



UML 9張圖大公開！ PowerDesigner UML 建模簡介

David Dichmann, PowerDesigner 產品經理, Sybase, Inc.

在您準備為企業作出系統和軟體投資前，必須首先瞭解企業的實際需求，使用何種技術將幫助您的企業獲取更大的成功。您可以使用 UML，借助用例圖、循序圖、活動圖來進行分析。這些圖表將幫助您規劃系統的範圍、動態性能、以及表現方式等。您所需要捕捉的是使用者需求以及系統的行為，而無需考量程式碼實作。本文將幫助您深入瞭解 UML 強大的技術，並幫助您的企業創建出高效的傳統商務和電子商務系統。

PowerDesigner UML建模技術

PowerDesigner 9.0 支援UML1.3建模技術，即所謂物件導向分析與設計(OOAD)，所支援圖形包含用例圖(Use Case Diagram)、類別圖(Class Diagram)、循序圖(Sequence Diagram)、活動圖(Activity Diagram)和元件圖(Component Diagram)。而在最新的 9.5 版中，PowerDesigner 加強了對 UML 的支援，新增了物件圖(Object Diagram)、狀態圖(StateChart Diagram)、合作圖(Collaboration Diagram)與部署圖(Deployment Diagram)，透過UML建模技術，可將系統的分析過程與開發過程完整描述與建構。

用例圖(The Use Case Diagram)

UML Use Case圖提供了一個系統環境的建模方式。它能夠幫助您確定系統/應用程式的外部 and 內部元素以及系統範圍。因為是以圖形化方式呈現，與使用者溝通系統行為需求將更加便利，對於研發產品的開發團隊來說，更是有著舉足輕重的重要性。對於企業的非IT人員，或第一次接觸該軟體產品的用戶也有很大的幫助作用。用例圖能夠以視覺化的方式，表達系統如何滿足所收集的業務規則，以及特定的用戶需求等資訊。

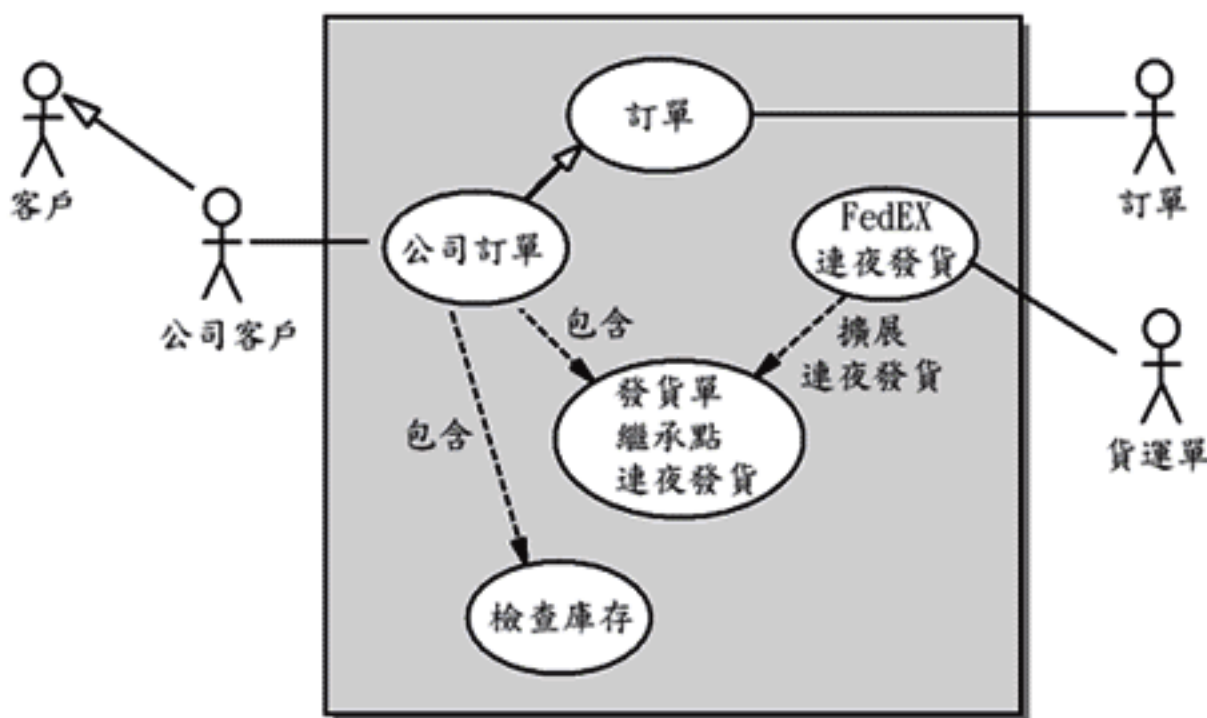


圖 1 - 公司客戶下訂單的用例圖

Use Case 圖的另一項功能表現在軟體開發後期，您可以透過Use Case圖中定義的需求來協助測試系統規格的相關功能，您不僅可以驗證系統功能是否無錯誤，還可以驗證系統運行時是否按照要求，執行了指定命令。這樣，您可以測試系統是否完全滿足了要求，以確保產品可以投入生產--也就是說，它已完全滿足了用戶的需求。只有確保滿足了合理、實用的各項需求，才能確保 IT 專案的成功。

循序圖(The Sequence Diagram)

您可以使用UML循序圖更詳細描述系統需求並對設計元素進行連結。循序圖敘述物件之間的互動關係。該互動關係在角色(role，與用例圖中的角色相同)和物件之間表示。透過循序圖，您可以按照系統特定方案中事件(消息)的精確順序來描述隨時間變化的系統行為。使用循序圖進行Use Case分析並引導設計：您可以決定將對Use Case圖所定義的管理任務負責的系統物件類型，並決定哪種物件將管理系統內外的"對話"或通信。由於消息已從序列圖中抽出，您可以描述類別和介面(我們最後要編譯和部署的程式碼)所需的某些關鍵操作(methods)。

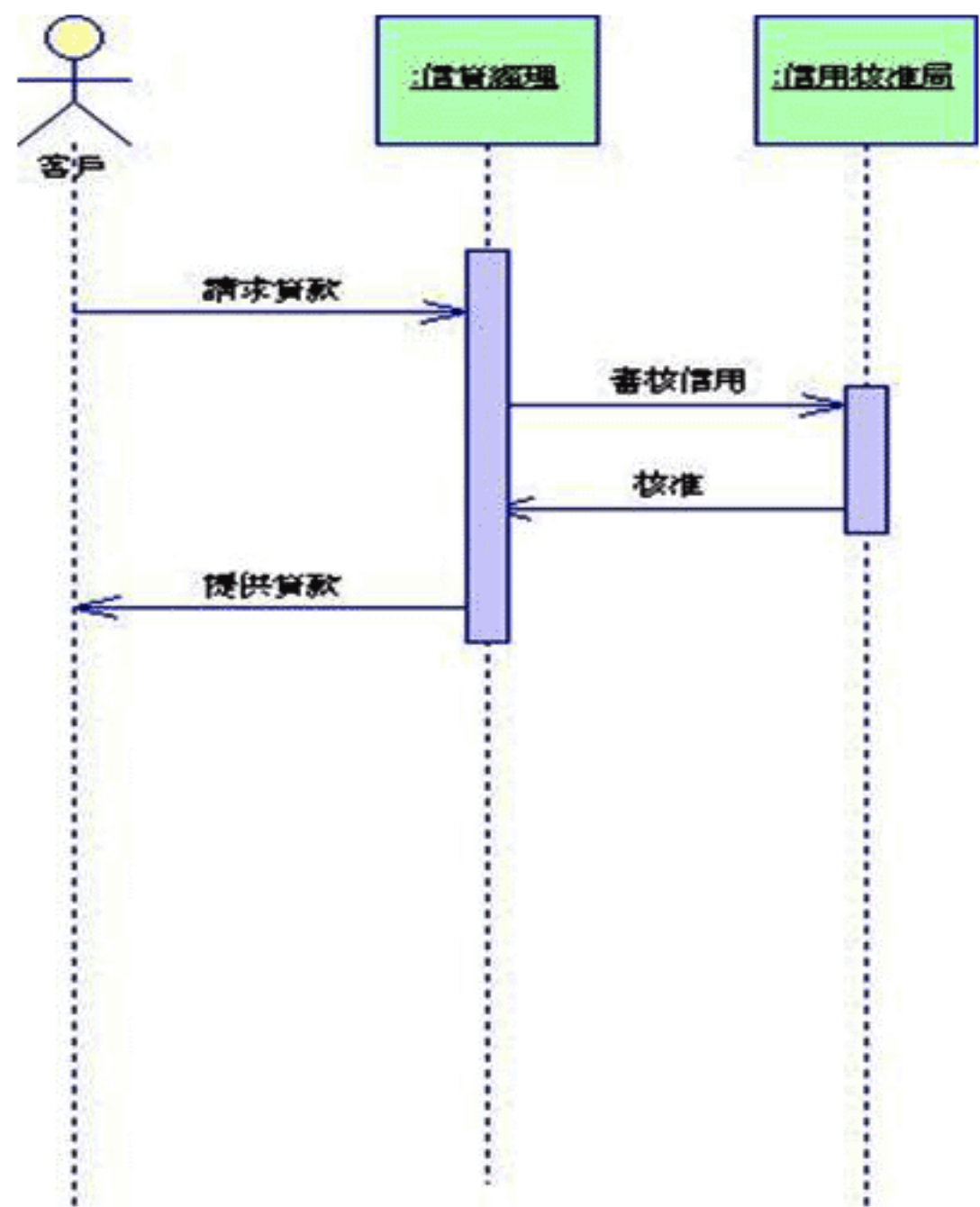


圖 2 - 貸款應用程式的序列圖





UML 9張圖大公開！ PowerDesigner UML 建模簡介

David Dichmann, PowerDesigner 產品經理, Sybase, Inc.

活動圖(The Activity Diagram)

UML活動圖設計用於幫助您瞭解系統中物件的動態變化。用於描述某一特定物件或多個物件如何協同工作。與循序圖有所不同，活動圖不是一系列與時間相關的通信，而是從一個任務到另一任務的控制轉移，同時指定誰（哪個物件）對發生的任務負責。

UML活動圖也是商業行為流程的技術視圖。可對商業流程進行分析或在“商業流程模組(Business Process Model)”完成後產生活動圖。

活動圖還可幫助構造系統內元素的詳細動態視圖（EJB如何互操作等）。

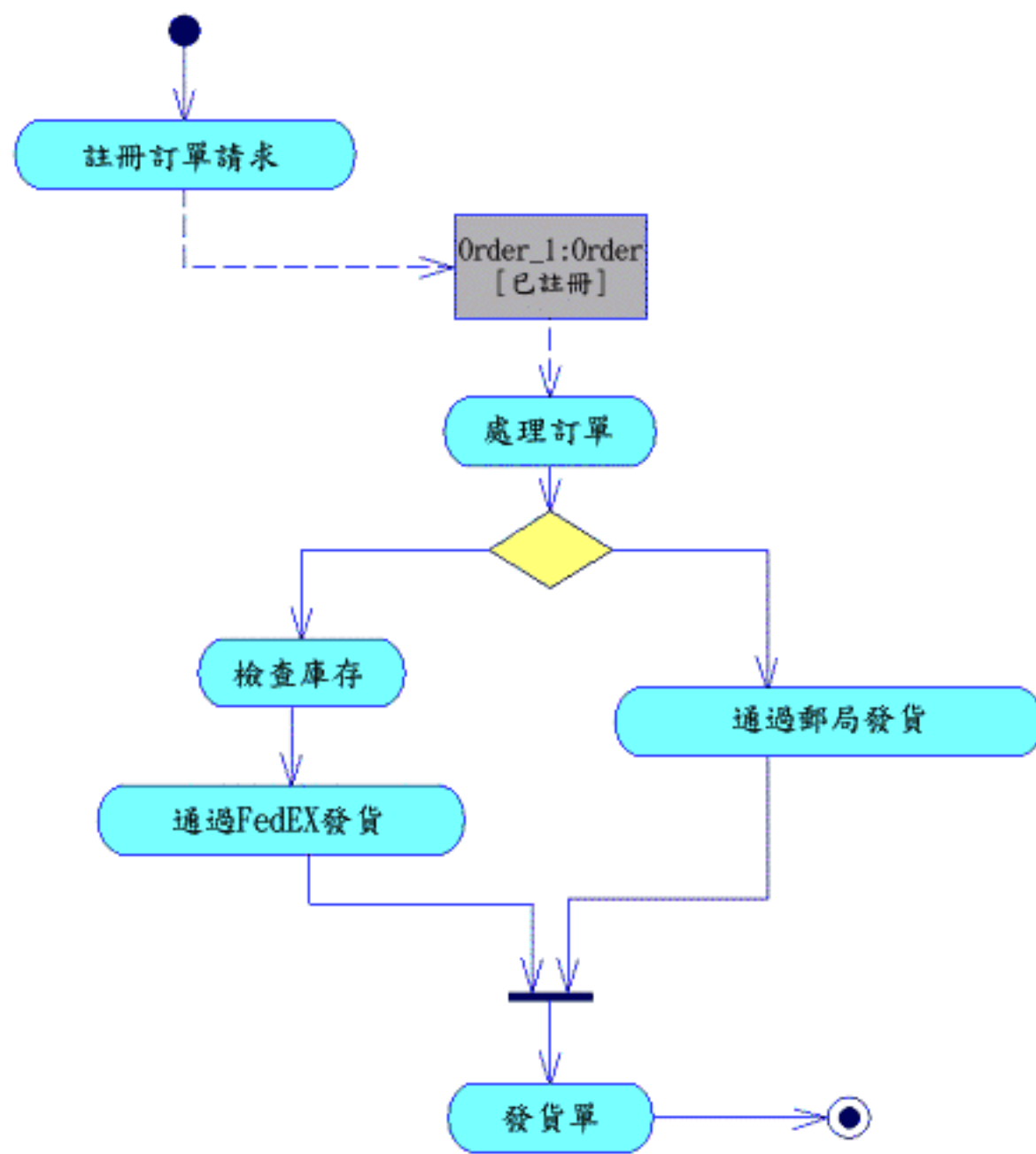


圖 3 - 活動圖 - 處理訂單

透過分析推動設計

透過分析模型可捕捉獨立於實作程式之外的系統意向和預期行為，與使用的語言、應用程式伺服器或任何的體系結構都沒有關係。但是，設計階段開始後，情形便不同了。您必須進入生產環境的細節並將軟體構建至特定的體系結構，設計是將系統加以實體化。

如果設計是由分析得到的，您可以更加確信所編寫的系統行為的正確性，確認所開發的成果將是一個按需求構建的系統。您將獲得高度成功--讓用戶得到所需要的系統。您還可以直接利用分析得出的資訊而無需在設計過程中重新生成，從而縮減開發時間，由於不必“重新複製”任何工作，因此減少了人為錯誤。

所以，到底透過分析，我們可獲得什麼呢？透過Use Case圖可以發現物件並進而建構類別(Class)和介面(Interface)。一或多個類別和介面可以實現一個角色，您可以在角色中直接定義類別和介面，還可以將角色連結到現有的類別和介面，因此在一個單一的元素中將包含所分析的多種元素。

透過循序圖之中物件的相互傳遞，可以進而找出物件的行為模式，如果您需要向某一類別發送消息(message)，可以直接以該物件的操作方式(operation)來顯示。循序圖中的消息可以用來自動建立操作或連結到現有操作。您可以通過連結跟蹤方法的功能，包括將哪些作為輸入內容和必須返回(return)哪些內容等等。

設計所包含的內容

您已經知道要構建的內容，現在您需要表達如何建構。您需要確定企業邏輯(Business Logic)所在的位置：可以置於應用程式伺服器上的EJB等元件中，也可以置於使用VB或PowerBuilder等語言、作為用戶端應用程式一部分的類別或元件中，或者做為觸發器和過程內置於關聯資料庫中。您需要根據系統需求(系統範圍、安全性、Performance)做抉擇。UML類圖和元件圖將用於定義詳細的技術系統靜態結構。

類別圖 (The Class Diagram)

透過UML 類別圖可定義100%的程式碼結構，包含企業邏輯以及物件中的程式結構。既然類別圖用來模擬開發中所維護的實際程式碼，顯然它是一種類似 Java 或 PowerBuilder 等物件語言的抽象性表示法。您還可以使用 UML 類別圖來概括 XML 中的複雜結構，令其更易於開發和理解。

透過PowerDesigner，UML 類別圖可以產生程式碼。還可以在開發過程中編輯該t程式碼以完善、測試和部署最終運行的應用程式。由於 PowerDesigner 在物件語言和 UML 類圖之間具有 1:1 的映射功能，您還可以實施反向工程代碼，讀取原始檔案並創建新的類別圖。您可以更深入地理解現有系統並簡化整合及維護工作。

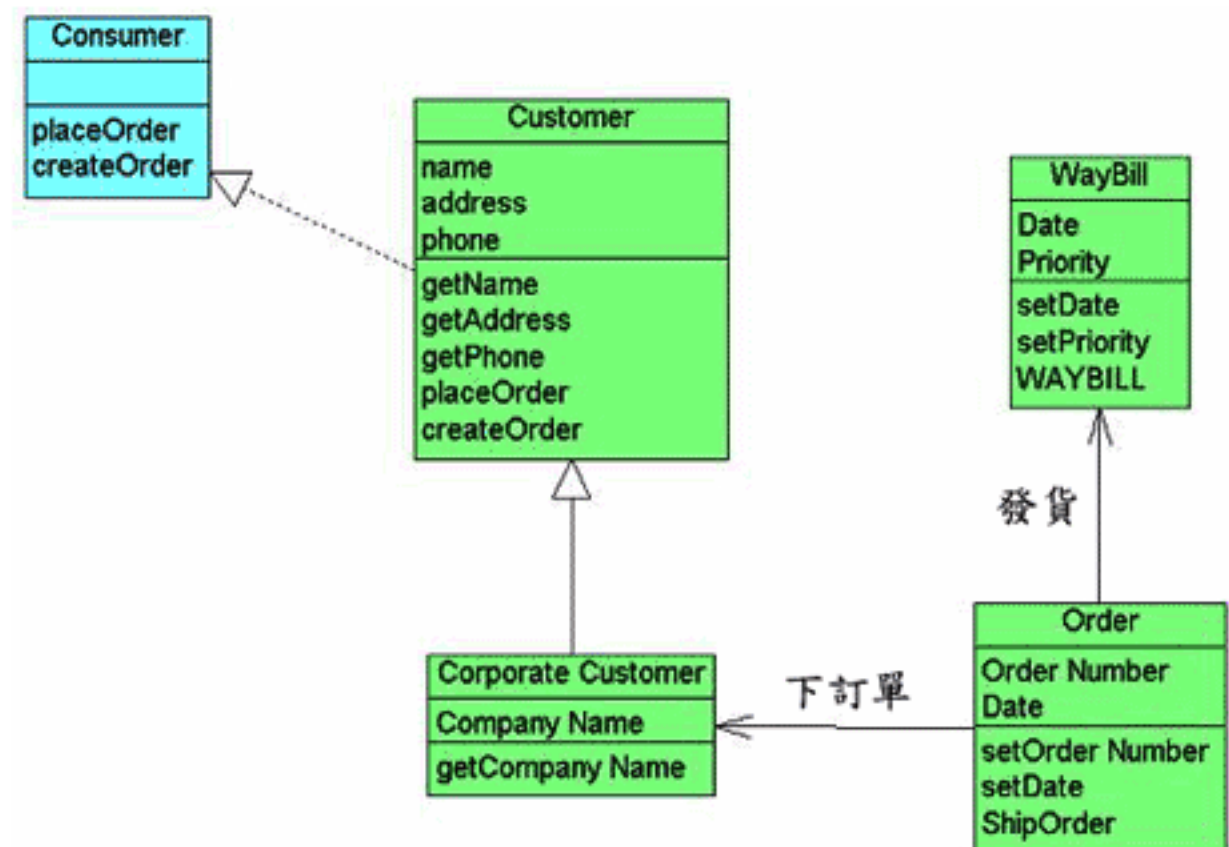


圖 4 - 訂單輸入系統的類別





UML 9張圖大公開！ PowerDesigner UML 建模簡介

David Dichmann, PowerDesigner 產品經理, Sybase, Inc.

元件圖(The Component Diagram)

UML的元件圖表示high-level的物件定義以及依賴性(dependency), 以一種黑盒子(black box)的觀點為出發。它是一個設計模型, 並且是程式碼的直接概念性表示法。例如, 一個 EJB 的元件標識直接連結到實作所必需的一系列類別和介面, 並將生成所需程式碼來推動最終這個EJB元件的開發。

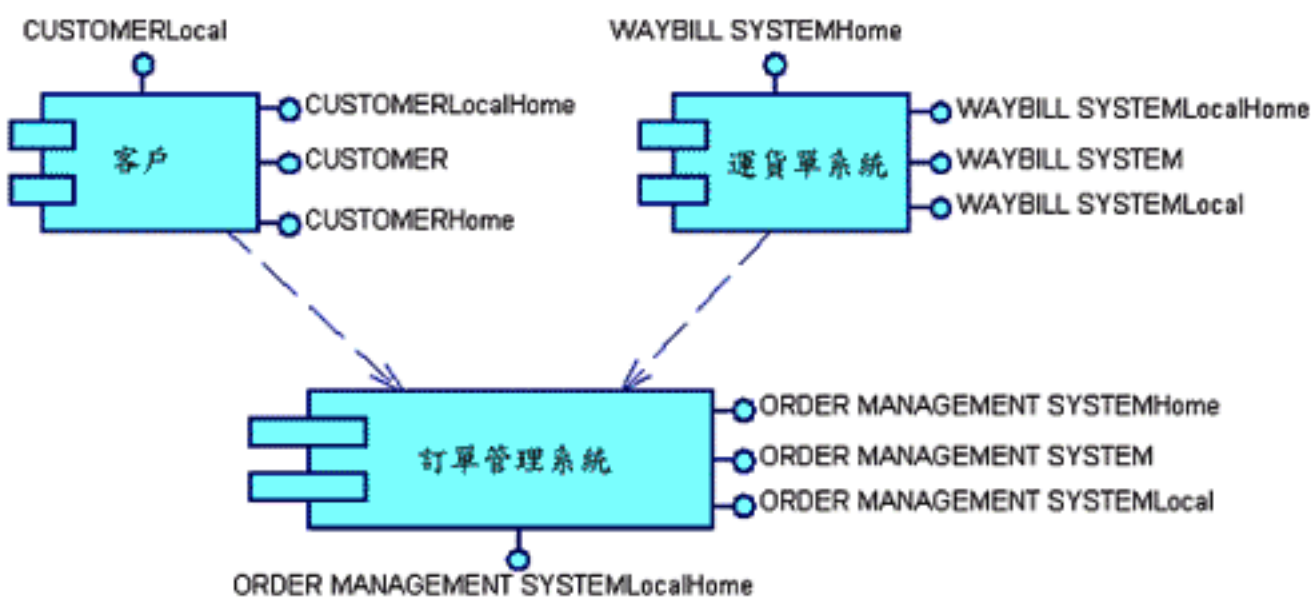


圖 5 - 訂單輸入系統的元件

元件圖比元件的程式碼更容易理解和管理。並且透過編寫元件介面的文檔來實現程式碼的共用(sharing)和重復使用(reuse), 使用者無需瞭解元件的實作內容即可在其他專案和系統中使用這些元件。

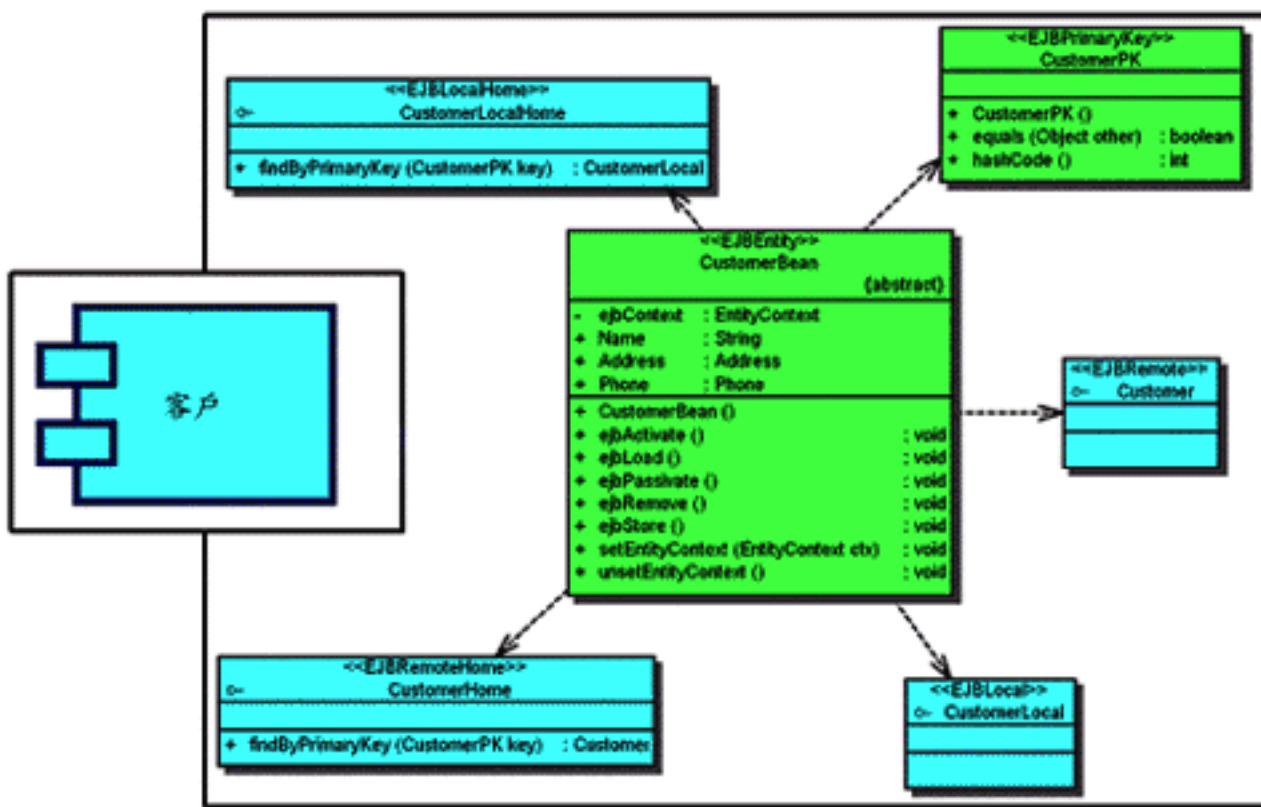


圖 6 - 客戶實體 bean 的 EJB 模型:圖

雙向互動式正反向工程

世界不是一成不變的, 您的 IT 系統也是如此。在您瞭解需求, 透過分析進行了設計, 並建構了系統的某些元素後, 必然還會遇到新的變化, 如要更新定義, 又或者現有用例圖中存在某些需要改正的錯誤, 程式碼在 IDE 和文本編

編輯器中被編輯以及資料庫被DBA 優化等。必須管理和掌握所有需要更改的細節, 以確保所建構的系統能夠與企業需求保持一致。

正反向工程的一個方案是當程式碼在開發過程中被更改時, 需要在類別圖中反映出來。具體細節如下:

1. 創建類別圖並將企業邏輯元素增添到模型中
2. 生成應用程式碼
3. 在 IDE 或文本編輯器中編輯程式碼
4. 編輯設計圖, 此時暫時忽略在程式碼中所作更改
5. 對編輯內容實施反向工程, 直到與現有類別圖一致
6. 將所設計的內容與開發時的編輯內容同步化(合併)
7. 產生新的程式碼, 該程式碼包含了設計以及研發人員所作的

當對類別圖進行了修改以反映新的設計內容時, 應該使用同步/合併技術防止遺漏開發人員的工作成果, 同時允許設計人員接受或拒絕開發過程中所做的更改。這樣, PowerDesigner 令 IT 能夠完全控制體系結構, 這正是制勝的關鍵。

PowerDesigner 的功能並不是僅限於此! 現在設計模型已被更新, 您可以將這些更改連結到分析中。有可能您在分析中發現了新的需求, 可以將這一更改反映到設計中並編寫程式碼。使用 PowerDesigner 中領先的 Compare/Merge 技術(在 September Blueprint 中討論過), 您可以在開發週期的所有模型和階段中獲得真正的正反向同步化。

上述探討了 5 種 UML 圖表: 用例圖(Use Case Diagram)、循序圖(Sequence Diagram)、活動圖(Activity Diagram)、類別圖(Class Diagram)和元件圖(Component Diagram), 它們可以幫助您掌握系統的需求, 設計其實體結構和預期的功能, 並轉換為程式碼。下面我們要接著介紹使用另外 4 個 UML 圖來進一步精簡前 5 個圖中包含的定義, 或者從完全不同的可取角度來定義系統。

這些圖表(物件圖、合作圖、狀態圖和配置圖)與前面的圖一起組成了PowerDesigner中的全部圖表, 並可在PowerDesigner9.5中使用。

物件圖(Object Diagram):

與類別圖一樣, 物件圖也是一個 UML 靜態結構圖; 它定義了系統在某個特定時刻具有的實體元素, 而無需考量系統的動態活動。它與程式碼有著一比一的對應關係, 但與類別圖不同, 物件圖僅顯示出實例(instance), 而非物件類別的內部定義。將物件圖描述為物件類別的實例圖可能最為合適。

物件圖的主要用途是進行分析。有一些類別圖無法表示的限制(constraint)可以在物件圖之中表示。此外, 在查看所管理的實例(instance)以釐清實例間的交互作用關係時, 物件圖還允許我們定義具體的"what if"情節(scenario)。



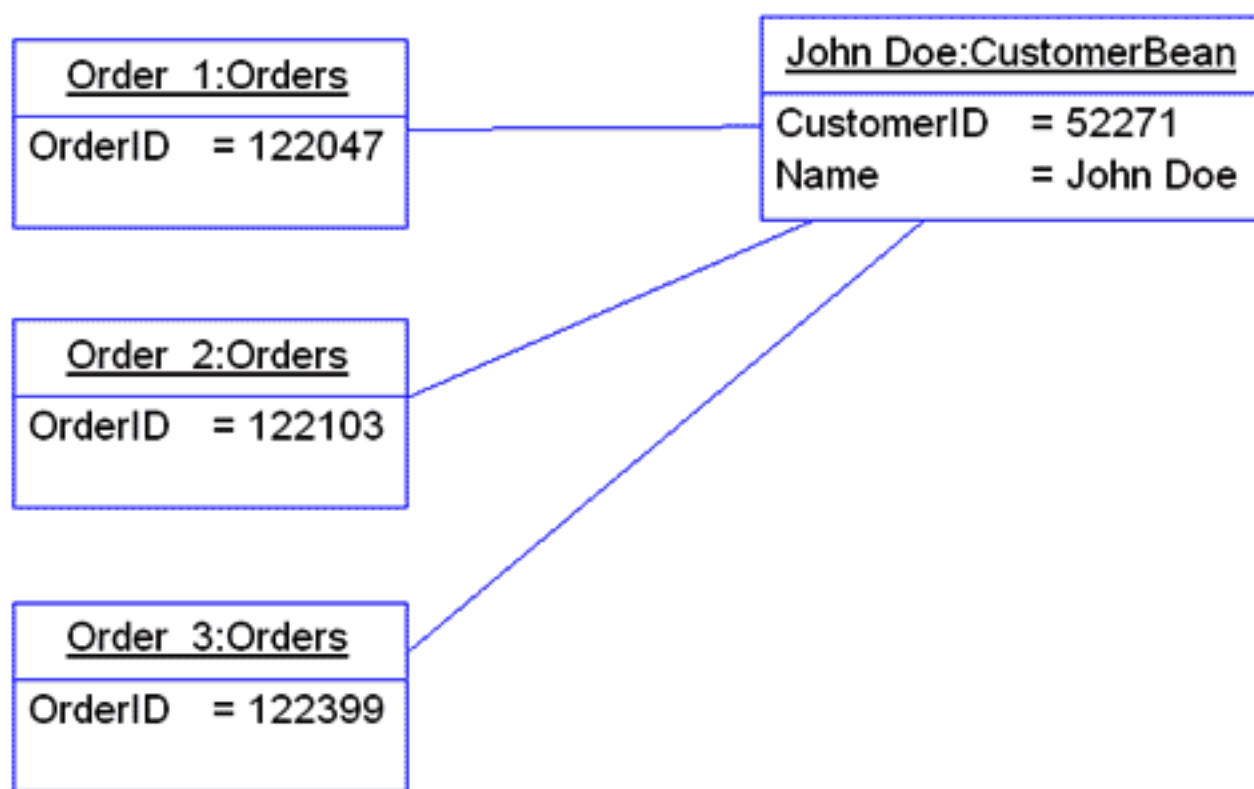


UML 9張圖大公開！ PowerDesigner UML 建模簡介

David Dichmann, PowerDesigner 產品經理, Sybase, Inc.

對於OO建模的初學者，以下有一些簡單的介紹：類別化(classifier)是抽象的物件結構定義。類別可以告訴我們所管理的是什麼類型的資料(屬性/成員表示資料元素)以及此類別具有的功能(操作/方法表示物件的行為)。實例即是一個類別化例子。假設定義一個名為 Customer 的類別，該類別具有 Name 屬性，而假設"Jane Doe"是一個由"Customer"這個類別產生出來的實例，代表一個客戶(customer)名為"Jane Doe"，實例通常具有比類別更豐富的含義，這是因為類別表示某種抽象的概述。收集某類別的若干個實例可能有助於您理解其用途並更容易地使用它。

因此，物件圖是類別圖的具體形式，表示類別實例樣本，並且顯示了鍵值和關係。例如，CustomerBean 類具有以下客戶實例：該客戶的 ID 為 52271，姓名為"John Doe"。該客戶實例與三個訂單實例(三份訂單)相關，訂單編號分別為122047、122103 和 122399。



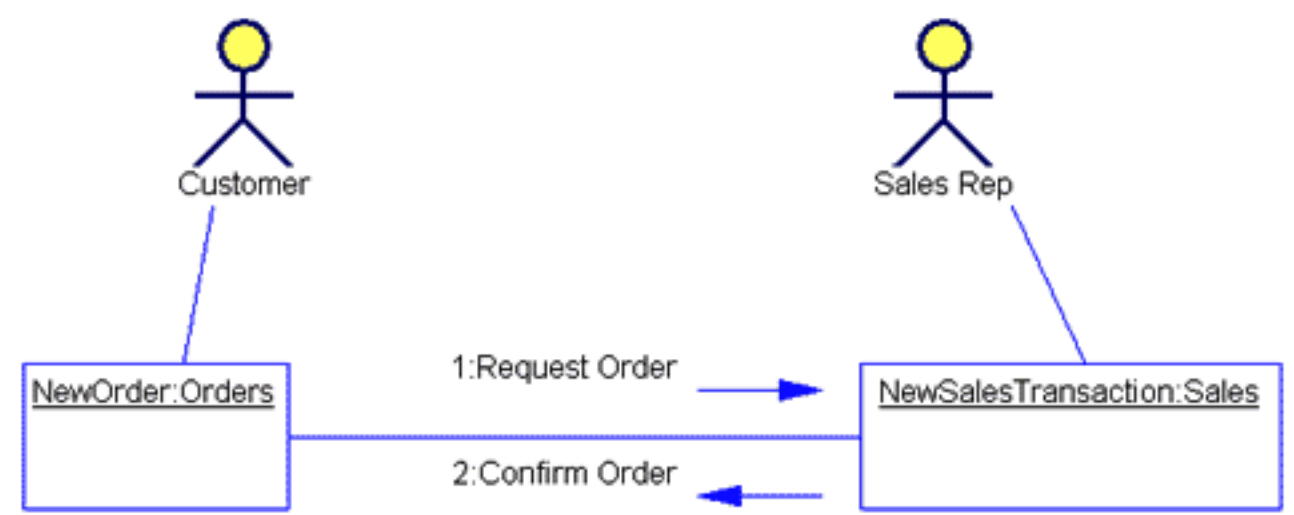
合作圖(Collaboration Diagram)：

合作圖和循序圖非常相似。實際上，循序圖和合作圖可以有效地交替使用，並可以簡便的相互轉換。其區別在於用戶閱讀和理解的方式不同。循序圖具有很好的層次性，並且圍繞時間建構，合作圖則主要是圍繞物件結構建構。通過在圖中對消息進行編號可以表示消息的順序。採用這種方式時，即使圖的結構不是基於時間的，也將保持定時關係。

合作圖借助於系統中元素或物件之間的交互作用，表示系統的動態方面，即在一段時間內的表現方式。它通過表示系統的靜態結構來對類別圖和物件圖進行補充，但不是借助於基於結構的關係，而是在系統物件之間傳遞交互作用"消息"。

建構合作圖時還可以在概念級測試靜態模型。在類圖中定義了類實例，這些類實例之間的交互作用定義了一個具體的使用方案以及將在這些元素之間發生的內部通訊。我們還可以使用其他角色來表示系統的外部作用者和內部使用者，如用例圖(Use Case Diagram)。

例如，我們可以建立一個訂單輸入系統，以供客戶和銷售代表使用。客戶通過建立一個新訂單與該系統交互作用。訂單物件與銷售物件之間進行對話，該對話由連結消息(message)表示，在此情況下，只有兩個消息：一個是來自 Orders 類的訂單請求，一個是來自 Sales 類的訂單確認。對一個連結上的消息數量沒有限制。我們在此討論的對話以一個訂單請求開始，然後是對該訂單的確認。

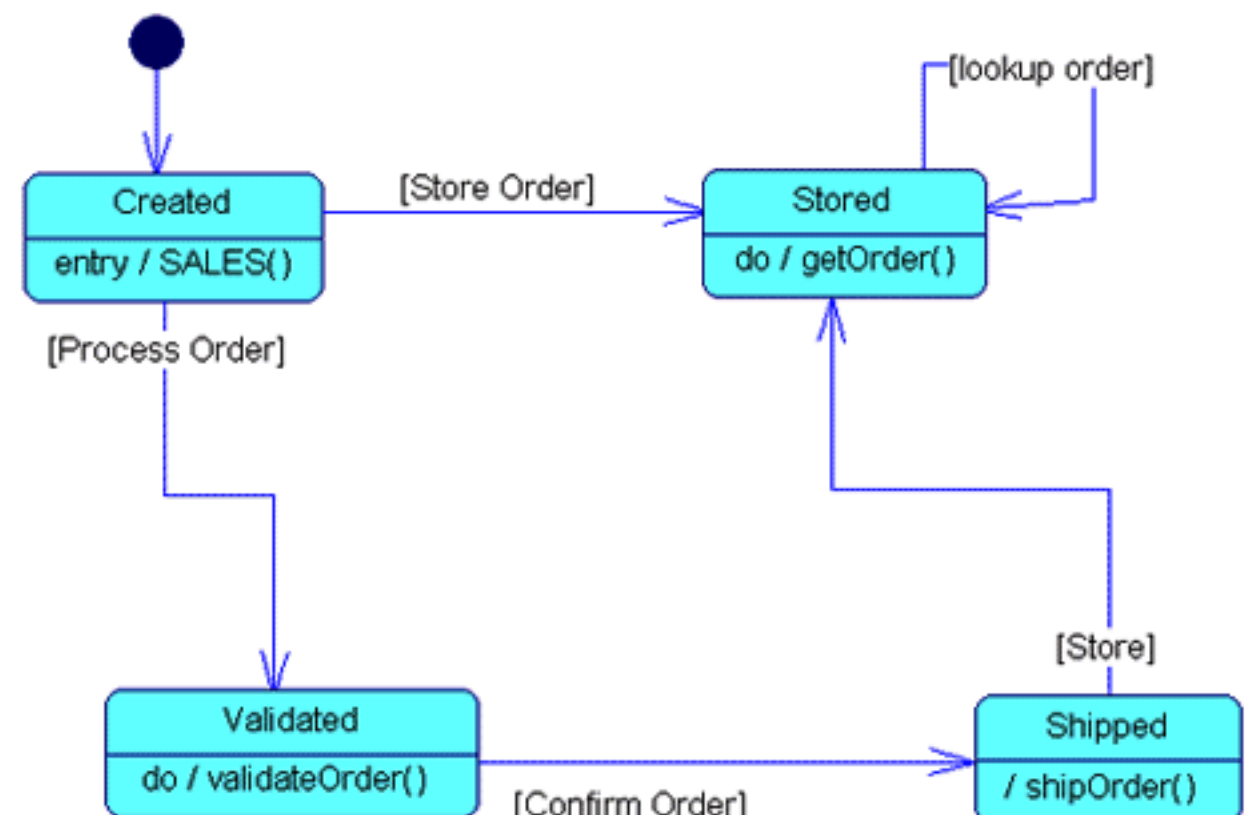


適用性

合作圖對於設計人員尤其重要，因為它闡明了物件的作用。您可以在循序圖之前建構合作圖(如果您計畫構造這兩個圖)，但通常是在完成類別圖之後構造合作圖以說明從類別中導出的物件之間的交互作用。可以使用一個或多個合作圖來實現一個用例(use case)，或者將複雜行為分割成多個邏輯子行為。

狀態圖(Statechart Diagram)：

狀態圖(State Diagram, 又名State Machine)描述了特定類別或元件在其整個生命週期中不斷變化時的行為。該圖顯示是什麼觸發了從一種狀態向另一種狀態的轉換，以及在該類別上呼叫哪些動作以提供該狀態的行為或觸發這種轉換。例如，訂單在被建立時處於初始狀態(initial state)。在客戶確認訂單正確後，訂單將進入確認狀態。在出貨以後，訂單需要從確認狀態(Validated)態進入出貨(Shipped)狀態。



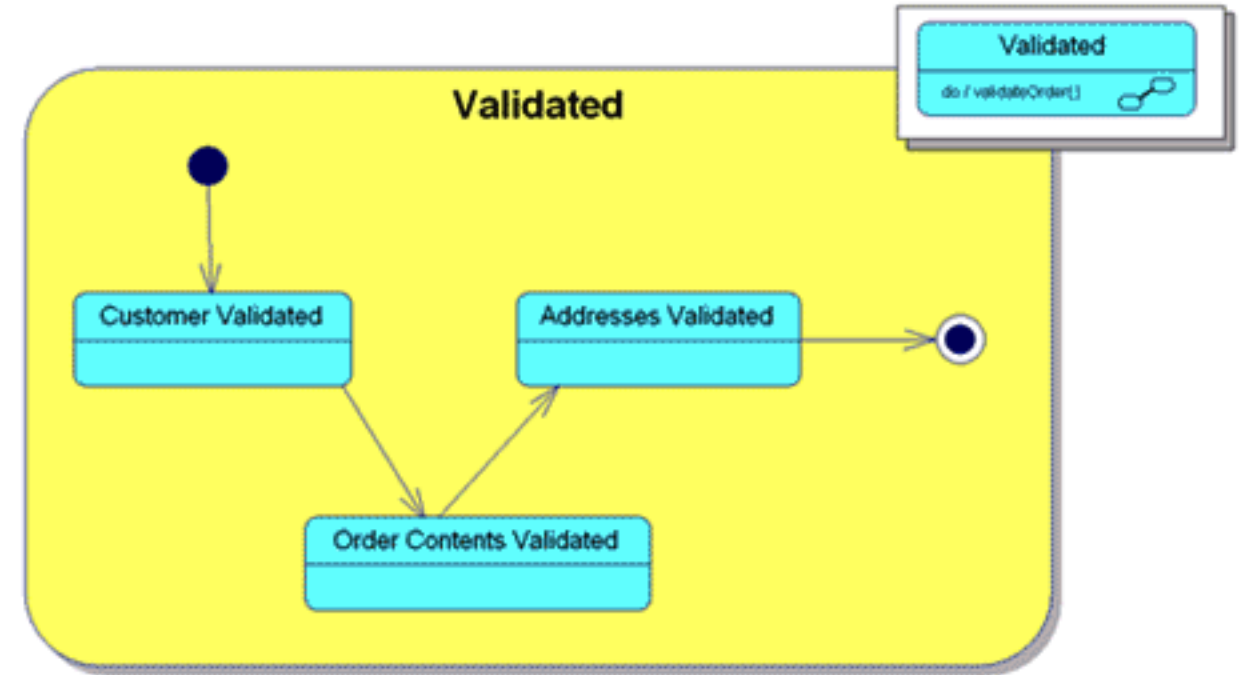


UML 9張圖大公開！ PowerDesigner UML 建模簡介

David Dichmann, PowerDesigner 產品經理, Sybase, Inc.

子狀態圖 (Sub-statechart Diagram)

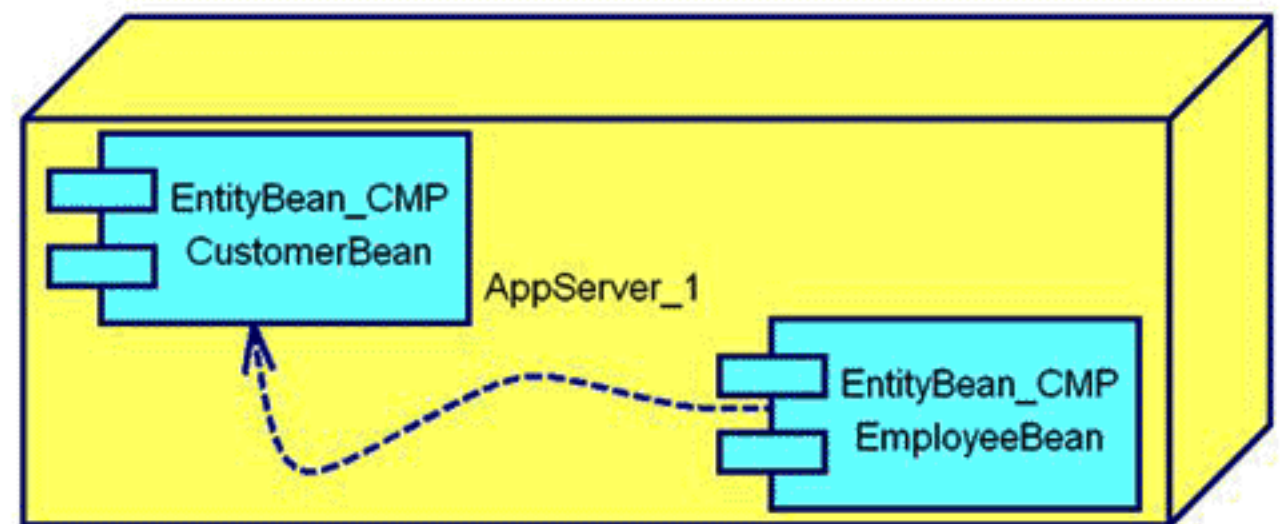
若要保持狀態圖簡單和易讀，您可能發現所定義的一個或多個狀態實際上涉及到更為複雜的行為，以至於它本身就可以定義為一個狀態圖。此時，與向主圖中添加大量複雜細節的做法相比，更好的做法是將這個單獨的狀態分解為多個子狀態，進而組成一個輔助圖，以定義父狀態的更為複雜的內部行為。



部署圖 (Deployment Diagram)

部署圖可以幫助我們確定所有代碼元素在伺服器、工作站和資料庫中的存放位置。有的節點需要依賴硬體或軟體框架來運行部分業務邏輯。這些節點交互作用以演示我們開發的多個電腦和系統是如何交互作用和整合的。節點中包含將部署到資料庫、應用程式或 Web 伺服器中的元件實例。部署圖用於將元件實際部署到伺服器中。通過定義希望元件運行的位置，我們可以快速的映射、部署和管理分佈在用戶端應用程式和應用程式伺服器端元件之間的業務邏輯或資料庫端伺服器邏輯。以下是要管理的物理體系結構的 1:1 模型。

例如，假定我們已決定實現兩個 Enterprise Java Beans，並且在應用程式伺服器上運行它們。下圖顯示了單個節點以及該節點內的兩個元件（每個 EJB 一個元件）。我們可以看出 EmployeeBean 依賴于同一應用程式伺服器內的 CustomerBean。



結論

在我們借助用例圖、循序圖、活動圖、類別圖和元件圖完成基本 UML 建模時，我們將需要其他一些工具來定義有關系統中某些特定元素的詳細資訊。我們可能希望在物件圖中使用精確的示例來表示物件的結構，或者借助於狀態圖來更詳細瞭解在其內部具有多個複雜狀態的類別的行為。我們需要使用合作圖從結構角度而不是從時間角度來考察系統元件之間的交互作用。最後，還需要使用部署圖來顯示所有系統元件在運行環境中的硬體或伺服器中所處的位置，從而更詳盡的瞭解分散式體系結構的使用方式。

UML 為我們提供了更加實用的圖表，以便完成對業務邏輯的技術分析、設計、開發、或部署。將這 9 種圖表與傳統的資料建模方法和新的業務流程建模方法相結合，將涵蓋從高階系統需求到技術和資料需求，以及程式實作面各個方面，我們可以全面瞭解推動軟體發展的所有因素。

