

第三章 物件導向技術



內容大綱

學習目標

- 3.1 導論
- 3.2 模組化系統理論
- 3.3 物件導向的基本概念
- 3.4 物件導向分析與設計及塑模工具
- 3.5 結論



學習目標

詳讀本章，你至少能瞭解：

- 物件導向的基本概念。
- 物件導向分析與設計及塑模工具。
- 物件導向系統開發的過程與產出。



3.1 導論

- ❑ 物件導向技術 (Object-Oriented Technique) 是繼結構化技術後，系統開發上的新思維典範 (Paradigm)。
- ❑ 物件導向 (Object-Oriented) 之觀念起源於模擬語言，以物件模式描述真實系統，並將「資料抽象化」、「封裝」、「繼承」與「同名異式」的觀念融入系統開發中。



3.2 模組化系統理論(1/5)

□ 模組化系統理論(Modular Systems Theory)：

- 將所有實體(例如社會的、生物的、技術的)視為是階層化地巢狀系統(Hierarchically Nested Systems)，意味在任何分析的單元，這實體是一種元件的系統，且其每一元件是更細小元件的系統，一直分解到元件是最基本粒子或直到我們無法再分解(Schilling, 2000)。
- 是系統分析與設計很重要的理論基礎，該理論闡明模組化(Modularity)、模組設計及模組化系統(Modular System)概念及應具備之特性等。



3.2 模組化系統理論(2/5)

- ❑ 模組化 (Modularity)：是一種一般化的系統概念 (General Systems Concept)，被用於描述一個系統的元件(Component)可被分開與重組的程度，也可說是元件間的耦合緊密度與系統結構的規則使元件能被分開與重組的程度。
- ❑ 模組化系統(Modular System)：一個模組化系統是由許多具有上述特性模組的有機組成，可完成某目的或功能。



3.2 模組化系統理論(3/5)

- 一個好的模組設計需同時具備高內聚力 (High Cohesion) 與低耦合力 (Low Coupling) 兩種特性。
 - 內聚力是指模組內機能(或是功能)之間相關的程度，亦即模組內的功能相關程度。
 - 耦合力係指模組之間的相關強度，亦即模組與外部模組間的相依程度。

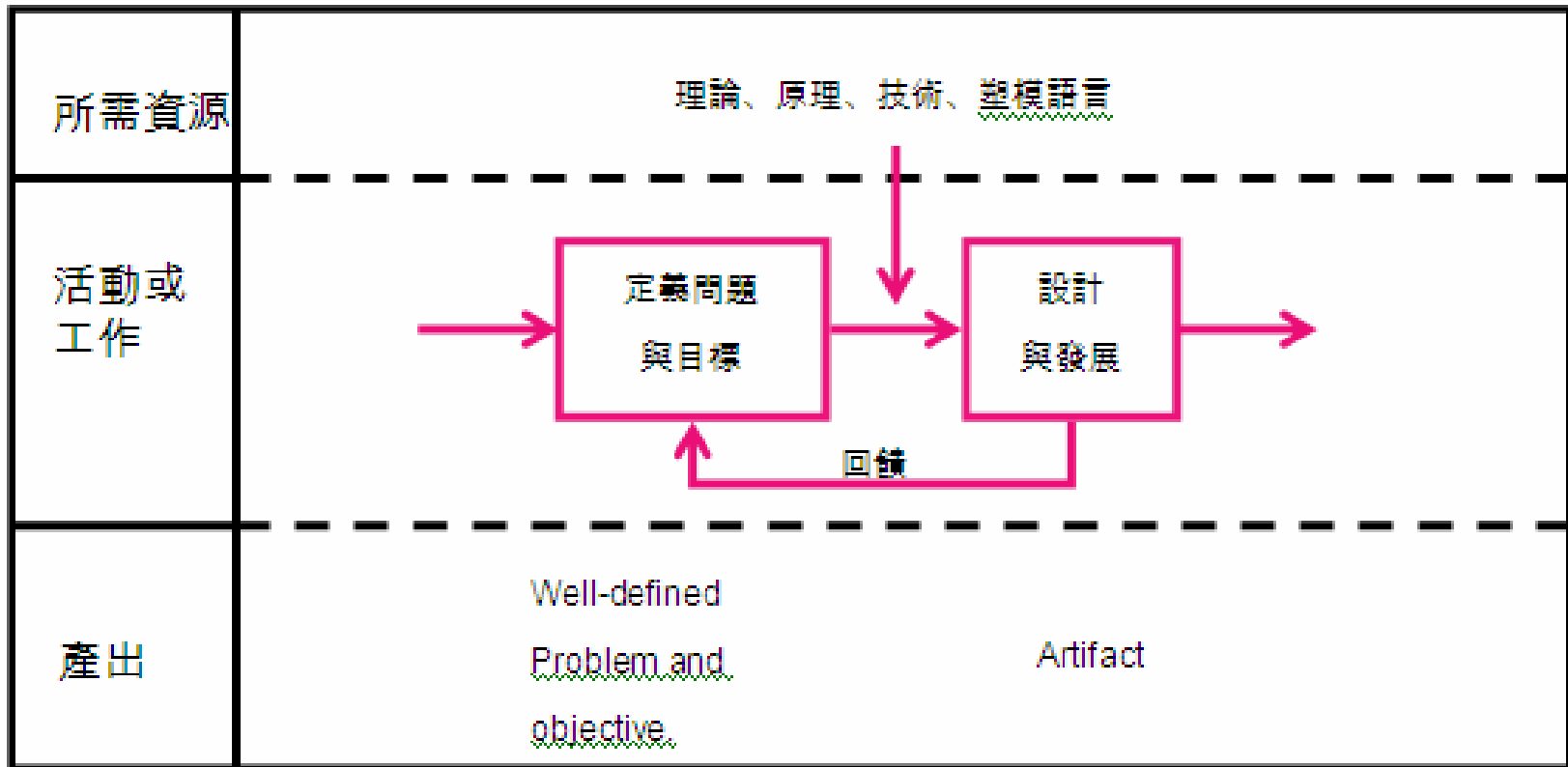


3.2 模組化系統理論(4/5)

- 當一個模組化系統的元件可被分解與重組成新的配置(可能用一些不同元件替代)，且僅降低些微的功能，則我們稱該系統具有高程度的模組化。
- 一個較佳設計的模組化系統與架構能夠提升軟體開發的效率。



3.2 模組化系統理論—設計科學研究方法(5/5)



3.3 物件導向的基本概念(1/13)

□ 物件導向的基本觀念：

- 物件
- 類別
- 抽象化
- 封裝
- 封裝
- 繼承
- 同名異式



3.3 物件導向的基本概念(2/13)

□ 物件

- 物件 (Object) 是物件導向的基本思維單位，為具有狀態 (State)、行為 (Behavior) 與識別 (Identity) 的實體 (Entity) 或抽象化概念 (Abstract Concept)，其行為會影響其狀態。
- 物件包括：
 - 屬性 (Attributes)
 - 操作 (Operations) 或稱方法 (Methods)
- 例如張三之物件，有姓、名、工時與工時薪資為其屬性，而工時薪資計算為其操作。



3.3 物件導向的基本概念(3/13)

□ 類別

- 類別 (Class) 為具有相同結構及行為之物件所組成的集合。一個類別是一種定義(Definition)、樣版(Template) 或模型(Mold)，是許多物件共同特徵(例如屬性、操作和語意) 的一種描述。
- 類別為物件經抽象化(Abstraction) 所得的結果，剔除物件間的差異，並僅考慮其相同性質後，將這些物件組成一個群體稱為類別。
- 例如，張三物件屬於「計時員工」類別。



3.3 物件導向的基本概念(4/13)

□ 抽象化

- 抽象化(Abstraction)意即將真實世界某個領域中的複雜現象整理出重要的特性，以簡化的模式、方法或模型予以描述，並剔除在領域外的特性。
- 抽象化有兩種特性：
 - 共通性：從物件中擷取共通的特徵，並在類別中描述其共同的屬性或操作。
 - 具有範圍：所謂的具有範圍指的是類別中已定義了屬性及操作，而其物件的屬性及操作則不會超出類別所定義的範圍。



3.3 物件導向的基本概念(5/13)

計時員工

— 姓名
— 工時
工時薪資

+ 組合姓名()
+ 計算工時薪資()

計件員工

— 姓名
— 件數
件數薪資

+ 組合姓名()
+ 計算件數薪資()



3.3 物件導向的基本概念(6/13)

- 類別使用一組相同的「屬性」和「操作」來描述一至多個物件。
- 類別有時亦稱物件類型 (Object Type)，類別中之任一物件則稱該類別之實例 (Instance)，因此物件與實例可視為同義詞。



3.3 物件導向的基本概念(7/13)

□ 封裝

- 物件導向技術將資訊及操作此資訊的方法包裝成一個物件，稱為封裝 (Encapsulation)。
- 封裝所成的物件，其結構可分為兩部分：
 - 介面 (Interface)：用以定義物件之外觀行為
 - 實作 (Implementation)：用以存放抽象化結果及描述如何達成外觀行為。



3.3 物件導向的基本概念(8/13)

- 封裝將物件的實作細節隱藏，使其與外界環境隔離，而只允許該物件包含之操作修改其資訊，稱為資訊隱藏(Information Hiding)。
- 使用物件時，僅需瞭解物件提供何種操作，而不需知其內部之資訊或行為如何表達或執行。外界之物件僅能透過訊息傳遞，要求該物件提供服務，而無法逕自改變該物件之資訊內容。
- 封裝之特性可使物件導向的系統較容易維護。



3.3 物件導向的基本概念(9/13)

□ 繼承

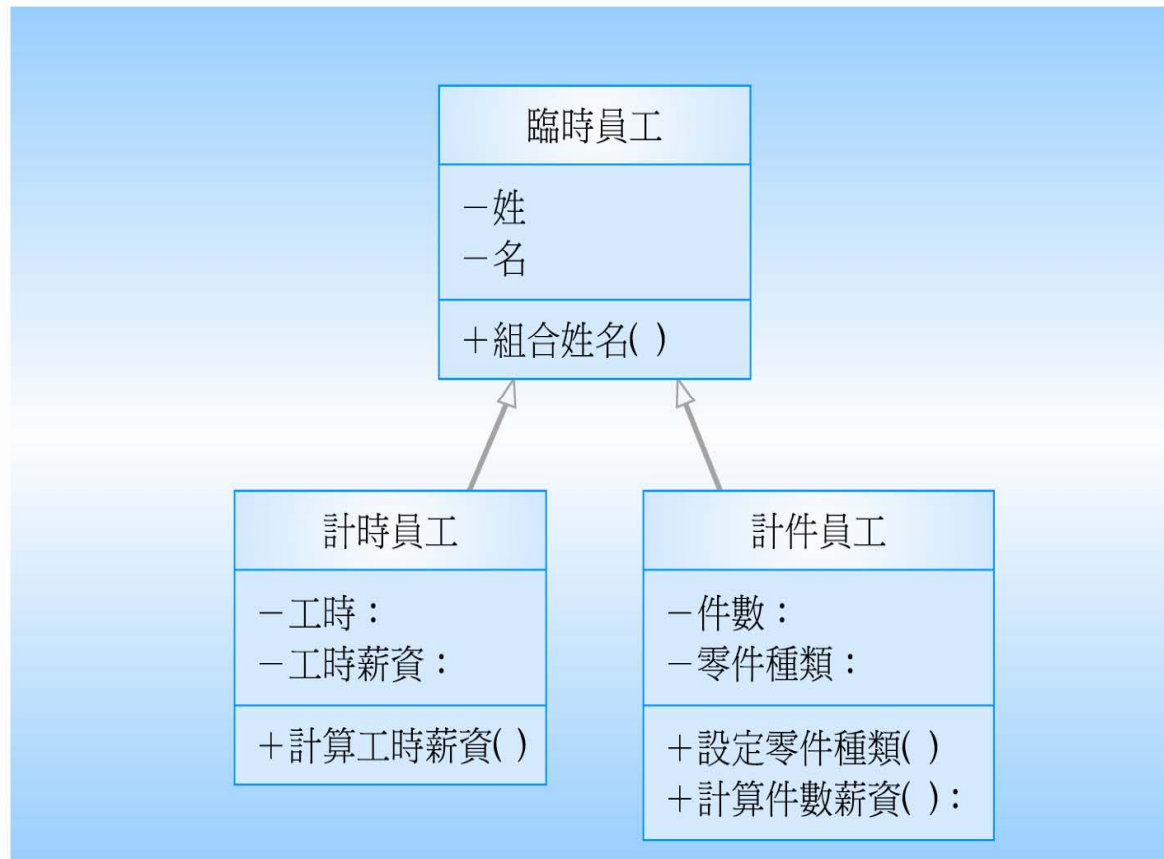
- 繼承 (Inheritance)在物件類別之間有一種“is a”的關係，也就是說繼承者是被繼承者的組成物件。
- 繼承之特性可以一般化 (Generalization) 與特殊化 (Specialization) 原則，萃取相關子類別 (Subclass) 中相同屬性和操作，歸類為一個父類別 (Superclass) 來達成。
- 物件導向的技術中，繼承概念可達成程式碼再用(Reuse) 與減少重複描述、產生可靠度較高的軟體，對軟體工程有革命性的影響。



3.3 物件導向的基本概念(10/13)

Figure

圖 3-2 類別間之繼承關係



3.3 物件導向的基本概念(11/13)

□ 同名異式，簡稱「多型」

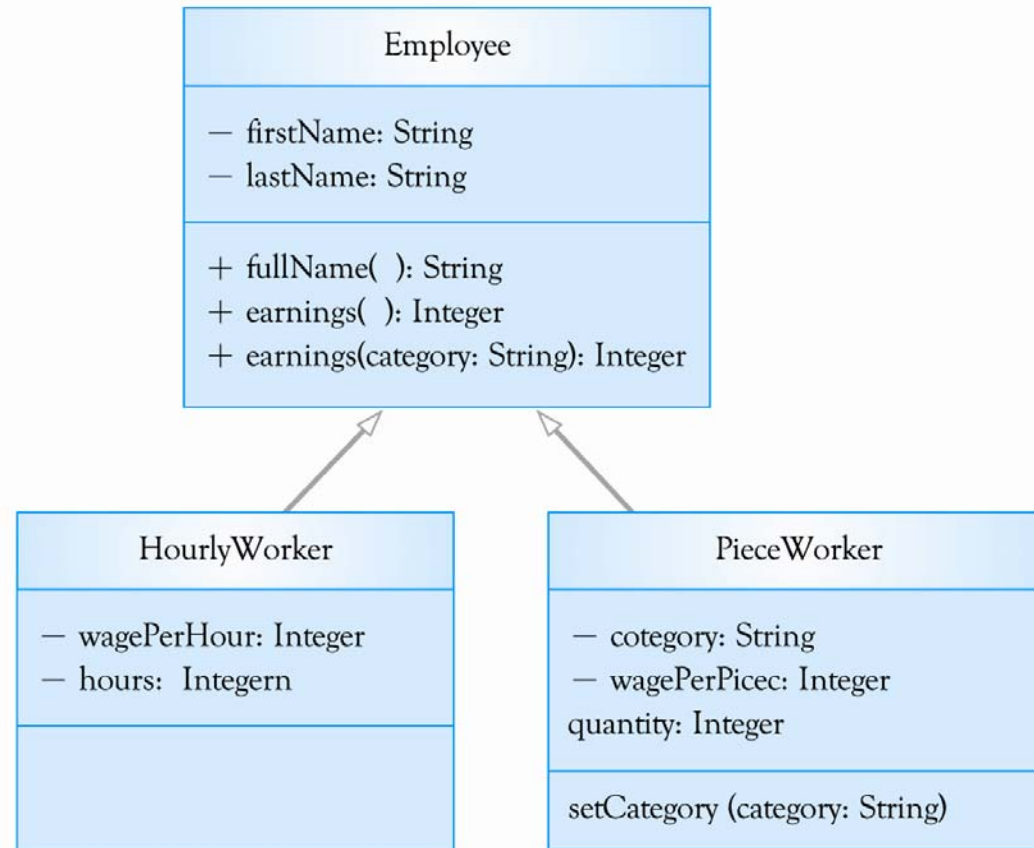
- 同名異式一般分為兩種不同的類型：超荷(Overload)與超載(Override)。
- 超荷：是在同一類別中，使用相同名稱的操作，然而每個操作的參數個數、參數資料型態不完全相同，當使用此操作時，是依參數個數及參數資料型態的不同來判斷要使用哪一個操作，超荷有時又稱靜態多型。
- 超載：超載的發生是來自於類別的繼承關係，一個子類別繼承父類別時，同時繼承父類別的屬性與方法(操作)。所謂超載，即是繼承父類別之後，在子類別中又再建立一個與父類別相同名稱的方法。
- 一般的物件導向程式語言在呼叫超載方法時，必須加上修飾子**this**或是**super**來區分呼叫的是子類別的方法，亦或是父類別的方法。例如，**this.method**是呼叫子類別的方法，**super.method**是呼叫父類別的方法。



3.3 物件導向的基本概念(12/13)

Figure

圖 3-3 類別間之繼承關係與多型



3.3 物件導向的基本概念(13/13)

- 程式設計上，運用多型或超荷之特性，可允許不同（或相同）的物件定義相同的操作名稱，待程式執行時再依動態連結 (Dynamic Binding) 的方法，判斷訊息之參數個數與參數型態來決定運作的物件或操作。
- 運用多型或超荷之目的是希望以相同的介面來處理不同的物件，或相同的物件但不同的操作方式以簡化系統發展的複雜性並增加彈性。

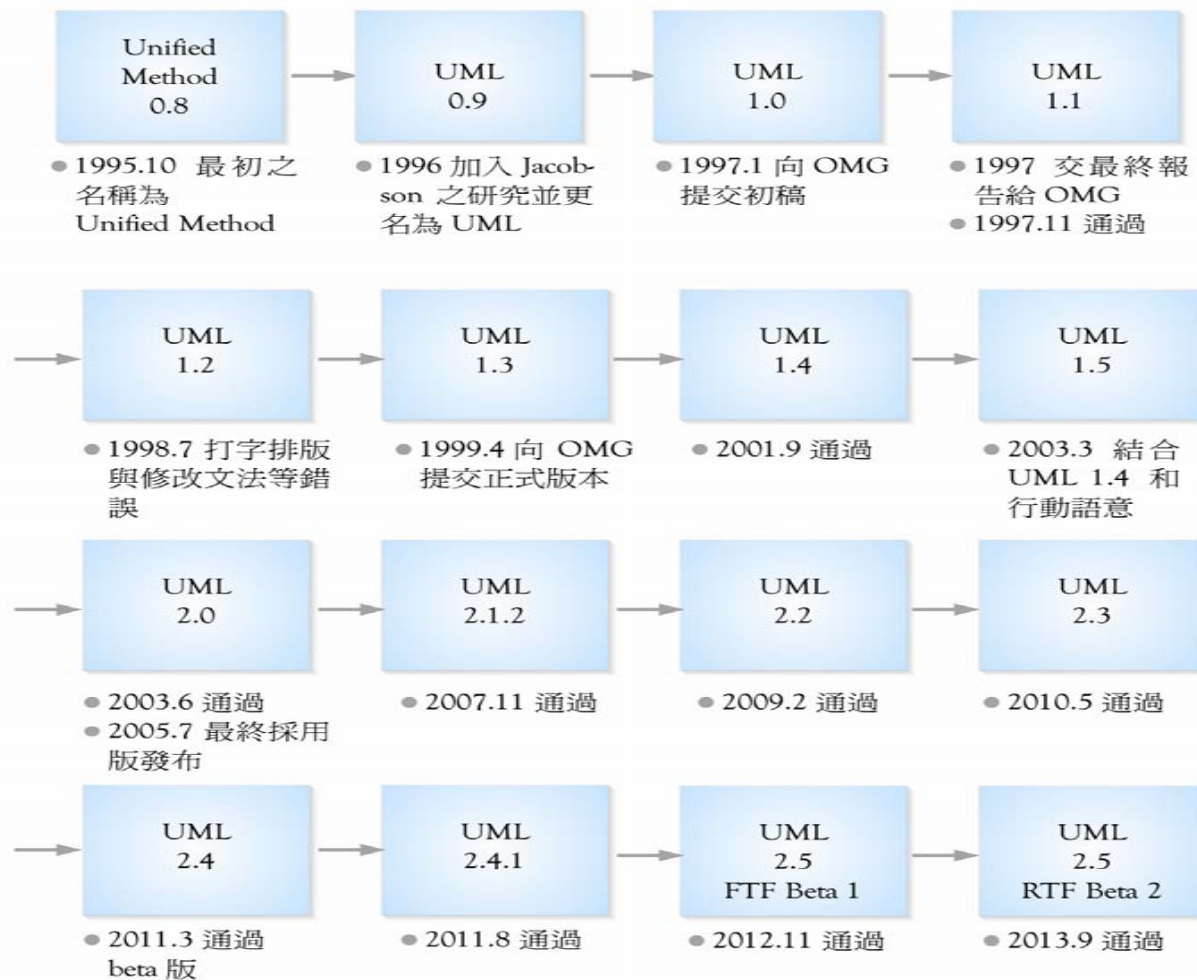


3.4 物件導向分析與設計及塑模工具 (1/21)

- ❑ 統一塑模語言 (Unified Modeling Language, UML) 為Rational 公司整合Booch、Rumbaugh 與Jacobson 三種方法而提出之物件導向塑模工具。起源於1995年10月，Booch 與 Rumbaugh 提供給OOPSLA 的統一方法(Unified Method)，當時的版本為Version 0.8。
- ❑ 1996年，Rational公司將Jacobson的研究加入統一方法，將版本更新為Version 0.9，並正式命名為統一塑模語言。
- ❑ UML 於2013年9月通過UML 2.5 RTF-Beta 2標準，其中強化重點在於整理規格的結構部份，以利更多不同的標記語言能以UML作為基礎設計。



圖3-4 UML之版本與年代演進

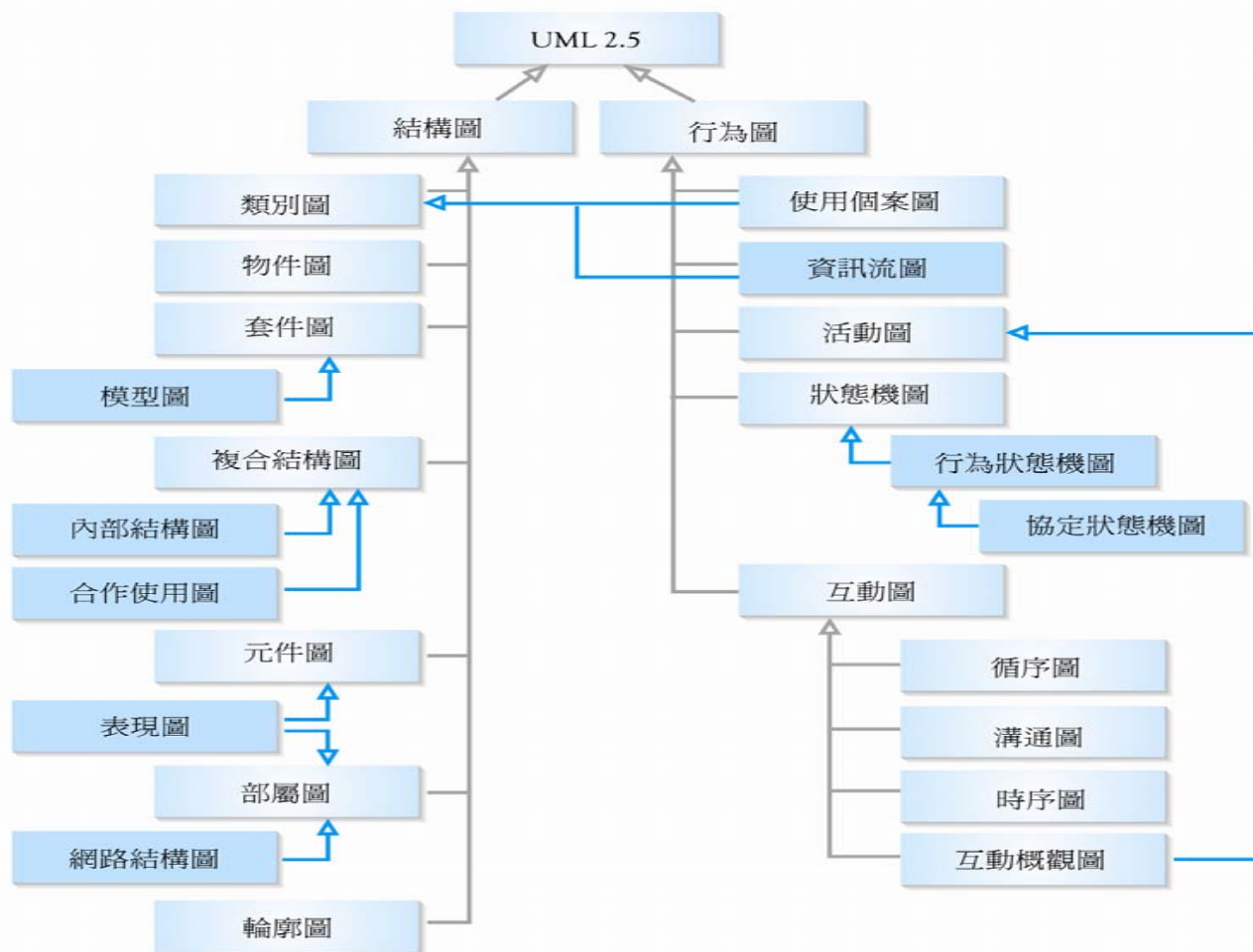


3.4 物件導向分析與設計及塑模工具 (2/21)

- 統一塑模語言是一種視覺化 (Visualizing) 、 文件化 (Documenting) 及 規格化 (Specifying) 的軟體塑模語言，共使用十四種模式圖進行塑模：
 - 使用個案圖
 - 類別圖
 - 物件圖
 - 循序圖
 - 溝通圖
 - 狀態圖
 - 活動圖
 - 元件圖
 - 部署圖
 - 套件圖
 - 時序圖
 - 互動概觀圖
 - 複合結構圖
 - 輪廓圖



圖3-5 UML 2.5 圖形架構圖 (參考uml-digrams.org)



3.4 物件導向分析與設計及塑模工具 (3/21)

1. 使用個案圖

- UML之使用個案圖 (Use Case Diagram) 為引用Jacobson方法中的使用個案模式，由使用者觀點描述系統的行為者與系統間之互動行為與關係。從內部觀點來看，使用個案可描述系統做什麼 (What)；從外部觀點來看，可描述行為者與系統如何互動 (How)。

2. 類別圖

- UML之類別圖 (Class Diagram) 為引用Booch與Rumbaugh 方法中的類別圖，以表示系統存在之物件型態 (或稱類別) 及各物件型態間之靜態資料結構與邏輯關係，也表達類別之屬性、操作與類別間連結之限制。



3.4 物件導向分析與設計及塑模工具 (4/21)

3. 物件圖

- UML之物件圖 (Object Diagram) 用以描述系統於某一時點之靜態資料結構，該圖由一群相關物件及其連結所組成，表示系統在某一時點之實例。

4. 循序圖

- UML之循序圖 (Sequence Diagram) 為結合Booch之互動圖與Rumbaugh 之訊息追蹤圖而成，主要描述系統運作時物件間的互動行為，著重以時間之先後順序為主軸，表達物件間訊息傳遞與處理程序。一個循序圖有一個與之對應的溝通圖，但表達重點與方式不同。



3.4 物件導向分析與設計及塑模工具 (5/21)

5. 溝通圖

- UML之溝通圖 (Communication Diagram) 即UML 1.x 版之合作圖 (Collaboration Diagram)，為與合作情形 (Collaboration) 之概念有所區別，UML 2.0版合作圖改名為溝通圖。由Booch 之物件互動圖與 Rumbaugh 之物件導向資料流程圖改進而成，主要描述系統運作時物件間的互動行為，但著重表達相關物件間之連結結構，並能同時展現物件間之訊息傳遞活動。



3.4 物件導向分析與設計及塑模工具 (6/21)

6.時序圖

- UML之時序圖 (Timing Diagram) 為互動圖的一種，著重物件間訊息傳遞的時間關係，用以顯示詳細的時間資訊，常用於即時 (Real-time) 或嵌入式系統之塑模上。時序圖藉由時間軸、條件與狀態等符號來表達物件在不同時點上的狀態改變。

7.互動概觀圖

- UML之互動概觀圖 (Interaction Overview Diagram)結合活動圖與互動圖來表達整體的控制流程，以活動圖為骨幹，說明主要活動與控制流 (Control Flow)，而細部活動則以互動圖 (包含循序圖、時序圖、溝通圖等) 來表示。



3.4 物件導向分析與設計及塑模工具 (7/21)

8. 狀態機圖

- UML 之狀態機圖 (State Machine Diagram) 以微觀物件為主，細分物件所發生的各項事件，並表達物件生命週期之狀態轉變及活動結果，從UML 2.4版開始，狀態機圖分成行為狀態機圖及協定狀態機圖，原本的狀態機圖由行為狀態機圖所取代。

9. 行為狀態機圖

- UML之行為狀態機圖(Behavioral State Machine Diagram)對狀態機圖做詳細的行為描述，主要用來表示系統在狀態轉換時的行為。



3.4 物件導向分析與設計及塑模工具 (8/21)

10協定狀態機圖

- 協定狀態機圖(Protocol State Machine Diagrams)是狀態機圖中的一種，協定狀態機圖是用來表達一種使用協定或某種分類器的生命週期，以表示協定溝通過程中狀態的改變以及事件的關聯。

11.活動圖

- UML之活動圖 (Activity Diagram) 用以表達使用個案中一系列事件的流程，即工作流程和所需之作業活動，及執行某一作業行為中之活動、轉換與條件等。一個活動可表示一個工作流程步驟或一個運算的執行動作，而一個活動圖則描述一群循序或同步的活動。



3.4 物件導向分析與設計及塑模工具 (9/21)

12.資訊流程圖

- 資訊流圖 (Information flow) 是 UML 行為圖 (Behavior Diagram) 之一，以抽象化方式表達系統中較高層級的資訊交換情形，資訊流 (Information flows) 能有效地描述整個系統內部的資訊流動情形，尤其是針對系統模型尚未完整規格化或描述較少的部分。

13.元件圖

- UML之元件圖 (Component Diagram) 起源於 Booch之模組圖，其中元件 (Component) 為系統之實體模組，是系統中可被替換的部分，適合提供一群介面做適當的實現 (或實作)。而元件圖則用以說明系統中各軟體元件間的配置結構與關係。



3.4 物件導向分析與設計及塑模工具 (10/21)

14. 部署圖

- UML之部署圖 (Deployment Diagram) 起源於Booch 之處理圖，說明系統各軟硬體（例如處理器、處理元件）於實際應用時之元件配置與關聯。例如硬體或網路基礎建設之拓樸，及同一處理器內執行處理的時程安排等。

15. 套件圖

- UML之套件圖 (Package Diagram) 用來表達系統中套件或元素的組織方式，描述套件與套件間之相依性 (Dependency)，可用於任一UML 模式。



3.4 物件導向分析與設計及塑模工具 (11/21)

16.輪廓圖

- UML之輪廓圖 (Profile Diagram) 能針對現存超模型(Meta-model) 中之超類別 (Meta-class) 來進行擴充(Extension)，使其適用於不同用途，如此便可隨不同平台（如J2EE、.NET等）或領域（如即時系統塑模或企業流程領域 (Business Process) 塑模)來調整UML 之超模型。

17. 模型圖

- UML之模型圖 (Model Diagram) 是用來顯示系統中抽象或特定觀點(Specific View)的結構圖，可以描述系統架構、邏輯與行為，以及顯示一個多層(Multilayered)的體系結構，為一多層應用模型。



3.4 物件導向分析與設計及塑模工具 (12/21)

18.表現圖

- UML之表現圖 (Manifestation Diagram)，係指介於元件圖與部署圖間的中介圖(Intermediate Diagram)，主要用於強化元件圖的產出與部署圖的元件之實作關係。

19.網路架構圖

- UML之網路架構圖 (Network Architecture Diagram) 為部署圖的一種，部署圖在節點上會產生產出，但網路架構圖不能顯示產出或部署，只用來顯示系統的邏輯或實體網路架構。



3.4 物件導向分析與設計及塑模工具 (13/21)

20.複合結構圖

- UML 之複合結構圖 (Composite Structure Diagram) 用來表達某個類別或元件在執行時期可能會產生的實例 (Instance) 與連結 (Link)，用以描述各物件如何在類別裡協同合作，或各物件如何達成目標。
- 複合結構圖可分為內部結構圖及合作使用圖，用以表達以下三種情形：類別之內部結構 (Internal Structure)、物件間之合作 (Collaboration) 關係、使用類別之方式。

21.內部結構圖

- UML 之內部結構圖 (Internal Structure Diagram) 用於表示分類 (Classifier) 的內部結構，並將分類解構成屬性 (Properties)、部份 (Parts) 及關係 (Relationships)。



3.4 物件導向分析與設計及塑模工具 (14/21)

22. 合作使用圖

- UML 之合作使用圖(Collaboration Use Diagram)用以描述物件之間在系統當中的互動關係(即合作行為)，以表達系統的行為，物件與物件間使用連結器 (Connector)連結，表示它們涉及完成一個任務或一組相關的任務。



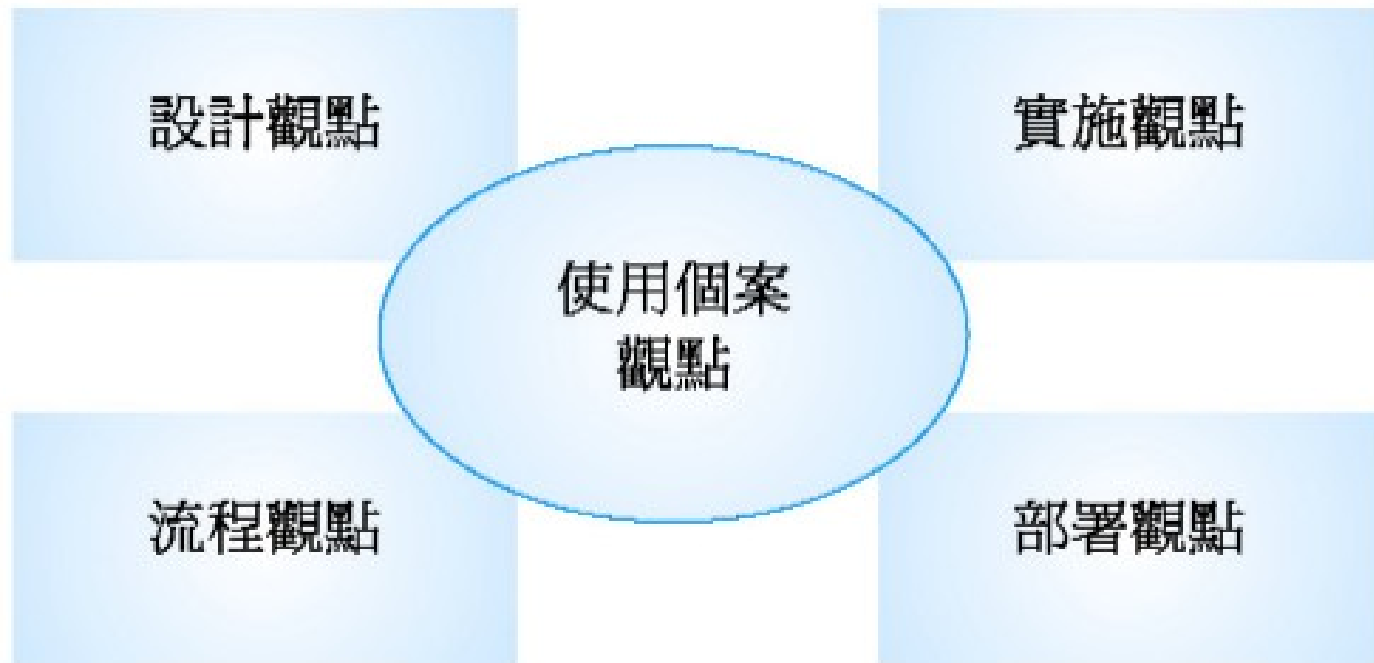
3.4 物件導向分析與設計及塑模工具 (15/21)

□ Booch et al, (1999) 由系統之概念面提出五個連鎖觀點的軟體系統結構 (參考圖3-6)：

1. 使用個案觀點 (Use Case View)
2. 設計觀點 (Design View)
3. 流程觀點 (Process View)
4. 實施觀點 (Implementation View)
5. 部署觀點 (Deployment View)



圖3-6 五個連鎖觀點的軟體系統結構



3.4 物件導向分析與設計及塑模工具 (16/21)

□ 使用個案觀點 (Use Case View)

- 由描述系統行為的使用個案組成，這些系統行為是以使用者、分析師、測試者的觀點來描述系統應做什麼及其操作情境，並不涉及實際系統的軟硬體架構。以使用個案觀點來描述高鐵站之自動售票系統為例，可說明系統如何支援使用者購票及購票之實際操作情境 (例如須先輸入購票資訊再行付款)。

□ 設計觀點 (Design View)

- 描述系統之類別、介面、合作情形及彼此間之互動方式。此觀點主要支援系統之功能面需求，意即著重表達系統應提供予使用者之服務。以設計觀點來描述高鐵站之自動售票系統，可說明此售票系統有選擇票種、目的地、車票張數、乘車日期時間與購票資訊等介面及車次資料等類別，並描述介面與類別間之互動及合作情形。



3.4 物件導向分析與設計及塑模工具 (13/21)

□ 流程觀點 (Process View)

- 用以描述系統內部的處理程序來說明系統之績效 (Performance)、產出 (Throughput) 與可擴充性 (Scalability)，而處理程序可能為平行或同步機制。以流程觀點描述高鐵站之自動售票系統，可說明乘客在購票過程中，系統讀取車次資料之處理程序；亦可說明若乘客選擇以信用卡付款，則系統與外部銀行連結之處理程序。

□ 實施觀點 (Implementation View)

- 表達系統版本的結構配置管理，描述系統由哪些獨立的元件與檔案組成，這些元件與檔案如何以不同方式組合可實際執行之系統。以高鐵站之自動售票系統為例，實施觀點可描述售票系統中所包含之元件，如計算車資之函式庫、列印車票之執行檔元件，並說明各元件於系統中之靜態結構關係。



3.4 物件導向分析與設計及塑模工具 (17/21)

□ 部署觀點 (Deployment View)

- 表達系統執行時，硬體拓樸上各節點 (Nodes) 之配置，並描述系統分析與設計的產出，及說明系統如何實作。以高鐵站之自動售票系統為例，此觀點可說明各站之自動售票機如何建置，例如售票軟體系統如何安裝至售票機、觸控式螢幕如何接收乘客輸入之購票資訊等。



3.4 物件導向分析與設計及塑模工具 (18/21)

表達工具	靜態面	動態面
使用個案觀點	➤ 使用個案圖 ➤ 活動圖	➤ 循序圖 ➤ 溝通圖 ➤ 時序圖 ➤ 互動概觀圖 ➤ 狀態圖 ➤ 活動圖
設計觀點	➤ 類別圖 ➤ 物件圖 ➤ 複合結構圖	
流程觀點	➤ 類別圖 ➤ 物件圖 ➤ 複合結構圖	
實施觀點	➤ 元件圖 ➤ 套件圖	
部署觀點	➤ 部署圖	



3.4 物件導向分析與設計及塑模工具 (19/21)

- 於系統實作面，上述五個觀點配合物件導向分析與設計（含需求分析、系統分析與設計）之活動，可具體地分為五種塑模：
 - 使用個案塑模 (需求塑模)
 - 物件結構塑模
 - 物件互動行為塑模
 - 使用者介面塑模
 - 系統元件與結構塑模



3.4 物件導向分析與設計及塑模工具 (20/21)

□ 使用個案塑模 (需求塑模)

- 主要應用使用個案圖、活動圖、藍圖與資料詞彙等工具進行使用者與企業需求塑模，此活動包含處理使用個案觀點之靜態與動態工作。

□ 物件互動行為塑模

- 主要塑模工具為循序圖、溝通圖、時序圖及互動概觀圖。

□ 使用者介面塑模

- 主要塑模工具為介面結構圖、介面藍圖、介面詞彙、行為狀態圖。



3.4 物件導向分析與設計及塑模工具 (21/21)

□ 物件結構塑模

- 主要塑模工具為類別圖、物件圖與複合結構圖，此活動包含處理設計觀點之靜態工作。

□ 系統元件與結構塑模

- 主要塑模工具為元件圖、部署圖，此活動可包含實施觀點與部署觀點之靜態工作。



圖3-7 物件導向塑模活動及塑模工具

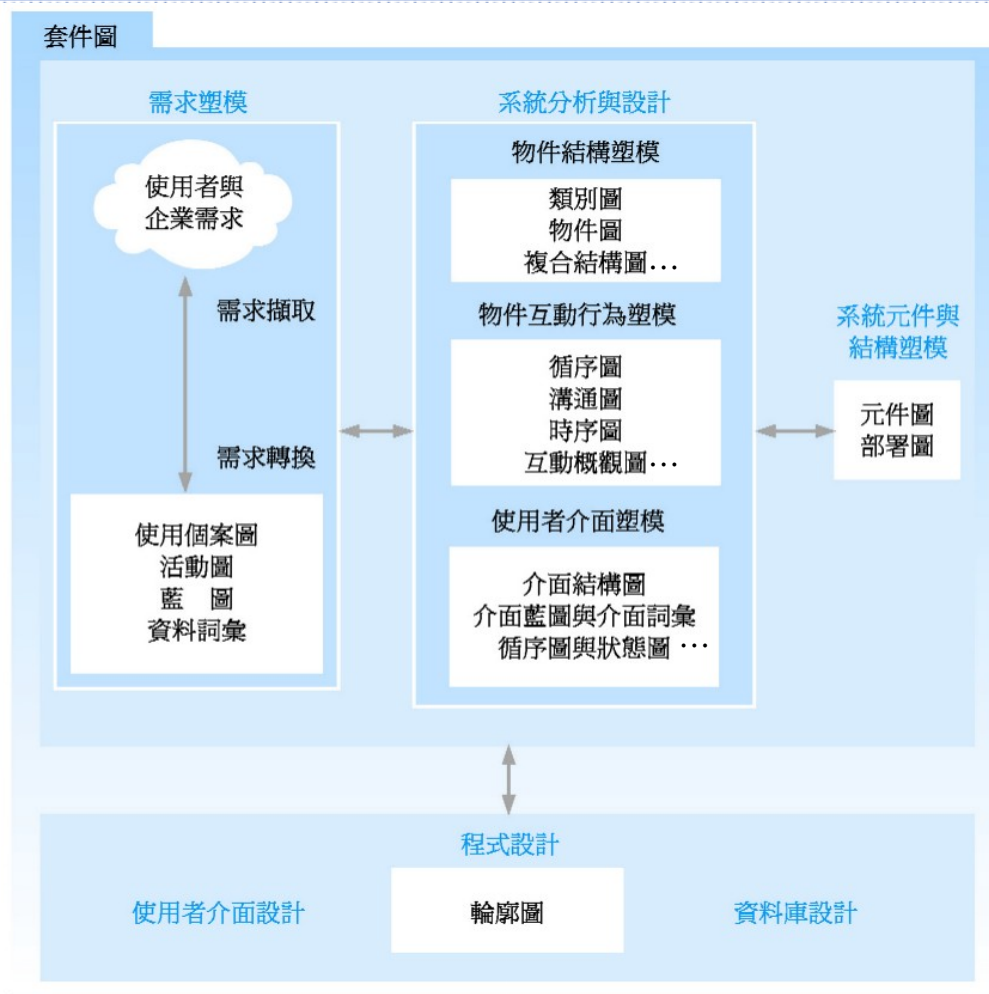
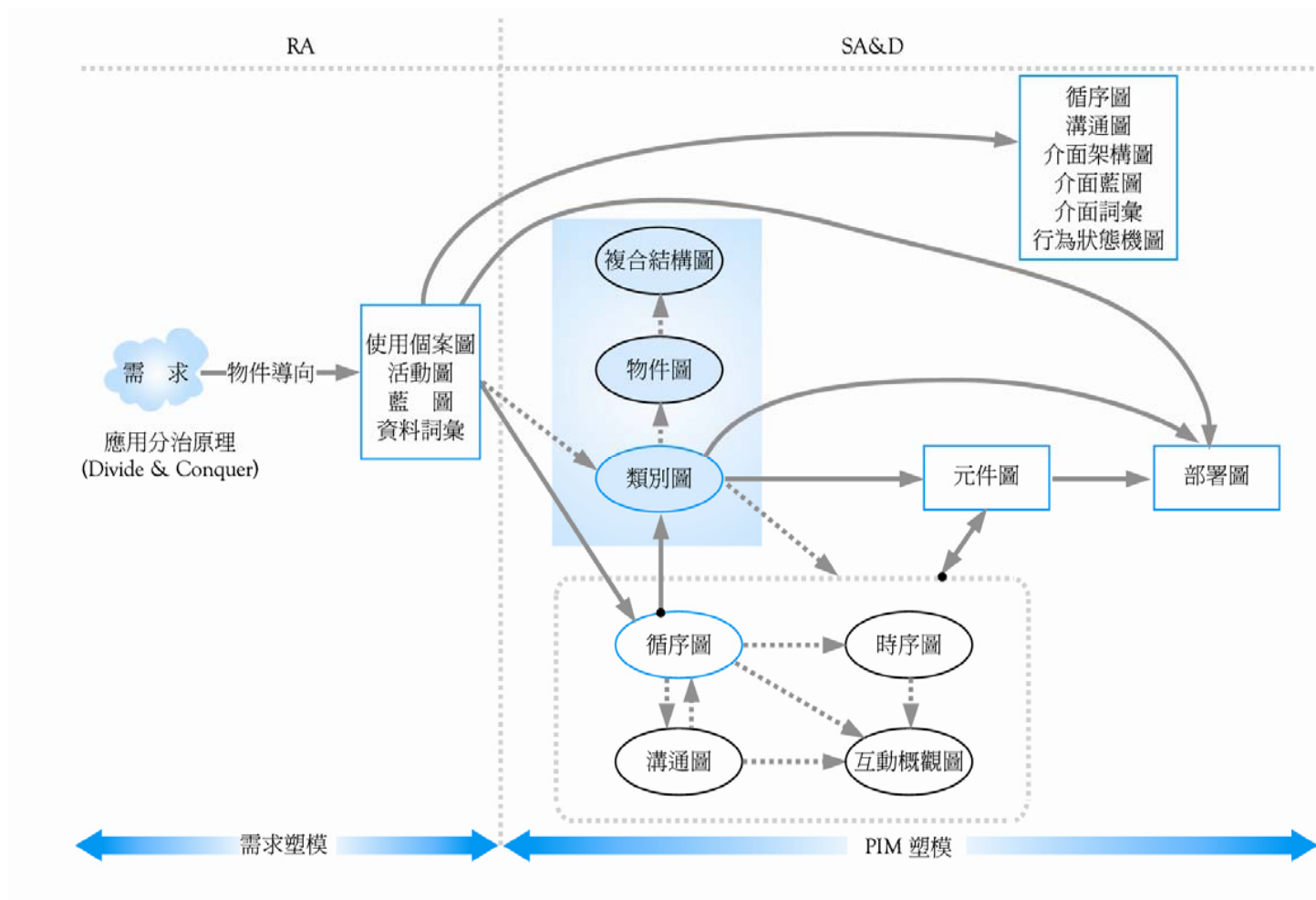


圖3-8 物件導向塑模活動及塑模工具



3.4 結論(1/2)

- ❑ 物件導向之基本概念包括物件、類別、封裝、繼承、同名異式與超荷，有別於結構化之系統，物件導向技術的引進，使軟體開發與維護更有效率，亦提升程式的重用與可維護性。
- ❑ 物件導向之系統開發過程是一道反覆的程序，包括需求分析、系統分析與設計、細部設計、程式編輯、測試與部署等階段。



3.4 結論(2/2)

- ❑ 於系統塑模的概念面，提供了使用個案、設計、流程、實施與部署五個連鎖觀點的軟體系統結構，供不同參與者，在不同的時間，從不同的角度來檢視系統。
- ❑ 在系統塑模的實作面，需求分析主要以使用個案圖、活動圖、藍圖、資料詞彙等作為表達工具；而系統分析與設計則主要以類別圖、物件圖、循序圖、溝通圖、狀態圖、活動圖、元件圖與部署圖等作為表達工具。

