



Vietnam National University of HCMC
International University
School of Computer Science and Engineering



Object – Oriented Analysis and Design

Activity Diagram

Instructor: Le Thi Ngoc Hanh, Ph.D

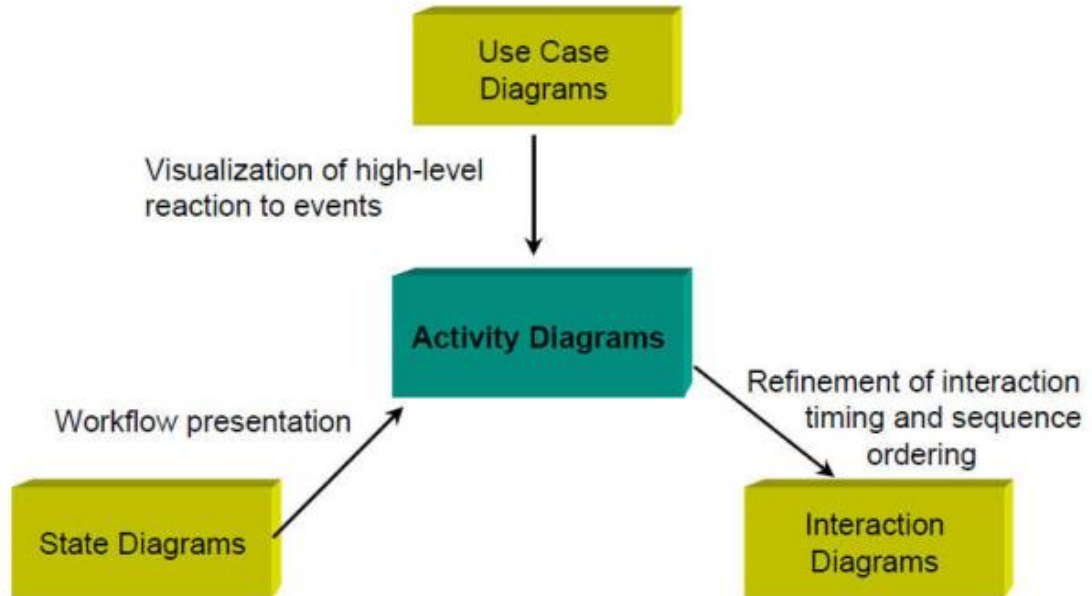
ltnhanh@hcmiu.edu.vn

Outline

- 💧 Activity diagram in UML
- 💧 Components in Activity diagram
- 💧 Reading:
 - [R2] - Chapter 13, Section 13.3
 - [R3] – Chapter 5, Section 5.6

@ Image credit: uml-diagram.org

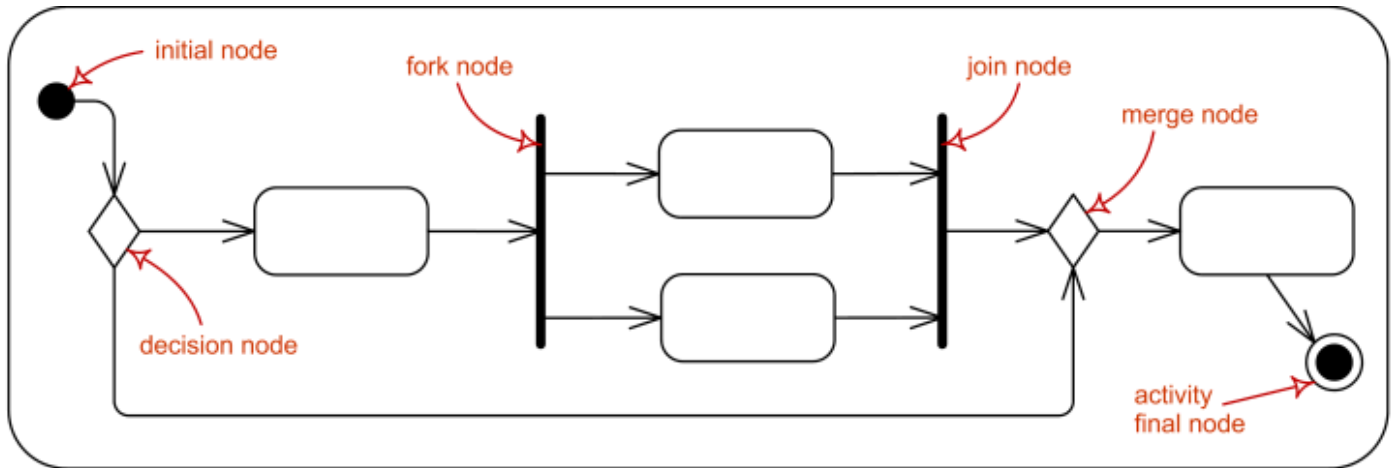
Role of Activity Diagrams in UML



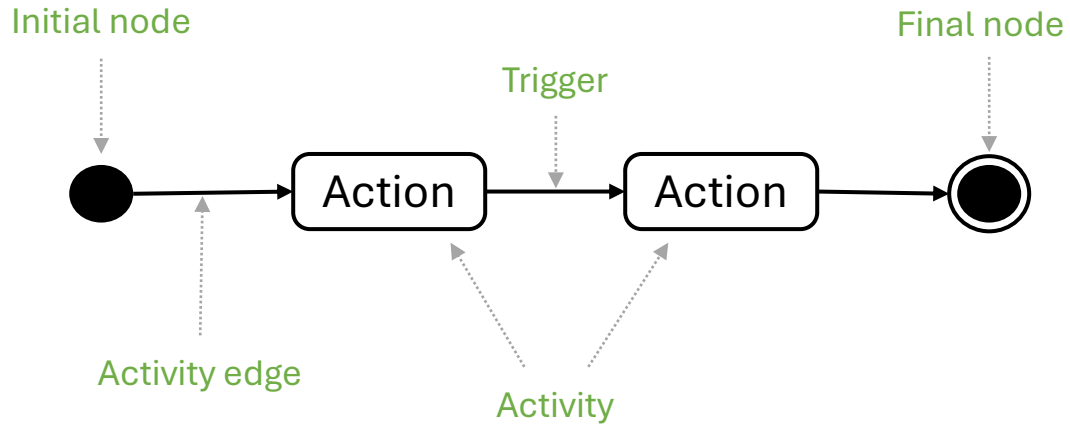
Why Activity Diagram?

- 💧 Activity diagrams describe the workflow of a system.
- 💧 Activity diagrams are useful for analysing a **use case** by describing **what actions** need to take place and **when** they should **occur**.
- 💧 Benefit on clarity, decision timing, and problem identification.

Activity Diagram at a Glance



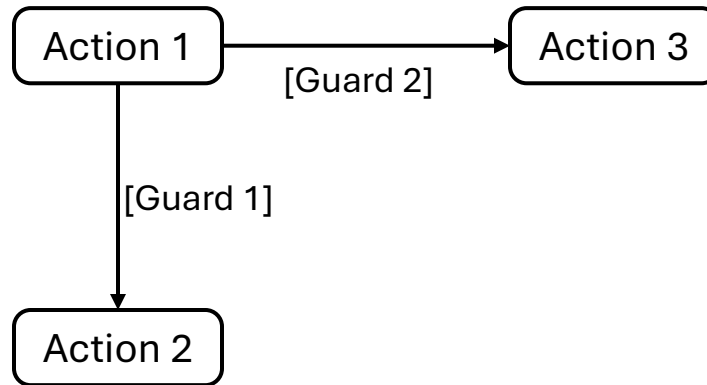
Structure of Activity Diagram



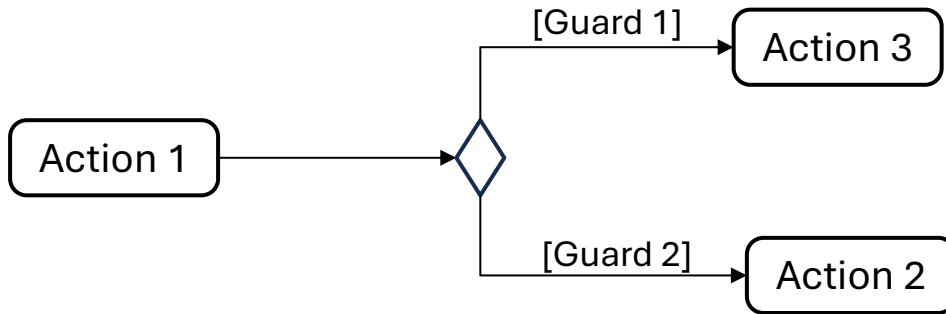
Decision Points

- ◆ A decision point shows where the exit transition from a state or activity may branch in alternative directions depending on a condition.
- ◆ A decision involves selecting one control-flow transition out of many control-flow transition based on a condition.
- ◆ Guard expression (inside **[]**) label the transitions coming out of a branch.

Guards

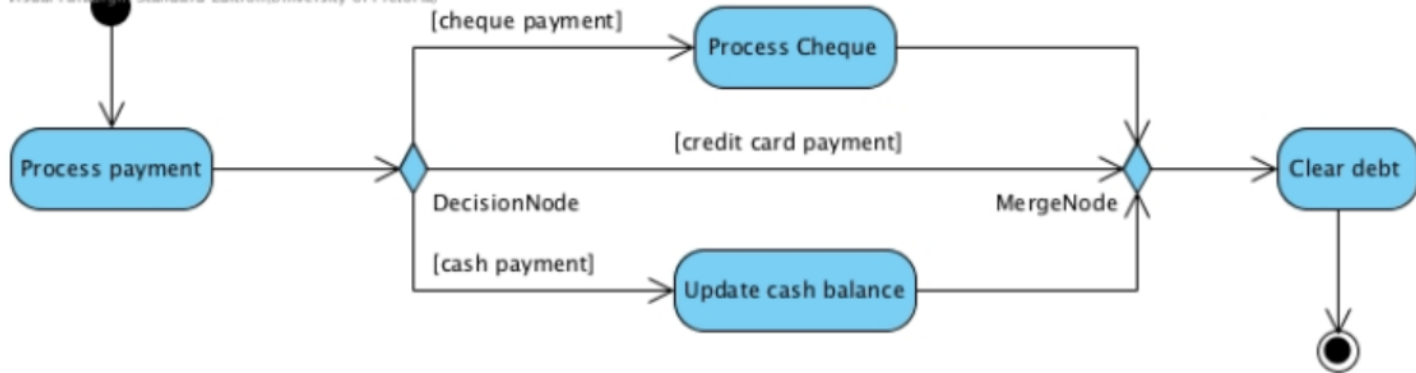


Guards



Decision vs Merge Node

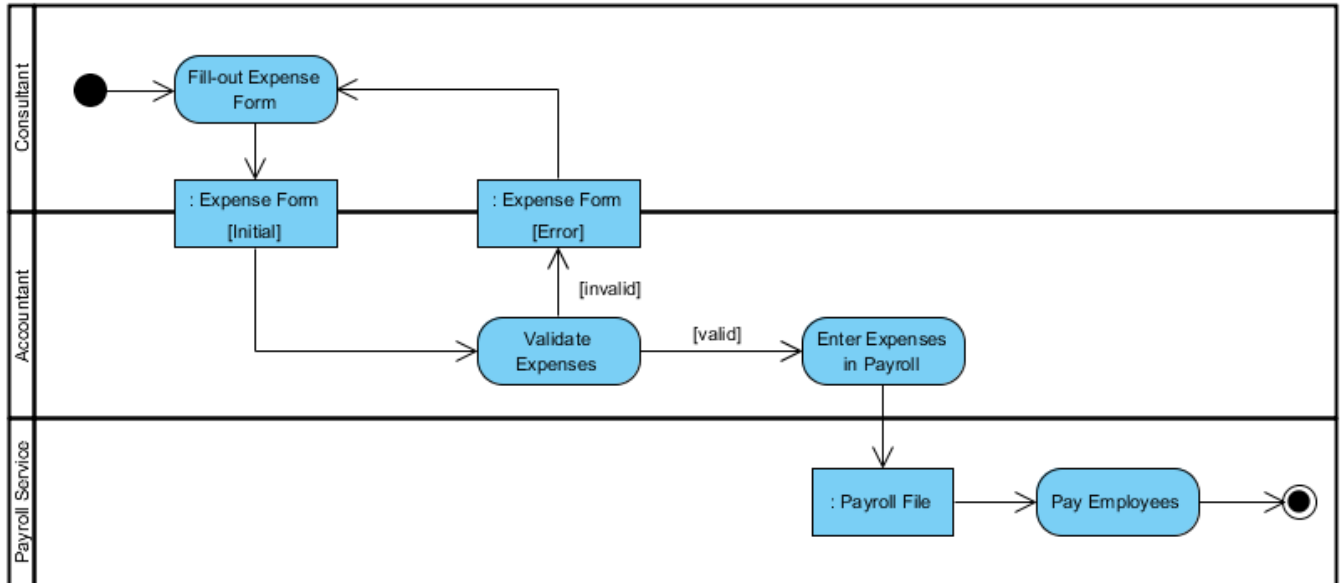
Visual Paradigm Standard Edition (University of Pretoria)



Swimlanes

- ♦ A swimlane is a way to group activities performed by the same actor on an activity diagram or activity diagram or to group activities in a single thread.
- ♦ Swimlanes are indicated by vertical dashed lines which separate the diagram into zones.
- ♦ Swimlanes allow the partition an activity diagram so that parts of it appear in a swimlane relevant to that element in the partition.

Swimlanes



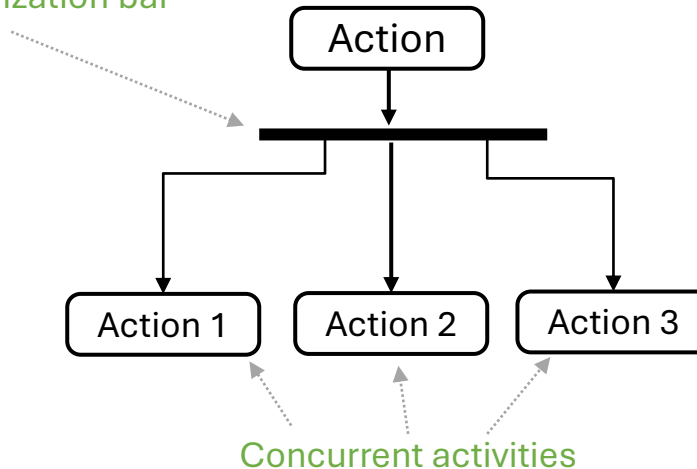
Concurrent Activities

- The difference between flowcharts and activity diagrams is that in activity diagram, parallel behavior can be expressed.
- This is important for business modeling, where unnecessary sequential process can be designed for parallel execution.
- Enabling to graphically lay out what threads you have and when they need to **synchronize**.

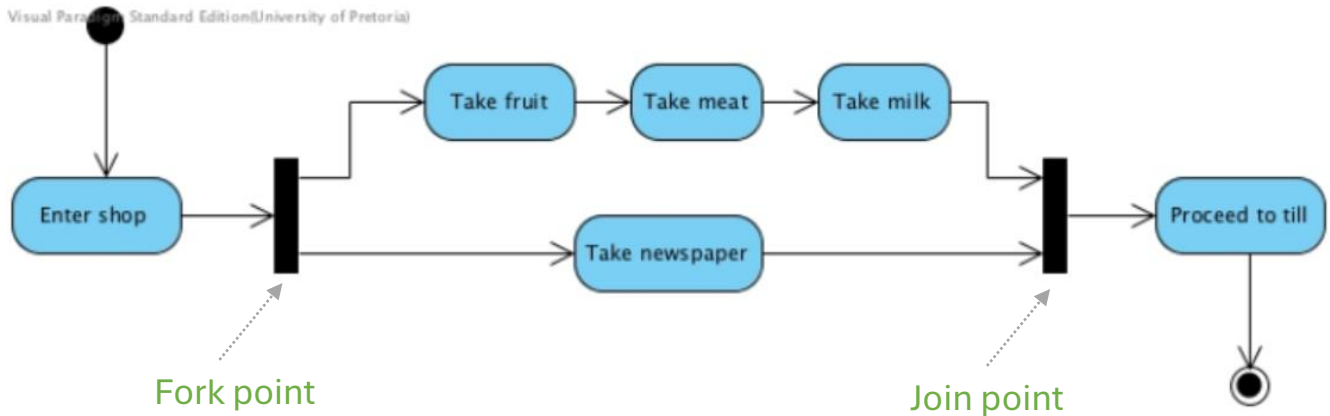
Synchronization Bars

- Synchronization bars initiate concurrent sections in AD. In these concurrent sections, triggers can occur in parallel and no sequential order is established.

Synchronization bar

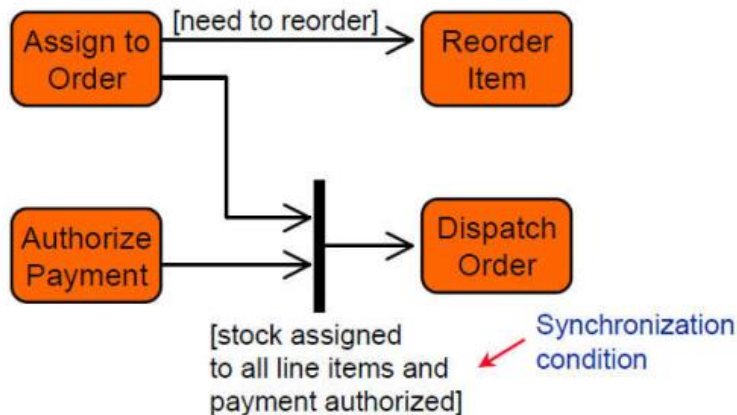


Concurrent Activities



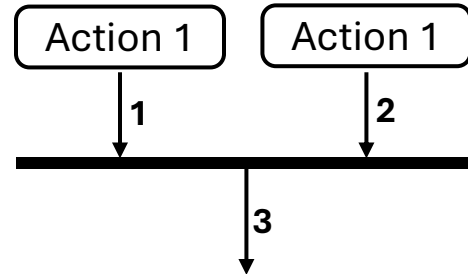
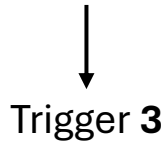
Synchronization Conditions

- The default behavior of synchronization bars is that the outbound trigger occurs as soon as all inbound triggers have occurred.
- In addition to this condition, you can specify an extra synchronization condition which is checked every time an inbound trigger occurs.

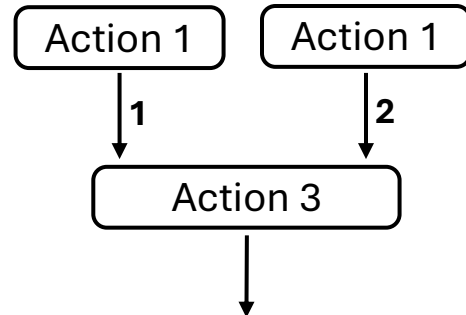
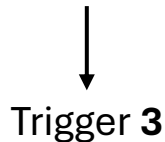


Multiple Upcoming Triggers

Trigger 1 **AND** Trigger 2



Trigger 1 **OR** Trigger 2



When to Use Activity Diagrams?

- ♦ Activity diagrams show behavior that spans over multiple use cases to describe the workflow of the overall process.
- ♦ For multiple objects and their high-level interaction, activity diagrams are particularly helpful for representing an overview of concurrent processes.
- ♦ Do not use activity diagrams to see how objects collaborate. An interaction diagram is simpler and gives you a clearer picture of collaborations.
- ♦ Activity diagrams are not accurate for describing how an object behaves over its lifetime. Use a state diagram instead.