

Scenario-Based Modeling

Activity Diagram : Control Flow

Elements of the Analysis Model

Object-oriented Analysis

Scenario-based modeling

Use case text
Use case diagrams
Activity diagrams
Swim lane diagrams

Class-based modeling

Class diagrams
Analysis packages
CRC models
Collaboration diagrams

Procedural/Structured Analysis

Flow-oriented modeling

Data structure diagrams
Data flow diagrams
Control-flow diagrams
Processing narratives

Behavioral modeling

State diagrams
Sequence diagrams

WHAT IS AN ACTIVITY DIAGRAM?

- The UML activity diagram supplements the use case by providing a graphical representation of the flow of interaction within a specific scenario.
- It is used to model the flow of activities, actions, and interactions within a system.
- It provides a visual representation of processes, workflows, and scenarios.
- It is similar to traditional flowchart, but that flowchart techniques lack constructs for expressing concurrency

Developing an Activity Diagram

- Describes how activities are coordinated.
- Is particularly useful when you know that an operation has to achieve a number of different things, and you want to model what the essential dependencies between them are, before you decide in what order to do them.
- Record the dependencies between activities, such as which things can happen in parallel and what must be finished before something else can start.

Activity Diagrams

Activity diagrams commonly contain

- Activity states and action states
- Transitions
- Objects

Purpose

- Activity diagrams help in understanding the dynamic behavior of a system, focusing on how activities and tasks are performed.
- They aid in modeling business processes, software behaviors, and system interactions.

Symbols and Notations

- **Start and End Nodes:**
 - The solid circle represents the start node, indicating where the activity diagram begins.
 - The solid concentric circle represents the end node, indicating the termination point.
- **Action State:**
 - Action states represent activities or actions performed within the system.
 - They are represented by rectangular nodes.
- **Decision Node:**
 - Decision nodes denote a branching point in the flow based on conditions.
 - They are represented by a diamond-shaped node.
- **Fork and Join Nodes:**
 - Fork nodes split a single flow into multiple parallel flows.
 - Join nodes synchronize parallel flows back into a single flow.
- **Merge Node:**
 - Merge nodes combine multiple flows into a single flow.

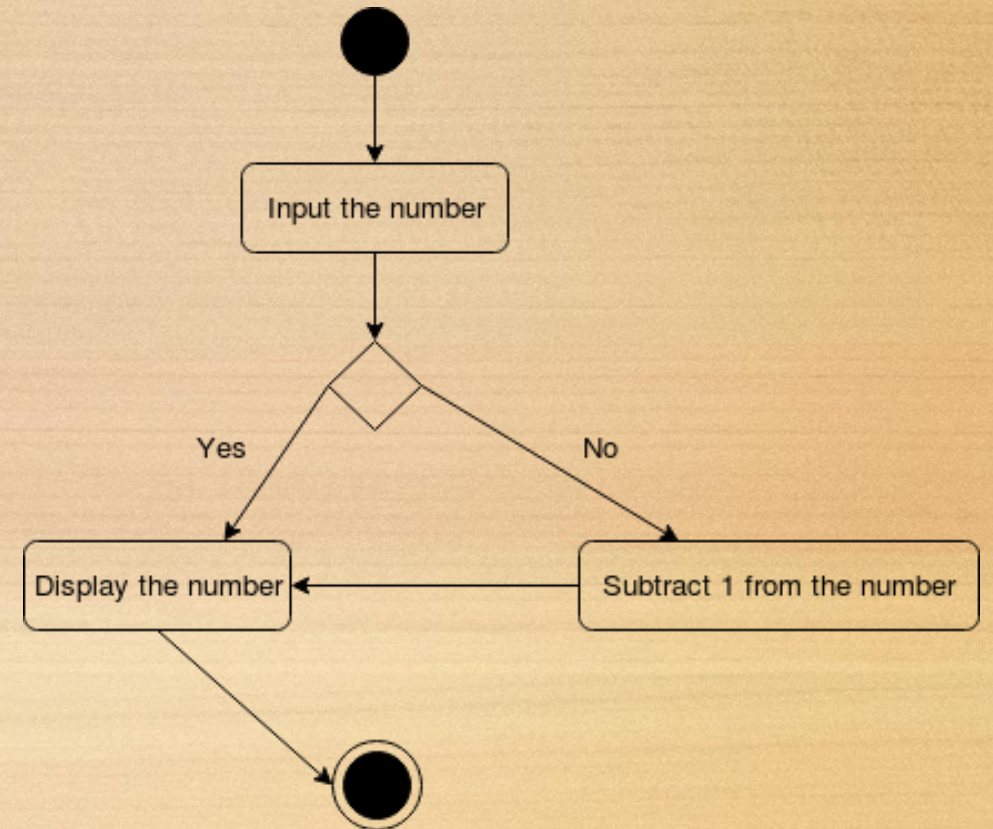
Start and End Nodes:



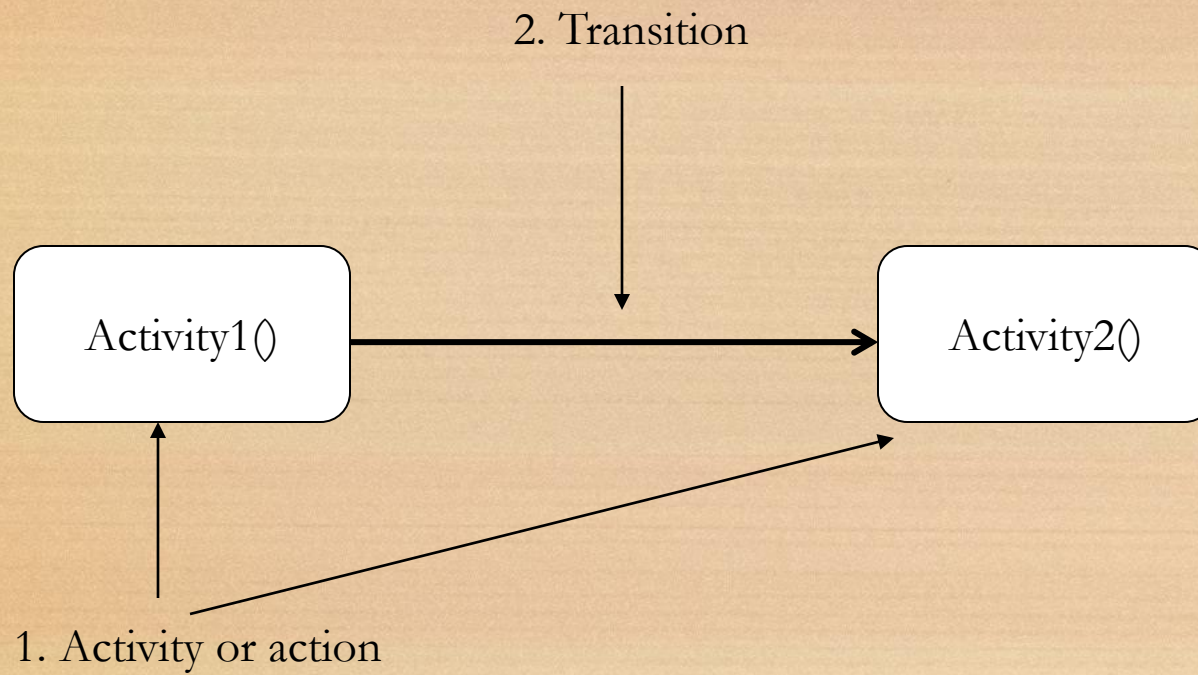
Start Marker



Stop Marker



ACTIVITY / ACTION



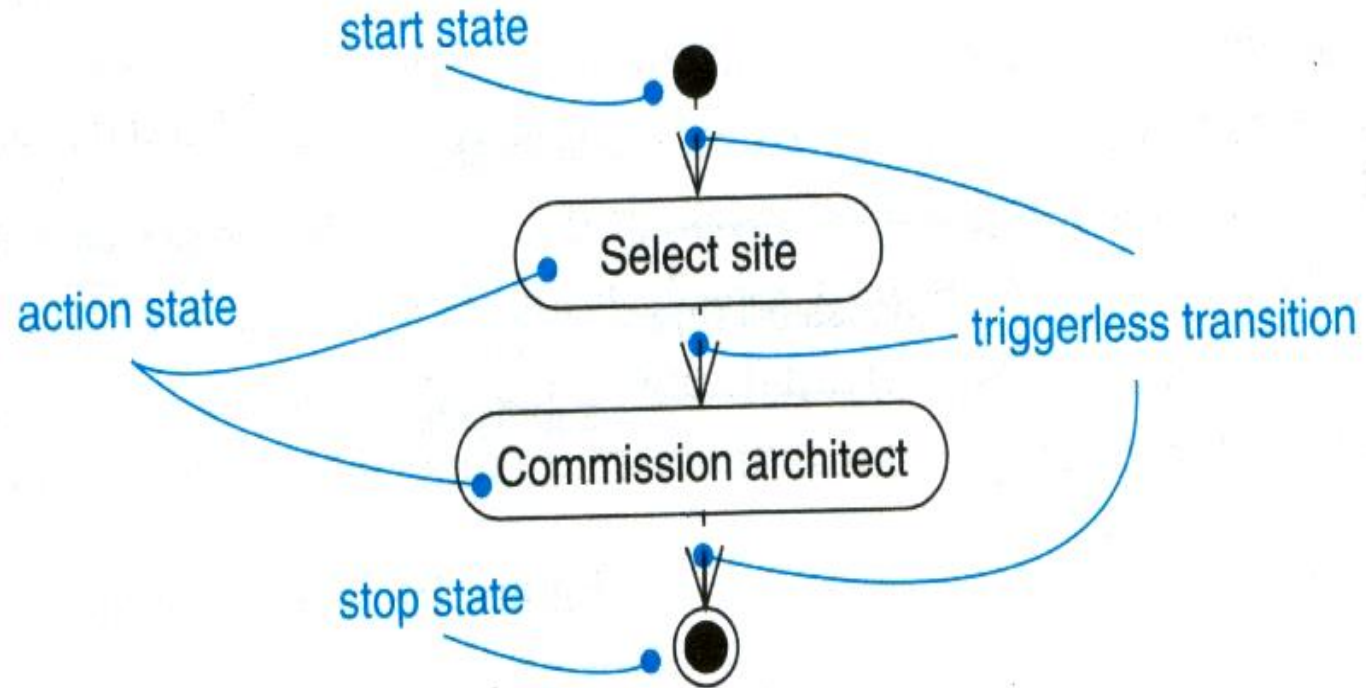
Action States and Activity States

- Action states are atomic and cannot be decomposed
 - Work of the action state is not interrupted
- Activity states can be further decomposed
 - Their activity being represented by other activity diagrams
 - They may be interrupted

Transitions

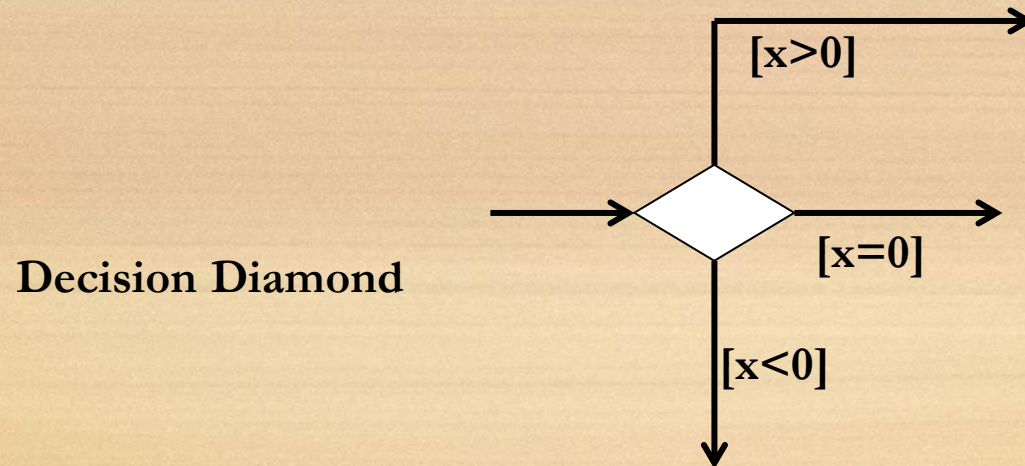
- When the action or activity of a state completes, flow of control passes immediately to the next action or activity state
- A flow of control has to start and end someplace
 - initial state -- a solid ball
 - stop state -- a solid ball inside a circle

Transitions

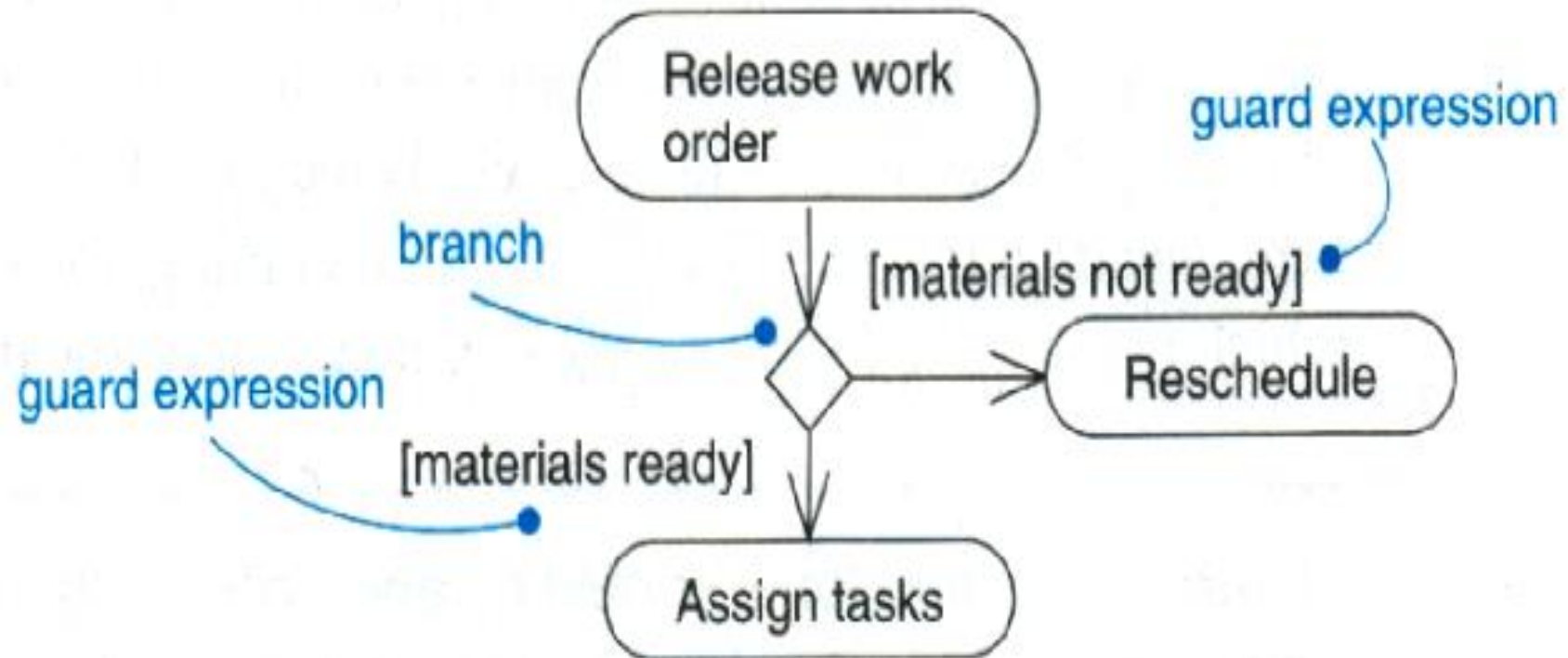


DECISION NODE

- A branch specifies alternate paths taken based on some Boolean expression
- A branch may have one incoming transition and two or more outgoing ones

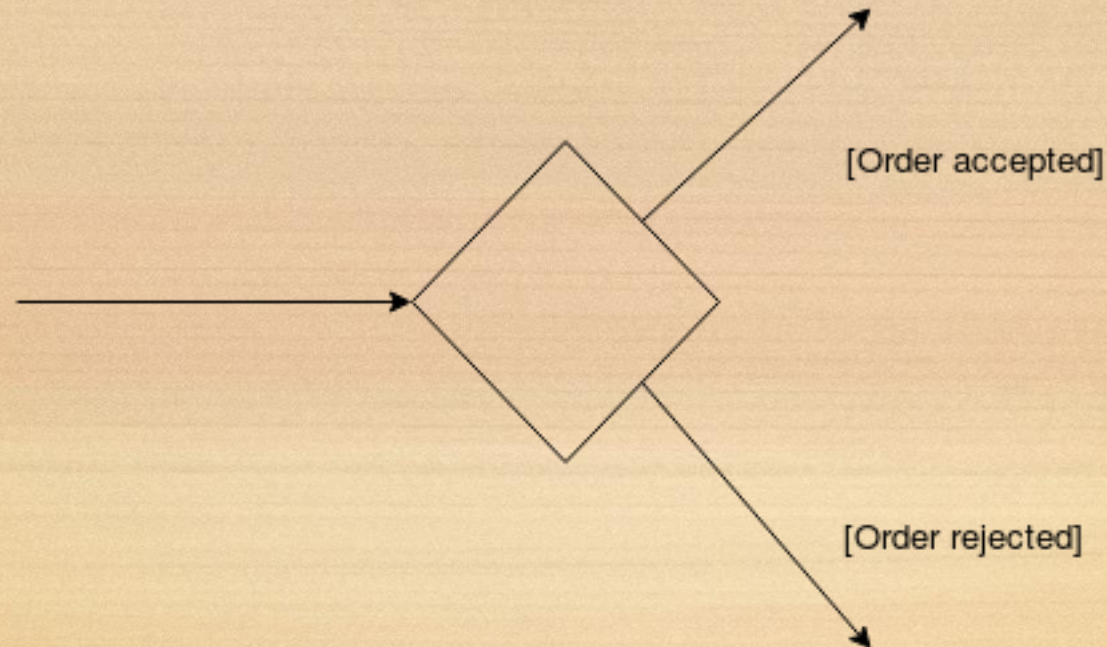


Decision or Branching



DECISION NODE - GUARD

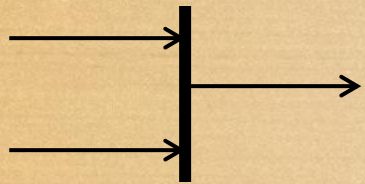
Guards – A Guard refers to a statement written next to a decision node on an arrow sometimes within square brackets.



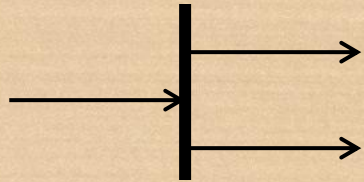
ORKING AND JOINING

- Use a synchronization bar to specify the forking and joining of parallel flows of control
- A synchronization bar is rendered as a thick horizontal or vertical line

FORK AND JOIN NODES:

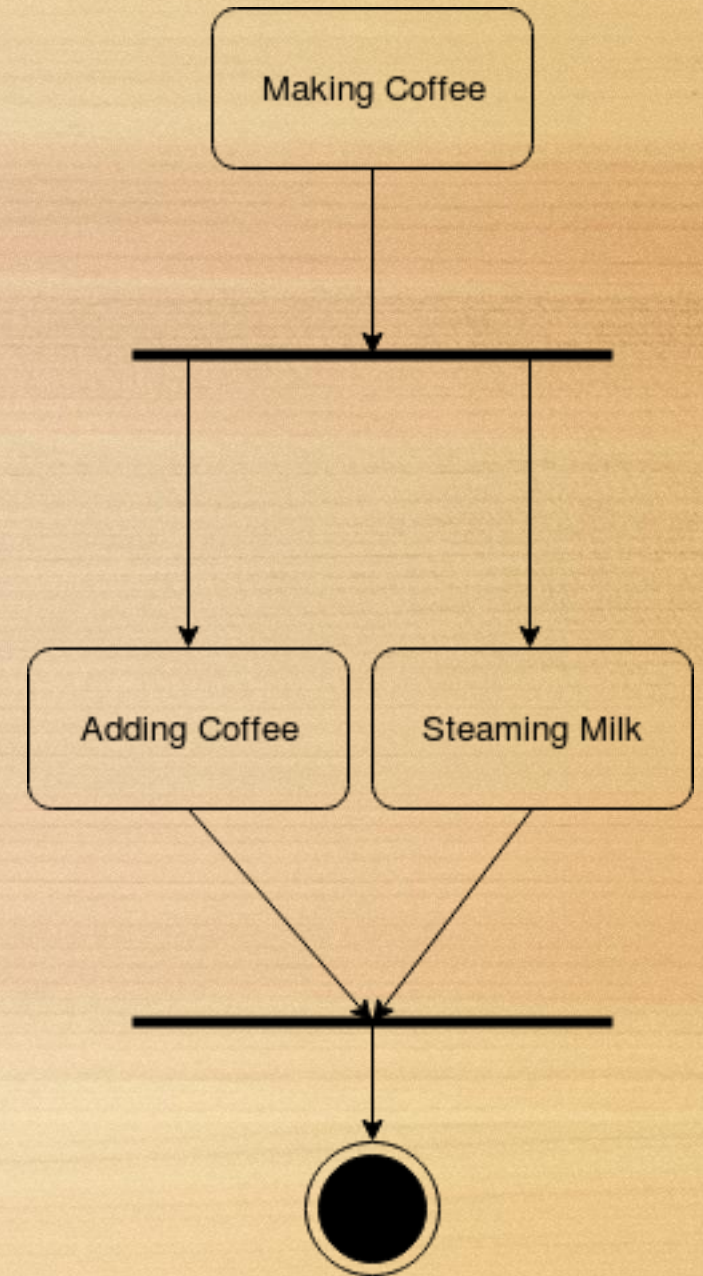


Synch. Bar (Join)



Splitting Bar (Fork)

For Concurrent Activities



Fork

- A fork may have one incoming transitions and two or more outgoing transitions
 - each transition represents an independent flow of control
 - conceptually, the activities of each of outgoing transitions are concurrent

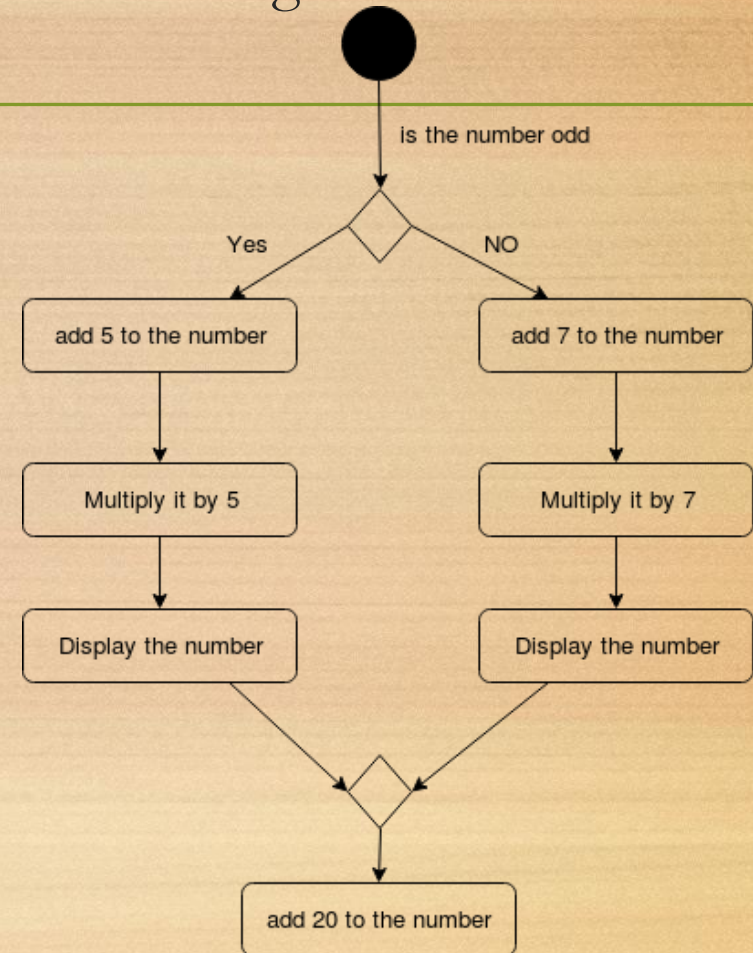
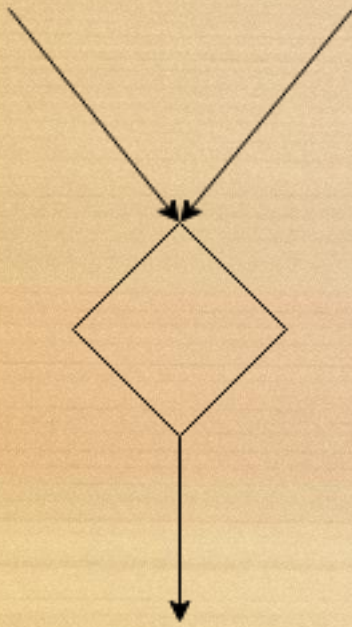
Join

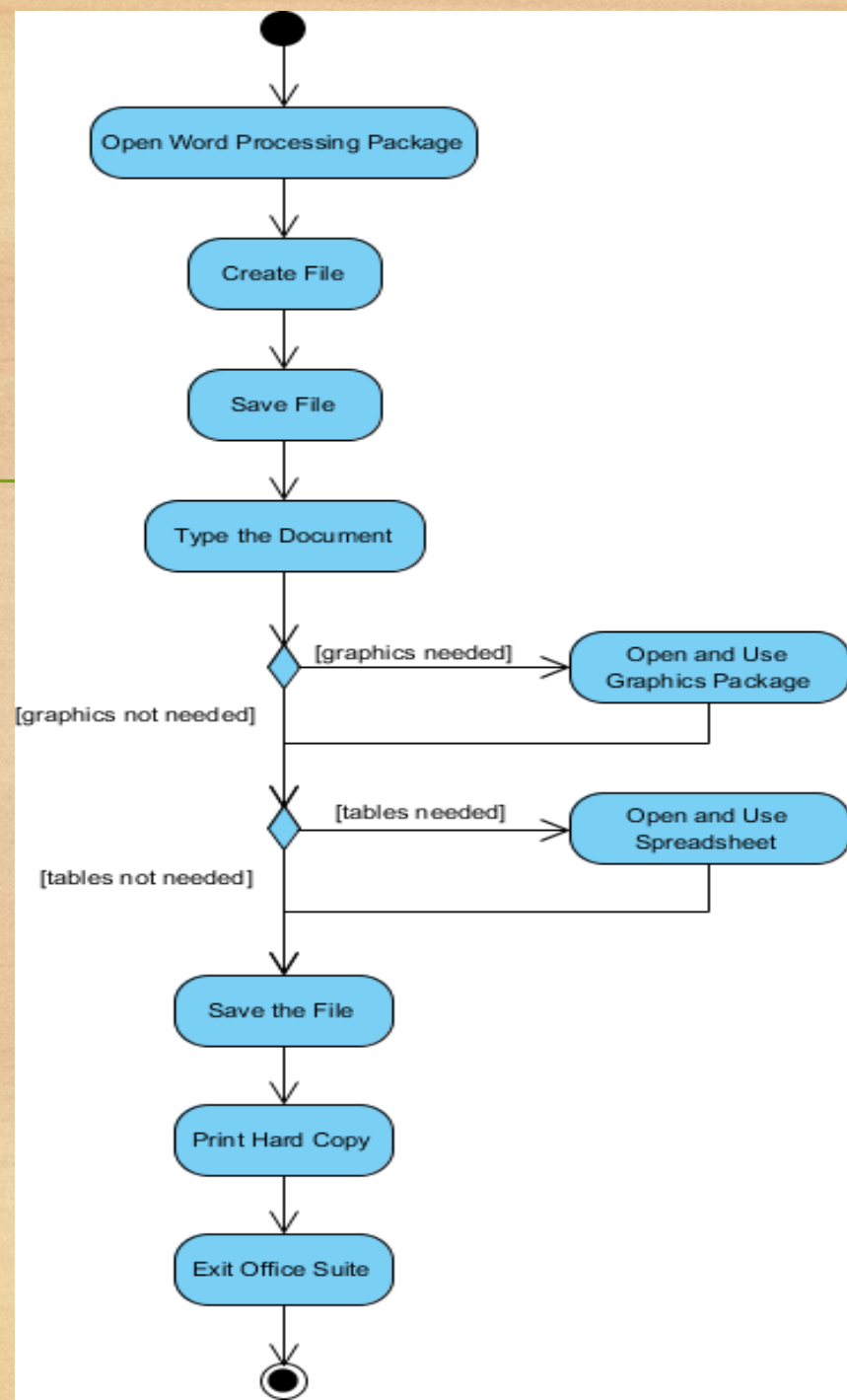
- A join may have two or more incoming transitions and one outgoing transition

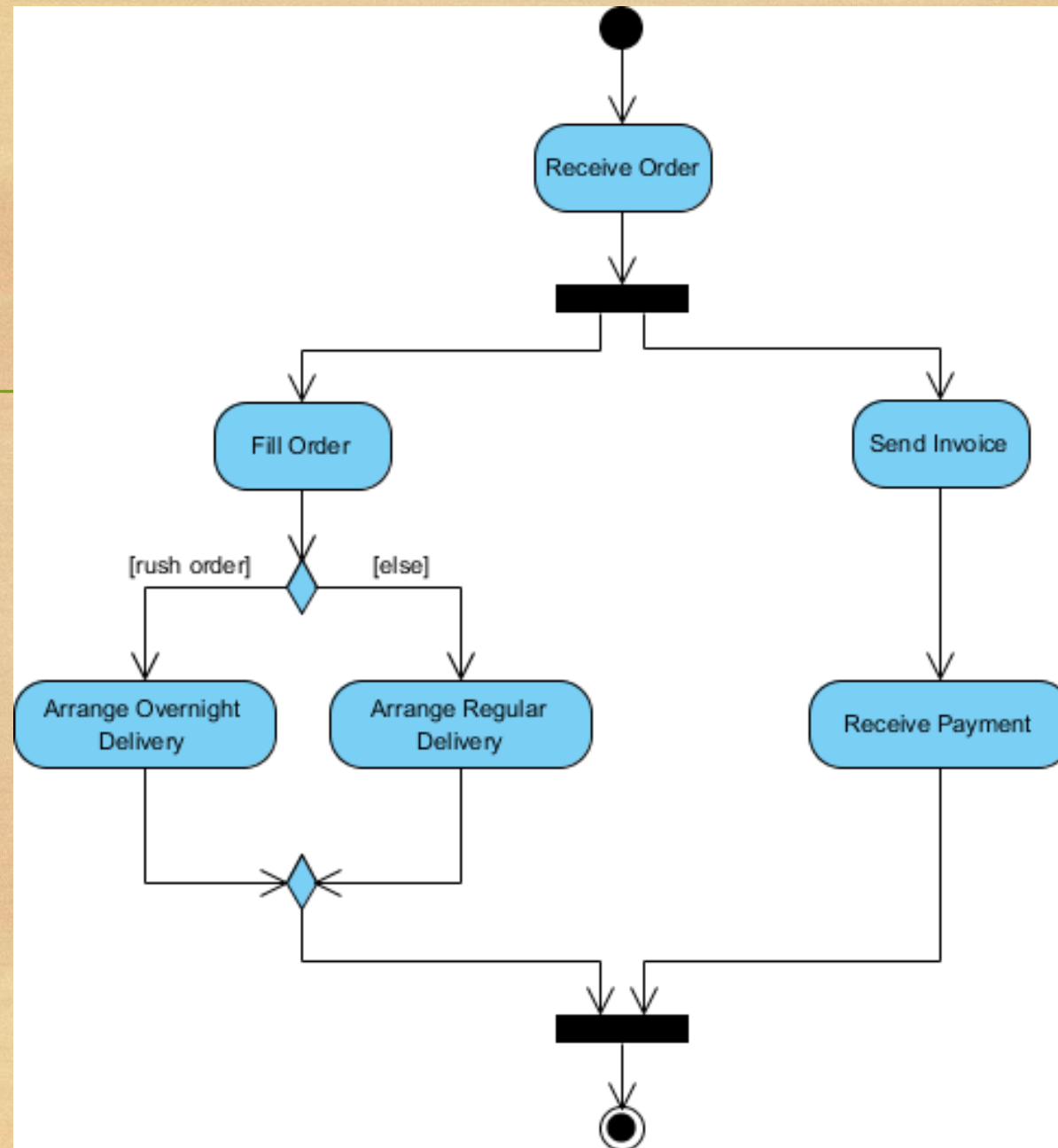
- above the join, the activities associated with each of these paths continues in parallel
- at the join, the concurrent flows synchronize
 - each waits until all incoming flows have reached the join, at which point one flow of control continues on below the join

Merge Node

- **Merge** – Scenarios arise when activities which are not being executed concurrently have to be merged. We use the merge notation for such scenarios. We can merge two or more activities into one if the control proceeds







Activity Diagram: Example

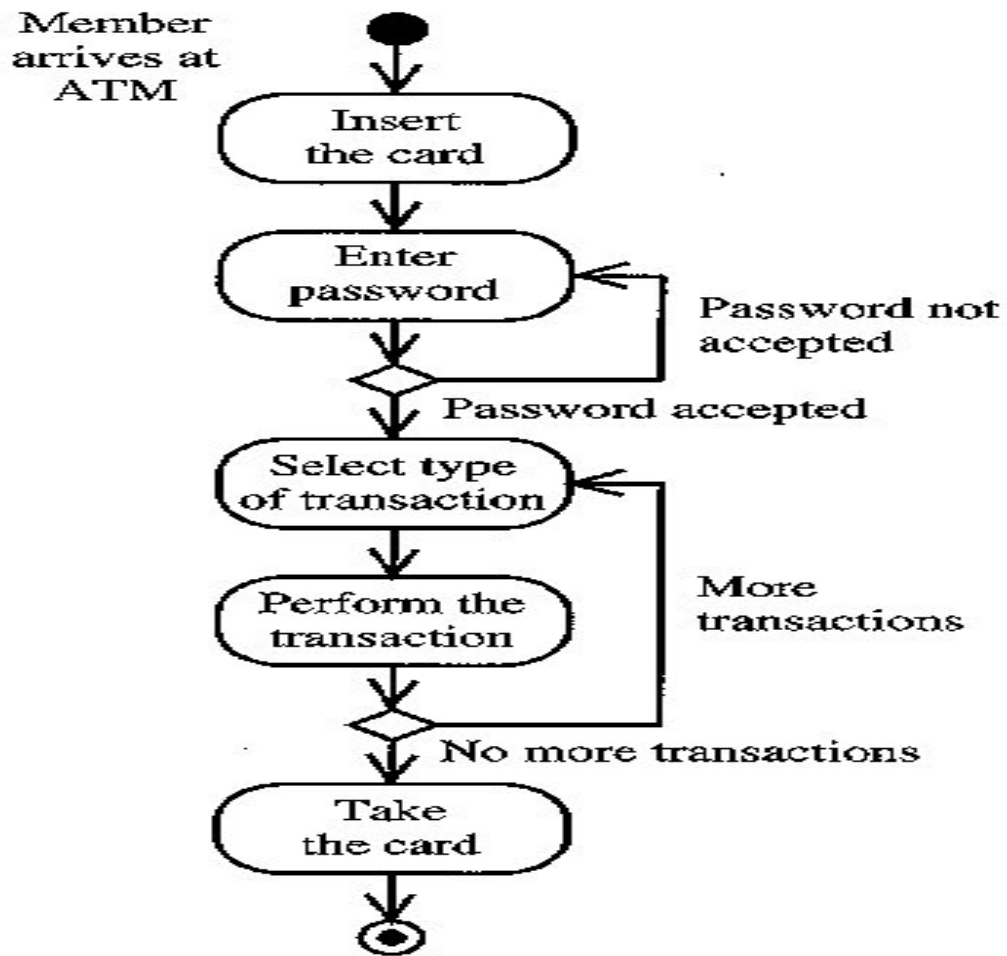
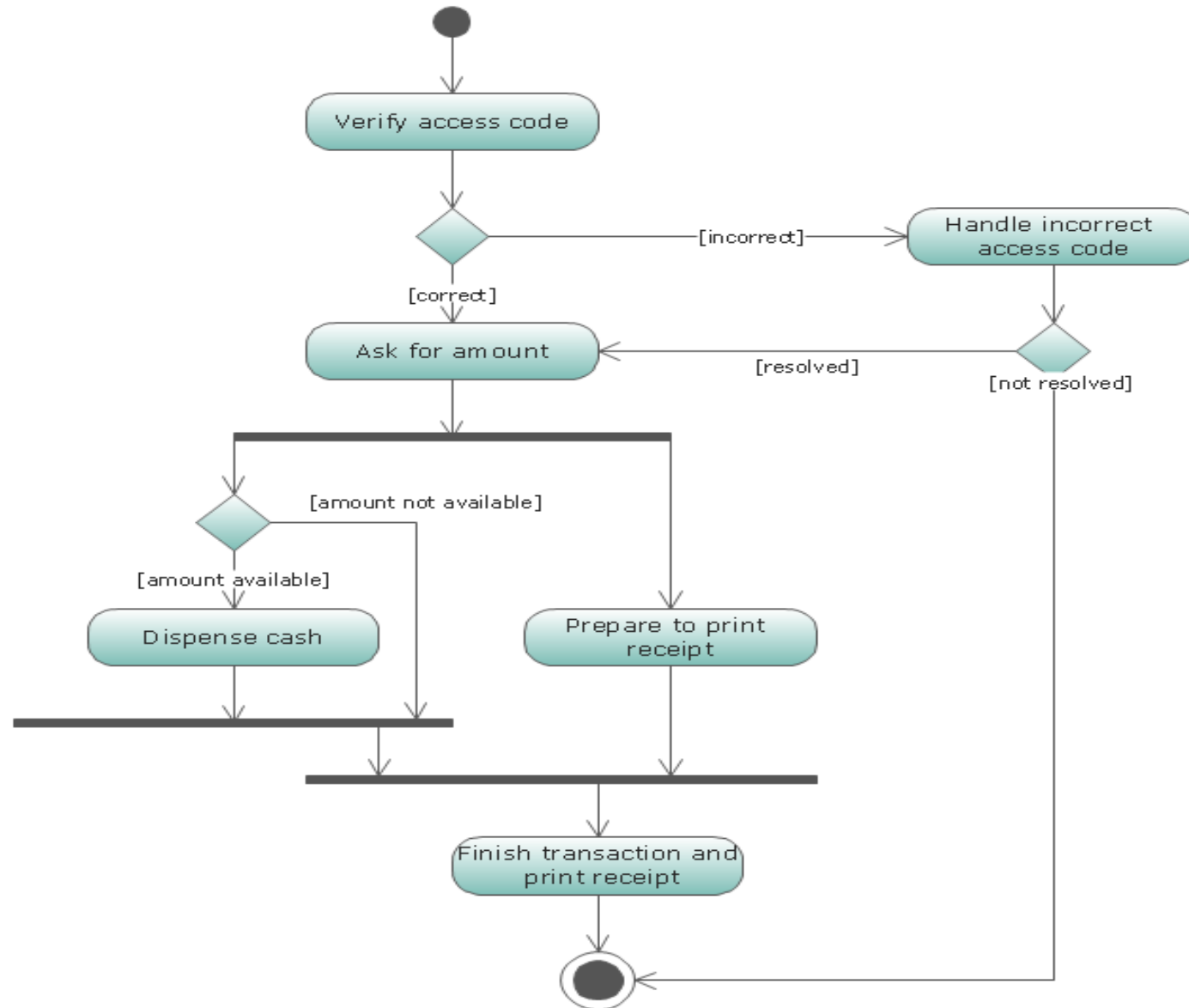
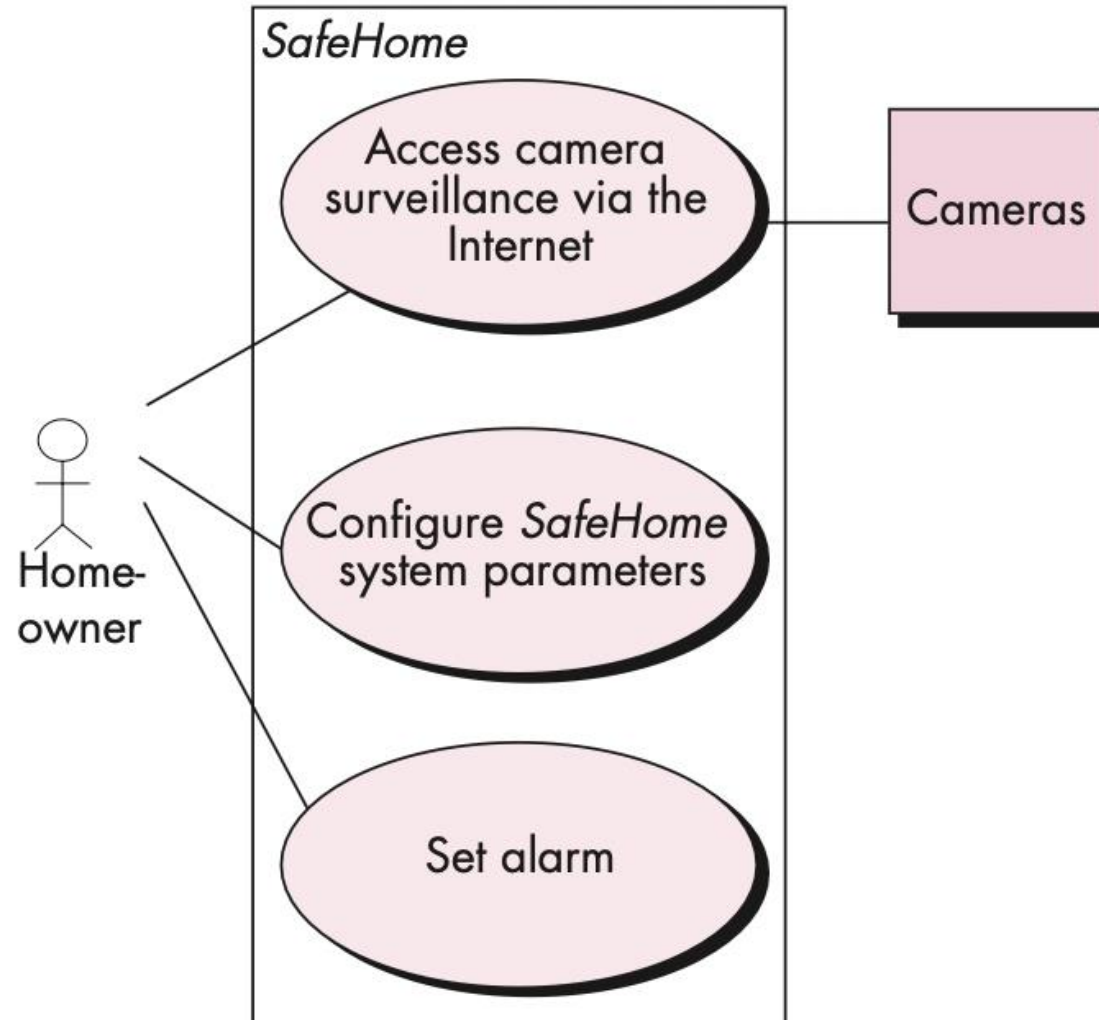


FIGURE 6-8
Activities involved in an ATM transaction.

UML Activity Diagram





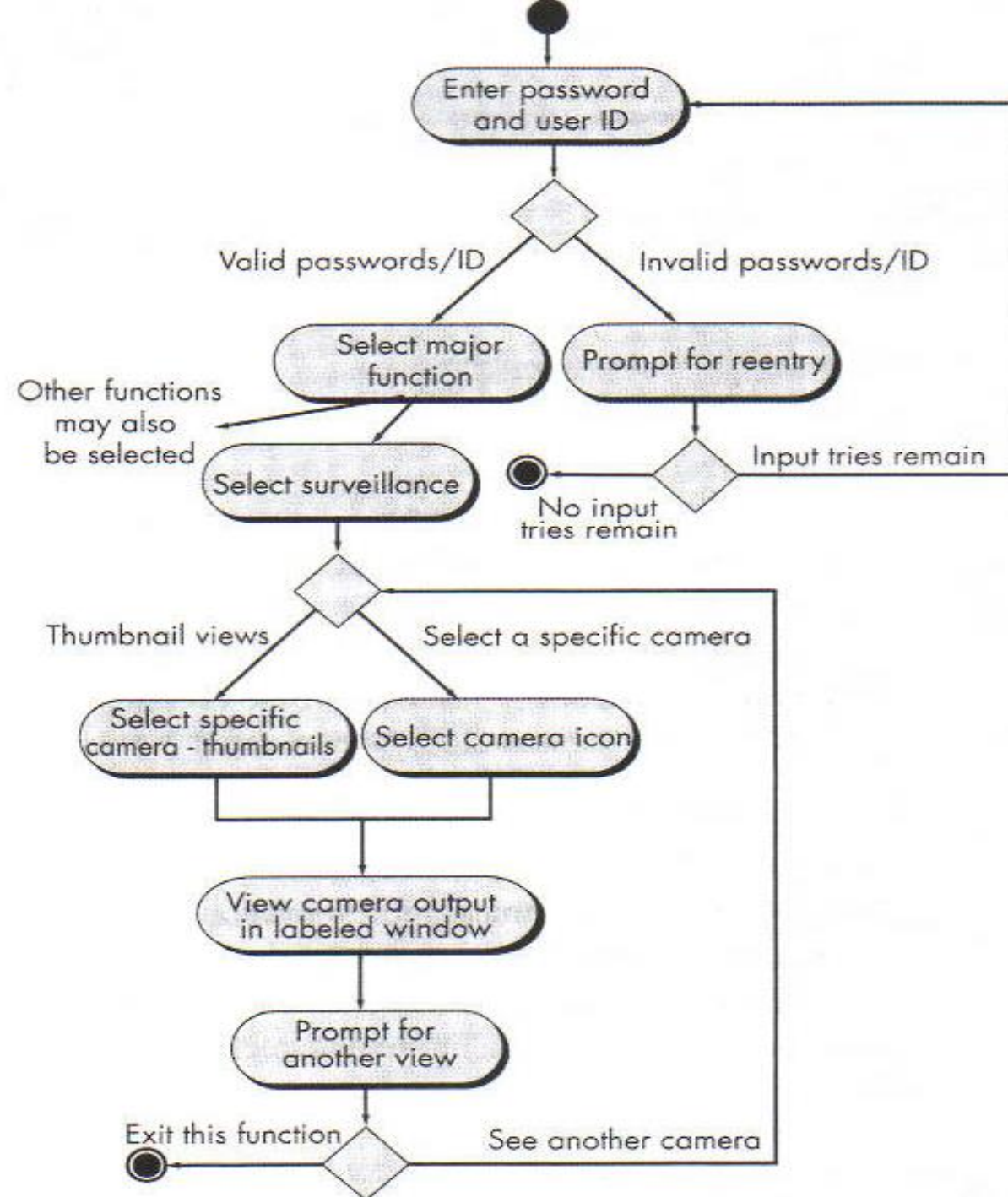
Use-case of Safehome system

Scenario:

1. The homeowner logs onto the *SafeHome Products* Web site.
2. The homeowner enters his or her user ID.
3. The homeowner enters two passwords (each at least eight characters in length).
4. The system displays all major function buttons.
5. The homeowner selects "surveillance" from the major function buttons.
6. The homeowner selects "pick a camera."
7. The system displays the floor plan of the house.
8. The homeowner selects a camera icon from the floor plan.
9. The homeowner selects the "view" button.
10. The system displays a viewing window that is identified by the camera ID.
11. The system displays video output within the viewing window at one frame per second.

UML Activity Diagram

Activity
diagram for
Access
camera
surveillance—
display
camera views
function



Swimlane Diagrams

- A variation of activity diagram
- Indicate which actor has responsibility for the action described by an activity rectangle
- A UML swimlane diagram represents the flow of actions, decisions and indicates which actors perform each

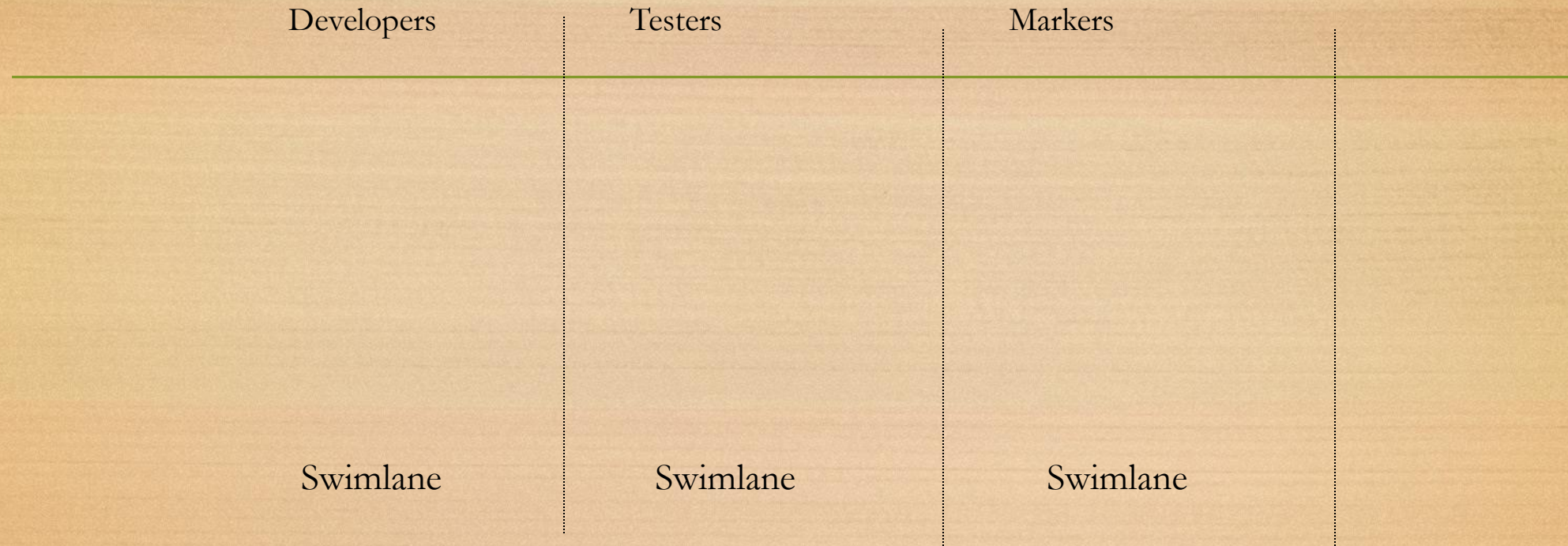
Swimlanes (1)

- A swimlane specifies a locus of activities
- To partition the activity states on an activity diagram into groups
 - each group representing the business organization responsible for those activities
 - each group is called a swimlane
- Each swimlane is divided from its neighbor by a vertical solid line

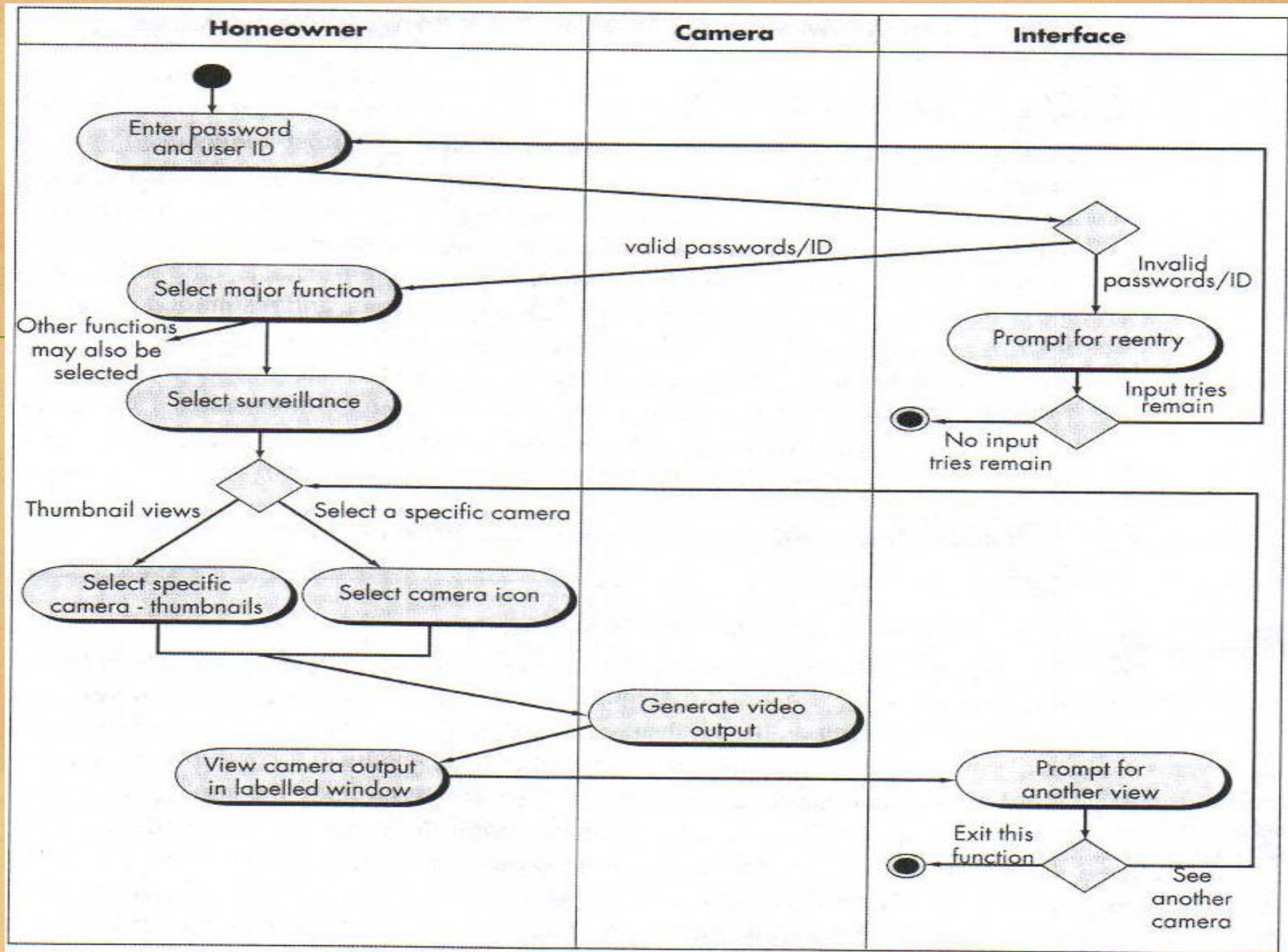
Swimlanes (2)

- Each swimlane has a name unique within its diagram
- Each swimlane may represent some real-world entity
- Each swimlane may be implemented by one or more classes
- Every activity belongs to exactly one swimlane, but transitions may cross lanes

Swimlanes and Actors



Application/Department/Group/Role Boundaries



Facebook is an online social media website, where the users must register before using the it. Once the user is authenticated by site, he can retrieve notifications, messages and requests. Further, in the next step the user can change the user preferences as online or hidden. Finally the status will be displayed in user's wall. User has option to login by entering credentials for three times. Third unsuccessful login will lock the account, send warning email of the user and display 'account locker' message on the screen. Else the system will prompt for login credentials again.

