

《系統分析》

命題意旨	今年考題重點著重在實務應用上如何透過學習的理論與技術上的實踐，這是近年來國家考試在系統分析上命題的特色。
答題關鍵	<p>第一題：關鍵在考學生對於 UML 實作的能力，以及互動圖之間的觀念特性，熟悉 UML 者答題取分不致太難。</p> <p>第二題：實際上也只是透過物件導向的精神來測驗考生對於物件實作案例(Instance)時的元件觀念，若能了解元件圖的概念及製作技巧，應可得分。</p> <p>第三題：涵蓋範圍較廣，首先必須了解企業系統生命週期的觀念，將系統生命週期觀念展開，再加入系統開發生命週期的觀念，最後再輔以專案管理工具的甘特圖畫法方能得分；考生如果不小心將系統生命週期和系統開發生命週期混淆，得分將大大減少。</p> <p>第四題：透過案例考流程造模的程序、逐步分解細化的結構化概念以及處理規格描述實作的結構化英文與決策表技術，如果有實作練習的考生應能獲取較高分數。</p>
高分閱讀	<p>1.上課講義第五回 PP. 40-41、上課補充 UML 實例、吳仁和《物件導向系統分析與設計》。</p> <p>2.上課講義第五回 P. 43、上課補充 UML 實例、吳仁和《物件導向系統分析與設計》。</p> <p>3.上課講義第二回 PP. 25-26, PP. 45-46、上課補充甘特圖製作實例。</p> <p>4.上課講義第四回 PP. 32-33, PP. 76-78、上課補充流程造模製作實例。</p>

一、請參考下列有關合作圖與循序圖的說明，以圖一新增訂購項目使用個案合作圖繪製出相對應的循序圖。並請說明兩圖間的一致性如何檢查？請以該實例說明之。(30 分)

合作圖與循序圖之說明：

(一)UML 之合作圖與循序圖均表達一個使用個案內，許多物件間之互動行為，兩者均由相同的資料中推演出，因此兩種圖在語意上是相同的。亦即可以從循序圖轉換成合作圖，亦可從合作圖轉換成循序圖，轉換中不應有資訊遺失。

(二)由以下之步驟將循序圖轉成合作圖：

1. 合作圖與循序圖之物件相同。
2. 合作圖之連結是來自循序圖之物件間訊息的傳送／接收。
3. 合作圖中之訊息與操作與循序圖相同：
 - ①循序圖是依時間順序來表達發生順序。
 - ②合作圖是以數字（自然數或杜威數）來表達發生順序。

新增訂購項目使用個案合作圖說明：

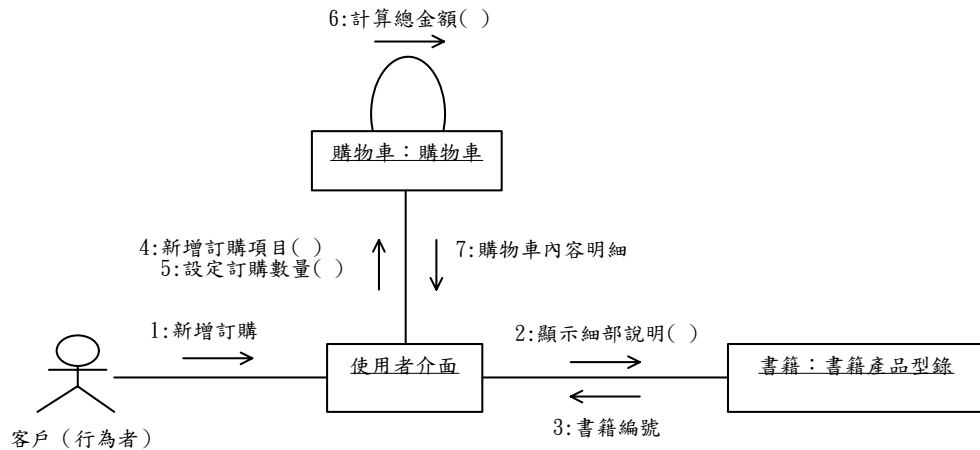
(一)物件有客戶、書籍產品型錄與購物車。

(二)循序圖之物件間有訊息傳送／接收，則合作圖之物件間必有一連結。

(三)訊息傳遞與操作依時間發生之順序分別有：

1. 新增訂購
2. 顯示細部說明
3. 書籍編號
4. 新增訂購項目
5. 設定訂購數量
6. 計算總金額
7. 購物車內容明細

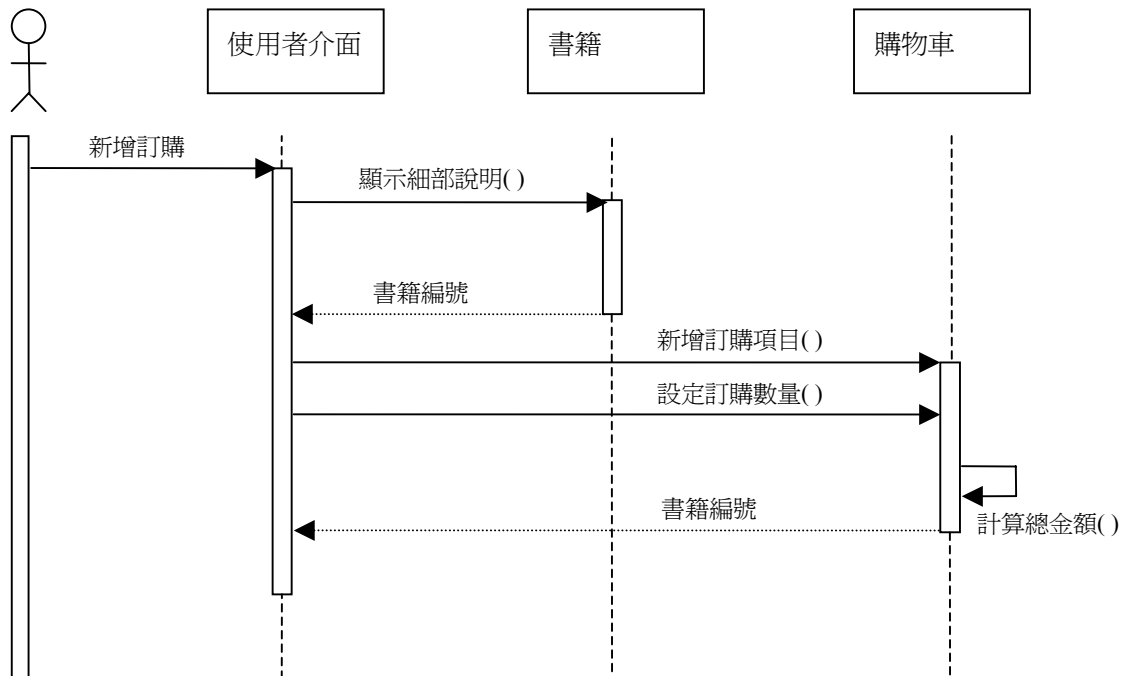




圖一 新增訂購項目使用個案合作圖

【擬答】

合作圖和循序圖在物件導向發展過程中均扮演著互動行為的說明，因此兩者有著一定程度的對應。而二者之關係如題目所示。根據題目之說明，可將循序圖轉換為合作圖，自然也可以在反向的運作中將合作圖轉而產生循序圖，依照循序圖產生的原則所建置的循序圖如下：



合作圖和循序圖既然是可以轉換的，二者便可以有一定的一致性，也因此在轉換中必須遵循下列原則來保持一致性：

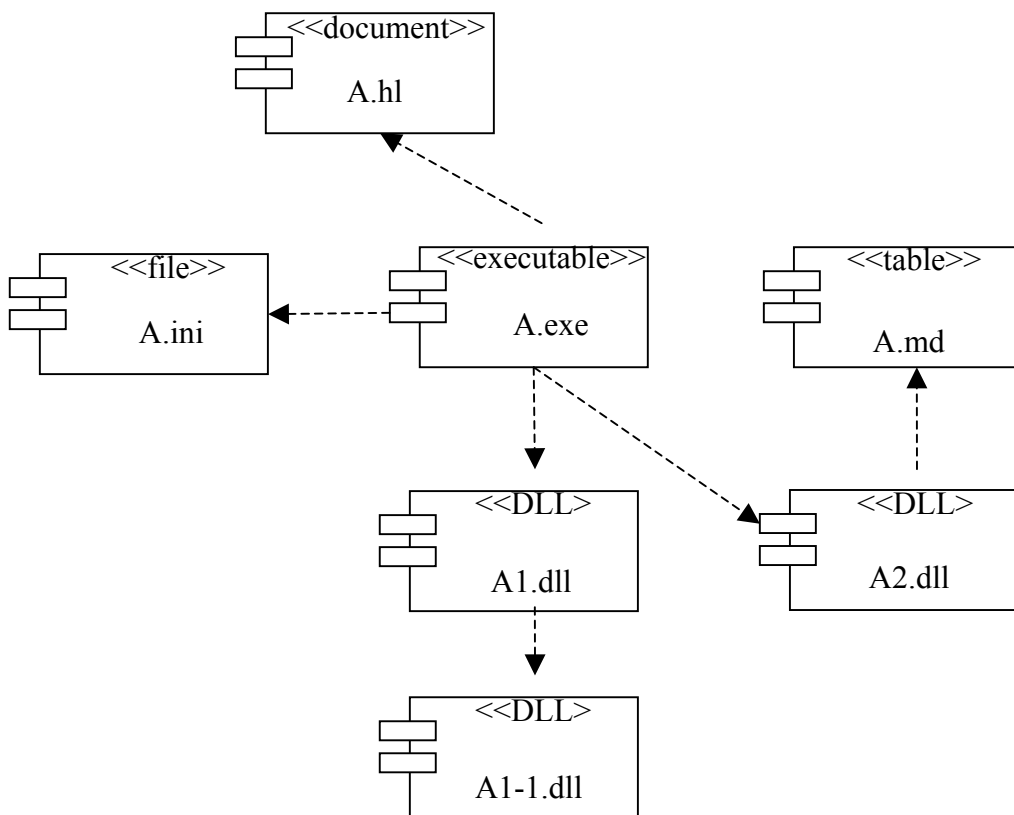
1. 合作圖中的物件與循序圖中的物件應該是相同。在題目中合作圖包含客戶、使用者介面、書籍和購物車等物件，所以在循序圖中也應該有這些對應物件。
2. 合作圖的連結是來自於循序圖中二個物件間有訊息傳遞或接收者。也就是在循序圖中的二物件間有訊息傳遞或接收，那麼在合作圖中的二物件間必定有連結。由此原則可檢證客戶和使用者介面、使用者介面和書籍、使用者介面和購物車間在循序圖中也要存在訊息傳遞或接收。
3. 合作圖中的訊息和操作應和循序圖中的順序一一對應，由此原則再配合循序圖裡的生命線起始及訊息傳遞的前後順序應該和合作圖裡的操作順序一致。



二、物件導向是將物件 (Object) 描述真實系統的觀念融入於系統開發中，且因為了使軟體成為獨立且容易拼裝的元件，物件的觀念因而形成。物件是基本的思維單位，是一個具有狀態 (State)、行為 (Behavior) 與識別 (Identity) 的實體 (Entity) 或抽象化概念 (Abstract Concept)，且其行為會影響其狀態者。因此請說明在物件導向系統開發中所謂的物件應包括那些元件？請舉一個物件的實例 (Instance) 說明之。(20 分)

【擬答】

1. 在物件導向系統開發中，元件是系統的實體模組，也是系統中可被替換的部份，被用來提供給一些介面做適當的實現。元件與類別雖然有點相似，但二者依然有點差異。類別用來表達邏輯的抽象化事物，元件則表達真實世界的實體事物。因應這樣的關係，物件導向系統開發中包含三類元件，分別為部署元件、工作產品元件和執行元件。
2. 部署元件一般是可執行的系統所需的元件，在物件實例中像是動態連結函式庫 (dll) 或執行檔 (exe) 均為此類；而工作產品元件則是屬於原始碼檔案或資料檔案，利用工作產品元件可以生成部署元件，另外執行元件則是一種系統執行的順序。
3. 在 UML 中對這些元件的對應形式分別建立執行檔、函式庫、資料表、檔案與文件圖形，茲利用物件實例說明不同元件的關係如下：



4. 在物件的實例 (Instance) 中 A.exe 是一個執行檔元件其執行依賴 A.ini 的檔案元件進行參數設定、依賴 A.hlp 的文件元件進行說明內容的提供。另外，其執行時期仰賴 A1.dll 與 A2.dll 的函式庫元件提供動態執行的函式功能，A1.dll 會在執行時呼叫 A1-1.dll，而 A2.dll 則與 A.mdb 的資料表元件存在關係。

三、大部分的資訊系統專案皆以團隊開發方式完成，通常會由團隊的某一成員擔任專案經理來控管專案的進行。請以開發檢察事務官所需的「偵訊筆錄管理專案」為例，說明完成專案開發所需的「作業事件」有那些？請依系統生命週期的循環過程的先後順序一一條列寫出來，並請繪出相對應的甘特圖 (Gantt Chart)。(30 分)



【擬答】

資訊系統生命週期依照前後順序為產生商業需要、系統發展、安裝、系統操作與使用到系統荒蕪，在系統荒蕪之後因為不再能達成企業的營運需求，因此會循環再發生產生商業需要、系統發展……生命週期的過程，重新再發生資訊系統的開發。然而新的資訊系統生命週期為了反應市場快速變遷，因此不會等到系統荒蕪的階段來重新啟動產生商業需要的步驟，而會直接在系統操作與使用階段便銜接產生商業需要的階段。因此，在建立「偵訊筆錄管理專案」中的系統生命週期循環過程包括有：產生商業需要、系統發展、安裝、系統操作與使用四個循環階段。茲將各階段的作業事件列式如下：

1.產生商業需要

確立系統發展目的；界定系統的功能目標與範圍。

2.系統發展

分析:探討系統所需的一切，如可行性分析、需求分析、系統分析等；

設計:探討系統如何被完成，如概念性設計、細部設計等；

製作:主要為編碼和測試兩件工作。

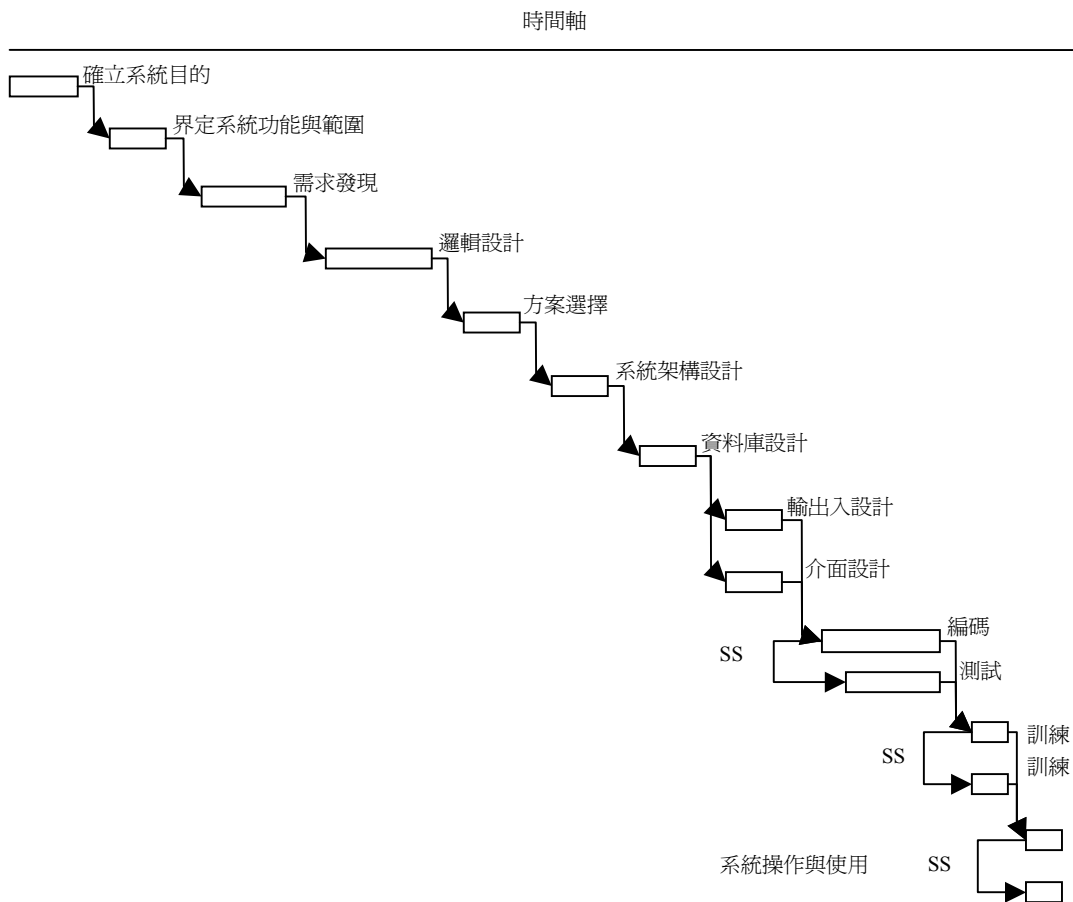
3.安裝

安裝；訓練。

4.系統操作與使用

實際在企業中運作；蒐集系統問題。

由資訊系統生命週期的工作項目，再輔以專案管理的運作，則可以產生甘特圖如下：



四、請以系統分析與設計樣板中有關環境圖與資料流程圖為例，說明如何完成專案開發的「分解圖」？請舉例繪出相對應的圖示，並說明那些部分的流程須以「處理規格描述」來說明？並敘述常用來作為處理規格描述的結構化語文的語法與圖表有那些？（20 分）

【系統分析與設計樣板】

壹、環境分析

使用個案

環境圖

貳、資料分析

實體關係圖

實體關係圖轉關聯表

正規化

關聯表資料字典

參、流程分析

資料流程圖（第 0 階至第 n 階）

分解圖（也稱為階層圖）

處理規格描述（結構化語文）

肆、模組設計

結構圖或 HIPO 圖

程式設計語言（PDL）模組描述

伍、使用者介面設計

介面結構圖

介面藍圖與元件規格

介面狀態圖與轉換表

陸、軟硬體環境設計及開發工具選擇

硬體網路架構

作業系統

應用系統架構

開發工具

【擬答】

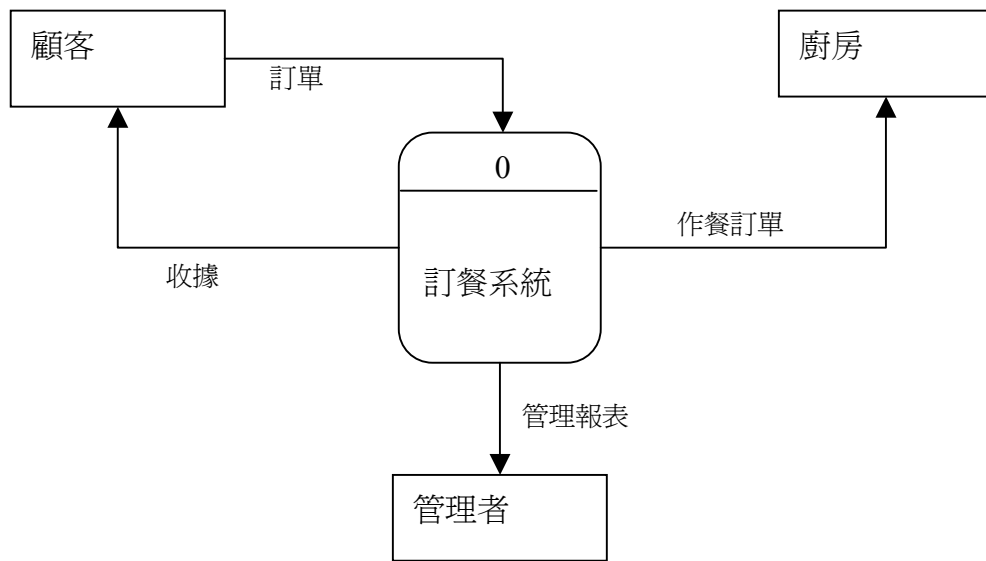
以流程造模的方式來進行系統分析與設計，可以遵循逐步系統化分解的概念進行分析的工作，要完成這樣逐步分解、細化的分析作業，開發專案可以利用如下的策略來進行：

- (1)建構系統的環境關連資料流程圖(context data flow diagram)，以建立初步的專案範疇。
- (2)繪製功能分解圖(functional decomposition diagram)，將系統分成邏輯子系統或功能。
- (3)收集事件-回應(event-response)或使用案例列表(use case list)，以找出並確認系統必須對它們提供回應的業務事件。這個列表也描述每個事件需要的或可能的回應。
- (4)在分解圖中為每個事件加入事件處理器(event handler)。現在，分解圖可作為系統的概要。
- (5)選擇性地，建構並驗證每個事件的事件圖(event diagram)。這個簡單的資料流程圖只顯示事件處理器和每個事件的輸入與輸出。
- (6)藉合併事件圖來建構一個或多個系統圖(system diagram)。這些資料流程圖顯示系統的「大圖」。
- (7)建立基本圖(primitive diagram)，這是為那些需要更詳細的處理流程細節的處理而建立的。這些資料流程圖顯示單一事件之所有的基本處理、資料儲存和資料流。
- (8)每個基本處理的邏輯和每一基本資料流的資料結構以處理規格描述的工具來描述。

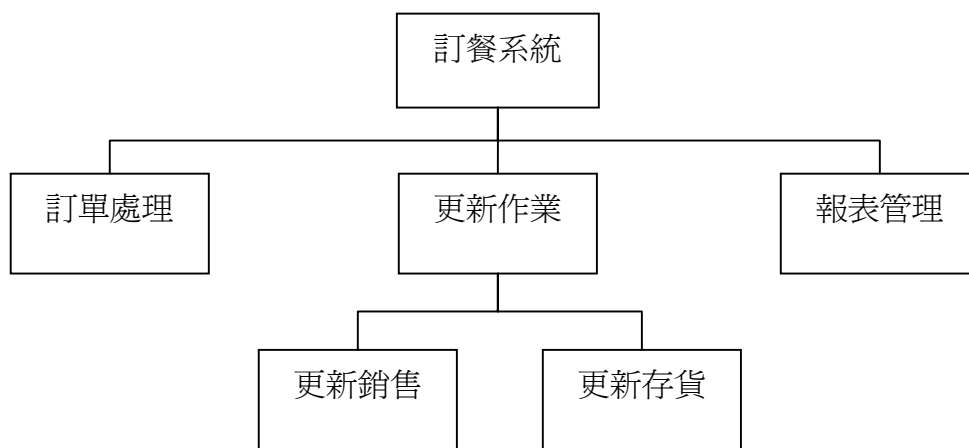


茲舉餐廳系統實例說明：

環境關聯圖：



功能分解圖：透過環境關聯圖將系統進行分解為子系統

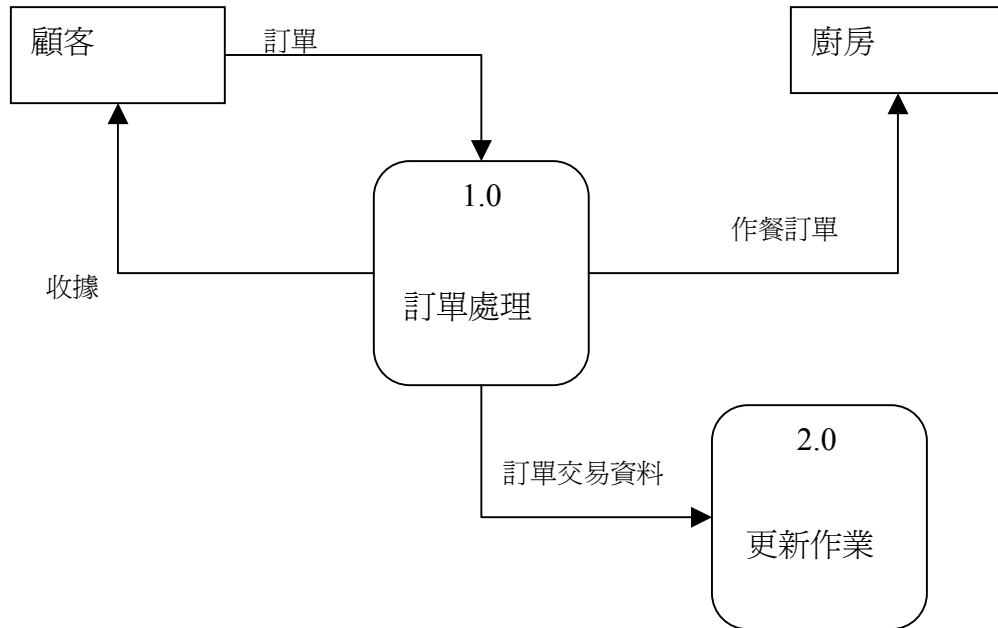


使用案例列表：舉客戶訂餐為例

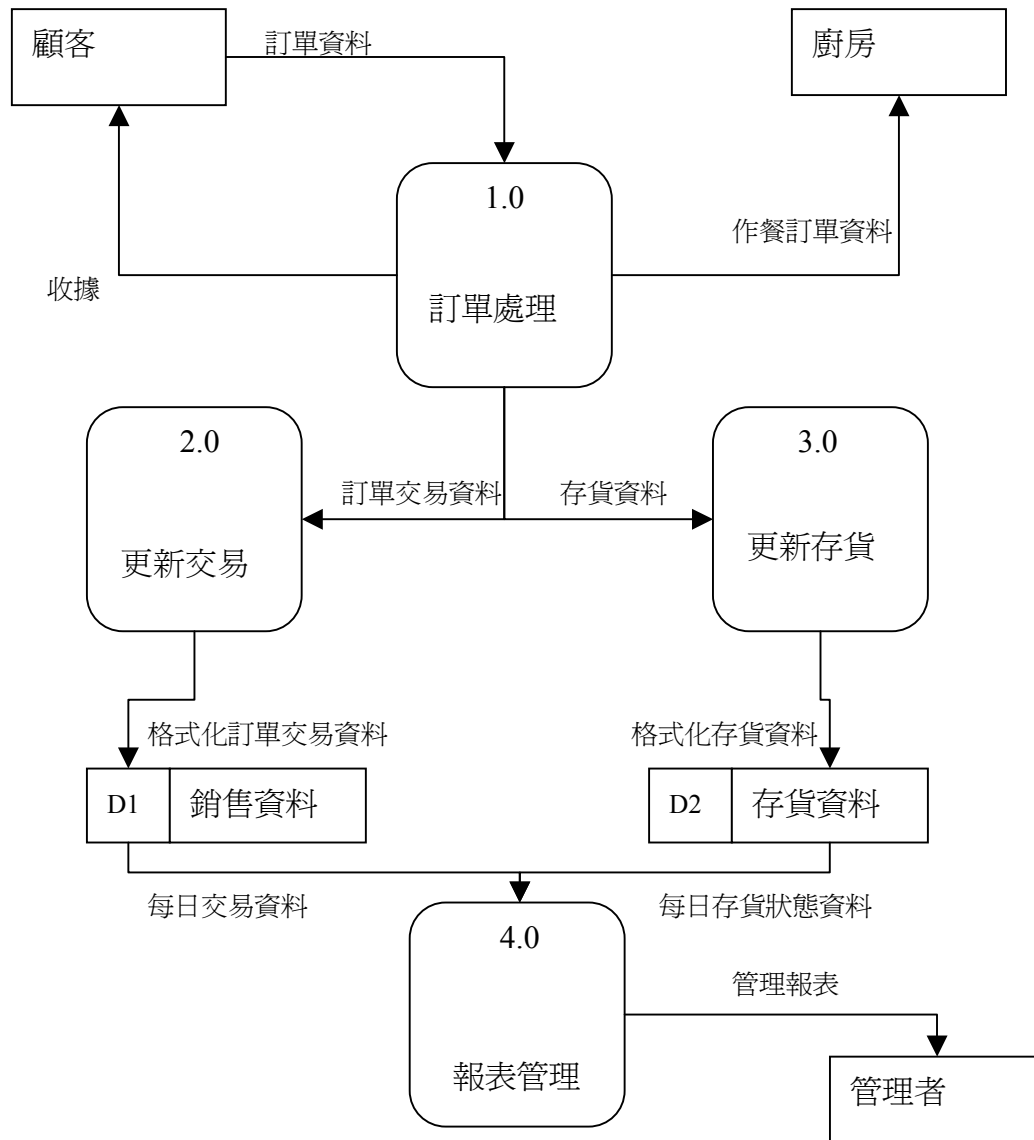
角色	事件	觸發因素	回應
客戶	訂單處理	下訂單	判別處理訂單後提出訂單收據



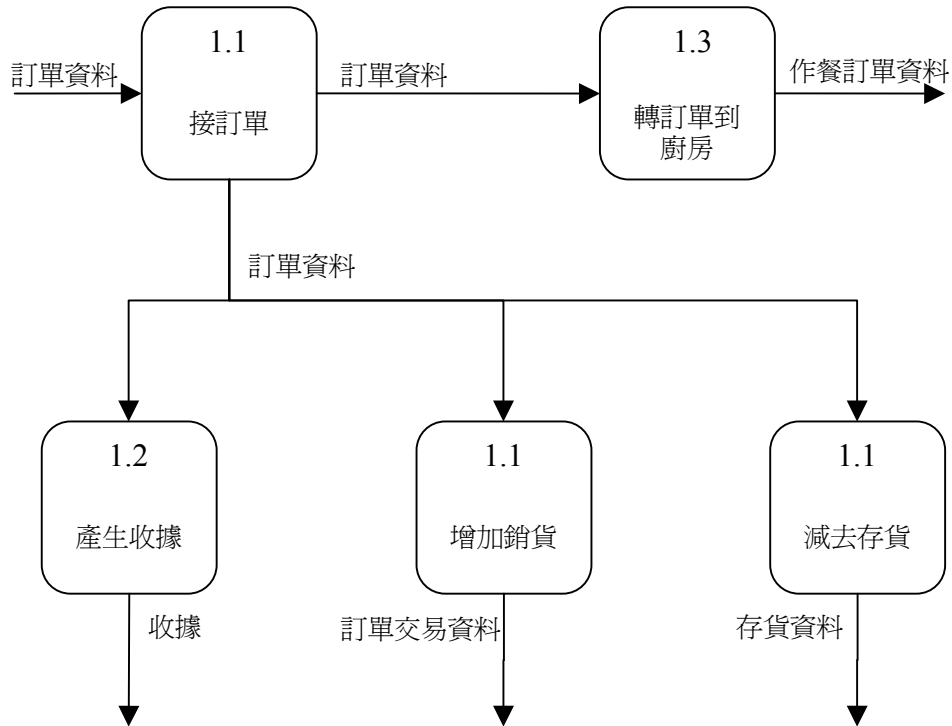
進行事件處理器的作業：透過事件圖



產生系統圖：



建立基本圖：將系統圖中之事件圖再予以細化分解



處理規格描述：

在利用基本圖予以分解細化直到最底層時，便可以利用處理規格描述來說明處理的實施方法，在結構化分析與設計的概念，當事件處理已經符合模組化精神時，便可以不必再行細化分割，此時便可以利用處理規格描述來協助開發工作的繼續進行。

處理規格描述可以使用結構化英文、決策樹或決策表來進行，通常是遭遇到必須進行邏輯條件的決策判斷時使用，結構化英文的語法有 IF.....THEN，DO.....UNTIL，READ，.....等等。

例如在增加銷貨處理：

Process 1.1：增加銷貨

READ 銷售項目資料 from 訂單資料

IF 會員 is N and 消費金額>3000 THEN subtotal = subtotal * 0.9

IF 會員 is Y and 消費金額>3000 THEN subtotal = subtotal * 0.85

IF 會員 is Y and 消費金額>5000 THEN subtotal = subtotal * 0.8

又如在接訂單時若有折扣，可以透過決策表進行判斷：

條件／行動	規則		
	1	2	3
會員	Y	Y	N
消費金額	>5000	>3000	>3000
9 折			x
85 折		x	
8 折	x		

