

Object-**O**riented **A**nalysis and **D**esign

U-03 Object Behavior Modeling

Neter Chao Ph.D.

內容大綱

- 導論
- 物件互動行為塑模工作
- 循序圖
- 合作圖
- 結論

導 論

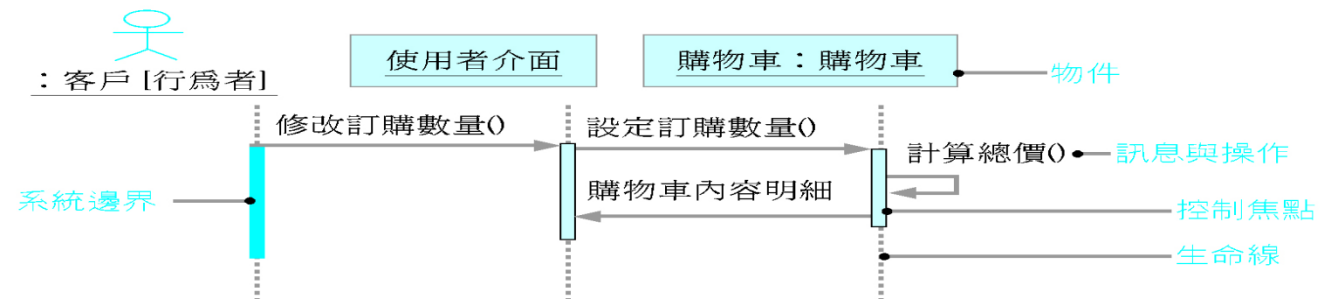
- 完成了使用個案（或稱需求）塑模後，便可進行物件結構塑模、物件互動行為塑模等活動以開始架構系統
- 物件結構塑模主要以類別圖與物件圖表達物件間之靜態結構
- 物件互動行為塑模主要以互動圖或狀態圖來表達物件間之互動行為，並以狀態圖來表達物件內元件的狀態、轉換與控制等。

物件互動行為塑模工作

- 物件互動行為塑模主要以互動圖或狀態圖來表達物件間之互動行為，並以狀態圖來表達物件內元件的狀態、轉換與控制等。
 - 循序圖
 - 合作圖
- 使用者介面(User Interface, UI)
 - UI循序圖
- 應用程式核心(Application Core)
 - Application Core循序圖

循序圖

- 循序圖是用來描述一個使用個案中之參與物件及物件間的互動行為，強調以時間發生之先後順序表達物件間的訊息傳遞與處理程序。



● 操作描述

1. 設定訂購數量()

- (1) 系統依客戶設定的訂購數量，將記憶體陣列變數〔購物車〕內數量變數設定為新值。
- (2) 執行計算總金額()。

2. 計算總價()

- (1) 系統針對記憶體陣列變數〔購物車〕內的資料進行下列計算。

① 計算單項產品金額 = 單價 × 數量

② 計算訂購總價： Σ (單項產品金額)

- (2) 將記憶體陣列變數〔購物車〕內的所有資料、單項產品金額及訂購總價顯示在網頁中。

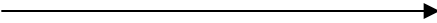
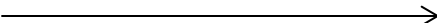




循序圖之元件(1/2)

- 物件
 - 邊界物件、控制物件與實體物件
 - 確認實體物件之準則
 - 找出名詞
 - 刪除不合適的物件
 - 確認屬性之準則
 - 確認操作之準則

循序圖之元件(2/2)

- 訊息
- 操作與操作描述
- 生命線
- 控制焦點
- 框架

循序圖之訊息共通表示法

-  : 同步
-  : 非同步
-  : 創造
-  : 發現
-  : 遺失
-  : 回應

框架(1/8)

- 框架(Frame)是一種圖示標記，用以表達循序圖中某一區段範圍內的一連串訊息與操作組合之控制流程、輸入、輸出等，簡稱組合區段(Combined Fragment)



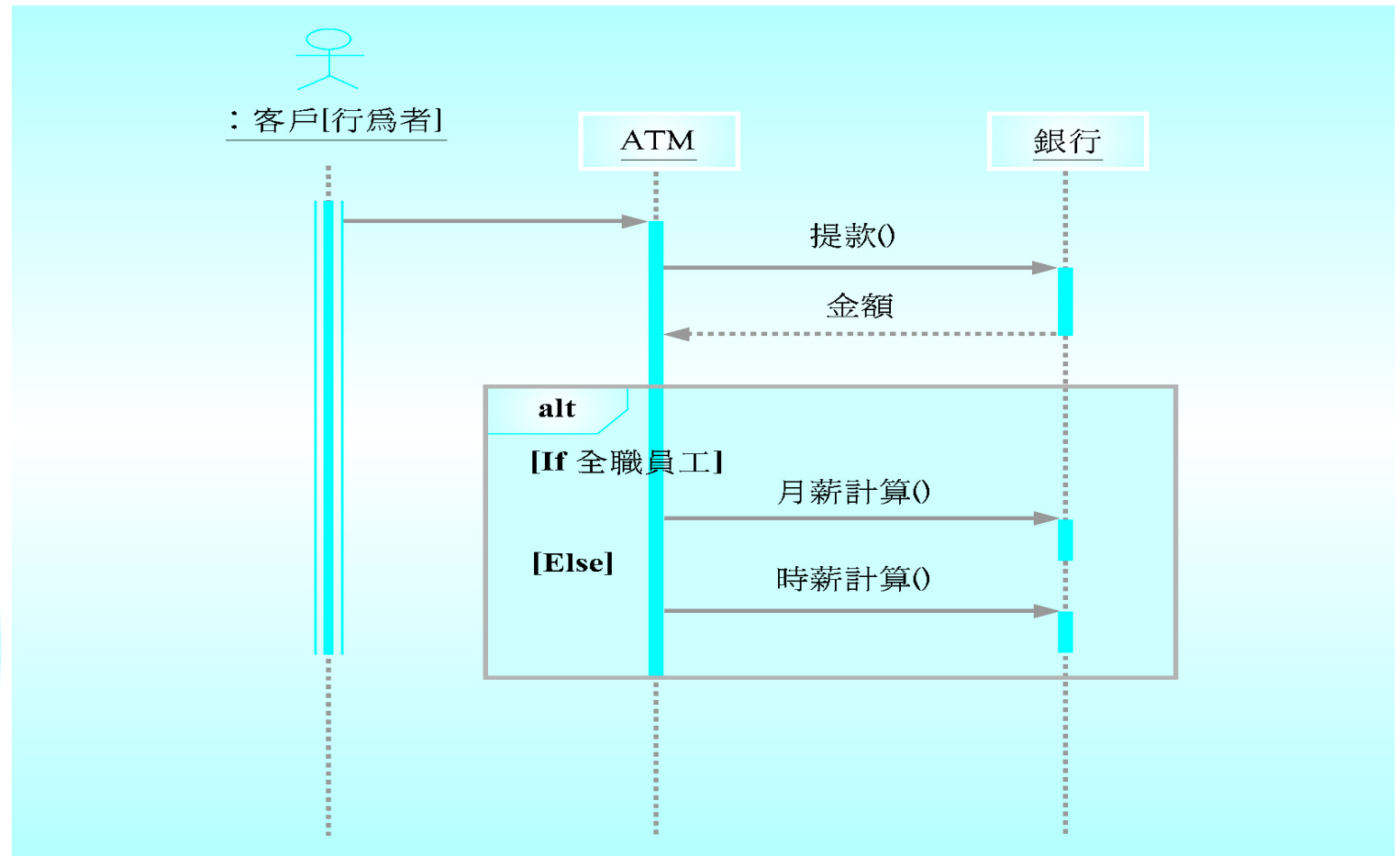
框架 (2/8)

- 多選一

- 多選一(Alternative)是用以表達彼此互斥訊息序列的選擇，典型的範例是「If 前提為真（執行運算域 A），Else（執行運算域 B）」，其中每個運算域可包含多個線式訊息傳遞。



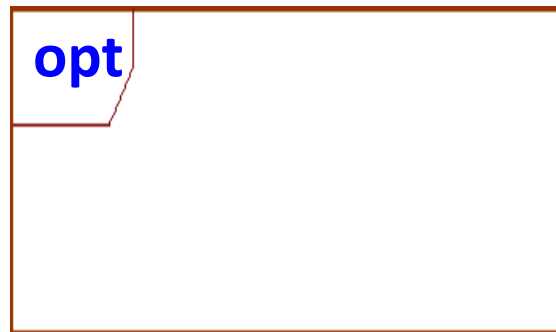
多選一範例



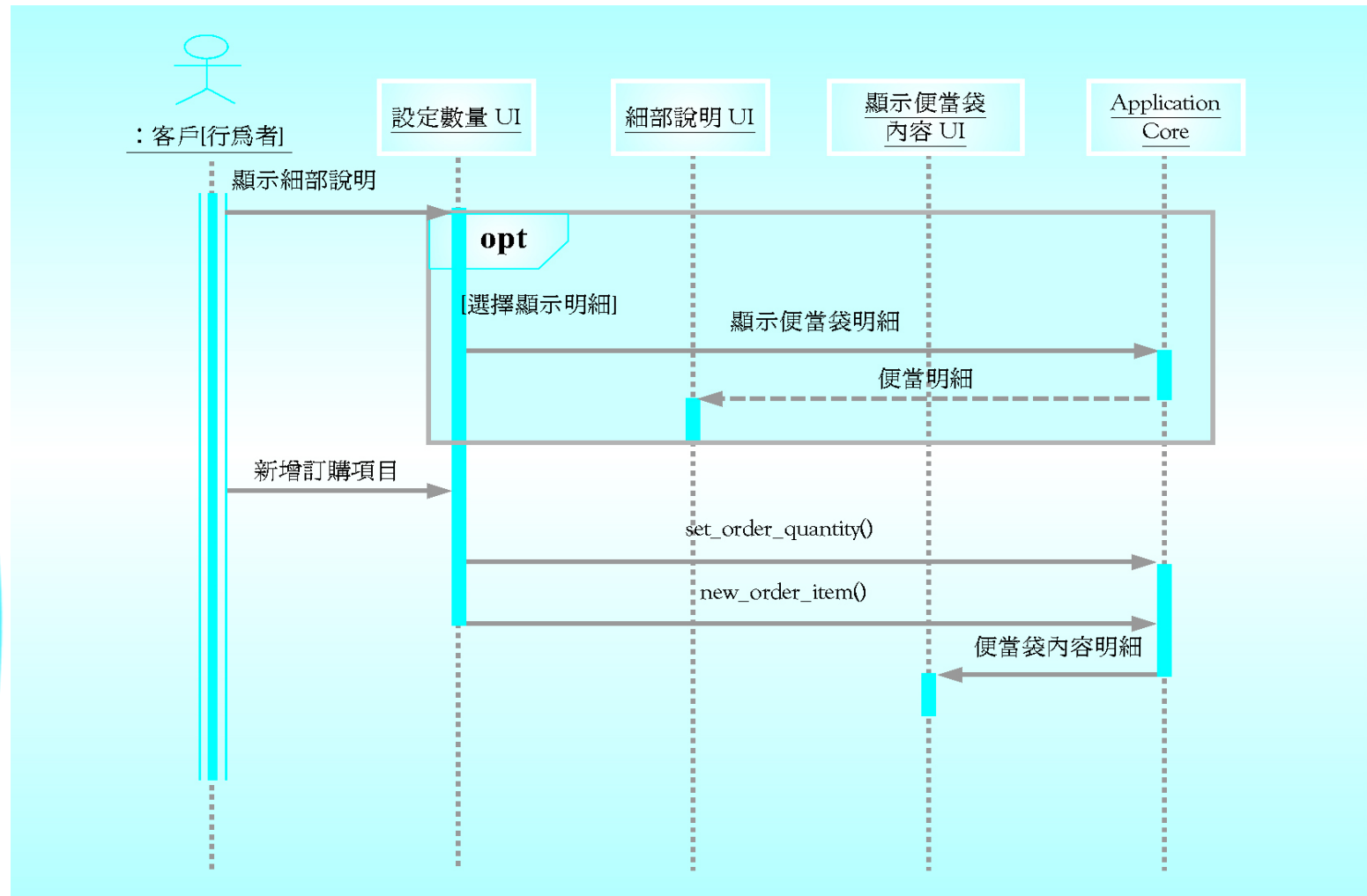
框架 (3/8)

- 自由選擇

- 自由選擇(Option)是用以達可能發生（或可能不發生）的訊息序列，其概念就如多選一的「If 前提為真（執行運算域 A）」，而沒有「Else（執行運算域 B）」；也就是說，它只有一個運算域，該運算域也可包含多個線式訊息傳遞，如果前提成立，則在運算域中的所有訊息會被送出。



自由選擇之範例



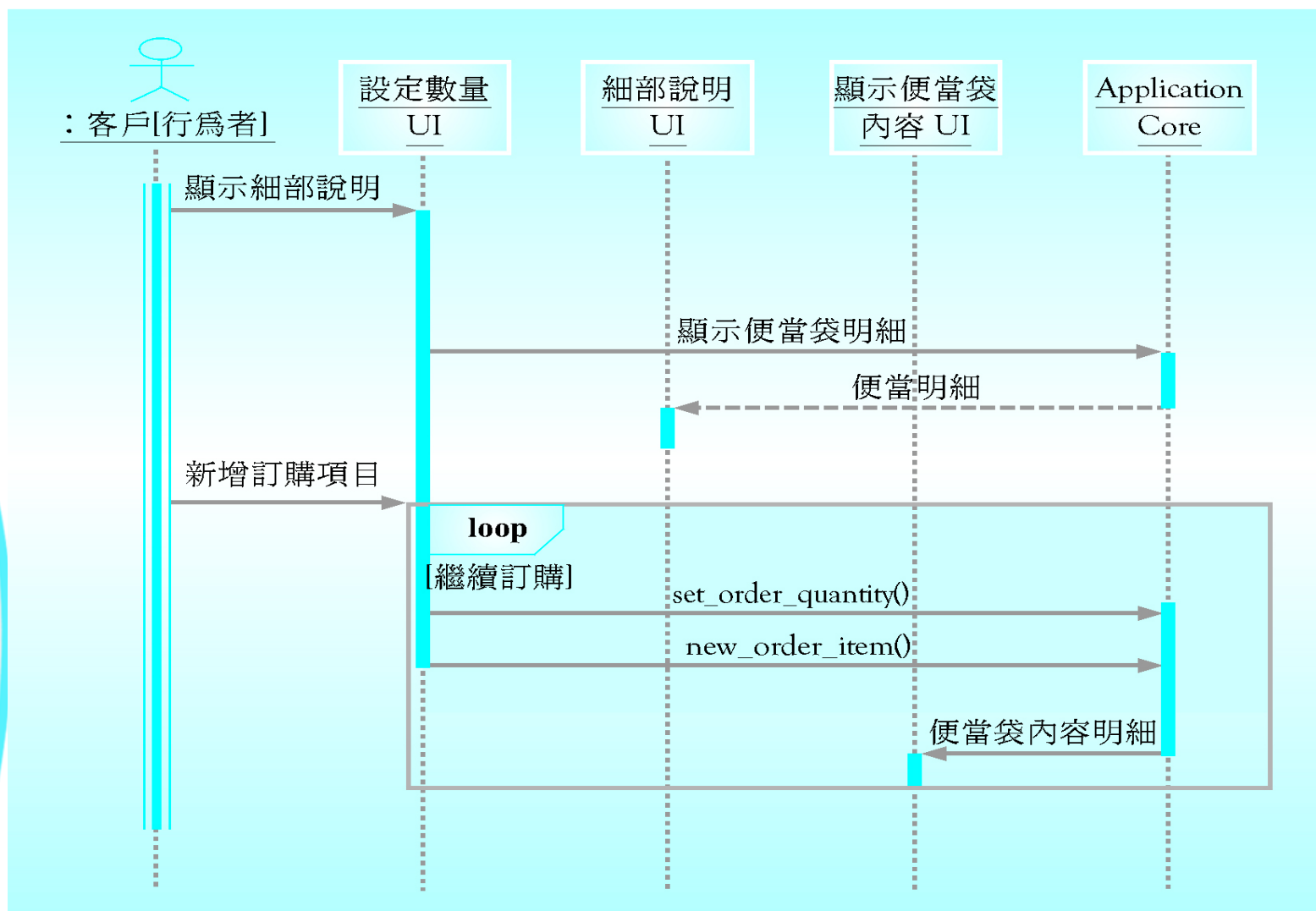
框架 (4/8)

- 迴圈

- 迴圈(Loop)之表達結構很像自由選擇，他有成立條件與一個運算域，該運算域也可包含多個線式訊息傳遞。當條件成立（也就是為「True」），運算域中的操作會持續地被執行，直到條件為「False」為止。



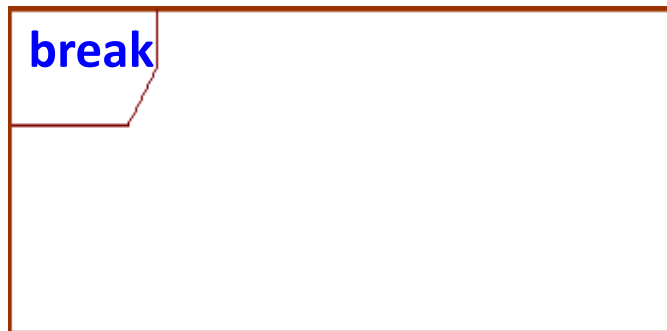
迴圈之範例



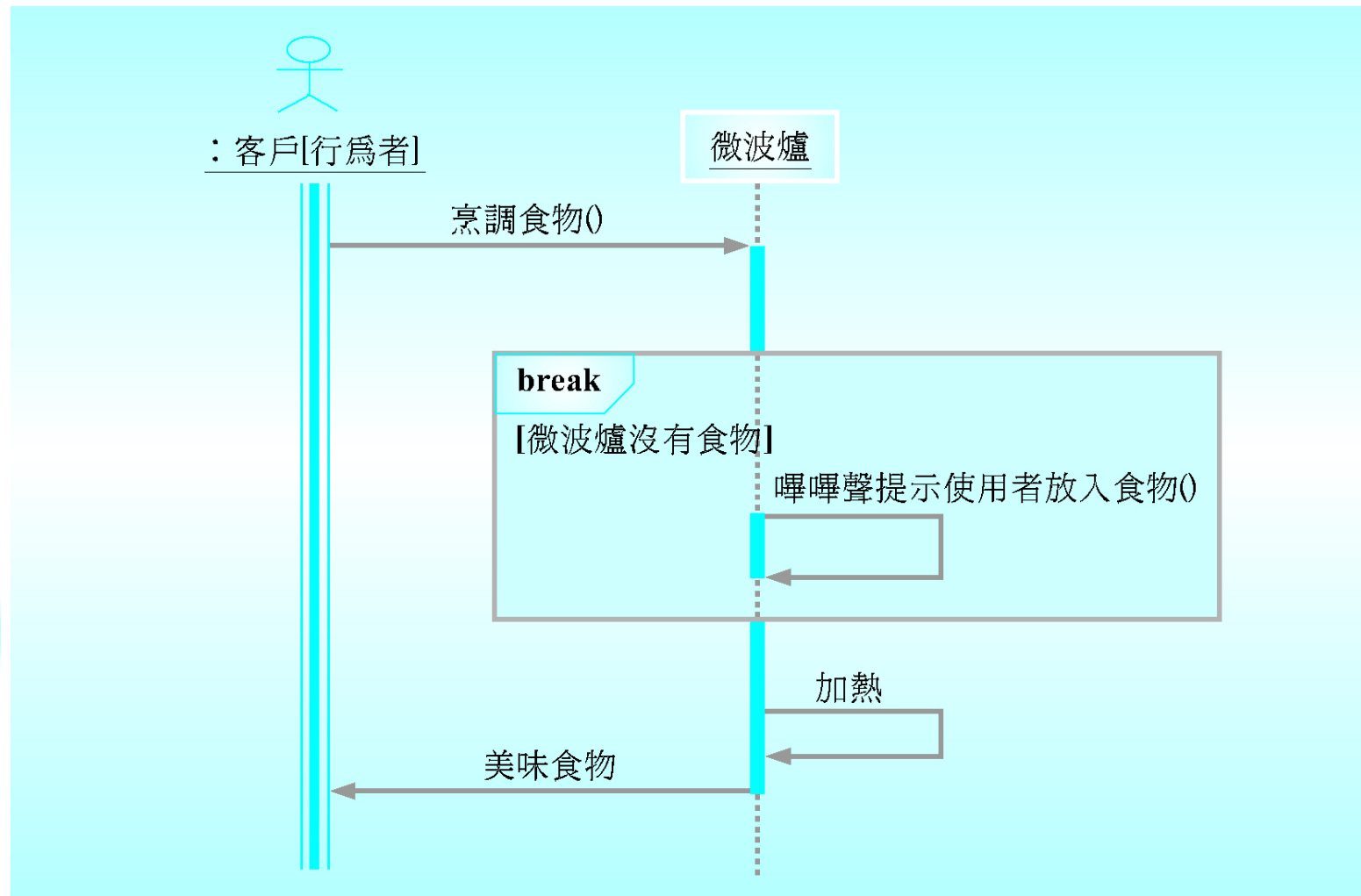
框架 (5/8)

- 終止

- 終止(Break)之表達結構很像自由選擇，他有成立條件與一個運算域，該運算域也可包含多個線式訊息傳遞。若條件不成立（也就是為「False」），運算域中的操作不會被執行（或被跳過）。當條件成立（就是為「True」），運算域中的操作會被執行，但當運算域中的操作被執行完畢，則該區段以下循序圖之所有訊息傳遞與操作將被終止。



終止之範例

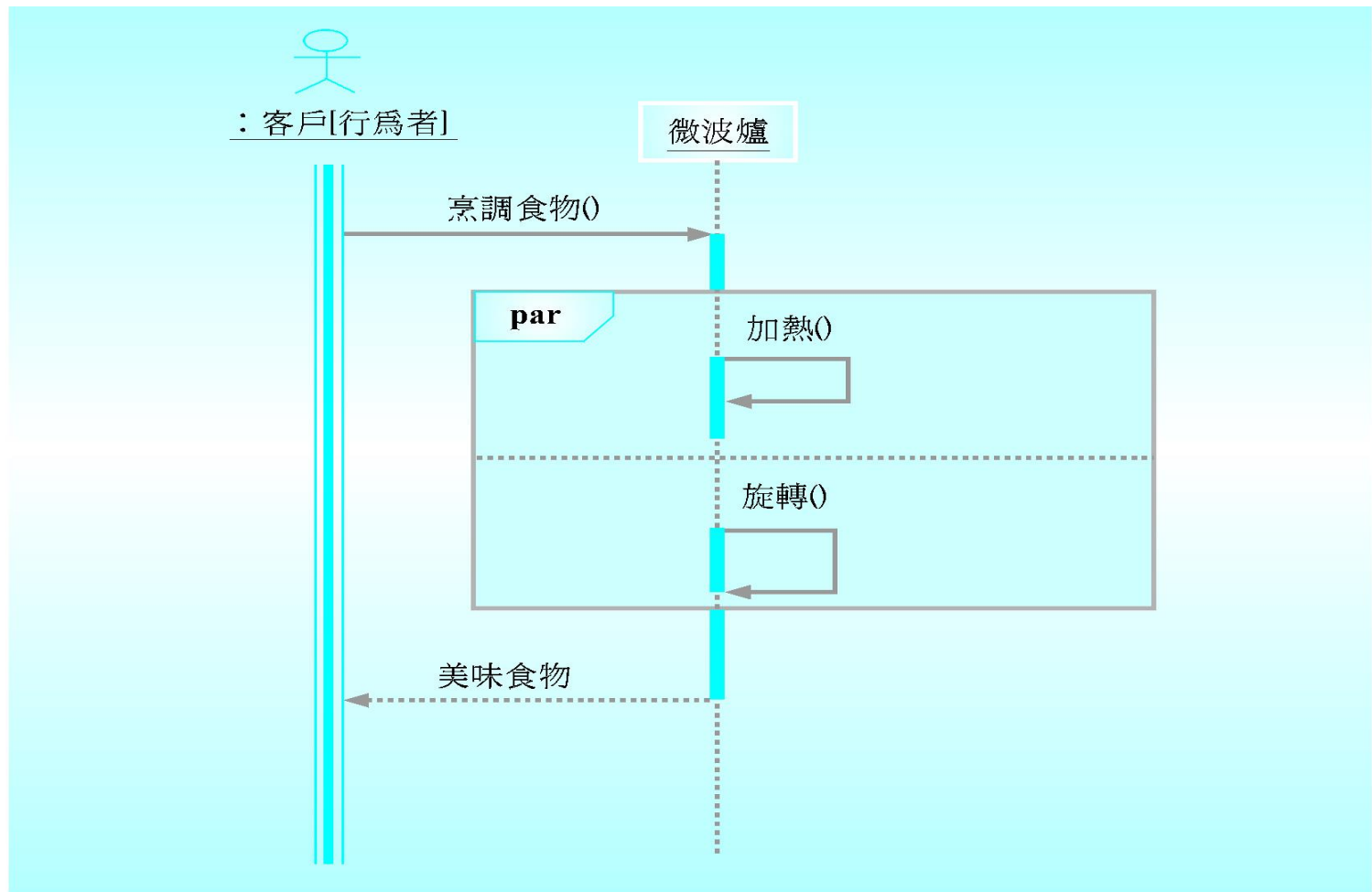


框架 (6/8)

- 平行

- 平行(Parallel)是用以表達可同時執行的多個訊息傳遞。平行的組合區段名稱用「**par**」命名，區段中可以有多個運算域，每個運算域間可用虛線隔開，以增加其易讀性。每個運算域可包含多個訊息傳遞，所有運算域的訊息傳遞可同時執行，且當所有訊息傳遞均完成之後，該區段工作才算完成。

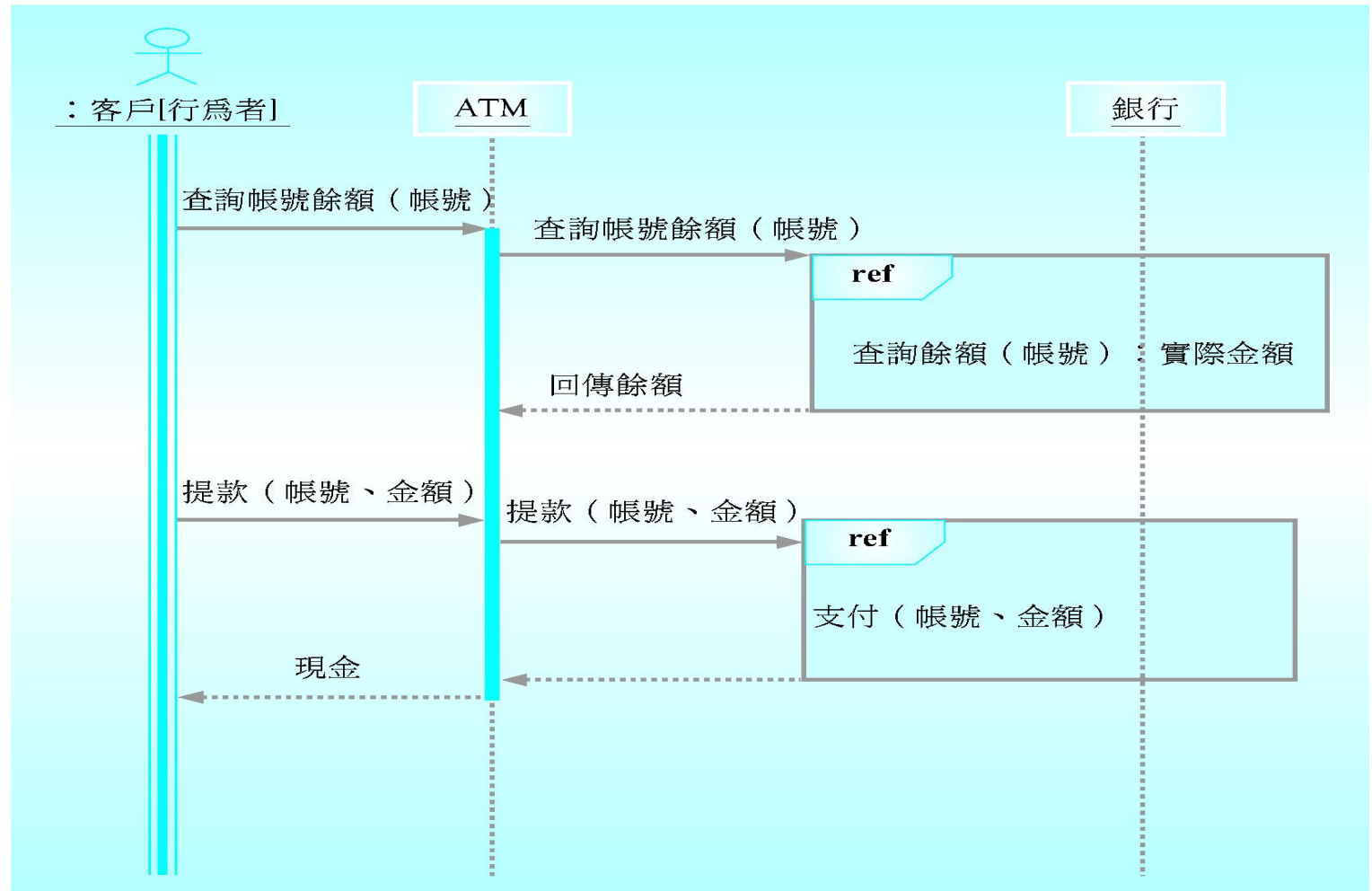
平行之範例



框架 (7/8)

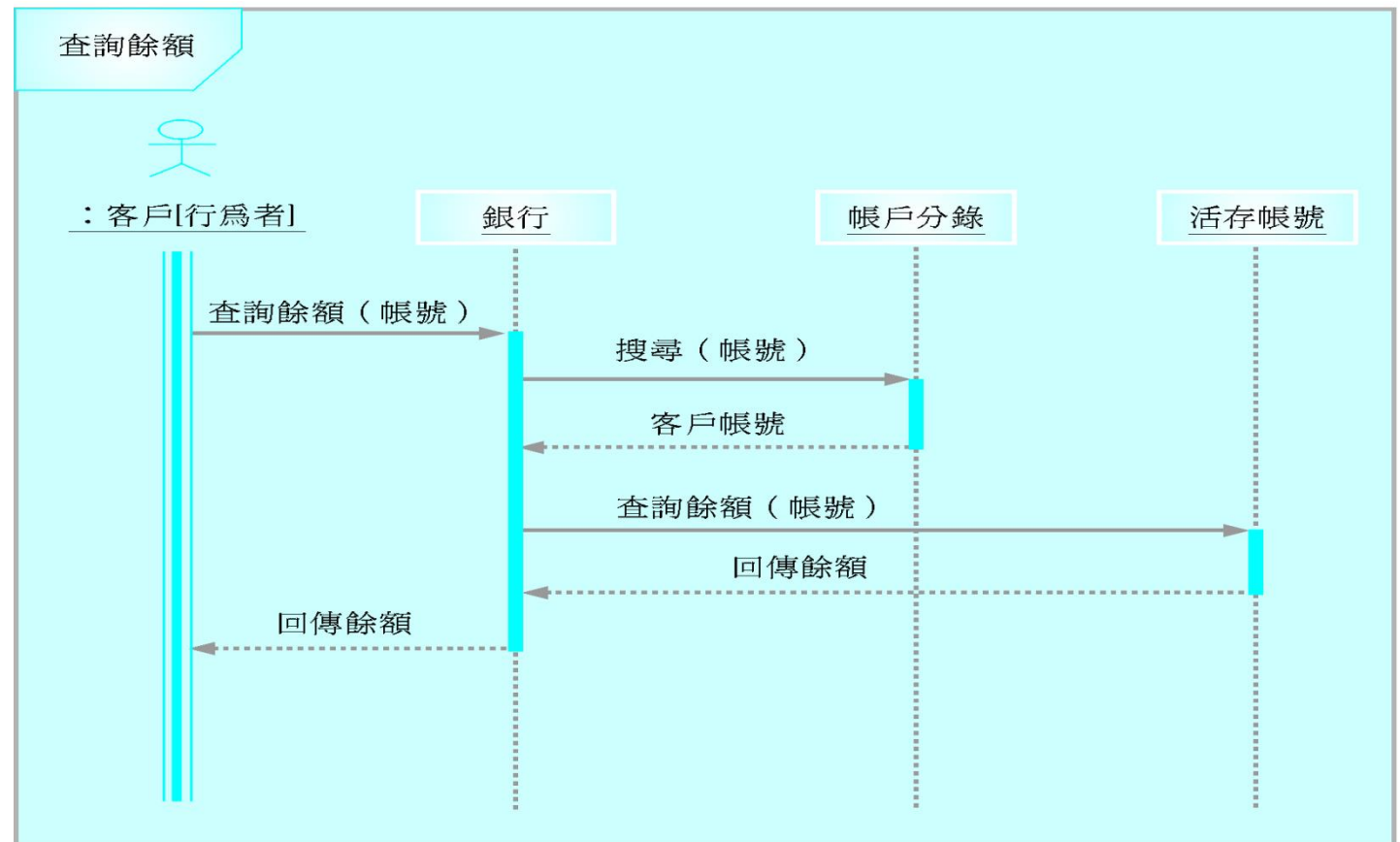
- 參考
 - 參考(Referencing)是用以表達某個循序圖可藉由一個訊息傳遞以參考其他循序圖，例如提供服務，使循序圖可被重用，也可讓循序圖具有組合能力。彼此互斥訊息序列的選擇，典型的範例是「If (前提為真)，Then (執行運算域 A)，Else (執行運算域 B)」，其中每個運算域可包含多個線式訊息傳遞。

參考之範例



框架 (8/8)

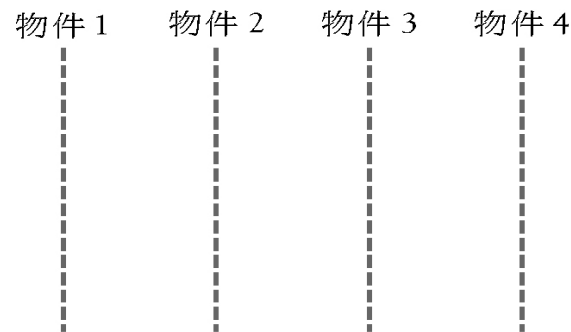
- 被參考循序圖



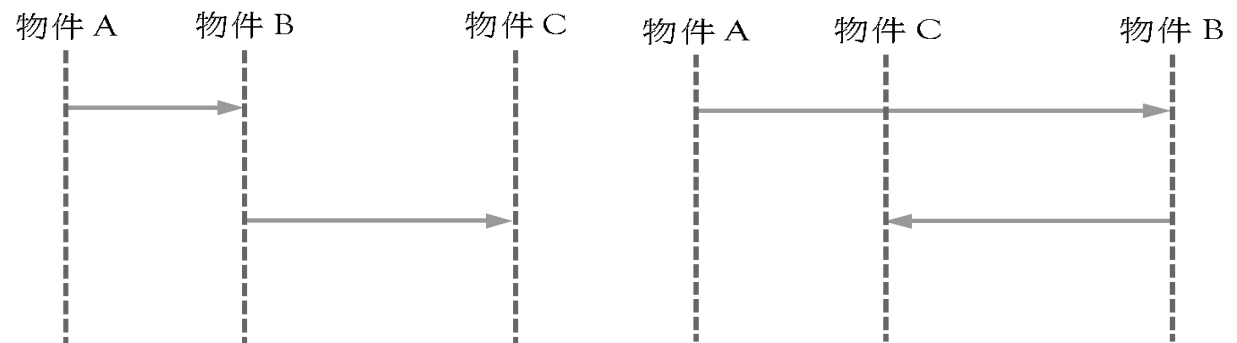
循序圖之建構步驟與準則(1/4)

- 確認物件

- 以物件出現之順序或按時間發生之順序排列

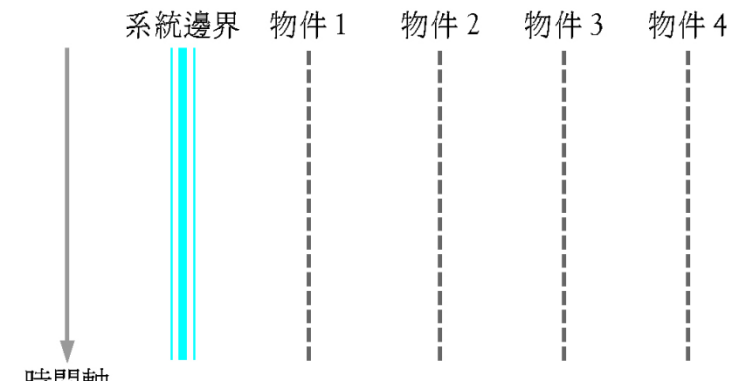


- 兩個物件間，表達訊息之水平線距離愈短愈好



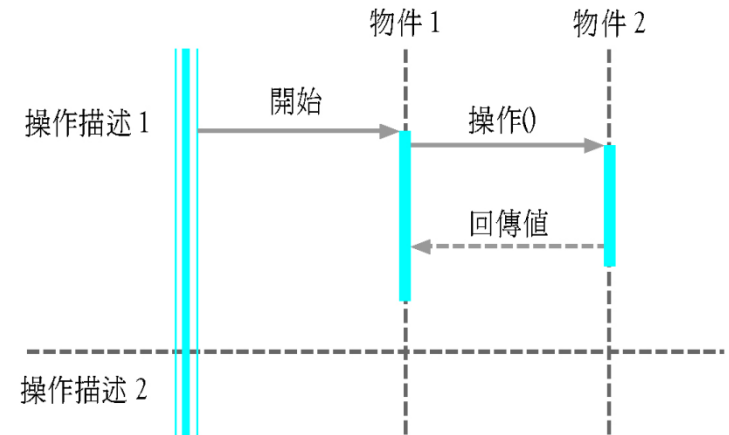
循序圖之建構步驟與準則(2/4)

- 系統邊界 (System Border)



- 描述操作

- 經驗法則
- 結構化英文
- 程式設計語言
- 或物件限制語言



循序圖之建構步驟與準則(3/4)

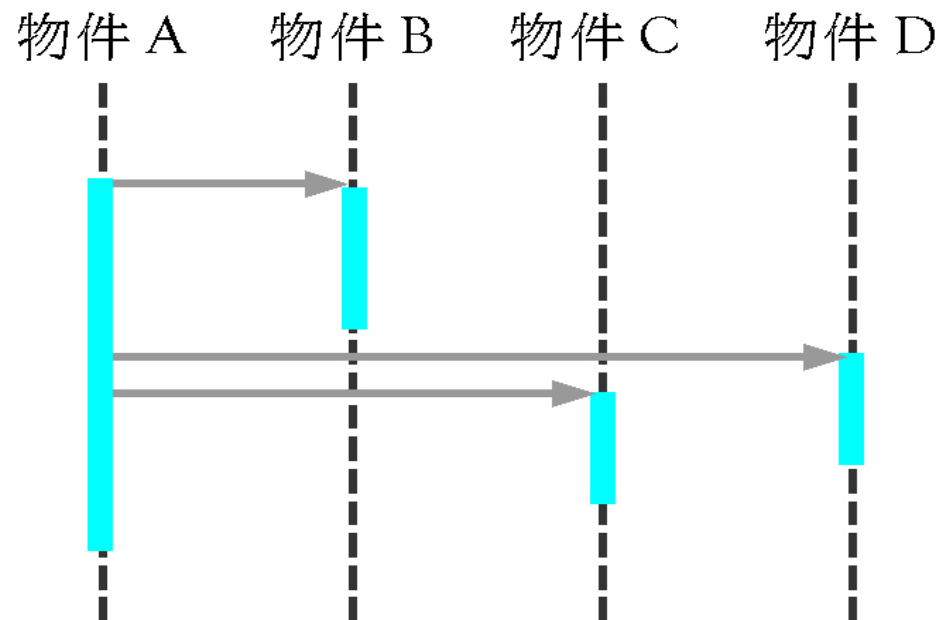
- 描述訊息
 - 盡可能將每一訊息之參數數目減到最少，如此才有利於再用；如果該訊息所需要的參數確實很多時，可將該訊息再拆成數個訊息處理。
 - 相同或類似的訊息處理，其命名應相同；因為從相同之訊息命名，可以瞭解物件間相似的訊息傳遞。
 - 命名時，應能從名稱上反映出物件間訊息傳遞的作用、訊息的內容或其意義。

循序圖之建構步驟與準則(4/4)

- 繪製循序圖
 1. 將物件置於循序圖之上方，並依物件間訊息傳遞／接收發生之時間順序等原則安排物件由左至右之順序
 2. 在物件下劃與物件垂直之生命線，再標示物件間傳送／接收之訊息與相關之操作
 3. 接著依每一物件執行某動作之時段劃控制焦點。

循序圖之架構(1/2)

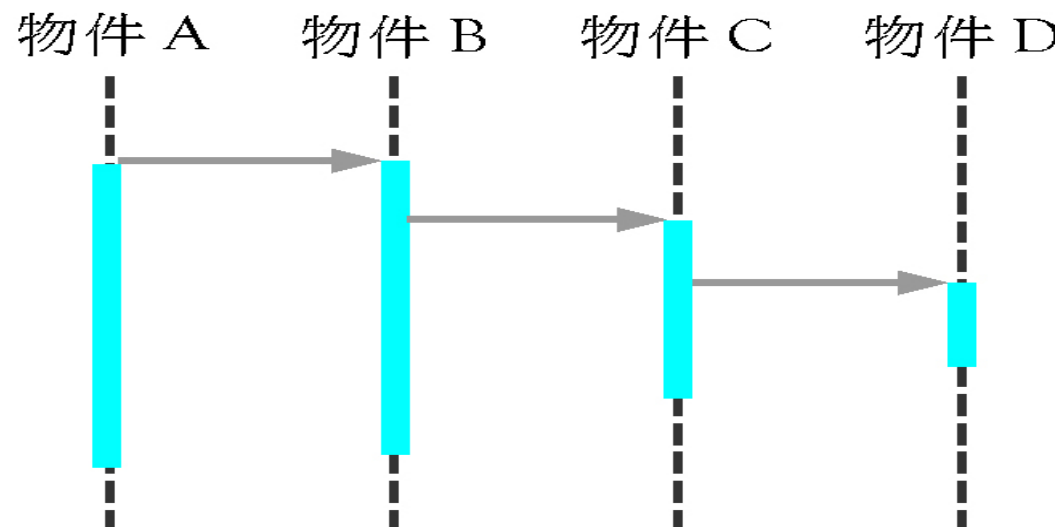
- 集中式
 - 集中式(Fork)架構之特徵是所有的物件訊息傳遞皆由一個物件負責。



循序圖之架構(2/2)

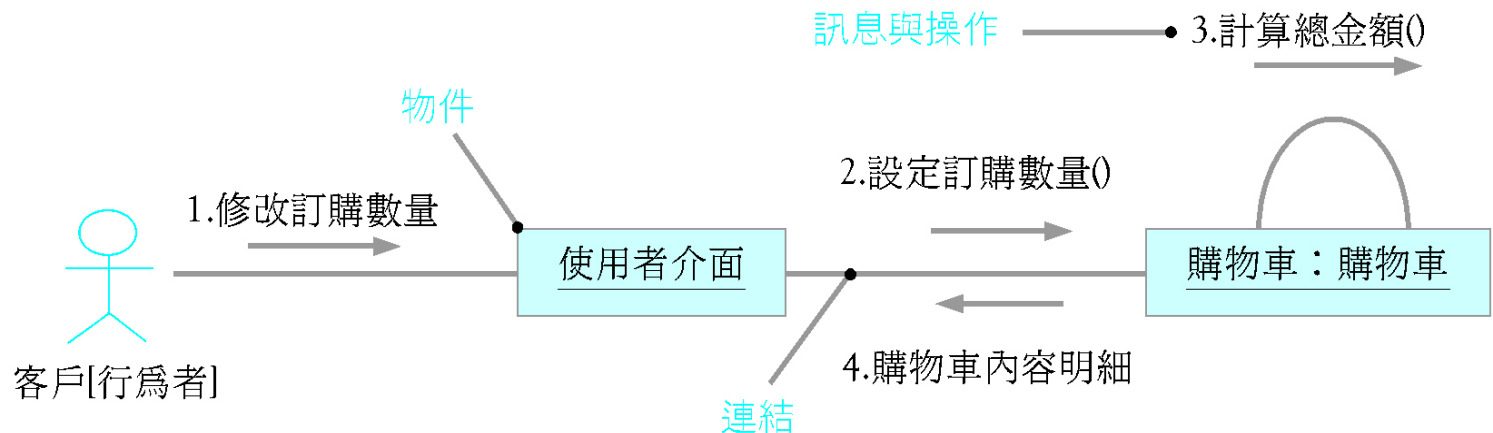
- 階梯式

- 階梯式(Stair)架構之特徵是物件訊息傳遞不全然由某一物件負責，每一物件均可涉及訊息之傳遞，但只負責處理本身或被傳遞到之操作。此種架構，系統設計之複雜度可分散到每一個物件上。



合作圖

- 合作圖強調以物件的結構化組織表達物件間的訊息傳遞／接收與處理程序。
- 合作圖之元件
 - 物件
 - 連結
 - 訊息與操作



合作圖之建構步驟與準則

- 合作圖中之物件與循序圖中之物件是相同的。
- 合作圖中之連結是來自於循序圖中兩物件間有訊息傳送／接收者
- 合作圖中之訊息與操作與循序圖也是相同的，只不過在表達時，循序圖是依時間順序來表達發生順序，而合作圖是以數字（自然數或杜威數）來表達發生順序。合作圖中之訊息與操作表達在相關物件間之連結旁。







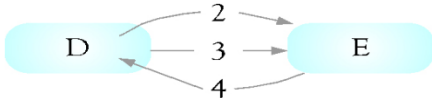

狀態圖

- 表達一個物件、一個使用個案、多個使用個案間或一個系統在其生命週期中之行為，且強調表達狀態及其轉換關係。

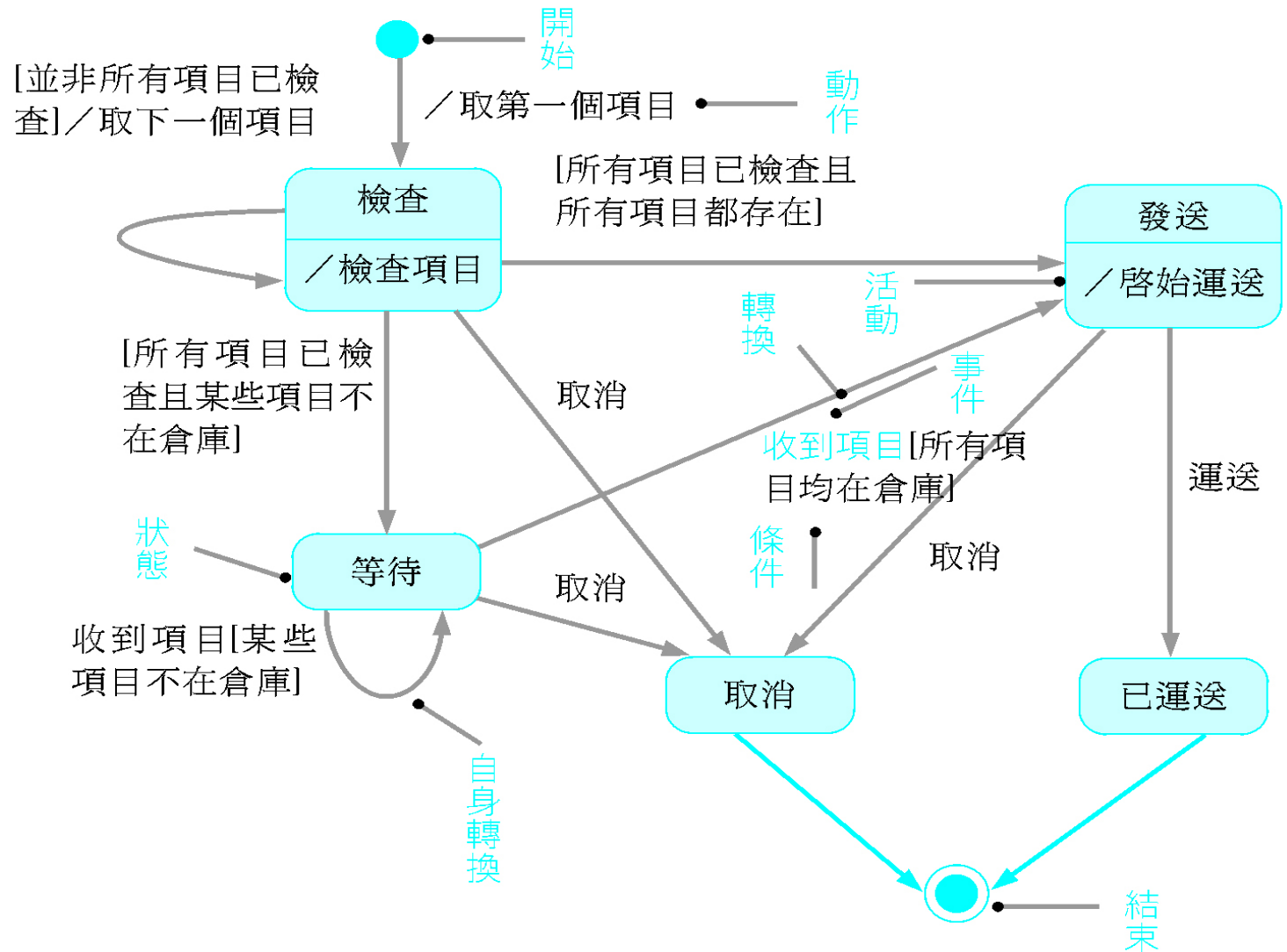


狀態圖之元件

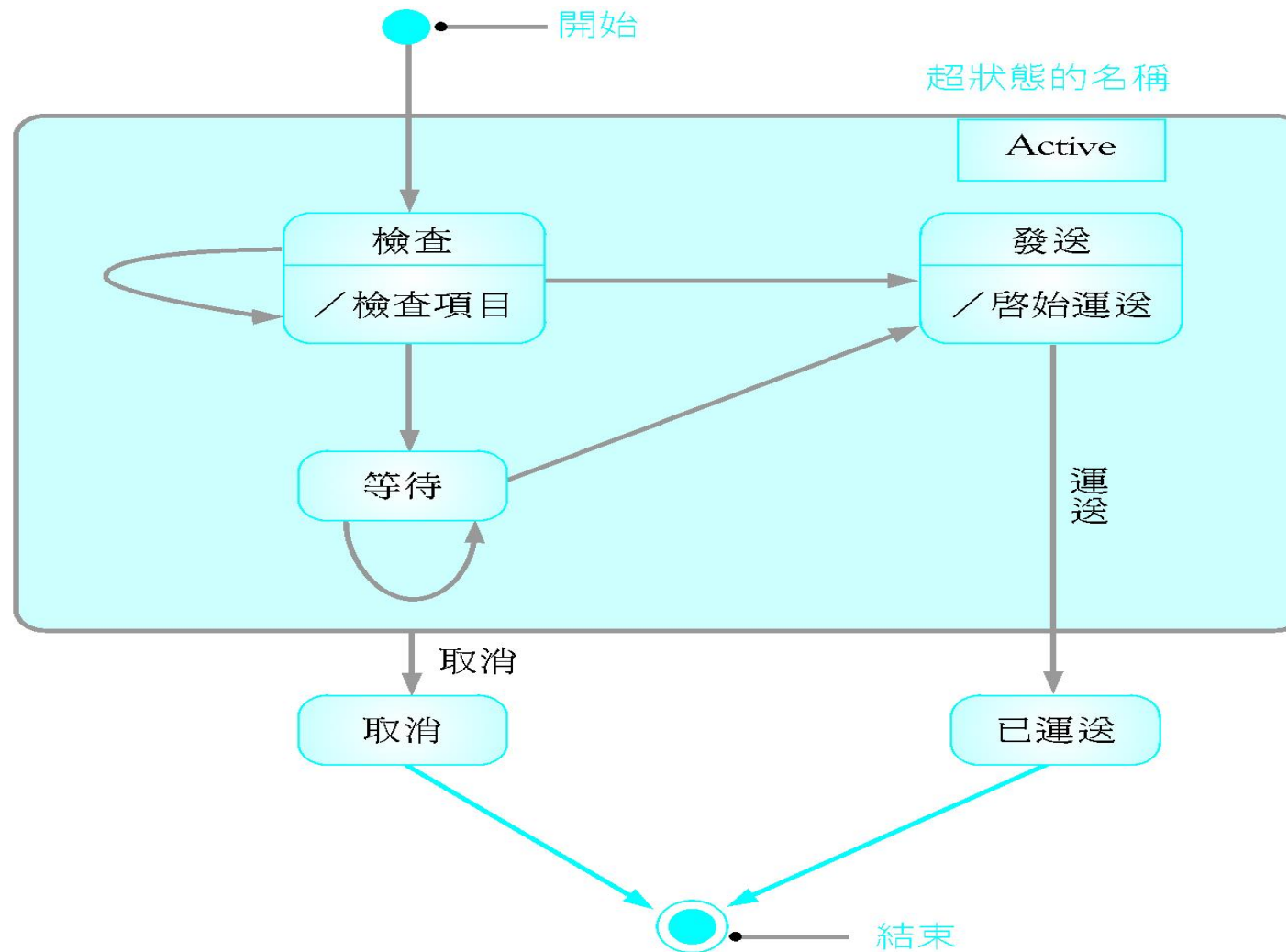
- 開始狀態
- 一般狀態
- 歷史狀態
- 同步組合狀態
- 循序組合狀態
- 轉換
(事件[成立條件]/動作)
- 狀態具深度
- 結束狀態

狀態圖元件	符 號
開始狀態	
一般狀態	
歷史狀態	
同步組合狀態	
循序組合狀態	
轉換 (事件[成立條件]/動作)	
狀態具深度	
結束狀態	

狀態圖範例



超狀態圖範例



狀態圖之建構步驟與原則 (1/2)

- 建構步驟
 1. 找出狀態
 2. 找出狀態間之轉換
 3. 繪製狀態圖
 4. 精鍊(Refine)狀態圖。

狀態圖之建構步驟與原則

(2/2)

- 原則

- 從循序圖中物件之操作描述、使用個案之描述，逐一找出狀態圖之狀態與轉換。
- 狀態之轉換，「事件〔成立條件〕 / 動作」，此三個部分是可選擇性的，不一定要同時都具備。
- 由狀態圖之上方或左上方以「開始」畫起，從系統的觀點，依物件之行為，將物件生命週期的活動狀態之順序，逐一畫出所有狀態及轉換。
- 自身轉換的表示法是，箭頭由該狀態伸出，繞一圓弧後，箭頭再指向該狀態，並在適當位置說明「事件〔成立條件〕 / 動作」。
- 繪製狀態圖時，其轉換符號應盡量避免交叉。

結論

- 應用互動圖（循序圖與合作圖）與狀態圖來表達一個使用個案內物件間與物件內元件之互動行為。
- 每一個使用個案須建立一個互動圖，建構互動圖所需之資訊主要來自於使用個案圖或類別圖，以及與使用者之互動。