Activity Diagrams

Introduction(I)

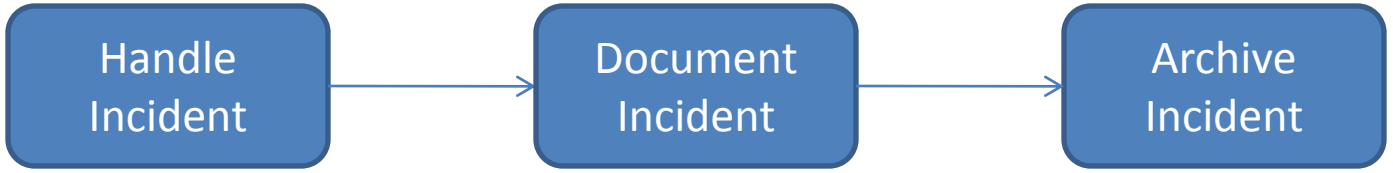
- Activity diagrams describe procedural logic, business process, and workflow.
- Activity diagrams focus on the action sequence of execution and the conditions that trigger or guard those actions.
- Activity diagrams focus on the activity's internal actions, not on the external interfaces

Introduction(II)

- Activity diagrams have similarities to flowcharts
  - But flowcharts notation does not support parallel behavior.
  - Business managers may prefer activity diagrams over flowcharts, because they are more understandable for non-technical people.
- An activity diagram is a special case of state chart diagram in which states are actions.

Introduction(III)

- An activity diagram shows flow control within a system.



Activity Diagram Elements

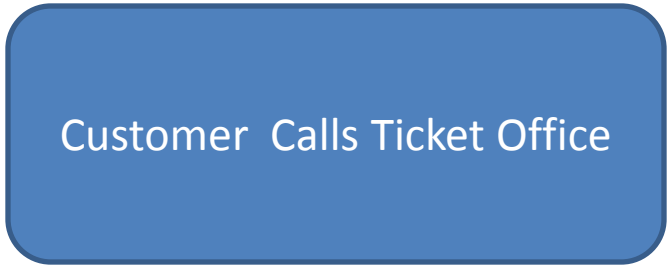
- Initial node
- Activity final node
- Action
- Flow/edge
- Fork
- Join
- Decision
- Merge
- Synch

Basic Elements—Action(I)

- Action in Activity Diagram Elements' official UML name is **action state**.
- Distinction between **action** and **activity**
  - Action state refers to it as **action**
  - Use term **activity** only refer to the whole task being modeled by the activity diagram

Basic Elements—Action(II)

- The rounded rectangle represents an action that occurs.
- E.g., Customer calls ticket office :

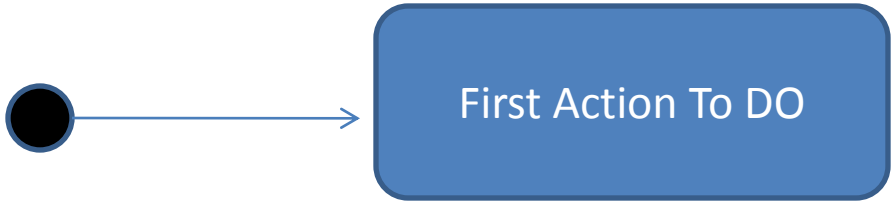


A sample action that is part of an activity diagram

Basic Elements--Initial state

- The filled circle is the staring point of the diagrams.

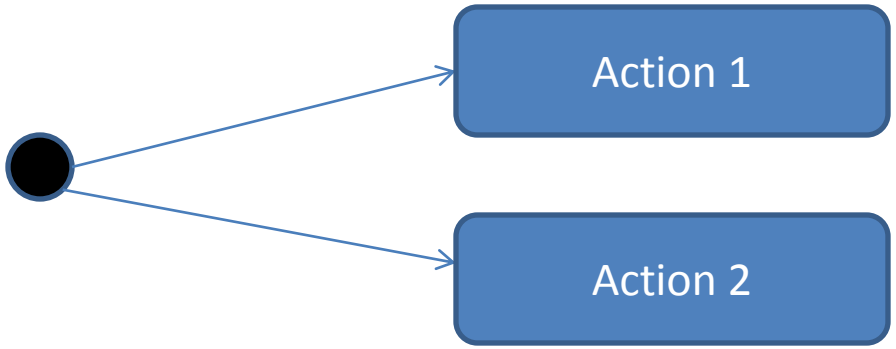
  An initial node isn't required.



The initial state shows the starting point for the action sequence within an activity diagram.

Basic Elements--Initial state(II)

- Initial state can indicate only **ONE** action.



Incorrect rendering of an initial state within an activity diagram. The initial state can indicate only ONE action

Basic Elements—Flow/edge

- The arrow on the diagram. There is a subtle difference between flows and edges.

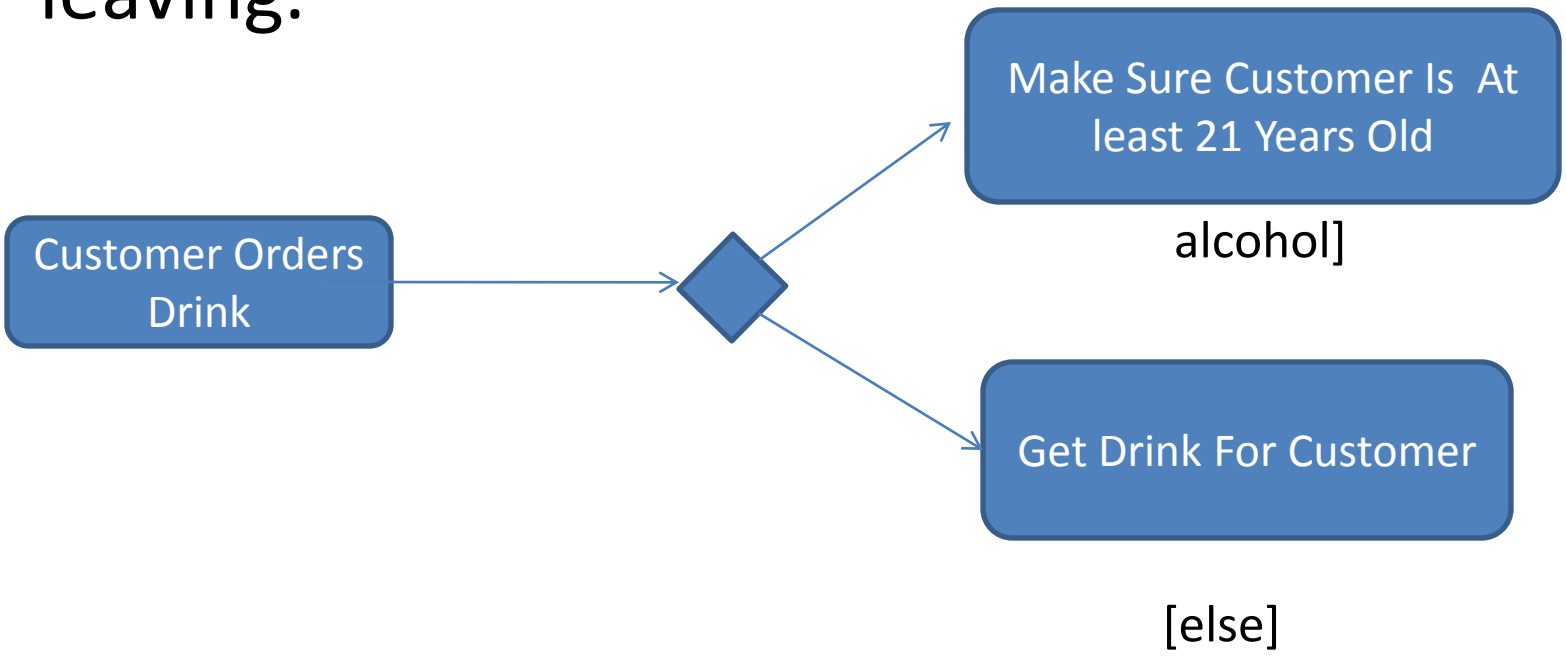
Basic Elements—Final node

- The filled circle with a border is the ending point.

  An activity diagram can have zero or more activity final nodes

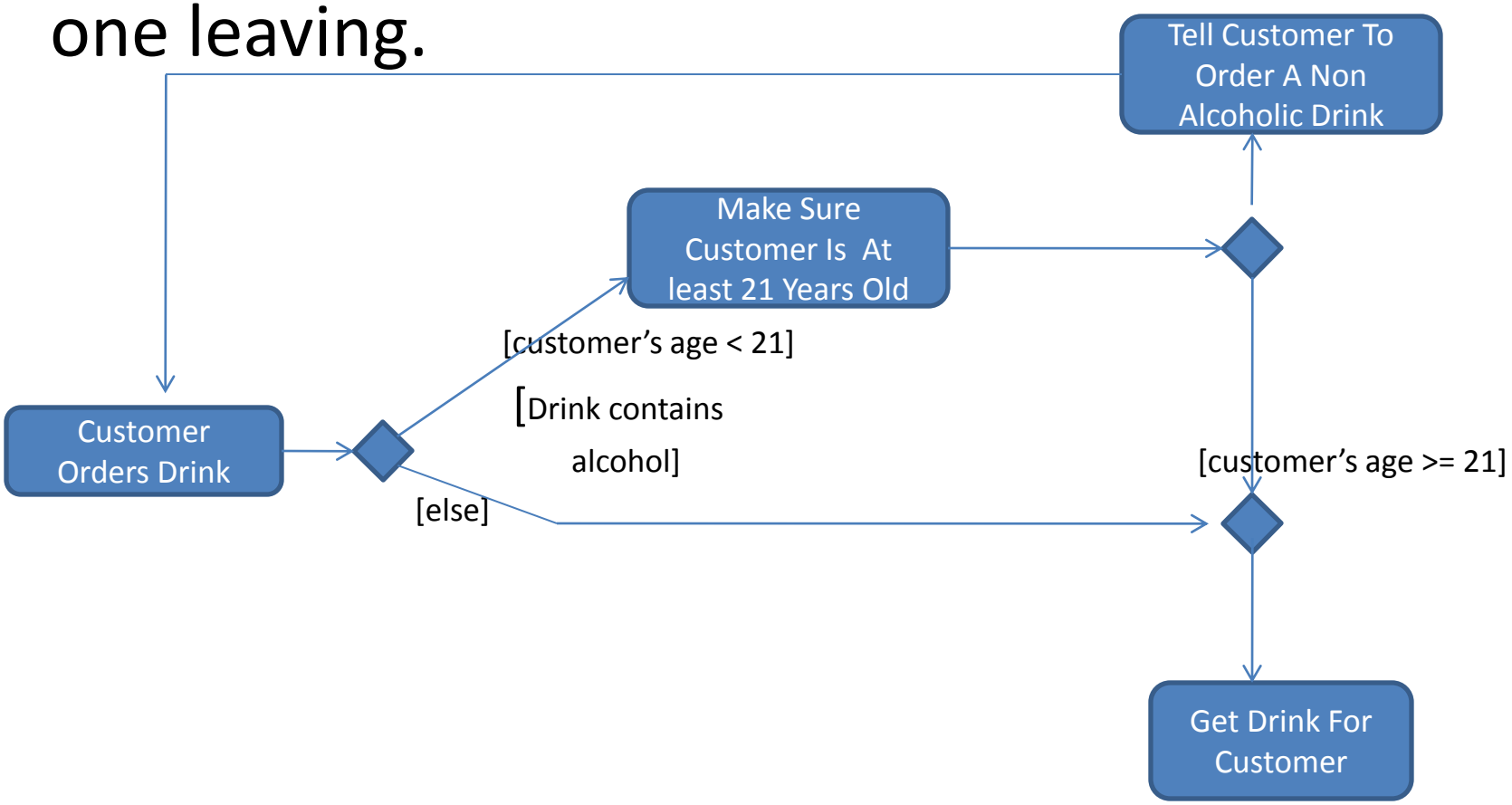
Decision

- A diamond with one flow entering and several leaving.



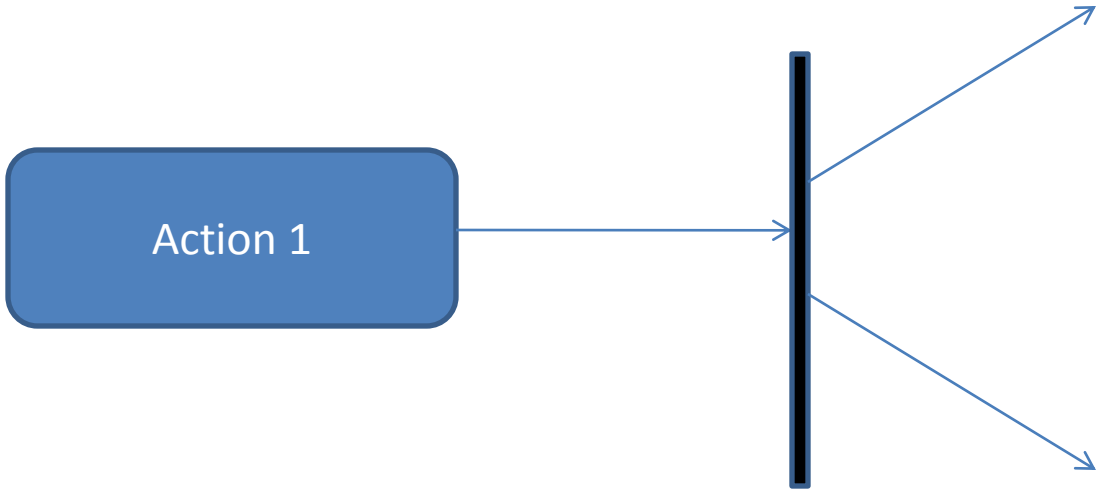
Merge

- A diamond with several flows entering and one leaving.



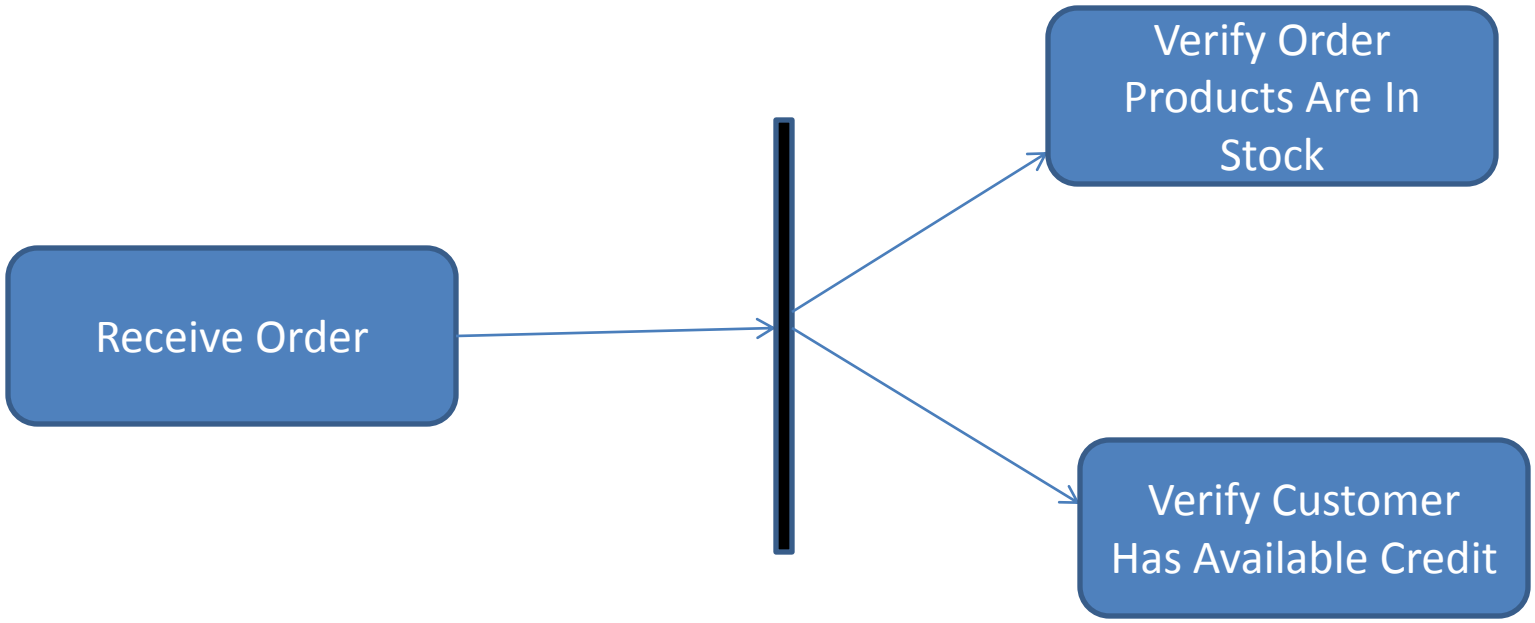
Synch

- A thick, solid line, allowing two or more action sequences to proceed in parallel



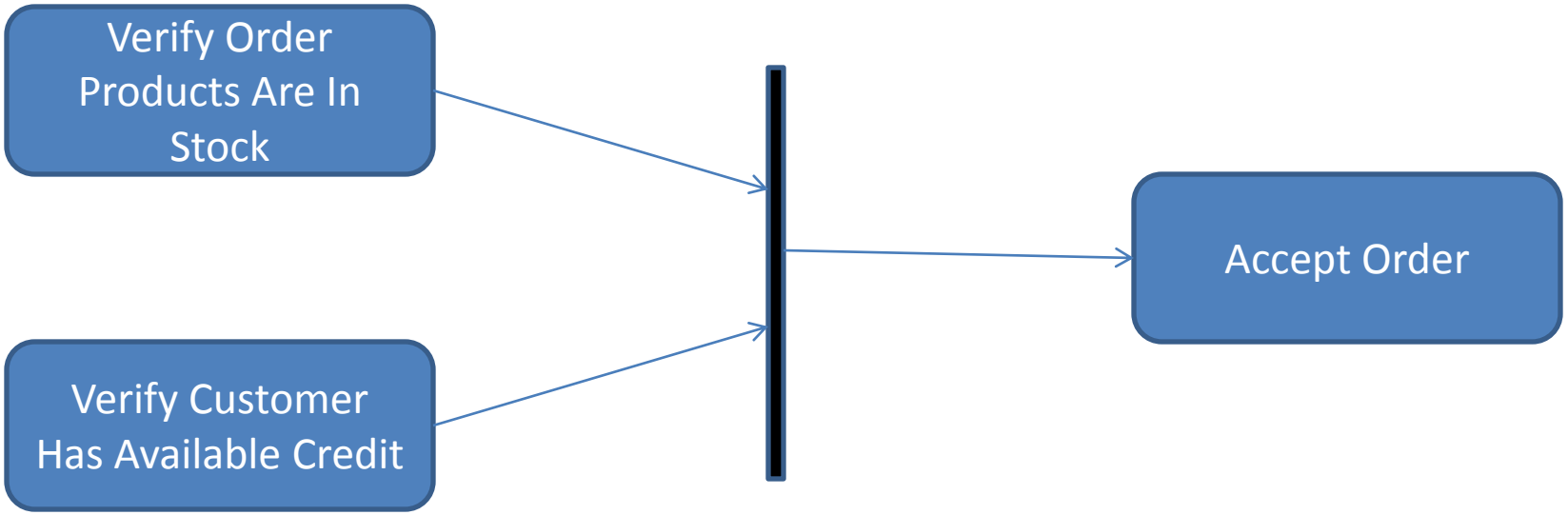
Fork

- Synch with one flow going into it and several leaving it.
- Denotes the beginning of parallel actions.



Join

- Synch with several flows entering and one leaving.
- All incoming flows must reach it before processing may continue. This denotes the end of parallel processing.

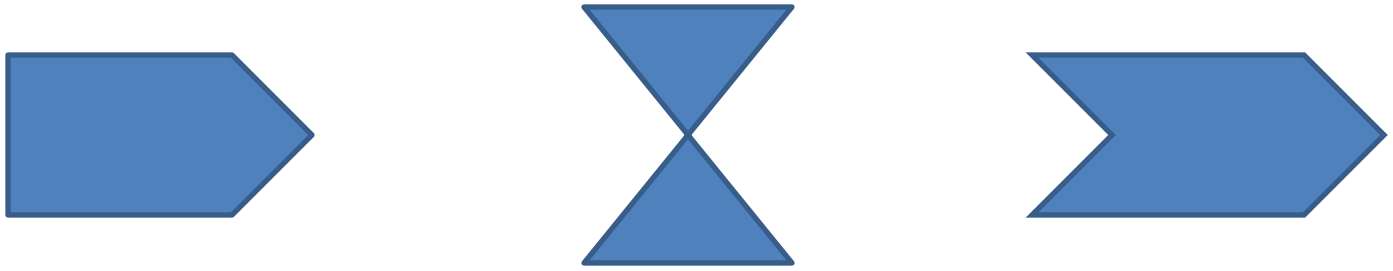


Signals

- An Activity diagram can have a clearly defined start point, which corresponds to an invocation of a program or routine.
- Activity diagram can also show response to signals.
- A time signal occurs because of the passage of time (for example, each month end might trigger a signal.)
- A real time signal indicates that the activity receives an event from an outside process.
- The activity listens for those signals, and the diagram defines how the activity reacts.

Signals

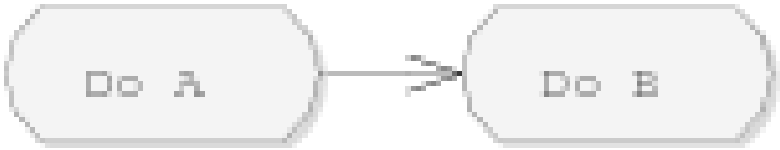
- Activity diagrams can show signals sent or received
- For example, we can send a message and then wait for a reply before we can continue.
- Basically, the signals are flow triggers.



Send signal Time signal Accept signal

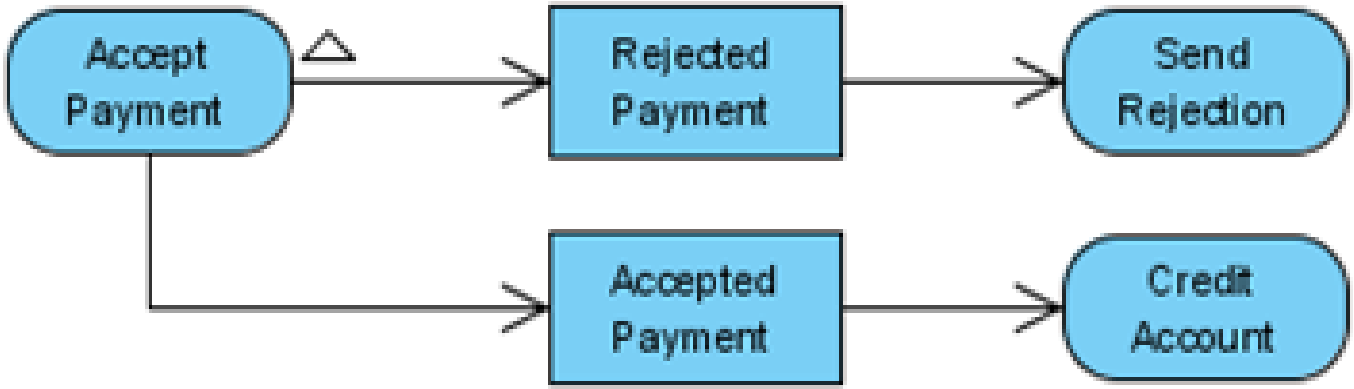
Flow

- **Connection between 2 actions**
- **Simple flow**



  - arrow
  - from a node to another

- **Flow with Exception**

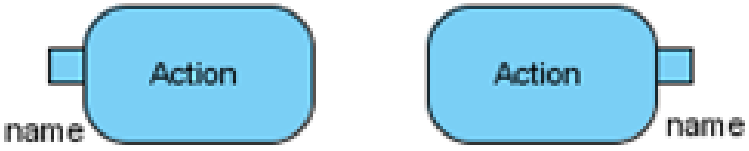


Flows (cont.)

- **Flow with objects**

- **Flow with pins**

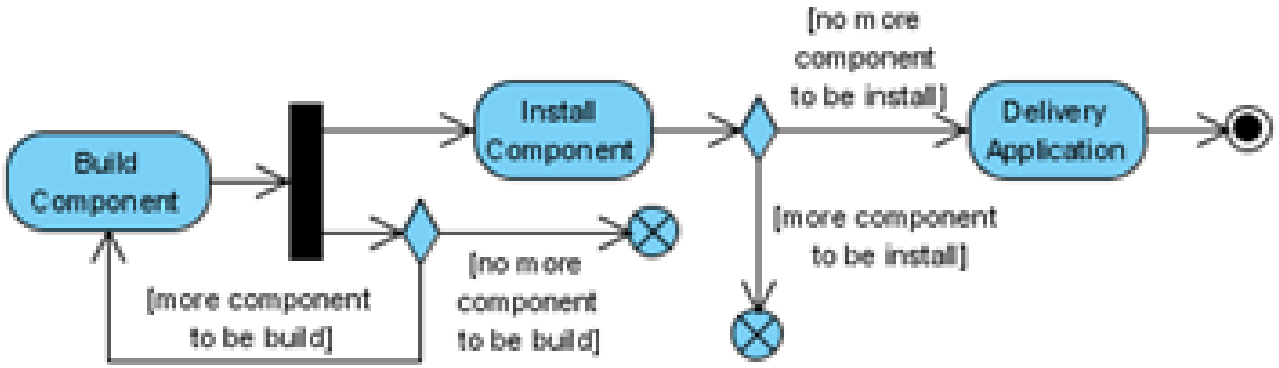
  - similar to flows with objects
  - data needed and data produced



Flows (cont.)

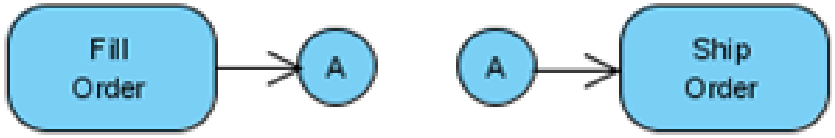
- **Decision flows**

-labeled



- **Connectors**

- does the same job as a simple arrow



Tokens

Tokens flow through the diagrams:

- The initial node creates a token, executes, passes the token to the next
- Fork produces a token on each of its outward flows.
- On a join, as each inbound token arrives, nothing happens until all the tokens appear at the join; then a token is produced on the outward flow.

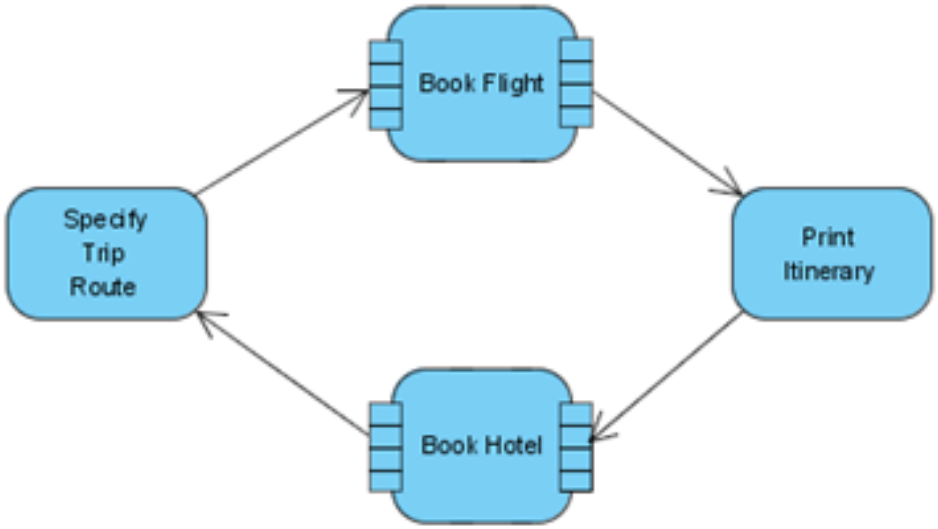
Join Specification

innovate achieve lead

- **Boolean expression** using the names of the incoming edges to specify the conditions under which the join will emit a token.
- **Evaluated** whenever a new token is offered on any incoming edge**.**
- **Default** - "and”

Expansion Region

- Structured activity region that **executes multiple times** corresponding to elements of an input collection.
- Example: The hotels may be booked independently and concurrently with each other and with booking the flight.



Advanced Notation

- Conditional threads
- Nested activity diagrams
- Partitions

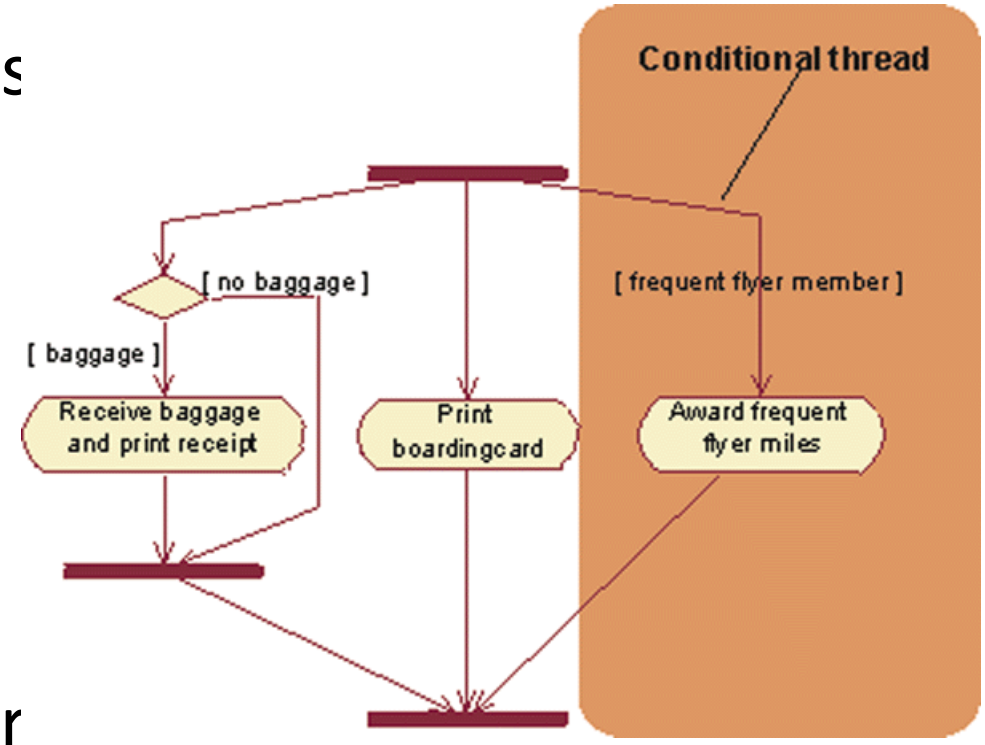
Conditional threads

- One of a set of concurrent threads is **conditional**.

- **Example**:

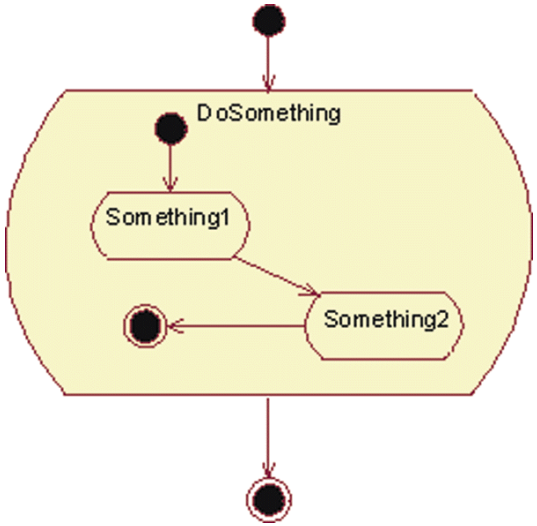
  Frequent-flyer member?

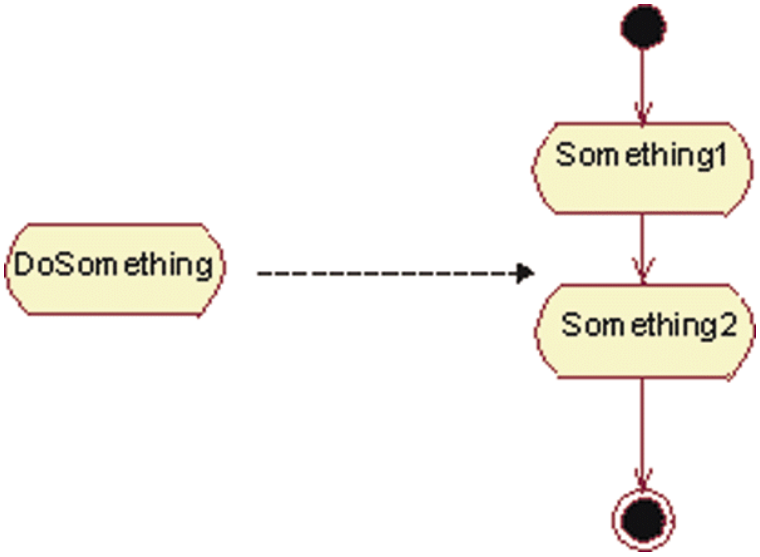
  Award the passenger frequent flyer miles.



Nested Activity Diagram

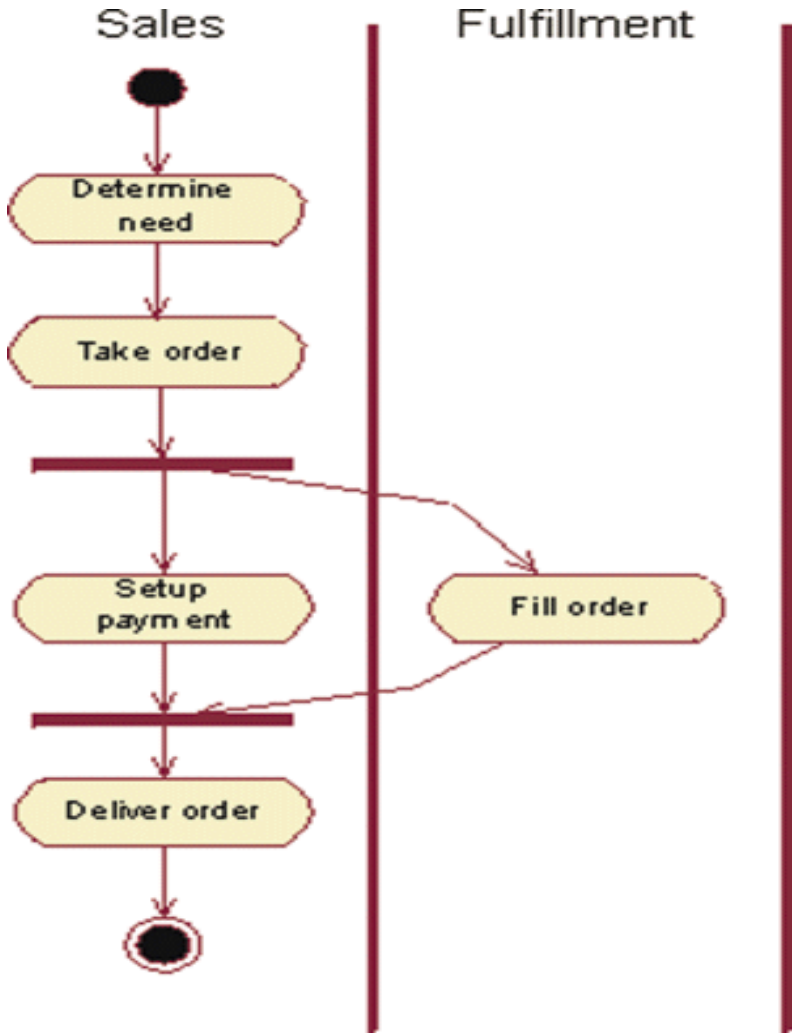
- diagram refers to an external one that uses more abstraction

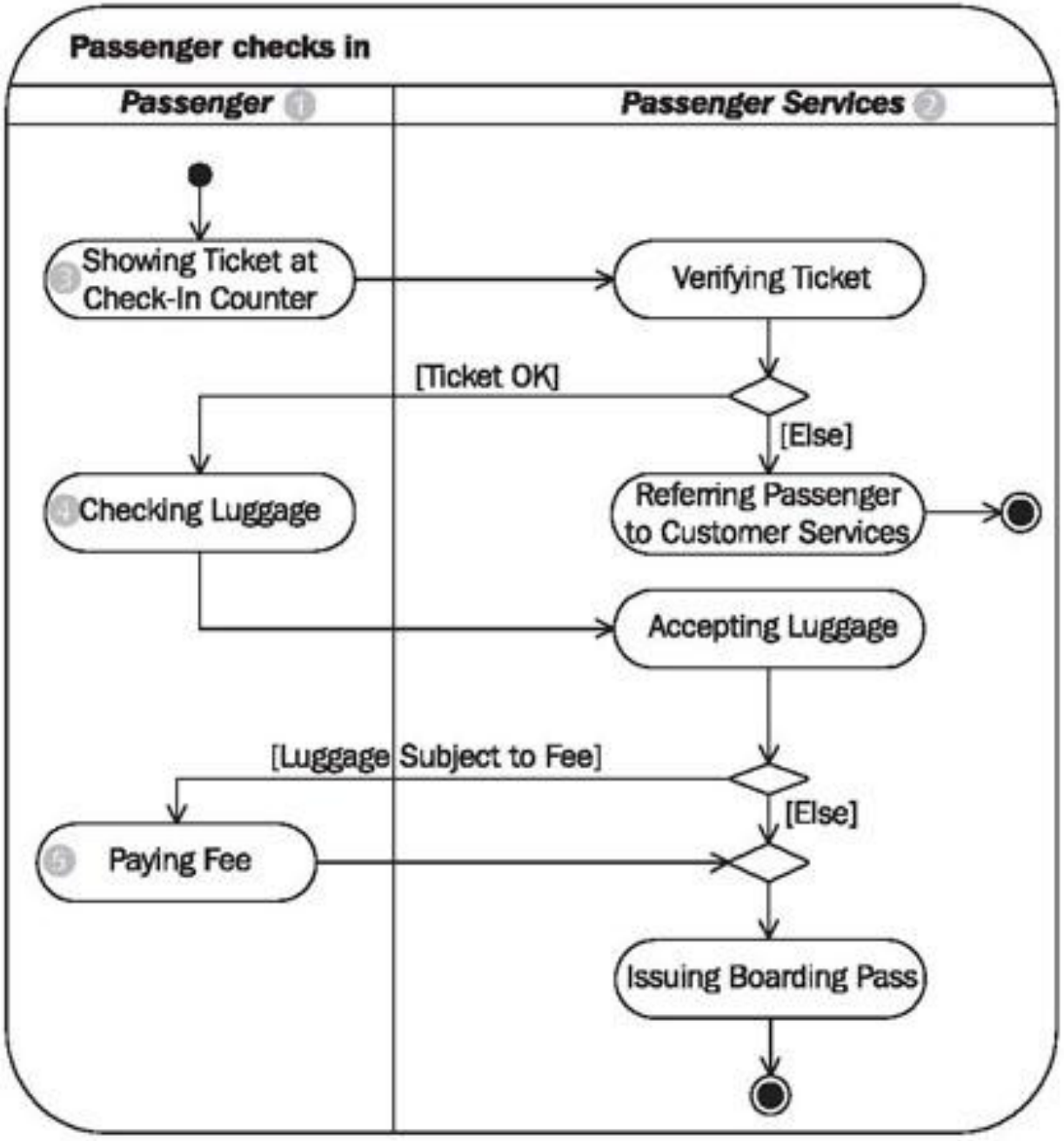




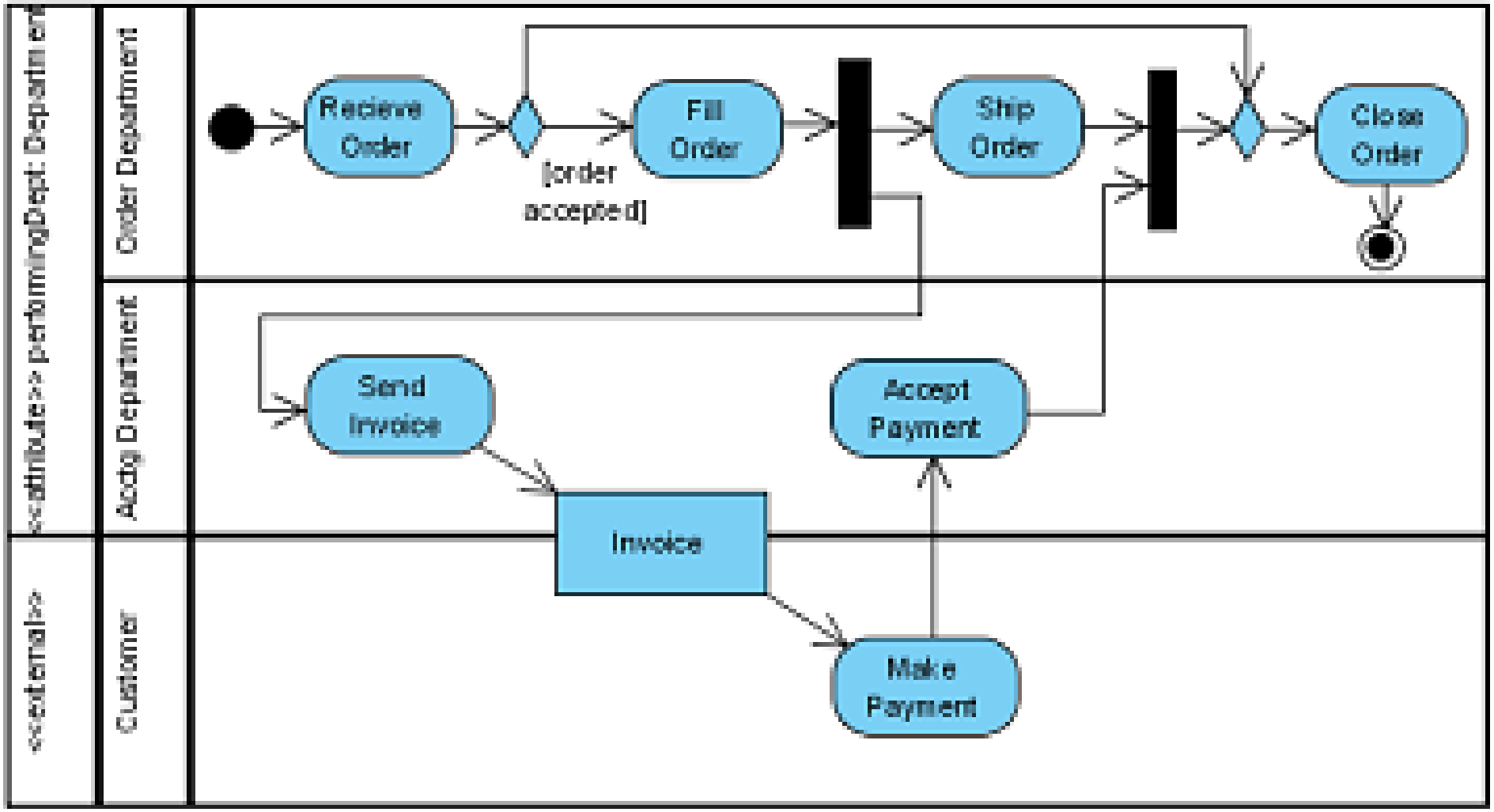
Partitions

- The contents of an activity diagram may be organized into *partitions*
- Does not have a formal semantic interpretation
- May represent organizational unit

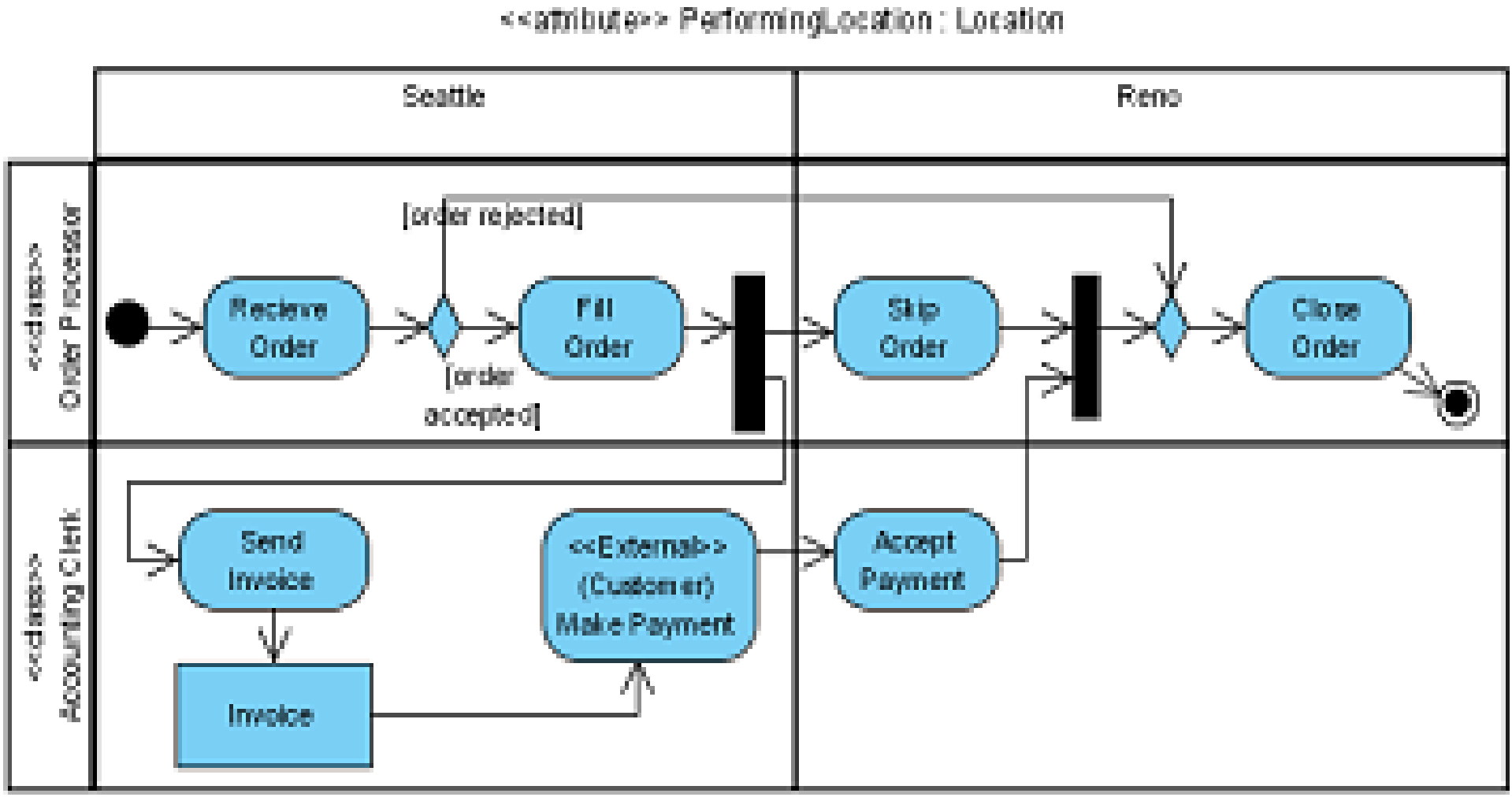



Passenger checks in
Passenger
Passenger Services
Showing Ticket at Check-In Counter
Verifying Ticket
[Ticket OK]
[Else]
Checking Luggage
Referring Passenger to Customer Services
Accepting Luggage
[Luggage Subject to Fee]
[Else]
Paying Fee
Issuing Boarding Pass

Dimensional Partition



Multidimensional Hierarchical Diagram



Activity vs. Sequence Diagrams

- Activity diagrams give focus to the **workflow**
- Sequence diagrams give focus to the **handling of business entities**.
- Activity diagram with partitions focuses on **how you divide responsibilities onto classes**
- The sequence diagram helps you understand **how objects interact and in what sequence.**

When to Use Activity Diagrams

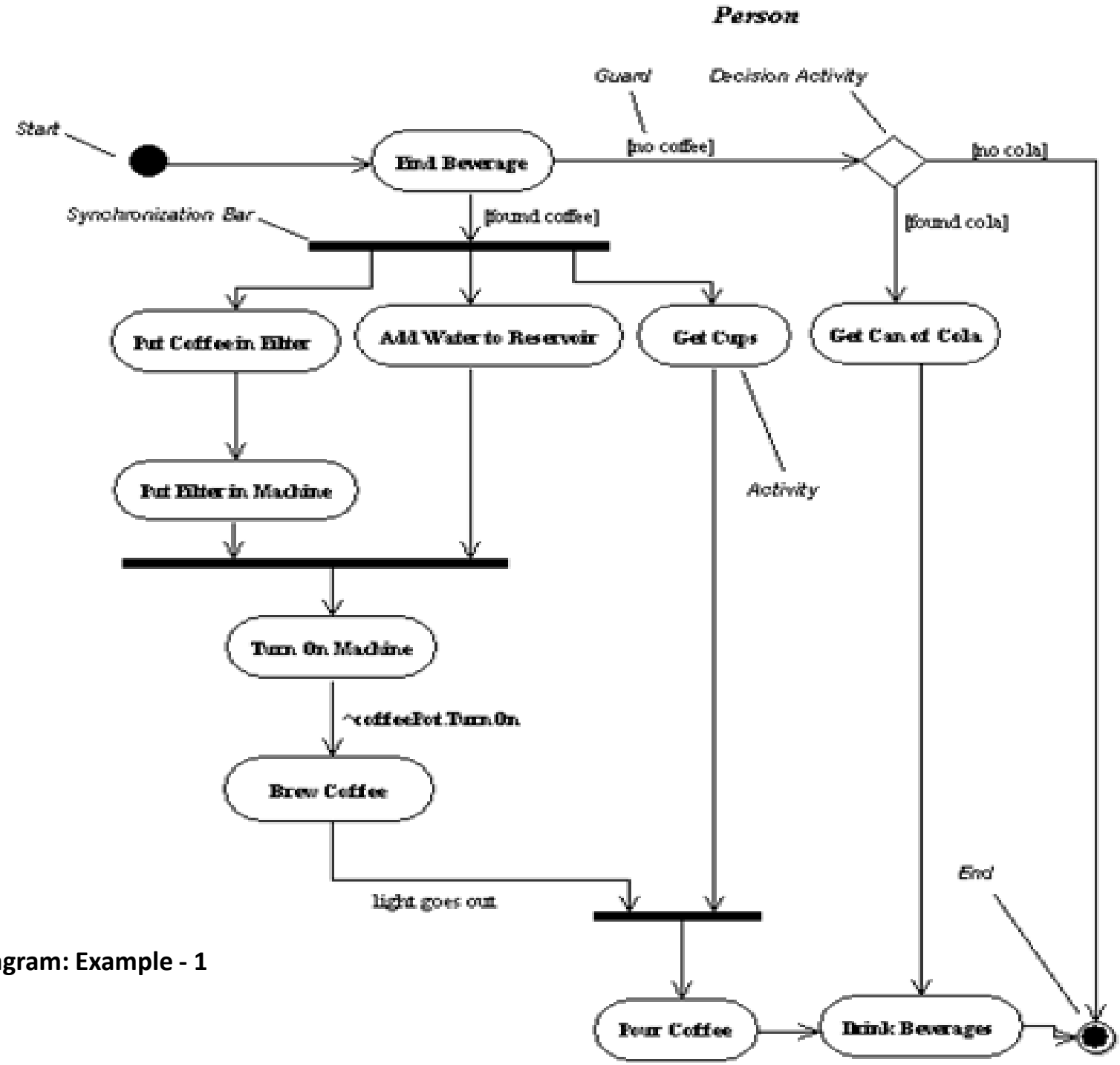
- Describe a behavior which contains parallel activities

Or

- Show how behaviors in several use-cases interact.

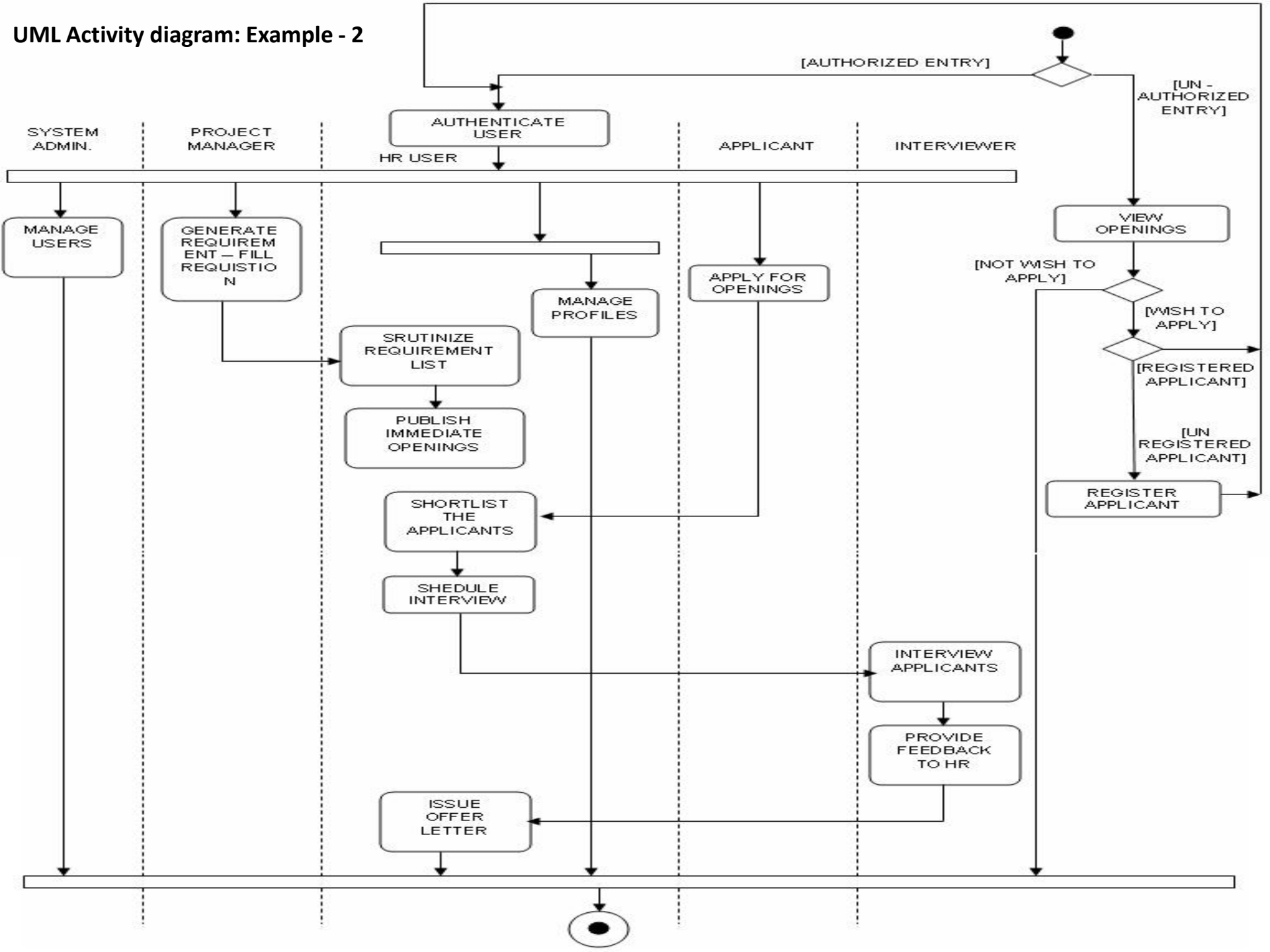
Specification standards

- No need for documenting the Activity diagrams beyond diagram itself.
- However, most UML tools provide in-built documentation capturing and printing capabilities for the Activity diagram and its elements.



UML Activity diagram: Example - 1

UML Activity diagram: Example - 2



Thank You