

使用 Rational RequisitePro 管理需求的新方法：使用架構方法來分析、管理及追蹤商業需求

Kumar Mani (kmani@us.ibm.com)，IBM 資深 IT 架構工程師

原文刊載於<http://www-128.ibm.com/developerworks/library/ar-reqframe1/index.html>

讓我們從 **Kumar Mani** 撰寫的系列文章中，探索擷取、管理及追蹤架構需求的新方法，這些方法以架構理論為基礎，適用於所有 IT 專案。如果您是 IT 架構工程師或專案管理者，需要面對企業或整合專案的複雜需求，便可使用這些方法來管理專案，並協助確保交付的時效。本文會探討使用 **IBM® Rational®** 工具組的實作方式，但使用其他產品也可以輕鬆採用此實作。

前言

本文會說明以需求工程為重點的新架構方法及技術，也就是在專案有效期中，擷取、分析和需求管理的技術。雖然本文使用 **Rational** 工具組來說明此技術，但並非 **Rational** 產品的指導教學，其目的是讓您使用基礎架構技術，並運用到您的專案上。當然，熟悉 **Rational** 的 IT 架構工程師，隨時可將這些技術運用到他們的專案上。

「需求」十分重要，因為這是開發架構的重點所在。圖 1 顯示「開放群組架構的架構開發方法」(TOGAF ADM) 及其八大階段。

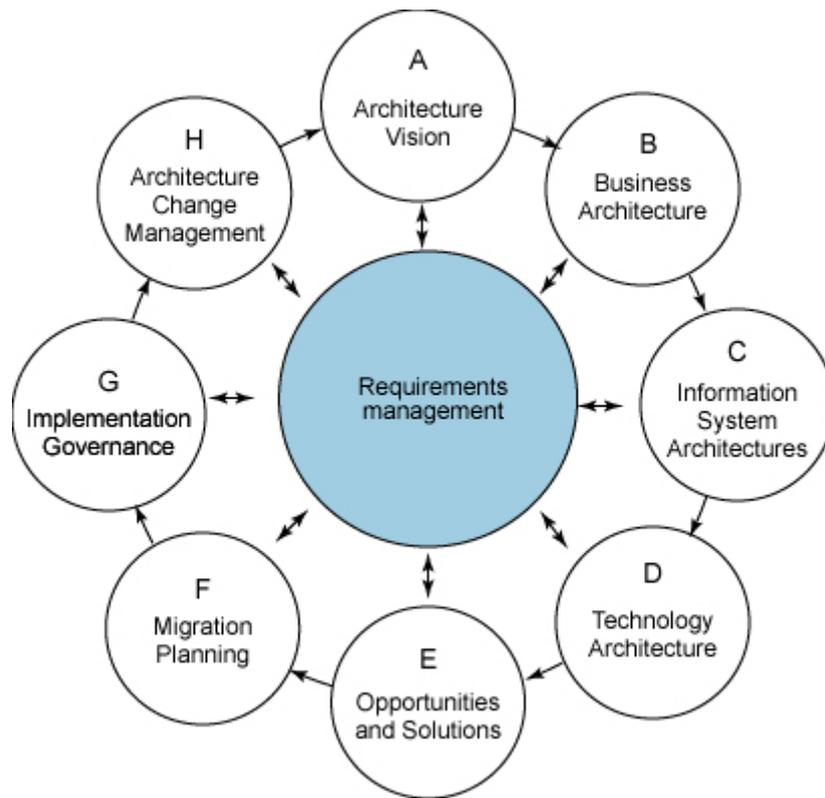


圖 1. TOGAF 架構開發方法

需求，意即需要、機會或不足所表現的形式，是整個流程其他階段的原動力。然而，基於許多不同原因，人們往往忽略或不夠重視專案中的需求管理。IT 架構工程師經常問：

- 「我一直是使用 Rational 產品來建立模型及開發，但要怎麼與需求結合呢？」
- 「我們的 XYZ 工具功能也不差，還有更好的選擇嗎？」

IT 專員都希望技術上的需求能夠清楚明瞭，才能著手編碼，但專案管理者及客戶並不清楚 Rational RequisitePro® 這類工具的好處，以及搭配專案管理工具（如 Rational Portfolio Manager）使用的方式，（部分專案使用所有 Rational 工具組，但不包括 RequisitePro！）這些都是本文探討的問題。

Rational 之類的工具，只要運用得宜，就能產生顯著的效益。部分專案管理者把工具（如 RequisitePro）當作需求儲存庫來使用，也說是說，將需求原原本本匯入工具，然後再從中輸出報告，結果專案管理者便會很納悶，因為看不到任何實質效益。

本文的目的，就是要彌補產品文獻及流程文獻之間的落差。Rational 工具文件能清楚說明其特色及功能，而流程文獻則提出大量方法及最佳實務。不過，IT 架構工程師仍難以發揮工具的最大效益。本文會介紹商業需求、使用案例和基本追蹤功能，以及測試期間的系統層次需求、元件化及可追蹤性。

Empire Systems Corporation

爲了說明如何善用架構需求技術，本系列文章會提到 **Empire Systems Corporation (ESC)** 公司的假設範例，這是一家虛構的個人電腦、筆記型電腦、電腦零件及相關硬體（如網路照相機和麥克風）製造商及供應商。ESC 已經有 Web 網站，許多應用程式也已成爲 Web 應用程式。現在，該公司領導者想要精簡流程、自動化商業功能、採用企業架構，以便提高收益及獲利，讓企業更上一層樓。

商業需求

成功的 IT 專案始於良好的商業需求，但技術文獻通常只著重需求工程的方法、技術及最佳實務，有關需求的討論，尤其是商業需求，卻含糊不清、雜亂無章，且通常難以理解。不夠明確會影響商業需求的詮釋，以及後續對映到技術需求的方式。接下來，讓我們簡單扼要重述需求工程的幾個基本原則。

1. 以商業爲重點

商業需求應著重在實際的商業需求，並且跟其他商業概念（如願景、使命、目標或目的）加以區隔。把商業目標（例如，「成功降低 20% 成本」）說成商業需求是常見的錯誤。

2. SMART

需求必須明確 (Specific)、可評估 (Measurable)、可執行 (Actionable)、實際 (Realistic) 且有時程 (Timebound)。這遠比想像中困難，然而，還是可以透過下列問題輕鬆達成：

- 爲什麼需要這樣做？
- 如果解決此問題，您會得到什麼成果或可以做什麼？
- 哪項流程特別會受到這個缺點影響？
- 改進流程所能達到的效益（儘可能量化）爲何？
- 有什麼作法可以改變現狀？

這問題通常很難回答，而且相關人員往往不知如何陳述問題。需求工程師可提出幾個假設狀況建議，以取得一些可能答案。不過，一旦相關人員進入狀況後，就要回到正題，這一點很重要。

- 有鑑於當時情勢及商業環境，何時（相關時程）應達成此目標？

這並不是專案交付時程的起始點，而是判斷商業環境的開始。例如，線上賀卡商家可能要鎖定假日（即其商業環境），來推出新的網站。

3. 找出並釐清範圍

內外部範圍爲何？要執行或交付的工作爲何？何時要完成？

4. 找出「主詞」及「動詞」

這一點最常忽略，而且是範圍逐漸擴大的主因。如果有一個以上的動詞，應考慮將

其列為不同需求，才能使目標明確，相關人員也能更充分瞭解。

5. 強調要做的事，而非方式

商業需求要強調所需執行的事情，且應該儘量避免提到執行方式。例如，企業可能需要以主觀、整合的角度，來檢視其商業資料。所以，重點應該是想做的事情，而不是需要得到的東西（如資料倉儲）。

需求範例

ESC 有數個 Web 應用程式。由於每個應用程式一直都是獨立發展，所以客戶會面臨一些問題，像是必須個別登入各應用程式，以及各項商業功能需使用不同的檢視畫面，使客戶怨聲載道。為了因應這些問題，ESC 業務團隊制定出下列需求：

BR264：客戶必須能透過一致的介面，存取所有 ESC 應用程式，且只需登入一次。所有客戶都會有相同的 ESCWeb（主要的商務網站）外觀與操作方式，減少使用者的錯誤，並且改善使用者體驗，而 ESCWeb 最後也能支援機動及遠端客戶。

可以明顯看出，這項需求有改善的空間。我們可以運用上述原則，重述此需求，如下所示。

BR264：在 ESC5.0，客戶只要登入一次即可存取所有 ESC 應用程式，包括 ESCWeb、ESCOrderStatus、ESCVendor 及 ESCSupport。

BR265：在 ESC5.0，所有 ESC 應用程式都會實施 ESC Web 標準 273-1 及 273-2，以降低 10% 的使用者錯誤。

方法一：在 RequisitePro 中指定商業需求

在 RequisitePro 中記錄需求的方法，關鍵是維持記錄及釐清需求所用的原則。此方法有三個步驟：

- A. 定義商業需求的 Requirement Type。所有商業需求都要有此屬性。BUS 是慣用的字首。
- B. 定義 Requirement Type 時，請使用 Requirement Must Contain 選項來指定定義符號（如 "|"）。主詞和動詞可分別置於定義符號兩邊。這並非強制規定，詳細資料請見 RequisitePro 手冊。目的是在找出需求的主詞和動詞時，審慎歸納出原則。
- C. 在最上層為商業需求建立一個套件，您很快就會看到這對可追蹤性的好處。

方法一即告完成。您已記錄明確又扼要的商業需求，並準備進入下一階段。圖 2 說明此方法。

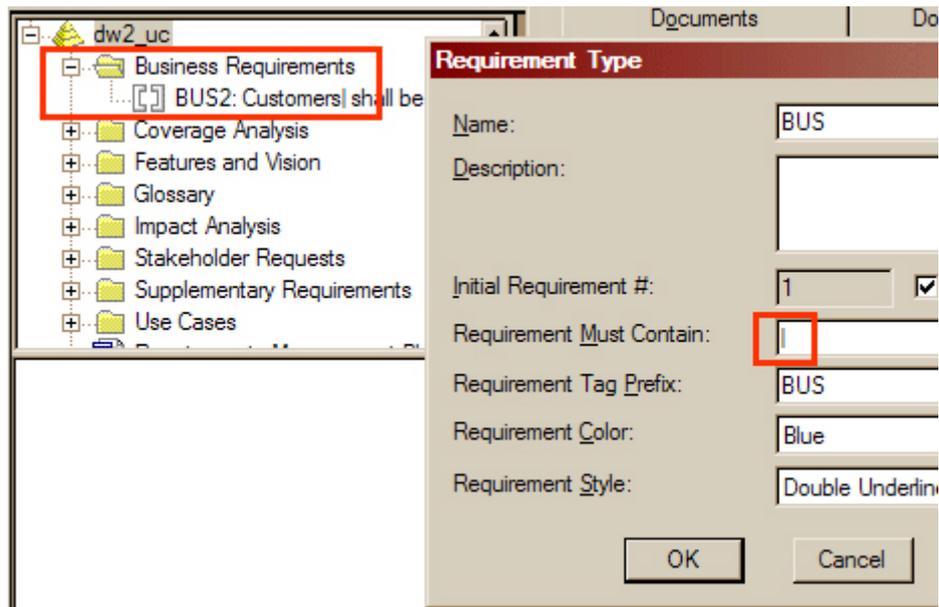


圖 2. 使用方法一擷取商業需求

使用案例

方法二（將使用案例連到需求）涵蓋專案的下一個階段。定義商業需求後，商業架構工程師或分析師即可開發使用案例來加以闡述。面對大型或複雜的專案，商業分析師特別容易碰到兩個問題：

- 需求置於文件中（除非使用「方法一」），而使用案例是在模型中，如 Rational Rose® Enterprise (Rose)。
- 隨著需求（及使用案例）的數量增加，這種不連貫會變得難以管理。

如何將商業需求連結到使用案例？使用案例會擷取使用者檢視畫面、指定使用者的動作及系統回應，但商業需求並非只是使用案例的來源，若指定為功能需求，就會影響到使用案例的內容；指定為特性或限制時，則會變為使用案例的屬性（或替代流程）。我們必須能記錄商業需求與使用案例之間的自然關係，並運用到有意義的地方，如追蹤或分析。

建立使用案例模型的相關資料非常多，「參考資源」中的文章是一個很好的入門。雖然討論使用案例並不屬於本文範疇，但我們仍會介紹用於方法實作的命名慣例。

使用案例的命名原則

為什麼使用案例的名稱很重要？請回顧上述用來定義需求的主詞與動詞結構，此處進一

步延伸這個概念。使用案例模型中的行動者代表主詞，而在我們的使用案例中，命名原則指定命名使用案例時，必須使用現在式或動名詞 (ing) 的動詞形式。有時候，加入動詞所修飾的受詞名稱也會有幫助。

方法二：將使用案例連結到需求

現在即可定義使用案例，並將其與需求連結。此方法的目的是確保使用案例的定義明確，並保持跟需求之間的關聯（追蹤），其步驟如下。

- A. 如果您的 RequisitePro 專案並不包含使用案例套件，或使用案例的需求類型，則請加以建立。
- B. 使用 Associate Model to RequisitePro 專案對話方塊，建立 Rose 模型與 RequisitePro 專案之間的關聯。確定將 Default Requirement Type 設為 Use Case。
- C. 現在便可開始在 Rose 中建立模型。首先，針對每個使用案例，建立 RequisitePro 需求類型 Use Case，其名稱需符合命名原則，如圖 3 所示。將此名稱輸入 Requirement Text。

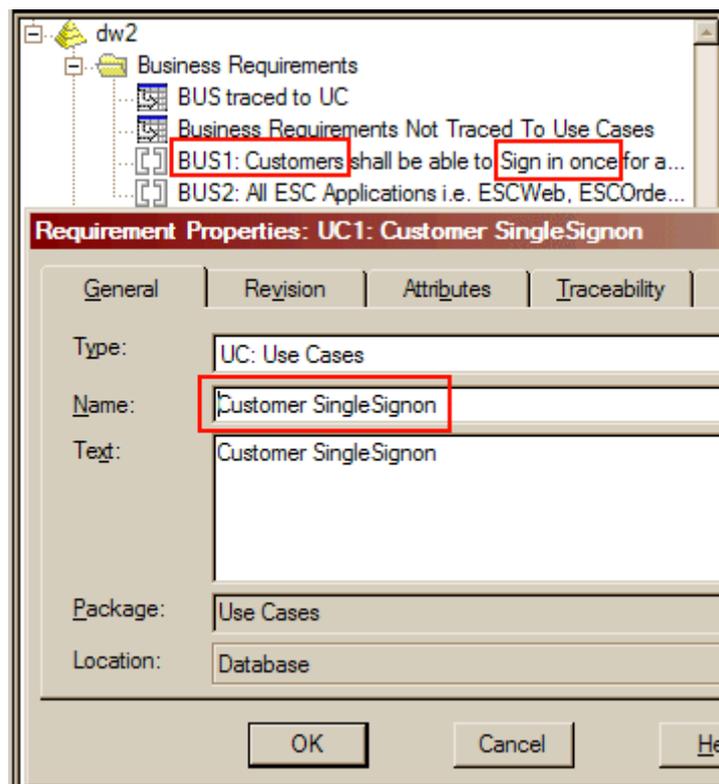


圖 3. 使用案例命名原則

- D. 在 Rose 中建立使用案例。用滑鼠右鍵按一下使用案例，開啓 Associate Requirement to Use Case 對話方塊。選取 Use Case in RequisitePro，然後按一下 OK。此時會啓動重要的 Resolve Use Case Name 對話方塊，此方塊會使用動詞將使用案例連結到需求，如圖 4 所示。

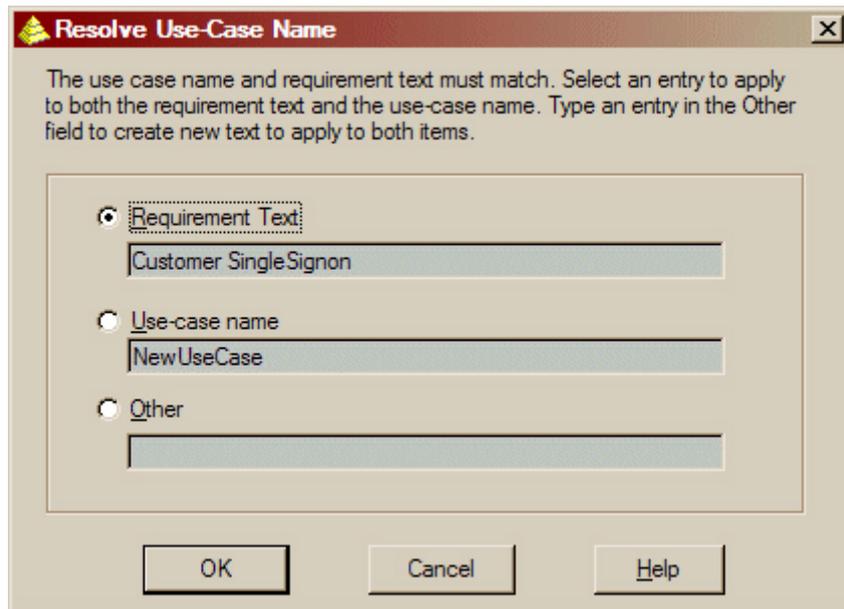


圖 4. 將使用案例連到結需求

E. 現在您會看到 **Requirement Properties** 對話方塊。選取 **Traceability** 標籤，然後將此使用案例追溯到 **BUS** 需求。

此方法即告完成。開始時先在 **RequisitePro** 中命名使用案例，在此步驟中您定義、詳盡闡述及完整追蹤 **Rose** 中的使用案例，也突顯出 **Rational** 的一大優勢：整合了需求、建模及設計。

可追蹤性

可追蹤性 (traceability) 為何如此重要？可追蹤性就是追蹤需求有效期的能力，包括原始的需求及各種不同變化形式，且能往前追蹤或往後回溯。隨著企業發展，支援企業的系統（及其需求）也會跟著改變。透過可追蹤性，就能將需求的過去、現在及未來緊密連結。可追蹤性報告會提供不同專案分析的資料，如成本、涵蓋範圍，以及可能的影響或作用。

但在實務層面，卻很難達成可追蹤性，主要原因在於，可追蹤性通常僅被視為需求工程的附屬。需求定義於文件中，而模型則建立在 **Rose**。建模工作完成後，可追蹤性通常會以手動方式記錄在試算表上，不過，試算表的維護工作十分麻煩，而且容易造成疏失。但對於專案最大的影響，是因為一般會將試算表當作可追蹤性報告，因而疏於檢查。

下一個方法可解決此問題，其中第一部分是在建立需求後，歸納出追蹤需求的原則（在上述方法中，我們已透過使用案例加以說明），目的是在整個需求有效期及進入測試階段內，維持一致的原則。第二部分是產生涵蓋範圍及 **Dangler** 報告。**Dangler** 報告是第一階段的影響分析。重點在於，在審閱期間，這些自動產生的報告均需要仔細檢查。

顧名思義，涵蓋範圍是確保某個層級的所有需求，在下一個層級都會含括在內（或進一

步指定)。比方說，測試案例必須含括每個使用案例。**Angler** 是無意中漸漸形成的需求，在上一個層級並沒有前例可循。最好在有效期的初期階段就把這種情況挑出來，因為此階段的警告，比起交付系統之後的錯誤報告，在處理時容易的多。此外，讓這些功能自動執行也很重要。

方法三：追蹤涵蓋範圍及 Dangler

此方法將說明如何建置 Coverage 及 Dangler 視圖，找出商業需求與使用案例之間的落差。這些視圖符合標準的 RequisitePro 視圖架構。請使用下列步驟。

- A. 從商業需求套件的 **Coverage**，選取新的 View，其中 View Type 是可追蹤性矩陣，Row Requirement Type 是 BUS，而 Column Requirement Type 是 UC。請注意我們如何重複使用「方法一」的 BUS 套件。
- B. 在 **View Properties** 方塊中，於列需求上建立名為 Business Requirements Coverage 的 Query。然後按 Add，將 Traced-from 屬性新增到 Query。
- C. 此時會啟動 **Query Requirements** 對話方塊。如圖 5 所示，選取 Not Traced 並勾選 Direct links only。

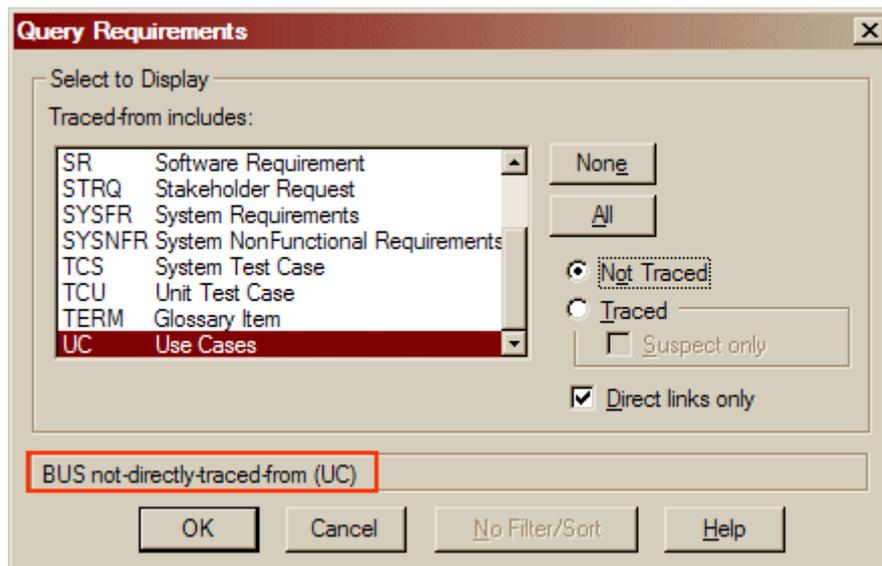


圖 5. 追蹤範圍

- D. 現在重新整理 View。您就會看到沒有相關使用案例的商業需求。您可跳過步驟 2 和 3，建立完整報告 View。
- E. 把這個方法中的 Query 準則反過來，即可建立 Dangler 視圖。從直欄需求類型（使用案例）著手，然後選取 Traced-to BUS 需求。此視圖會顯示所有沒有相關商業需求 (dangling) 的使用案例。

此方法即告完成。追溯到起始商業需求，並刪除不實的使用案例及不完整的商業需求。到此，您就能將專案安全移入解決方案的範圍。

總結

本文已探討需求工程的基本概念，並採用三項新方法來管理架構需求。此外，本文也檢視了需求的基本原則，同時說明管理商業需求、使用案例及可追蹤性的三種方法。

然而，我們的問題尚未完全解決。敬請期待「第二部分」，我們將進入解決方案領域及解決方案開發（從架構觀點）的不同階段，並探討管理解決方案交付的新方法。

作者簡介



Kumar Mani 是 IBM Global Business 解決方案的資深 IT 架構工程師，在設計及架構企業應用程式與整合解決方案方面，擁有超過九年的經驗。他的專業技能還包括使用 IBM WebSphere® Application Server、WebSphere Process Modeler、WebSphere Business Integration (InterChange)、Web services 及 Rational 工具組來設計解決方案。**Kumar Mani** 曾指導架構工程師、分析師及 IT 專員，說明如何管理端對端需求及解決方案。同時，他還曾受邀擔任產業研討會及各種 IBM 活動的演講者。**Kumar** 擁有哥倫比亞大學電腦科學及工程的理科學士及碩士學位。