

# 系统变体

Hökan Dyrhage

Rational Software 白皮书

---

TP 155

# 目录

简介 ...	...1
系统变体 ...	.1
系统的不同部分 ...	.1
不同的语言 ...	...1
多平台 ...	..2
修补程序发布版 ...	..2
子系统变体 ...	...4
可变性机制 ...	.4
对 Rational Unified Process 的影响 ...	..5
对实施工作流的影响 ...	..5
对其他工作流的影响 ...	.5

## 简介

该附录讨论系统变体是什么,以及如何管理。理解 Rational Unified Process 并不需要阅读本附录;相反,它应该作为 Rational Unified Process 的扩展。最后一部分简要地讨论变体和可变性的引入将如何影响 Rational Unified Process。

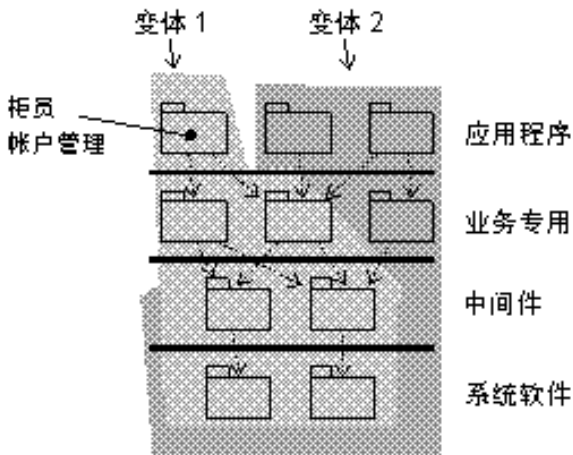
这是一个 Rational Unified Process 在未来有待改进和扩展的领域。本附录对此作了前瞻性说明。

## 系统变体

许多系统在交付时有不止一个变体。这意味着系统会为不同(类型)的客户进行不同的配置、打包和安装。有时系统安装和适应的方式稍有不同,就会产生不同的变体。在其他时候,把系统的不同部分交付给不同的客户,可以实现可变性。以下几部分列举了一些变体的例子。

### 系统的不同部分

完整系统的不同部分交付给不同类的客户。例如,一个银行系统可作为两种不同的产品交付。系统的变体 1 包含与电话银行业务有关的所有事务,变体 2 则涉及柜员帐户管理的方方面面。可执行文件在应用程序层的子系统中定义。这就是说,变体(下图中的变体 1)是使用子系统(如柜员帐户管理子系统)及其需要编译和执行的所有子系统(即它直接或间接导入的所有子系统)的一个工作版本。



开发为两个变体的银行系统。

### 不同的语言

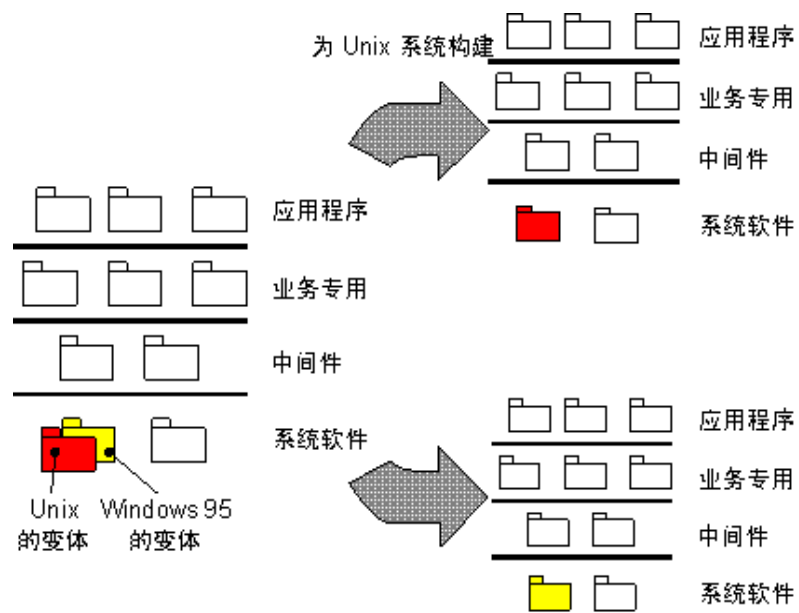
如果系统用不同的语言开发,例如,英语、法语和日语,则需要为每一种语言交付一个系统变体。变体间的区别在于所有的文本,如菜单和帮助文本,都应使用特定的语言表述。

处理不同语言的一种方法是,收集文件中的所有文本,为每一种语言准备一个文件。交付特定语言的系统即意味着交付系统的一切,再加上包含有特定语言文本的文件。软件在启动时读入该文件,并初始化所有相关变量。

多平台

如果系统支持多个不兼容的平台，那么应为每一种平台开发一个变体。例如，如果系统可在 Windows NT 和 UNIX 两种平台上运行，就需要开发两个系统变体。

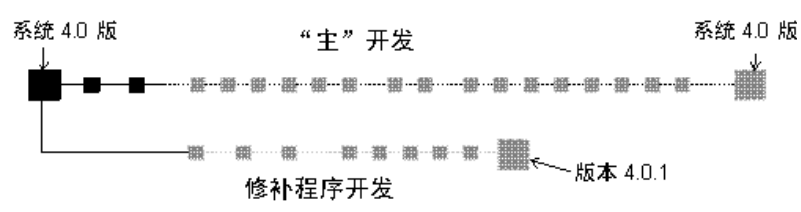
在下面的例子中，特定于平台的代码都放在一个子系统中。在这种情况下开发了两个系统变体。编译文件（一个“makefile”）指定应该一同编译每个源代码文件的哪个版本。可以说，“makefile”指定了每个子系统的哪个变体应该成为工作版本的一部分。因此，每个平台上都需要一个“makefile”。



运行在若干平台上的系统。

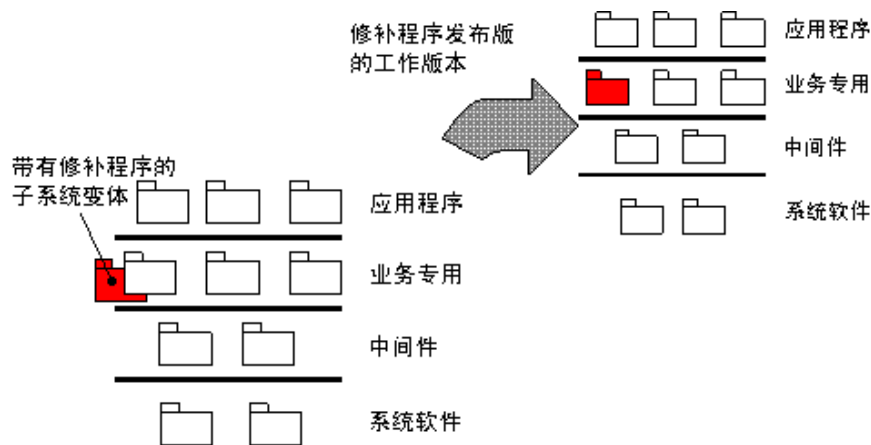
修补程序发布版

有时有必要开发系统的修补程序发布版。这通常和开发项目同时进行。这就是说，修补程序发布版是系统的另一个变体；它与系统的“主要”版本同时存在。



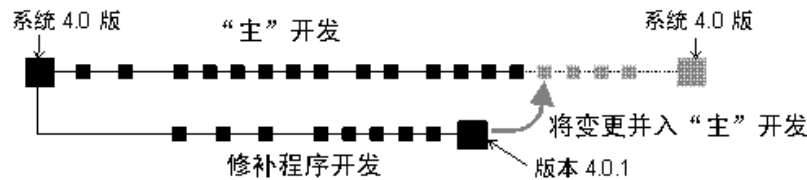
修补程序发布版的开发与主要开发项目同时进行。灰色方框表示未来的发布版和基线。

需作的修改放在一个或几个实施子系统中。由于正常的开发项目是并行开发的，因此这些子系统变体的开发工作要与主要开发工作一起进行。要创建一个工作版本，在“makefile”文件中指定每个子系统的哪个变体应该成为工作版本的一部分。



作为修补程序发布版的工作版本应使用包含该修补程序的一个子系统变体创建。

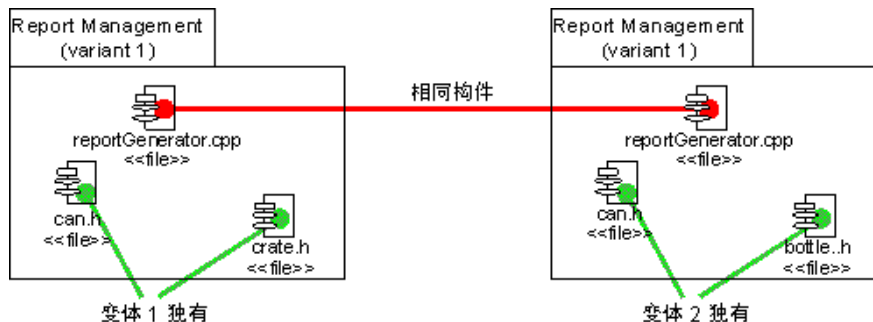
这类变体一般寿命很短。修补程序发布后，这个变体停止开发，有价值的代码变更并入主开发路径。



修补程序发布版中的相关变更在适当时并入主开发路径。

## 子系统变体

上例表明，一个特定的子系统通常需要有两个或者更多变体。有些组件是这些变体所共同的，而其他组件则是每个子系统变体特有的。



“报告管理”实施子系统的两个变体。有一个组件是两个变体所共有的，其他组件对每个变体而言是独一无二的。

通常子系统的每个变体由一位实施员开发，而且子系统变体的开发是并行的。如果组件被开发某个变体的实施员改动了，应将变更传达给另一个变体。为使之成为可能，需要来自配置管理和版本控制工具(CMVC)的工具支持，用来管理在两个变体中应保持一致的组件。

## 可变性机制

存在几种创建系统变体的机制。前面几部分已经提到了其中的一些机制。每种机制各有特色。通常要综合使用这些机制在系统中创建可变性。

- **编译（和链接）文件。**在编译文件（在 UNIX 环境下是一个 makefile）中，指定应将每个源代码文件的哪个变体编译并链接在一起，形成可执行文件。
- **动态装载的组件。**开发动态链接库、Applet 或 ActiveX 等组件，作为系统的一部分，在运行时链接至运行中的程序。这些组件可以由 CMVC 工具进行管理，这样即可向客户交付这些组件的子集了。
- **启动文件。**使用包含系统启动时软件读入信息的文件来初始化系统。例如，Windows 的资源文件、启动文件或初始化文件。在系统启动时，软件读入这些文件，相应对系统进行设置。例如，系统要为不同的语言进行定制，在这种情况下，应将所有的文本放到系统在启动时读入的文本文件中。
- **将系统分为几个可执行文件。**将系统作为几个可执行文件进行开发。如 .exe 文件。可以向客户交付这些文件的组合。可执行文件的组合由 CMVC 工具管理。

## ***对 Rational Unified Process 的影响***

---

这部分简要说明变体的引入对 Rational Unified Process 的影响。

### **对实施工作流的影响**

实施工作流需要在以下方面进行扩展：

- 向**定义实施视图**活动添加以下步骤：
  - 如何确定应该开发系统的哪个变体
  - 如何决定哪些子系统应该有变体
  - 如何决定使用什么可变性机制以实现可变性
- 在制定**系统集成计划**活动中，必须考虑变体，计划如何集成不同的系统变体。
- 对于子系统变体“之间”的并行开发，说明应更详细。例如，如果要分割“同一个”组件，或合并各个组件，应采取什么办法？这也会影响实施员的若干活动。

还可能影响其他实施活动。

### **对其他工作流的影响**

开发不同的变体或系统的各个系列会影响所有的工作流。前面几部分讨论了对实施工作流的一些影响。下面简要说明对其他工作流有什么影响：

- **需求捕获**—工作流程应说明如何确定系统变体。每一类客户需要什么。
- **分析设计**—工作流程应说明如何在设计模型中对变体建模、如何设计子系统变体以及如何定义变体。
- **测试**—工作流程应说明如何测试系统的各个系列（变体）。将系统的每个变体作为一个单独的系统进行测试。变体还影响集成测试。
- **在管理**—工作流程中，您需要组织项目，为每个子系统变体组织一支独立的开发团队。在大项目中，每个系统变体甚至可能有各自的项目经理。
- **在环境**—工作流程中，您需要帮助管理系统变体的工具。
- **部署**工作流程则复杂得多，因为要交付到的客户可分为若干类。



两家总部：

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

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

免费电话：(800) 728-1212  
电子邮件：[info@rational.com](mailto:info@rational.com)  
Web: [www.rational.com](http://www.rational.com)  
全球网址：[www.rational.com/worldwide](http://www.rational.com/worldwide)

Rational、Rational 徽标和 Rational Unified Process 是 Rational Software Corporation 在美国和 / 或其他国家或地区的注册商标。Microsoft、Microsoft Windows、Microsoft Visual Studio、Microsoft Word、Microsoft Project、Visual C++ 和 Visual Basic 是 Microsoft Corporation 的商标或注册商标。其他所有名称均仅用于标识目的，它们是其相应公司的商标或注册商标。ALL RIGHTS RESERVED.

Copyright 2006 Rational Software Corporation.  
如有更改，恕不另行通知。