IT 561: Adv Software Engineering

Activity Diagram
(Use Case to Activity Diagram)

Saurabh Tiwari
Activity Diagram

- Activity diagrams represent the dynamic (behavioral) view of a system.

- Activity diagrams are typically used for business (transaction) process modeling and modeling the logic captured by a single use-case or usage scenario.

- Activity diagram is used to represent the flow across use cases or to represent flow within a particular use case.

- UML activity diagrams are the object oriented equivalent of flow chart and data flow diagrams in function-oriented design approach.

- Activity diagram contains activities, transitions between activities, decision points, synchronization bars, swim lanes and many more...
Activity Diagram

Elements of Activity Diagram

- Initial Node
- Final Node
- Activity (Node)
- Control Flow (Edge)
- Decision (Branch)
- Fork/Join Node
- Swim lanes
Student Enrollment in DAIICT

System

Student Enrollment

Student

<<include>>

Registrar

Course registration
SEDAIICT System

- Here different activities are:
  - Received enrollment form filled by the student
    - Registrar checks the form
    - Input data to the system
    - System authenticate the environment
  - Pay fees by the student
    - Registrar checks the amount to be remitted and prepare a bill
    - System acknowledge fee receipts and print receipt
  - Hostel allotment
    - Allot hostel
    - Receive hostel charge
    - Allot room
  - Medical check up
    - Create hostel record
    - Conduct medical bill
    - Enter record
  - Issue library card
  - Issue identity card
Activity Diagram for the Use Case

1. Received form
2. Payment fees
3. Hostel allotment
4. Medical check
5. Issue identity card
6. Issue library card

Flowchart: Start with Received form, followed by Payment fees, Hostel allotment, Medical check, Issue identity card, and Issue library card.
Basic Components in an Activity Diagram

- **Initial node**
  - The filled circle is the starting point of the diagram

- **Final node**
  - The filled circle with a border is the ending point. An activity diagram can have zero or more activity final state.

- **Activity**
  - The rounded circle represents activities that occur. An activity is not necessarily a program, it may be a manual thing also

- **Flow/edge**
  - The arrows in the diagram. No label is necessary

```plaintext
Received form

Payment fees --> Hostel allotment --> Medical check

Issue identity card

Issue library card
```
Basic Components in an Activity Diagram

- **Fork**
  - A black bar (horizontal/vertical) with one flow going into it and several leaving it. This denotes the beginning of parallel activities.

- **Join**
  - A block bar with several flows entering it and one leaving it. This denotes the end of parallel activities.

- **Merge**
  - A diamond with several flows entering and one leaving. The implication is that all incoming flow to reach this point until processing continues.
• Difference between Join and Merge

  – A join is different from a merge in that the join synchronizes two inflows and produces a single outflow. The outflow from a join cannot execute until all inflows have been received.

  – A merge passes any control flows straight through it. If two or more inflows are received by a merge symbol, the action pointed to by its outflow is executed two or more times.
Basic Components in an Activity Diagram

• **Decision**
  – A diamond with one flow entering and several leaving. The flow leaving includes conditions as yes/ no state

• **Flow final**
  – The circle with X though it. This indicates that Process stop at this point

• **Swim lane**
  – A partition in activity diagram by means of dashed line, called swim lane. This swim lane may be horizontal or vertical
Detailed Activity Diagram

1. Received form
   - Payment fees
   - Hostel allotment
   - Medical check

2. Issue identity card
   - Issue library card
Detailed Activity Diagram

1. Fill-in form
2. Check form
3. Display student screen
4. Input student information
5. Verify the applications
6. Search for Student selection list
7. Create record
8. Regret message
   - Correct
   - Incorrect
   - [Not Found]
   - [Found]
   - [Match]
   - [No Match]
   - Regret registration
   - 1
Detailed Activity Diagram

1. Display enrollment form -> Calculate registration fees -> Display fees payable screen
   -> Pay fee

2. Process payment
   -> Print receipt
Activity Diagram with Swim Lane

Student
- Fill-in form
  - [Incorrect]
  - [Correct]
- Check form
  - [Correct]
- Display student screen

Registrar
- Verify the applications
  - [Found]
  - [Not Found]
- Regret message
  - ✗

System
- Input student information
- Search for Student selection list
  - [Match]
  - [No Match]
- Regret registration
  - ✗
- Create record
  - 1
Activity Diagram with UML 2.0

Vending Machine
Some more features in Activity Diagrams
Object and Object Flow

- An object flow is a path along which objects can pass. An object is shown as a rectangle.

- An object flow is shown as a connector with an arrowhead denoting the direction the object is being passed.
Input and Output Pin

- An object flow must have an object on at least one of its ends. A shorthand notation for the above diagram would be to use input and output pins.
A data store is shown as an object with the «datastore» keyword.
Interruptible Activity Region

• An interruptible activity region surrounds a group of actions that can be interrupted. In the very simple example below, the Process Order action will execute until completion, when it will pass control to the Close Order action, unless a Cancel Request interrupt is received which will pass control to the Cancel Order action.
An Example

![Activity Diagram]

- Requested Order
  - Receive Order
  - Process Order
    - Fill Order
    - Ship Order
    - Close Order
  - [order rejected]
    - Send Invoice
    - Make Payment
    - Accept Payment
  - [order accepted]

In this activity diagram, the process begins with receiving an order. If the order is rejected, it moves to the Send Invoice and Make Payment stages. If the order is accepted, it proceeds to the Fill Order, Ship Order, and Close Order stages.
Importance of Activity Diagram

• An activity diagram can depict a model in several ways

• It can also depicts “Basic course of action” as well as “detailed courses”

• Activity diagram can also be drawn that cross several use cases, or that address just a small portion of use case

• Activity diagrams are normally employed in business process modeling. This is carried out during the initial stages of requirement analysis and specification

• Activity diagrams can be very useful to understand the complex processing activities involving many components

• The activity diagram can be used to develop interaction diagrams which help to allocate activities to classes
Problems to Ponder

• How activity diagram related to flow chart? How it defers from flow chart?

• How methods in classes and activities can be correlated?
Exercises?

- Prepare an activity diagram for computing a restaurant bill. There should be a charge for each delivered item. The total amount should be subjected to a tax and a service charge of 18% for group of six of more. For smaller groups, there should be a blank entry for a gratuity according to the customer’s discretion. Any coupons or gift certificates submitted by the customer should be subtracted.
Activities

• Total items
• Add tax
• Credit coupons and certificates
• Customer determines gratuity [less than six]
• Add 18% [six or more]

Activity diagram???
Exercises?

• Prepare an activity diagram that elaborates the details of logging into an email system. Note that entry of the user name and the password can occur in any order.

• Draw the activity diagrams for
  – Library Information System
  – Bank ATM
Use Case to Activity: Graph - Example

Use Case Model

Graph
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 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.)
• 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.
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.
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
Exercise?

<table>
<thead>
<tr>
<th>Use case Name</th>
<th>Process Sale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Cashier, Catalog System, Inventory System</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Basic Flow</th>
<th>Steps</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>Cashier starts a new sale.</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Cashier enters item identifier.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>POS System retrieve item information from the catalog system and, records sale line item and presents item description, price, and running total. Cashier repeats steps 2 until indicates done.</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>POS System calculates and presents total price.</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Cashier tells Customer the total, and asks for payment.</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Customer pays and POS System handles payment</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>POS System records completed sale and sends sale information to the external Inventory system for stock update.</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>POS System prints receipt.</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>Customer leaves with receipt and goods.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alternate Flow</th>
<th>Steps</th>
<th>Branching Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.1</td>
<td>IF the item entered by the Cashier is Invalid identifier THEN POS System Indicate error and Cashier enters the item manually.</td>
</tr>
<tr>
<td></td>
<td>7.1</td>
<td>IF the items stock gets below a predefined minimum place a reposition order THEN Cashier deletes the item</td>
</tr>
<tr>
<td></td>
<td>6.1</td>
<td>IF The Customer not have enough money THEN Customer asks the cashier to Cancel the transaction</td>
</tr>
<tr>
<td></td>
<td>6.2</td>
<td>IF Customer says they intended to pay by cash but don’t have enough cash THEN Customer uses an alternate payment method. The cashier tells customer to pay by card.</td>
</tr>
</tbody>
</table>

Precondition | Cashier is identified and authenticated. |
Postcondition | Sale is saved. Accounting and Inventory are updated. Receipt is generated. Payment authorization approvals are recorded. |

Prepare an activity diagram??
DESIGNING A CALL CENTER

GIVE THE EMPLOYEES SIX MINUTES OF BATHROOM BREAKS PER SHIFT.

TINY CUBICLES...AND WE’LL MONITOR CALLS AND HAVE INCOMPATIBLE OBJECTIVES, SUCH AS SPEED AND CUSTOMER SERVICE.

HOW’S THE PROJECT GOING?

I’M STILL COLLECTING THE ABUSER REQUIREMENTS.
Questions?

Next Lectures...
Sequence diagram: Identification of Domain Objects
Use Case Realization

Use Case Model

Use Case

<<realizes>>

Use Case Realization

Design Model

Sequence Diagrams

Collaboration Diagrams

Class Diagrams

Use Case Realization Documentation