

Data Flow Diagram

資科系
林偉川

DFD

- 一般的軟體，最基本上可分為“輸入”、“處理”、“輸出”三部份
- DFD圖的精神就在於資料流程的規劃、各處理單元的動作、並且可以作更深一步的切分。

DFD之定義

- DFD圖有幾個符號：



外部儲存體 (External Entity)



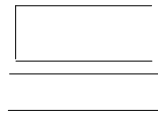
轉換程序 (Process)

執行輸入資料處理轉換成輸出資料的單元



資料流 (Data Flow)

用以連接不同的程序，以表示資料傳送的方向上方必有資料標籤

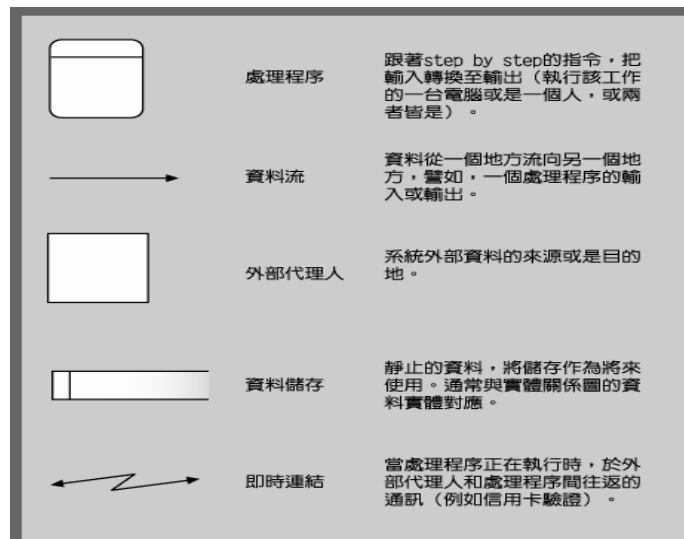


資料儲存所 (Data Store)

資料儲存的地方，箭頭表示資料的來源及輸出的方向

3

資料流程圖的符號



4

資料流程圖

- 一個圖形模型，它顯示資訊系統的全部主要需求
 - 輸入與輸出
 - 處理程序
 - 資料儲存
- 容易閱讀與了解

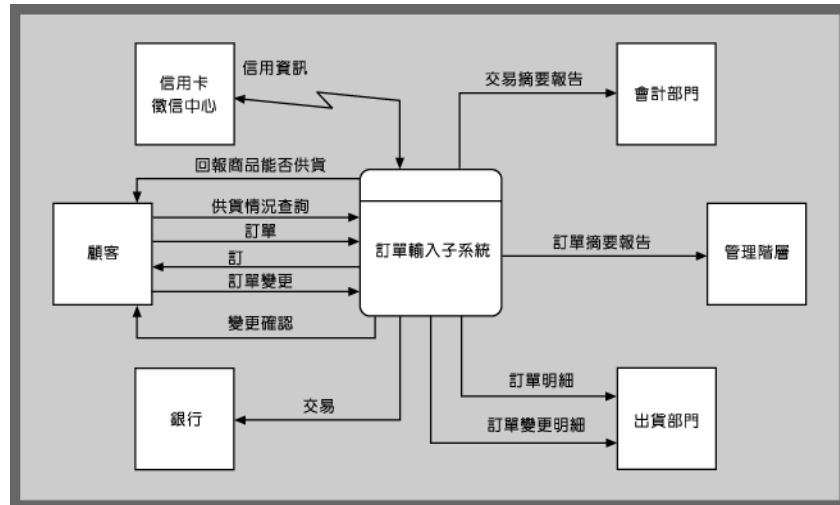
5

環境圖 (Context Diagram)

- DFD的第一張圖，有時候稱為圖0
- 摘要所有處理活動的DFD
- 描述系統最抽象之觀點
- 顯示系統範圍
- 系統的範圍藉著單一處理程序與外部代理人表示

6

訂單輸入子系統的環境圖



7

DFD 片段

- 表示在**單一處理程序**符號中，系統對於一個**事件的回應**
- **獨立的模型**
- 重點放在系統的**單一部份**上
- 只顯示需要**回應事件**的**資料儲存**

8

事件分割系統模型

- 一個塑模系統需求的 DFD，它使用**單一的處理程序**給系統與子系統中的各個**事件**
- 可分解內容的**層級**
- 可分解為**更詳細的 DFD 分段**

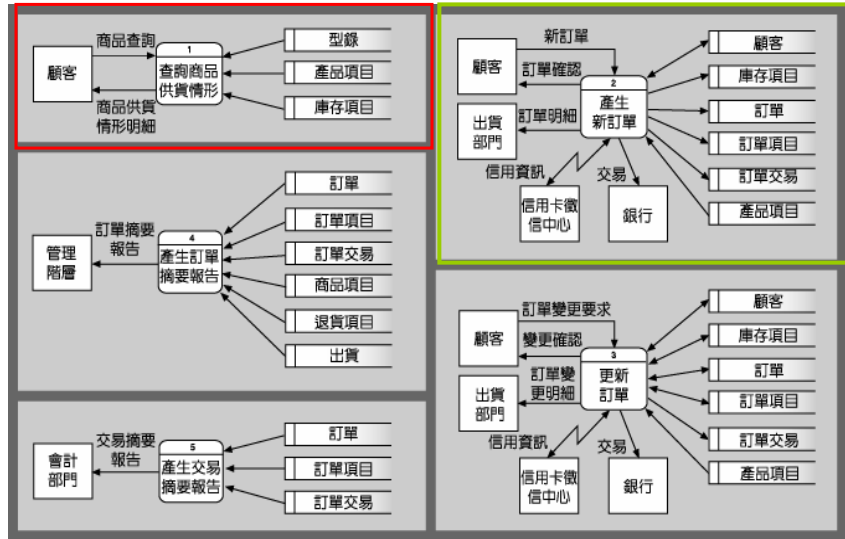
9

分解 DFD 分段

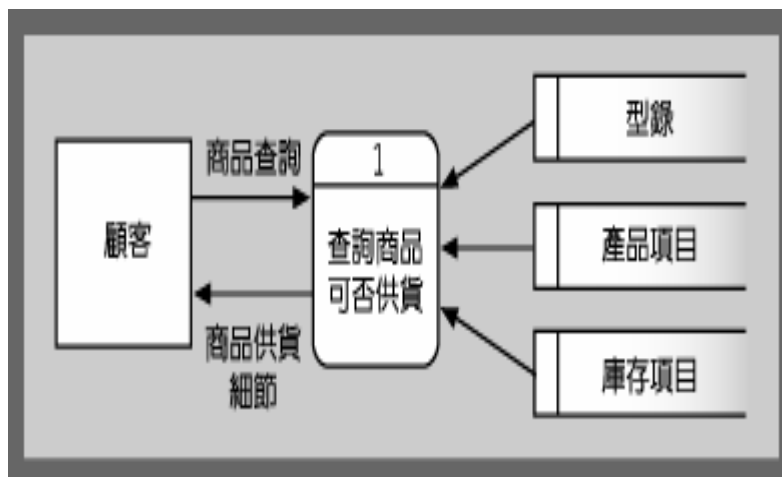
- 有時候需要對 DFD 分段做**更詳細的探索**
- 分解為**子處理程序**提供**更詳細的內容**
- **編號結構不等於執行的順序**

10

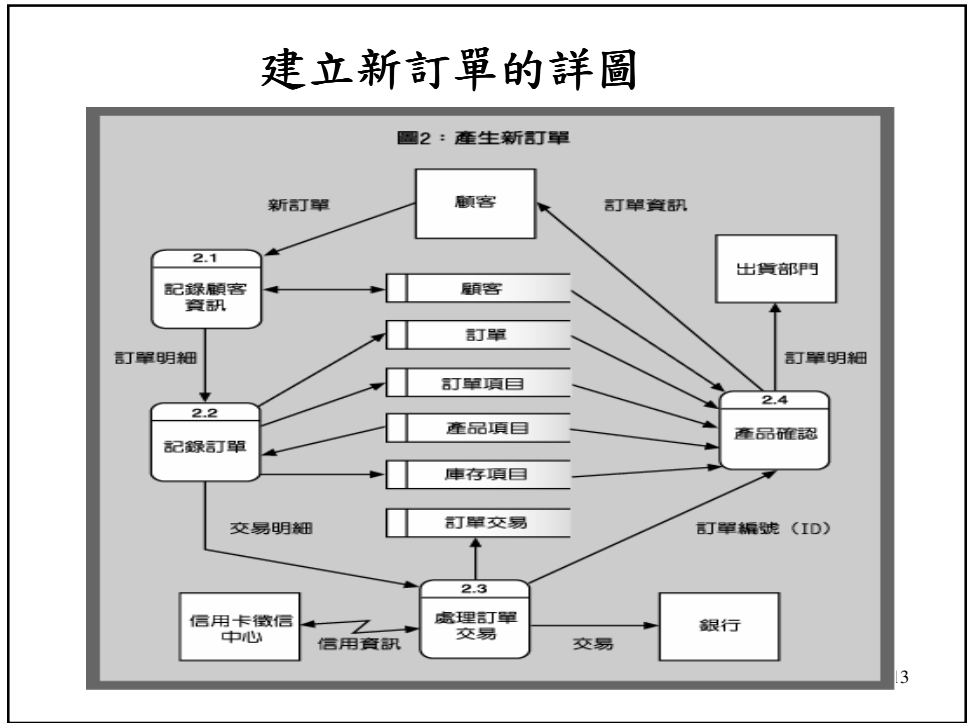
訂單輸入子系統的 DFD 分段



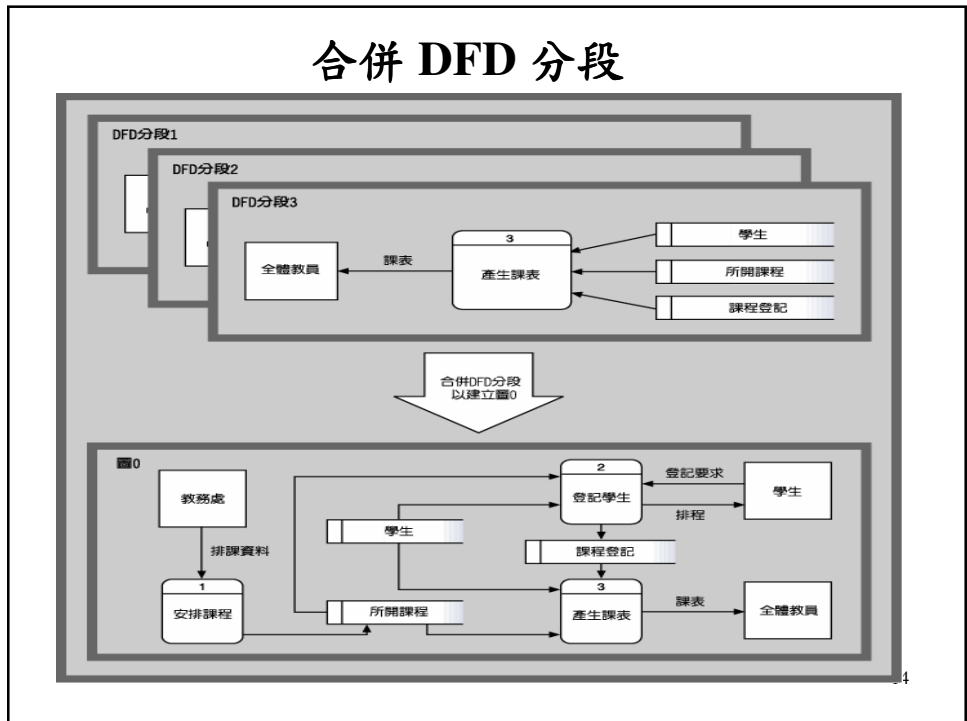
個案中的 DFD 分段



建立新訂單的詳圖



合併 DFD 分段



實體與邏輯 DFD

- 邏輯模型
 - 假定系統是在完美科技的環境下實作
 - 無法解說系統是如何實作的
- 實體模型
 - 描述實作技術的假設
 - 在分析的最後階段或設計的初期開發

15

評估 DFD 的品質

- 可讀性
- 內部一致性
- 正確表示系統內容
- 降低資訊超載
- 將需要的介面數減到最少

16

資料流的一致性問題

- ◆在處理程序與它的處理程序分解間資料流內容的差異
- ◆資料流出卻沒有相對應的資料流入
- ◆資料流入卻沒有相對應的資料流出
- ◆造成不平衡的 DFD

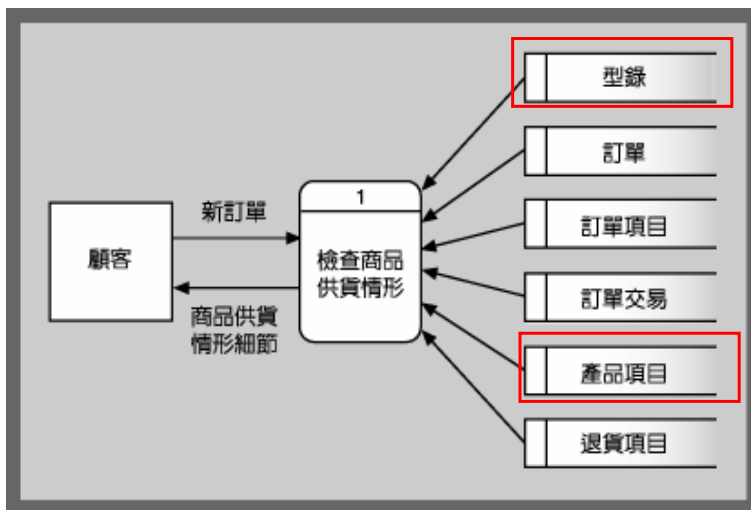
17

一致性規則

- 全部流入處理程序的資料必須流出該處理程序，或是被用來產生流出該處理程序的資料
- 全部流出處理程序的資料必須已流入該處理程序，或是從流入這個處理程序的資料所產生

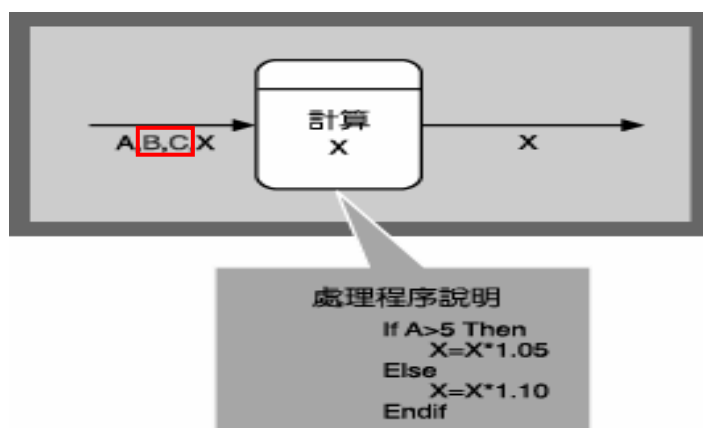
18

不必要之資料輸入：黑洞(因無輸出)



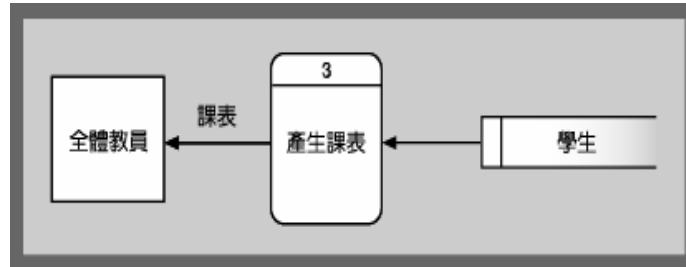
19

不必要之資料輸入的處理程序



20

不可能資料輸出的處理程序：奇蹟(不可能產出)



21

不可能之資料輸出的處理程序



處理程序說明

```
If A>5 Then  
  Y=100  
Else  
  Y=250  
Endif
```

22

DFD 元件的說明文件(Data Dictionary)

- ◆最低層級的處理程序都需要詳細描述
- ◆需要描述資料流的內容
- ◆需要以資料元素描述必要的資料儲存
- ◆需要描述每個資料元素
- ◆有不同選項可以定義處理程序

23

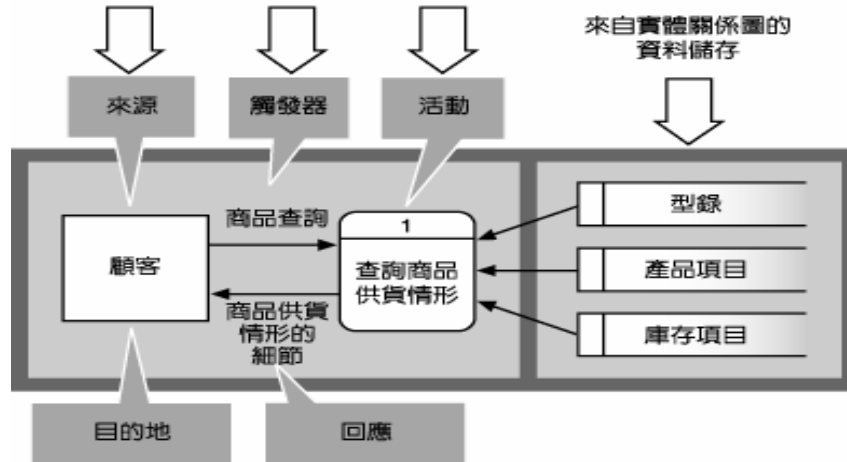
DFD之資料字典定義

- 資料字典(Data Dictionary)：用於說明DFD中Data Flow的內涵架構，用來定義資料流中的各部分（與DFD須一起考慮）符號： $=$ 表“全等”， $+$ 表“和”， $[]$ 表“其中之一”， $\{ \}$ 表“重覆”， $()$ 表“可有可無”， $*$ 表“說明部分”。其依賴資料流、資料元素、檔案及處理程序之分類編製各項目。
- 結構化英語(Structured English)：用於說明DFD中Process的詳細邏輯，由一些簡單的字彙和文法組成，含有三種結構：順序結構、選擇結構與重複結構。也就是一種擬似碼(Pseudo-Code)，其代表的是控制流程！

24

DFD 整合了事件表與 ERD

來自事件表中關於外部事件
代理人、資料流程、處理
程序的資訊



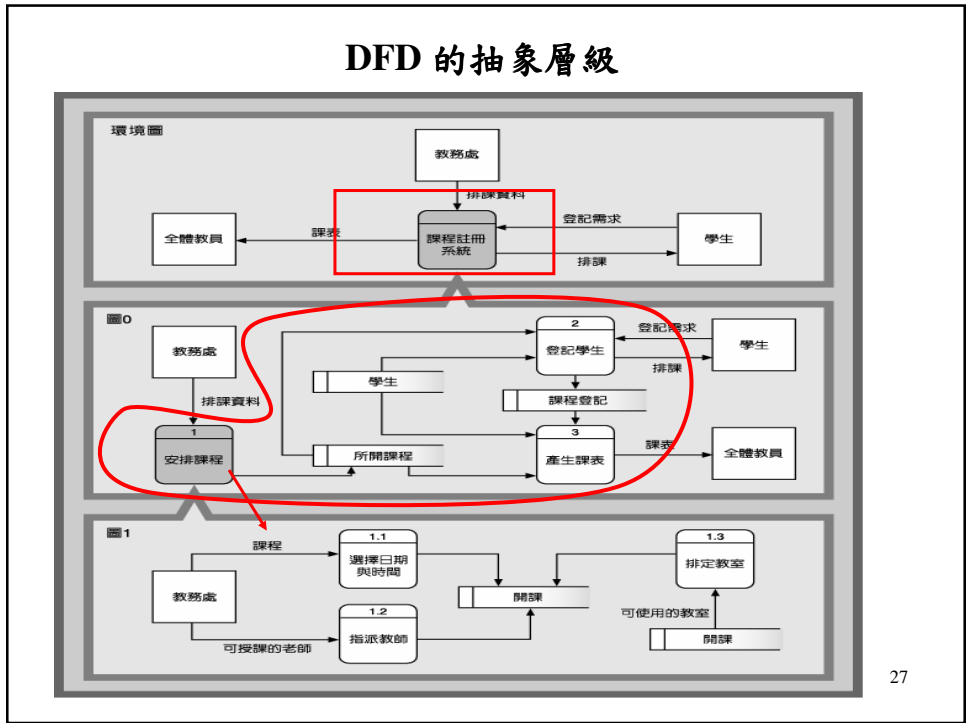
25

DFD 與抽象的層級

- DFD 可以分解為**其他的圖表**提供不同的**詳細程度**
- 層級愈高的內容愈一般性(數字愈小)
- 層級愈低的內容愈詳細(數字愈大)

26

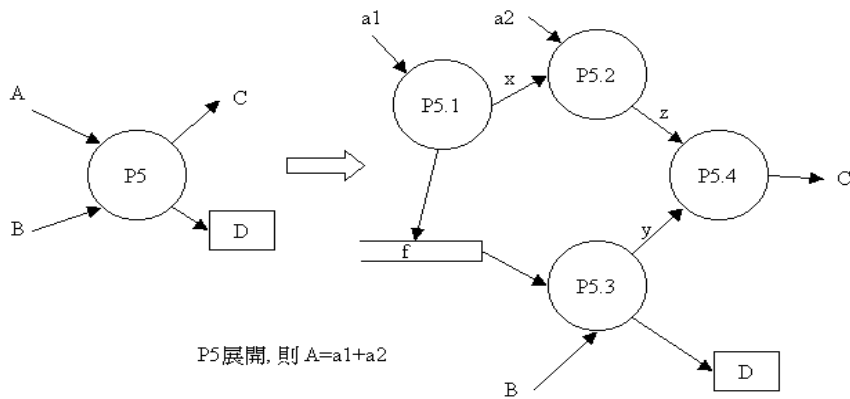
DFD 的抽象層級



27

DFD之檢驗

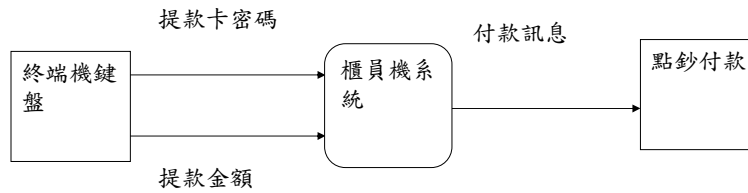
- DFD資料流一致性檢驗



28

DFD 範例

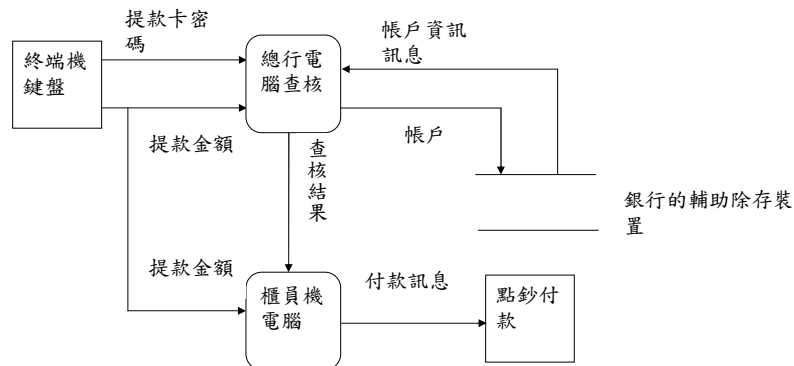
- 舉一個銀行自動櫃員機的 DFD 圖



29

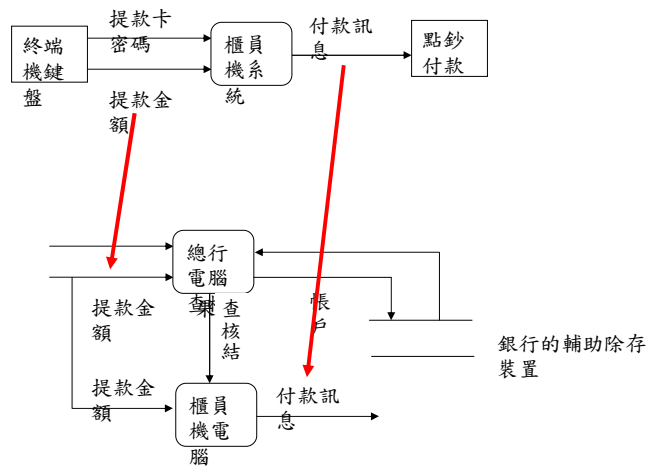
DFD 範例

- 這是最初步且未經細部規劃的，一個雛型已經產生了。下一步工作是逐步的將 DFD 圖分解成細節。



30

DFD 範例



31

DFD之模組

- 一個模組沒有什麼好限制的，也沒什麼標準可以用來明確訂定模組該有多大大小。在某些軟體專案可能會用長度來限制模組大小，以達到模組化目的。但設計師仍然可以投機取巧，**先撰寫好程式，再依一定長度來切**，不過以這種方式所產生的模組大概都不會是很好的模組。如何衡量某個模組是不是太大，該減肥了呢？應該可由**衡量模組緊密性**的觀點來看。太長的模組不僅不易了解及偵錯，或許還能再被細分成更小的模組。

32

DFD之模組

- 由模組本身“內部”的內容來看，分析其內容的『緊密性』，這就像一個工廠的生產線，其工作內容是否相關，可能某些工人的工作是在鍛造鋼板，有些是在焊結鋼板，這些都算是蠻密切的關係，但有些工人作的事是織布，這之間的關係就小了。
- 由模組與模組間的『關聯性』，這就像一棟大廈中的住戶間的關係，有的是親戚，有的是朋友或同事，但也可能毫無關係。

33

DFD之模組緊密性

- 緊密性或內聚力(愈大愈好)，為量度一個模組內部相互功能的強度(coherence)
 - 偶然緊密性：一個單一模組或許做了幾件工作，但這幾件工作都沒什關聯，充其量只是一種隨機的組合。這種情況像是一家百貨公司，東西賣得很多，不過彼此實在沒什關聯。

34

DFD之模組緊密性

- 邏輯緊密性：一個模組裡的工作是有關係的。像是“美食街”這類餐館一樣，賣的都是食物，是要填飽肚子的。
- 時間緊密性：如果一個模組裡的工作必需在同一段時間內執行，便具有此特性。例如開門鎖這件工作一樣，在轉動鑰匙的當時，還要用手轉動門把才行。
- 程序緊密性：如一模組內的單一工作有一定的執行順序，便稱為具備了程序緊密性。必需先打開電源，才能用電腦一樣。

35

DFD之模組緊密性

- 溝通緊密性：如一模組內的工作集中的存取某一些特定的相同資料，例如班上的同學為了交老師的報告，必需參考一些資料，但相信不少人，甚至大部份人都會找同一本書來參考。
- 順序緊密性：如果一個模組內的“很多”工作有先後的順序，和程序緊密性的“單一”不同。例如先得爬上二樓，才能上三樓，然後再上五樓。
- 功能緊密性：假如一個模組的工作都是相同的話。例如某個模組的功能就僅僅是作兩整數A與B的相加而已。

36

DFD之模組關聯性

- 關聯性或聯結力(愈小愈好)，為測量系統中各模組之間相關的程度 (Coupling)
 - 缺乏關聯性：這由字面上就知道，模組和模組間毫無瓜葛。
 - 記號關聯性：模組與模組間傳遞資料的方式不是傳遞一個個單一的參數列，而是整個資料結構（資料的聚合體）。就像寫信給某人，寄的不是一張張的信紙，而是一本書加上幾盒磁片，或許還有油畫等。

37

DFD之模組關聯性

- 控制關聯性：模組間所傳遞的資料若不是”單純的資料“，而是一些控制的旗號。這就像美國總統下給駐波斯灣官兵的命令不是一本厚厚的攻擊計劃，而只是攻擊的代碼而已。
- 資料關聯性：模組之間傳遞的資料是一個個參數。

38

DFD之模組關聯性

- 外界關聯性：模組與模組共同運用一個特定的 I/O 裝置。
- 共同關聯性：模組與模組之間共用同一部份的資料。就像薪資模組和出席模組共用人事資料一樣。
- 內容關聯性：若一個模組會用到另一個模組的控制資料，或者可以從另一模組中間開始執行。這種情況必需盡力避免。

39

模組最好為何

- 模組內的緊密度越高越好，而模組間的關聯度越低越好
- 透過模組化的設計方式，可以一步步的建立通用性的程式庫(libraries)，也可以引用過去所建立好的程式庫。

40