.
  11. If all four checks are successful, the system dispenses the cash.
  12. System prints a receipt with a transaction number, the transaction type, the amount withdrawn, and the new account balance.
  13. System ejects card.
  14. System displays the idle welcome message.
- **Postcondition**:
  - Funds have been withdrawn from the customer's account.

Elaboration of ATM Use Case

- **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.

Wait a Minute...

- What does this have to 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**

Idaho State University
Computer Science

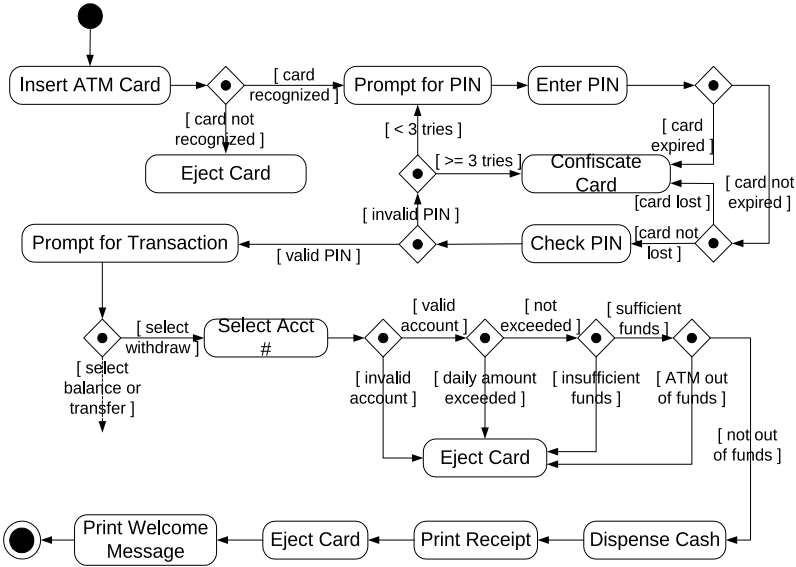
Use Cases to Activity Diagrams

- 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**
- Flow among steps are **edges**
- Activity diagrams usually have some helpful characteristics
  - Few Loops
  - Simple predicates
  - No obvious DU pairs

ROAR

Idaho State University
Computer Science

ATM Withdraw Activity Graph



ROAR

Covering Activity Graphs

- **Node Coverage**
  - Inputs to the software are derived from labels on nodes and predicates
  - Used to form test case values
- **Edge Coverage**
- Data flow techniques do **not** apply
- **Scenario Testing**
  - **Scenario**: A complete path through a use case activity graph
  - Should make **semantic** sense to the users
  - Number of paths often **finite**
  - If not, scenarios defined based on **domain knowledge**
  - Use "**specific path coverage**," where the set S of paths is the set of scenarios
  - Note that specified path coverage does not necessarily subsume edge 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

Are there any questions?