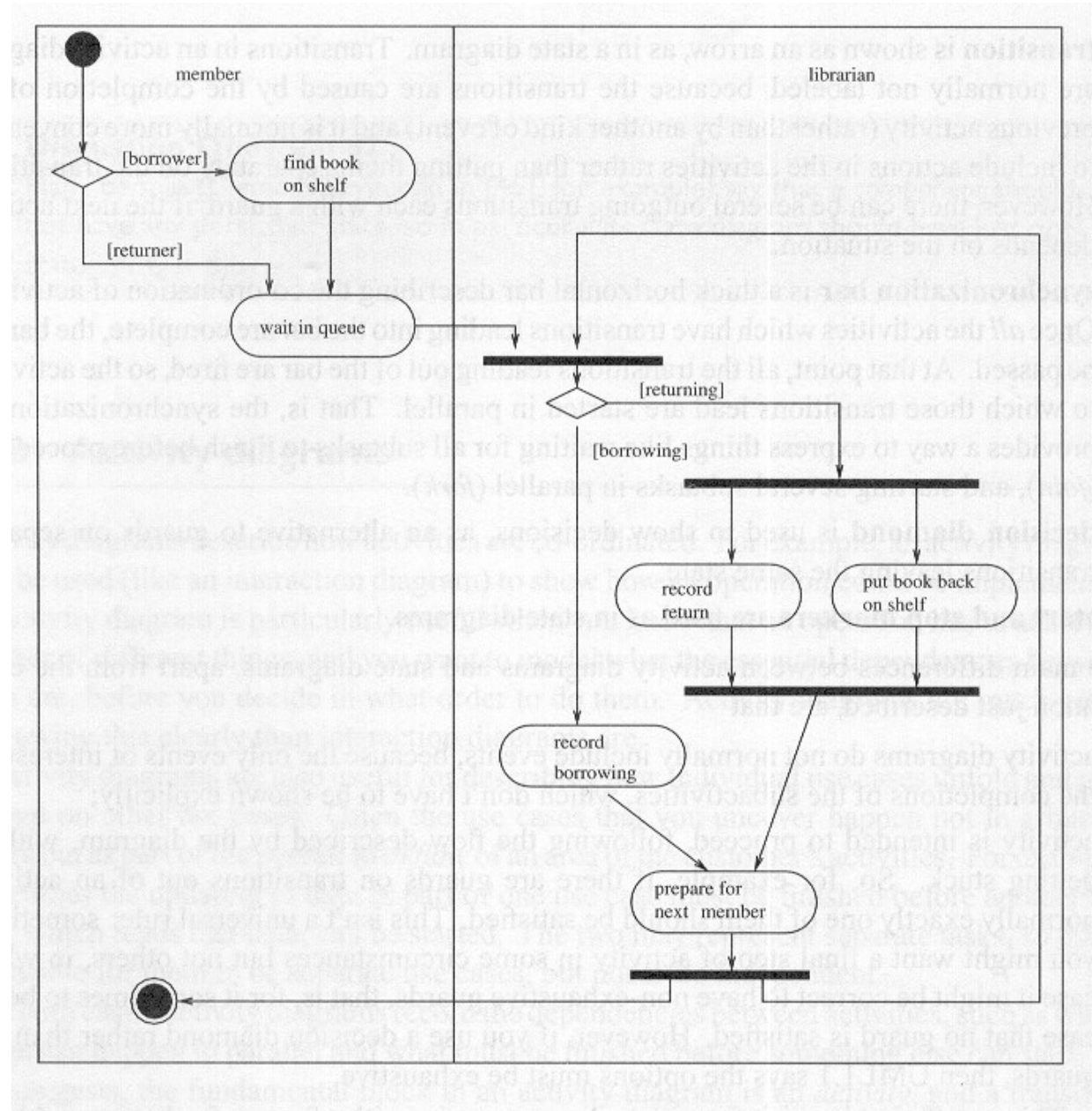


UML Activity Diagrams

Example



Activity Diagram: what is it?

- Describes activities and flows of data or decisions between activities
- Provides a very broad view of business processes
- Can be used to break out the activities that occur within a use case
- Good for showing parallel threads

Activity Diagram: when to use it?

- When describing work flow across many use cases
- When analysing a use case, and before methods are assigned to symbols
- When dealing with multi-threaded applications

Creating an Activity Diagram

This diagram is useful in showing work flow connections and describing behaviour that has a lot of parallel processing.

When you use an activity diagram you can choose the order in which to do things.

It expresses the essential sequencing rules to follow.

It is different from a flow chart in that it shows parallel processes, not just sequential processes.

- Activity diagram is to visualize and exploit the opportunity of parallel and concurrent processing
Flow chart is limited to sequential process with condition checking points.

Activity Diagram

- Shows a set of activities, the flow from activity to activity.
- Represents the dynamic view of a system.
- Models the function of a system.

Activity Diagram

- Activity diagrams show the flow of control between activities
 - They can model the sequential and concurrent steps in a computational process
 - They can also model the flow of an object as it moves from state to state at different points in the activity

UML Activity symbols

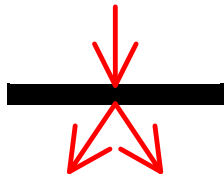
ActionState



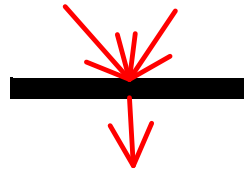
initial state



final state



Transition
(Fork)



Transition
(Join)



Control Flow



Decision



Swimlane

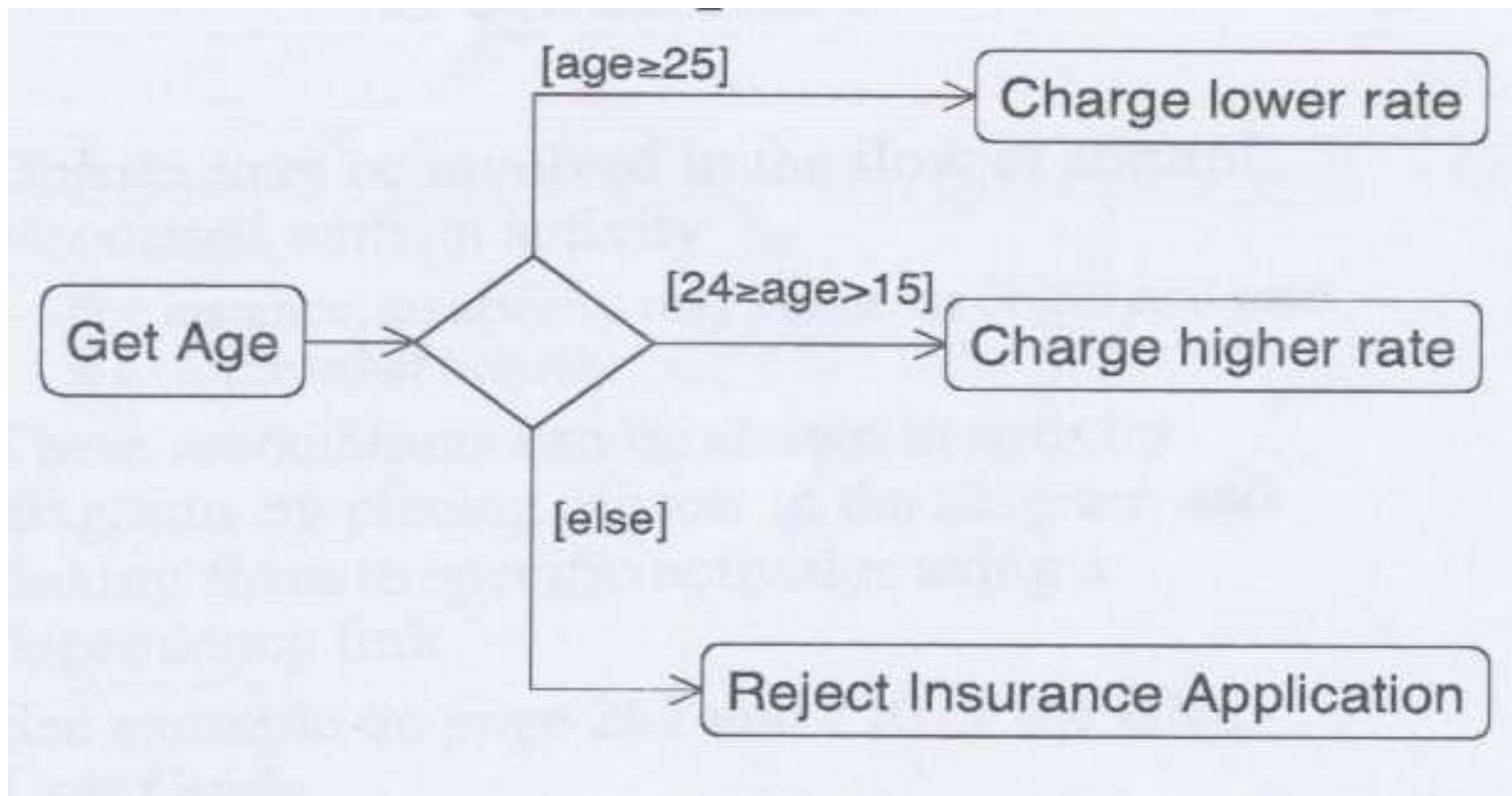
Action State

- In a conceptual diagram an activity is a task that needs to be done – either by a human or a computer
- In a specification-perspective diagram or an implementation-diagram, an activity is a method on a class

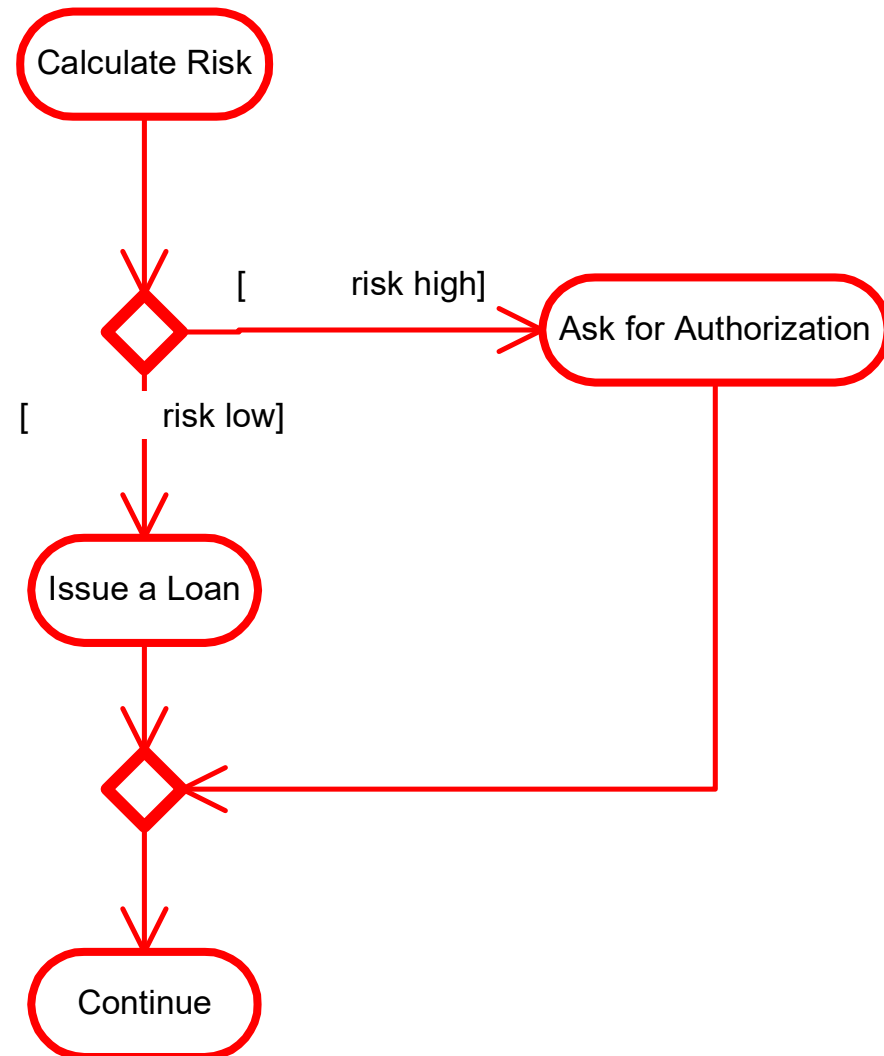
Sequential Branching

- A sequential branch is represented as a diamond
 - It may have one incoming transition and two or more outgoing transitions
 - Guards are associated with each transition. The guards are evaluated upon entering the branch, and the one that evaluates to true is then taken

Example



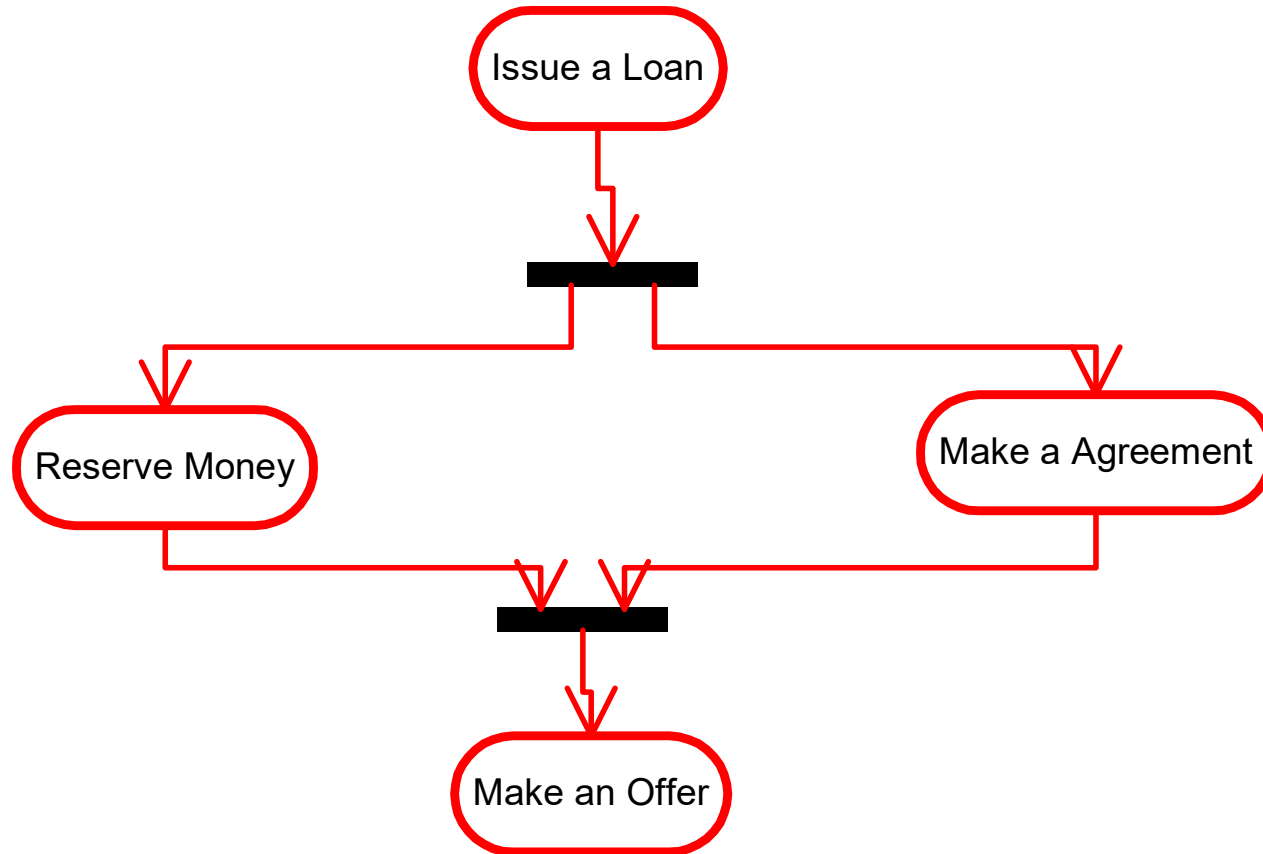
Branch & Merge

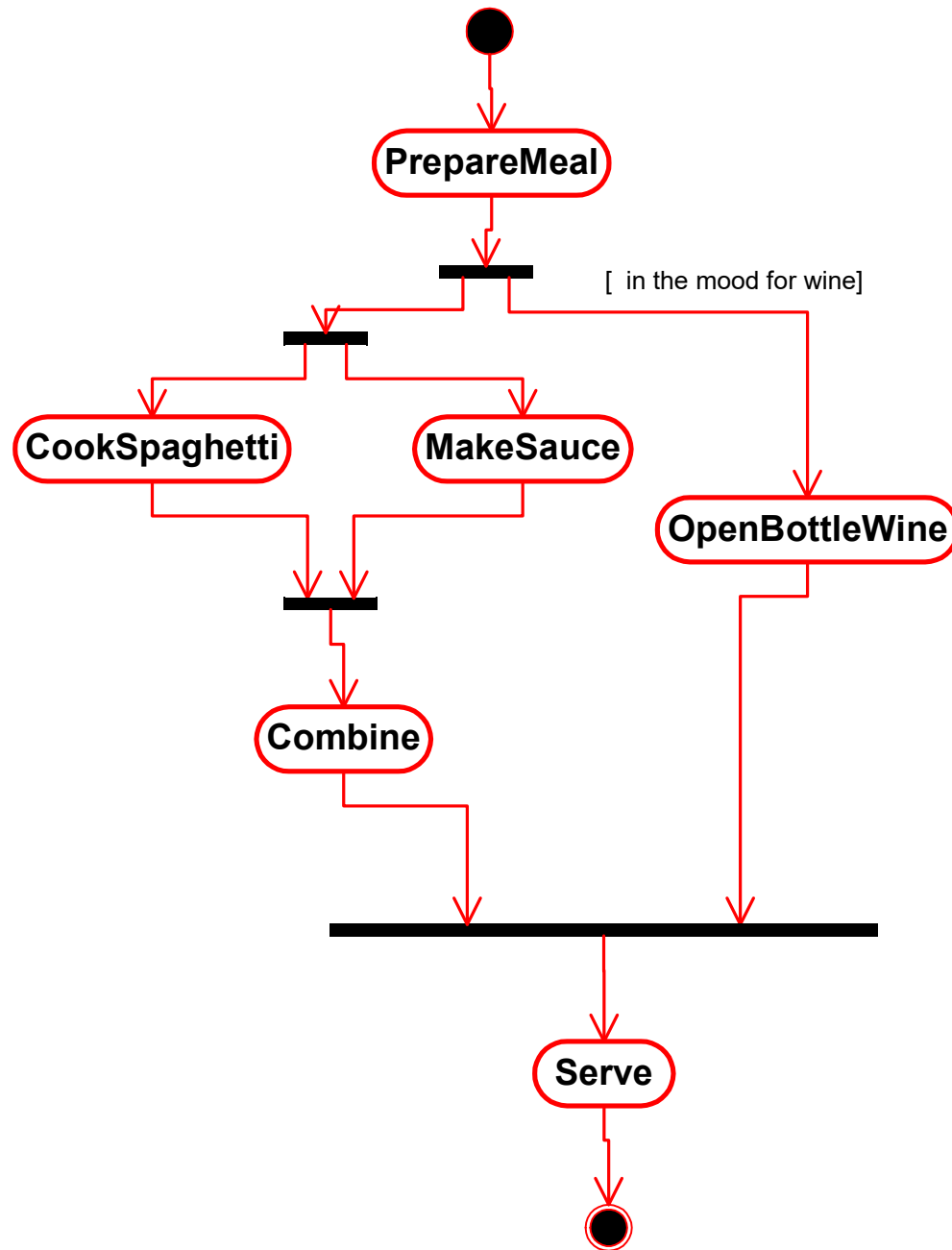


Forking and Joining

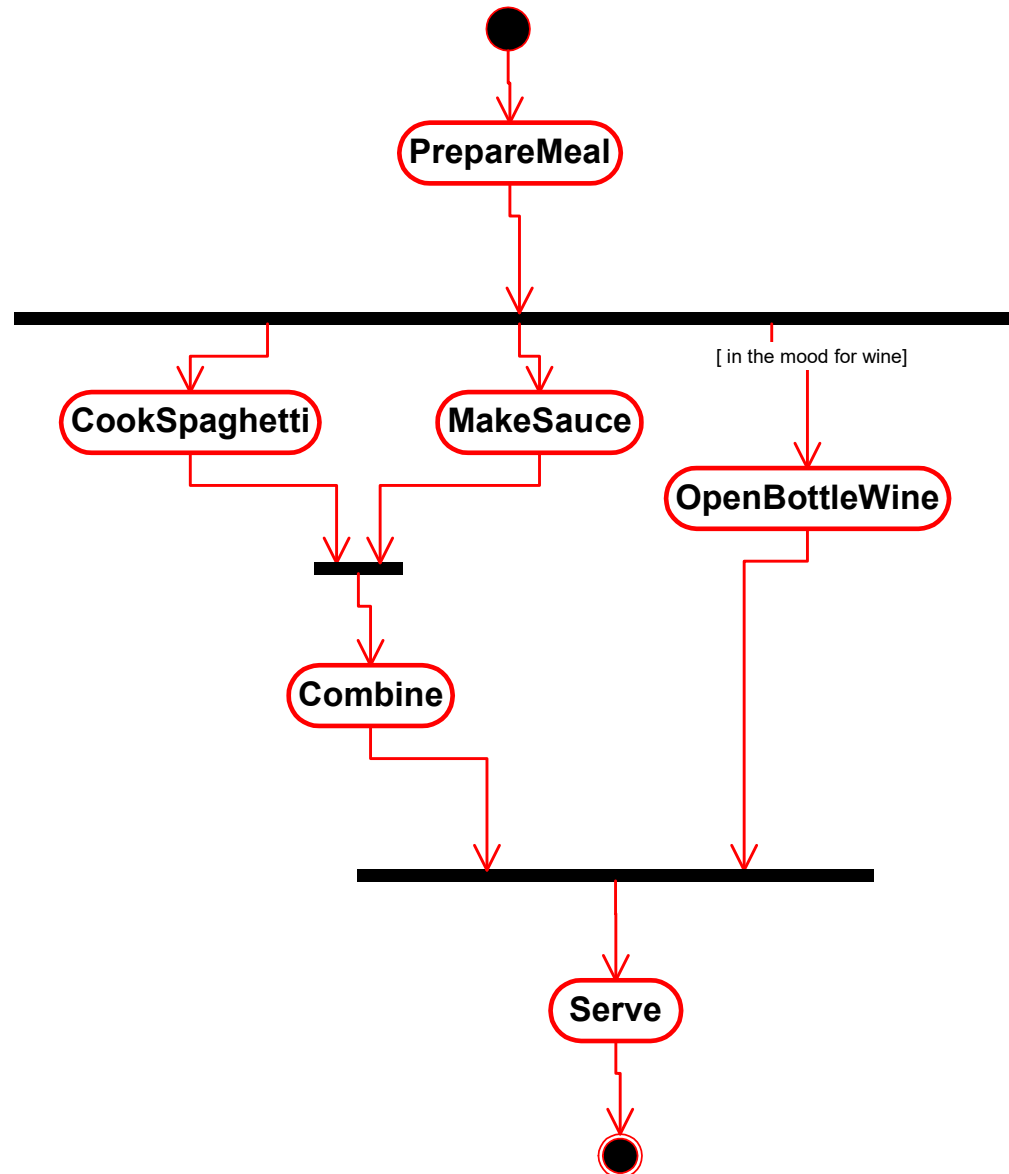
- Concurrent activities within an activity diagram are modelled with the use of synchronised bars
- Synchronisation bars are drawn as a thick horizontal or vertical line
- Joins and Forks should balance
 - The number of flows that leave a fork should equal to the number of flows entering the corresponding join.

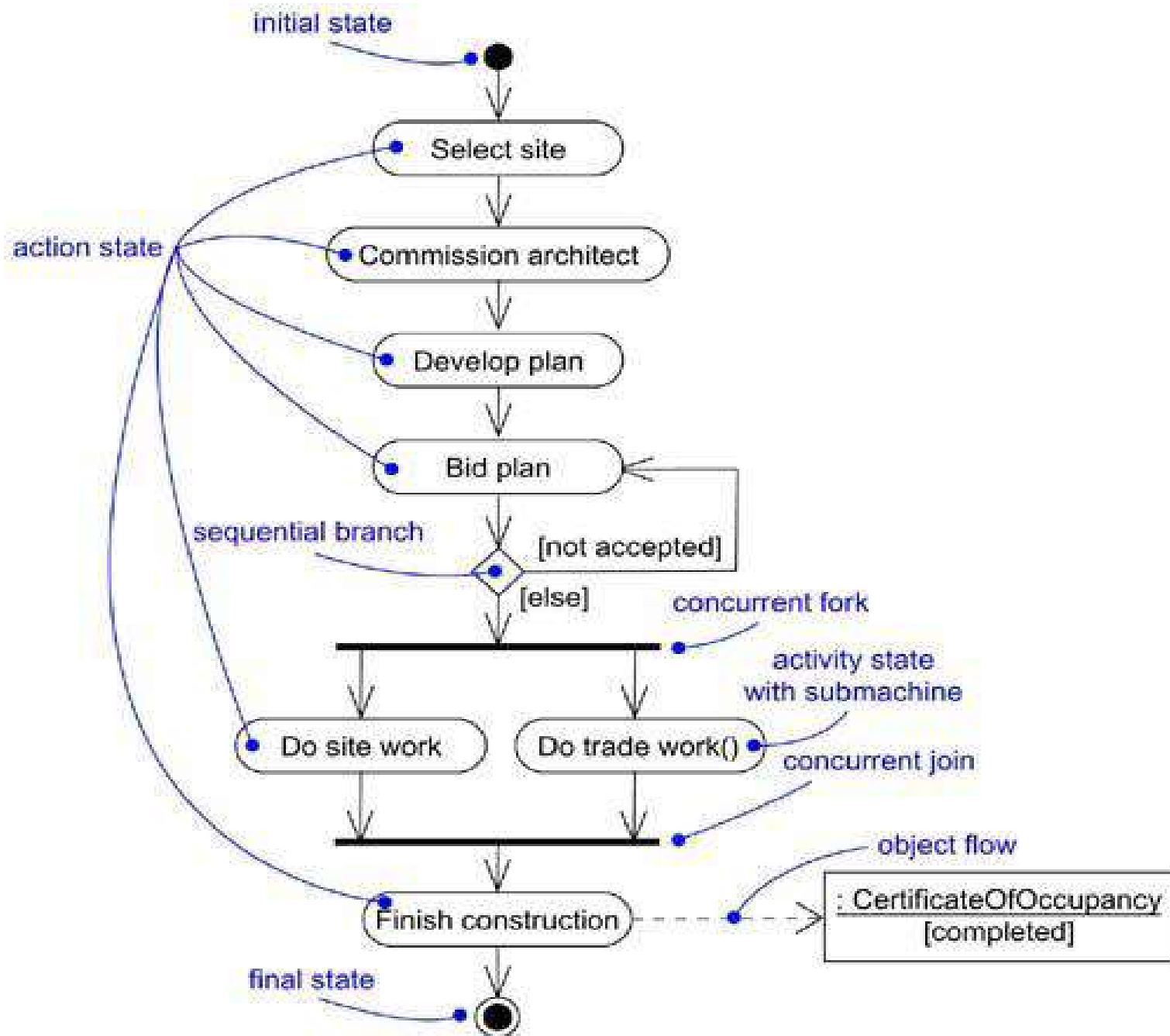
Fork & Join





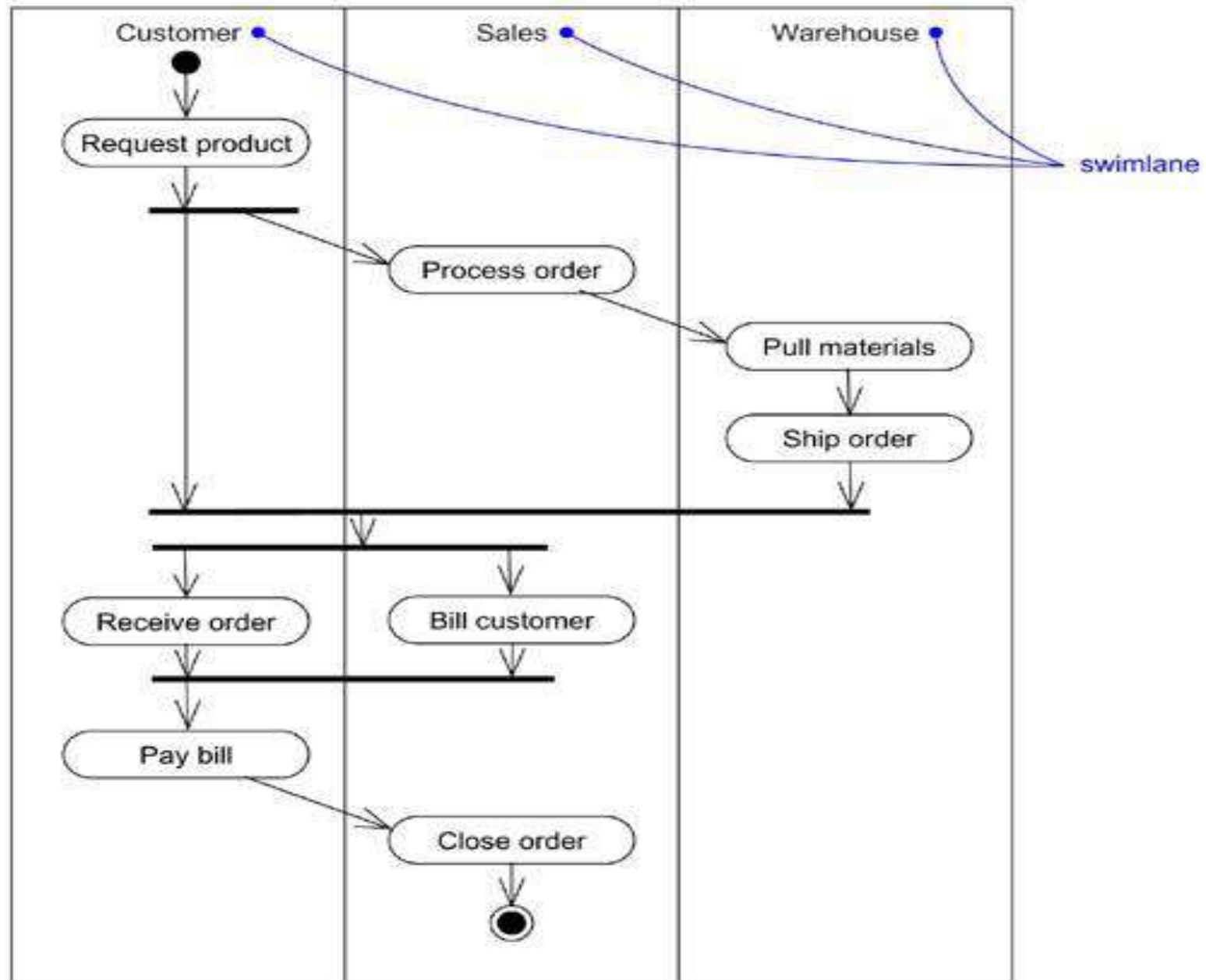
Forks, Joins & conditional Threads



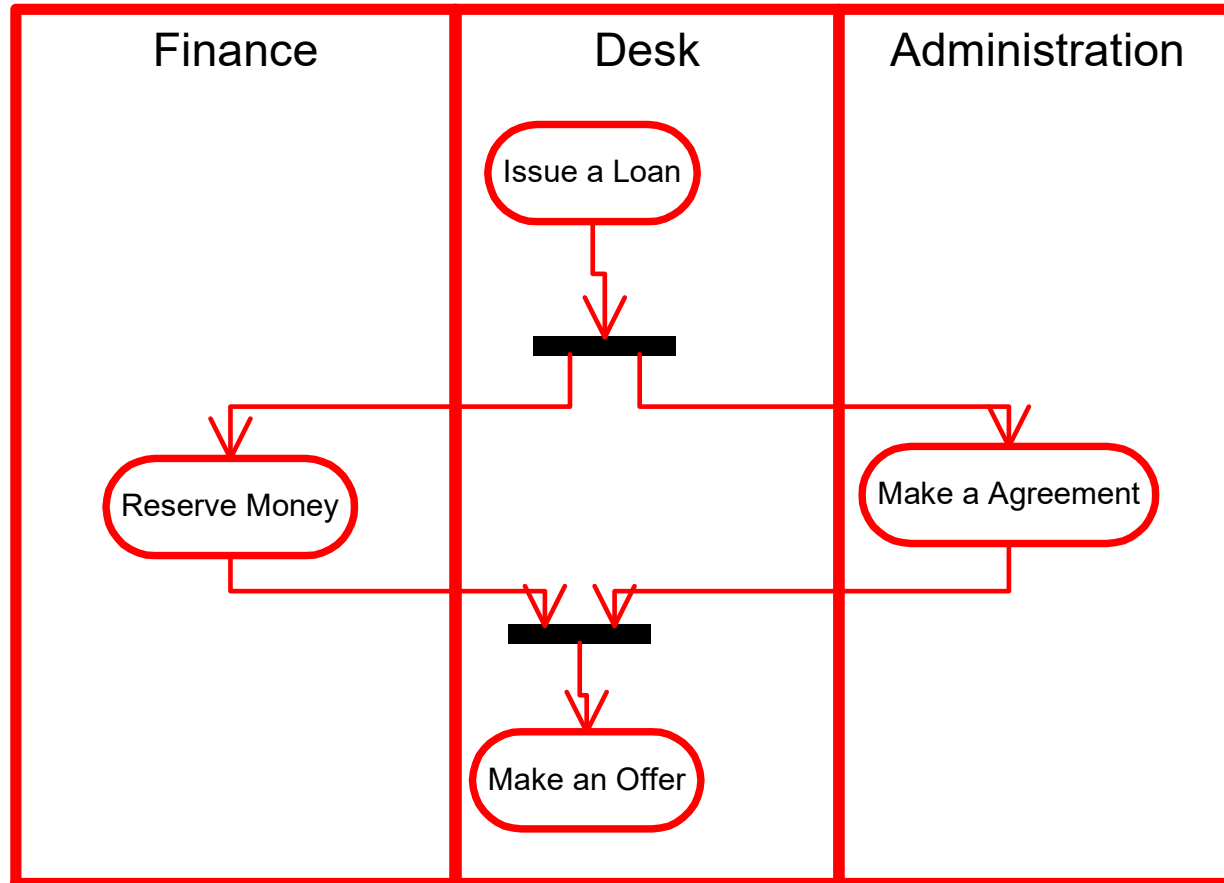


Swim lanes

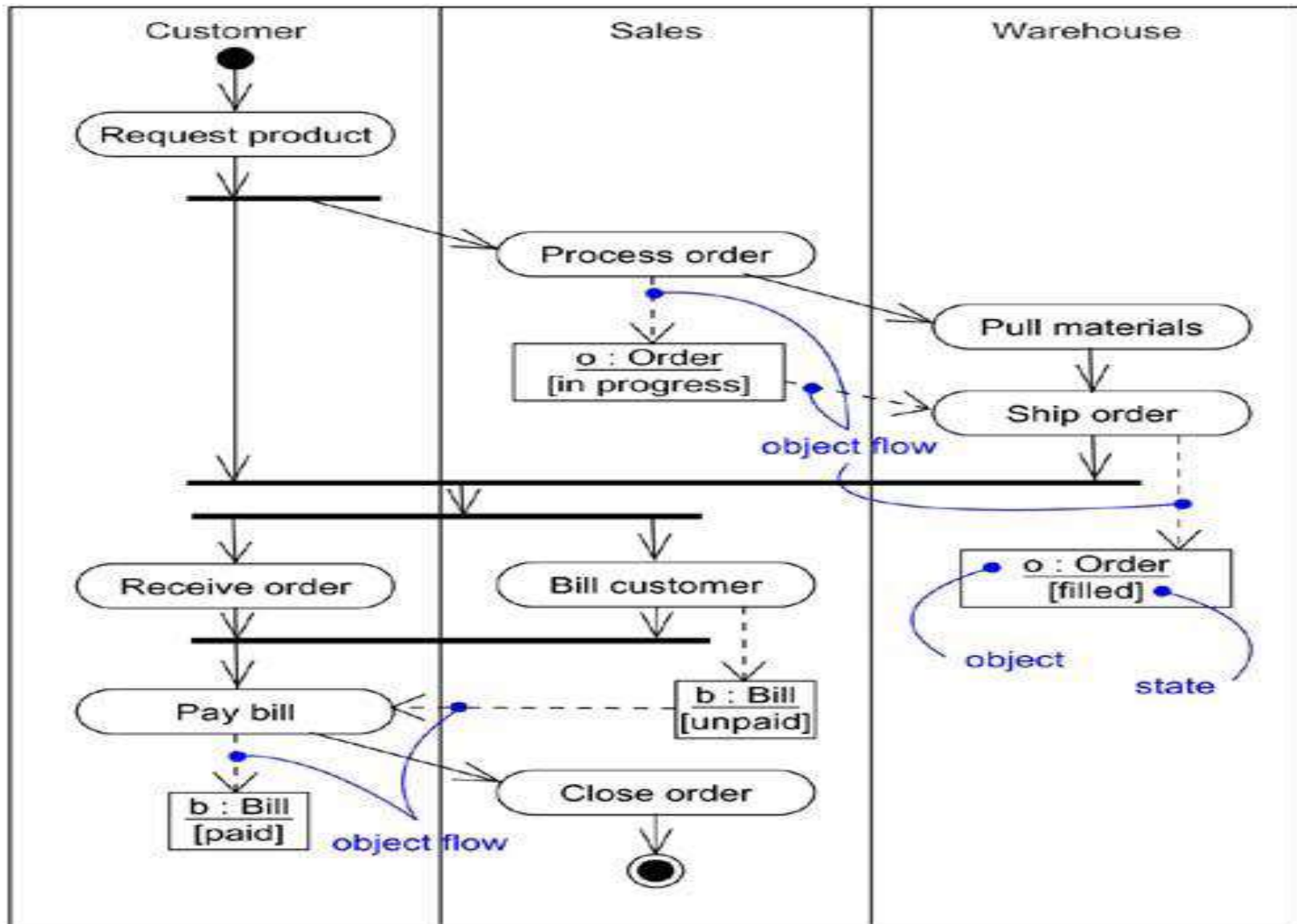
- The activities of an activity diagram may be performed by different groups.
- Each zone or lane represents the responsibilities of a particular group.
- If swimlanes are used, each activity can belong to one and only one swimlane.
- To use swimlanes, you must arrange your activity diagrams into vertical zones separated by dashed lines.

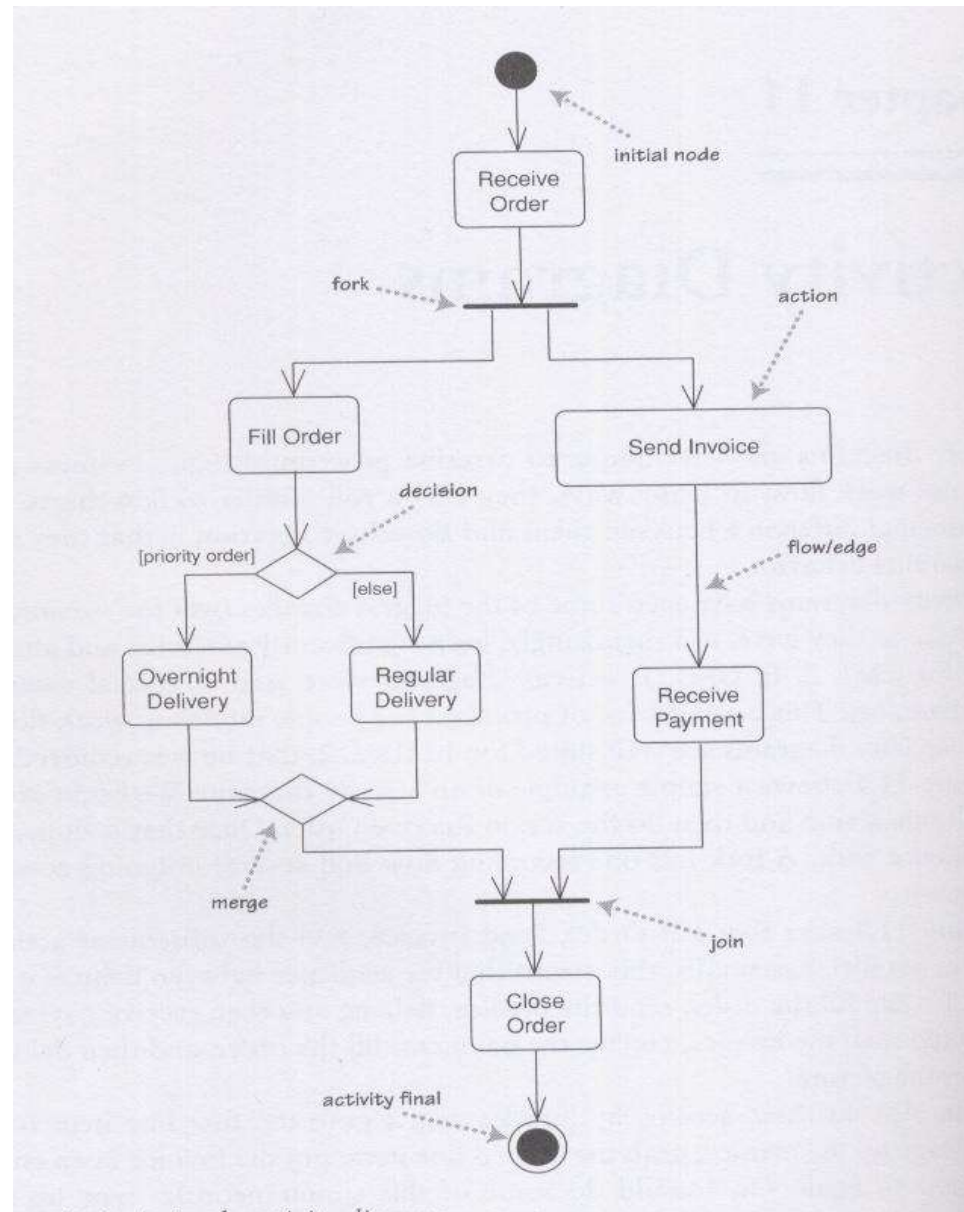


Swim lanes

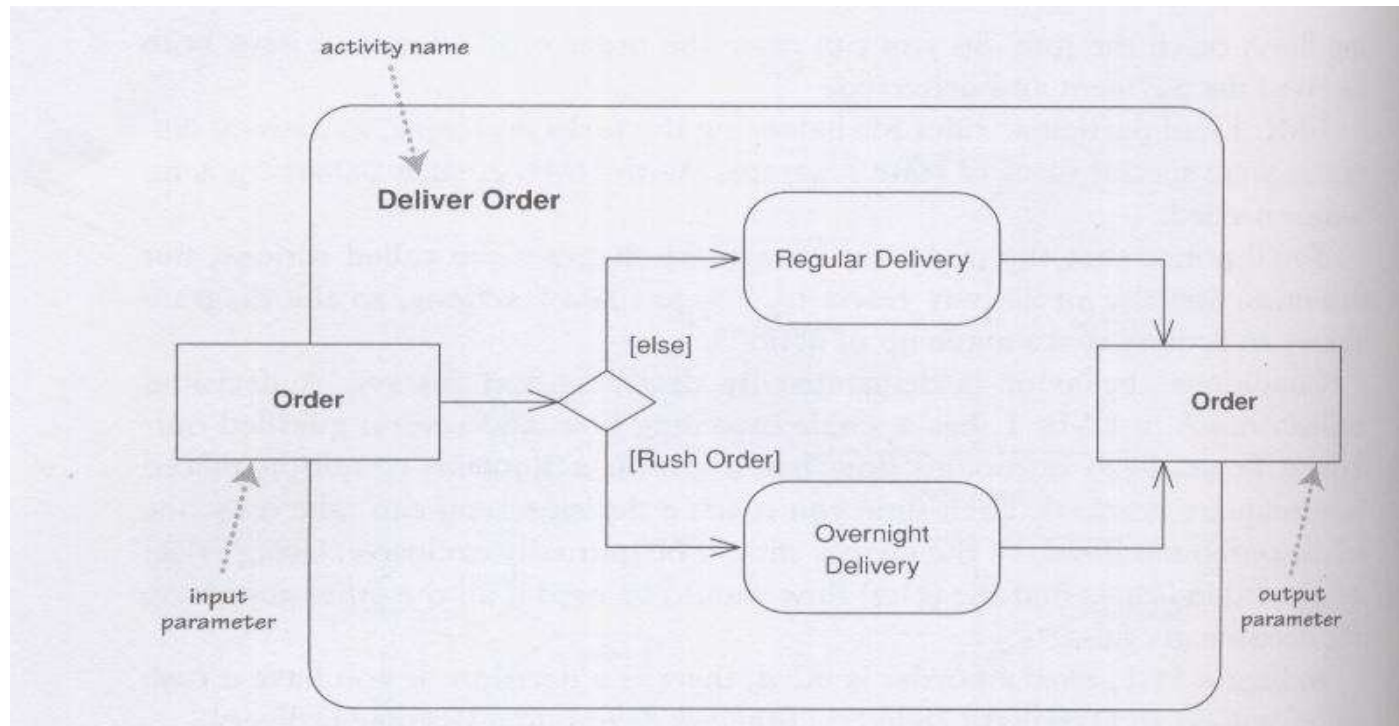


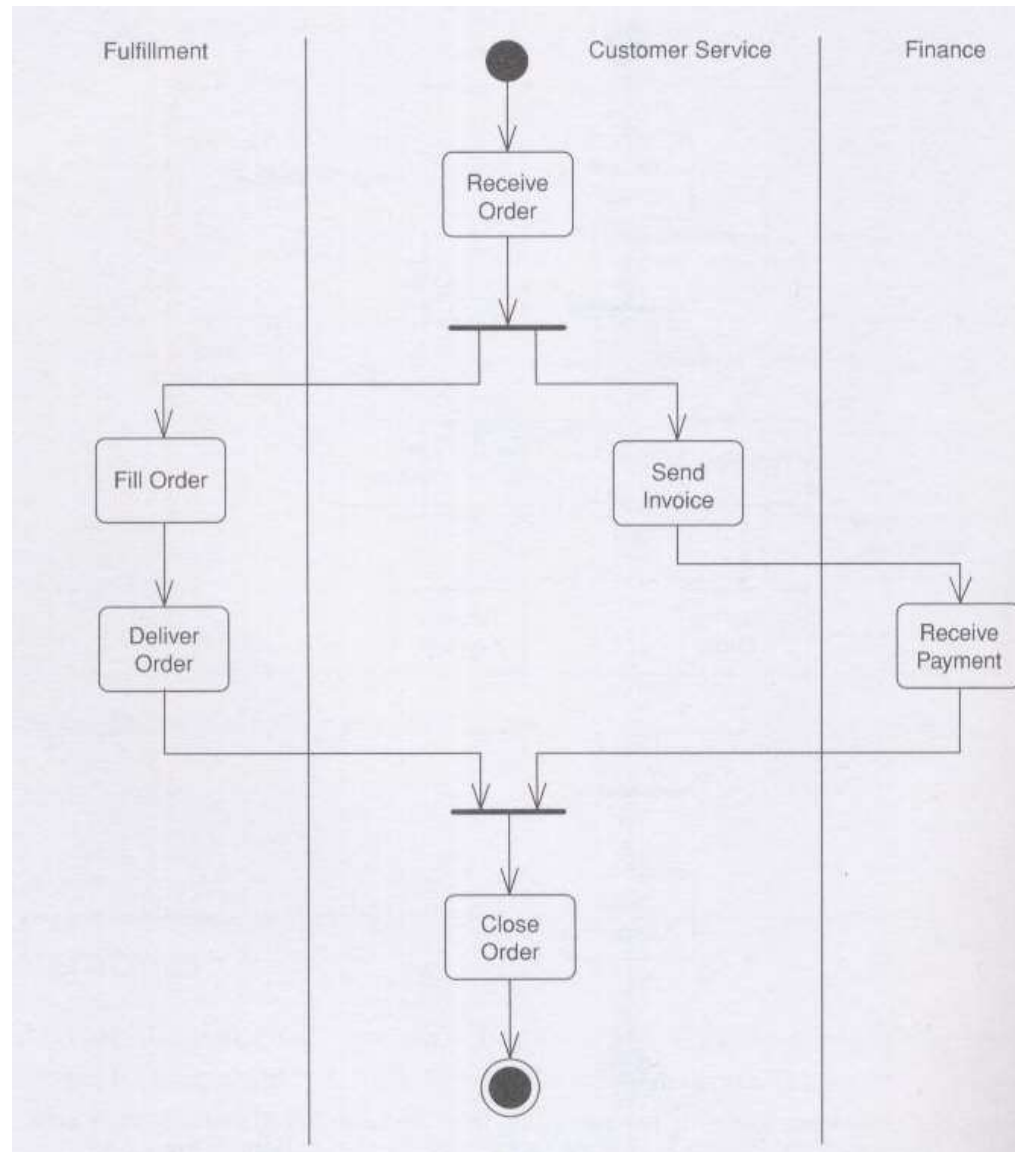
Object Flow

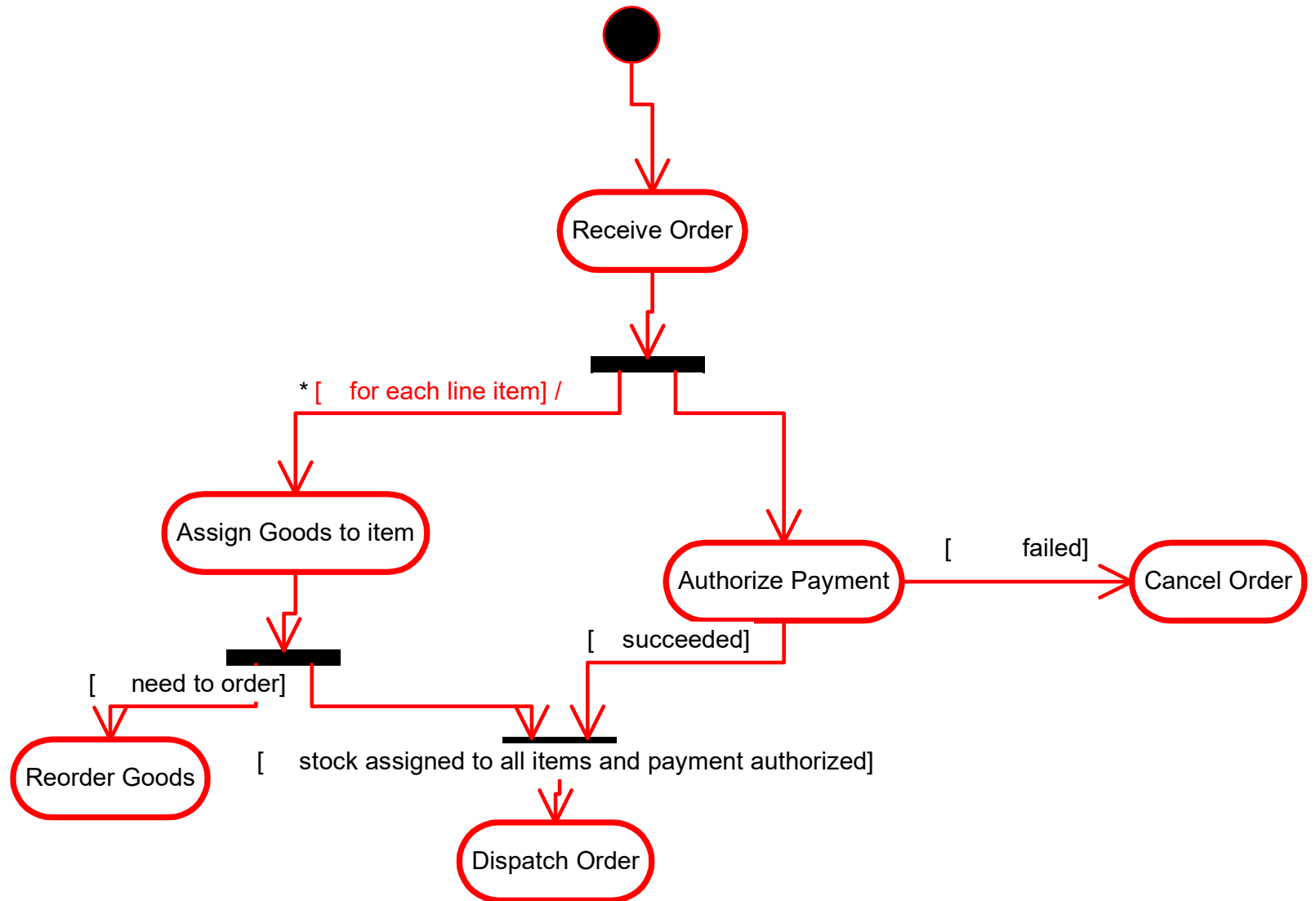




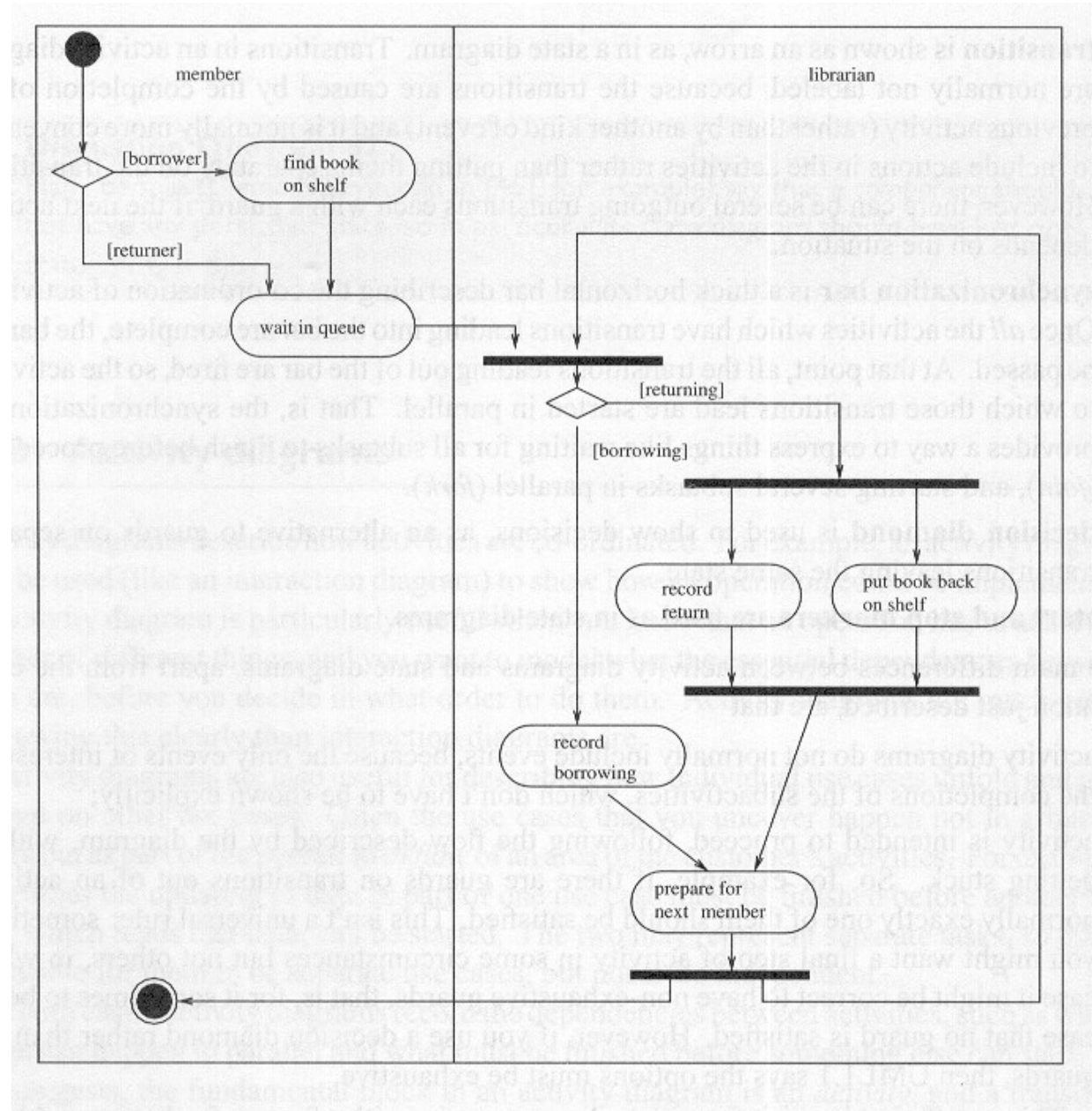
Composite Activity







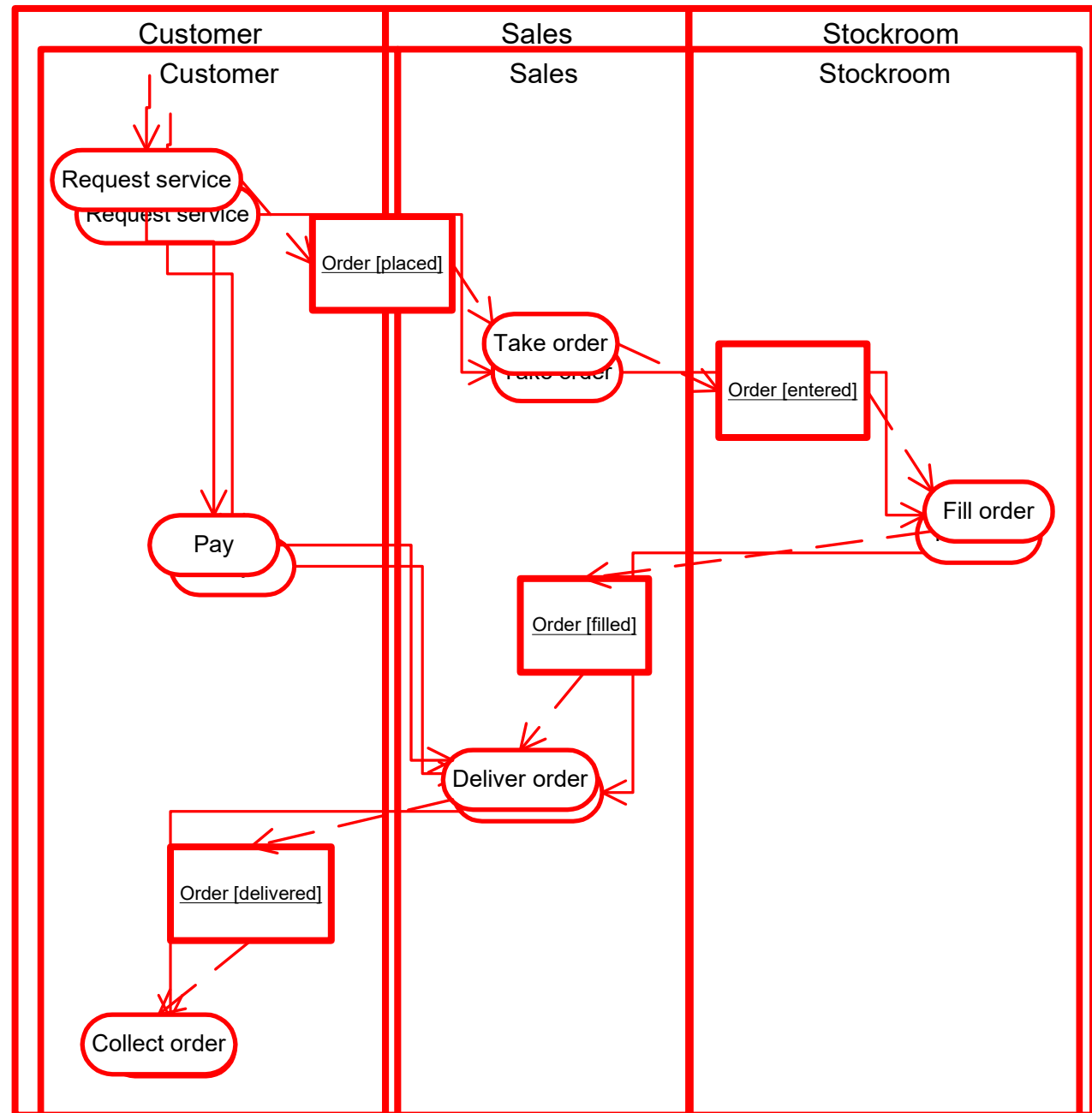
Example



Object Flow

- Objects may be involved in the flow of control associated with an activity.
 - These associations can be shown in activity diagrams by placing objects in the diagram and linking them to specific activities using a dependency link.
 - The same object may be the output of one action and the input of one or more subsequent activities.
 - To distinguish the various appearances of the same object, the state of the object at each point may be placed in brackets and appended to the name of the object.

Object Flow Example



Date2date example: scenario's

Use case

Log in Subscriber

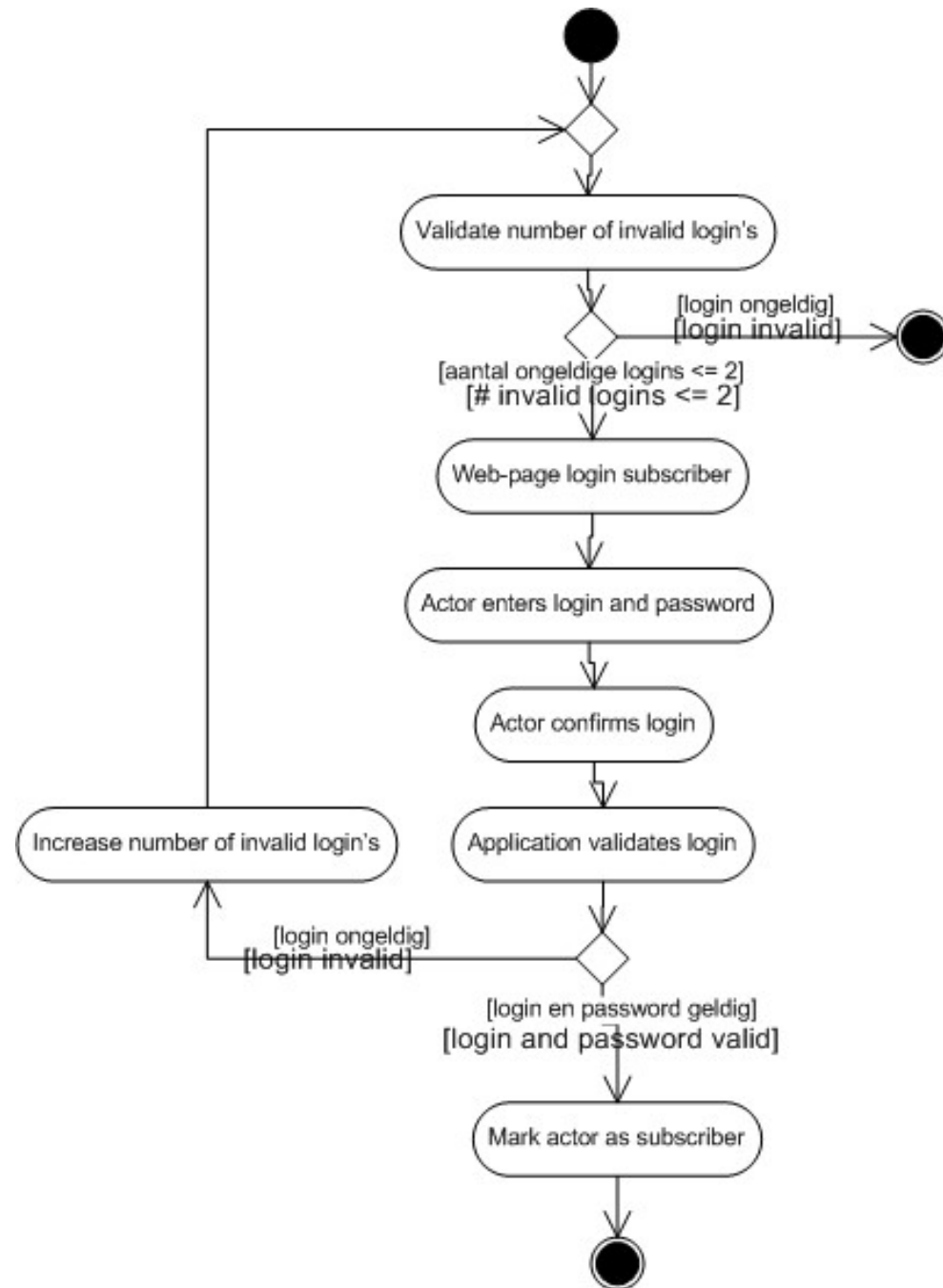
Step-by-step Plan

1. *Validate number of invalid login's*
2. *If number of invalid login's more than 2, stop*
3. *Show web-page*
4. *Actor enters login and password*
5. *Actor confirms login*
6. *Application validates login*
7. *If login is valid*
 - 7.1 *Actor is marked as subscriber*
 - 7.2 *Stop*
8. *If login is invalid*
 - 8.1 *Increase the number of steps*
 - 8.2 *Repeat from 1.*

Date2date : Log in Subscriber – use case text

Use Case	Log in subscriber
Actors	Subscriber, Visitor
Description	Check validity actor as subscriber
Pre-conditions	----
Basic flow	<ol style="list-style-type: none">1. Validate number of invalid login's2. Show web-page3. Actor enters login and password4. Actor confirms login5. Application validates login6. Actor is marked as subscriber
Extension /	
Exception flow	<ol style="list-style-type: none">1a Login is invalid 1a₁ Number of logins >2, stop5a Login is invalid 5a₁ Increase number of logins, repeat 1
Post-conditions	Actor is known as subscriber and actor is logged in

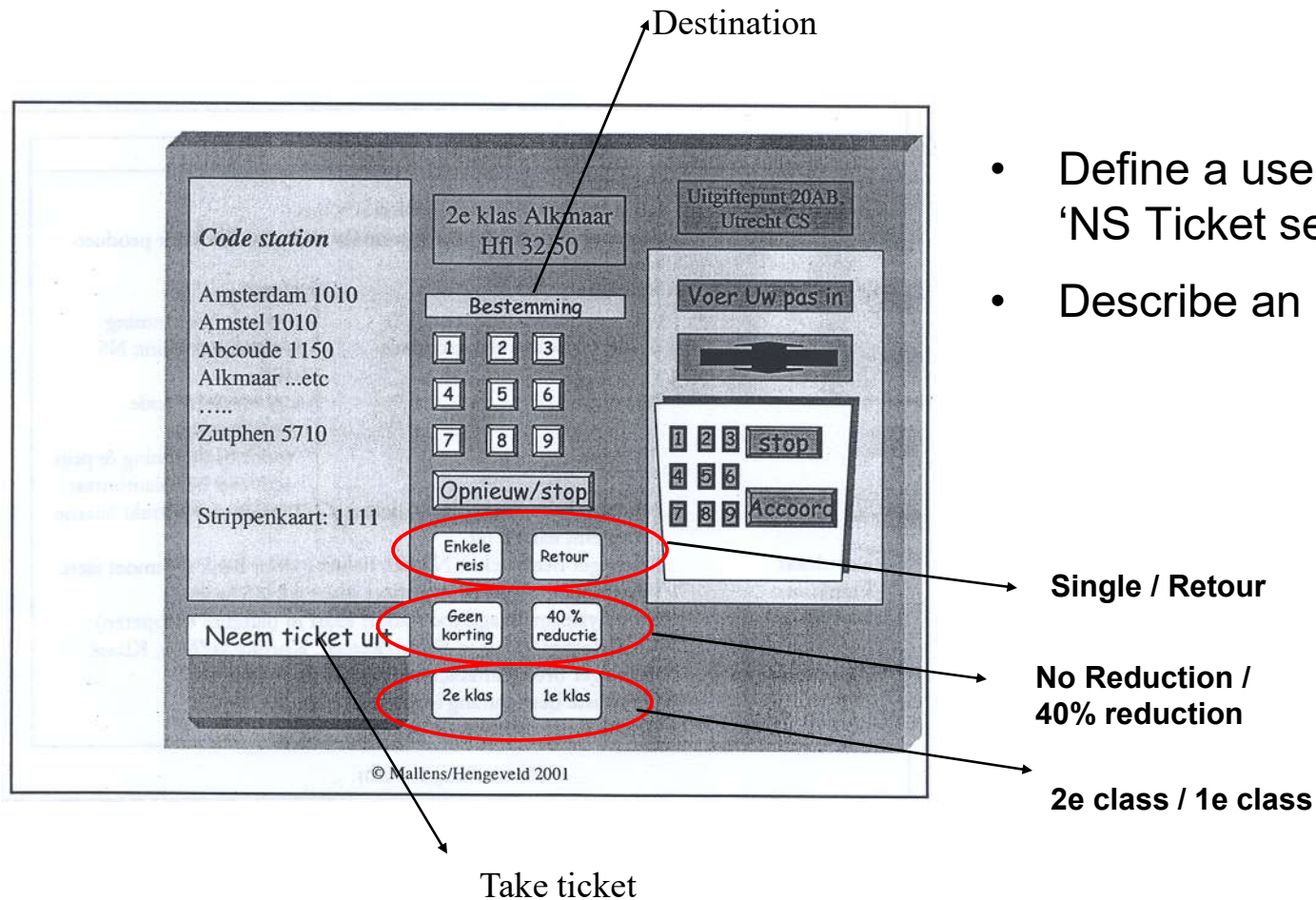
Date2date :
Log in
subscriber –
activity diagram



Key point

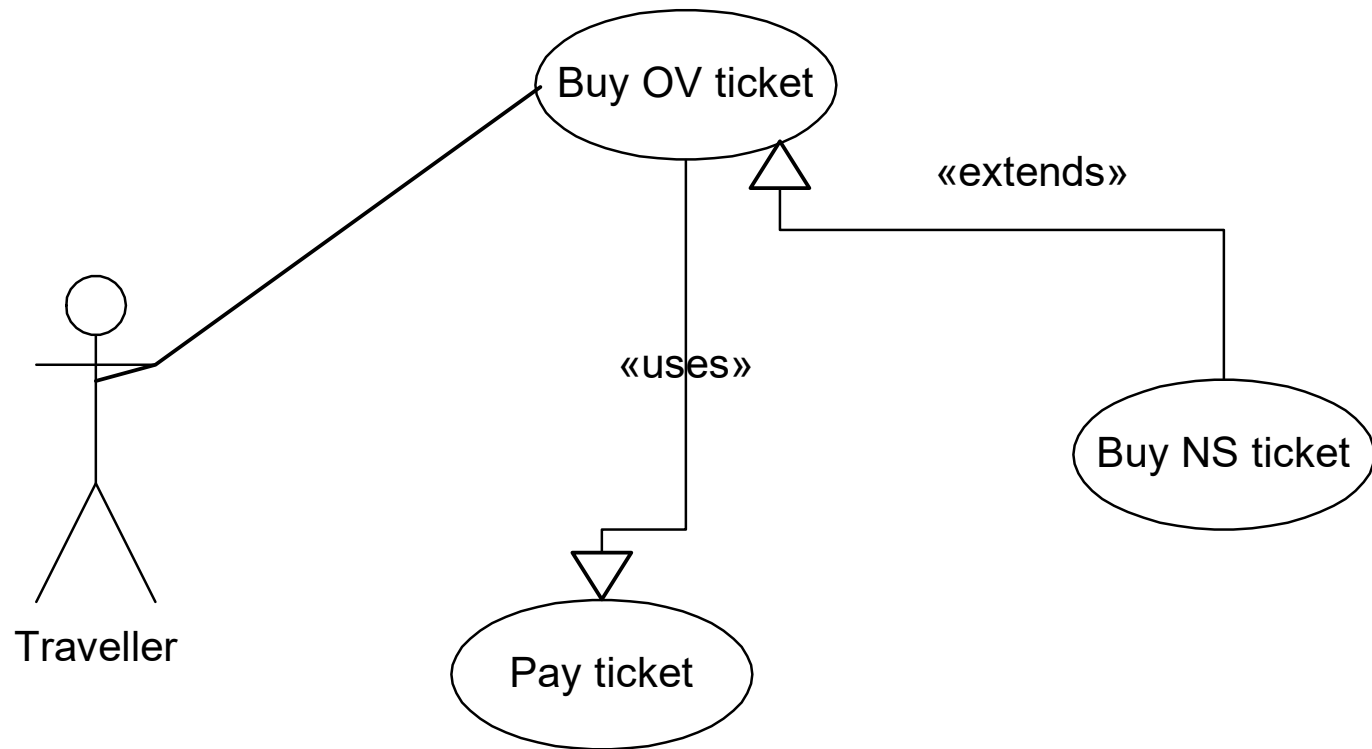
- Activity diagrams are useful for
 - Business Process Redesign
 - Work Flow Modelling
 - Analysing Use Cases

NS Ticket service



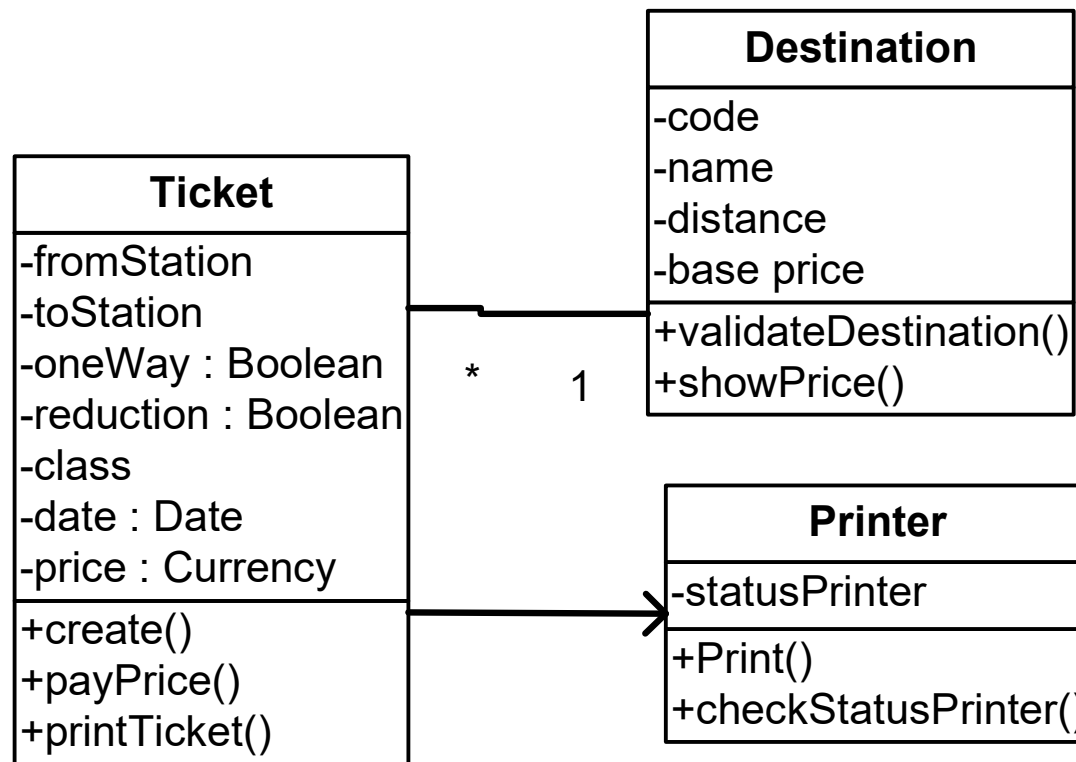
- Define a use case diagram of 'NS Ticket service'
- Describe an use case.

Use Case diagram 'NS Ticket service'



Use Case	Buy OV Ticket
Actors	Traveller
Preconditions	Traveller has a valid pass
Description	<ol style="list-style-type: none"> 1. Ticket device expects destination code 2. Traveller enters destination code 3. Extension point: NS ticket 4. Ticket device checks code and calculates the charge. Shows destination code & fare. Activates ticket machine for paying 5. Traveller pays (use case: Pay ticket) 6. Ticket device print and supplies ticket 7. Traveller takes ticket
Extension	Destination code = NS station. 3a. Ticket device expects ticket type 3b. Traveller enters Single/Return, Discount Y/N, Class
Exceptions	Traveller interrupt the interaction or walk away Traveller enters an incorrect destination code Payment is not finished off successful
Result	Traveller has ticket. (NS can look forward to the payment)

Make a class diagram of 'NS ticket service'
including the classes 'Ticket' and
'Destination'



Make an activity diagram for the actor 'Traveller'

