



# 物件導向系統分析與設計

## Object Oriented Analysis and Design

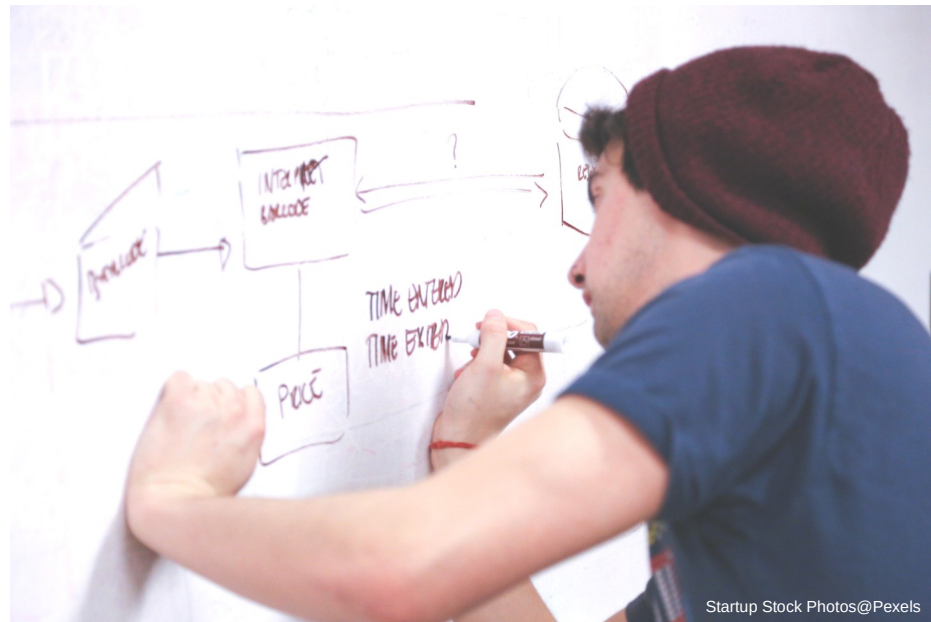
### Activity Diagram

劉儒斌 Paladin R. Liu

paladin@ntub.edu.tw

# AGENDA

- 活動圖簡介
- 活動圖圖示
- 如何繪製活動圖

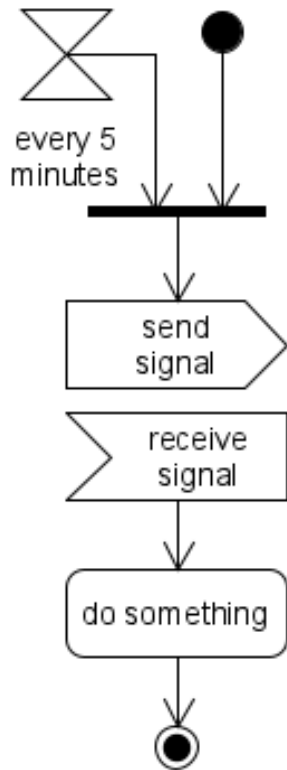


# 活動圖簡介

- 使用案例圖用來說明系統該做些什麼
- 活動圖用來描述系統應該如何完成這些目標
- 活動圖顯示了鏈接在一起的活動 / 操作，以表示系統中發生的一個過程
  - 例如，可以使用活動圖來描述建立 Blog 帳戶所需要的步驟

# 活動圖簡介

- 由節點 (Node) 及邊 (Edge) 所構成
- 節點表示活動圖裡的…
  - 動作、控制、或物件
- 邊表示活動的流向
  - 指示流程的程序 / 步驟



# 活動圖圖示

# 活動圖圖示

- **Start / Initial Node**

- 黑色的實心圓點
- 代表活動 / 流程的開始



Initial Node

- **End / Final Node**

- 空心圓中，帶有實心圓點
- 代表活動 / 流程的結束



Final Node

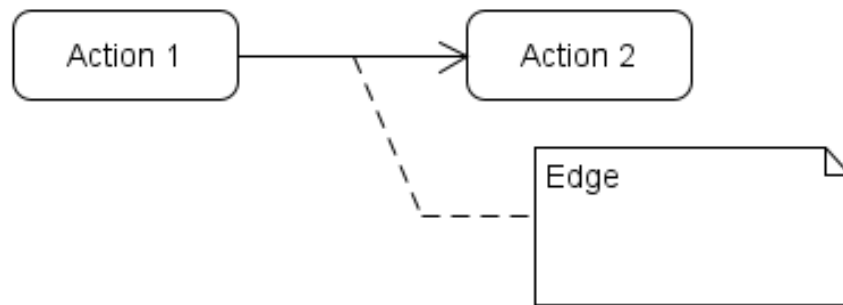
# 活動圖圖示

- **Action**

- 帶有圓角的長方形
- 代表流程中發生的活動

- **Edge**

- 帶有箭頭的實心直線
- 代表流程中的活動流向

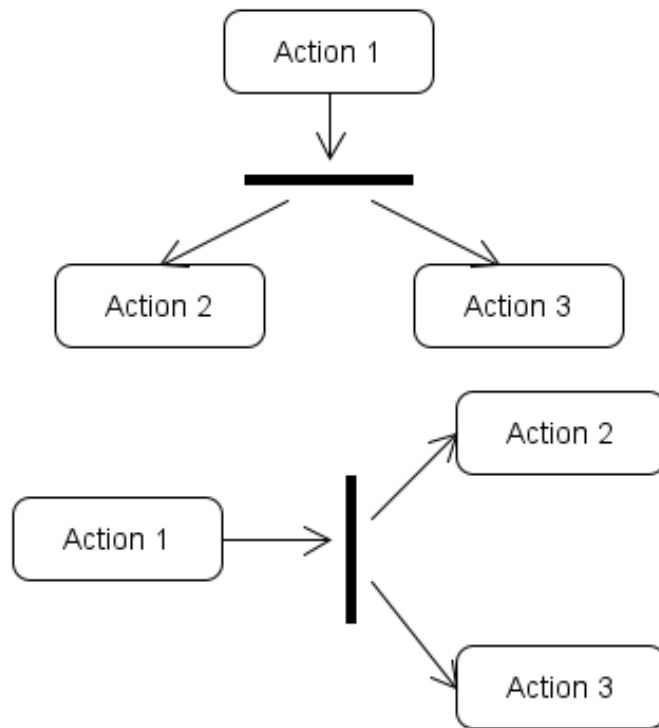




# 活動圖圖示

- 分岔 (Fork)

- 以一個黑色實體的長條圖形 (Bar) 表示
- 表示接下來的動作，會分為兩個或以上的動作，同時進行。
- 與 Join 搭配使用

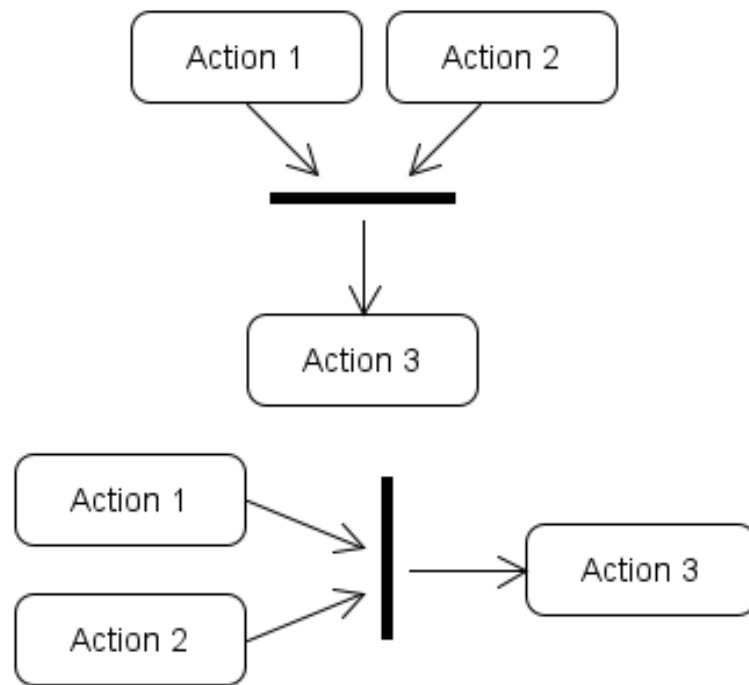




# 活動圖圖示

- 收合 (Join)

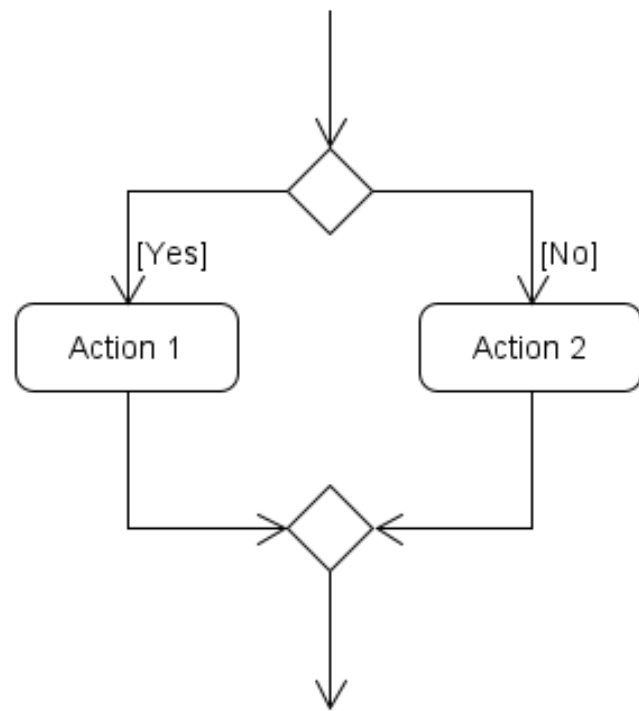
- 以一個黑色實體的長條圖形 (Bar) 表示
- 表示先前的動作，需要全部完成了，接下來的動作才會被執行
- 與 Fork 搭配使用



# 活動圖圖示

- 決策 (Decision) / 合併 (Merge)

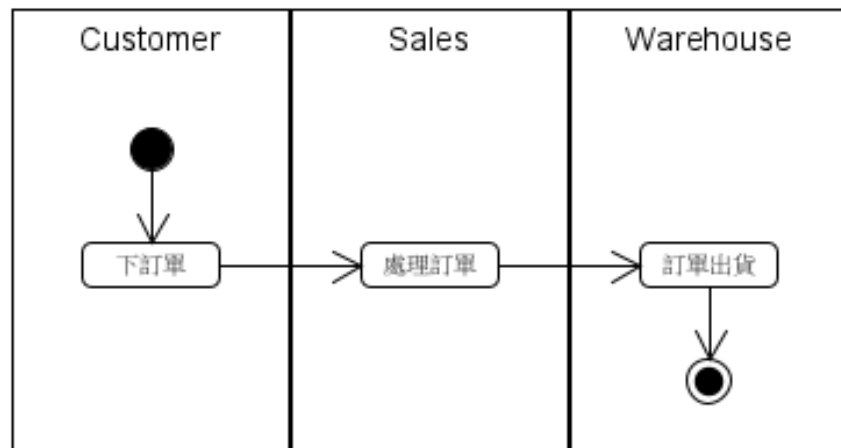
- 以空心的菱形表示
- 當一個流入、多個流出時，稱為決策
- 當多個流入、一個流出時，稱為合併



# 活動圖圖示

- 水道 (Swimlane)

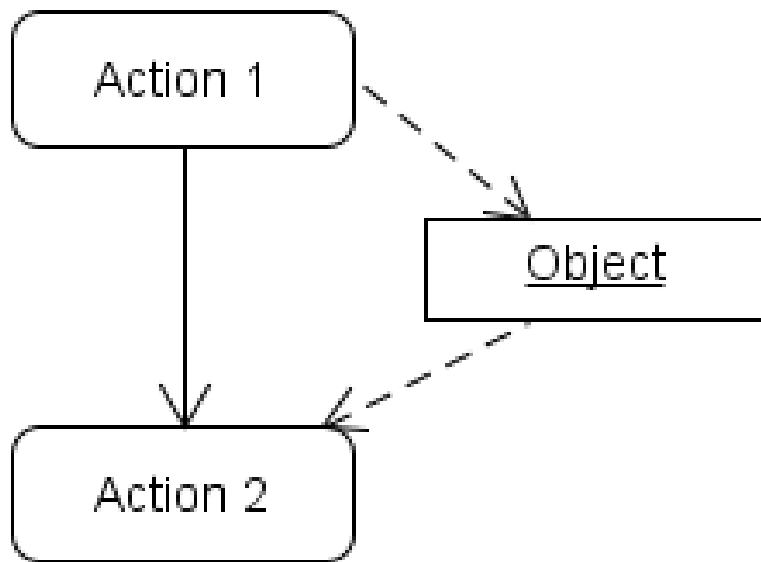
- 有如游泳競賽中的水道
- 把流程中的活動，依照 Actor 或企業物件 (Business Object) 來分類
- 塑模時，可以更容易清楚的表達與企業流程相關的概念



# 活動圖圖示

- 物件 (Object)

- 以類似於「類別」的圖形表示
  - ▶ 虛線箭頭代表物件流 (Object Flow)
- 代表 Action 的輸入 / 輸出資料
- 不一定是系統裡的 Object
  - ▶ 如果是描述一般的作業流程，也可以是作業裡的工作表單



# 活動圖圖示

- **Input Pin**

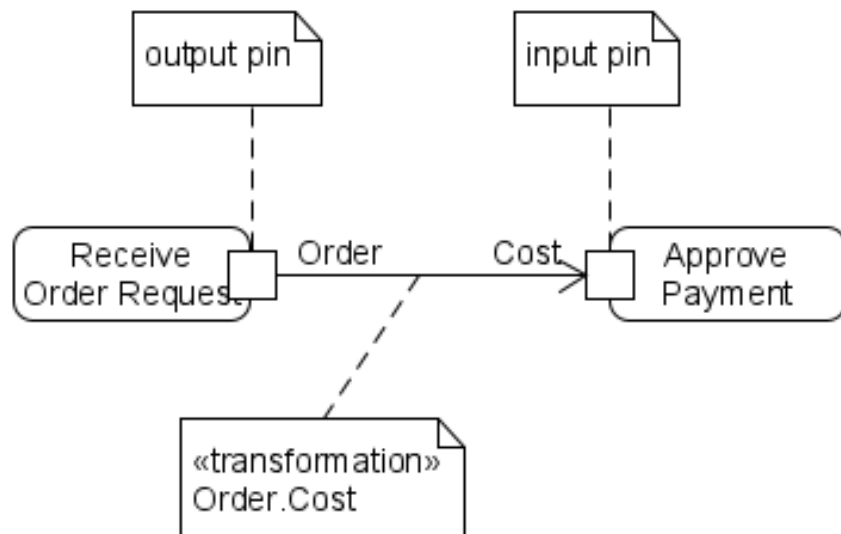
- 輸入指定的物件到 Action 中

- **Output Pin**

- 從 Action 中輸出指定的物件

- **«transformation»**

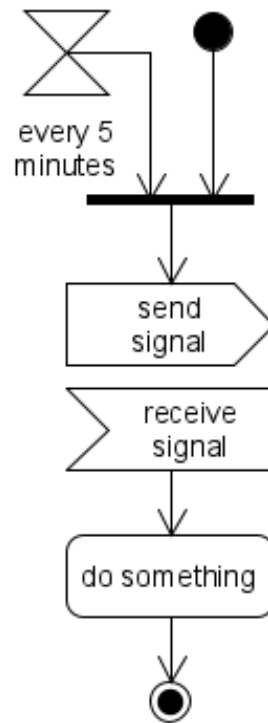
- 有時並不是整個物件輸出到下一個 Action ；透過 «transformation» 來指定哪些部份是必要的



# 活動圖圖示

- 訊號

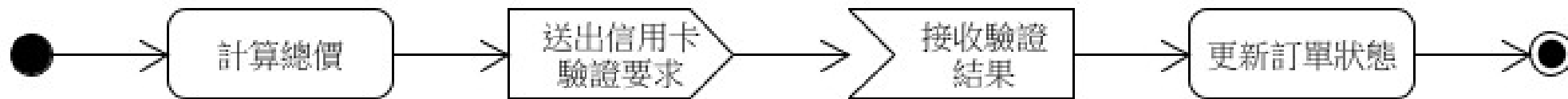
- 送出訊號
- 接收訊號
- 時間訊號
  - ▶ 時間間隔
  - ▶ 指定時間或日期



# 活動圖圖示

- 範例：信用卡驗證

- 收到訂單並計算總價後，送出信用卡驗證要求
- 接收驗證結果，更新訂單狀態





# 活動圖圖示

- 中斷

- 帶有箭頭的閃電線條
- 表示活動執行中產生中斷後活動之流向

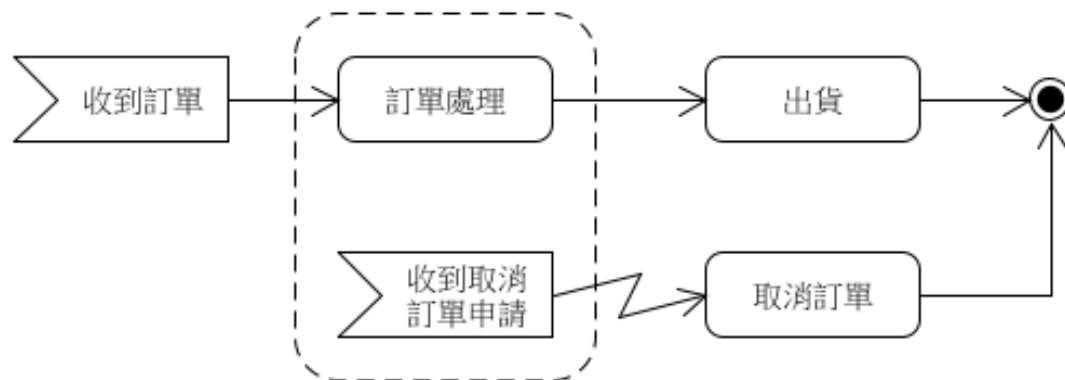
- 可中斷區

- 以虛線圓角框表示
- 表示活動過程中，哪一個區域所含之動作可能被中斷

# 活動圖圖示

- 範例：中斷訂單處理

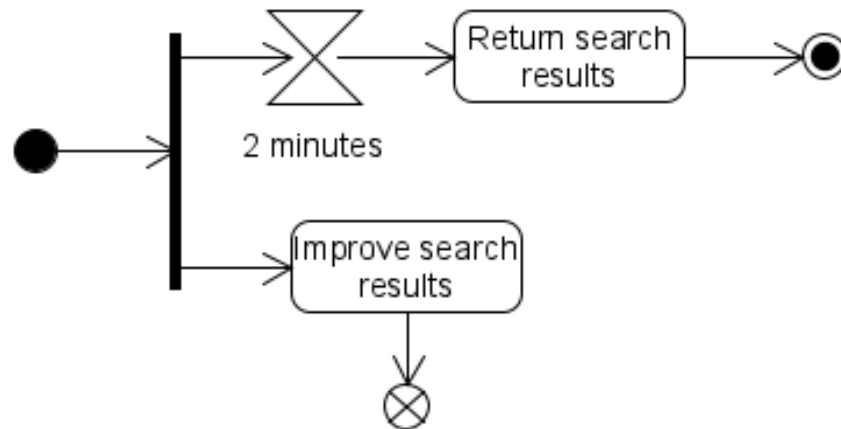
- 處理訂單的活動可能需要比較長的作業時間
- 過程中可能遇到客戶取消訂單



# 活動圖圖示

- 流程中止

- 用來表示流程結束
- 如果是在 Fork 之後出現，其他路徑的流程還會繼續執行



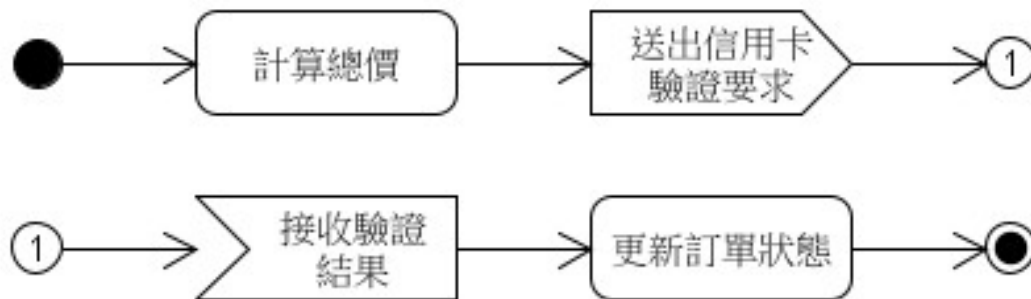
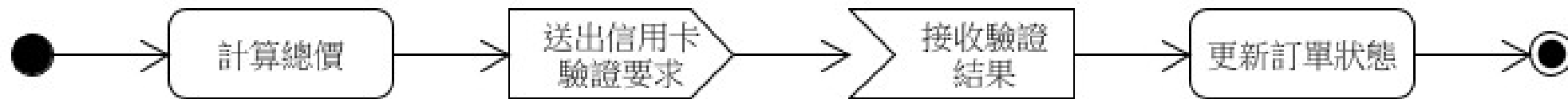
# 活動圖圖示

- **連結點**

- 有時圖形太複雜，互相交錯
- 有時圖形太長，不容易呈現與閱讀
- 以一個空心圓形表示，中間帶有數字或符號

# 活動圖圖示

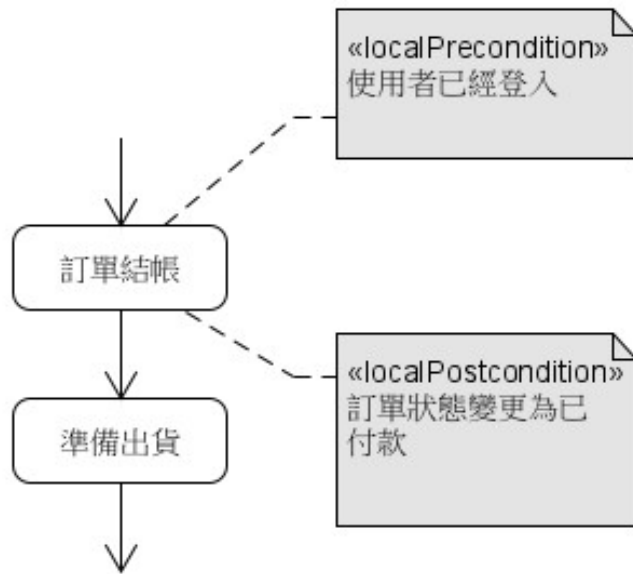
- 連結點範例



# 活動圖圖示

- 前置 / 後置條件

- 使用 stereotype
- «localPrecondition» :  
在執行動作前進行檢查
- «localPostcondition» :  
在執行動作前進行檢查

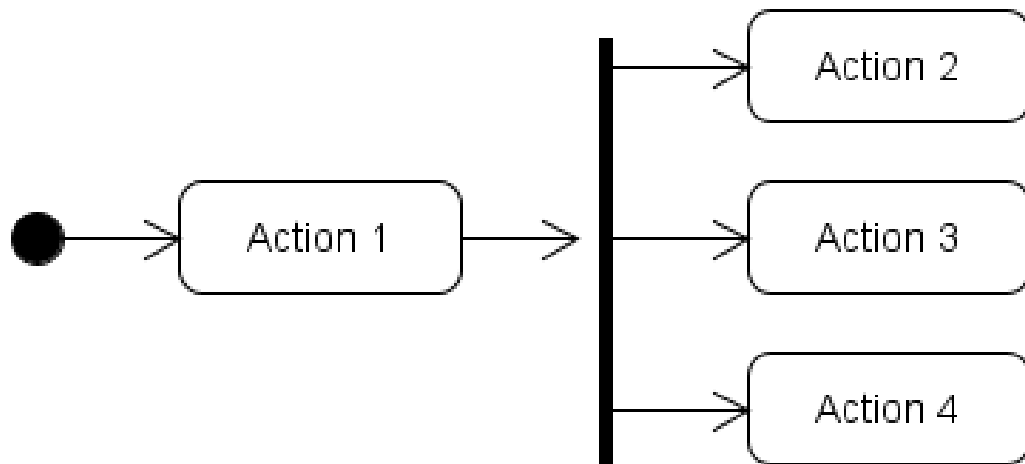


# 如何繪製活動圖

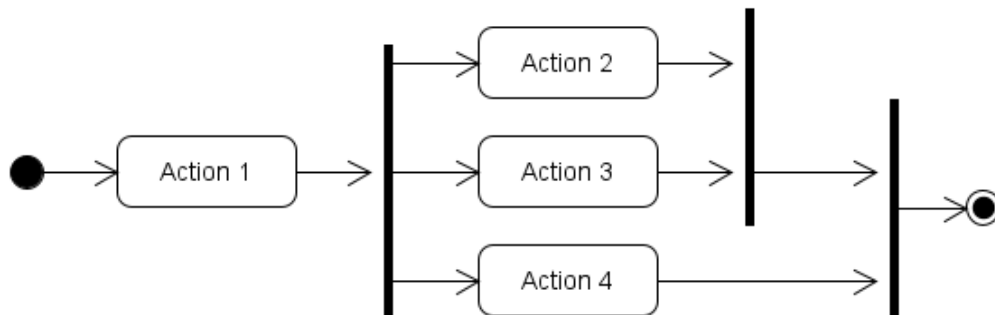
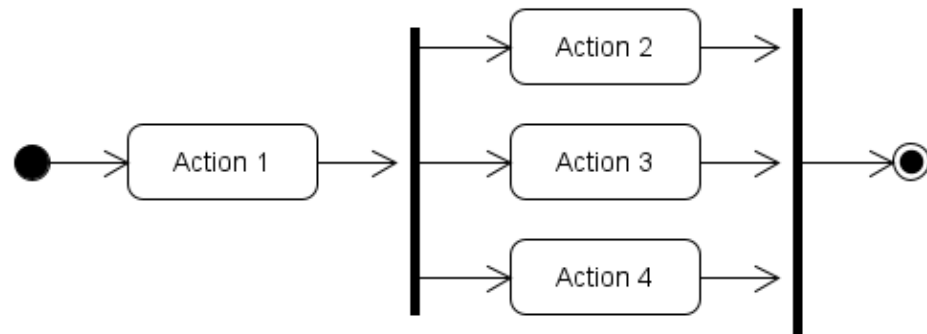


# 如何繪製活動圖

- 有 Fork 就會有 Join

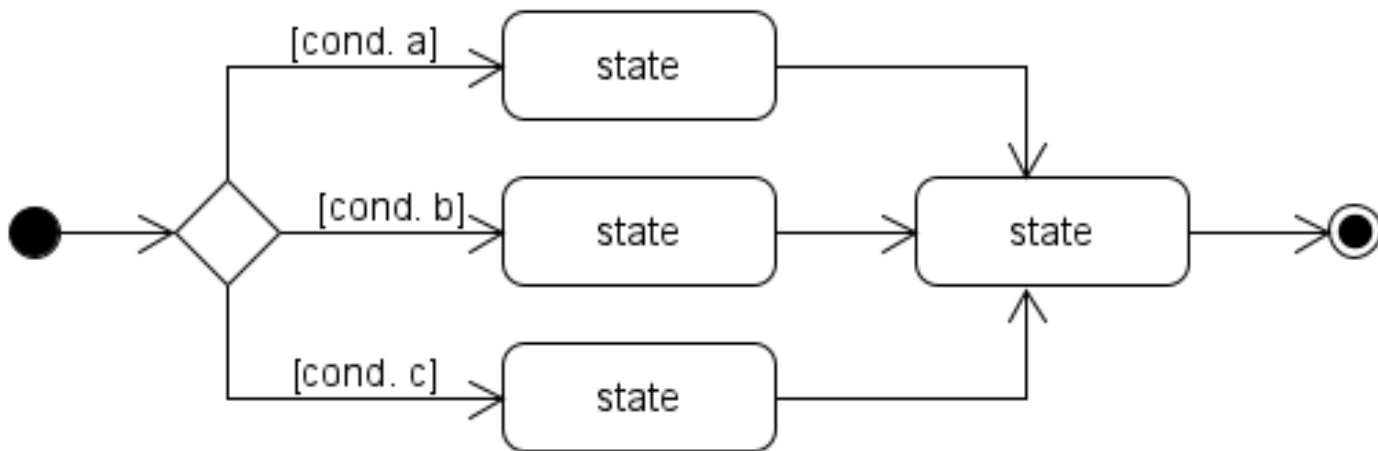


# 如何繪製活動圖

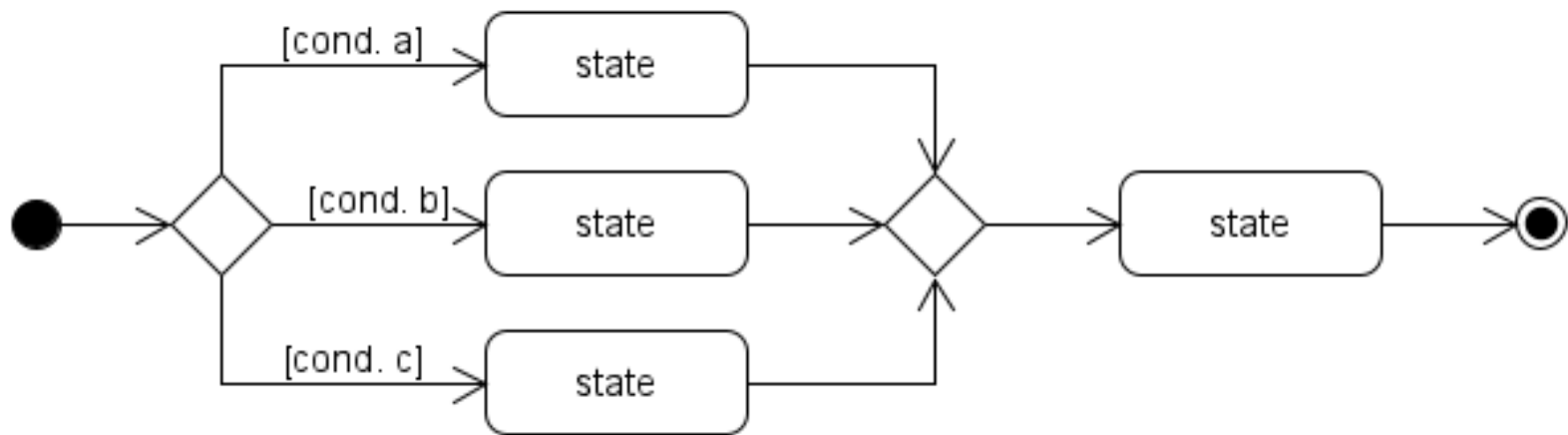


# 如何繪製活動圖

- 使用 Merge 爲 Branch 收攏
  - 標記判斷式的結束

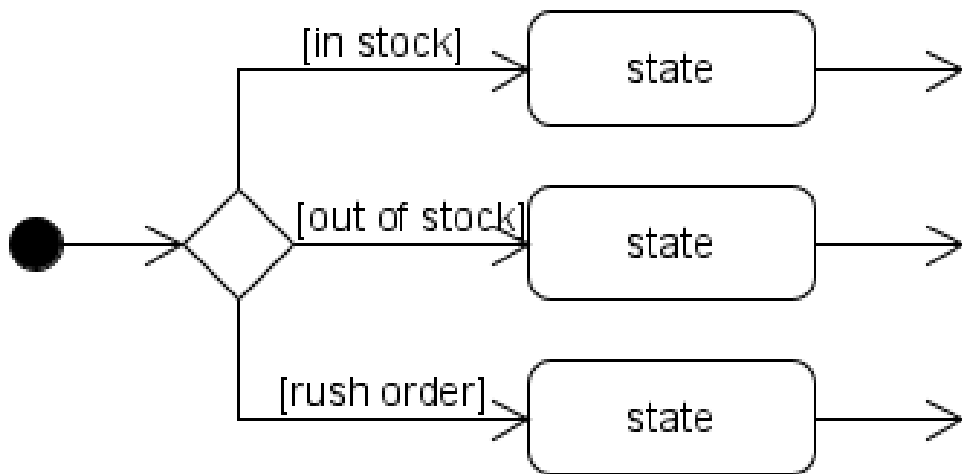


# 如何繪製活動圖



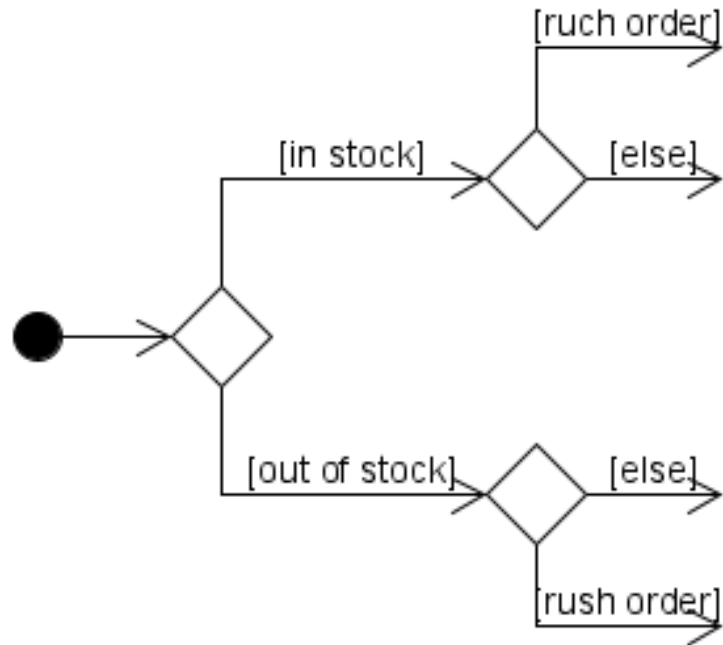
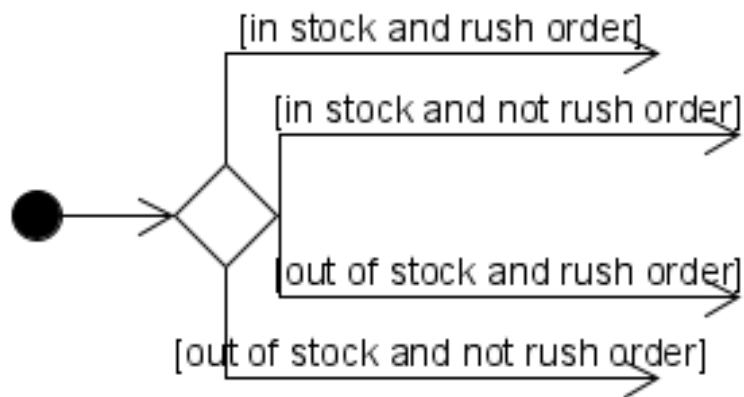
# 如何繪製活動圖

- 如果商品在庫，而且是急單，要走哪一條？？



# 如何繪製活動圖

- 這樣或許會更好…



# 如何繪製活動圖

- 從企業流程的面向
  - 從 Actor 的觀點，看 Actor 與系統如何互動
  - 從較高的層次，來描述與系統相關的企業流程
  - 識別重要的物件流

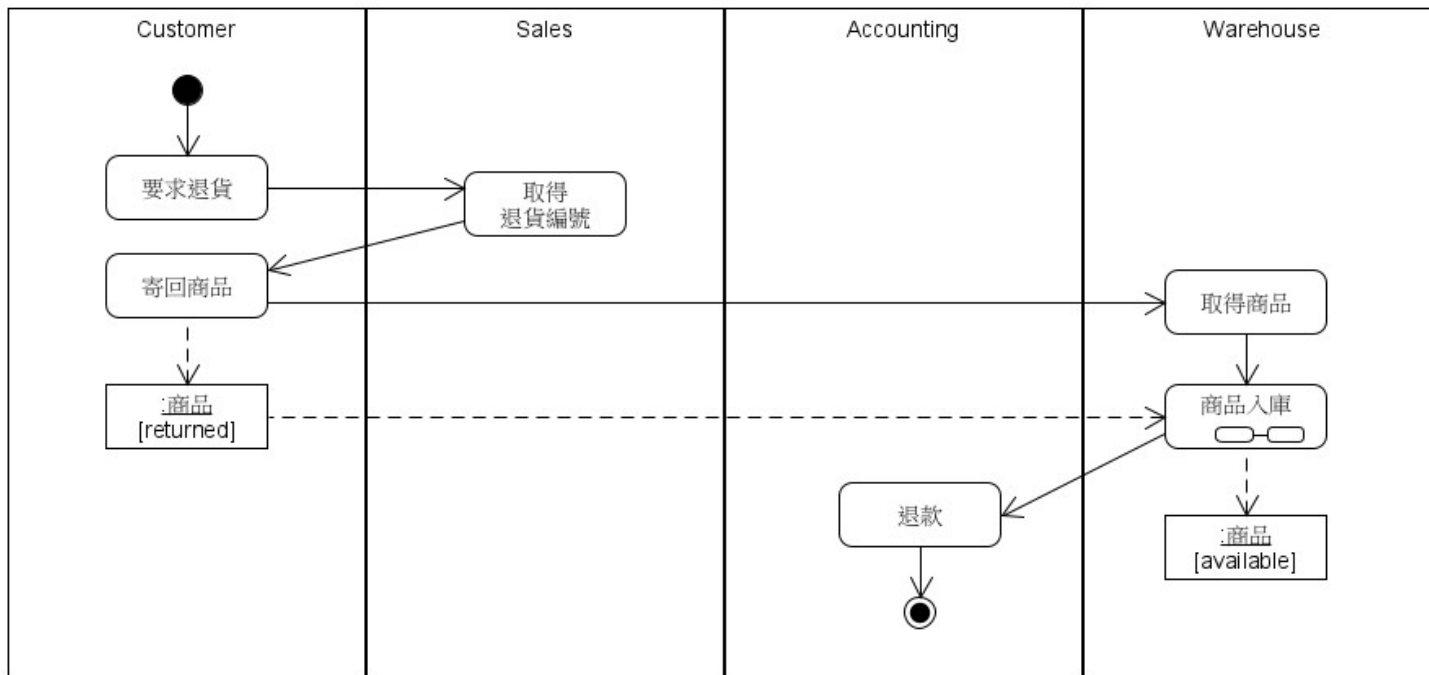


# 如何繪製活動圖

- 專注在企業流程之上
- 找出企業物件 (Business Object) ，並為它們建立 Swimlane
  - 領域問題中的詞彙，通常是名詞
- 識別流程啓始的前置條件，以及結束時的後置條件
  - 更好的描繪流程的邊界在哪裡
- 把重覆出現的部份流程，收合為一個活動
  - 再用另一張活動圖來描述這個部份的動作
- 如果在作業流程中存在重要的物件，就必需把它們也畫入活動圖中
  - 使用物件流來描述

# 如何繪製活動圖

- 範例：銷貨退貨 – 客戶訂購商品後，進行退貨的一連串流程



# 如何繪製活動圖

- 從操作的面向
  - 當成流程圖使用
  - 描述操作中的步驟、決策 / 判斷、分支、合併等
  - 說明執行操作所需提供的參數條件

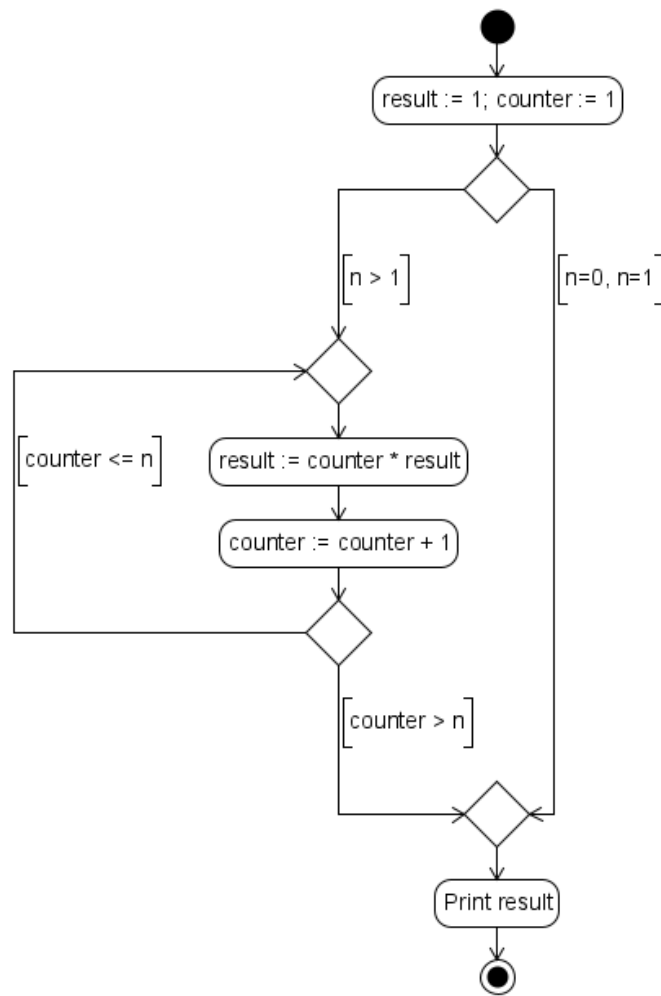
# 如何繪製活動圖

- 試著捕捉操作中的概念
  - 參數、特徵、相鄰的類別等
- 識別操作執行的前置及後置條件
- 從操作的初始狀態開始，定義相關的活動
  - 以及隨時間推移所發生的狀態改變
- 必要時使用分支，以表達有條件的路徑及迴圈

# 如何繪製活動圖

- 範例：計算  $N$  階層
  - $N=0$  or  $N=1$ 
    - ▶  $\text{Result} = 1$
  - $N > 1$ 
    - ▶  $\text{Result} = 1 * 2 * \dots * N$

Calculate n!



# 綜合練習

- **REQ-A001**：管理員可以在 CMS 中建立 Blog 帳號，同時使用作者證書資料庫驗證作者的詳細資訊。

# 綜合練習

- STEP 1：找出 Actor 及 Use Case





# 綜合練習

## • STEP 2：撰寫使用案例描述

主要角色

管理員

次要角色

作者證書資料庫

觸發事件

管理者要求 CMS 建立新的 Blog 帳號。

主要流程

步驟

動作

1

管理員要求系統建立一個新的 Blog 帳號。

2

管理員選取一種帳號分類。

3

管理員輸入作者的詳細資訊。

4

作者的詳細資料通過證書資料庫的驗證。

5

新 Blog 帳號已建立。

6

新 Blog 帳號的明細資訊以 Email 的方式寄送並通知作者。

延伸流程

步驟

分支動作

4.1

作者的詳細資料並未通過證書資料庫的驗證。

4.2

新 Blog 帳號建立被拒絕。

# 綜合練習

- STEP 3：繪製活動圖

