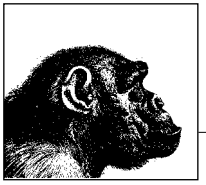


前言

關於本書

歡迎閱讀《UML 2.0 技術手冊》。UML 的全名是 Unified Modeling Language，中文翻譯為**統一塑模語言**。從 UML 誕生至今，UML 已經做了相當大的擴充，可以被用在許多領域，但它的根源還是在軟體開發領域。我們試圖盡量讓這本書能夠適合各類型的讀者，但是讀者至少還是必須懂一些物件導向程式設計（OOP），畢竟 UML 有許多術語來自 OOP。

在開始這本書的內容之前，我們希望先澄清本書對於 UML 的稱呼方式。就文法正確性來說，提到 UML 的時候，前面應該加上 the，也就是「the UML」，但是這聽起來又怪怪的，所以我們就用比較口語的稱呼方式，省略 the，直接稱呼「UML」。

《UML 2.0 技術手冊》從使用者的角度來介紹《the UML 2.0 Superstructure》。如果透過具體的範例有助於澄清某些概念時，我們就會使用 Java 程式當例子。

我們通常假定讀者懂 OOP，以及 OOP 相關的成分（類別、方法、繼承…等）。然而，我們並未假定讀者懂 UML。每一章都是利用「由上而下」的方式解說該章的主題，一開始的步調快而周詳，適合已經有基礎知識的讀者很快地學會 UML 的許多細節，後面的各節內容則會用比較和緩的步調詳細討論內容，包括了範例的展示，讓你瞭解如何將該主題運用在典型的問題中，幫助你更進一步地更新模型、消除歧異、抓住易於遺漏的細節、在運用工具開發的模型中增加資訊。

稍微警告一下：UML 對於**塑模**（modeling）有嚴格的術語，這是有必要的，可以盡可能地消除歧異、減少混淆。然而，每天我們運用 UML 在溝通時，常常會交替使用不同的稱呼方式，一個常見的例子就是，**操作**（operation）和**方法**（method），在軟體開發過程

中這兩個詞常常被視為是同義字而交替使用，但是在 UML 的定義中「操作」和「方法」其實指的是不同的概念。對於這種狀況，我們會使用 UML 正式定義的詞彙，而不使用不精確的詞彙。

如何使用這本書

這本書主要是根據 UML 圖的型態來進行內容的劃分，但是因為有些圖會依賴到別種圖的概念，所以有一些內容無法被很清楚地切割。第一章〈UML 基礎〉包含了 UML 的基本概念，並介紹一些背景資訊，幫助你瞭解後續章節的內容。如果你熟悉以前版本的 UML，你就可以略過本章。如果你對 UML 沒有足夠的背景知識，請務必從這一章開始讀。

接下來數章的內容是 UML 的**靜態塑模**（static modeling）。靜態塑模用來展現軟體的物質結構（假設軟體是可以觸摸的有形物體）。比方說，某個類別包含什麼操作和屬性，某類別**實踐**（implement）哪些**介面**（interface），以及某個**包裹**（package）包含哪些類別和介面。靜態塑模的章節包含了：

第二章〈類別圖〉

本章介紹**類別圖**（Class Diagram），討論類別圖中所能使用的多種元素（element）、如何繪製這些元素、如何擴充這些元素。因為類別圖常常是 UML 模型的中心，所以你應該要對這一章的內容滾瓜爛熟。本章最後會介紹類別圖在 UML 整體模型中所扮演的角色，以及類別圖通常如何被對應到程式碼中。

第三章〈包裹圖〉

本章介紹**包裹**（package）與 UML 模型內的**分組**（grouping）。

第四章〈合成結構〉

本章介紹**合成結構**，這是 UML 2.0 新加入的概念。合成結構特別設計來呈現**模式**（pattern），合成結構對於 UML 來說是一個重要的新元件。

第五章〈元件圖〉

本章介紹**元件**（component）以及**元件圖**（component diagram）。內容包括元件圖中所使用的**刻板**（stereotype）、元件之間的關係、元件之間的**中介資訊**（meta-information）。本章最後討論，在程式語言中一般是如何**實現**（realize）元件的。

第六章〈部署圖〉

本章介紹利用**部署圖**（deployment diagram）來表達系統部署的方式。本章解釋部署的基本概念，包括了**節點**（node）、節點刻板（node stereotype）、和元件的關係。本章也介紹如何利用部署圖塑造分散式系統的模型。

接下來的章節是 UML 的另外一半，也就是**行為塑模**（behavior modeling）。行為塑模用來表現執行時系統內的元素互動狀況。每種圖的作用不同，例如**使用案例圖**（use case diagram）可以呈現外部**演員**（actor）的觀點的需求，**循序圖**（sequence diagram）呈現物件之間的互動以實踐某個特定的使用案例。行為塑模的章節包含了：

第七章〈使用案例圖〉

本章介紹使用案例、演員、以及**系統邊界**（system boundary）。這章的內容有一點超出純 UML 的範圍，有討論到使用案例的常用技巧，包括使用案例**範圍劃分**（scoping）、使用案例文件，**使用案例實現**（use case realization）。

第八章〈狀態圖〉

本章介紹使用**狀態**（state）、**行為**（action）、以及**轉變**（transition）為**狀態機器**（state machine）進行模型塑造。**狀態圖**（statechart diagram）可以被用來為簡單的演算法塑造模型，一路到為複雜的系統塑造模型都沒問題。

第九章〈活動圖〉

本章介紹一個和狀態圖關係密切的圖，也就是**活動圖**（activity diagram）。活動圖類似傳統的**流程圖**（flowchart），通常用來為演算法塑造模型，或者為「使用案例實現」（use case realization）塑造模型。

第十章〈互動圖〉

本章介紹 UML 2.0 所支援的許多互動圖，其中大家最熟悉的是**循序圖**（sequence diagram）以及**合作圖**（collaboration diagram）。本章也會討論以時序為焦點的新互動圖。

本章最後一部分涵蓋 UML 2.0 的應用與擴充：

第十一章〈附標值、刻板、以及 UML 描述檔〉

本章介紹 UML 2.0 如何被擴充與調整。

第十二章〈有效的繪圖〉

本章和 UML 2.0 的規格完全沒關係，而是給你一些忠告，讓你知道真實世界中要如何塑模、何時該使用 UML 2.0 的哪個部分、何時以及如何有效地傳遞資訊。

附錄 A〈MDA：以模型驅動的架構〉

本附錄介紹**以模型驅動的架構**（Model-Driven Architecture，MDA）。MDA 不是一個新的想法，但是 UML 的設計中許多地方都受到 MDA 的影響，下一代的工具或許就能實現 MDA。

附錄 B〈OCL：物件規範語言〉

本附錄介紹**物件規範語言**（Object Constraint Language，OCL），這是一個用在 UML 圖中的簡單語言，用來表達某些限制條件。OCL 可以被採用在各式各樣的地方，本附錄只簡單地介紹它的基礎形式。

如果你熟悉 UML 的基本概念，你可以不必照順序讀這本書，可以直接跳著讀。然而，因為許多種元素都可能出現在不同的圖中，所以每一章的內容多少有一些重疊的地方。為了不要在每章重複某些資訊，我們在書中第一次出現的地方完整地定義了元素（以及它們相關的刻板、屬性…等）。後續章節如果有牽涉到此元件，如果有必要的話，則會說明交互參考到前面定義的地方。

字型慣例

由於本書保留了部份術語的原貌，為了避免讀者的混淆，我們運用了一些字型變化，讓讀者能輕易辨別詞彙本身的含意。

定寬（`constant width`）字型：

- 任何在程式碼中可能出現的元件，包括關鍵字、運算子、資料型態、函式名稱、變數名稱、類別名稱和界面名稱。
- 在 HTML 文件中出現的標籤（tag）。

斜體字型（Italic）：

- 用於可被替代的部份。例如：

  ```
  protocol://www.host.com:port/file#ref
  ```

  代表其中的 protocol、host、port、file、ref 必須代換成實際有意義的值。

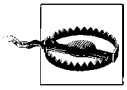
粗體定寬字（`constant bold`）

- 出現在螢幕上的字樣。為了區分手動鍵入的文字，以及程式自行產生的輸出訊息，前者會以粗體定寬字（constant bold）的樣式來表示，後者則是一般的定寬字型。
- 專有名詞。某些重要的術語在第一次出現時，會將其中文譯詞、英文全名、縮寫（不一定三者全部都有）用下列格式表示：

  檔案傳輸協定（File Transfer Protocol，FTP）

某些有縮寫的術語，在後文再次出現時，會視情況以縮寫出現，不一定會使用較長的中文譯詞。此外，若是在內文中列舉多個項目時，為了顯目起見，會以粗體字來表示。

表示訣竅、建議、或者一般性的註解。



表示你必須特別小心的地方。

請注意，UML 經常用到大括號（{}）以及**海鷗括號**（«»），當這些符號用在語法定義的時候，這是 UML 所要求的。

幾乎任何 UML 符號都是「可以從缺的」（optional），所以就不需要用某個特定的符號來表示「可以從缺的」這件事。如果有某個語法是「不可或缺的」，文中會特別說明。

線上資源

與本書相關的資訊，包括範例程式碼、相關資源連結、勘誤表，皆可在本書的中文網頁取得：

http://www.oreilly.com.tw/product_others.php?id=a185

批評和建議

我們永遠樂意聽到讀者對出版品的意見，包括如何讓本書可以更好的建議、指正本書的錯誤、或是讀者建議本書往後改版時，應該再加進來的其它主題。以下是本公司的聯絡資料：

美商歐萊禮股份有限公司台灣分公司

電話：（02）2709-9669　　傳真：（02）2703-8802

Web：http://www.oreilly.com.tw

電子郵件：

sales@oreilly.com.tw	（業務部）
edit@oreilly.com.tw	（編輯部）
bookquestion@oreilly.com.tw	（書籍內容的問題）

請以電子郵件的方式與我們聯絡，這會比電話和傳統郵件方便。有興趣為本公司翻譯書籍的高手，可與編輯部聯絡；如果您買到的書有印刷品質上的問題，可以寫信到業務部；若您對書籍內容有疑義，或是發現錯字，請寫信到 bookquestion@oreilly.com.tw 與我們聯絡，謝謝您！

感謝

Dan 的感謝

這本書是許多人努力的成果。如果沒有來自親友與同事的支援、電子郵件、評論、騷擾、建議，這本書就不可能完成。首先，我要謝謝我的編輯 Jonathan Gennick，謝謝他驚人的耐心。他是一個絕佳的工作伙伴，他讓這本書能朝著正確的方向持續前進。

接著，我要謝謝技術審閱者，他們的建議和評論未曾稍歇。有了他們所提供的豐富想法，我感覺這本書就像是第四版一般地精彩，而不是像第一版書籍的生澀。技術審閱者包括了：Stephen Mellor、Michael Chonoles、Mike Hudson、Bernie Thuman、Kimberley Hamilton、Russ Miles、以及 Julie Webster。

最後，我要謝謝我的家人：我的雙親從一開始就支持我，並成為我在專業和生活雙方面的典範；我的妻子 Tracey，在我寫作的同時，將一切事情處理得有條不紊。她所做到的這一切簡直是神奇透頂，相形之下，我寫書這件事簡直是一件易如反掌的小事。最後，但很重要，我也要謝謝我的兒子 Vinny，現在爹爹可以帶你去公園玩了！

Neil 的感謝

我要謝謝 Artifact-Software 的 Ron Wheeler 以及 Jacques Hamel 讓我們使用 XML 範例。也謝謝 Mindset Corporation 的 Derek McKee 讓我們使用 LamMDA 範例。最後，要特別感謝 Jonathan Gennick 對我們這麼有耐心