Software Architecture

Lab 05

System Modeling

Néstor Cataño

Innopolis University

Spring 2016

Outline

1. **Activity Diagrams**
2. Sequence Diagrams

Based on Book UML Distilled, Third Edition by Martin Fowler

Activity Diagrams

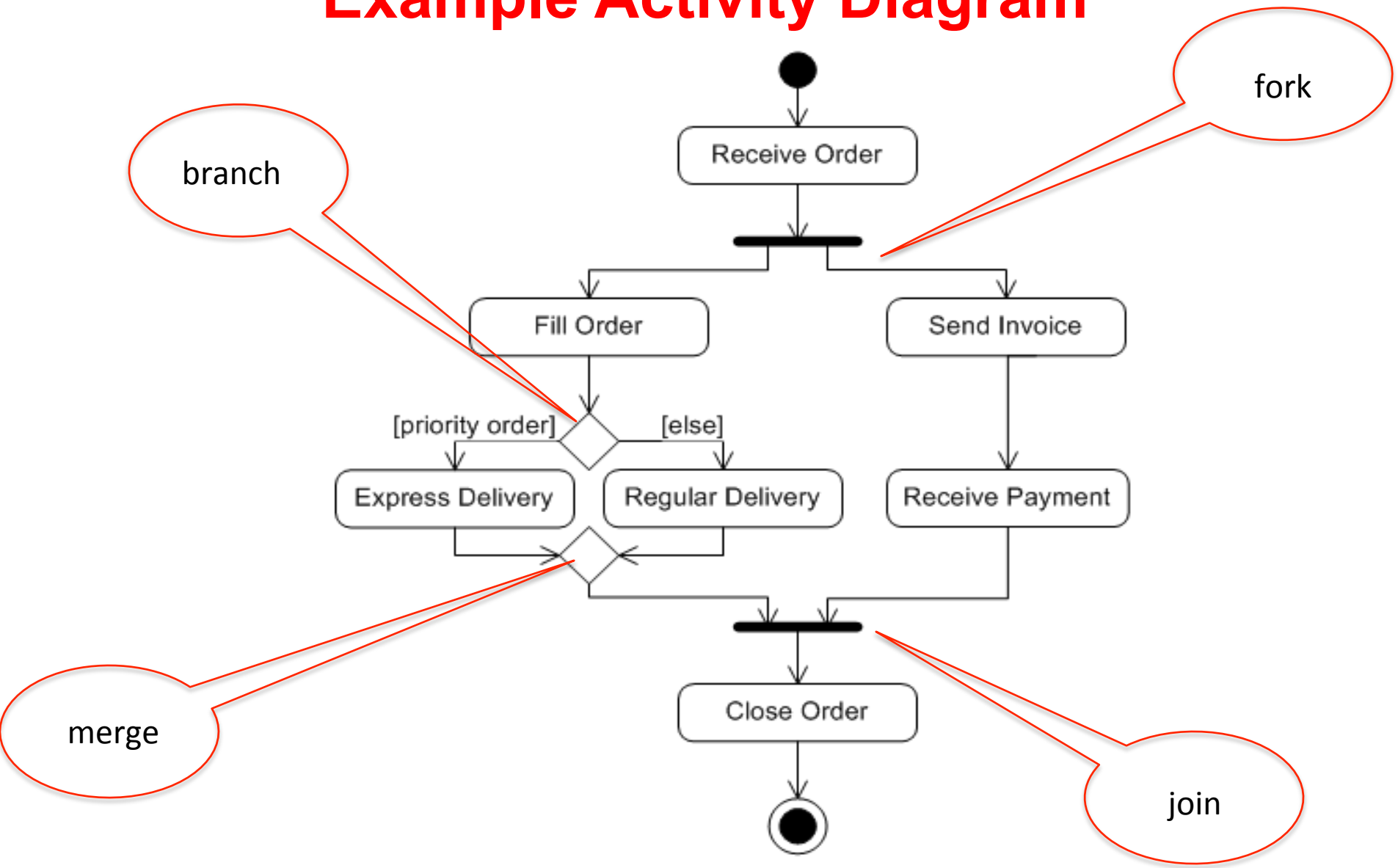
- Focus on the execution and flow of actions
  - > Events internal or external
  - > Based on Flowcharts and Petri nets
    - » Unlike Flowcharts, allows parallel actions
    - » Unlike Petri nets, allows control
- Can be applied to many kinds of behavioral modeling
  - > Business processes
  - > Software processes
  - > Workflows
  - > …

Activities

- Activities are high-level structures, composed of sequences of actions:
  - > "An activity is the specification of parameterized behavior as the coordinated sequencing of subordinate units whose individual elements are actions."

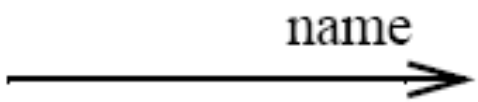
- In UML 1.x, actions were called activities

Example Activity Diagram



Activity Diagram Basic Notation

Action node

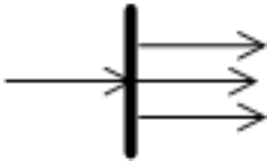


Activity edge with name

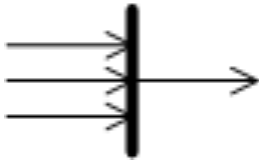
Control nodes

Fork and Join Nodes

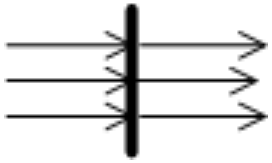
- **Used to model parallelism**



Fork node



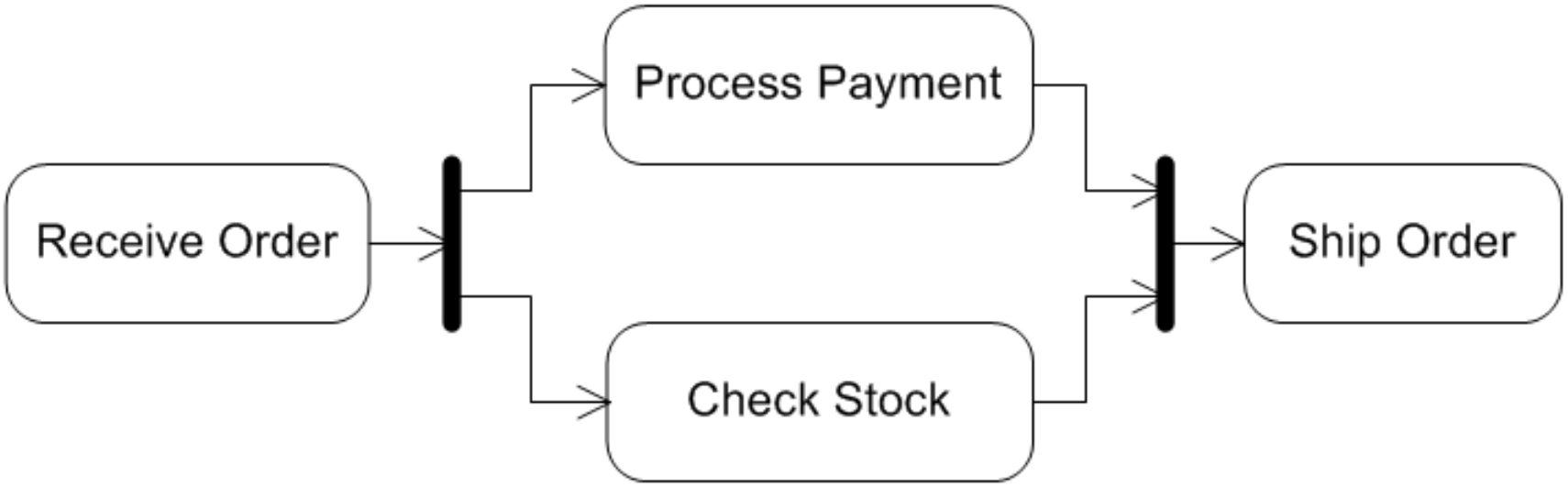
Join node



Fork & Join node

- **Join waits for all incoming flows**

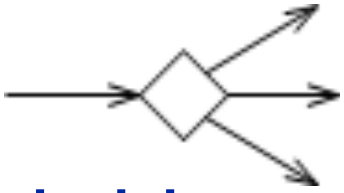
Example of Fork and Join



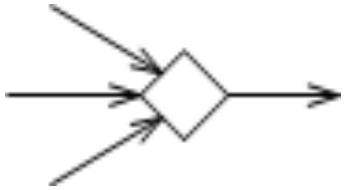
Decision and Merge Nodes

- A decision node represents conditional behavior
  - > Each outbound flow has a guard; written in brackets
  - > Guards are mutually exclusive

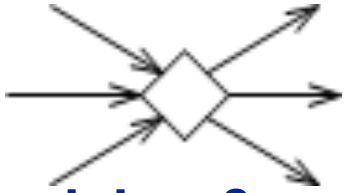
- A merge node brings together multiple flows.
  - > Not used to synchronize concurrent flows;
  - > But to accept one among several alternate flows.



decision



merge



decision & merge

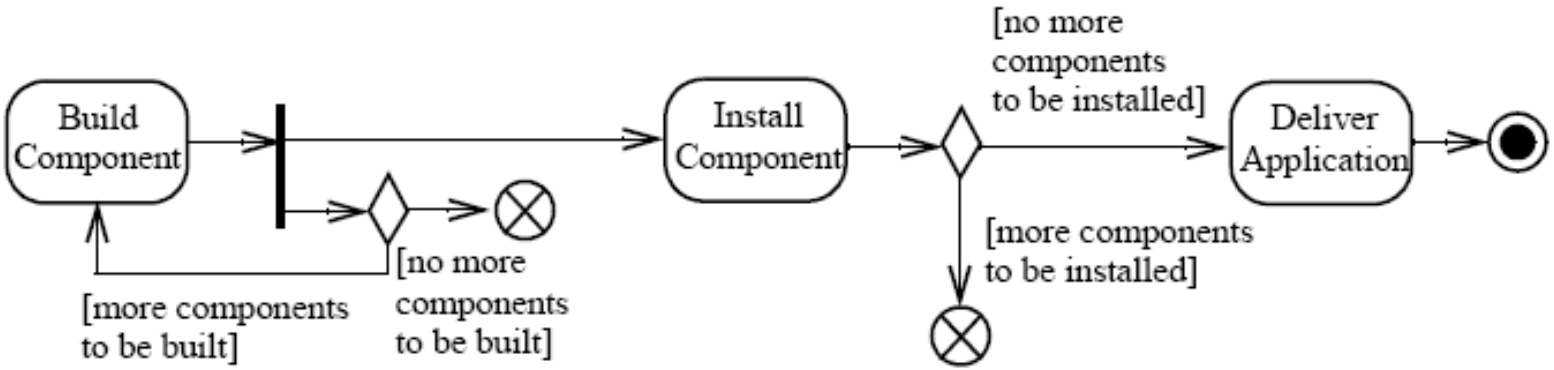
Flow and Activity Final Nodes

- A **flow final node** indicates that a single flow is complete.
  - > Other flows may still proceed.

- An **activity final node** indicates that the activity has been completed.
  - > Cancels all other outstanding ongoing flows

- Activity final node and flow final node do not have outgoing edges because they are termination points.

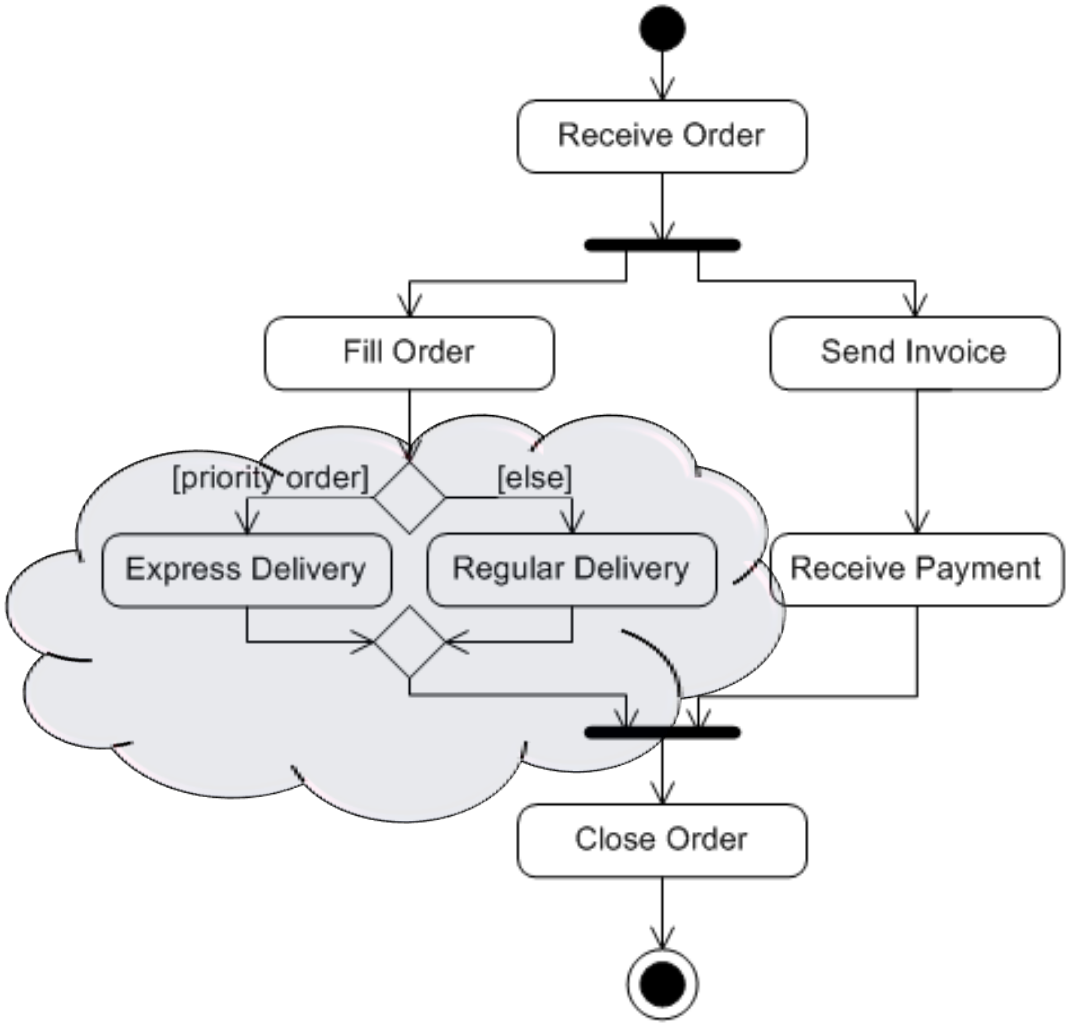
Example of Flow and Activity Final Nodes



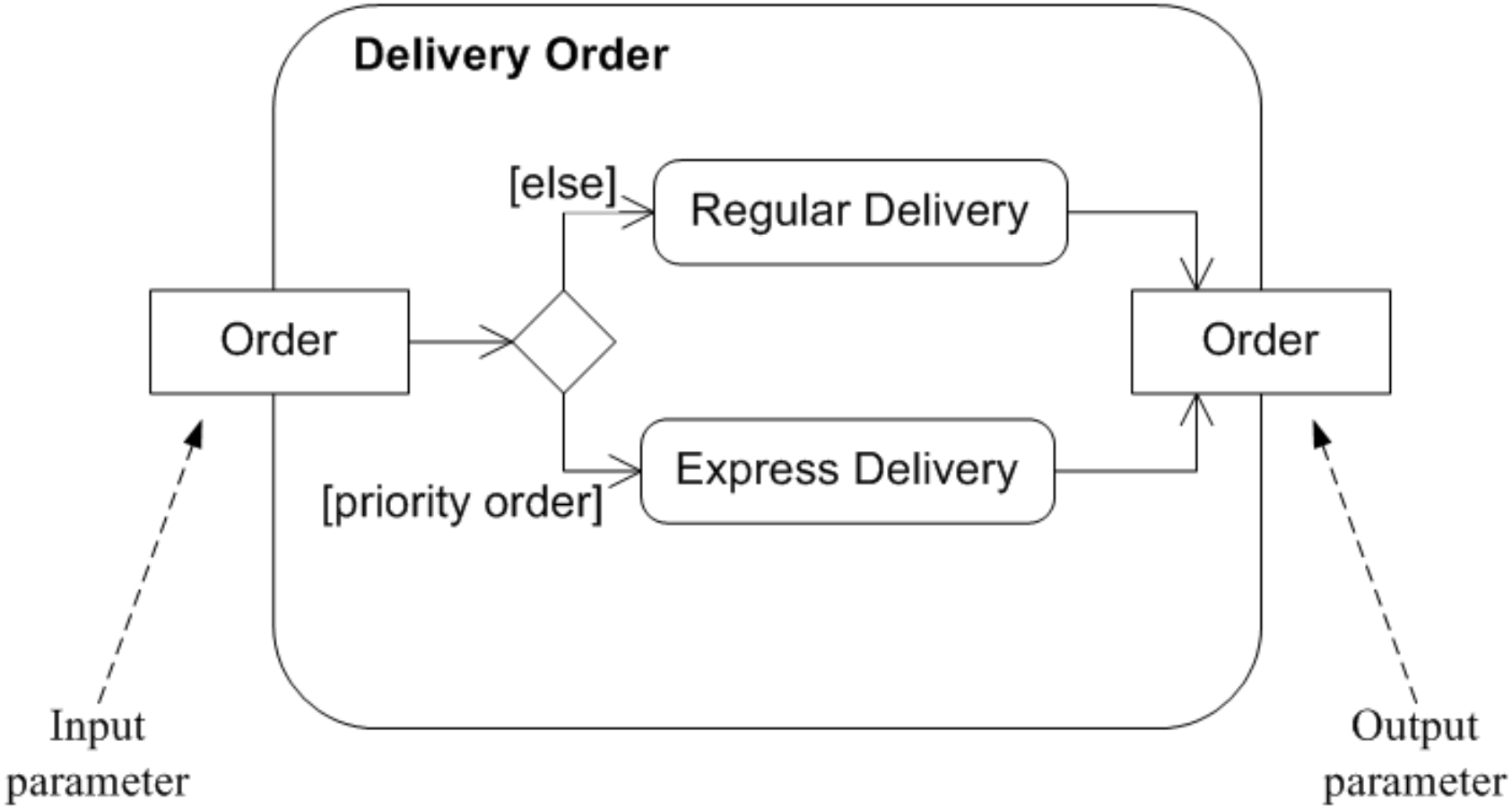
Grouping

- Activity diagrams tell you what happens but they don't tell you who does what
  - Solution → **Partitions**
- There are three ways in which activities can be **grouped**
  - > In **sub-activities**
  - > Using **swimlanes**
  - > In **expansion regions** (not discussed here)

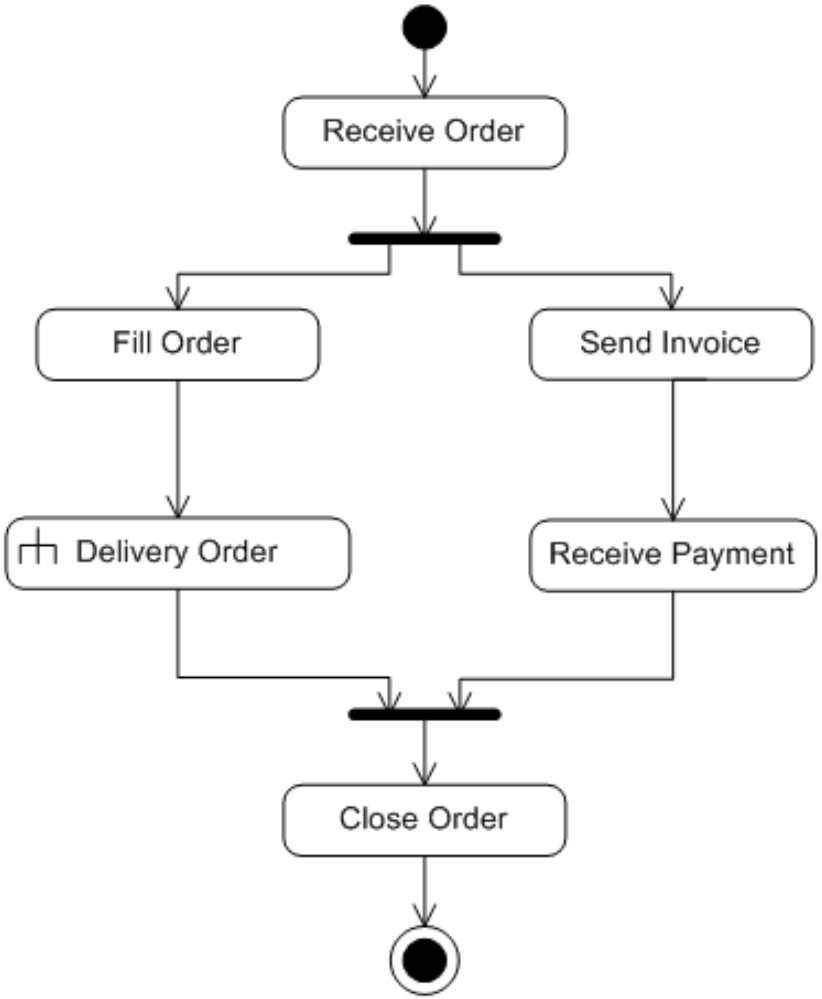
Subactivities



Subactivities



Calling the Subactivity

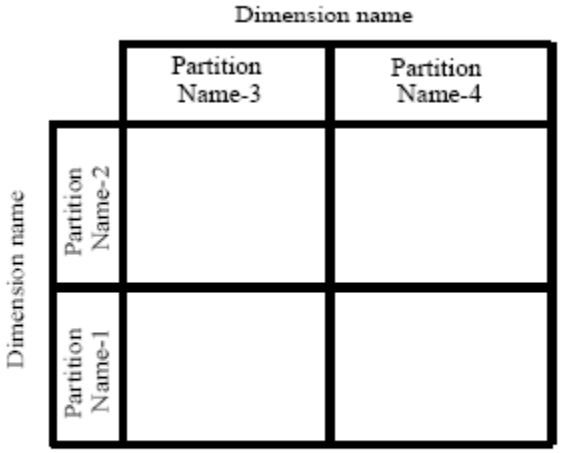


Rake indicates Sub-activity diagram

Swimlanes

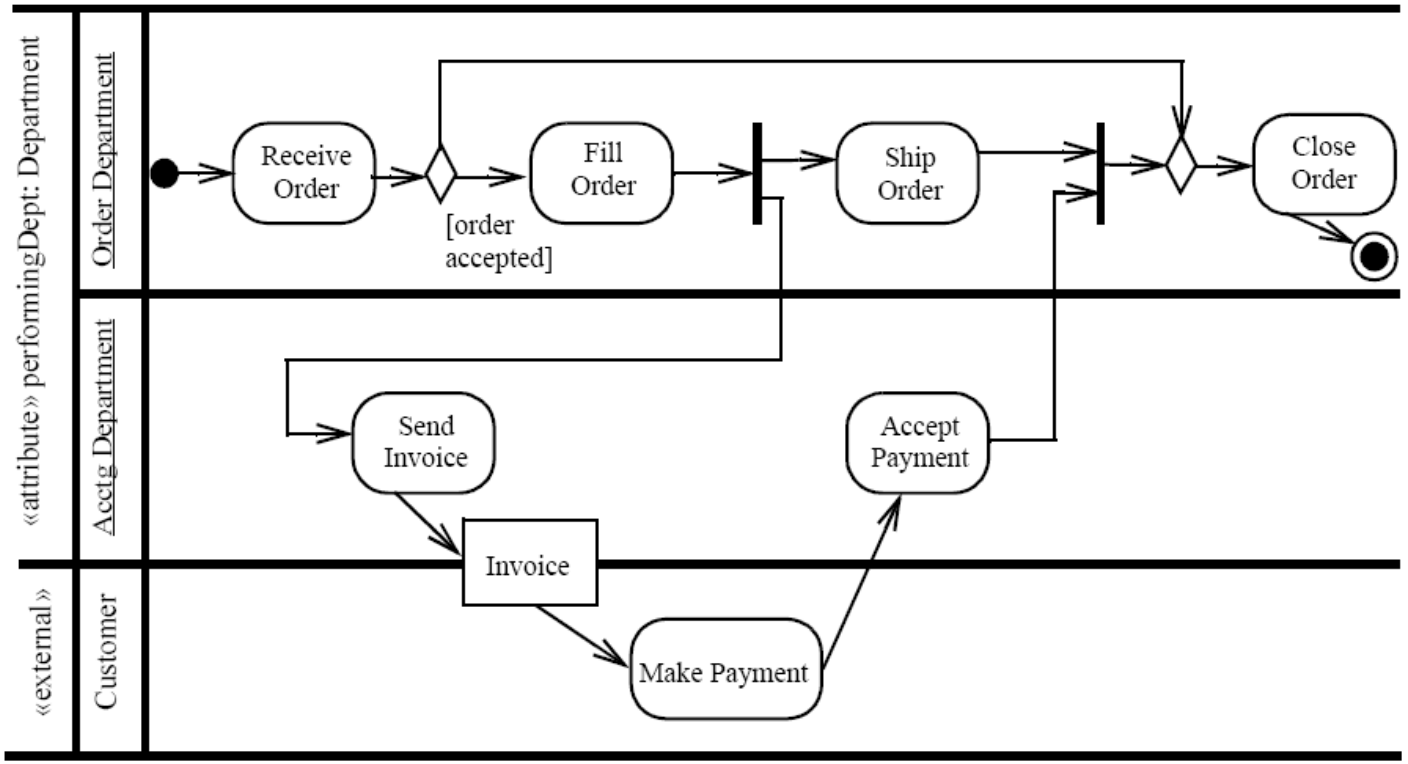
a) Partition using a swimlane notation

b) Partition using a hierarchical swimlane notation



c) Partition using a multidimensional hierarchical swimlane notation

Swimlanes Example – 1



Outline

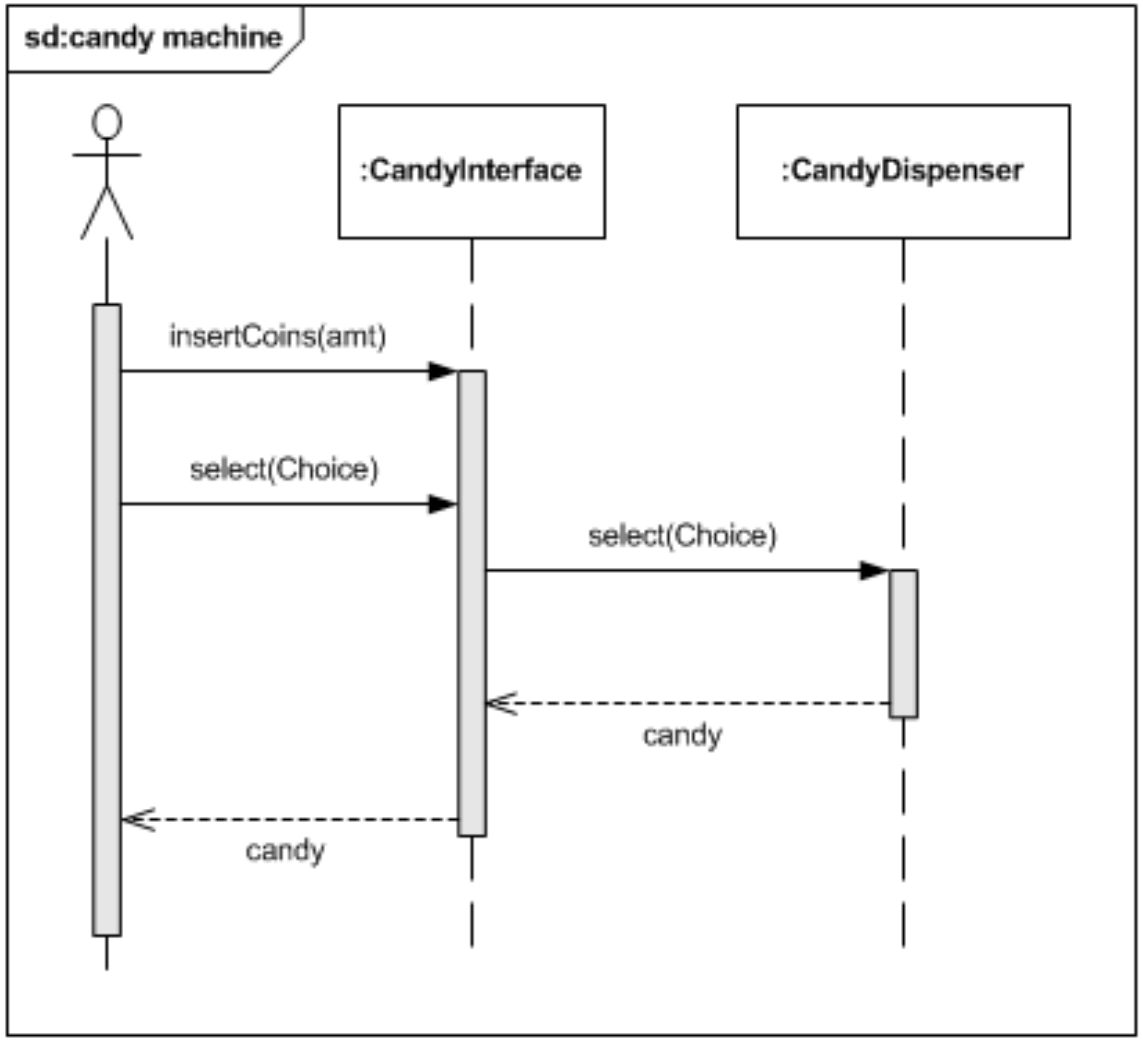
1. Activity Diagrams
2. **Sequence Diagrams**

Sequence Diagrams

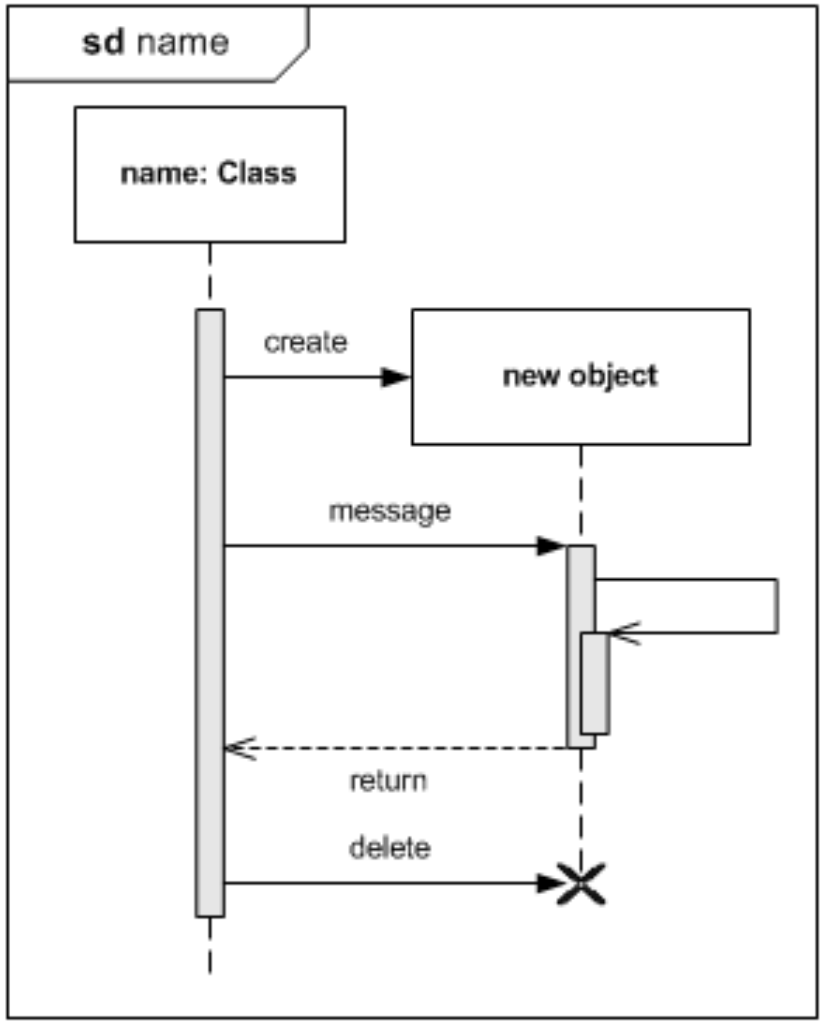
- Used to show:
  - message flows between objects
  - ordering relationships in message sequences
  - lifetime of objects, and threads

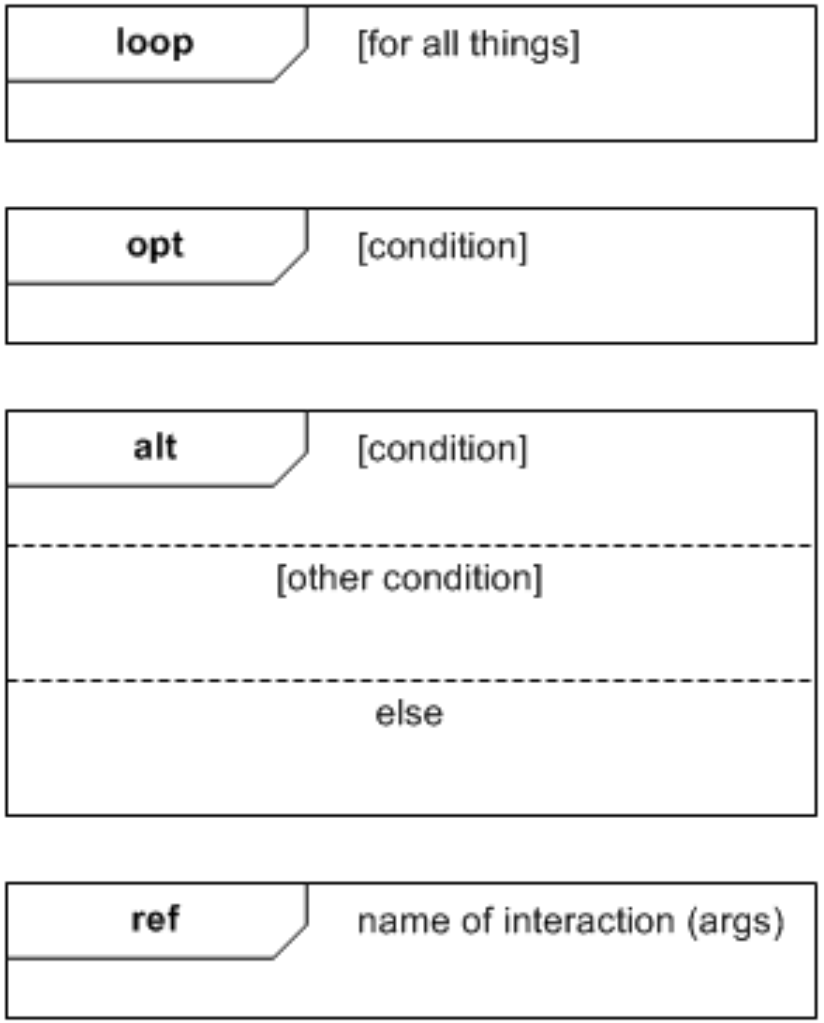
- Can be broken down into “interaction fragments” which may themselves be represented in the same or a separate diagram

Sequence Diagram Example



Sequence Diagrams: Basic Notation



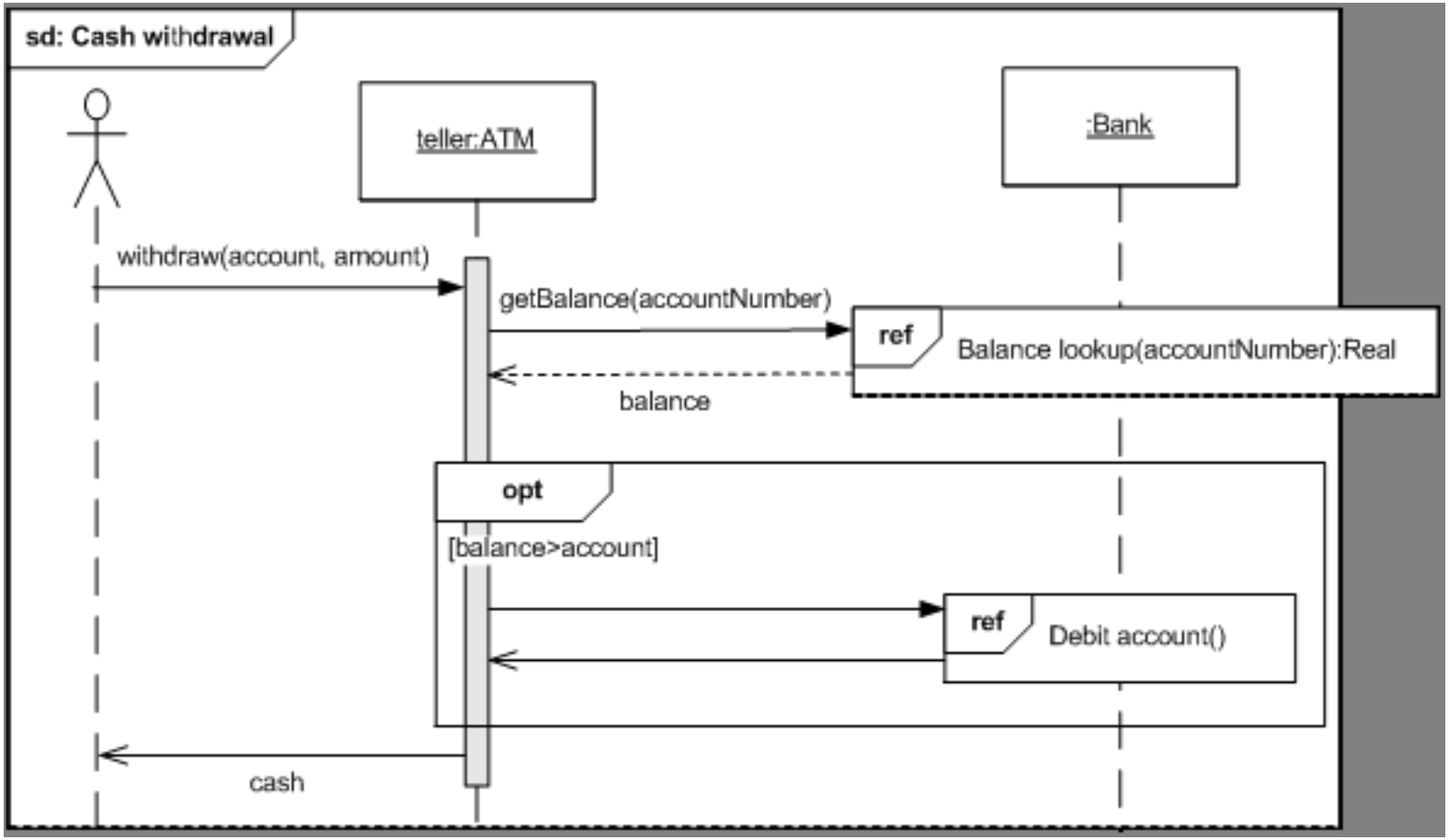


Sequence Diagrams: Basic Notation

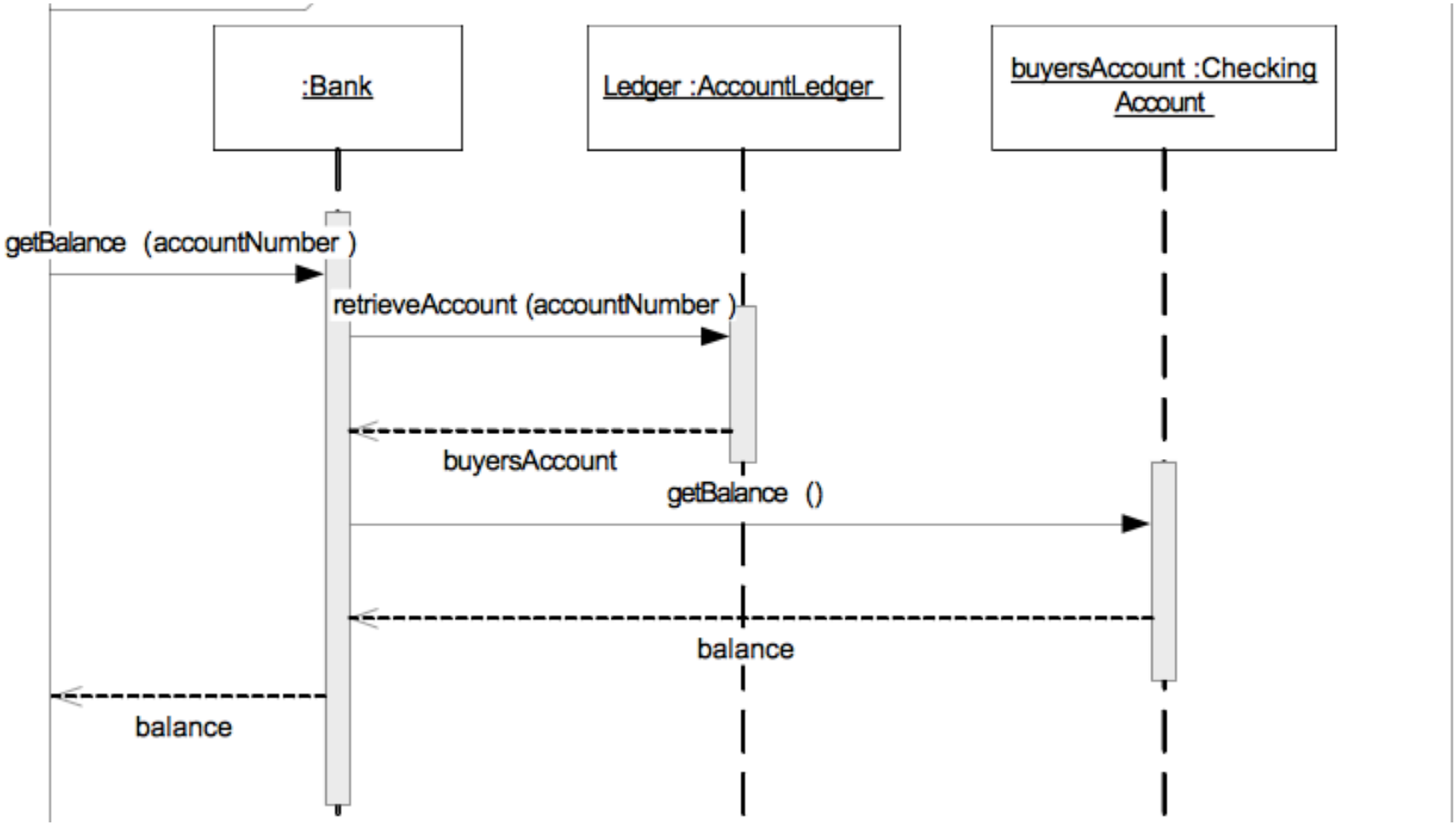
- **Diagrams can be "nested" using operators:**
  - »sd – named sequence diagram
  - »loop – repeat interaction fragment
  - »opt – optional "exemplar"
  - »ref – reference to "interaction fragment"
  - »alt – selection

  - »par – concurrent (parallel) regions
  - »seq – partial ordering (default) (aka "weak")
  - »strict – strict ordering
  - »assert – required (i.e. causal)
  - »neg – "can't happen" or a negative specification
- **The message entry/exit points are called "gates"**
- **Gates allow tools to ensure that the diagrams are compatible and consistent with each other.**

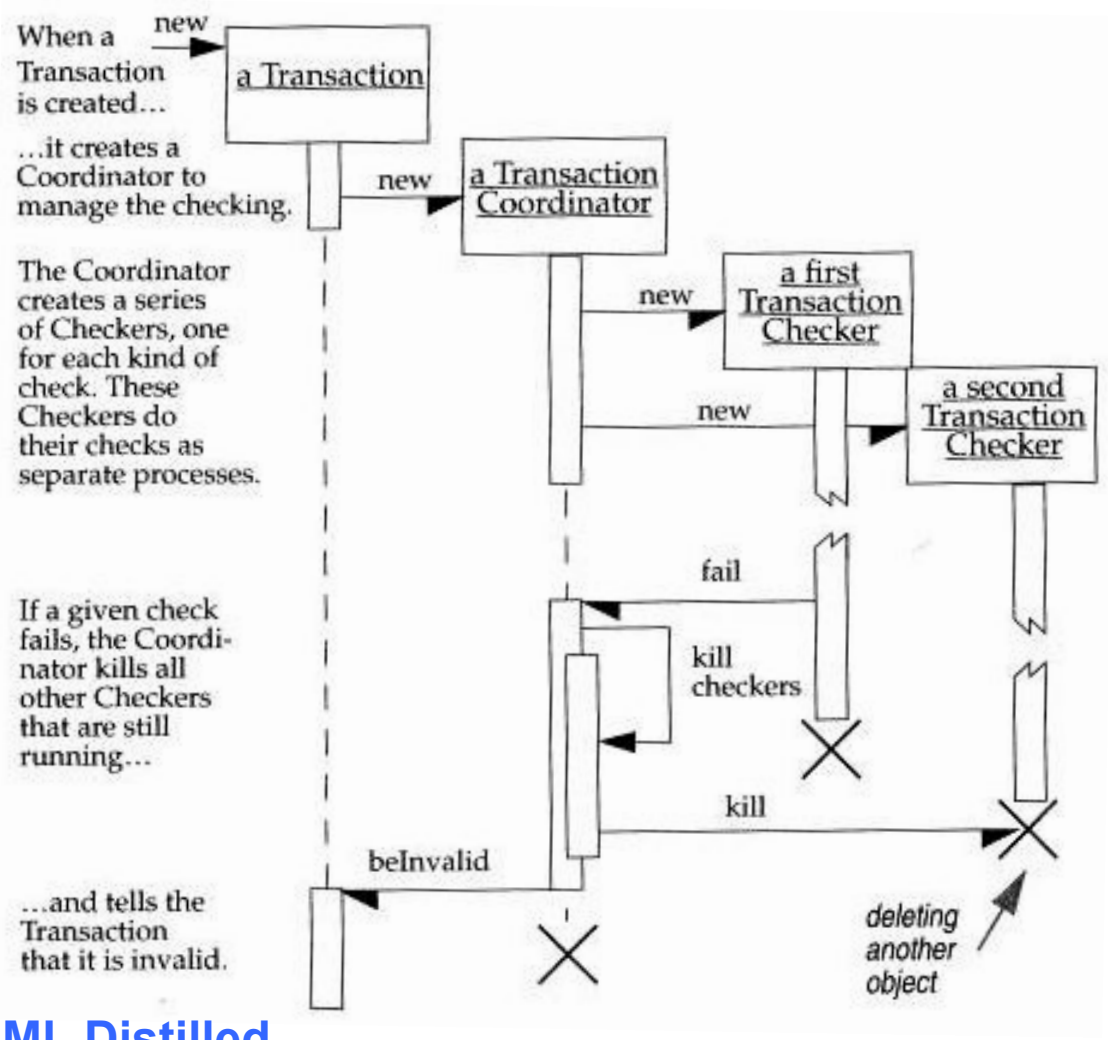
A More Complex Example



A More Complex Example (2)



Asynchronous calls, Self-delete, Notes



From Book UML Distilled