物件導向資訊系統分析與設計 -UML研討會

2005.04.27

參考資料

- IBM UML Resource Center
- http://www.omg.org/uml/
- Systems Analysis and Design with UML : An Object-Oriented Approach , Second Edition, Alan Dennis
- 系統分析與設計—理論與實務應用, 吳仁和
- 系統分析教材, 劉志俊教授
- TBLink 技轉資料

Agenda

- State Transition Diagram
- Component Diagram
- Deployment Diagram
- Interface Design
- Conclusion

State Transition Diagram

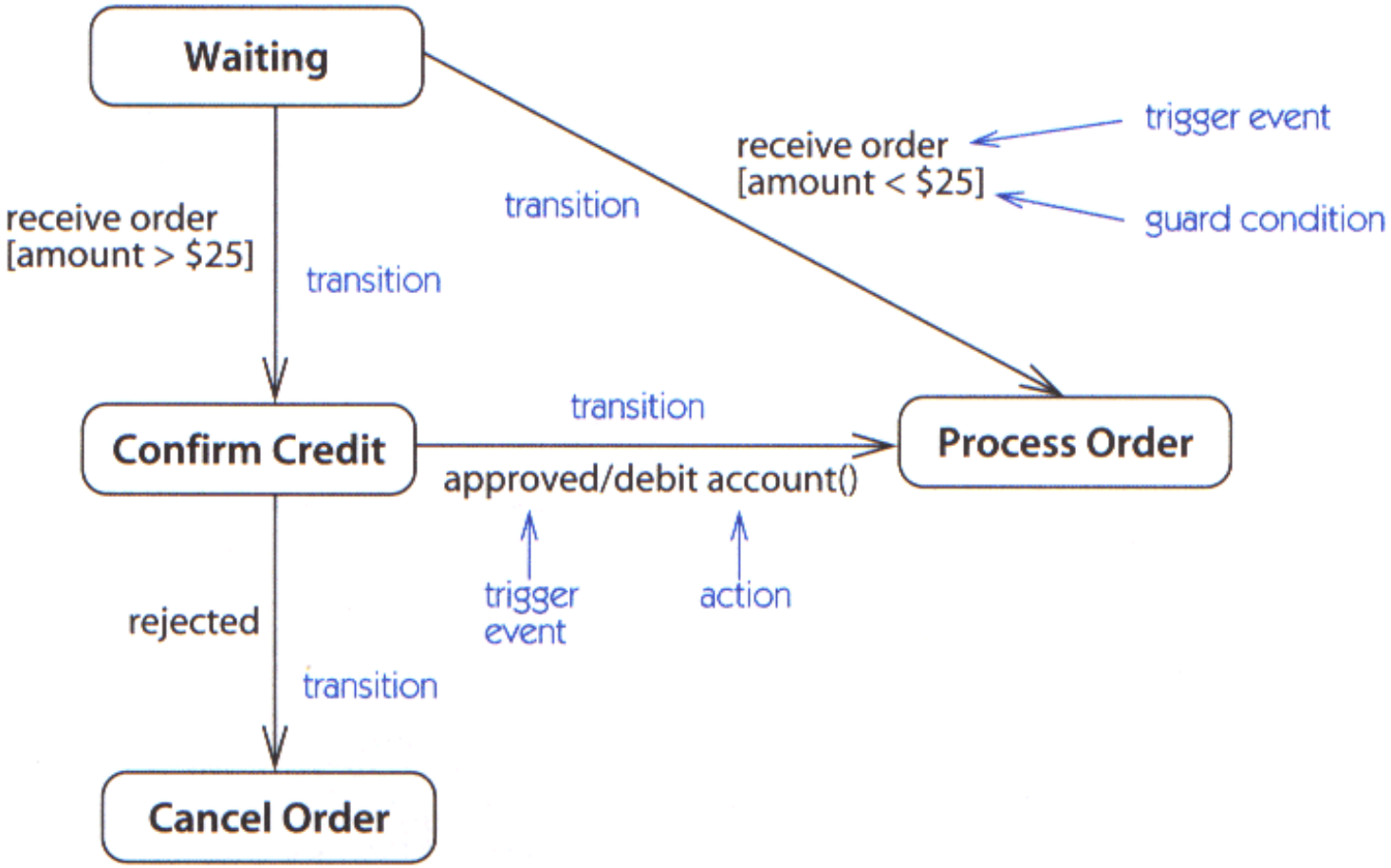
State Transition Diagrams

- ***State transition diagrams*** model the dynamic behavior of individual classes or any other kind of object.
  - the sequences of ***states*** that an object goes through
  - the events that cause a ***transition*** from one state to another
  - the ***actions*** that result from a state change

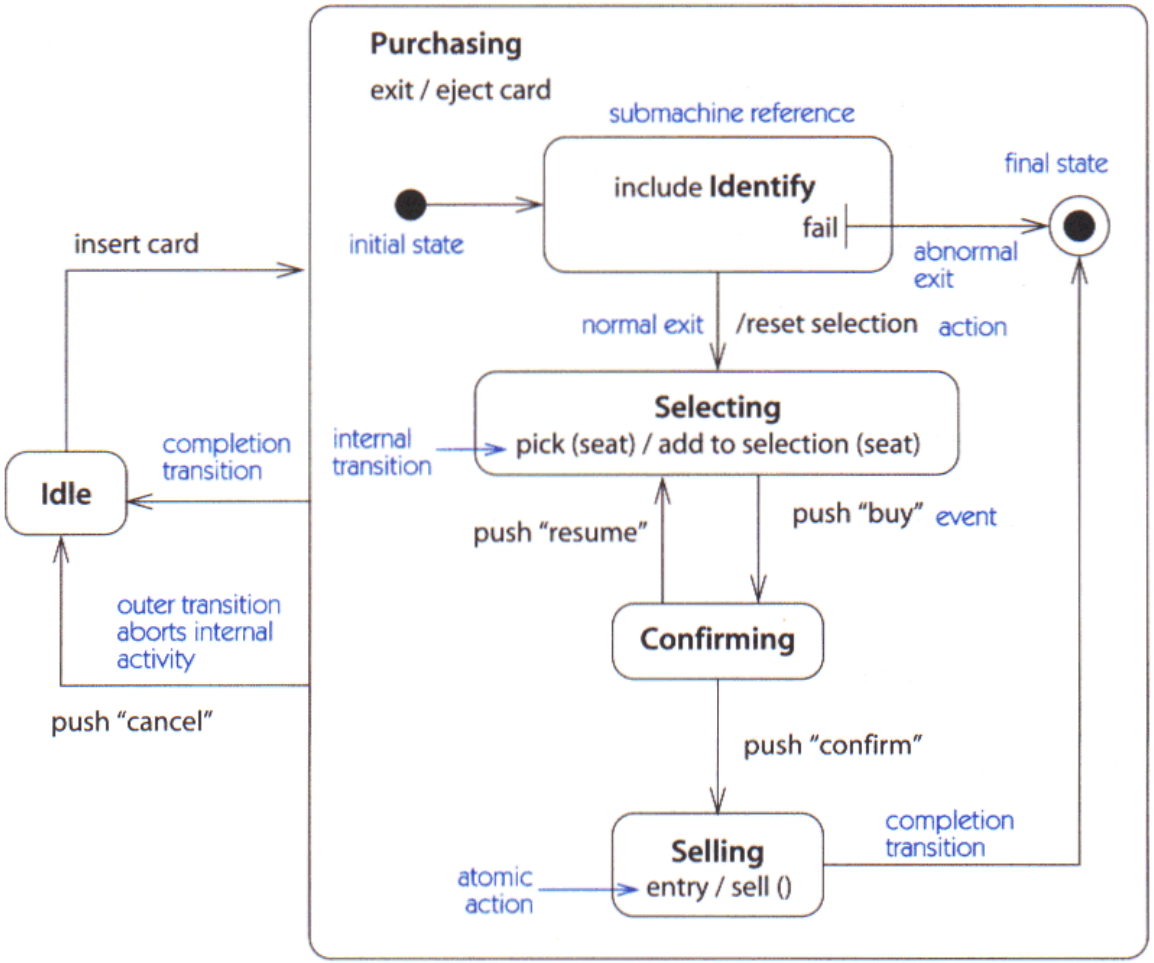
State Transition Diagrams vs. Activity Diagrams

- ***State transition diagrams*** are closely related to ***activity diagrams***.
- The main difference
  - State transition diagrams are state centric
  - Activity diagrams are activity centric
  - A state transition diagram is typically used to model the discrete stages of an object lifetime
  - An activity diagram is better suited to model the sequence of activities in a process

State Transition Diagrams
Notation: Summary

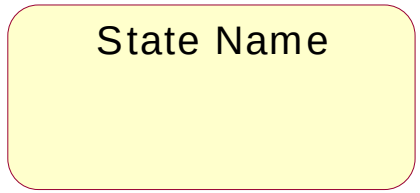


State Transition Diagrams
Notation: Summary



State Transition Diagrams
Notation: State

- A ***state*** represents a condition or situation during the life of an object during which it satisfies some condition or waits for some event.
- The state icon appears as a rectangle with rounded corners and a name.

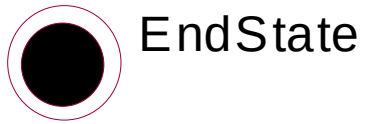


狀態與轉換樣板

State Transition Diagrams
Notation: State

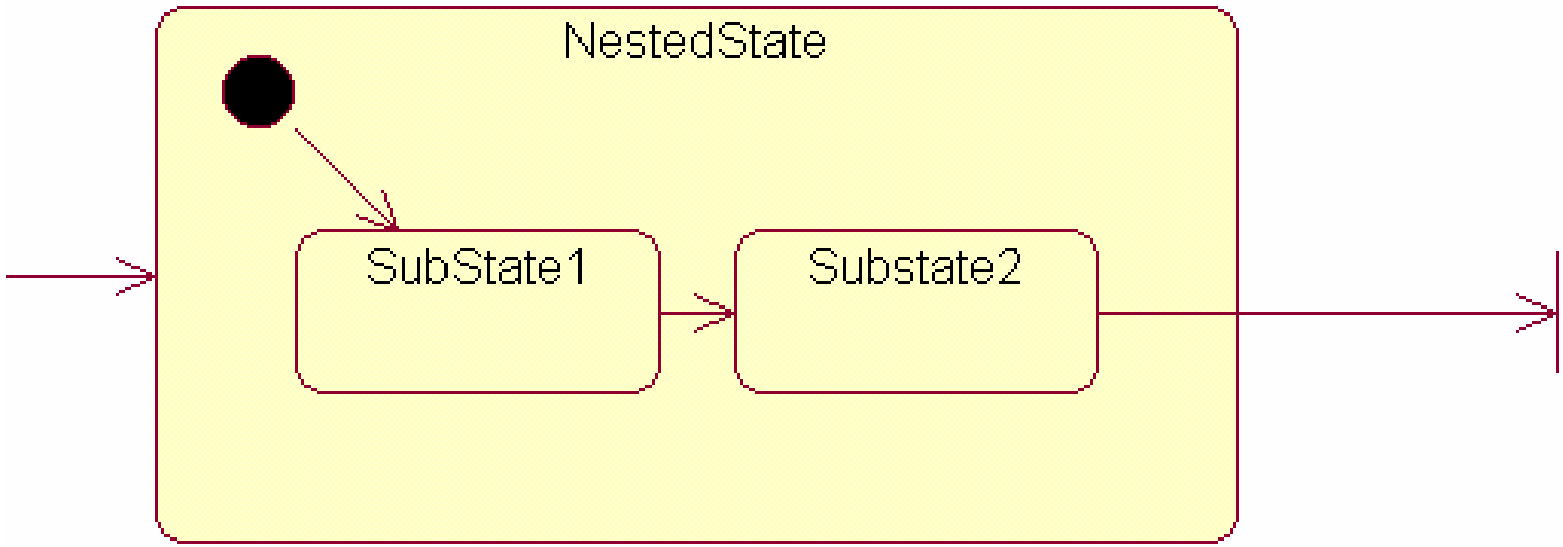
- A ***start state*** explicitly shows the beginning of the events that cause a transition on a state transition diagram.

- An ***end state*** represents a final or terminal state on an.



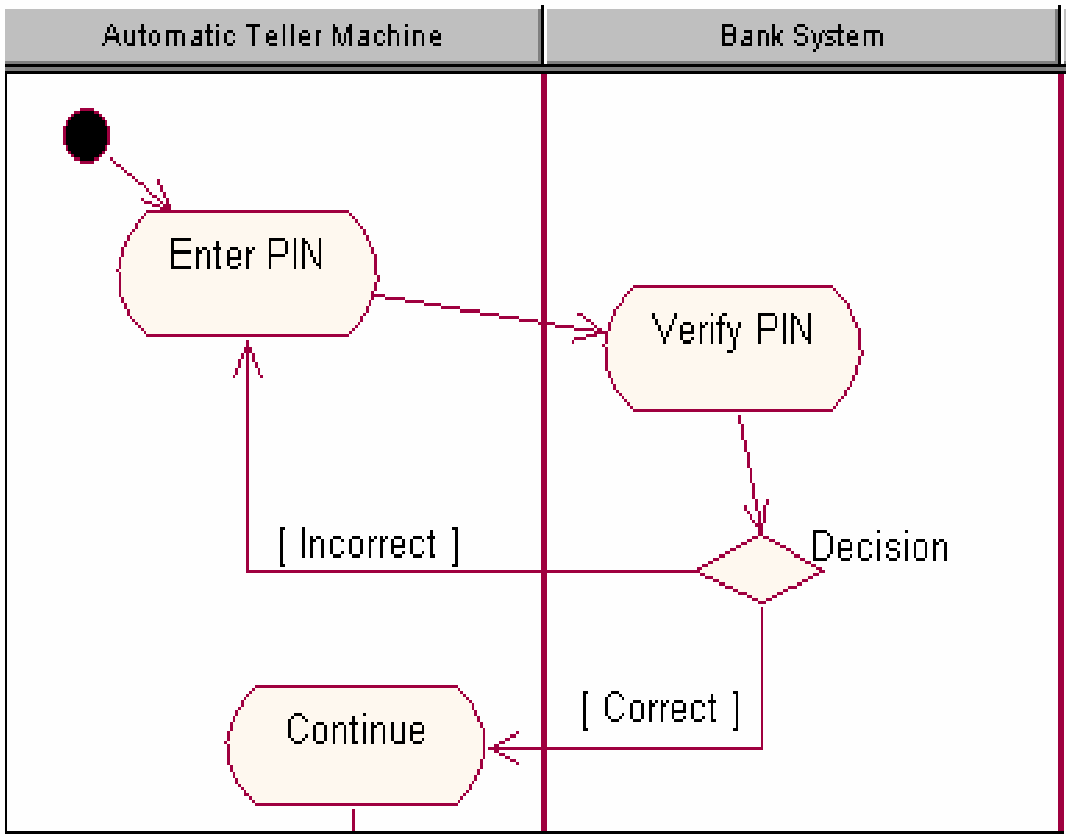
State Transition Diagrams
Notation: State

- States may be ***nested*** (called ***substates***) to any depth level.



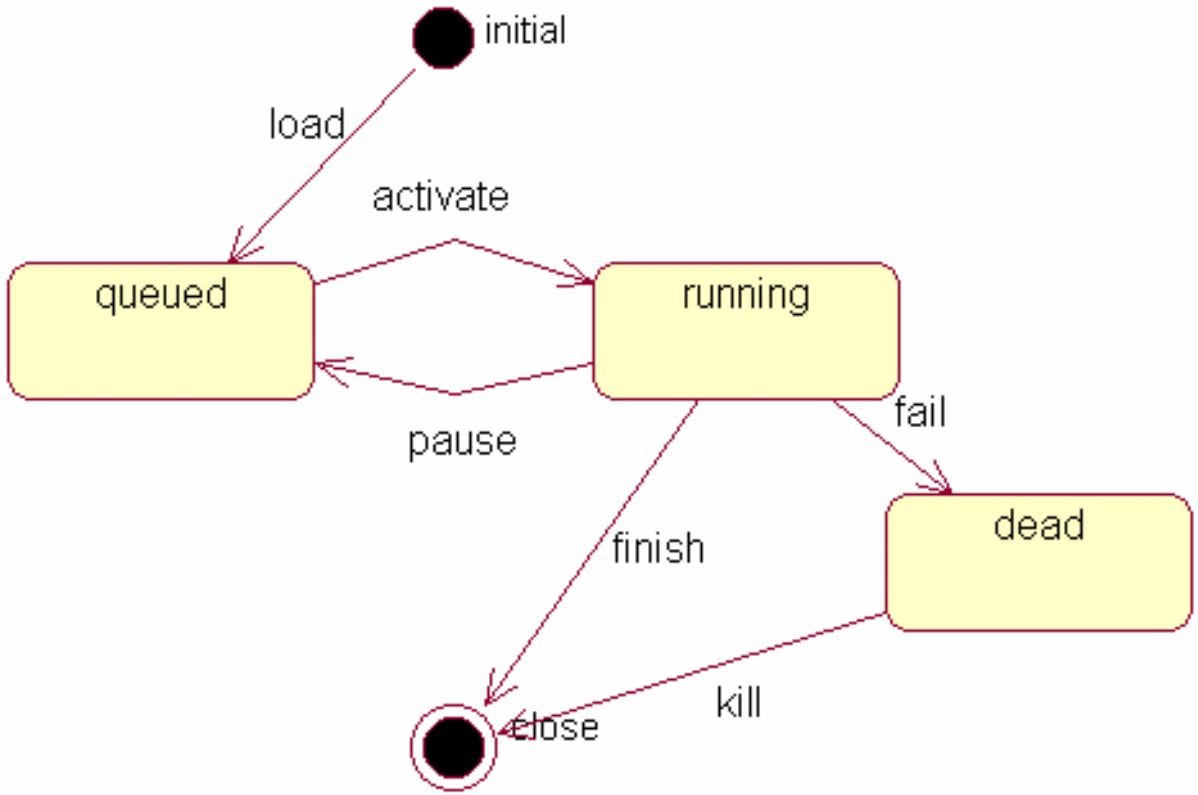
State Transition Diagrams
Notation: Decision

- Decision Example:
  - guard conditions: [correct] and [incorrect]

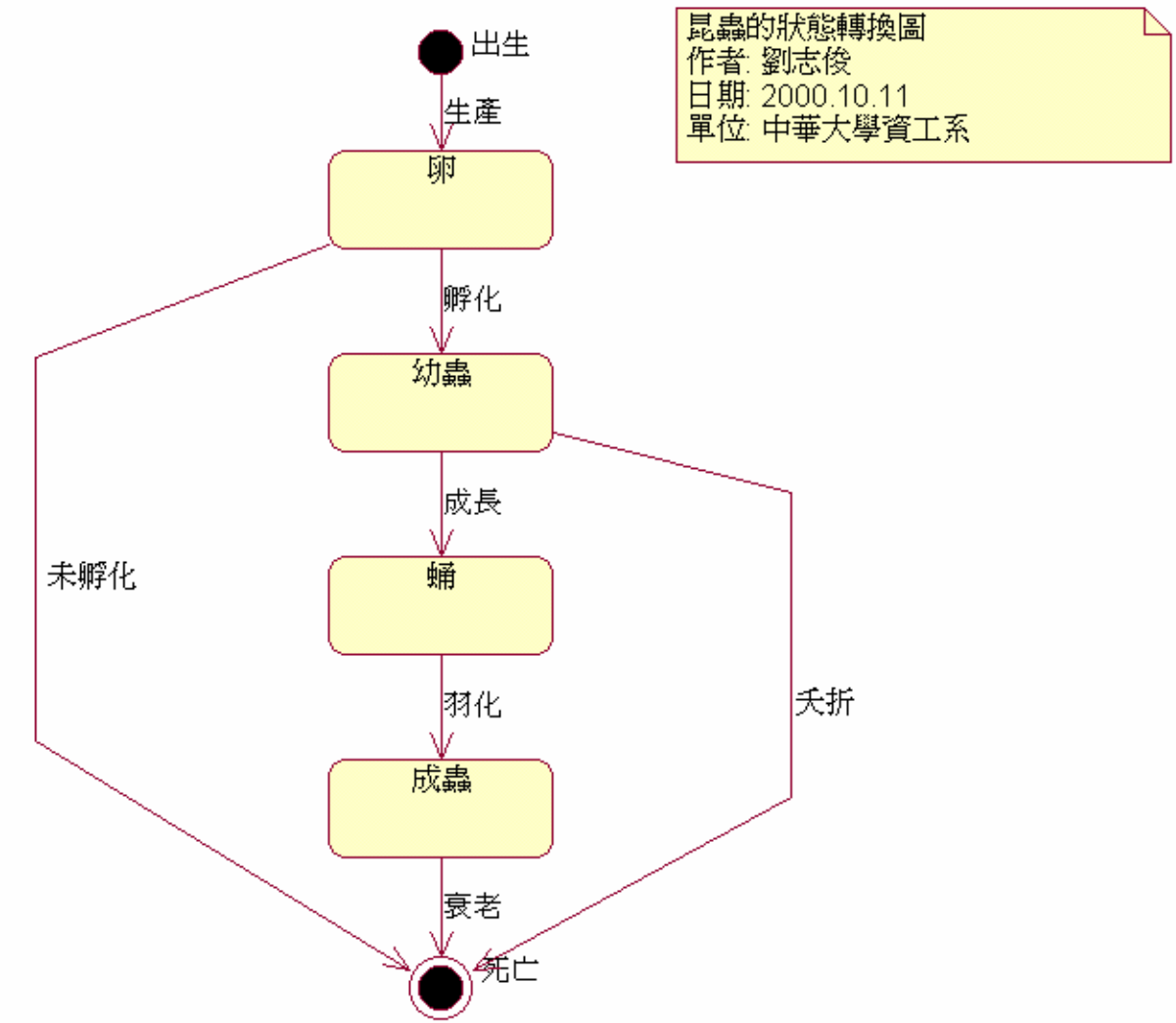


State Transition Diagram

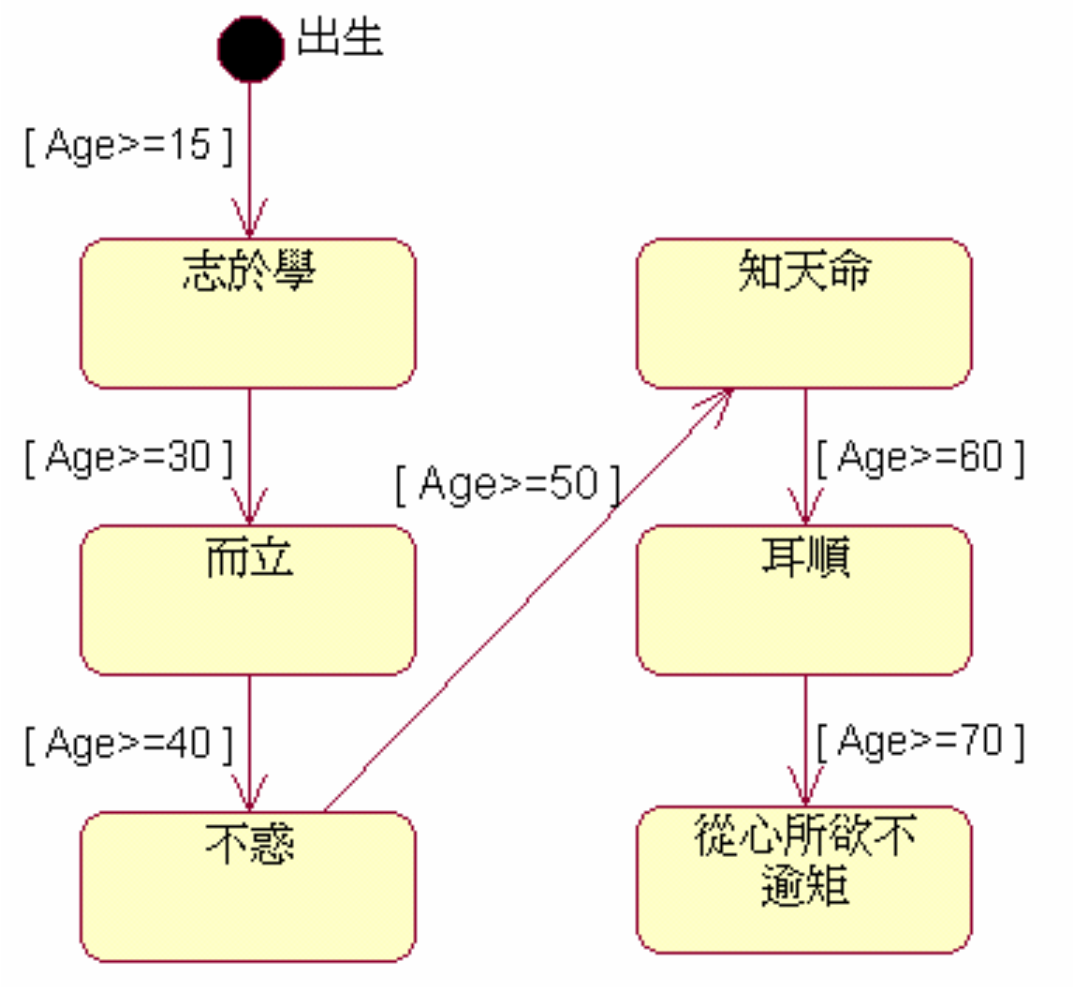
- Example 1: Process



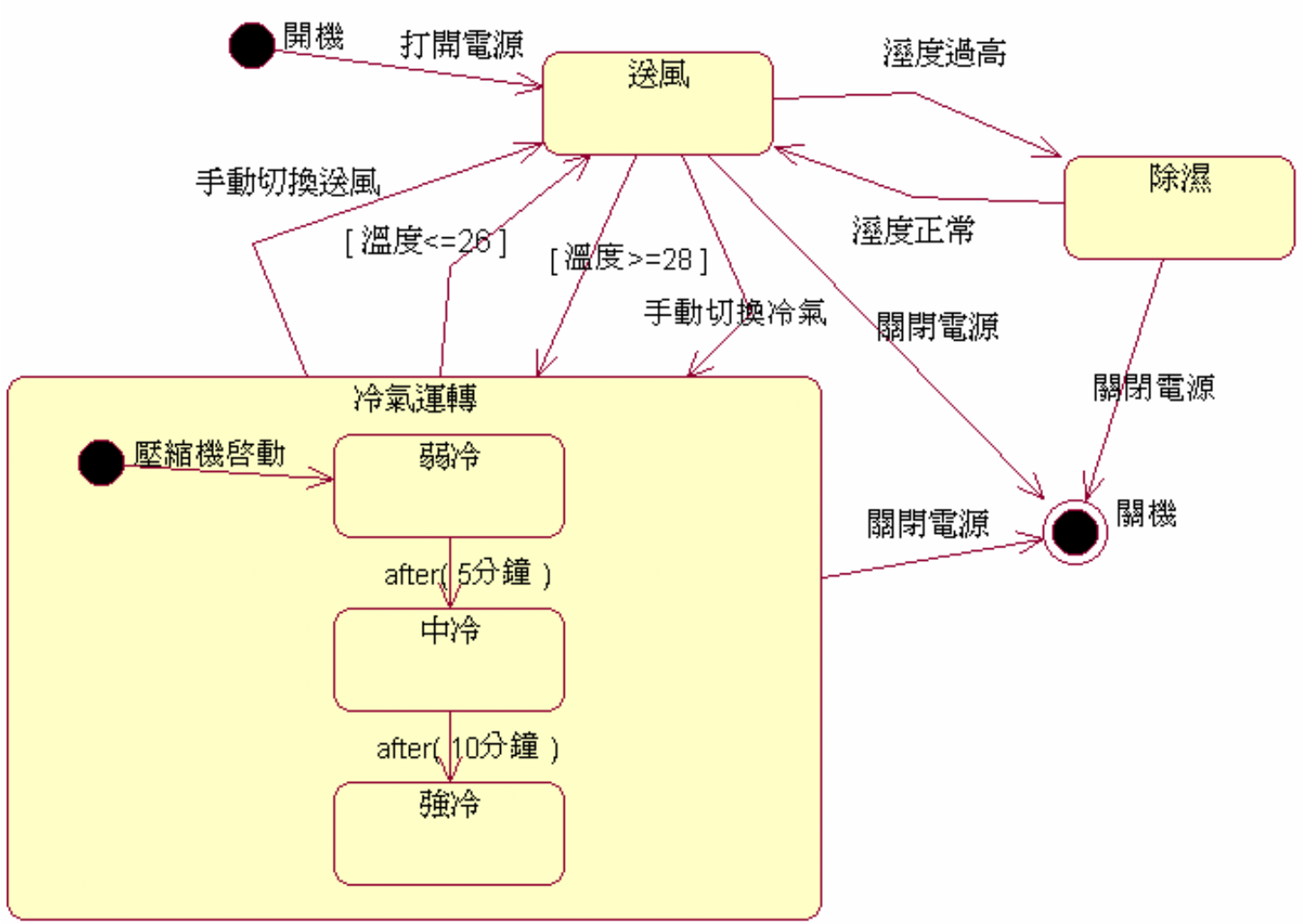
State Transition Diagram
Example 2: 昆蟲



State Transition Diagram
Example 3: 孔子



State Transition Diagram
Example 4: 冷氣



狀態圖之建構步驟與原則

- 狀態圖之建構步驟
  - 找出狀態
  - 找出狀態間之轉換
  - 繪製狀態圖
  - 精練狀態圖
- 建構狀態圖可參考下列原則：
  - 從循序圖中類別物件之操作描述、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。

狀態圖之建構步驟與原則（續）

- 由狀態圖之上方或左上方以「開始」畫起，從系統的觀點，依類別物件之行為，將物件生命週期的活動狀態順序，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。

狀態圖之建構案例

- 本訂購系統為例，依狀態圖之建構步驟針對購物車之類別物件進行其作業行為塑模，步驟如下：
  - 找出狀態
    - 本案例僅有購物車物件需要做狀態圖描述，因為該物件之行為包括許多的狀態與轉換。從購物車之操作描述可得知有五個狀態：新增訂購項目、等待、修改訂購數量、刪除訂購項目、取消採購訂單。

狀態圖之建構案例（續）

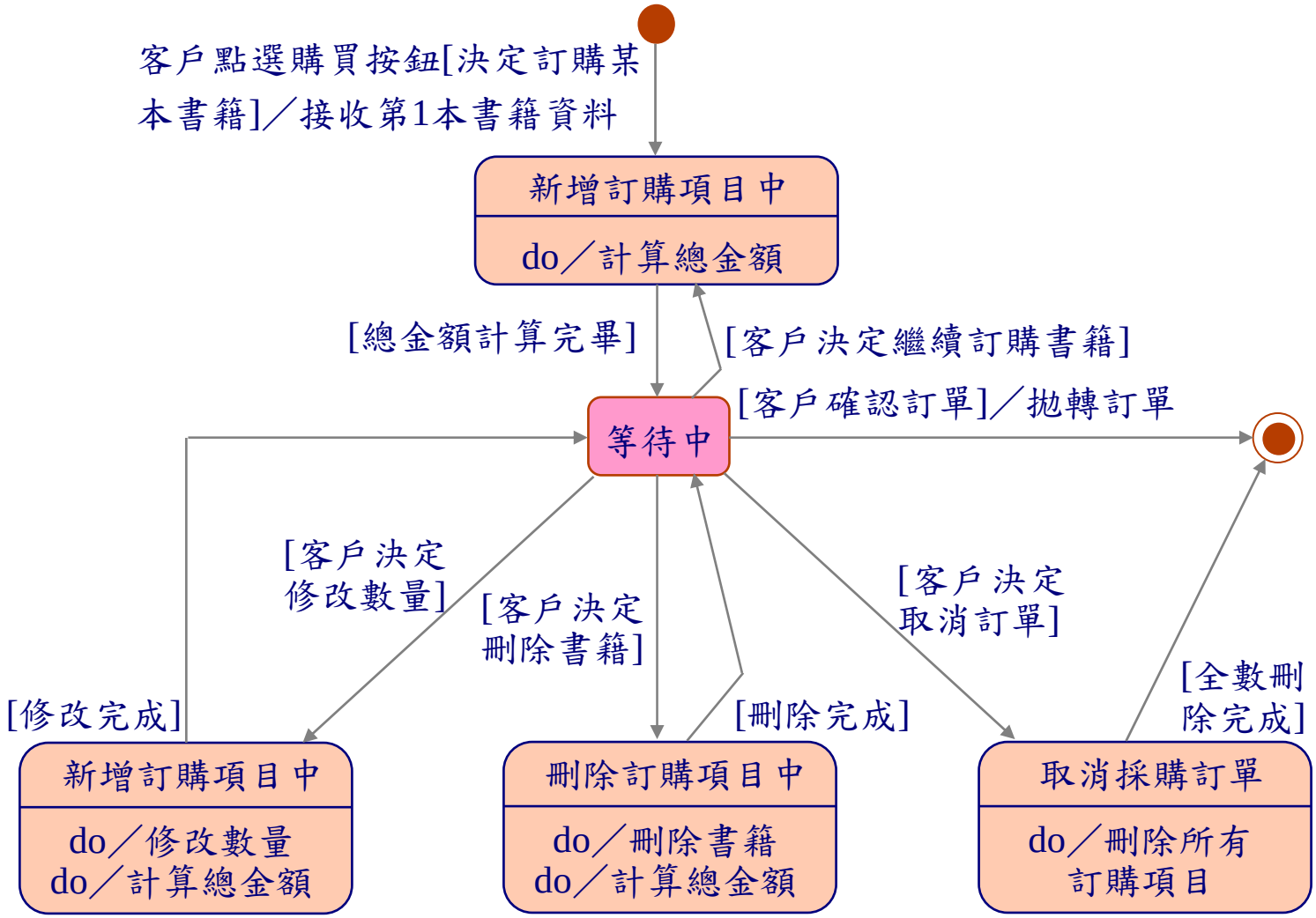
- 其中，新增訂購項目狀態之活動是計算總金額；等待狀態僅有狀態；修改訂購數量之活動是修改數量與計算總金額；刪除訂購項目之活動是刪除書籍與計算總金額；取消採購訂單之活動為刪除所有訂購項目。

- 找出狀態間之轉換：依購物車之操作描述，購物車之狀態、轉換（包括事件、成立條件與動作）與其間之關係。
- 繪製狀態圖。
- 精練狀態圖。

購物車之狀態與　轉換

項次	來源狀態	轉　　換	目的狀態
1	起始	客戶點選購買按鈕[決定購買某書]／接收第一本書籍資料	新增訂購項目
2	新增訂購項目	[總金計算完畢]	等待
3	等待	[客戶決定繼續訂購書籍]	新增訂購項目
		[客戶決定修改數量]	修改訂購數量
		[客戶決定刪除書籍]	刪除訂購項目
		[客戶決定取消訂單]	取消採購訂單
		[客戶確認訂單]	結束
4	修改訂購數量	[修改完成]	等待
5	刪除訂購項目	[刪除完成]	等待
6	取消採購訂單	[全數刪除完成]	結束

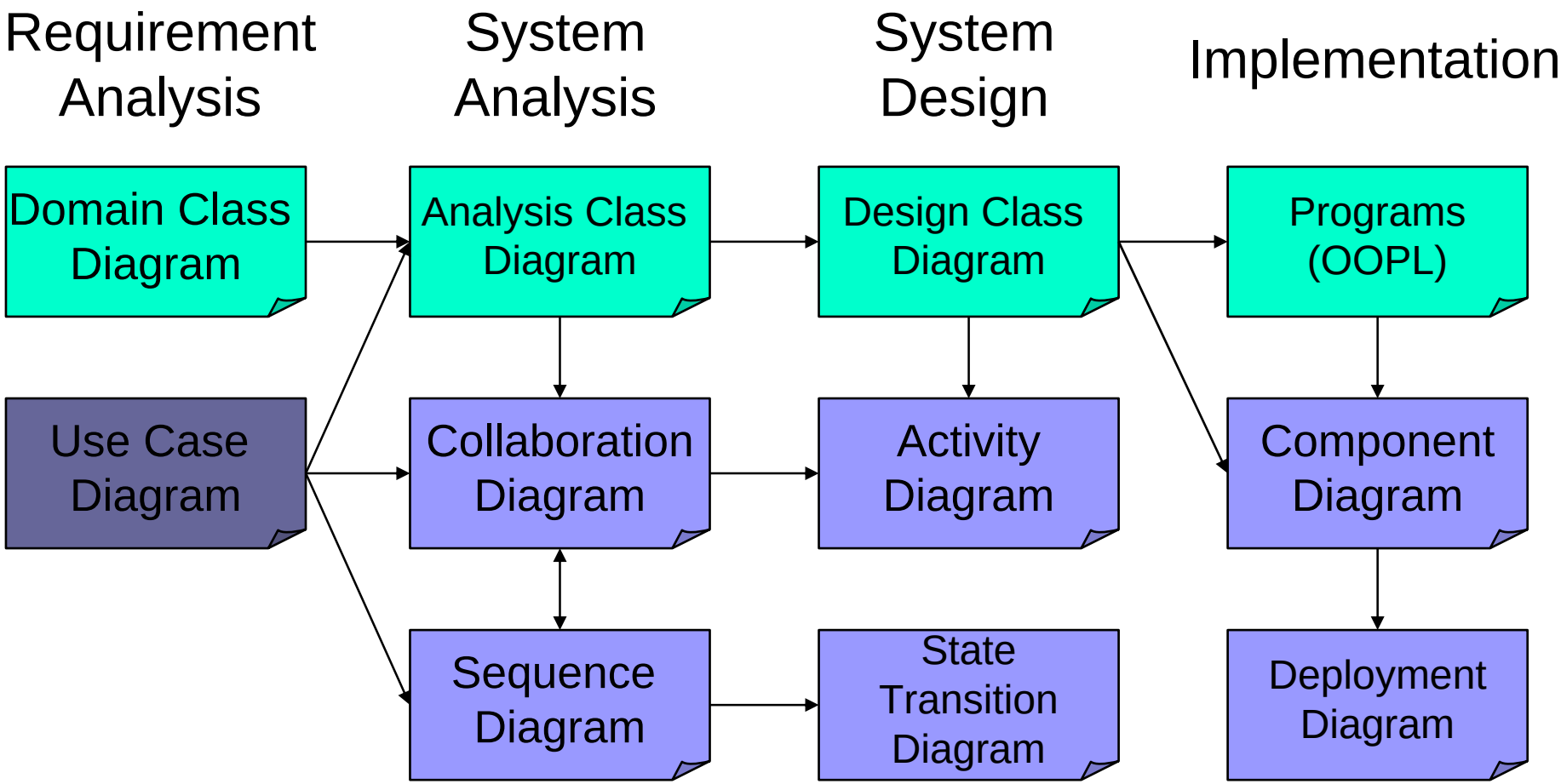
購物車之狀態圖


客戶點選購買按鈕[決定訂購某本書籍]／接收第1本書籍資料
新增訂購項目中
do／計算總金額
[總金額計算完畢]
[客戶決定繼續訂購書籍]
[客戶確認訂單]／拋轉訂單
等待中
[客戶決定修改數量]
[客戶決定刪除書籍]
[客戶決定取消訂單]
[修改完成]
[刪除完成]
[全數刪除完成]
新增訂購項目中
do／修改數量
do／計算總金額
刪除訂購項目中
do／刪除書籍
do／計算總金額
取消採購訂單
do／刪除所有訂購項目

Component Diagram

Implementation

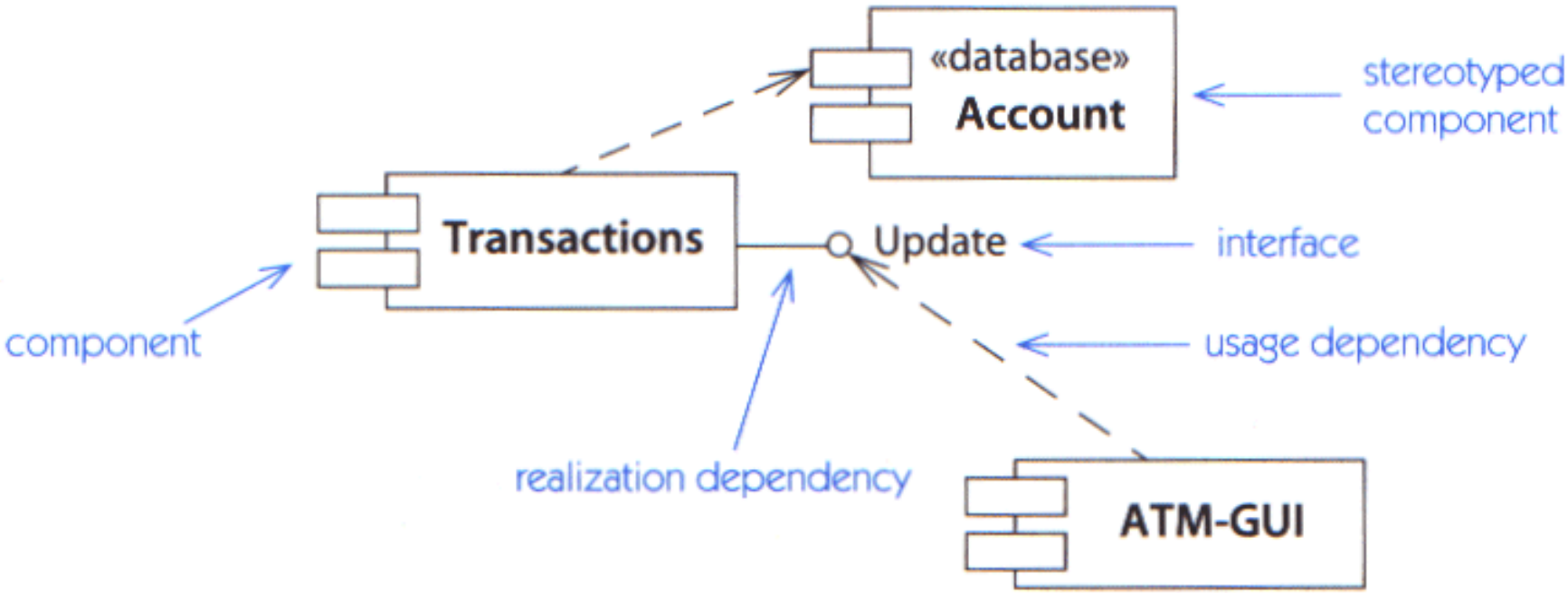
Artifacts for Implementation



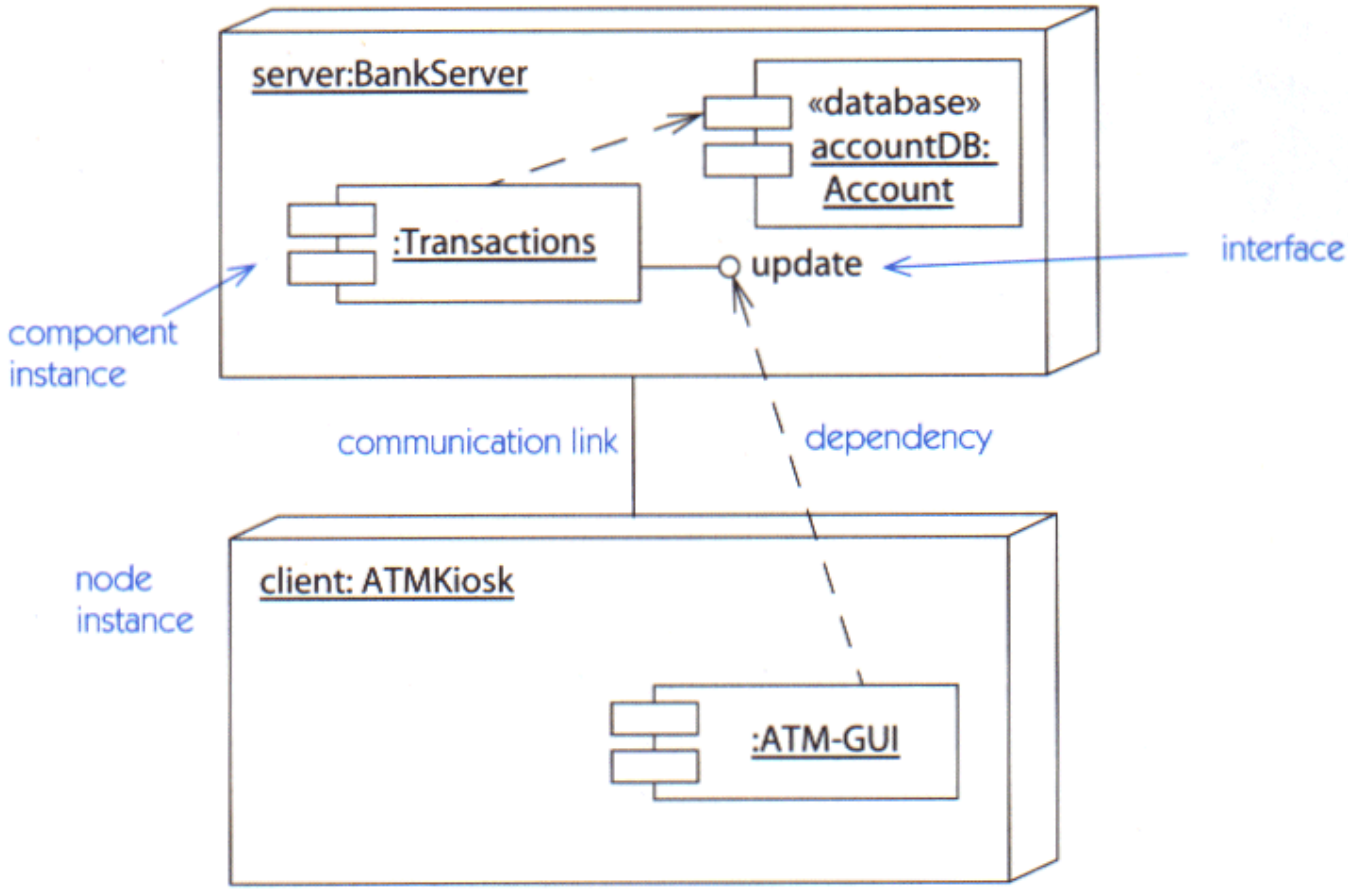
Artifacts for Implementation Programs

```
public class Account {
1     // In this example the Account has a balance only
2     private Money balance = new Money (0);
3     public Money withdraw(Money amount) {
4                 // First we must ensure that the balance is at least
5                 // as big as the amount to withdraw
6                 if balance >= amount
7                 // Then we check that we will not withdraw negative amount
8                       then { if amount >= 0
9                             then {
10                                  try {
11                                        balance = balance – amount;
12                                        return amount
13                                        }
14                                  catch (Exception exc) {
15                                  // Deal with failures reducing the balance
16                                  // … to be defined …
17                            }
18                      }
19                      else {return 0}
20                }
21                else {return 0}
22          }
  }
```

Artifacts for Implementation
Component Diagram



Artifacts for Implementation
Deployment Diagram

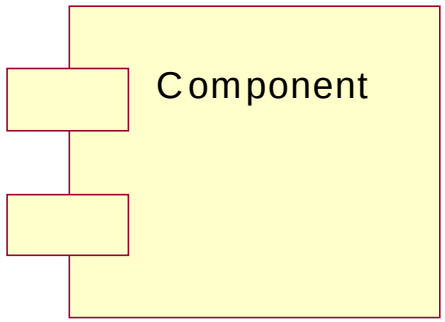


Component Diagram Notation: Component

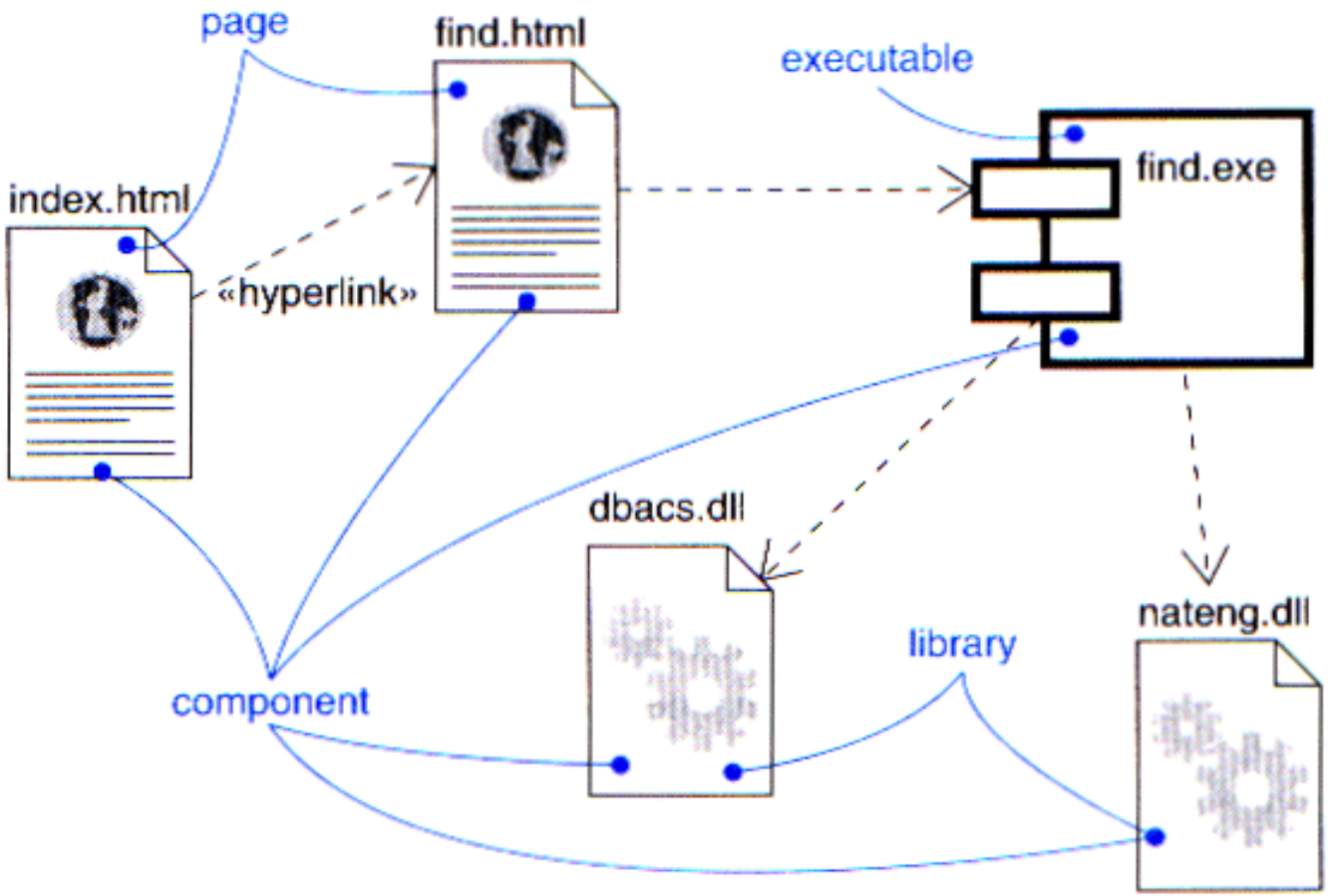
- A ***component*** represents a software module with a well-defined interface.
  - Source Code
  - Binary Code
  - Executable
  - DLL

Component Diagram
Notation: Component

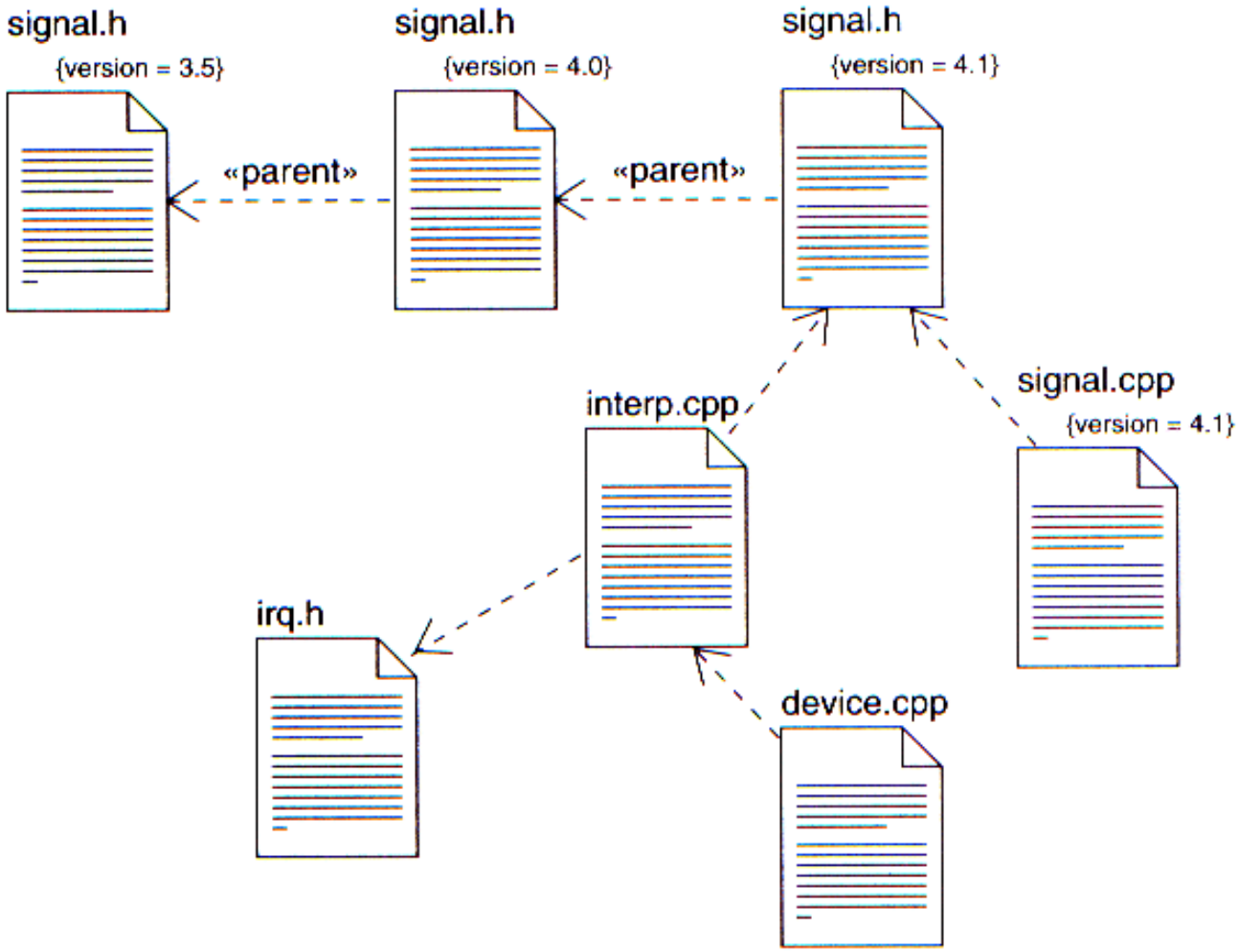
- A ***component icon*** is drawn as a large rectangle with two smaller rectangles attached to its left side.



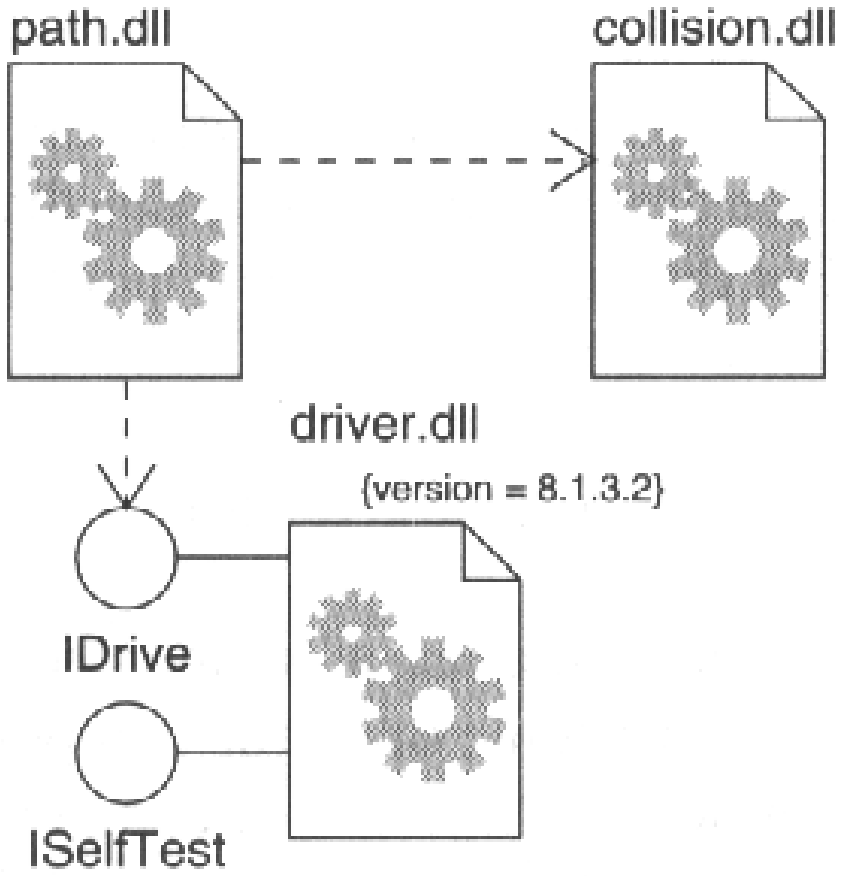
Component Diagram
Example 1: System



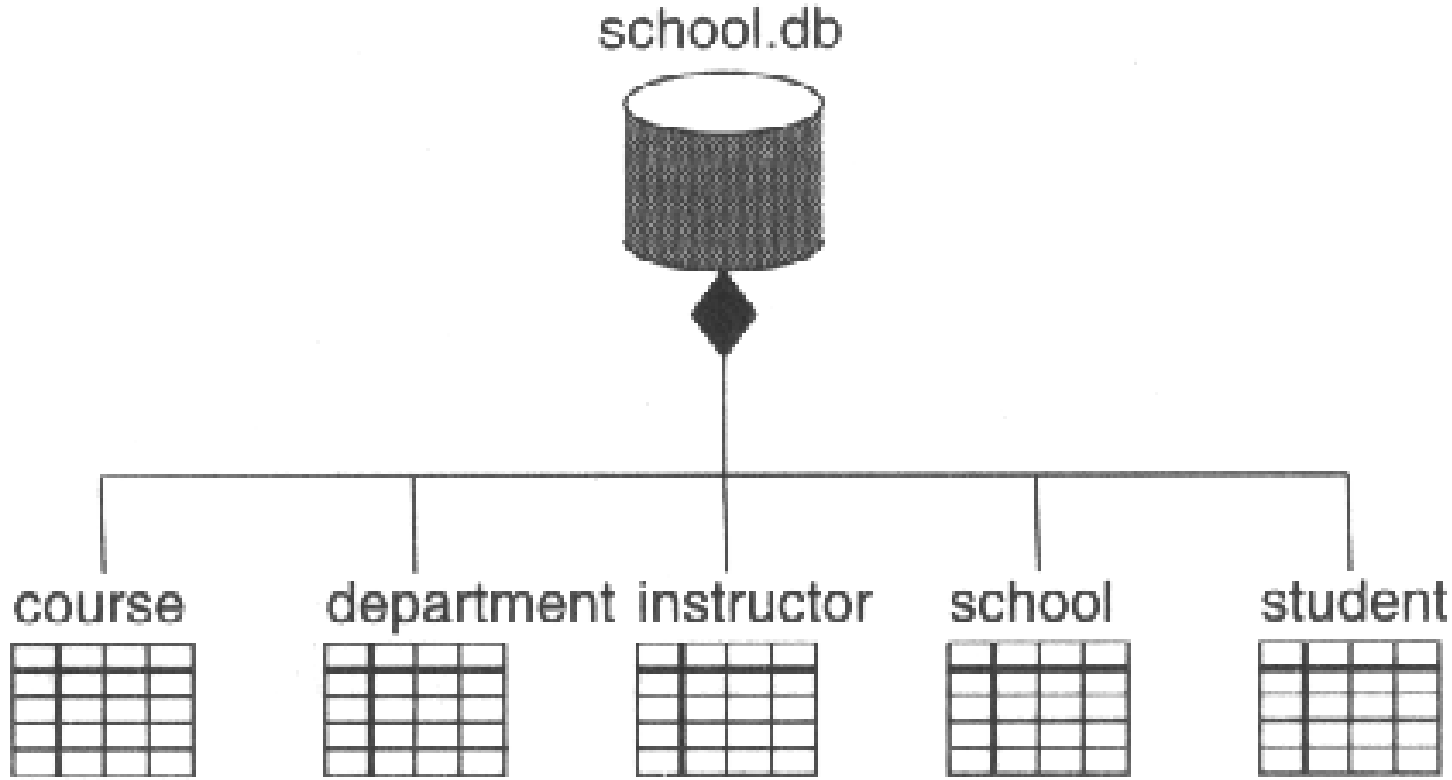
Component Diagram Example 2: Source Code



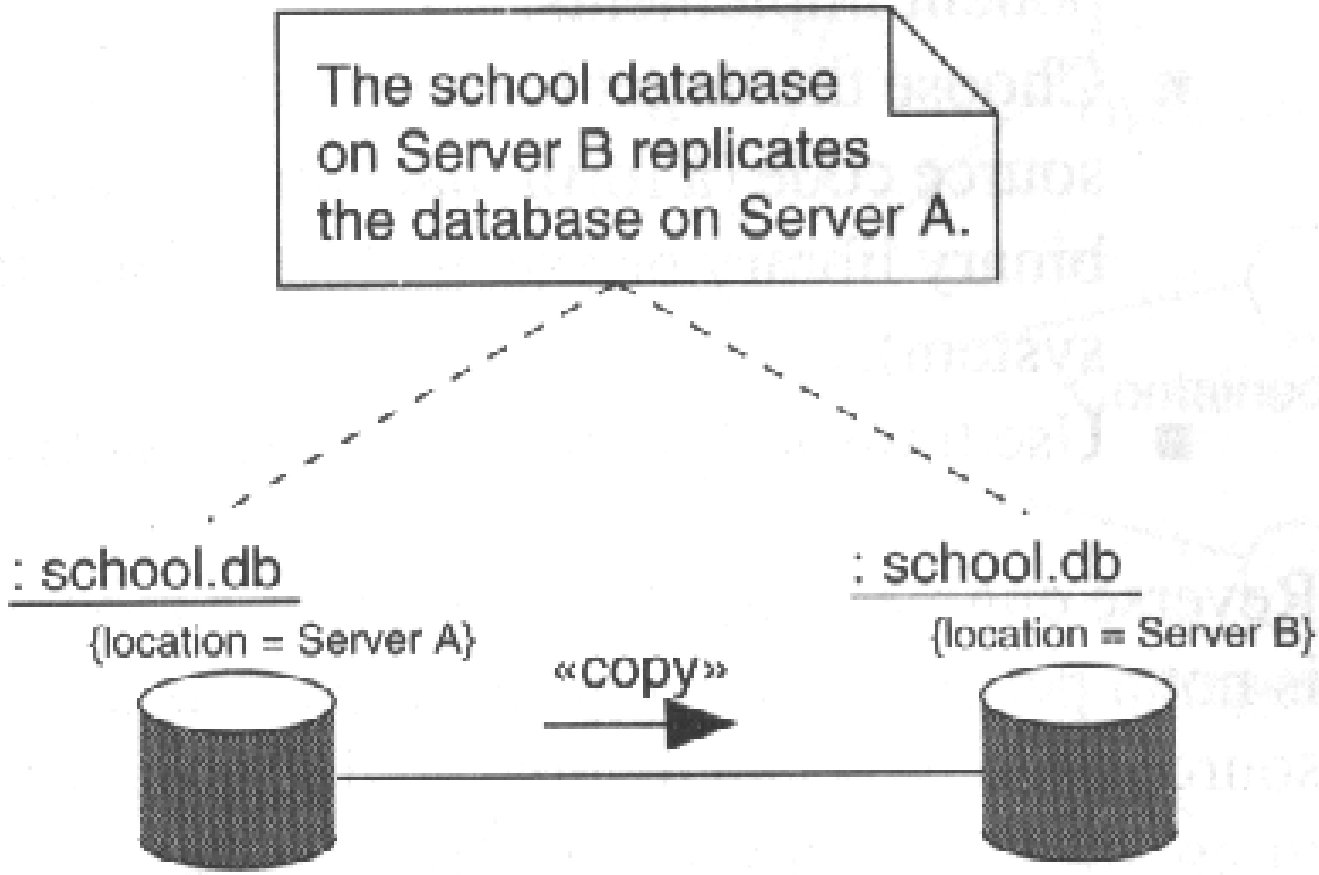
Component Diagram
Example 3: Executable Release



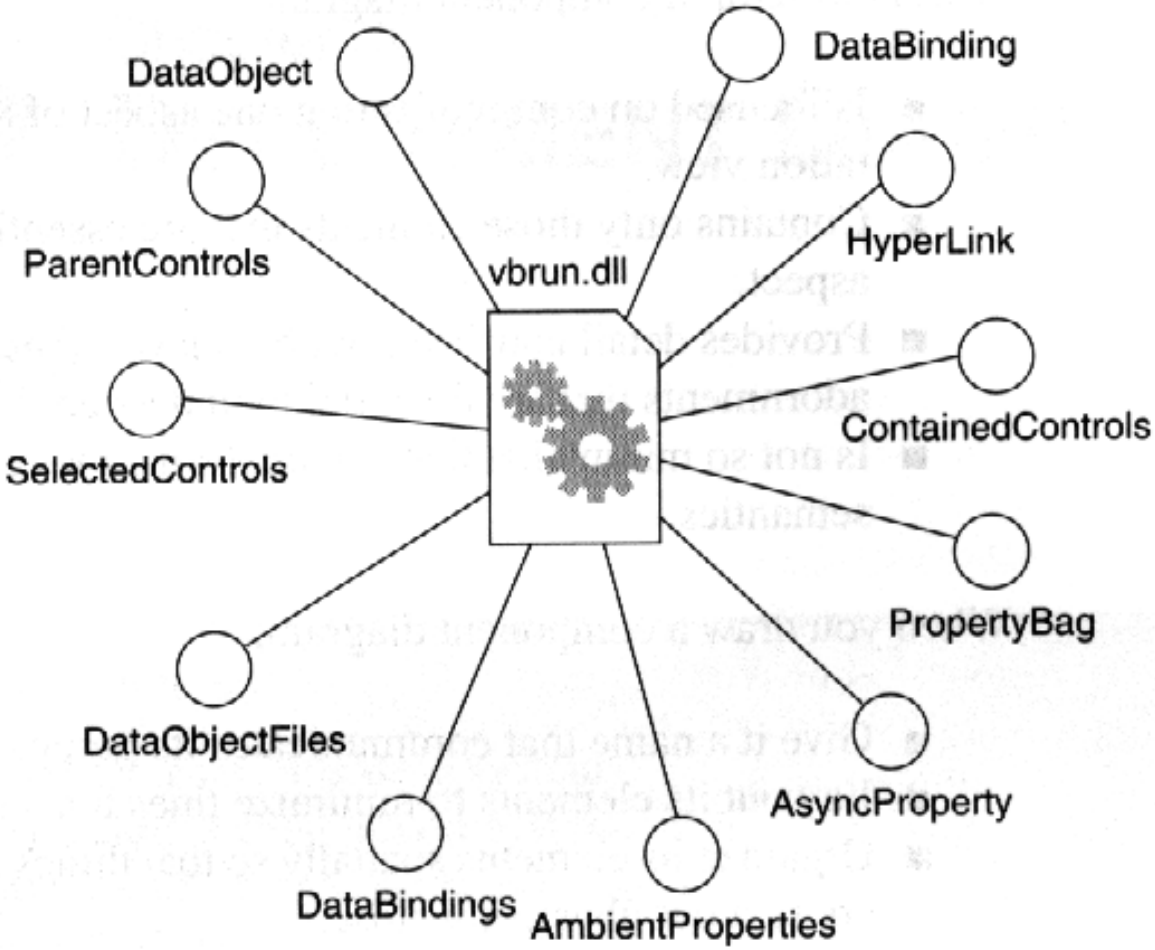
Component Diagram
Example 4: Physical Database



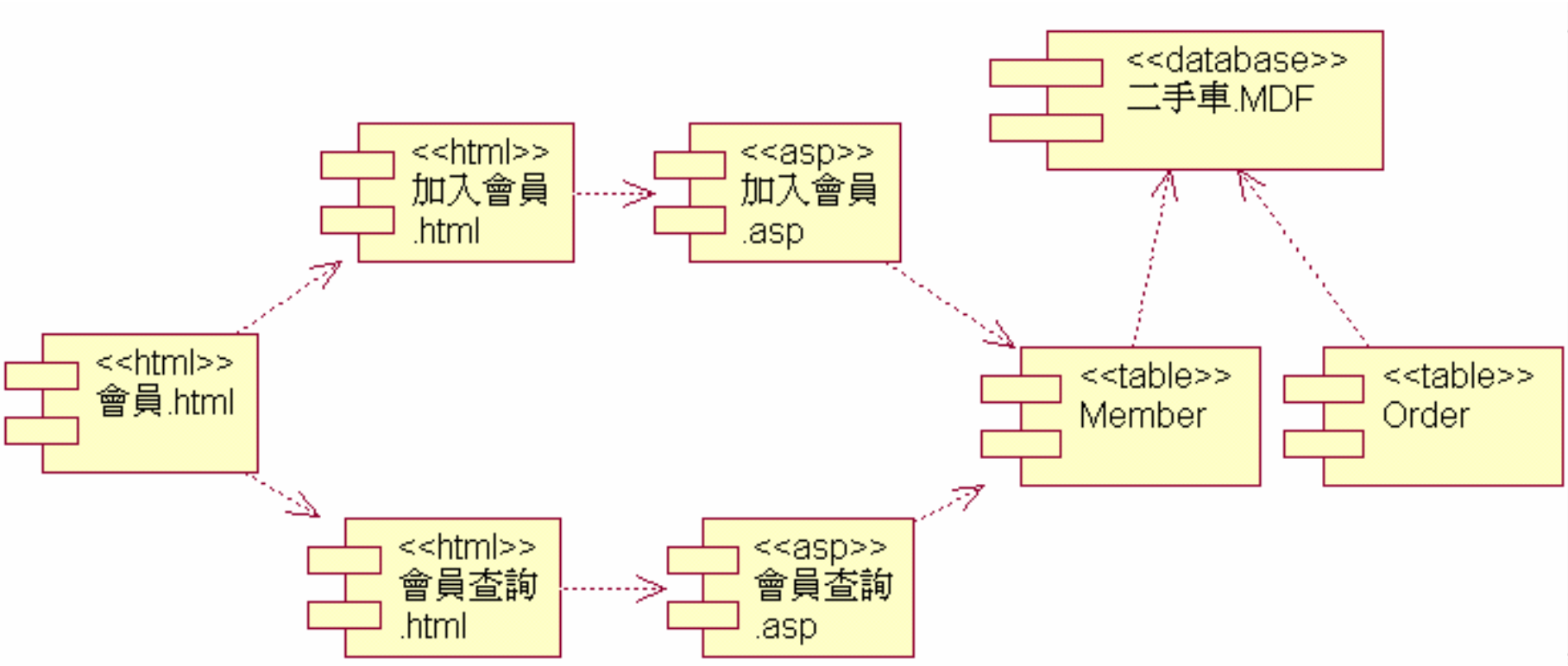
Component Diagram
Example 5: Adaptable System



Component Diagram Reverse Engineering



Component Diagram Example 6: 會員管理



表購物系統案例的元件

項次	類別　名稱	操作名稱	元件（模組）　名稱	元件（模組）功能
1	書籍產品型錄	顯示細部說明	書籍產品型錄物件類別模組(以VB程式語言撰寫之ActiveX　-DLL)	將書籍的所有欄位資料（編號、ISBN、出版社、單位、售價、作者、書名、類別、出版日期、版次）顯示在新網頁中
2	購物車	新增訂購項目 設定訂購數量 計算訂購總金額 刪除訂購項目 清空所有產品 確認採購訂單	購物車物件類別模組（以VB程式語言撰寫之ActiveX-DLL）	需完成以下的功能：新增訂購項目、設定訂購數量、計算訂購總金額、刪除訂購項目、清空所有產品、確認採購訂單

購物系統案例的元件（續）

項次	類別名稱	操作名稱	元件（模組）名稱	元件（模組）功能
3	訂單	新增訂單	訂單物件類別模組（以VB程式語言撰寫之ActiveX-DLL）	將記憶體變數（客戶編號、交貨地址）、及記憶體陣列變數（購物車）內的資料轉至後檯資料庫
4			使用者介面模組（以ASP程式語言撰寫之互動式網頁)	提供客戶上網訂購書籍時選擇執行下列功能：新增訂購項目、修改訂購數量、刪除訂購項目、取消採購訂單、確認採購訂單之操作介面
5			資料庫	儲存客戶、訂單、書籍產品型錄之資料

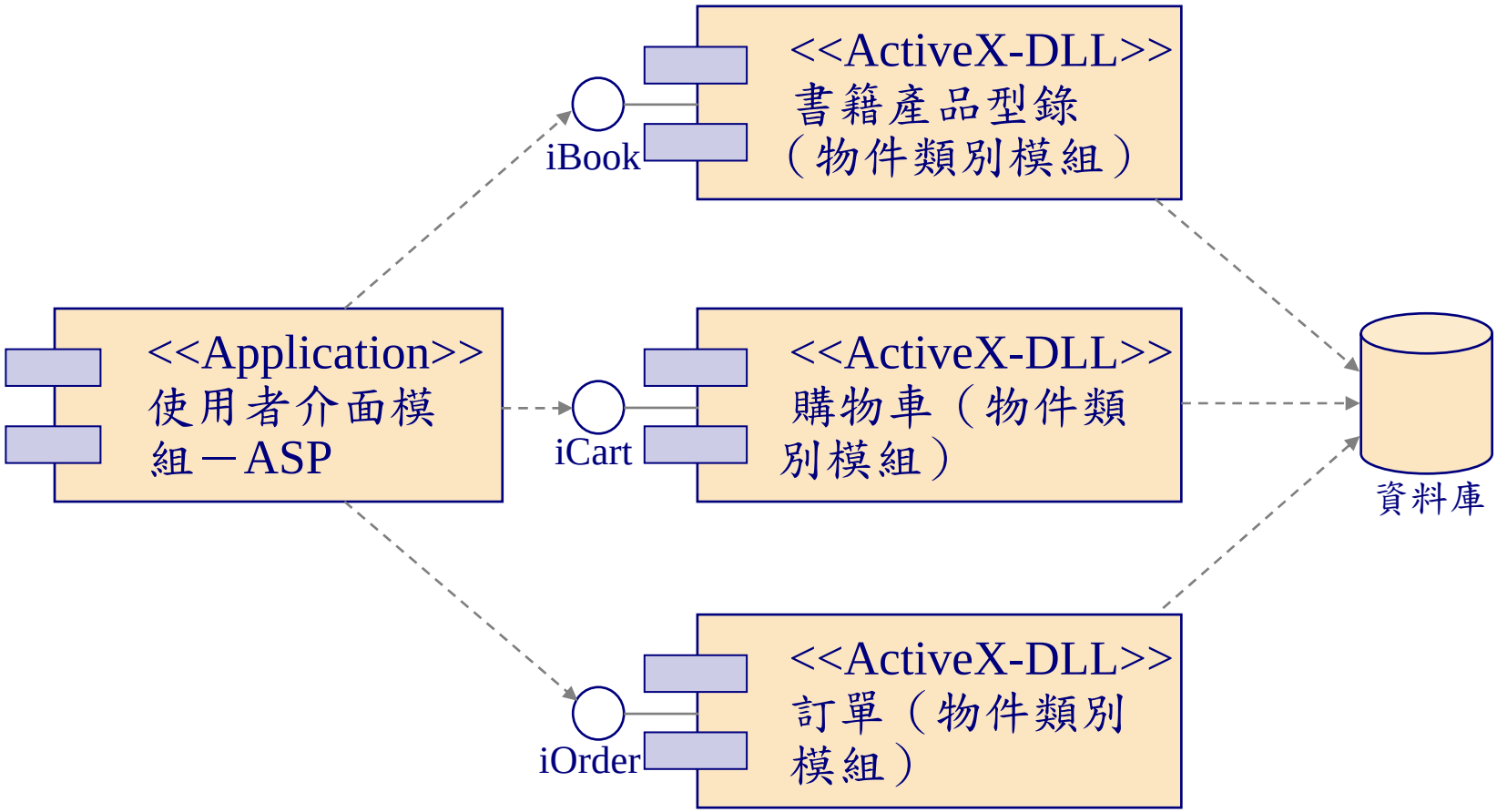
元件圖之建構案例

- 找出元件與關係
  - 「使用者介面模組」須透過iBook、iCart、iOrder三個介面完成下列功能：新增訂購項目、修改訂購數量、刪除訂購項目、取消採購訂單、確認採購訂單
  - 三個ActiveX-DLL：「書籍產品型錄物件類別模組」、「購物車物件類別模組」、及「訂單物件類別模組」皆可存取後端資料庫。

元件圖之建構案例（續）

- 繪製元件圖
  - 書籍產品型錄、購物車與訂單三個元件屬執行檔，且皆可存取後端資料庫，因此這三個執行檔與資料庫間有相依關係。
  - 使用者介面需透過iBook、iCart、iOrder三個介面完成訂購功能，所以使用者介面與這三個介面間也分別有相依關係。

購物系統元件圖

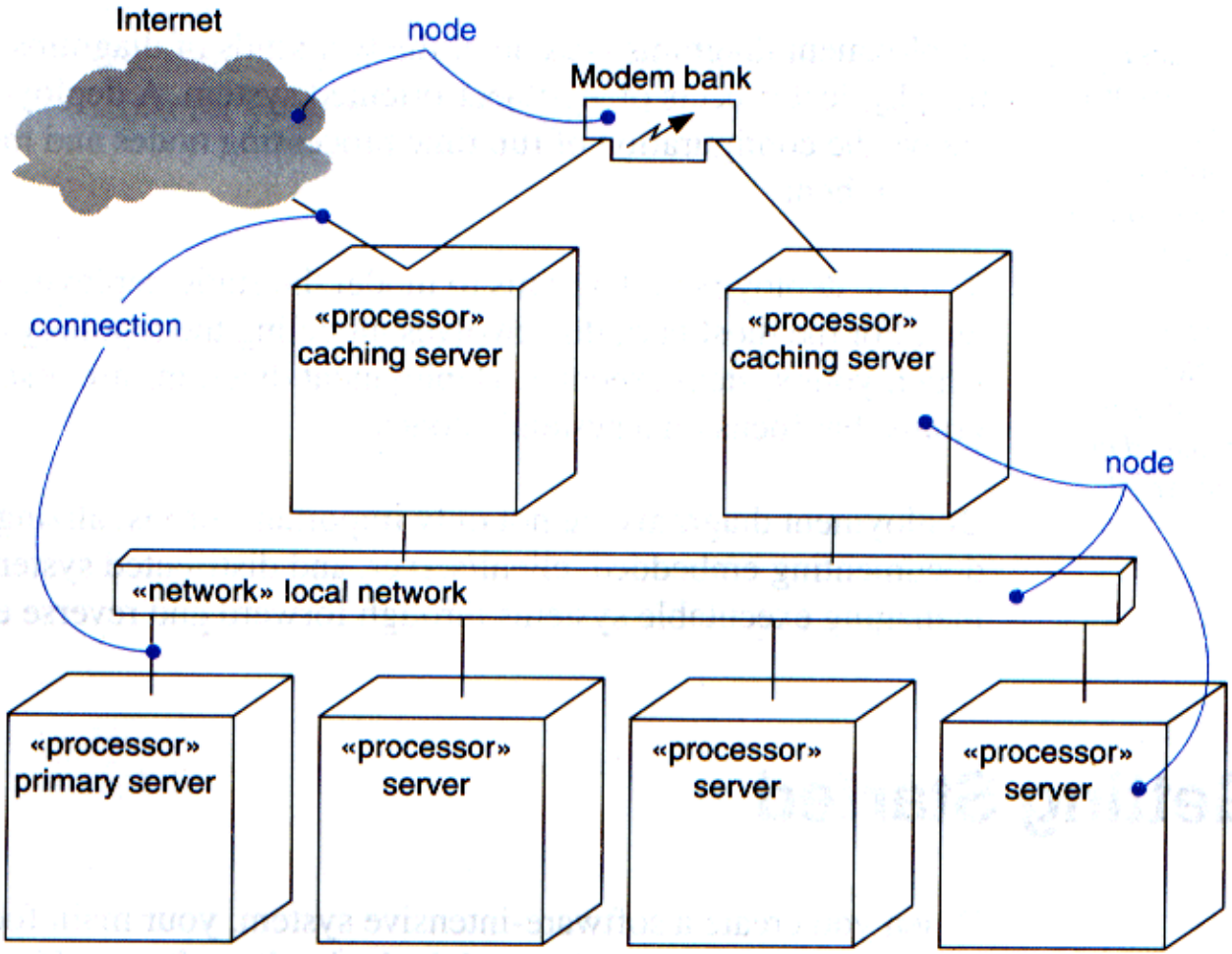

<<ActiveX-DLL>>
書籍產品型錄
（物件類別模組）
iBook
<<Application>>
使用者介面模
組－ASP
iCart
<<ActiveX-DLL>>
購物車（物件類
別模組）
資料庫
<<ActiveX-DLL>>
訂單（物件類別
模組）
iOrder

Deployment Diagram

Deployment Diagrams

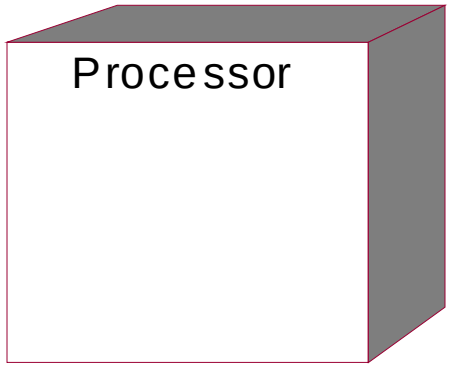
- A ***deployment diagram*** shows the configuration of run-time processing elements and the software components, processes, and objects that live on them.
- A deployment diagram consists of
  - ***Processors***
  - ***Devices***
  - ***Connections***

Deployment Diagrams
Notation Summary

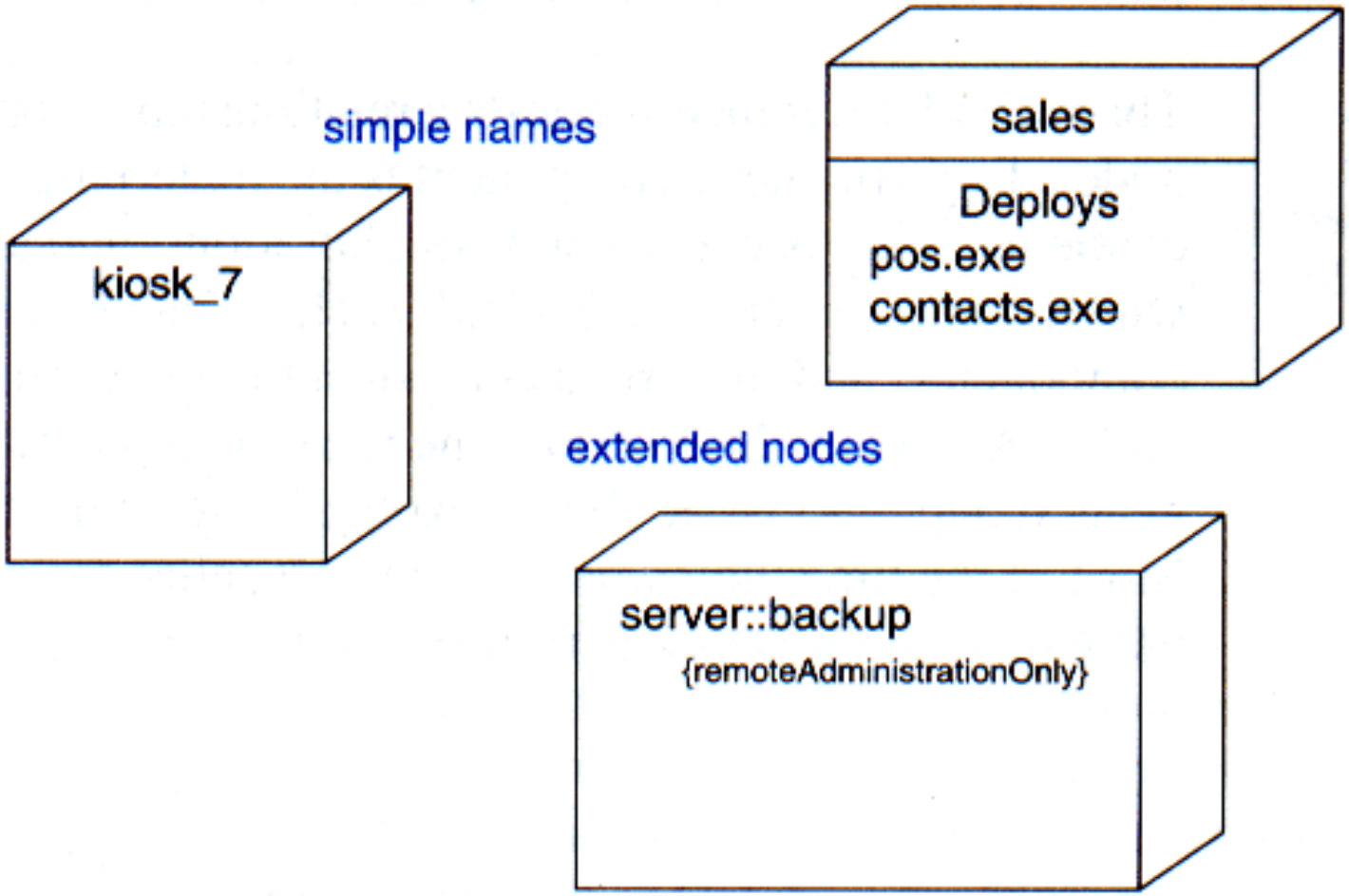


Deployment Diagrams
Notation: Processor

- A ***processor*** is a hardware component capable of executing programs.
- The icon for a processor is a shaded box:

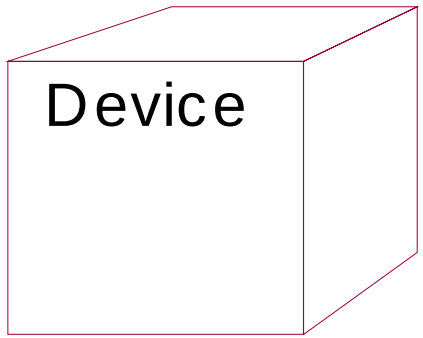


Deployment Diagrams
Notation: Processor

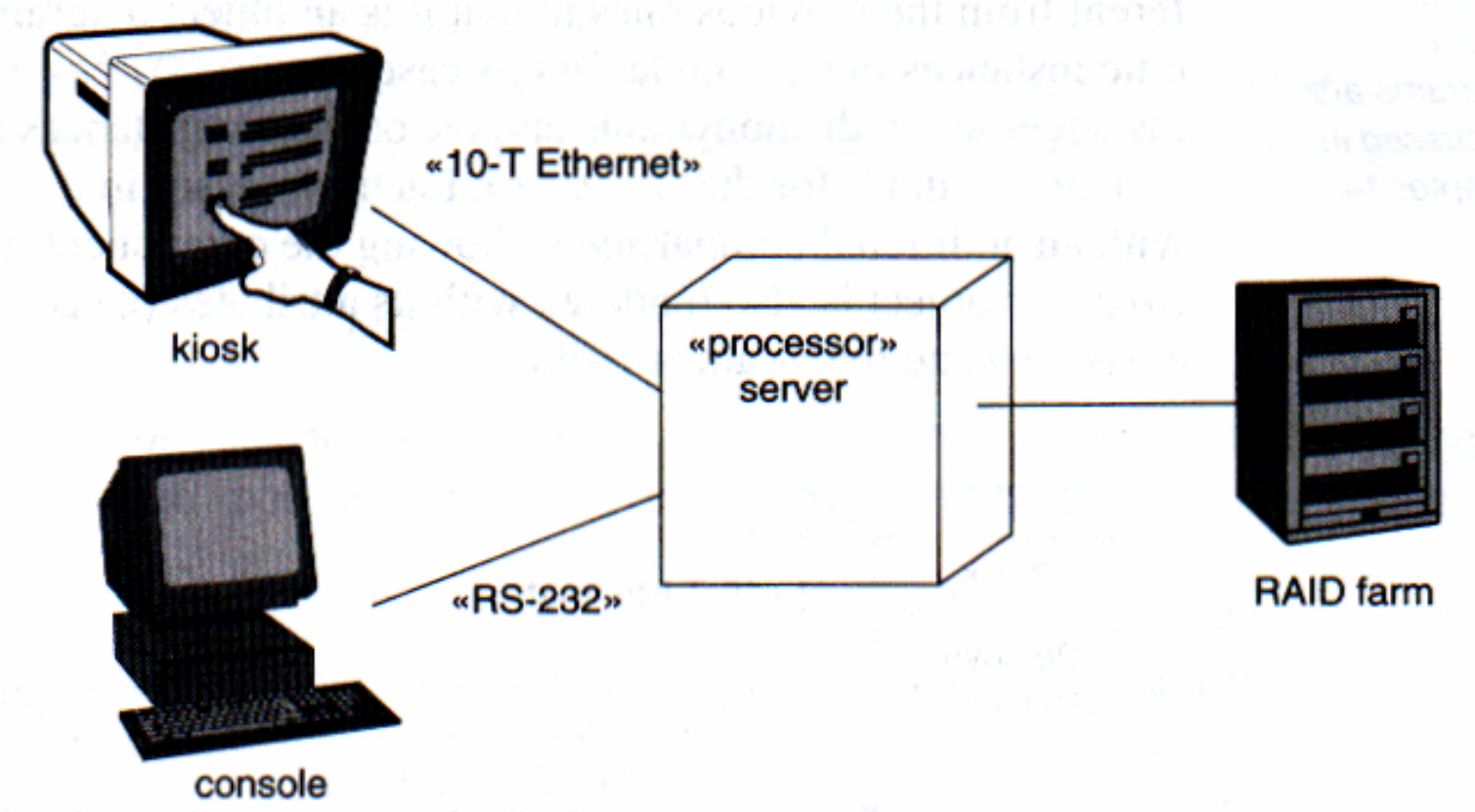


Deployment Diagrams
Notation: Device

- A ***device*** is a hardware component with no computing power.
- Each device must have a name. Device names can be generic, such as "modem" or "terminal."
- The icon for a device is a box.



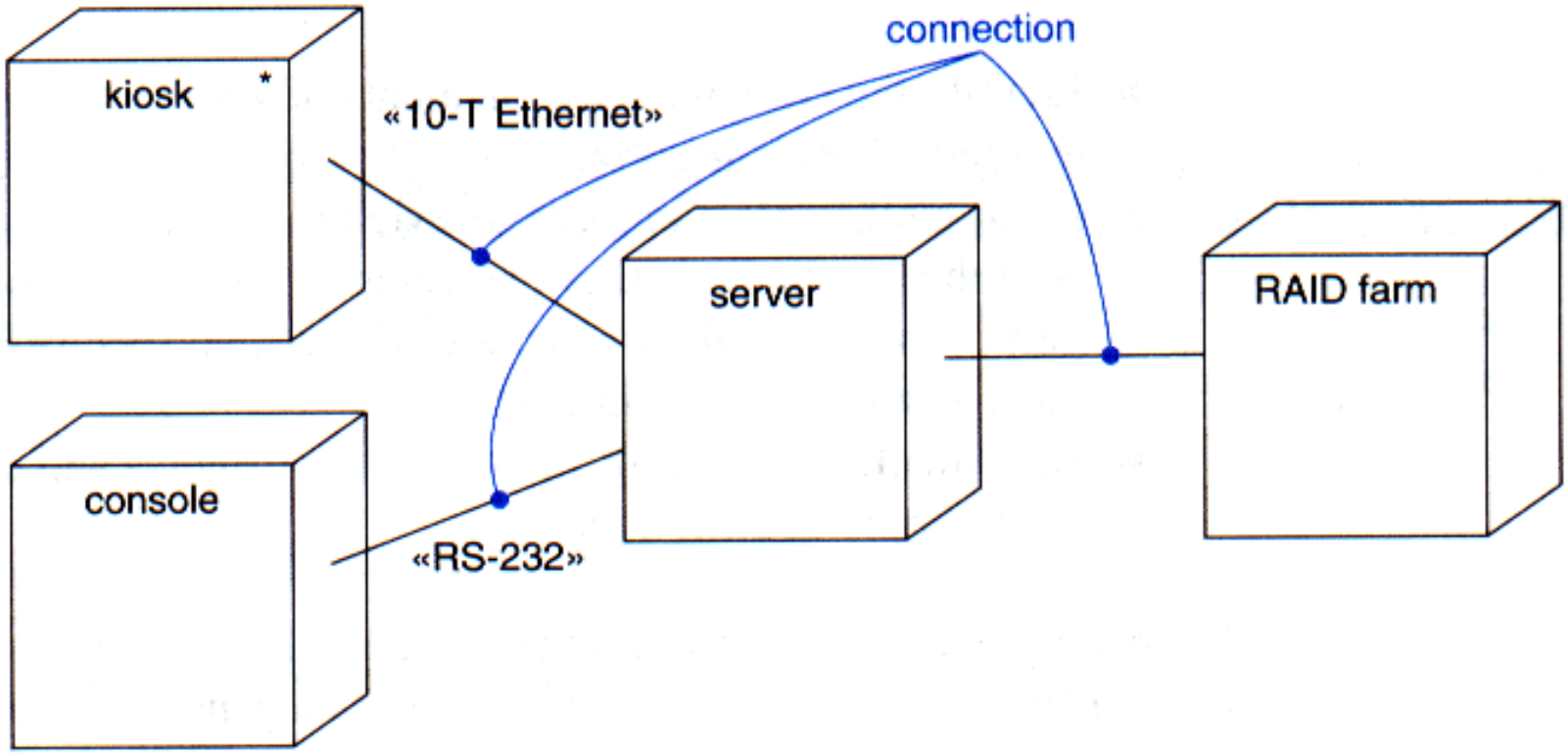
Deployment Diagrams
Notation: Device



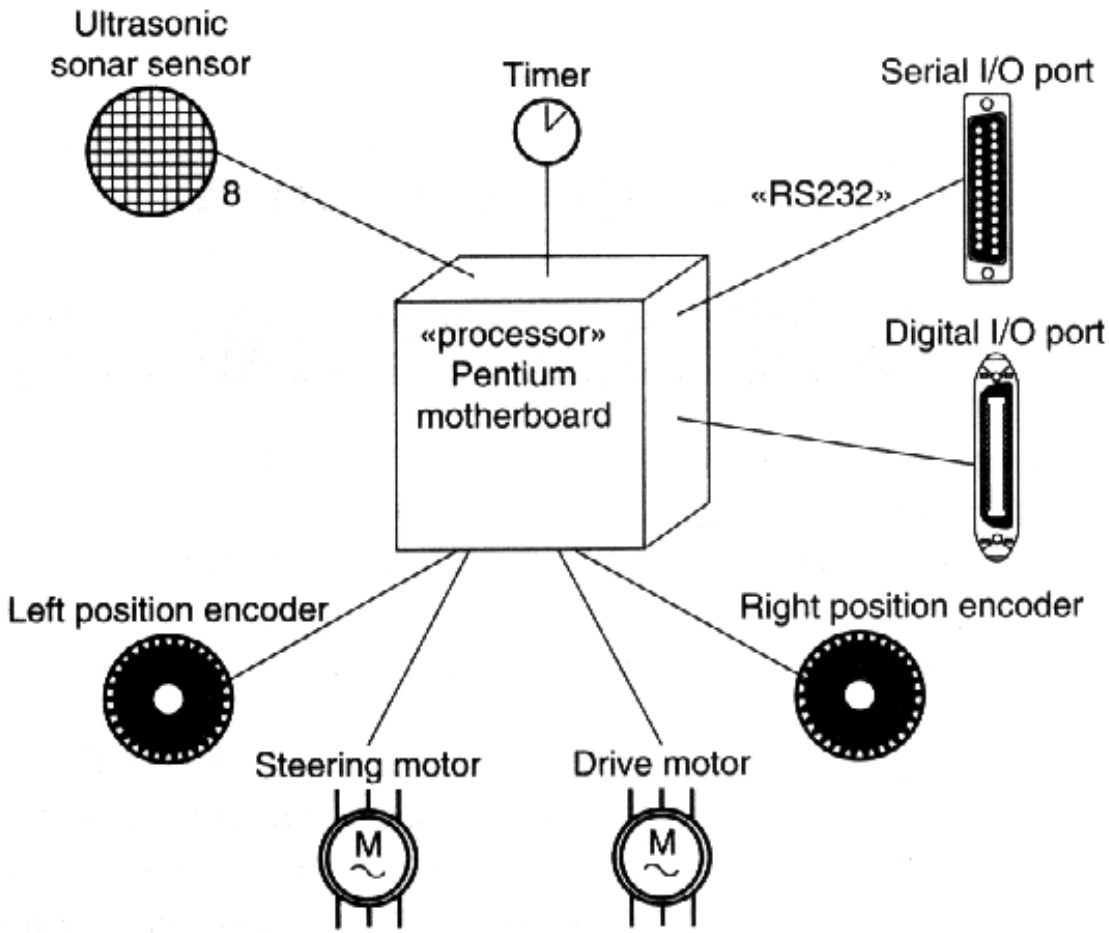
Deployment Diagrams
Notation: Connection

- A ***connection*** represents some type of hardware coupling between two entities.
  - An entity is either a processor or a device.
- The hardware coupling can be direct, such as an RS232 cable, or indirect, such as satellite-to-ground communication.
- Connections are usually bi-directional.
- The icon for a connection is a straight line.

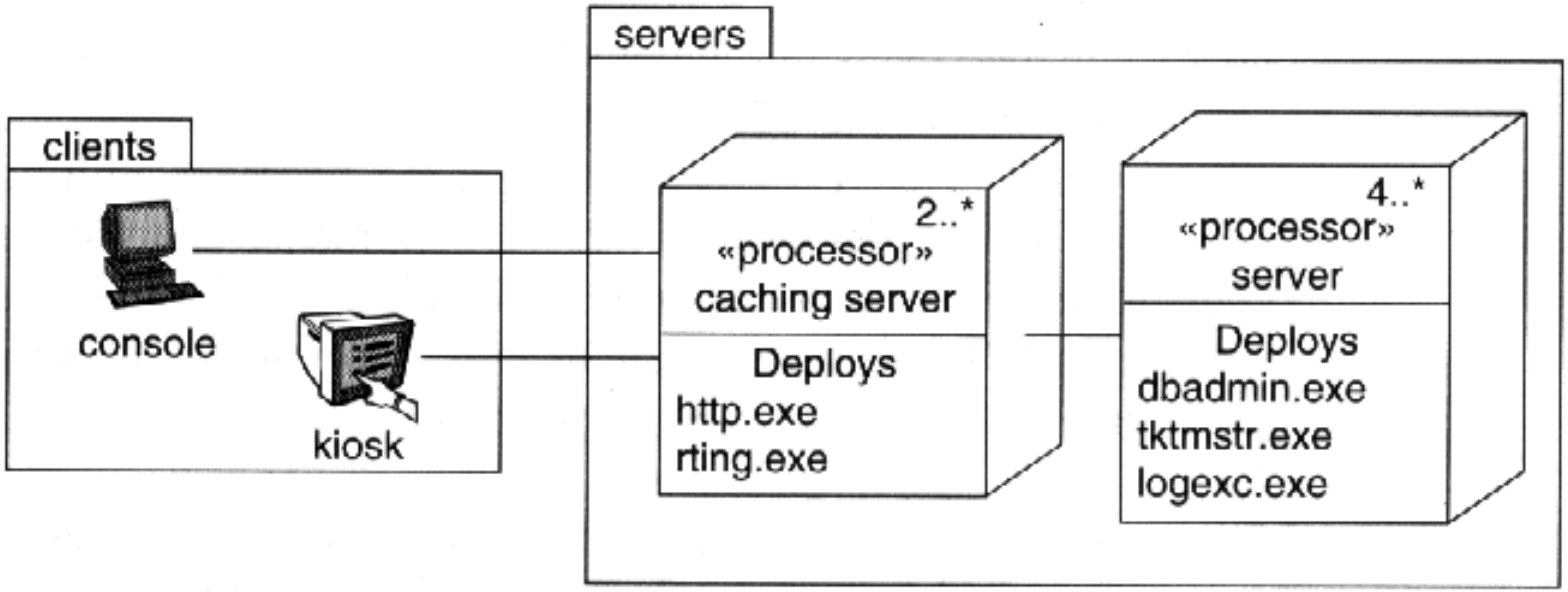
Deployment Diagrams
Notation: Connection



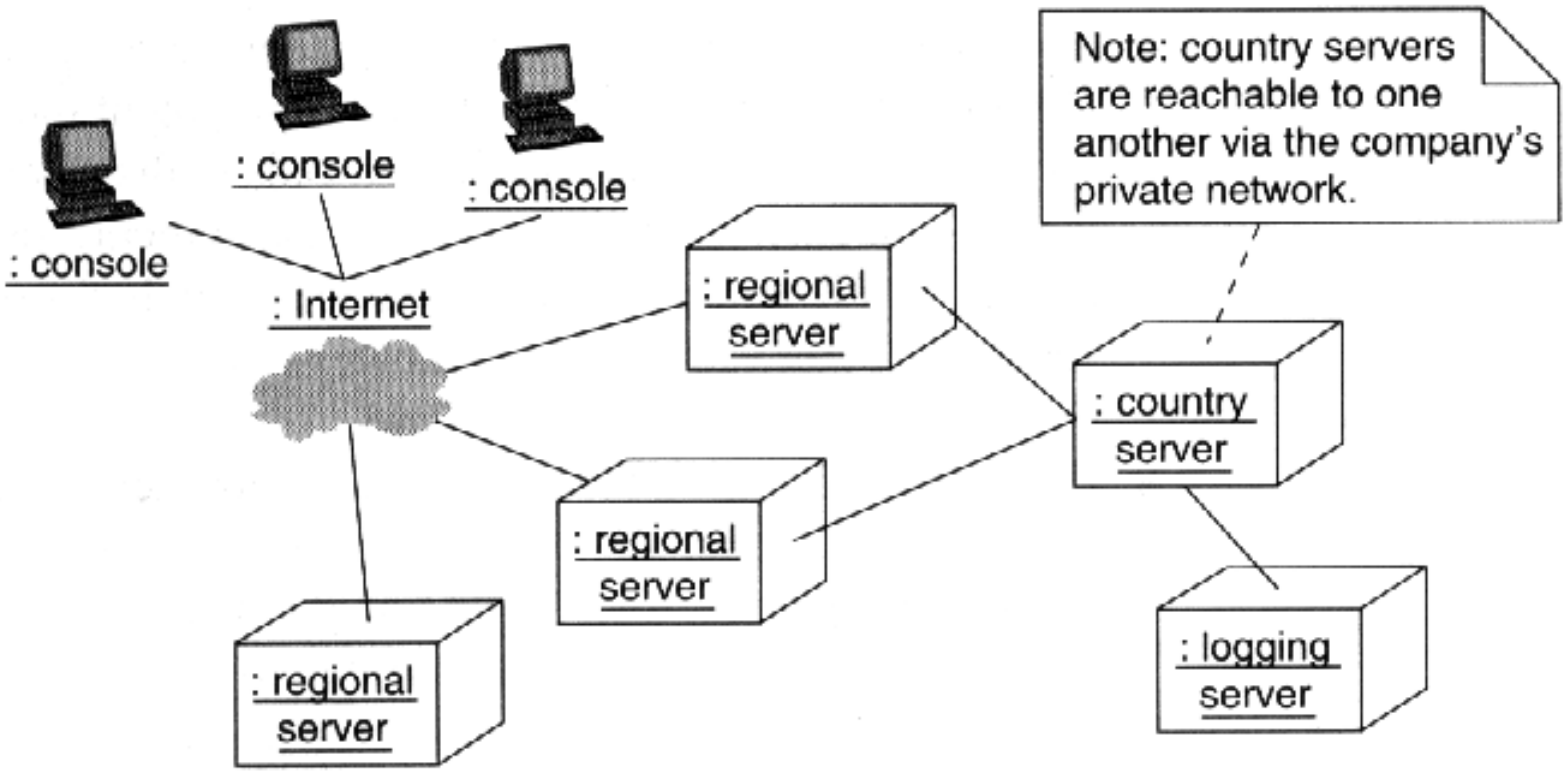
Example 1: Embedded Systems



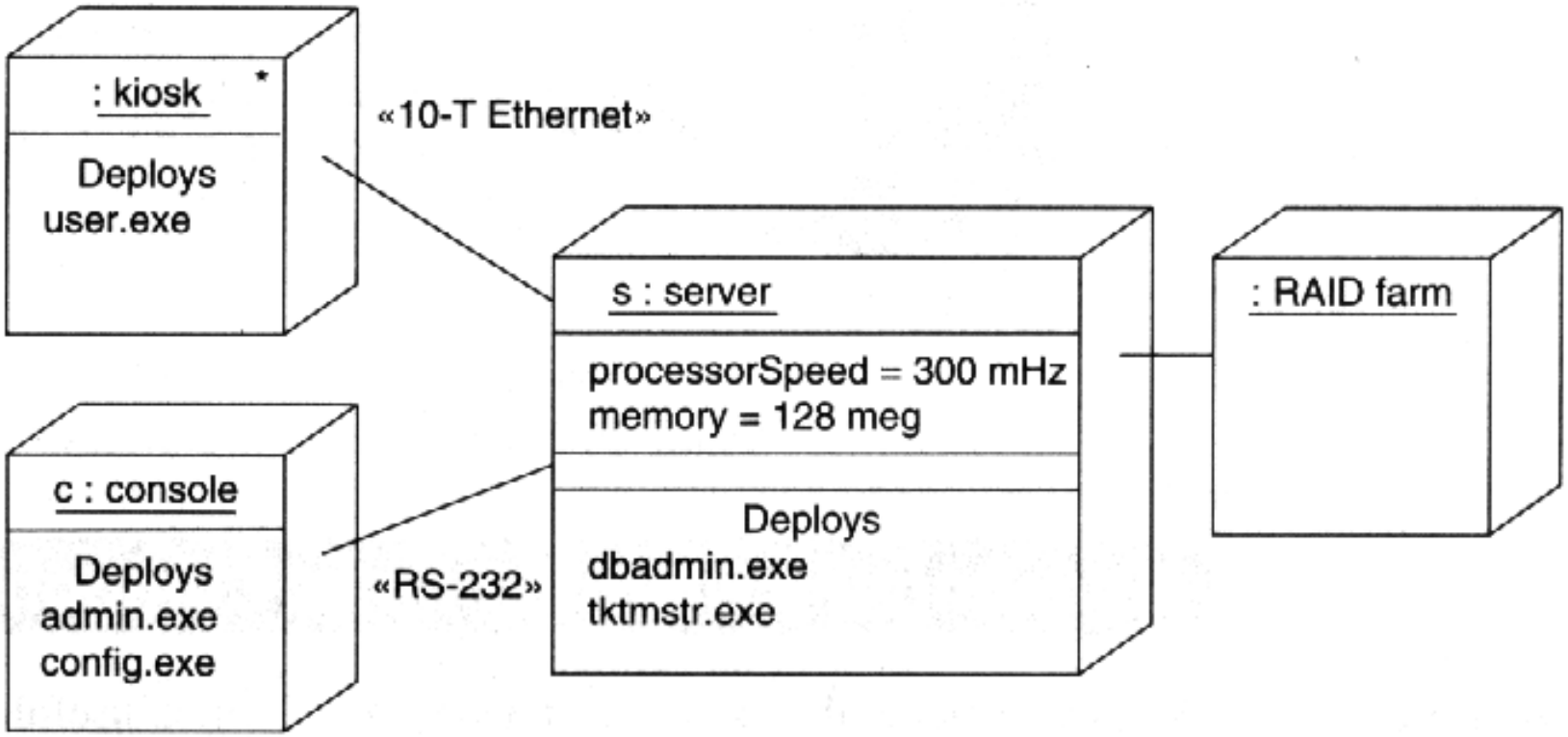
Example 2: Client/Server Systems



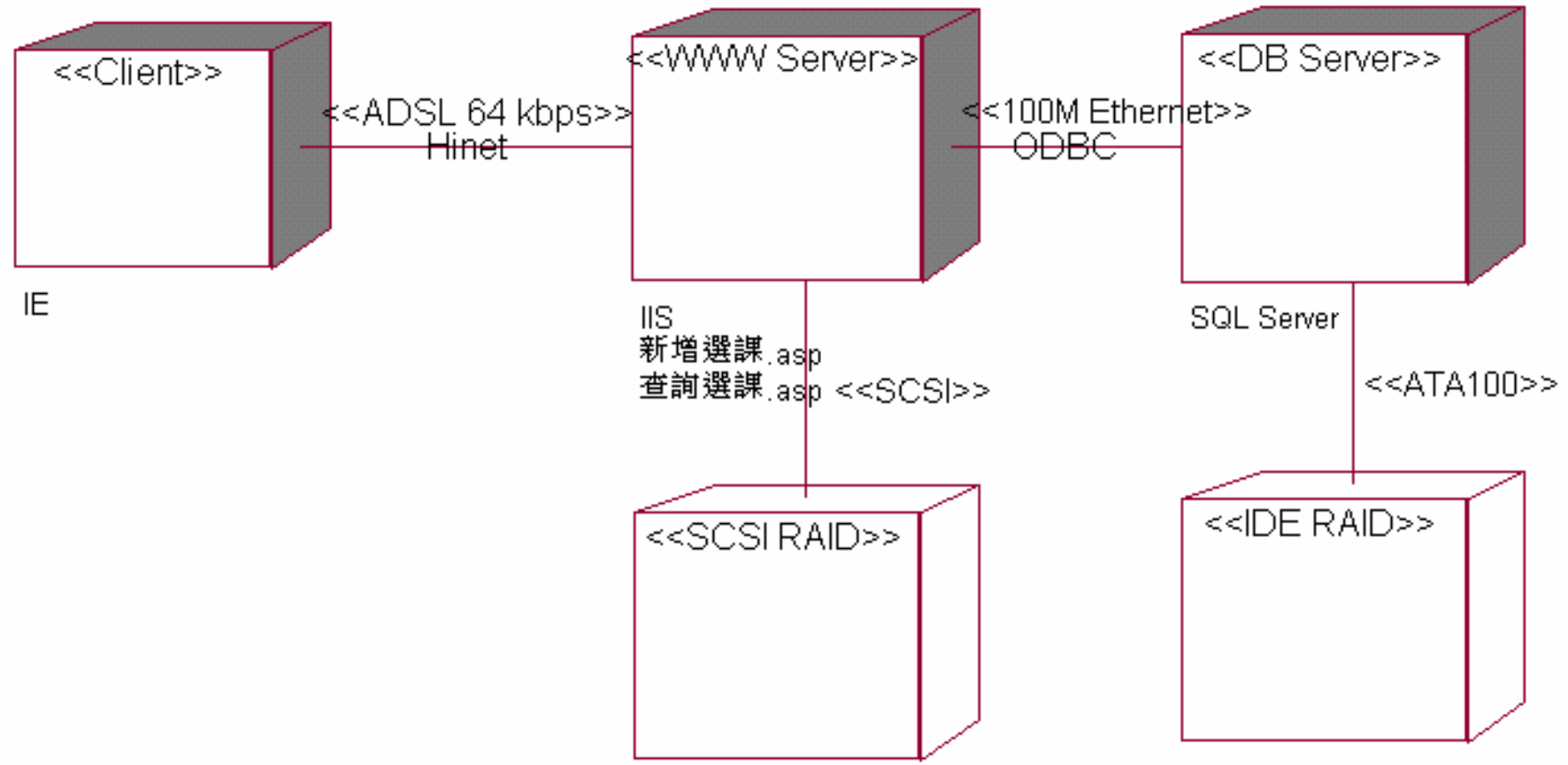
Example 3: Distributed Systems



Example 4: Distributed Database Systems

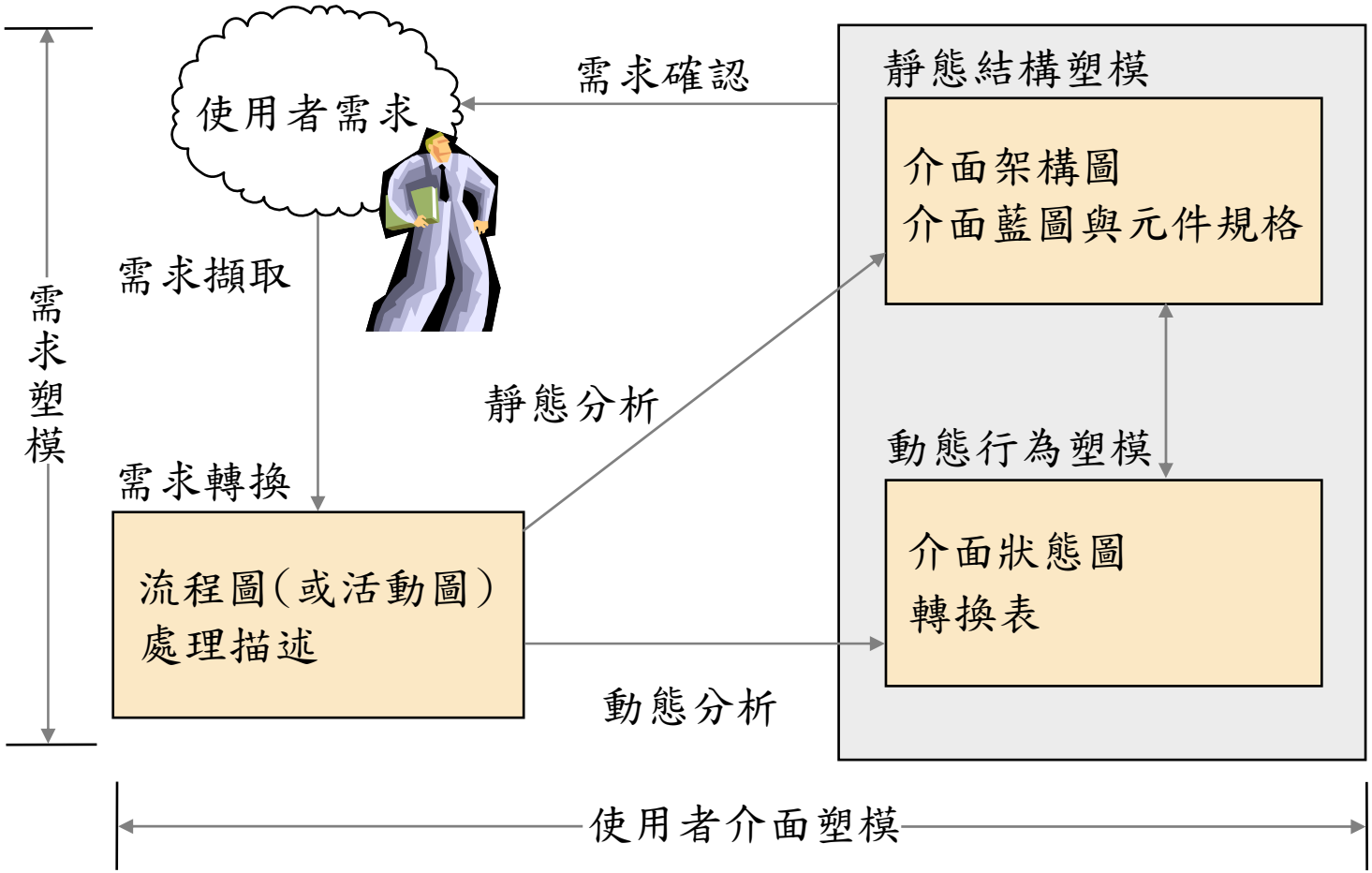


Example 5: Distributed Database Systems (ASP)



Interface Design

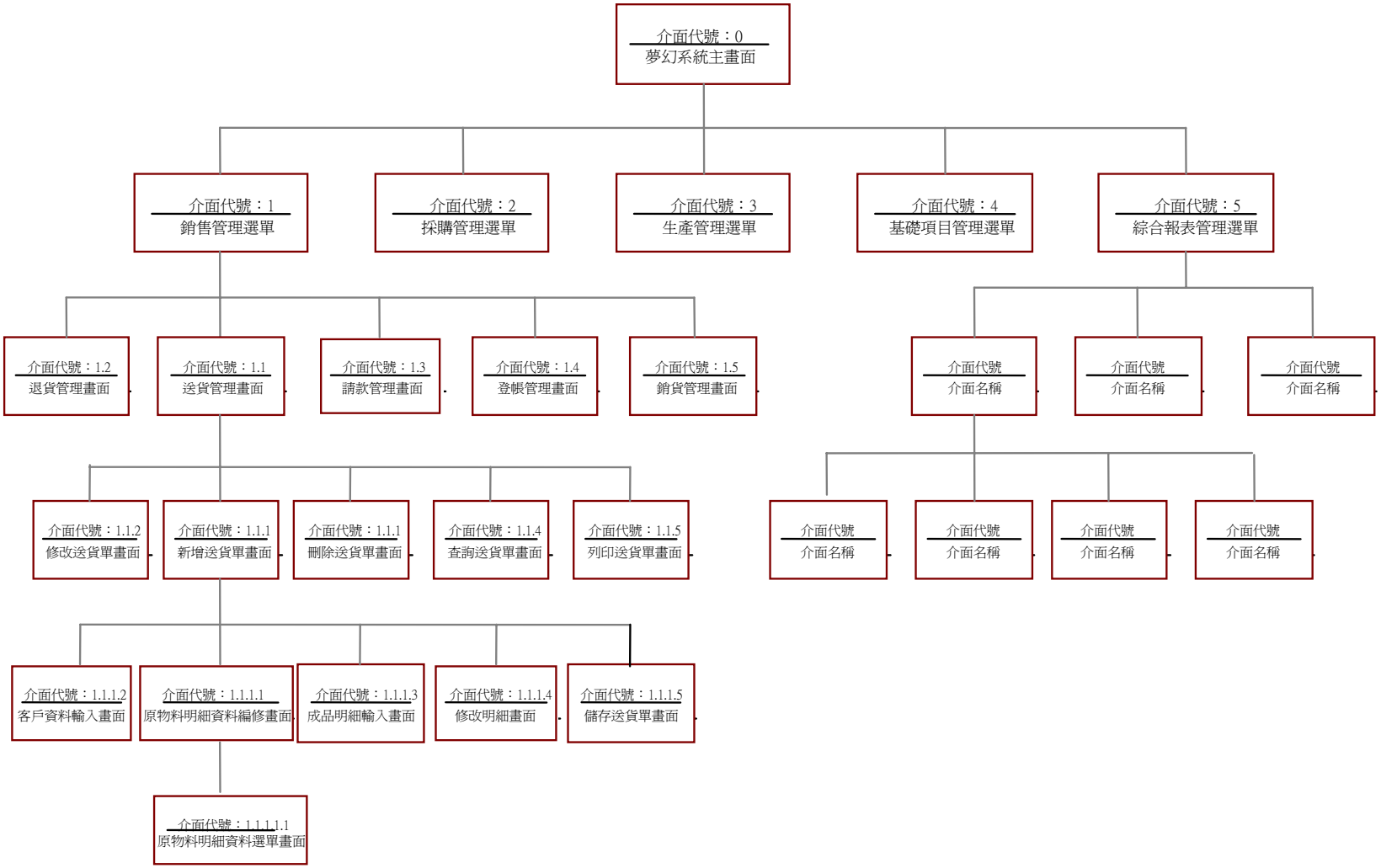
使用者介面塑模方法論


使用者需求
需求確認
靜態結構塑模
介面架構圖
介面藍圖與元件規格
需求擷取
需求塑模
靜態分析
動態行為塑模
需求轉換
介面狀態圖
轉換表
流程圖(或活動圖)
處理描述
動態分析
使用者介面塑模

介面元件範例

元件名稱	圖	元件名稱	圖
TextBox		Frame	
OptionButton		ResultTable	
CommandButton		ListBox	
SSTab		PictureBox	
Image		Grid	
TreeView		Label	
SelectedList		SortList	
MenuReference		HyperLink	
ComboBox		CheckBox	

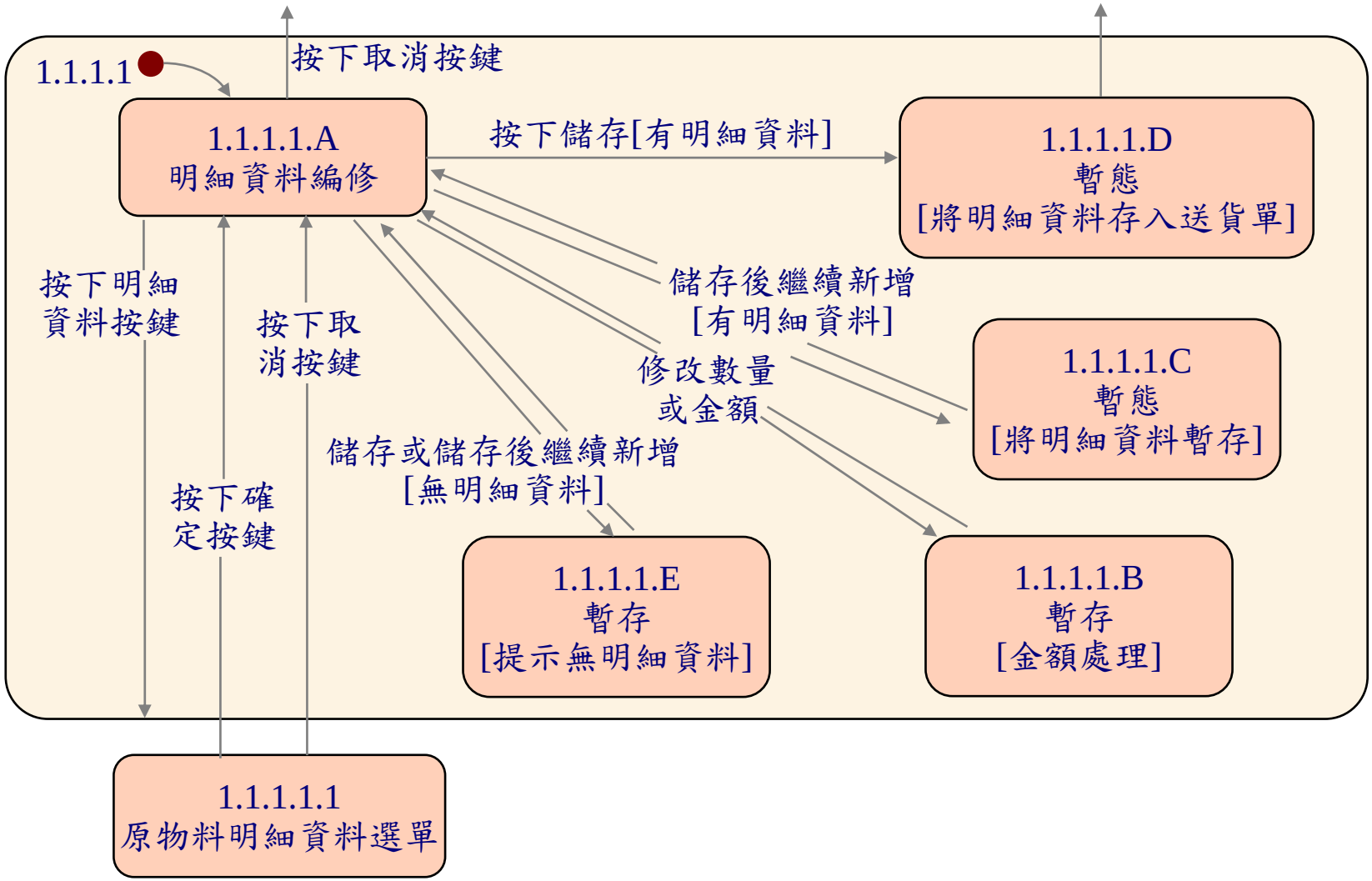
階層式介面架構圖


介面代號：0
夢幻系統主畫面
介面代號：1
銷售管理選單
介面代號：2
採購管理選單
介面代號：3
生產管理選單
介面代號：4
基礎項目管理選單
介面代號：5
綜合報表管理選單
介面代號：1.2
退貨管理畫面
介面代號：1.1
送貨管理畫面
介面代號：1.3
請款管理畫面
介面代號：1.4
登帳管理畫面
介面代號：1.5
銷貨管理畫面
介面代號
介面名稱
介面代號
介面名稱
介面代號
介面名稱
介面代號：1.1.2
修改送貨單畫面
介面代號：1.1.1
新增送貨單畫面
介面代號：1.1.1
刪除送貨單畫面
介面代號：1.1.4
查詢送貨單畫面
介面代號：1.1.5
列印送貨單畫面
介面代號
介面名稱
介面代號
介面名稱
介面代號
介面名稱
介面代號
介面名稱
介面代號：1.1.1.2
客戶資料輸入畫面
介面代號：1.1.1.1
原物料明細資料編修畫面
介面代號：1.1.1.3
成品明細輸入畫面
介面代號：1.1.1.4
修改明細畫面
介面代號：1.1.1.5
儲存送貨單畫面
介面代號：1.1.1.1.1
原物料明細資料選單畫面

元件規格

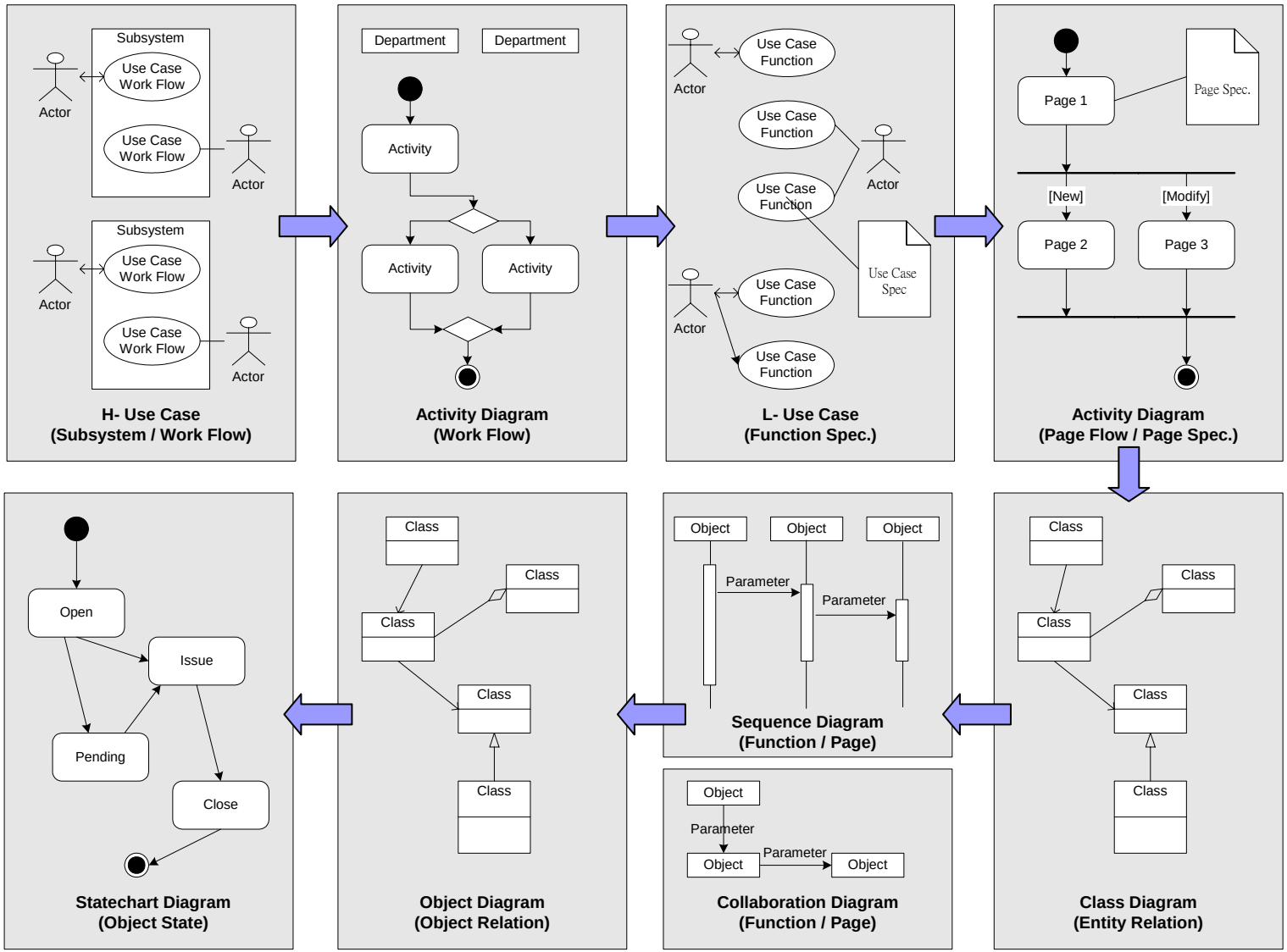
畫面代號	1.1.1.1	畫面名稱	原物料明細編修
畫面說明	原物料明細資料編修畫面		
元件名稱	元件類型	元件功能及概念說明	
送貨單編號	Label	呈現介面1.1.1中的送貨單編號	
折數	Label	呈現介面1.1.1中的折數	
原物料編號	TextBox	查詢使用者輸入的原物料編號之資料	
新增原物料	Command Button	進入介面1.1.1.1.1（原物料資料選單）	
名稱	Label	呈現使用者輸入編號之原物料名稱	
單位	Label	呈現使用者輸入編號之原物料單位	
規格	Label	呈現使用者輸入編號之原物料規格	
尺寸	Label	呈現使用者輸入編號之原物料尺寸	

介面1.1.1.1之介面狀態圖範例

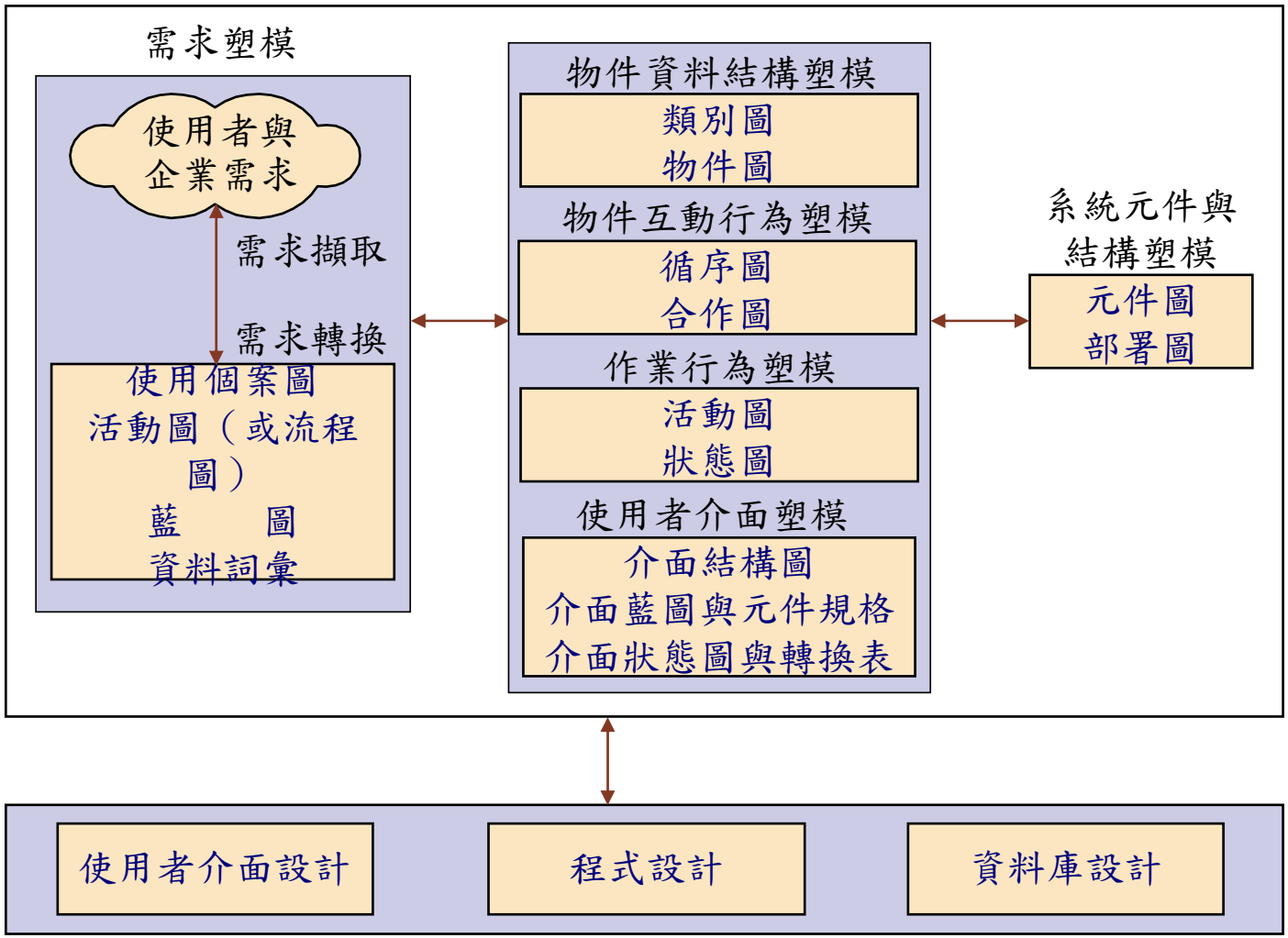


Conclusion

(UML-Diagram’ s Relationship)



物件導向塑模活動及塑模工具



物件導向分析與設計與塑模工具之關係