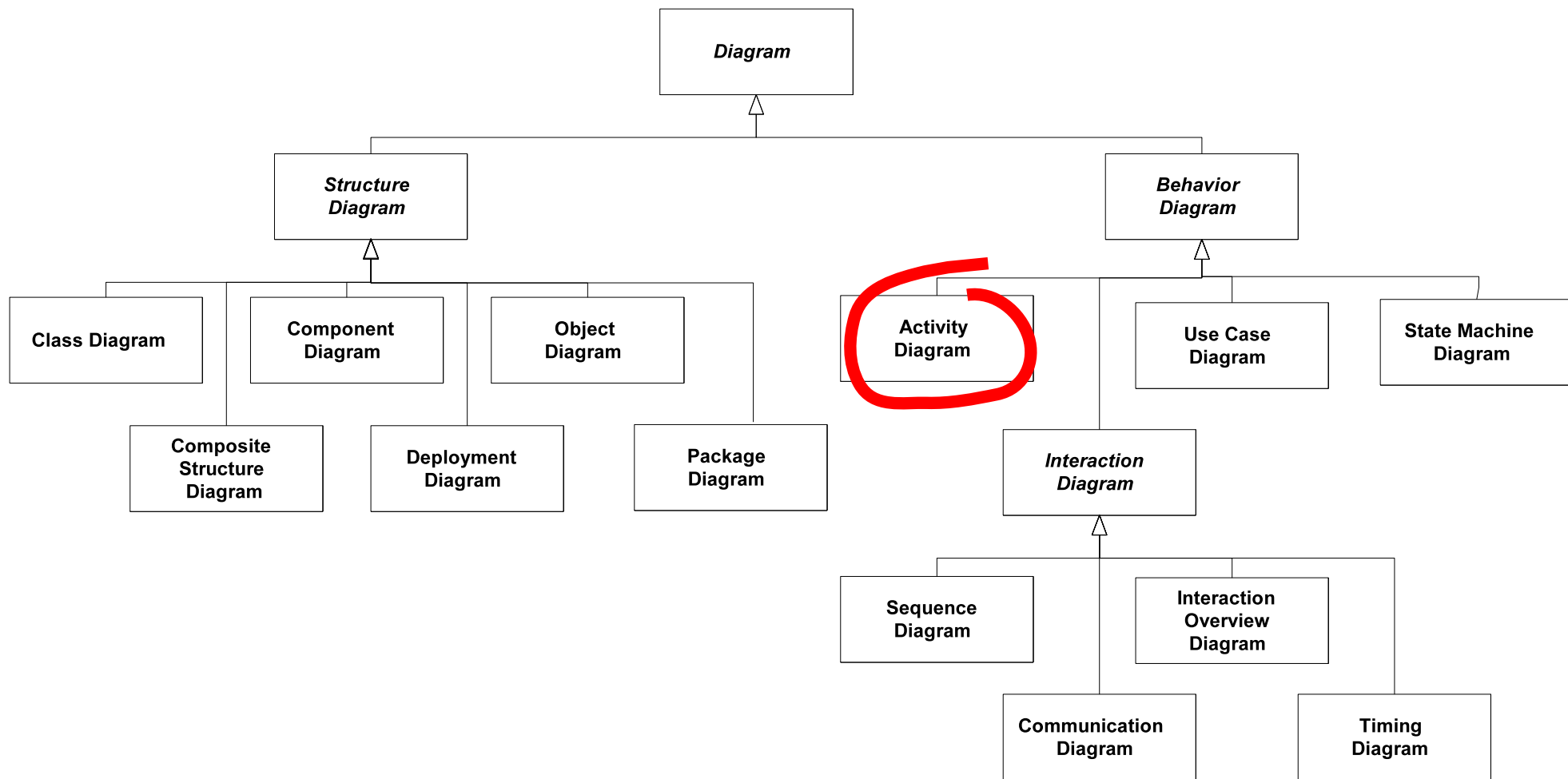# UML: Activity diagrams

Davide Rossi
Dipartimento di Informatica – Scienze e Ingegneria
Università di Bologna

```
                              Diagram

          Structure                            Behavior
           Diagram                             Diagram

Class Diagram    Component      Object        Activity         Use Case      State Machine
                 Diagram        Diagram       Diagram          Diagram       Diagram

         Composite      Deployment    Package                  Interaction
         Structure      Diagram       Diagram                  Diagram
         Diagram

                                                    Sequence        Interaction
                                                    Diagram         Overview
                                                                    Diagram

                                              Communication              Timing
                                              Diagram                    Diagram
```

Activity Diagrams

An activity diagram is an UML behavioral diagram which shows the flow of control or object flow with emphasis on the sequence and conditions of the flow. The actions coordinated by activity models can be initiated because other actions finish executing, because objects and data become available, or because some events external to the flow occur.

Activity Diagrams

Activity diagrams can be used to depict the behavior of elements such as: classes, use cases, interfaces, components, operations of a class.

They can model: business processes, behavior associated to use cases, behavior of a class operation, an algorithm.

Semantics

- Activity diagrams, in UML 2.x are formalized to be based on Petri net-like semantics.

- The semantics of the system is described in terms of transitions between markings (distributions of tokens in the network).

- The keyword is: concurrency.

Main elements of an AD

- Activity
- Activity nodes
  - Action
  - Object
  - Control
- Activity edges

Resolve Issue

Create ticket

Reproduce issue

Update ticket

[can't reproduce]

[issue reproduced]

[known issue]

Identify issue

[new issue]

Determine resolution

Fix Issue

[issue not resolved]

Verify Fix

[issue resolved]

Close ticket

© uml-diagrams.org

Process Order

Requested Order → Receive Order

[order accepted] → Fill Order → (fork) → Send Invoice → Invoice → Accept Payment → (join)

[order accepted] → Fill Order → (fork) → Ship Order → (join)

(join) → Close Order

[order rejected]

Activity

An activity is a parametric behavior represented as coordinated flow of actions.
The flow of execution is modeled as activity nodes connected by activity edges.
An activity could be rendered as round-cornered rectangle with the activity name in the upper left corner and nodes and edges of the activity inside the rectangle.
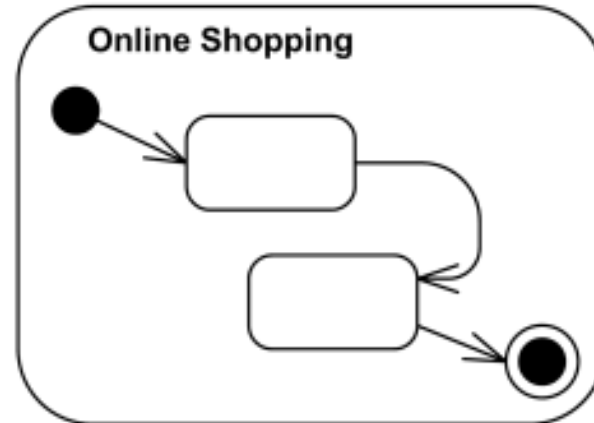
Actions

An action is a named element which represents a single atomic step within an activity.
There are various kinds of actions:

- Occurrences of primitive functions or call to operations.

- Communication actions, such as sending or receiving signals.

- Manipulations of objects, such as reading or writing attributes or associations.

- Invocations of behavior, such as activities.

Actions

Actions are notated as round-cornered rectangles. Name or description of the action is placed inside of the rectangle. An action may have sets of incoming and outgoing activity edges that specify control flow and data flow from and to other nodes. An action will not begin execution until all of its input conditions are satisfied.

Process
Order

Activity Edge

An activity edge is a directed connection between two activity nodes along which tokens may flow, from the source activity node to the target activity node. It is a generalization of control flow and object flow edges. Activity edge can have a guard - specification evaluated at runtime to determine if the edge can be traversed.
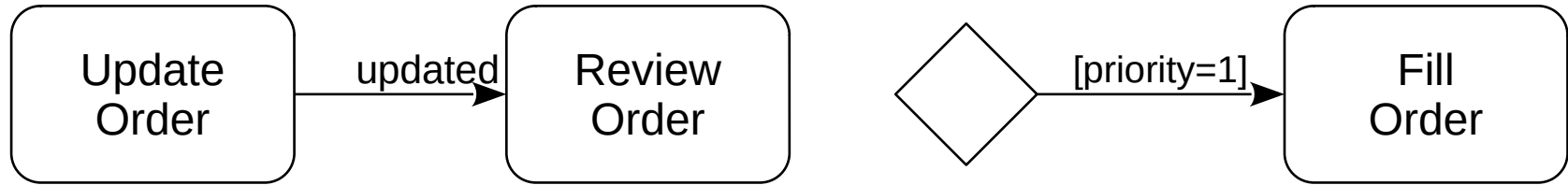
Object

An object node is an activity node that is part of defining object flow in an activity. It indicates that an instance of a particular Classifier, possibly in a particular state, may be available at a particular point in the activity.

```
┌─────────┐                    ┌─────────┐
│  Fill   │                    │ Review  │
│  Order  │                    │  Order  │
└─────────┘                    └─────────┘
      ↘                          ↗
        ┌──────────────┐
        │    Order     │
        └──────────────┘
```
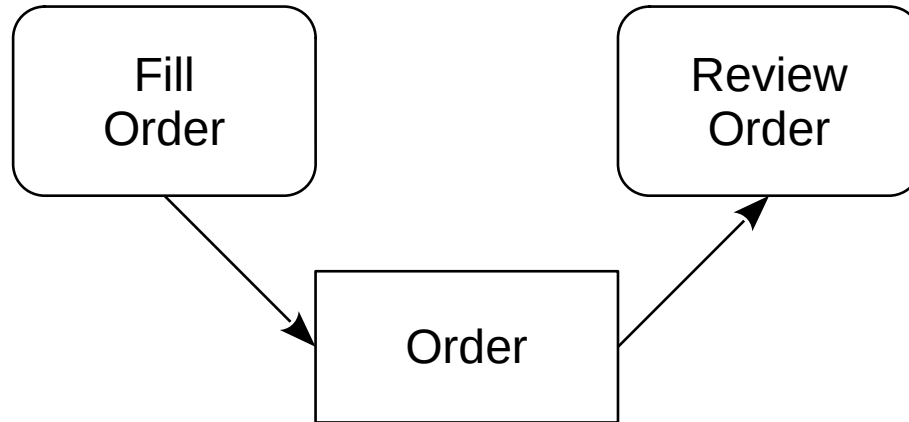
Input/output pins

Input pins are object nodes that receive values from other actions through object flows. Output pins are object nodes that deliver values to other actions through object flows.
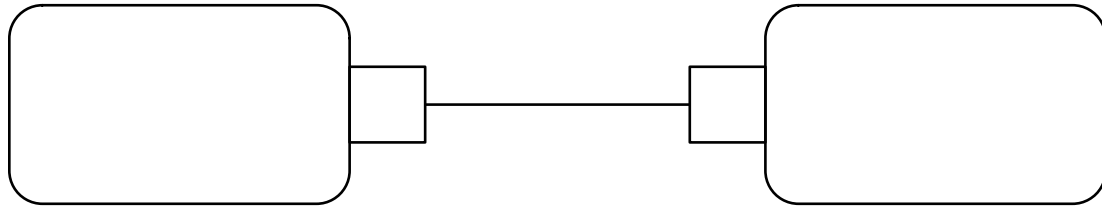
Activity Parameter Node

Activity parameter nodes are object nodes at the beginning and end of flows that provide a means to accept inputs to an activity and provide outputs from the activity, through the activity parameters.
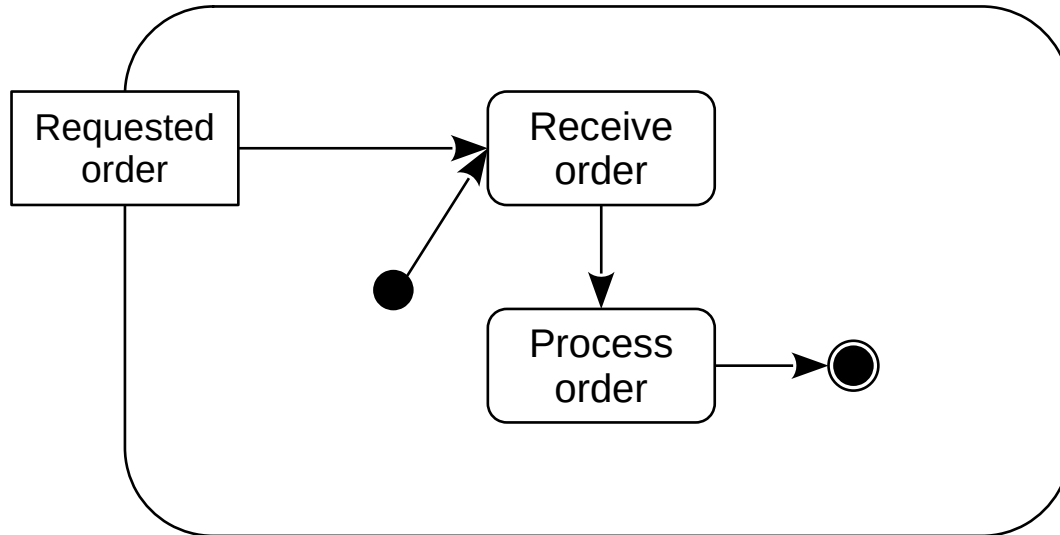
Connectors

An activity edge can be notated using a connector, which is a small circle with a name inside. Connectors are generally used to avoid drawing a long edge. This is purely notational. It does not affect the underlying model.
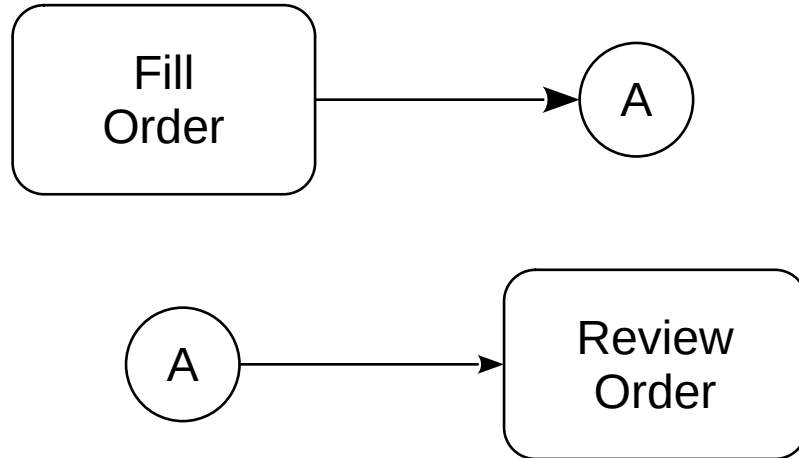
Control

A control node is an activity node used to coordinate the flows between other nodes. This includes: initial node, flow final node, activity final node, decision node, merge node, fork node, join node.


initial node

fork node

join node

merge node

decision node

activity final node
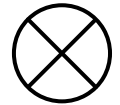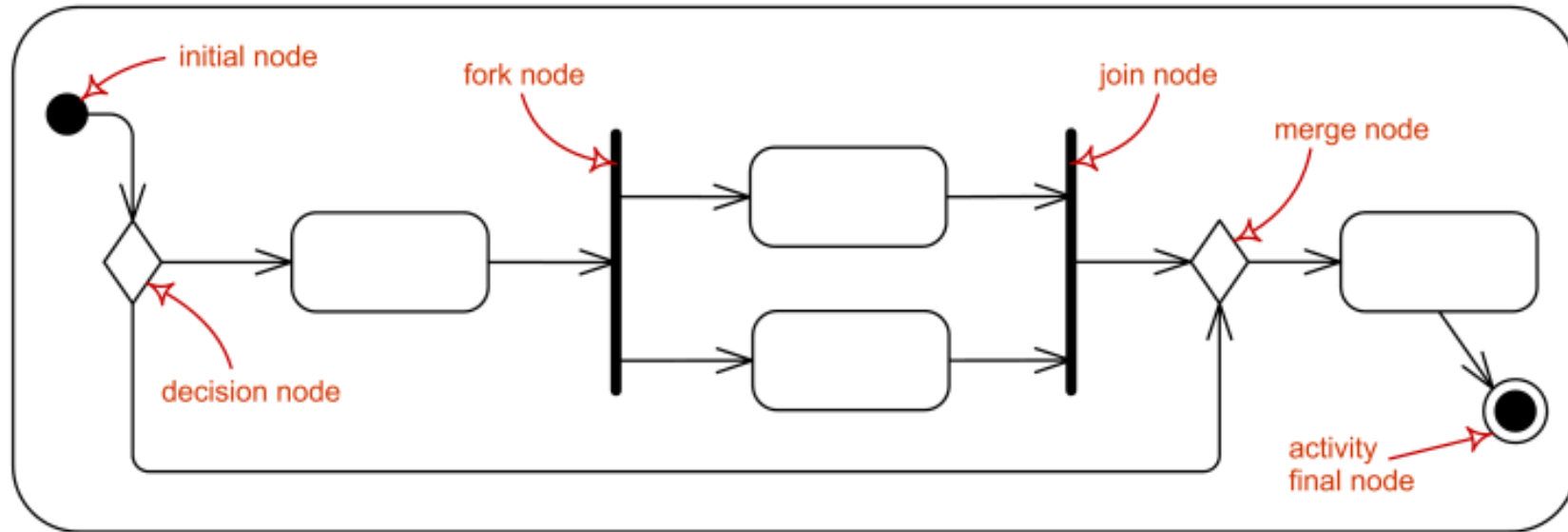
Flow final node

Token competition

Token are not "pushed" into actions, actions *accept* tokens.
When an action completes, the token is released and it is *offered* to other action**s** downstram.
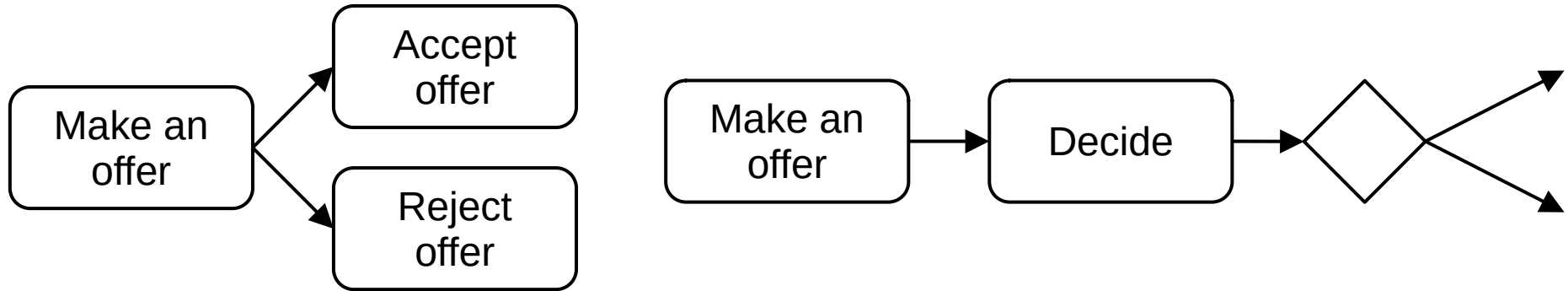
Expansion region

An **expansion region** is a structured node that takes collections as input, acts on each element of the collections individually and produces elements to output collections. Elements processing can take place sequentially («iterative»), concurrently («parallel») or in a streamline («stream»).
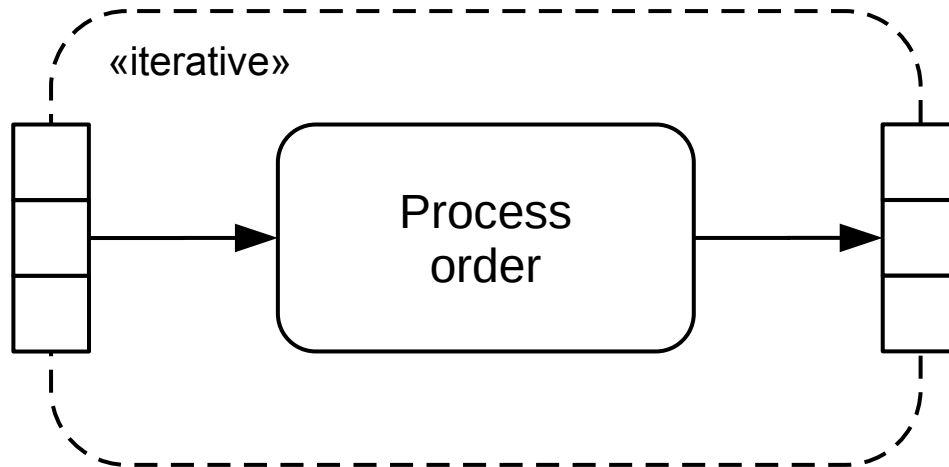
Activity Partition

An activity partition is an activity group for actions that have some common characteristic.

Partitions provide a constrained view on the behaviors invoked in activities and often correspond to organizational units or business actors in a business model.
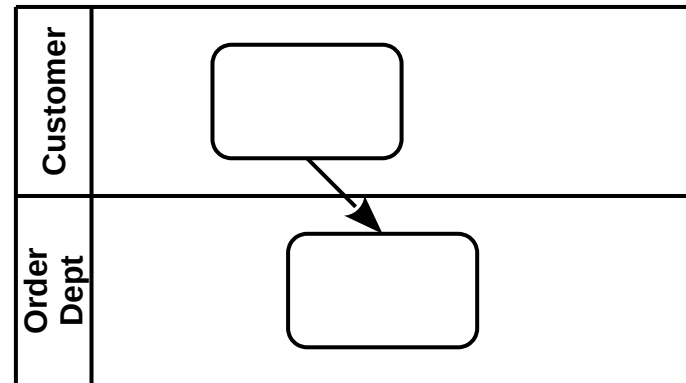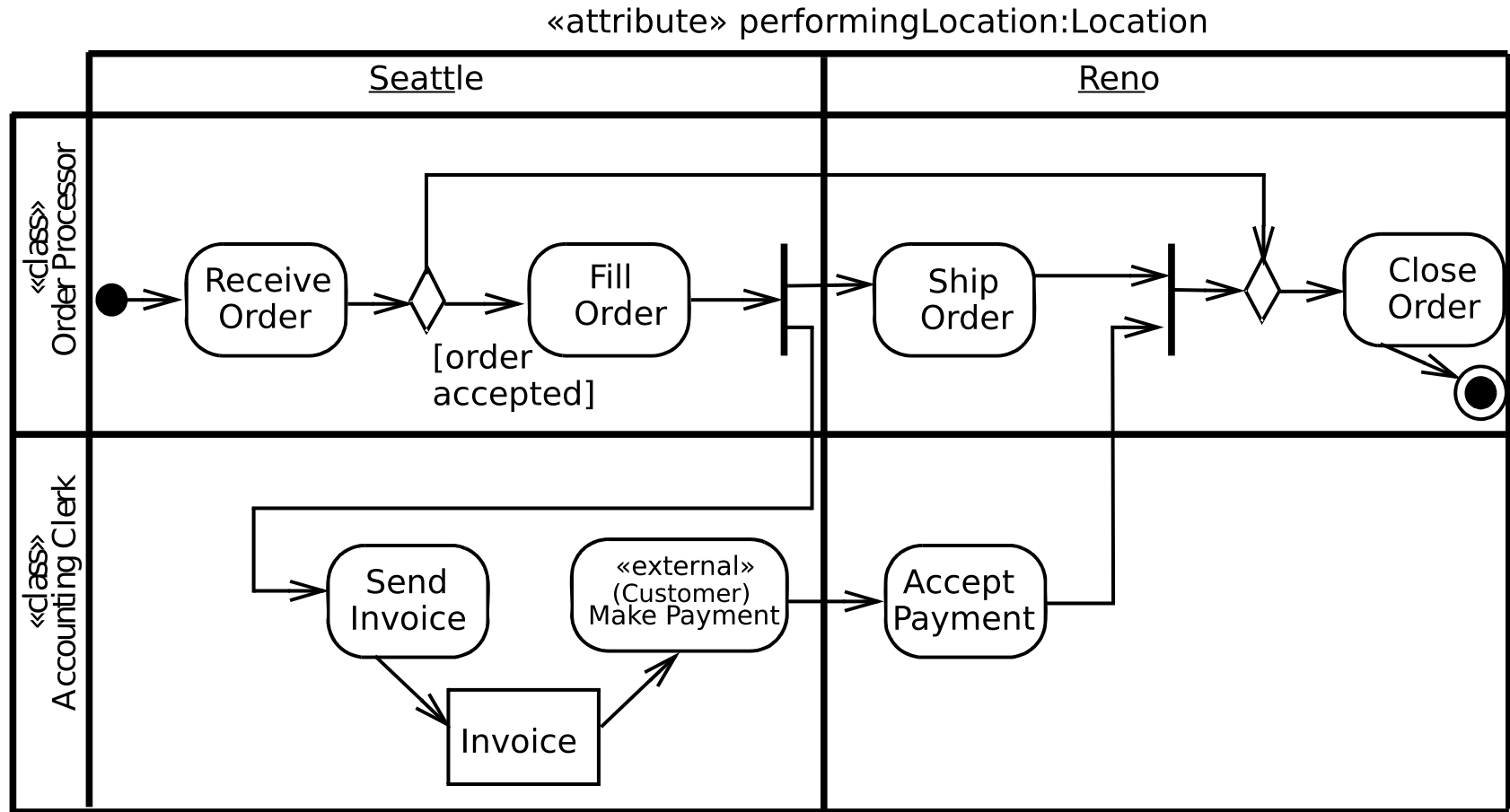
Alternative notations

Using activity partitions is not the only way to associate actions and actors/classifiers. Preceding the name of the action with the actor/classifier name between parentheses can be used as well.

(CustomerService)
Close order

Activity Partition example

Interruptible regions and edges

An interruptible activity region is a type of activity group which provides a mechanism for destroying all tokens and terminating all behaviors in the section of the activity enclosed within the boundary of the region. When a token is accepted by a special kind of edge called an interrupting edge, which is designated by a lightning bolt, it leaves the region and all other tokens are destroyed and behaviors within the region are terminated.
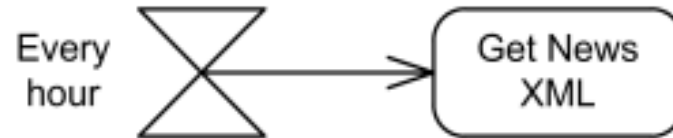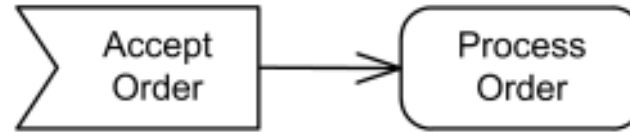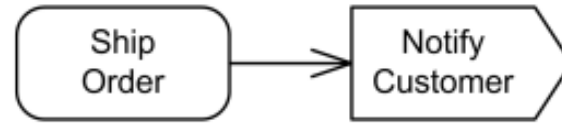
Interruption example
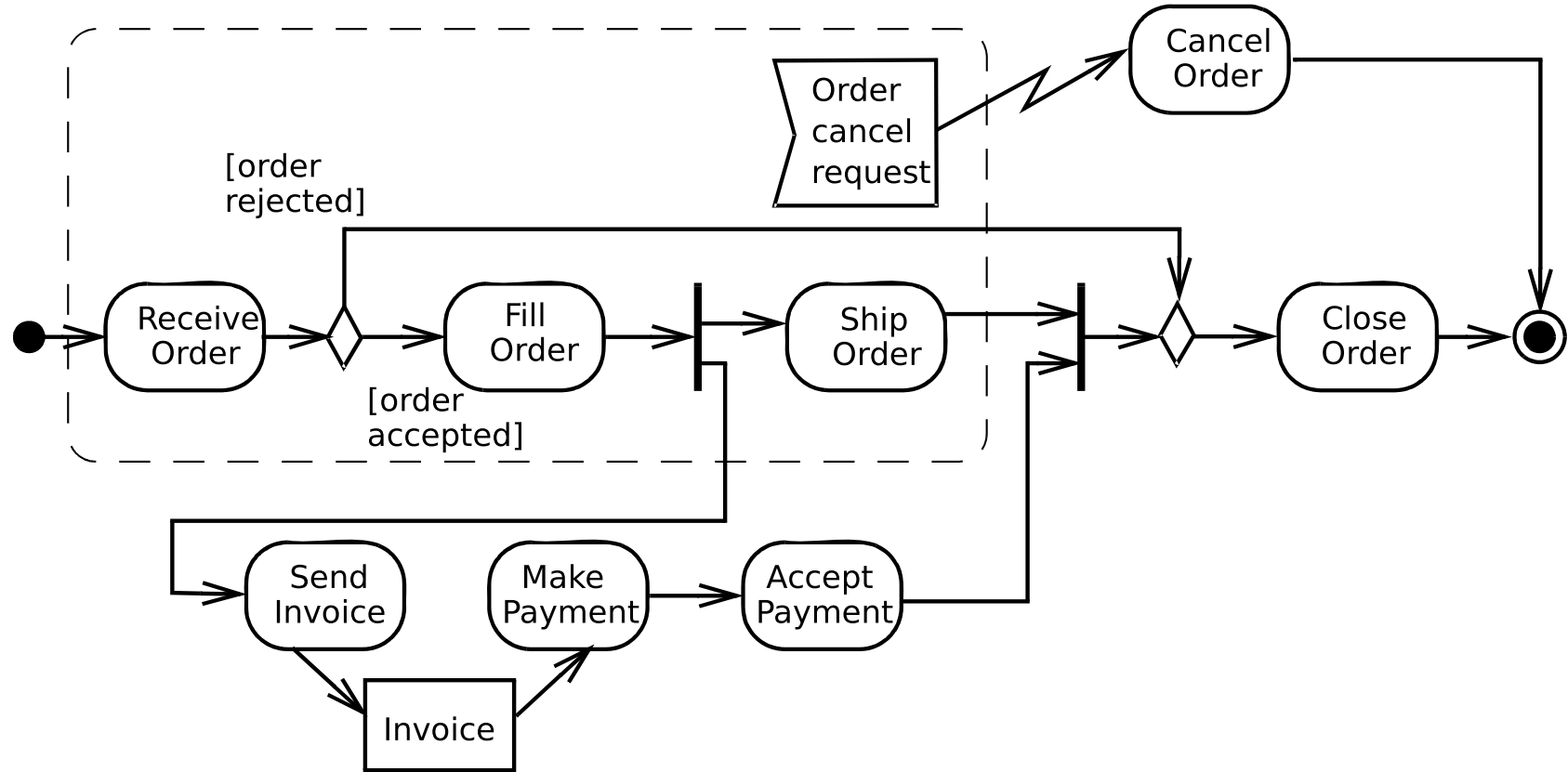
Event Actions

Event actions are used to model interaction taking place with elements outside the current activity.

Producing events is asynchnous, consuming events is synchrnous (i.e. blocking).

Event Actions

- Send signal

- Accept signal

- Repetitive time

Exercise

Draw an UML activity diagram depicting the purchase process for a beverage can from an automatic selling machine that accepts payments with coins (eventually dispensing change), pre-paid cards or credit cards.

The beverage can can be dispensed while payment is finalized (change is given, card is returned).

If the cancel button is pressed at any point, the process terminates immediately, and any inserted coins are refunded.