

Object-**O**riented **A**nalysis and **D**esign **U-01 R**equirement **A**nalysis

Neter Chao Ph.D.

內容大綱

- 需求分析導論
- 需求擷取方式
- 需求塑模工具與方法
- 需求分析之重要工作與文件樣版
- 案例介紹與需求描述
- 需求塑模 - 建構使用個案圖
- 需求塑模 - 建構活動圖
- 強韌分析 - 建構強韌圖
- 結論

學習目標

- 常見的六種需求擷取方式及其執行要點。
- 需求分析之步驟與應注意之事項等。
- 何謂使用個案模式。
- 如何建構使用個案圖、藍圖與資料詞彙。
- 何謂活動圖及其用途。
- 如何以使用個案圖、活動圖、藍圖、資料詞彙等進行需求塑模。
- 需求分析之重要工作與文件樣版。

學習目標(續)

- 如何建構使用個案圖。
- 何謂活動圖及其用途。
- 如何建構活動圖及其建構之準則。
- 何謂強韌分析及其用途。
- 如何運用強韌分析精鍊使用個案模式。
- 如何以使用個案圖、活動圖、藍圖、資料詞彙等表達需求。
- 如何以CASE工具建構使用個案模式。
- 如何以CASE工具建構強韌圖。

1.1 導論(1/6)

- 使用者需求係指使用者期待系統解決的問題與希望從系統中獲得之資訊。
- 使用者需求是資訊系統開發過程中最關鍵、最重要且最容易發生錯誤的部分，亦是資訊系統失敗的主因之一。

導論(2/6)

- 需求分析是資訊系統發展過程中之一項非常重要的活動，亦是一個重要之階段。理論上，該階段之主要工作是應用已通過驗證之原理(Principles)、技術(Techniques)、語言(Languages)與工具(Tools)，以幫助分析師瞭解問題或描述新系統之外部行為。

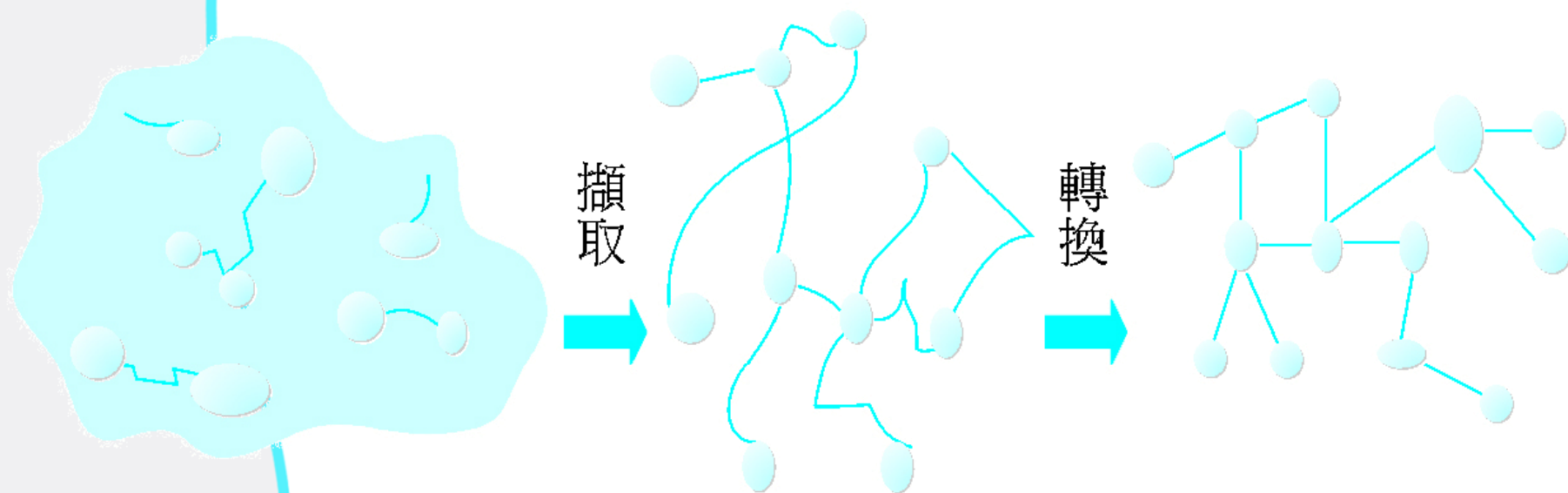
導論(3/6)

- Hooper 與 Hsia (1982)認為，需求分析階段主要包括三個活動：(1) 需求判斷；(2) 需求分析；(3) 需求溝通。
- 需求判斷強調如何判斷真正的需求及需求的正確性。
- 需求分析則重視在分析已有的需求下，所產生的不一致、不完整或矛盾等問題。
- 需求溝通重視以最佳的方式組織及描述需求，以使得需求容易令人瞭解，並經由相互溝通達到需求確認的目的。

導論(4/6)

- 需求分析階段基本上可分為兩大步驟：需求擷取與需求轉換（如圖 1-1）。
- 需求擷取主要是對系統範圍內之各種事物與相關現象，加以瞭解、判斷及選擇，並設計成描述性綱目。
- 需求轉換主要將描述性綱目以系統模式語法轉換成概念性綱目。

需求分析之重要步驟



各種事物現象



描述性綱目



概念性綱目

導論(5/6)

- 使用者需求可分為巨觀需求與微觀需求。
 - 巨觀需求包括欲電腦化之環境、作業程序與範圍、輸出與輸入所需之資訊或表單及系統目標、限制、主要功能等，這些需求應盡可能的在需求分析階段中釐清與確定。
 - 細部需求指的是電腦化之微觀範圍，可包括使用者介面之要求、例外狀況之處理與錯誤及輔助訊息之顯示等要求，這些需求通常需到設計階段才較容易處理，在此之前這些細部需求不易被掌握。

導論(6/6)

- 需求分析階段之重要工作包括：
 - 瞭解現有問題
 - 瞭解新系統目標
 - 瞭解新系統之限制
 - 瞭解使用者巨觀之需求等

1.2 需求擷取方式

- 在擷取使用者需求之前，必須瞭解系統之潛在使用者及可能之人機互動。
- 常用的需求擷取方式有查閱文件、觀察、訪談、問卷、開會討論與聯合開發等六種，這些方式可單獨應用亦可混合使用。

1.查閱文件

- 研究企業的內部文件，例如工作說明書、企業表單(Business Forms)與手冊等，是瞭解企業運作邏輯之初步工作。
- 一般來說，組織中很少有完整的文件詳細地描述出系統之全貌，再加上系統可能已經過多次的修改，文件往往未能配合更新，因此以該方式蒐集之資訊常有過時之慮。

2. 觀察

- 一般來說，實地觀察所獲得的資料正確性會比查閱文件為高，亦能驗證所蒐集資料之正確性及補充不完整的部分，而且透過實地觀察可以獲得第一手的資料。
- 僅用觀察仍無法完整地反映出組織的真實情況與需求
，例如被觀察者的行為可能改變。
- 選擇正常與例外情況之時機或對象來做觀察，以便獲得各種可能的資料。

3. 訪談(1/10)

- 訪談是需求擷取方式中最有效且最普遍的資料蒐集方法。訪談時，系統分析師親自與使用部門的主管或相關作業人員面對面討論實際作業的情況、報表和資訊需求等。
- 在訪談期間，系統分析師蒐集到的可能是事實、選擇或推測，並且可觀察到人們的肢體語言、情緒和他們對於現行系統之觀感等。

訪談(2/10)

- 訪談可分成兩種方式：
 - 開放式訪談(Open Interview)。
 - 結構化訪談(Structured Interview)。
- 開放式訪談
 - 系統分析師事先不預定表格、問卷或固定的標準程序，訪談過程全由使用者自由談論其工作。
 - 應用在系統分析師對問題領域不熟悉的情況。

訪談(3/10)

- 結構化訪談
 - 結構化訪談又稱為標準化訪談或導向式訪談，其訪談過程近似於詢問(Interrogation)而非交談(Conversation)，所要求資訊的深度與專業程度亦較深。

訪談(4/10)

- 結構化訪談的特點是把問題標準化，然後由受訪者回答或選擇。
- 所有的受訪者都回答同一形式的問題，其作法如下：
 - (1)每討論一個主題時，先由系統分析師依其現有瞭解的知識，提出簡短敘述。
 - (2)使用者針對主題作深度分析，但系統分析師應在深入到適當程度時加以中止。
 - (3)對某個主題有初步瞭解，就可以進行另一主題的訪談。

訪談(5/10)

- 訪談之問題依其性質可分成開放性與封閉性問題兩種：
 - 開放性問題
 - 用來探索系統分析師對於受訪者之回答無法預期或明確詢問之問題。
 - 優點：能讓先前不知道的資訊浮現出來，系統分析師可以用一些非預期中的問題，不斷地來探究新的資訊。
 - 缺點：回答問題所花的時間較長，也較難做結論。

訪談(6/10)

— 封閉性問題

- 適用於問題可預期且回答可明確描述之情況。
- 優點：訪談的時間較短，問題可較廣泛。
- 缺點：所列回答之選項未必包含受訪者所要回答的答案。

訪談(7/10)

- 封閉性問題可以有列幾種設計形式：
 - (1)對與錯的選擇方式。
 - (2)多重選擇的方式。
 - (3)Likert尺度的衡量方式。用一些級距來衡量受訪者意見的強弱程度，例如用很好、好、普通、差與很差五個等級。

訪談(8/10)

- 訪談前須有周詳的規劃。
- 一些有助於訪談進行之原則包括：
 - (1)要仔細地聆聽受訪者的回答，同時將一些重點記錄下來，在得到允諾的情況下，還可以將訪問的內容用錄音機或錄影機錄下來，以有效地掌握訪談內容。
 - (2)訪談結束後，須在48小時之內將訪談內容整理出來，因為經過48小時之後，訪談的內容會慢慢地從記憶中消失。

訪談(9/10)

- (3)不管是面對開放性或是封閉性的問題，系統分析師不要強調問題答案的對或錯。
- (4)在訪談時，不要對新的系統做任何預期的想法。需讓受訪者知道他們的意見會被仔細地考慮，並讓受訪者知道專案的完成需要很多的步驟，同時還有很多人的觀點都需要一起納入考量。

訪談(10/10)

(5)從系統潛在的使用者、管理者與對現行系統有經驗的專業人員等尋找各式各樣的觀點，以對新系統做全盤性的瞭解，如此才能設計出大多數使用者都可接受的系統。

4.問卷(1/6)

- 當潛在之使用者太多或分布太廣時，此時可考慮以問卷之方式擷取需求。
- 一般來說，問卷調查適合於大型企業或公眾資訊系統的設計，因為它所涉及的作業範圍或對象太廣，系統分析師無法逐一親自調查，故利用問卷方式來蒐集使用者需求較為可行。

問卷(2/6)

- 設計一份好的問卷需有相當的練習與經驗，因為問卷上的問題是以文字靜態的表達出來，因此問題之語意與邏輯必須很清楚且有條理。
- 問卷設計時也可以用各種不同的方法來問同一個問題，以觀察各種可能的答案？

問卷(3/6)

- 進行正式問卷調查前需有先導測試。
- 經由先導測試後之檢討與回饋可進一步修飾問卷，及早發現問卷可能之問題，對提升問卷之信效度有很大之幫助。

問卷(4/6)

- 機率抽樣方法：
 - 簡單隨機抽樣：將系統使用者的名單逐一給予編號，再以摸彩法或亂數表來抽樣，以決定哪些人將成為受測的對象。
 - 分層抽樣：先將所有可能之系統使用者分成若干互斥的組或層，然後再從各組或各層中，隨機抽選預定數目之使用者作為受測對象。

問卷(5/6)

- 非機率抽樣方法：
 - 便利抽樣：以方便性為抽樣之主要考量，這些抽樣之對象可能是最容易找到的使用者、願意接受調查的人或是動機高的使用者。
 - 判斷抽樣：又稱立意抽樣(Purposive Sampling)，係根據抽樣設計者的判斷來選擇受測之使用者。設計者必須對使用者之特徵有相當地瞭解，然後針對符合某些特徵的名單抽樣，例如使用系統已經兩年以上的使用者或者是經常使用的人。

問卷(6/6)

- 上述抽樣策略可單獨使用，亦可混合使用。當問卷回收後，應逐一檢查回收問卷之回答是否完整，並去除異常之問卷，以便進行資料分析。

5. 開會討論

- 開會討論是一種很有效率的資料蒐集方式。使用者代表與系統開發人員齊聚一堂，將所知道的事實、觀念說出，讓所有與會人員一起相互溝通意見。
- 此方法的優點是較易獲得正確的資料，因為縱使有不正確的意見與觀念，經由眾人研究亦能加以修正。此外，亦可發揮腦力激盪的效果。
- 缺點是要安排共同的時間來進行，在溝通與協調上較困難。

6. 聯合開發(1/7)

- 聯合開發 (Joint Application Development, JAD) 主要之精神，是透過一個二至五天的集會，讓開發者與顧客能夠快速有效而且深入地檢討需求，並取得共識。
- 聯合開發的具體結果是產生完整的需求文件。

聯合開發(2/7)

- JAD依下列五個步驟來進行：
 - (1) 範圍界定
 - (2) 關鍵人員的熟悉
 - (3) 會議準備
 - (4) 會議進行
 - (5) 文件產生

聯合開發(3/7)

- 範圍界定
 - 首先由專案出資單位的高階主管定義專案的範圍，並以文字記載，且由高階主管和JAD的召集人一起簽訂契約。這個步驟使JAD的召集人得到來進行需求分析的授權，對於目標與範圍也有了約定。

聯合開發(4/7)

- 關鍵人員的熟悉
 - JAD的召集人要花一些時間訪談關鍵性的使用者及管理人員，以瞭解專案的背景資料及重要的需求。

聯合開發(5/7)

- 會議準備
 - JAD會議事前的準備工作應包括下列項目：
 - 整理需求文件草稿
 - 分送需求文件草稿
 - 安排助理人員
 - 準備會議室

聯合開發(6/7)

- 會議進行
 - 會議進行時，召集人引導大家充分利用各種視覺上的輔助工具，如貼紙、白板、投影片、圖表或簡報檔等，將需求表達出來並作有效地溝通及使用者與企業達成共識。

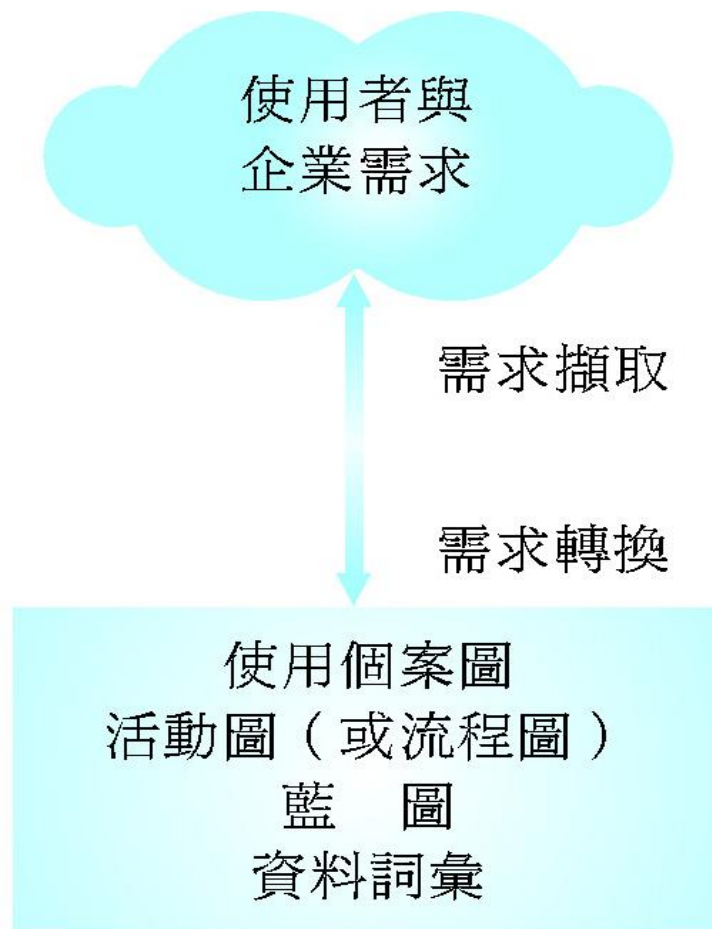
聯合開發(7/7)

- 文件產生
 - 最後階段將JAD會議所蒐集的需求整理成需求文件
 - ，為達到會議的效果，需求文件的準備要必須非常快，例如二、三天內完成。最後再召開一次審查會議，以確認需求文件的內容。

需求塑模工具與方法(1/2)

- 企業程序是企業處理有組織的結合，其執行結果須能完成某企業功能，例如行銷或生產管理等。
- 需求分析階段之重點工作是先擷取使用者（或企業）巨觀需求，並以使用者之觀點應用具有完整定義的（Well-Defined）工具、圖形或語言將需求表達出來，再進一步對需求進行合理化，最後經由使用者確認，以作為系統分析與設計的基礎。

需求塑模



需求表達工具與方法(2/2)

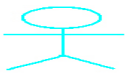




- 需求塑模的工具包括：
 - 使用個案圖、活動圖
 - 主要表達實體(Entity)與作業程序及所需資訊間（輸入與輸出）之關係。
 - 藍圖(Drawing)
 - 主要表示資訊之展示格式與內容等，例如表單之格線與縱橫項目。
 - 資料詞彙(Data Glossary)
 - 主要描述藍圖內資訊之詳細內容與規則。

使用個案圖(1/2)

- 使用個案圖是一種使用者需求表達之塑模工具，該工具從使用者之觀點描述系統提供之功能與定義系統內部之作業，並以使用個案圖(Use Case Diagram)表達之。
- 使用個案圖主要是由行為者(Actors)、使用個案(Use Cases)與關係等元件所組成，其可標示行為者與使用個案間之互動及使用個案與使用個案間之關係。

使用個案圖(2/2)

- 使用個案圖中，行為者常以人形表示，使用個案以橢圓表示，行為者與使用個案間之互動以實現表示，使用個案間之關係以箭頭表示，表達符號摘述如下圖：

使用個案圖元件	符 號
行為者	
使用個案	
行為者與使用個案間之互動	
使用個案間之 Include 與 Extend 關係	
使用個案間之 Generalization 關係	

使用個案圖之元件(1/8)

- 行為者

- 行為者是環境中的人或事物，它扮演與企業互動之角色且須與系統交換資訊之任何事物。簡言之，行為者是使用者在系統中所扮演的角色之一，該角色可與系統交換資訊與互動。

使用個案圖之元件(2/8)

- 行為者和使用者不同，使用者是一個真正使用系統的人，而行為者只是用來表示使用者所執行的某一種角色。
- 例如：業務人員是系統之使用者，參與銷售與採購兩個使用個案，且是執行這兩個使用個案之主要行為者，而客戶與廠商分別是銷售與採購使用個案之次要行為者。

使用個案圖之元件(3/8)

- 使用個案
 - 一個使用個案是系統中一系列的交互，以完成某一特定工作，並對系統之行為者產生可衡量的價值。也就是說，工作結的表現行為者產生一些可看得見、可量化或質化的效益。
 - 每一個使用個案就是使用者透過介面要求系統所做一系列相關的事件流，有起點亦有終點，又稱為一個案例(Instance)。

使用個案圖之元件(4/8)

- 使用個案應描述基本路徑與替代路徑。
- 基本路徑是事件之最重要路徑，它最讓人瞭解使用個案的作業內容。替代路徑所描述的可包括基本路徑之衍生及可能發生之錯誤等。一般來說，一個使用個案僅有一個基本路徑，但可以有多個替代路徑。
- 舉例來說，訂購一批貨物會有一個訂貨的使用個案，它包括好幾個對應的情節，例如一個一切都正常的情節、一個貨物不夠的情節、一個信用卡被拒絕的情節等。

使用個案圖之元件(5/8)

- 使用個案間的關係主要可以分為三種
 - 包含(Include)
 - 可視為一種 “Delegation” (委派) 或是 “Aggregation” (整合)，也就是說某一使用個案「A」會用到另一使用個案「B」，則稱A委派或整合B，關係之箭頭符號應由A指向B。
 - 延伸(Extend)
 - 可視為是某使用個案 (例如A) 在某情況時 (並非所有情況)，會被插入至另一使用個案的定義中 (例如B)，而形成一新的組合使用個案，則關係之箭頭符號應由A指向B。
 - 一般化(Generalization)
 - Generalization之關係意味著使用個案間之繼承關係。

使用個案圖之元件(6/8)

- 銀行匯票處理方式的使用個案：
 - a. 查驗密碼：要求使用者輸入密碼後，查核使用者是否為合法使用人員。
 - b. 掃描視網膜：確認使用者是否具有高優先權時，須利用攝影機將使用者眼睛影像截取後，比對其視網膜 影像資料是否為合法使用人員。
 - c. 確認使用者身分：要求使用者比對其身分是否為合法使用人員，正確時才可進入系統。

使用個案圖之元件(7/8)

d. 匯票追蹤：使用者欲追蹤特定匯票狀況時：

- (1)須 先核對使用者身分是否合法
- (2)為合法人員者便可進入系統查詢其指定匯票狀況。

e. 匯入匯票：操作員欲將匯票存入帳戶系統時

- (1)須先核對使用者身分是否合法
- (2)為合法人員者便可實施匯入一般程序，若該使用者具有高優先權利時，便可實施快速匯票處理。

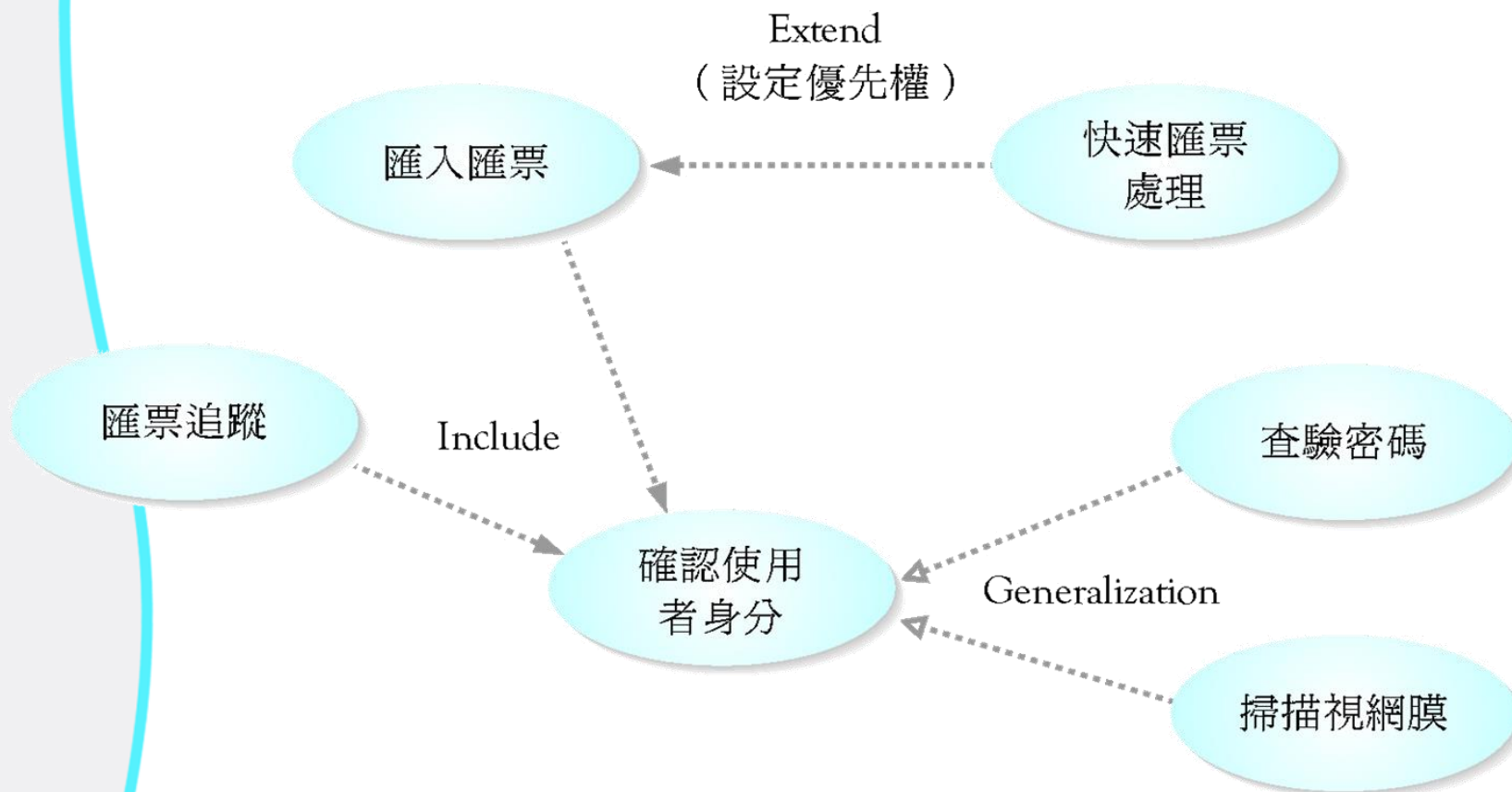
f. 快速匯票處理：直接快速將指定匯票匯入系統處理，排除其他限制因素。

- 主要事件流(1)：Include (確認使用者身分)，取得匯票匯入資料與確認的帳號，設定優先權，依權限處理匯票匯入。
- 主要事件流(2)：Include (確認使用者身分)，查核匯票的狀況。

使用個案圖之元件(8/8)

Figure

圖 4-4 使用個案關係圖範例



內聚力(1/9)

- 內聚力是一種衡量模組內部之工作相關程度之方法。換句話說，模組的內聚力是衡量模組完成一件單一旦定義清楚之工作的程度，也可以被用來衡量一個使用個案內部工作之相關程度。內聚力的種類大概可分為七種：
 - 功能內聚力(Functional Cohesion)
 - 順序內聚力(Sequential Cohesion)
 - 溝通內聚力(Communication Cohesion)
 - 暫時內聚力(Temporal Cohesion)
 - 程序內聚力(Procedural Cohesion)
 - 邏輯內聚力(Logical Cohesion)
 - 偶發內聚力(Coincidental Cohesion)

內聚力(2/9)

- 功能內聚力

- 功能內聚力指的是當一個模組（或使用個案）只做一件事情，亦即具有唯一的功能，是為功能型的內聚力。

檢查身分證
號碼的正確性

以異動檔更
新庫存主檔

計算營業稅

內聚力(3/9)

- 順序內聚力

- 順序內聚力是指模組內（或使用個案）具有多個功能或處理多件事情，且一項功能的輸出立即成為下一個功能的輸入，也就是共用相同資料，則此模組（或使用個案）具有順序內聚力。

讀取某數值 X

計算 X 之平方

將計算所得結果
顯示於螢幕上

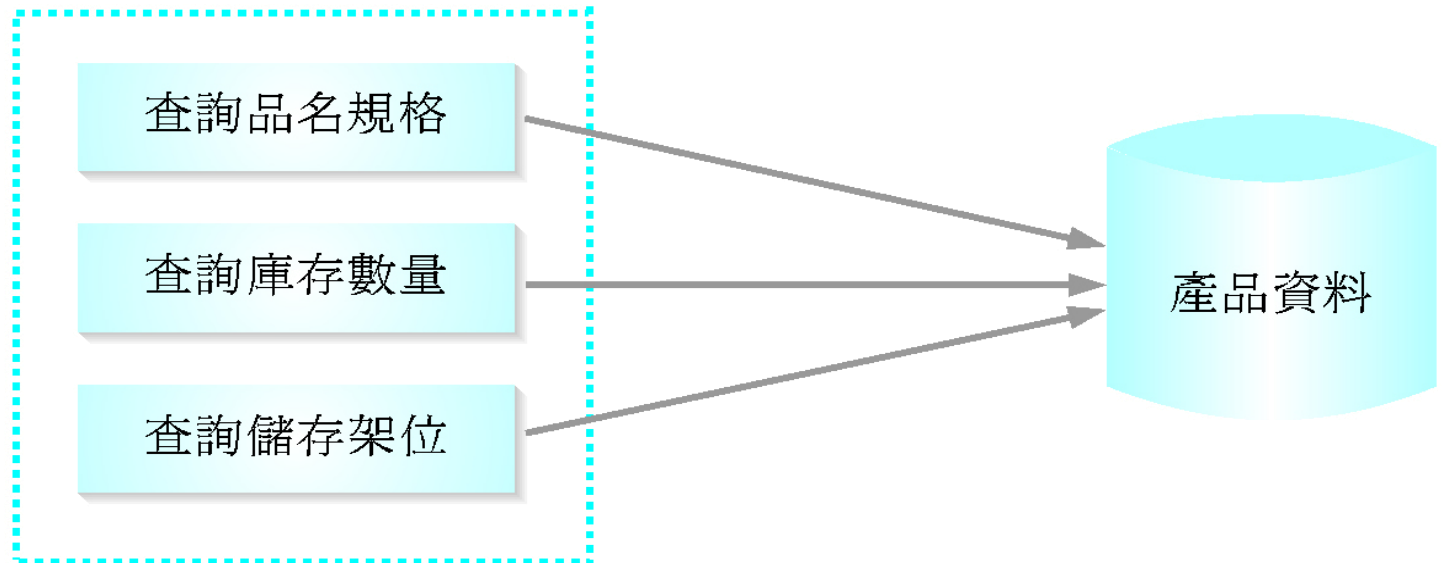


內聚力(4/9)

- 溝通內聚力

- 溝通內聚力是指模組（或使用個案）內具有多個功能或處理多件事情，且這些功能使用相同的資料(輸入)，但它們的執行順序沒有相關性

○



內聚力(5/9)

- 暫時內聚力

- 模組（或使用個案）內具有多個功能或處理多件事情，但是這些功能僅僅在時序上有所關聯，也就是必須在同一時間內執行完成，所以這種模組（或使用個案）具有暫時內聚力。

設定日期格式

指定資料檔路徑

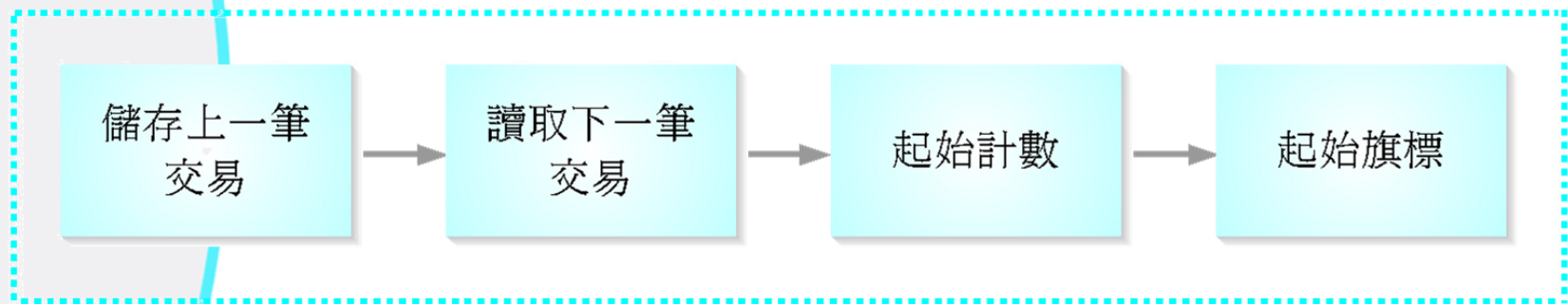
清除所有變數

設定變數起始值

內聚力(6/9)

- 程序內聚力

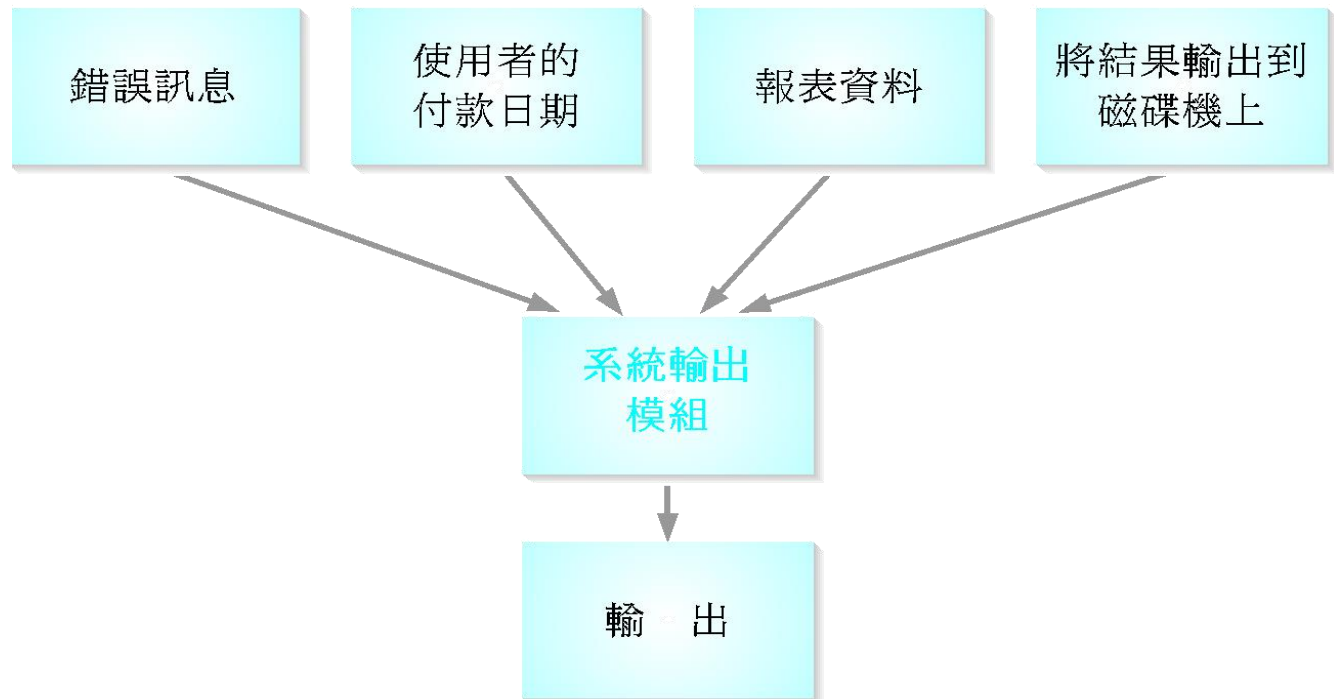
- 程序內聚力是指模組（或使用個案）內具有多個功能或處理多件事情，這些功能必須按照一定的順序來執行，且不共用資料，這些功能群集在一個模組（或使用個案）內僅為了確保它們的執行順序，則此模組（或使用個案）具有程序內聚力。



內聚力(7/9)

• 邏輯內聚力

- 邏輯內聚力是指模組（或使用個案）內具有多個邏輯上相關聯的功能。



內聚力(8/9)

- 偶發內聚力

- 若一個模組（或使用個案）內部要執行好幾件工作，且每一件工作都不相干，則該模組（或使用個案）具有偶發內聚力。在設計時，偶發內聚力應盡量避免，例如可將個別的工作分別獨立出來自成一個模組（或使用個案），使各模組（或使用個案）具有功能內聚力。

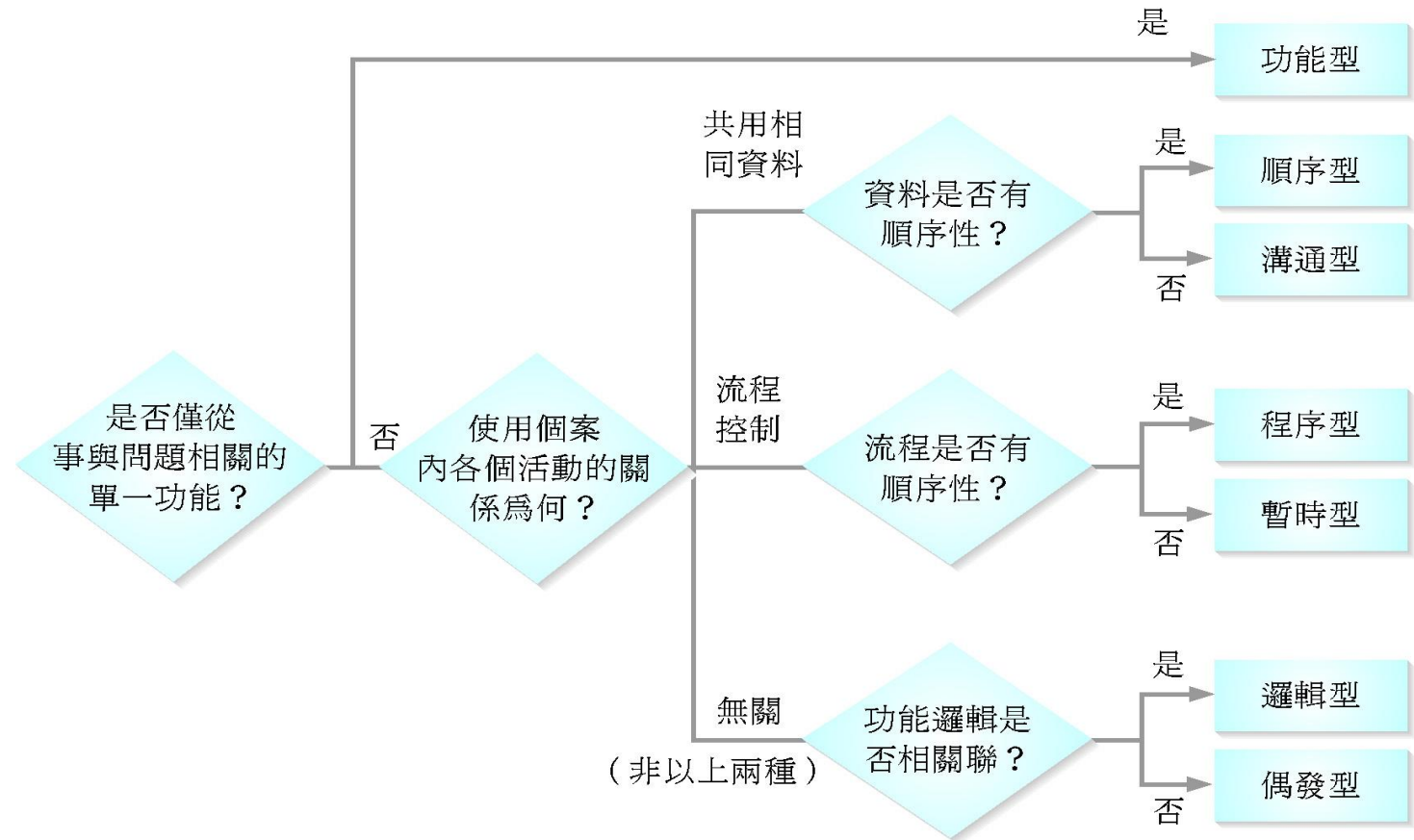
列印資產負債表

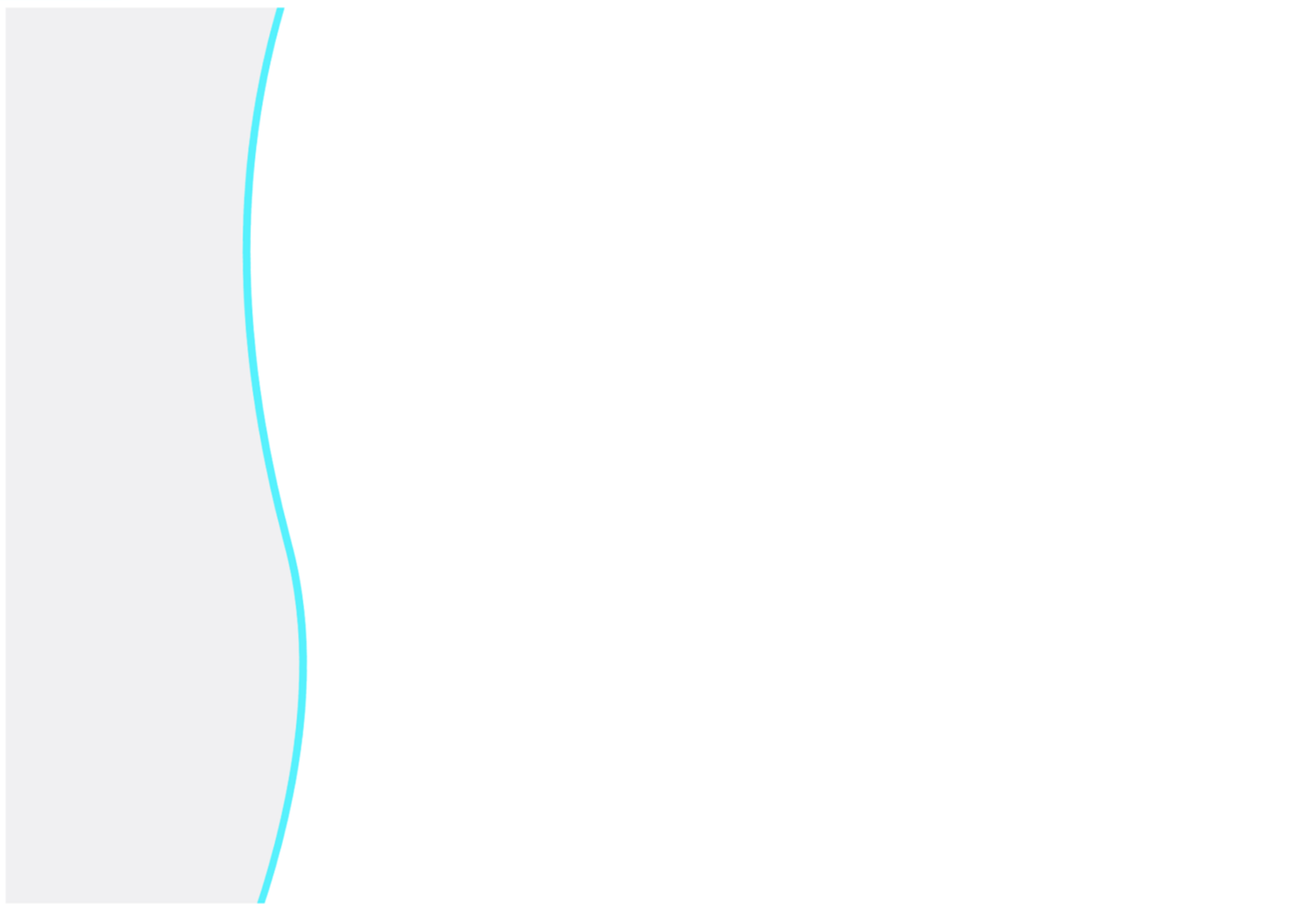
計算所得稅

查詢庫存量

內聚力(9/9)

- 模組內聚力之判定決策樹





使用個案圖建構步驟與準則-找出行為者(1/2)

- 找出使用個案圖行為者之方式很多，例如：
 - 可從組織中的成員來找。
 - 可從問題的陳述、使用者領域之專家或檢視蒐集到的文件，也可從其中之名詞、代名詞與名詞片語等，找出合乎行為者定義的人或相關系統。
 - 從環境圖可找出行為者，因環境圖中之外部實體與系統有互動關係，所以這些外部實體將成為系統中使用個案的行為者。

找出行為者(2/2)

- 亦可以從幾個方向來找出初步與系統相關的行為者，例如：
 - 所開發的系統用在什麼地方？
 - 這個系統會使用一些外在的資源嗎？
 - 誰對系統中的某些項目有興趣？
 - 誰是系統的資訊提供者？
 - 誰是系統的資訊使用者？
 - 誰是此一系統的操作者？
 - 誰會支援及維護這個系統？
 - 一個行為者扮演許多不同的角色嗎？
 - 或是多個行為者扮演相同的角色呢？

找出使用個案(1/3)

- 先找出某一行為者，再逐一檢討行為者，以找出其所參與的每一個使用個案。完成後，再以此方式檢討其他行為者，直到檢討完其他所有行為者為止。
- 一般來說，行為者使用系統時，必先啟動一個事件，之後會有相關事件之回應，直到完成使用系統的目的為止，將這一系列事件有組織地集合起來便成為一個使用個案。

找出使用個案(2/3)

- 一個使用個案是使用者透過介面要求系統所做的一連串相關的事件流（參考內聚力），應有起點亦有終點。
- 從使用者之角度來看，每一個使用個案就是一個案例，可完成某一功能。

找出使用個案(3/3)

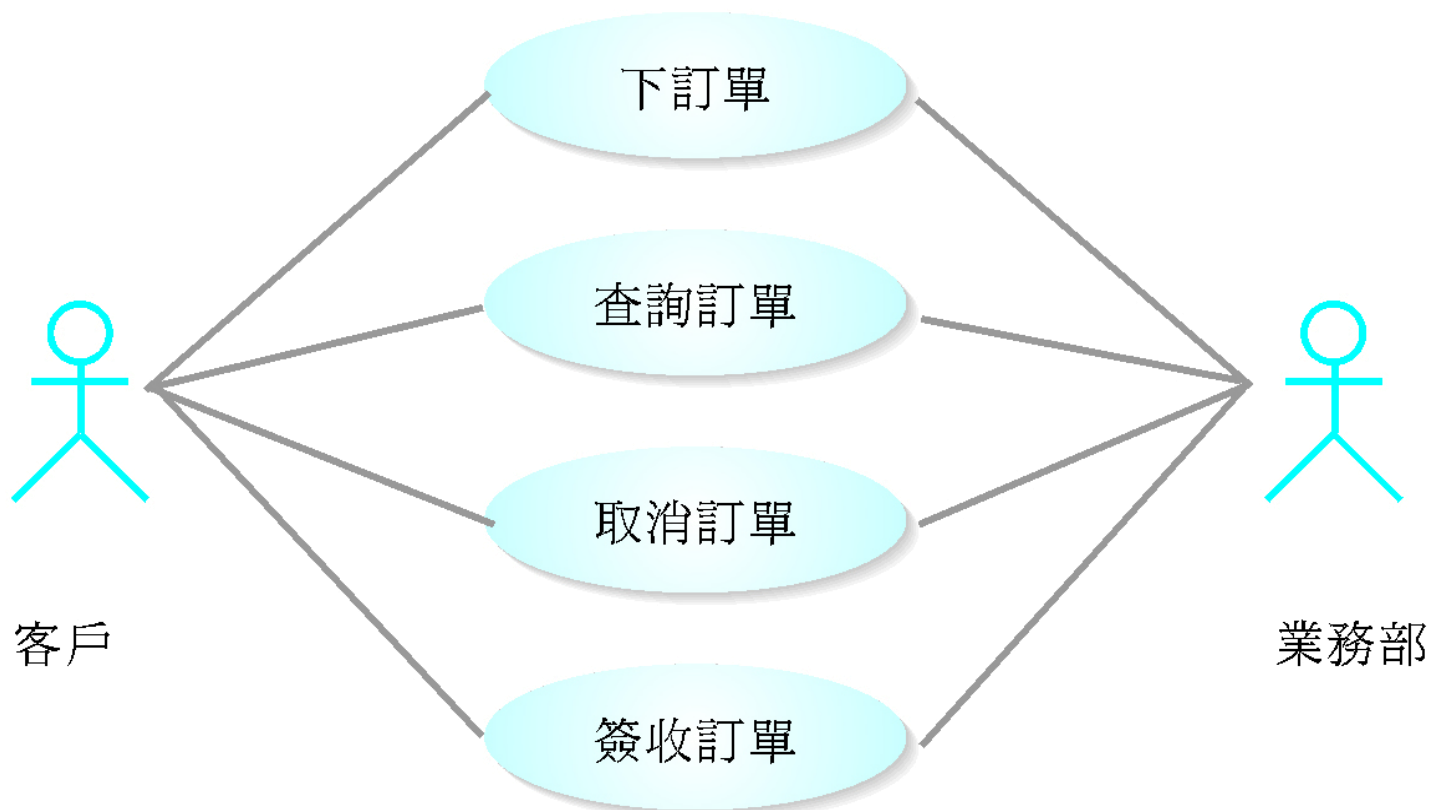
- 此外，也可以從一些方向來找出初步與系統相關的使用個案，例如：
 - 行為者做什麼工作？
 - 行為者是否會建立、儲存、修改、移除、或讀取此系統資訊？
 - 行為者是否會通知系統有關外在環境的改變？
 - 若系統發生任何狀況，是否會通知行為者？
 - 使用者如何支援及維護這個系統？

描述使用個案(1/6)

- 使用個案描述是從使用者之觀點描述使用者達成某目標、目的或功能之作業行為，此時應著重企業處理或功能描述，而不應涉及電腦化之操作。
- 使用個案的描述方式有兩種：
 - 自然語言
 - 事件條列式

描述使用個案(2/6)

- 使用個案範例



描述使用個案(3/6)

- 以取消訂單為例，該使用個案之描述如下：
 - (1) 以自然語言方式描述
 - 當業務部收到取消訂單的要求時，先找到系統中的訂單並取消之。然後，再將這個要求送到會計系統去更正客戶的訂單。
 - (2) 用以事件條列式描述
 - a. 業務部收到客戶取消訂單的要求。
 - b. 業務員輸入訂單編號，並按下尋找的按鈕。
 - c. 系統展示所尋得的訂單，並將訂單標示為「已取消」。
 - d. 會計系統被告知更正客戶的訂單。

描述使用個案(4/6)

- 以自然語言之文句敘述來描述使用個案時，若使用個案中包含許多替代路徑，則很容易變得難以瞭解，因此建議用一些結構化之寫作方式來描述較佳。也就是說，以事件條列式描述優於自然語言方式。
- 使用個案主要描述行為者達成某目標之行為，故事件描述之內容應包括：
 - (1) 工作項目與流程
 - (2) 工作內容
 - (3) 資料特性

描述使用個案(5/6)

- (1) 工作項目與流程
 - 事件描述應確定每一事件有哪些工作、工作間之流程與控制等，這些工作在系統分析與設計階段將被轉換成電腦化的操作項目。
- (2) 工作內容
 - 工作內容描述每一工作項目之細節，包括事件所傳送資料之每一項目(Item)或欄位(Field)及其來源，如送貨單之客戶名稱、地址、電話號碼。
 - 若有計算公式亦應列出，若涉及其他資料之異動更應詳加描述。

描述使用個案(6/6)

– (3) 資料特性

- 對於工作內容之資料應進一步描述其特性，資料特性應包含：
 - 資料型態(Data Type)，如字串或數字與長度。
 - 鍵(Key)，如主鍵或副鍵。
 - 排序(Order)，如以ASCII CODE作為排序順序。
 - 資料範圍(Range)，如設定數值必須小於100。
 - 資料出現次數(Occurrence)，如一張送貨單有六筆送貨明細。
- 以上這些描述將有助於物件或類別之屬性描述與資料庫之設計。

找出使用個案間之關係(1/7)

- 完成使用個案之描述後，可從使用個案之描述看出：
 - (1) 哪些個案之描述中有共同的行為，便可將之獨立成一個新的使用個案及修改原個案之描述，並考慮新使用個案與原使用個案間之關係，例如這可能是 **Include**或**Extend**。
 - (2) 哪些個案一定會用到某些個案，這是**Include**關係。
 - (3) 哪些個案僅在某些情況下會用到某些個案，這可能是**Extend**關係。
 - (4) 哪些個案某部份與另一個案相同且另外多出一些功能時，這可能是**Generalization**關係。
- 本節將以門禁系統為例，說明**Include**與**Extend**兩種關係，該系統包含以下八個使用個案(如圖4-14)：

找出使用個案間之關係(2/7)

- 1. 進入未啟動安全監控的門。
 - 當門禁系統未啟動，學生及管理員可自由的經由這個門進出。
- 2. 進入安全監控門。
 - 學生及管理員經由這個門進入的方式：
 - (1) 在數字鍵盤上輸入進入密碼
 - (2) 經由這個門進入
 - (3) 進入以後關上門

找出使用個案間之關係(3/7)

- 3. 改變進入密碼。
 - 管理員改變進入密碼的方式：
 - (1) 按下控制板上的進入密碼更改按鈕
 - (2) 在數字鍵盤上輸入安全密碼作為認證
 - (3) 在數字鍵盤上輸入新的進入密碼
 - (4) 在數字鍵盤上重新輸入新的進入密碼作為確認

找出使用個案間之關係(4/7)

- 4. 啟動門禁系統。
 - 管理員啟動門禁系統的程序：
 - (1)按下控制板上的啟動按鈕
 - (2)在數字鍵盤上輸入安全密碼作為認證
 - (3)未啟動指示燈熄掉
 - (4)啟動指示燈打開
 - (5)將門鎖上
 - 3. 改變進入密碼。

找出使用個案間之關係(5/7)

- 管理員改變進入密碼的方式：
 - (1) 按下控制板上的進入密碼更改按鈕
 - (2) 在數字鍵盤上輸入安全密碼作為認證
 - (3) 在數字鍵盤上輸入新的進入密碼
 - (4) 在數字鍵盤上重新輸入新的進入密碼作為確認
 - (5) 打開門鎖

找出使用個案間之關聯(6/7)

- 6. 輸入進入密碼。
 - 學生及管理員先按 “Enter” 鍵，再輸入八碼的進入密碼。
 - 門禁系統每按一個按鍵就發一聲「嗶」，並確認輸入密碼。
- 7. 輸入安全密碼。
 - 學生及管理員先按 Enter鍵，再輸入八碼的安全密碼。門禁系統每按一個按鍵就發一聲「嗶」，並確認輸入密碼。
- 8. 啟動警報器。
 - 如果門打開太久或者當門禁系統啟動而門沒有被關上，警報器發出響聲。管理員輸入安全密碼解除警報器。

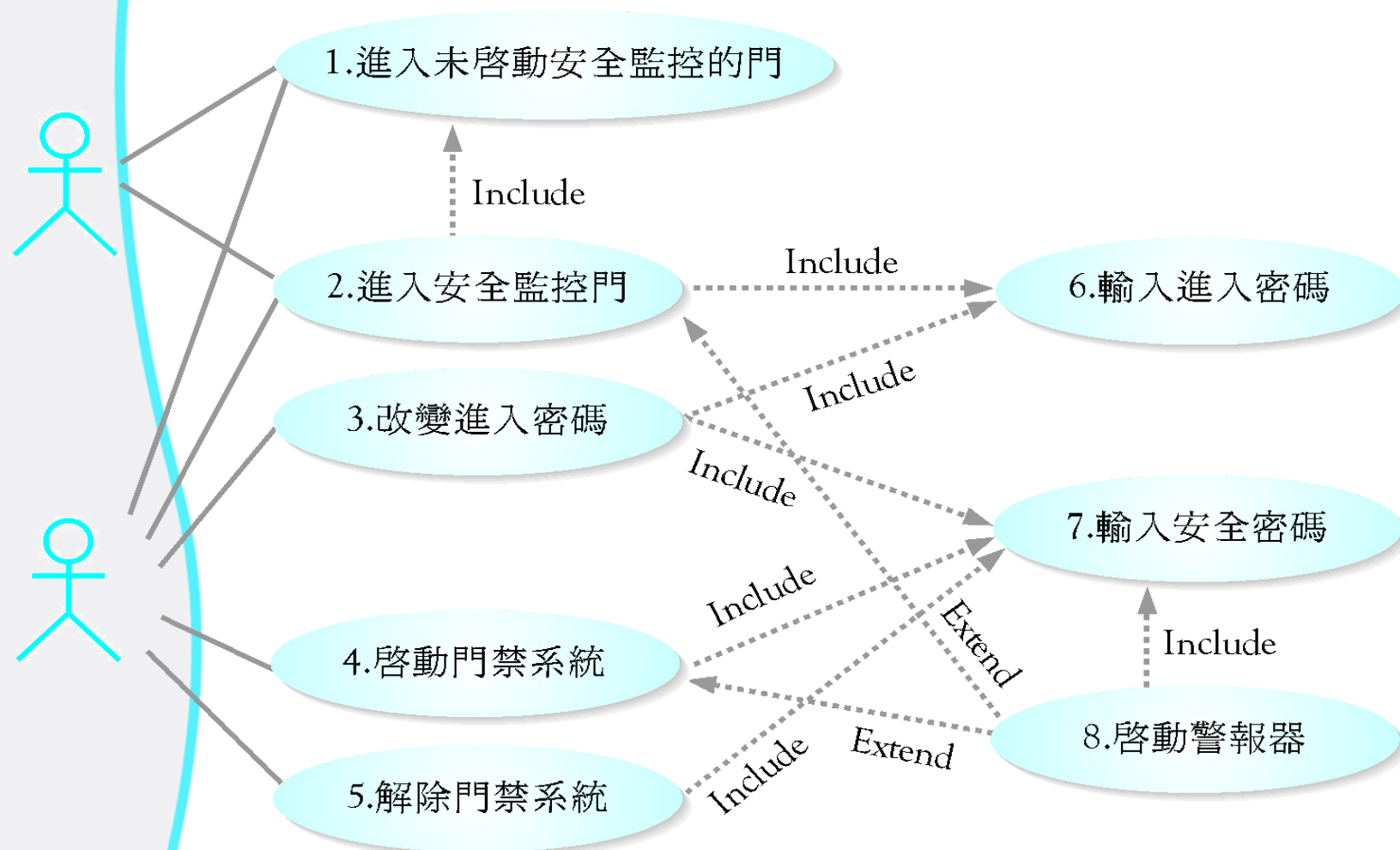
找出使用個案間之關係(7/7)

- Extend可視為兩使用個案間之繼承關係，也就是某使用個案的定義在某情況發生時會被插入至另一使用個案的定義中，而形成一新的組合使用個案。
- 以圖4-14為例，使用個案「啟動警報器」與「進入安全控制門」間有Extend 之關係，且是從前者 Extend 至後者。
- Include 是一種使用關係，如使用個案 “啟動警報器” 與 “輸入安全密碼” 間有 Include之關係，且是前者Include 後者。

繪製使用個案圖

- 完成上述工作後，最後是繪製使用個案圖繪製步驟為先繪出將所有行為者與使用個案，將行為者與使用個案間有互動者以互動符號連結，接著再確定使用個案間之關係，並以適當之關係符號連結。
- 完成使用個案圖後，必要時可以用活動圖來表達使用個案間之執行順序（例如循序或同步），以清楚表達系統之外部（巨觀）作業行為，因為這些關係在使用個案圖上不易表達。

門禁系統使用個案圖



藍圖

- 藍圖主要用於表達使用個案描述或活動圖中，有關之表單、介面等各項資訊需求之名稱、展示位置、格線、圖表與說明等，這些資訊常無法在使用個案描述或活動圖上具體地表達，因此須另以藍圖做進一步地表示。

資料詞彙(1/2)

- 資料詞彙進一步說明藍圖所無法表達之內容，如資訊之長度、型態、格式、公式、規則、範圍與限制等，並分別舉例說明之。

資料詞彙(2/2)

- 資料詞彙之內容項目除了藍圖中之欄位編號與名稱外，欄位資料之長度與型態、資料是否唯一、資料產生之規則、格式、範圍、公式等資訊，均有助於系統分析與設計，因此均可列入資料詞彙中，其形式可表達如下表：

欄位編號	欄位名稱	長度 / 型態	鍵	規則 / 格式 / 範圍 / 公式	範例
...

活動圖

- 活動圖是一種塑模工具，它可被用於表達一個物件、一個使用個案、許多使用個案間或一個系統在其生命週期中之循序或同步的操作(Operation)、作業流程(Workflow)或行為，例如在其生命週期中之所有活動及其轉換關係。
- 在UML 2.0版之後，循序圖之表達能力大幅提升，已可表達上述複雜行為，因此活動圖在這方面之工作已可由循序圖取代，但有關行為者與系統之互動行為、使用者之作業行為或系統之作業行為描述等方面，活動圖仍是一種很好的塑模工具。

活動圖之元件(1/6)

- 活動圖之主要元件：活動與轉換，兩者之關係與表達如圖所示。



活動圖之元件(2/6)

- 活動包括開始、一般活動、結束流程與結束活動；轉換包括一般轉換、分岔、結合、分支與合併等，這些元件所代表的意義分別說明如下：
 - 1.開始：開始(Initial)用於描述一連串活動的起點，以
一實心圓形表示。
 - 2.結束活動：結束活動(Activity Final)用於描述一連串
活動的結束（或終點），以一實心圓形外加一空心
圓圈表示。
 - 3.結束流程：結束流程(Flow Final)用於描述某一個流
程的結束（或終點），以一空心圓形內加一個
“x”
表示。

活動圖之元件(3/6)

- 4.活動：活動(Activity)是真實世界的一個動作處理、一組動作程序或是可執行的一段副程式。活動以一圓角矩形表示，內部表達名稱或其他行為描述。
- 5.條件或註記：條件(Condition)可分前提條件(Pre-condition)、結束條件(Post-condition)，可用於描述某一個活動的起始、結束（或終點）、分支或合併等的條件；註記(Note)可以包括活動之輸入與輸出、以一矩形外加右上角有折角表示。
- 6.活動物件：活動物件(Activity Object)可用於描述與某一個活動有關的狀態、輸入與輸出或資料儲存等，一矩形表示。

活動圖之元件(4/6)

- 7.轉換：與狀態圖的轉換相同，即是當一個動作或活動完成時直接到下一個活動的控制，與狀態圖的轉換不同的是此轉換只有[成立條件]的控制，以「箭頭」旁註 [成立條件] 來表示。
- 8.分岔：分岔(Fork)是用於表達當轉換發生後，有兩個 或兩個以上之平行活動發生的情況。分岔之表達符號是以一橫向黑實線條，外加一條流入之垂直箭頭與多條流出之垂直箭頭表示。
- 9.結合：結合(Join)是用於表達平行活動結束之情況，表達符號與分岔類似是以一橫向黑實線條，外加多條流入之垂直箭頭與一條流出之垂直箭頭表示。






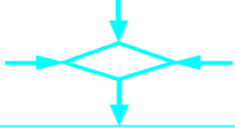

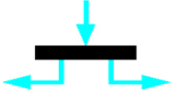

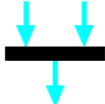

活動圖之元件(5/6)

- 10.分支：分支(Branch)是用於表達當轉換發生後，有多個選擇路徑，但僅能依條件選擇其中一個路徑執行之。分支之表達符號是以一菱形，外加一條流入菱形之箭頭與多條流出菱形之箭頭表示。
- 11.合併：合併(Merge)是用於表達有多個路徑匯集於某點，之後再依條件往下一個路徑執行。合併之表達符號與分支相似，是以一菱形外加多條流入菱形之箭頭與一條流出菱形之箭頭表示。
- 12.責任區：責任區(Swimlane)是以游泳池的水道方式來表達活動圖中，哪些活動是由誰、哪個部門、類別或元件負責的一種方式，每一水道代表一個負責的人、部門、類別或元件。

活動圖之元件(6/6)

Figure

圖 4-16 活動圖之元件與表達符號

活動圖之元件	符號	活動圖之元件	符號
開始活動		轉換[條件]	
結束活動		轉換 分支	
結束流程		轉換 合併	
一般活動		轉換 分岔	
條件或註記（例如限制、值或決策選項等）		轉換 結合	
活動物件（狀態、使用者介面、資料儲存或其他物件等）			

活動圖之建構步驟與原則(1/3)

- 活動圖之建構步驟包括：（1）找出活動，必要時也需找出執行該活動之實體；（2）找出活動間之轉換；（3）繪製活動圖；（4）精鍊活動圖。

建構活動圖可參考下列原則：

- 1.從使用個案描述或類別之操作描述中找出相關的活動與轉換。
- 2.由活動圖之上方或左上方以「開始」畫起，接著依行為者、系統或物件之操作、行為或作業流程等活動發生之順序畫出活動及其間之轉換，而在最後的活動之後以「結束」表達之。

活動圖之建構步驟與原則(2/3)

3. 遇到有平行處理或多執行緒的活動時以分岔描述之，此時在分岔之前會有一個進入轉換(Incoming Transition)，在分岔之後會有數個離開轉換(Outgoing Transition)。
4. 有分岔就必須有結合，在所有分岔出去的平行處理的活動執行完畢後需有結合。此時在結合之前會有數個進入轉換，在結合之後會有一個離開轉換。
5. 遇到數個有「成立條件」的擇一執行後續活動時，以分支來表達，此時在分支之前會有一個進入轉換，在分支之後會有數個具有互斥條件的離開轉換。

活動圖之建構步驟與原則(3/3)

- 6.合併用來描述之前以分支為開端的條件式活動的結束，此時在合併之前會有數個輸入轉換(Input Transition)，在合併之後會有一個輸出轉換(Output Transition)。
- 7.可以用責任區方式表達哪些活動是由誰、哪個部門、類別或元件負責，並將這些活動放置於同一「水道」內。

應用使用個案模式應注意事項

- 由於使用個案模式具有許多的好處且普遍被應用，它已成為物件導向技術的基礎部份，且已被整合入主要的物件導向開發方法中，但使用個案模式仍有許多限制。
- 以下提供一些與使用個案模式誤用有關之情況：
 - 使用個案模式不是真正物件導向的工具。
 - 使用個案圖之每個元素缺乏正式的定義。
 - 使用個案是由一個或多個行為者與系統之間的交互作用定義而成。然而，並不是所有的系統都一定要由行為者輸入才有反應。

使用個案模式的文件樣版(1/2)

- 完成使用個案塑模後，有很多相關的文件需加以整理，以便日後之系統維護或修改，這些文件至少應包括完整之使用個案圖，及各個使用個案文件之記錄。
- 使用個案之文件至少應包括：
 - (1)使用個案名稱：一個使用個案有唯一的名稱，命名要簡短且有意義，通常是以一個名詞或片語來描述。
 - (2)行為者：應列出參與該個案之所有行為者。

使用個案模式的文件樣版(2/2)

- (3)**目標**：應列出該使用個案完成後須達成之目標
- (4)**前提與結束狀態**：前提是指這個使用個案開始前必須成立的條件，而結束狀態是指這個使用個案結束之後必須成立的條件。
- (5)**Generalization、Extend與Include之使用個案**：列出該使用個案所Extend與Include及被Extend與Include之使用個案。
- (6)**使用個案描述**；描述其一系列之事件。

使用個案文件樣版

使用個案名稱

〈列出使用個案之名稱〉

行為者

〈列出所有與這個使用個案有互動之行為者〉

目標

〈列出該使用個案完成後須達成之目標〉

前提

〈描述這個使用個案開始前必須成立的條件〉

結束狀態

〈描述這個使用個案結束之後必須成立的條件〉

Extend 之使用個案

〈列出與這個使用個案有 Extend 關係者〉

Include 之使用個案

〈列出與這個使用個案有 Include 關係者〉

Generalization 之使用個案

〈列出與這個使用個案有 Generalization 關係者〉

一系列之事件

〈列出基本路徑和替代路徑〉

需求分析之重要工作與文件樣版

需求分析文件樣版

- 一、問題描述
- 二、新系統目標
- 三、新系統限制
- 四、使用者需求
 - (一)使用個案圖
 - (二)使用個案(1)
 - 活動圖(1)
 - 藍圖 1-1
 - 資料詞彙 1-1
 -
 - 藍圖 1-m
 - 資料詞彙 1-m
 - (三)使用個案(2)
 - 活動圖(2)
 -
 -

小結(1/2)

- 需求分析主要是擷取與表達使用者之巨觀需求，這包括欲電腦化之環境、作業處理、輸出與輸入所需之資訊或表單與系統之主要功能等。
- 需求分析階段對問題領域之瞭解範圍應盡可能地廣泛，到了分析與設計階段再縮小到專案範圍，如此有助於對新系統之瞭解與規劃。

小結(2/2)

- 以目前之開發工具來說，需求分析工作約占整個專案時間的10%左右，且需求分析往往無法在一個階段完全做完，常在分析與設計階段仍有需求分析工作之進行。當然，系統分析師之專業知識與經驗，對需求分析之成效有密切之影響。