

STATE DIAGRAMS

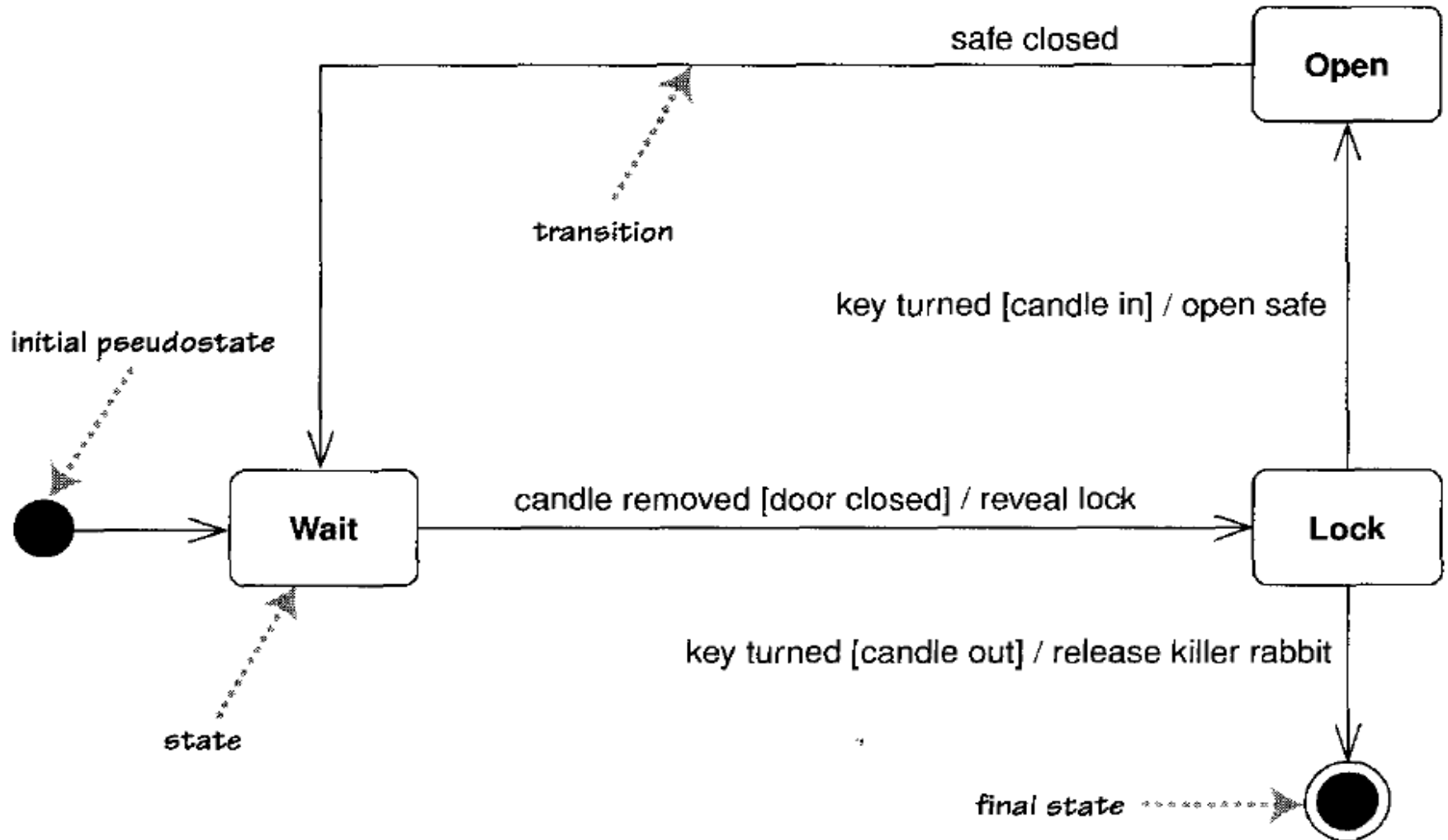
Introduction

- In object-oriented approaches, you draw a state diagram
 - for a single class, to show the lifetime behavior of a single object

Example

- A controller for a secret panel in a castle.
 - In this castle, you want to keep your valuables in a safe that's hard to find.
 - So to reveal the lock of the safe, you have to remove a strategic candle from its holder, but this will reveal the lock only while the door is closed.
 - Once you can see the lock, you can insert your key to open the safe.
 - For extra safety, you make sure that you can open the safe only if you replace the candle first.
 - If a thief comes before you open the safe, you will unleash a nasty monster to kill him.

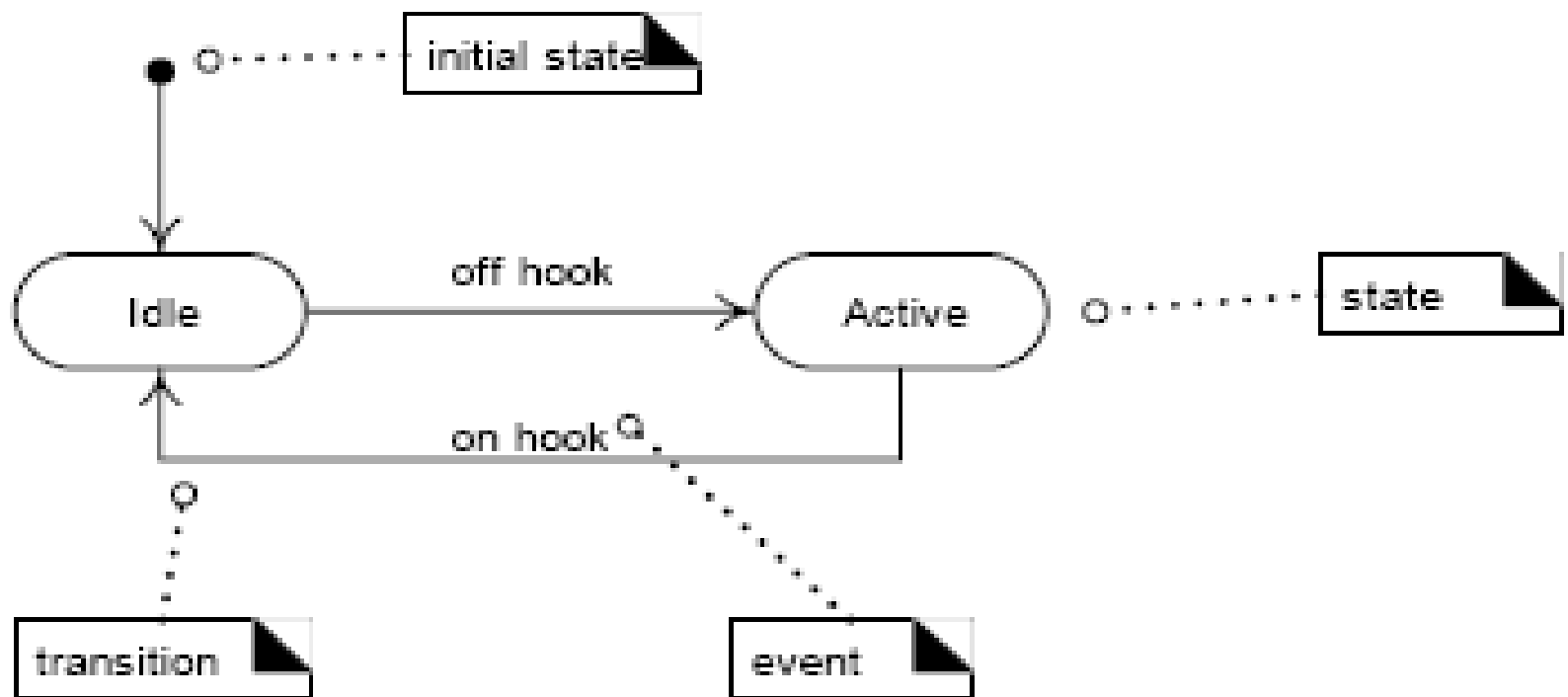
Example



Events, States and Transitions

- An **event** is a significant occurrence.
 - A telephone receiver is taken off the hook.
- A **state** is the condition of an object at a moment in time—the time between events.
 - A telephone is in the state of being "idle" after the receiver is placed on the hook and until it is taken off the hook.
- A **transition** is a relationship between two states that indicates that when an event occurs, the object moves from the prior state to the subsequent state.
 - When the event "off hook" occurs, transition the telephone from the "idle" to "active" state.

Telephone

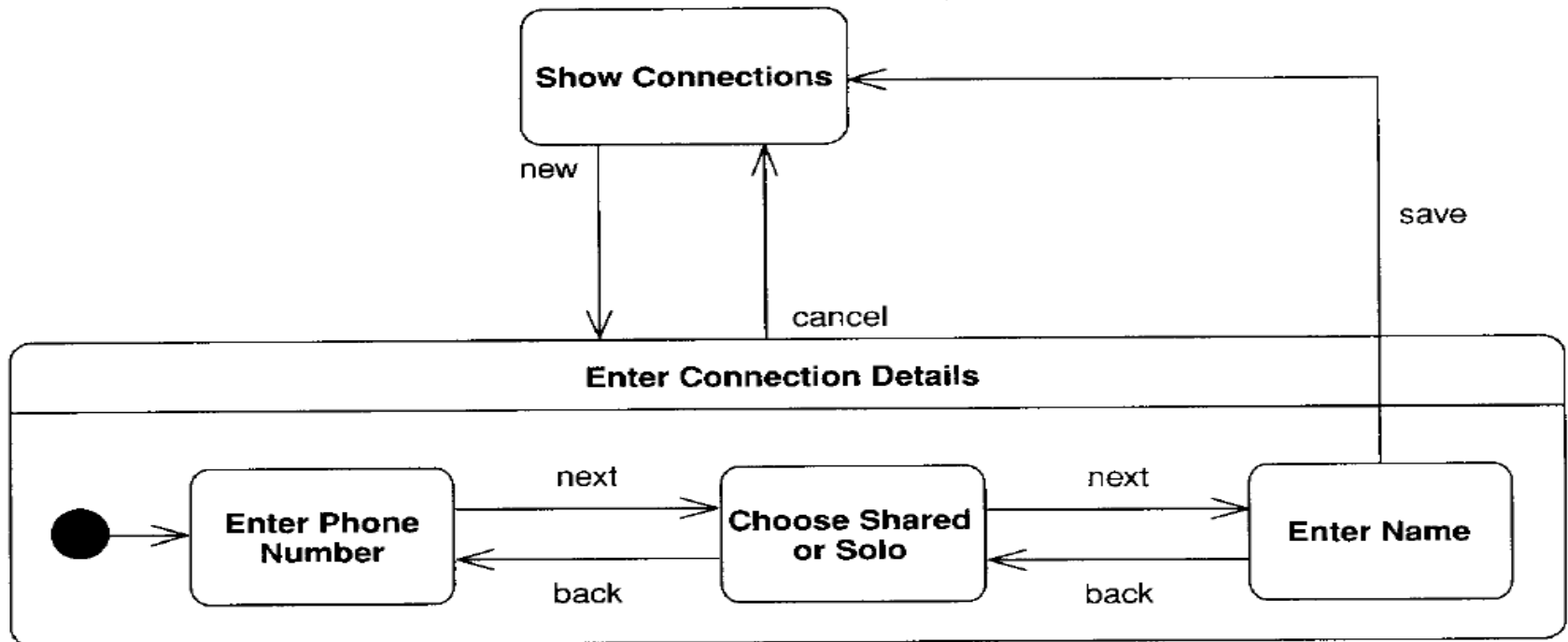


Superstates

- Often, several states share common transitions and internal activities.
- In these cases, you can make them substates and move the shared behavior into a superstate.

Superstates

- Without the superstate, you would have to draw a cancel transition for all three states within the Enter Connection Details state.



When to Use State Diagrams

- Good at describing the behavior of an object across several use cases.
- Are not very good at describing behavior that involves a number of objects collaborating.
- So, it is useful to combine state diagrams with other techniques.
 - interaction diagrams are good at describing the behavior of several objects in a single use case,
 - activity diagrams are good at showing the general sequence of activities for several objects and use cases.

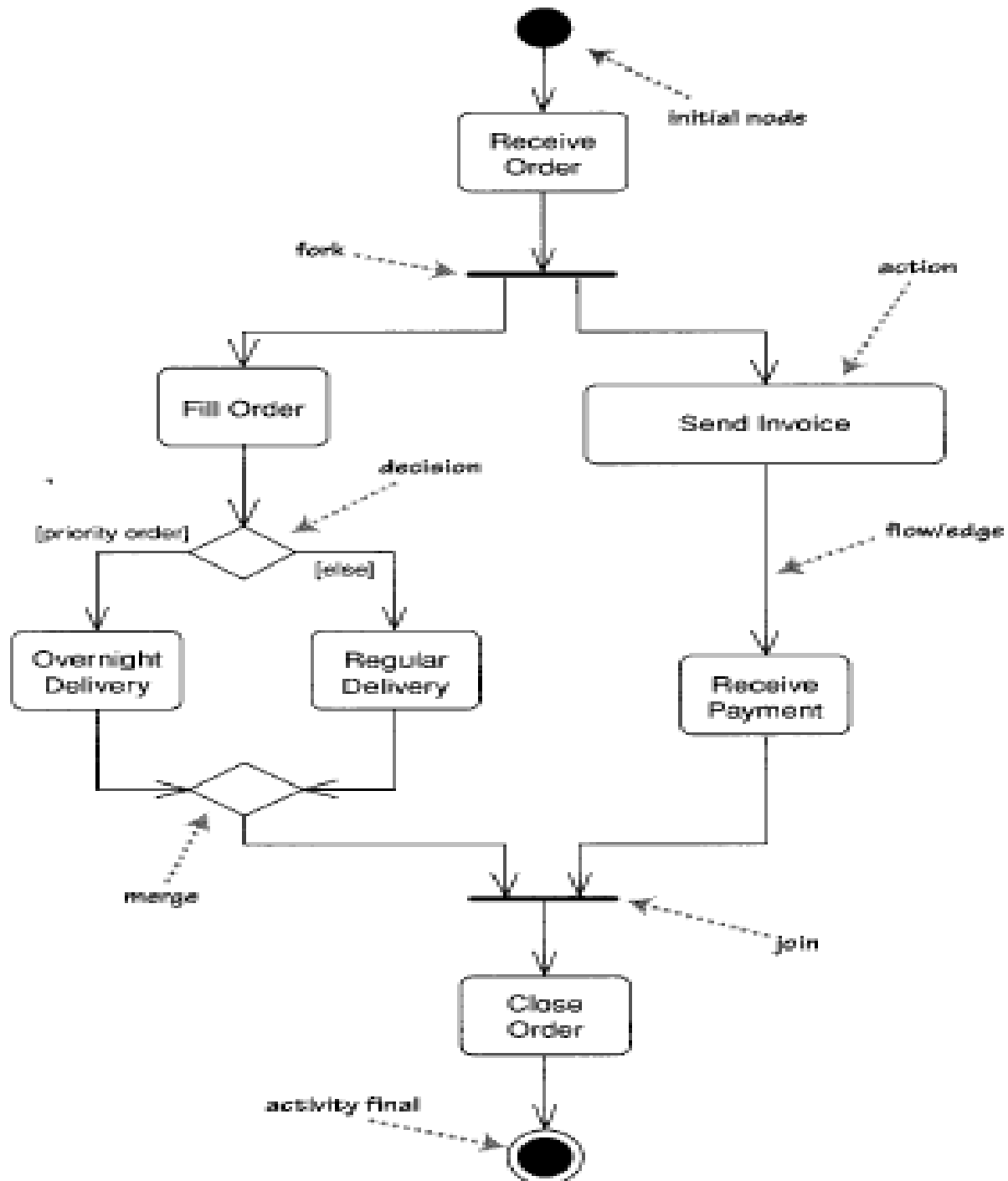
When to Use State Diagrams

- If you do use state diagrams,
 - Do not try to draw them for every class in the system.
 - Use state diagrams only for those classes that exhibit interesting behavior, where building the state diagram helps you understand what is going on.
- Many people find that User Interface have the kind of behavior that is useful to depict with a state diagram .

ACTIVITY DIAGRAMS

Activity Diagrams

- Activity diagrams are a technique to describe procedural logic, business process, and work flow.
- Similar to flowcharts, but the principal difference between them and flowchart notation is that,
 - they support parallel behavior.



Example

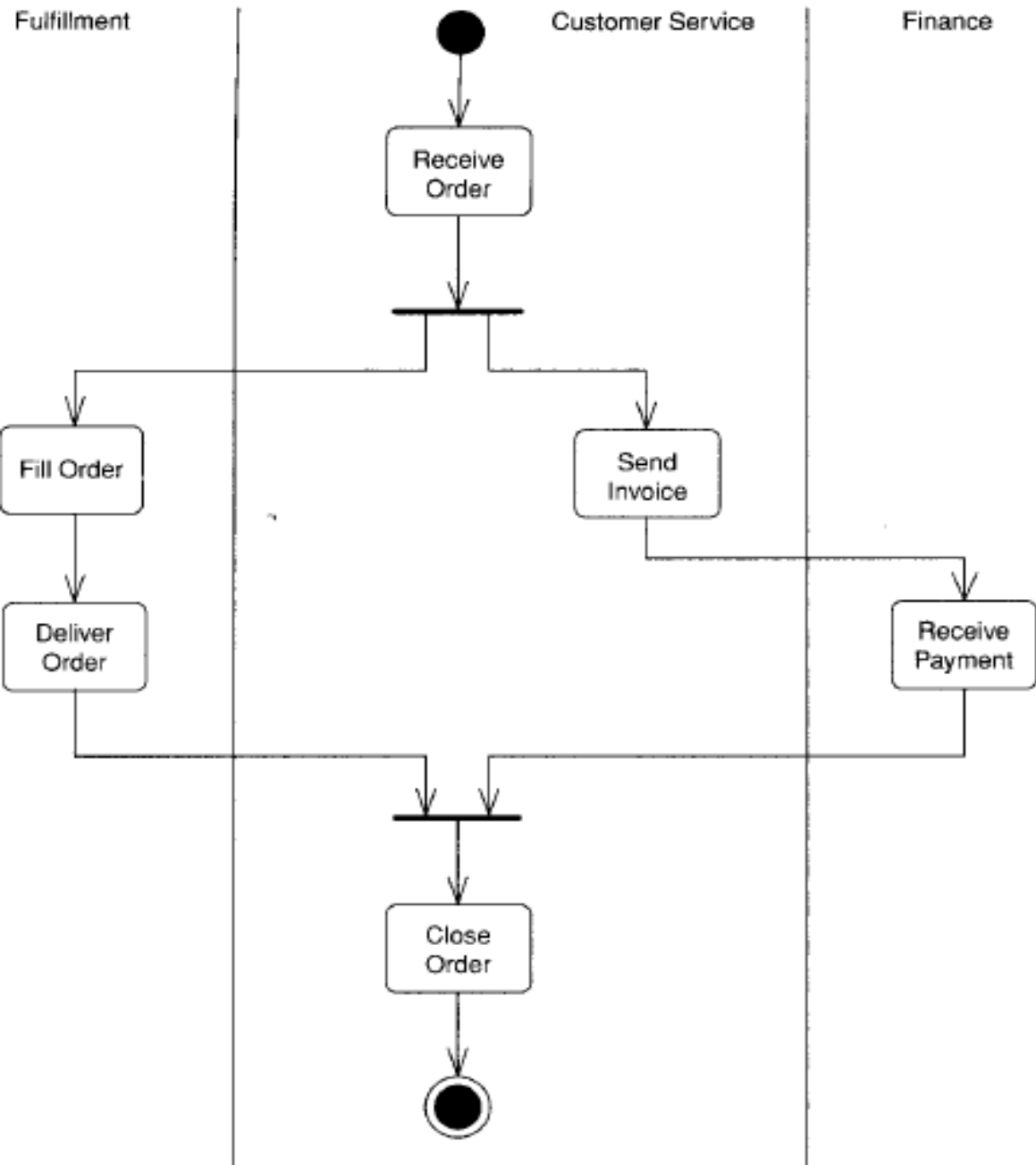
- We begin at the initial node action and then do the action Receive Order.
- Once that is done, we encounter a fork.
- A fork has one incoming flow and several outgoing concurrent flows.
- Fill Order, Send Invoice, and the subsequent actions
- occur in parallel.
- Essentially, this means that the sequence between them is irrelevant.
- I could fill the order, send the invoice, deliver, and then receive payment; or,
- I could send the invoice, receive the payment, fill the order, and then deliver.

Activity Diagrams

- Useful for concurrent algorithms, in which independent threads can do things in parallel.
- When you have parallelism, you'll need to synchronize.
- We don't close the order until it is delivered and paid for.
- We show this with the join before the Close Order action.
- With a join, the outgoing flow is taken only when all the incoming flows reach the join.
- So you can close the order only when you have both received the payment and delivered .

Partitions

- Activity diagrams tell you what happens, but they do not tell you who does what.
- In programming, this means that the diagram does not express which class is responsible for each action .
- If you want to show who does what, you can divide an activity diagram into partitions,
 - which show which actions one class or organization unit carries out .



When to Use Activity Diagrams

- They support and encourage parallel behavior. This makes them a great tool for work flow and process modeling.
- You can also use an activity diagram as a UML-compliant flowchart.
- You can take advantages of the forks and joins to describe parallel algorithms for concurrent programs.
- Often used to describe a use case .