

系統變體

Håkan Dyrhage

Rational Software 白皮書

TP 155

目錄

簡介.....	1
系統的變體.....	1
系統的不同組件.....	1
不同語言.....	1
多重平台.....	1
修補程式版本.....	2
子系統的變體.....	4
變化的機制.....	4
對 Rational Unified Process 的影響.....	5
對實作規範的影響.....	5
對其他規範的影響.....	5

簡介

本白皮書討論何謂系統變體以及如何處理它們。您不需要閱讀此資訊就可以瞭解 Rational Unified Process —相反地，它應該被視為 Rational Unified Process 的延伸。上一節簡短討論過變體和變化的引入對 Rational Unified Process 有何影響。

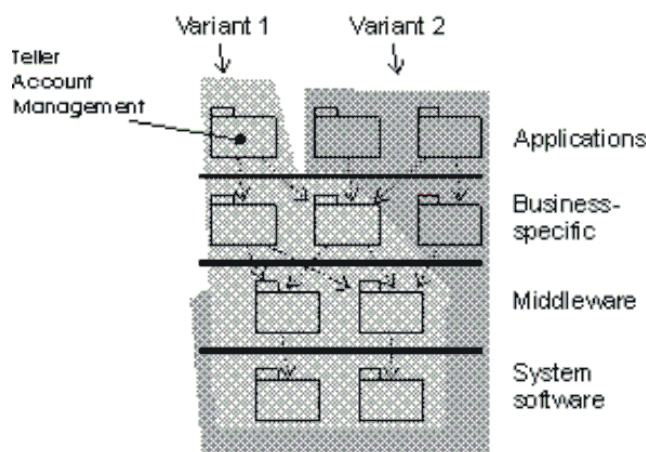
這是 Rational Unified Process 未來將改進和擴大的領域。本白皮書率先領您一睹為快。

系統的變體

許多系統以不止一個變體交付。這表示系統會針對不同（類別）客戶以不同方式來配置、套裝和安裝。有時候，只要以不同方式來安裝和調整系統，就可以達到不同變體。有時候，可向不同客戶交付系統的不同組件來達到變化。下列章節包含一些變體的範例。

系統的不同組件

完整系統的不同組件分送給不同類別的客戶。例如，銀行系統是以兩個不同的產品分送。假設系統的變體 1 包含電話銀行的一切內容，變體 2 包含櫃員帳戶管理的一切內容。執行檔是定義在應用程式層的子系統中。這表示變體（下圖的變體 1）是這樣的建置：例如它包含 Teller Account Management 子系統及它需要編譯和執行的所有子系統——亦即，它直接或間接匯入的所有子系統。



發展為兩個變體的銀行系統。

不同語言

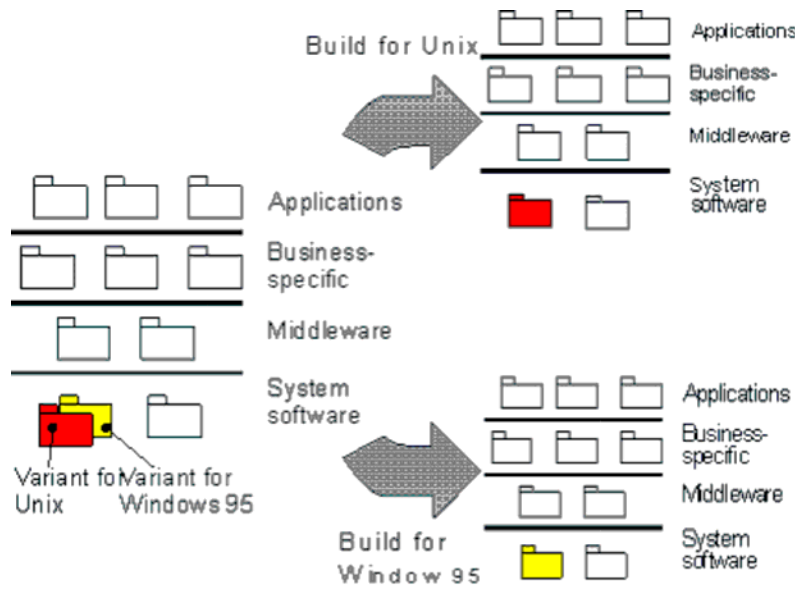
如果系統是針對不同語言而製作，例如英文、法文和日文，您可以為每一種語言交付一個系統變體。變體之間的差異在於所有文字（例如功能表和說明文字）應該以特定語言表示。

處理不同語言的方法之一是收集檔案中的所有文字，並為每一種語言提供一個檔案。交付特定語言的系統表示您要交付一切，包括含有特定語言文字的檔案在內。啟動時，軟體會讀取此檔案，並起始設定所有相關變數。

多重平台

如果系統支援多重不相容平台，則每一個平台需要系統的一個變體。比方說，如果系統是在 Windows NT 和 UNIX 上執行，則會產生系統的兩個變體。

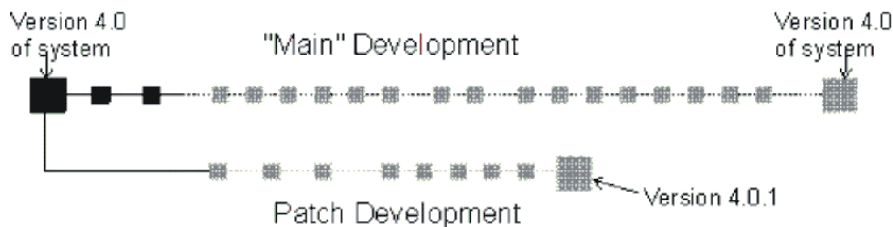
在下列範例中，平台專用程式碼位於一個子系統中。在此情況下，會開發子系統的兩個變體。編譯檔—即 **ake** 檔”—指定每一個程式碼檔案的哪一個版本要一起編譯。您也可以說 **Make** 檔指定每一個子系統的哪一個變體是建置的一部分。因此，您需要每一個平台有一個 **Make** 檔。



系統是爲了在數個平台上執行而開發的。

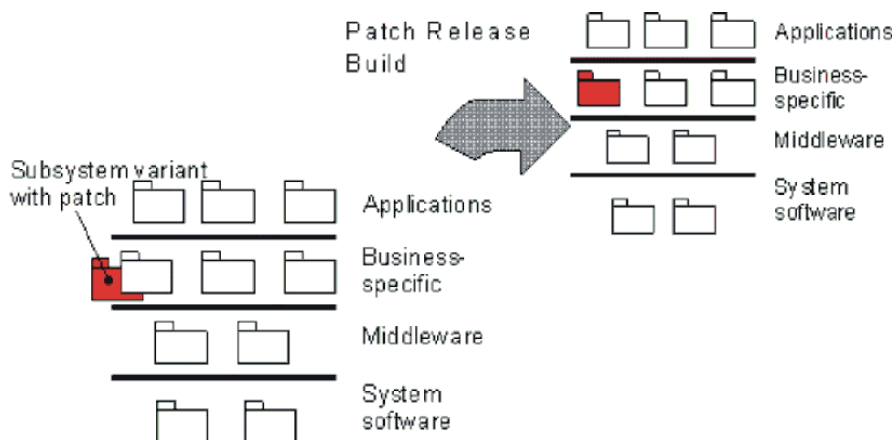
修補程式版本

有時候，需要開發系統的修補程式版本。這通常與開發專案並列完成，這表示修補程式版本是系統的另一個變體；它與系統的主要 版本並列存在。”



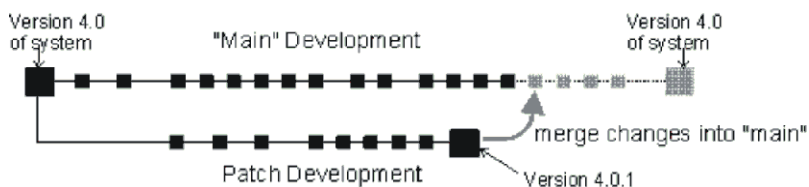
修補程式版本的開發與主要開發專案並列完成。灰色方塊指示未來版本和基準。

您需要做的變更位於一個或數個實作子系統中。因爲正常開發專案是並列進行的，所以您需要並列開發這些子系統的變體與主要開發成果。若要建立建置，您必須在 **Make** 檔中指定每一個子系統的哪一個變體是建置的一部分。



修補程式版本建置是以包含該修補程式的子系統的變體建置的。

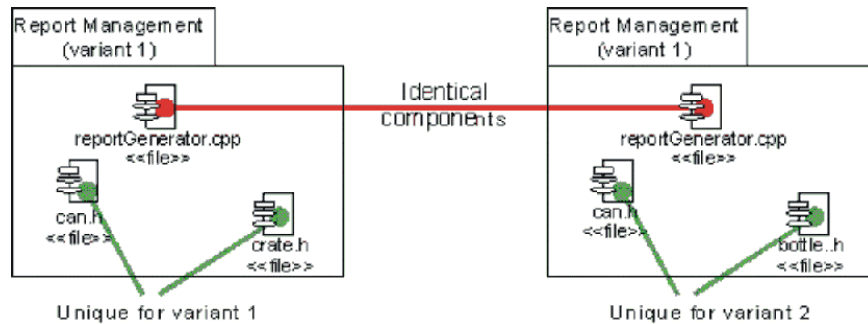
通常此種變體的有效期限較短。在發行修補程式之後，此變體不會再進一步開發，且所有重要程式碼變更會納入主要開發路徑中。



如果方便，修補程式版本的相關變更會合併到主要開發路徑。

子系統的變體

先前的例子說明了通常一個特定的子系統需要兩個以上的變體。這些變體將有一些共用元件，而其它元件則是每一個子系統變體特有的。



一個叫作 **Report Management** 的實作子系統有兩個變體，其中一個元件已決定作為兩變體的共同元件。其他元件則是每一個變體特有的。

通常，子系統的每一個變體是由一個執行者開發，而子系統的變體是並列開發的。如果元件由開發一個變體的一個執行者變更，則所做的變更應該傳送至其他變體。若要達到此目的，您需要來自**配置管理和版本控制工具 (CMVC)**的工具支援，來管理在兩個變體中應該相同的元件。

變化的機制

建立系統變體有幾個機制。其中有些在前幾節已提過。每一個機制有不同的性質。通常，您使用這些機制的組合在系統中產生變化。

- **編譯（和鏈結）檔案。**在編譯檔—即 Unix 環境中的 Make 檔—中指定每一個程式碼檔案的哪一個變體應該一起編譯及鏈結到執行檔中。
- **動態載入的元件。**開發系統的組件作為動態鏈結程式庫、Applet 或 Active X 元件，它們可以在執行時期鏈結到執行中的程式。這些元件可由 CMVC 工具來管理，並可將這些元件的子集傳送給客戶。
- **開機檔。**使用包含系統啟動來起始設定系統時軟體所讀取的資訊的檔案。例如，Windows 中的資源檔、開機檔或起始設定檔，當系統啟動並以不同方式設定系統時軟體會讀取它們。比方說，如果系統要針對不同語言而自訂，在該情況下，您要保留系統啟動時讀取的文字檔的所有文字，這時請使用此檔案。
- **以數個執行檔分割系統。**以數個執行檔開發系統；例如 .exe 檔。可向客戶交付這些組合。執行檔的不同組合由 CMVC 工具來管理。

對 *Rational Unified Process* 的影響

本節簡短說明變體的引入對 Rational Unified Process 有何影響。

對實作規範的影響

在下列領域，需要延伸實作規範：

- 新增下列步驟至**結構實作模型**作業中：
 - 如何定義應該開發系統的哪些變體
 - 如何決定哪些子系統應該有變體
 - 如何決定要使用哪一個變化機制來達到變化
- 在**規畫系統整合**作業中，您必須考量變體及規劃如何整合不同的系統變體。
- 您需要詳細說明有關子系統的變體「之間」的並列開發。比方說，如果「相同」元件應該分離時該怎麼辦？或元件是如何合併的？這也會影響執行者的數個相關作業。

其他實作作業和/或活動也會受到影響。

對其他規範的影響

開發系統的變體或系列會影響所有規範。在前幾節，已討論過對實作規範的一些影響。以下是其他規範受到何種影響的概略清單：

- **需求**—將說明如何識別系統的變體。每一個類別的客戶想要什麼。
- **分析和設計**—將說明如何在設計模型中建立變體模型，如何設計子系統變體，及如何定義變體。
- **測試**—將說明如何測試系統系列（變體）。以個別系統來測試系統的每一個變體。變體也會影響整合測試。
- **管理**—您需要組織專案，可能會針對每一個子系統變體組成個別團隊。在大型專案中，針對每一個系統變體，您甚至還有個別的專案經理。
- **環境**—您需要工具來幫助管理系統變體。
- **部署**—更加複雜，因為您要交付最終產品的客戶有數個類別。



兩個總公司：

Rational Software
18880 Homestead Road
Cupertino, CA 95014
電話：(408) 863-9900

Rational Software
20 Maguire Road
Lexington, MA 02421
電話：(781) 676-2400

免付費專線：(800) 728-1212

電子郵件：info@rational.com

網址：www.rational.com

國際辦事處：www.rational.com/worldwide

Rational、Rational 標誌和 Rational Unified Process 是 Rational Software Corporation 在美國和/或其他國家的註冊商標。
。 Microsoft、Microsoft Windows、Microsoft Visual Studio、Microsoft Word、Microsoft Project、Visual C++ 和 Visual Basic 是 Microsoft Corporation 的商標或註冊商標。所有其他名稱爲其他公司的商標或註冊商標，只做識別用途。
ALL RIGHTS RESERVED. Made in the U.S.A.

© Copyright 2002 Rational Software Corporation.
如有變更，恕不另行通知。