CORIOLIS GROUP BOOKS

北京科海培训中心

怎样用 Java Beans 开发 Web

(美) Jalal Feghhi 著
陈强璋 王俊文 吴洪来 译
徐国定 审校

机械工业出版社

CORIOLIS GROUP BOOKS　　北京科海培训中心

怎样用 Java Beans 开发 Web

（美） Jalal Feghhi　　著

陈强璋　王俊文　吴洪来　译

徐国定　　校

机械工业出版社

著作权合同登记号:01-97-1778

内 容 提 要

本书详细介绍了软件组件结构(SCI)的基本概念、Java Beans 技术的各个细节和应用。软件组件结构是软件工程领域中又一新的逻辑模型,建立在这一模型上的软件开发技术,实现了最大程度的代码重用,从而把开发工作量降低到最低限度。这一开发技术将在客户机/服务器的技术中获得广泛应用。

本书分 3 个部分共 15 章,第 1 部分给出了 SCI 的总体描述,第 2 部分介绍了 Java Beans 技术的各个细节,第 3 部分分别介绍了 Java Beans 的应用程序。

全书叙述细致,配以丰富实例,是一本 Java Beans 的优秀参考书,适合广大 Java 语言爱好者自学使用。

(欲购原版书配套光盘的读者请与北京科海培训中心联系)

联系电话:62562449　　62589259

通信地址:北京海淀路 82 号科海培训中心　　邮编:100080

图书在版编目(CIP)数据

怎样用 Java Beans 开发 Web/(美) 费格依(Feghhi. J.)著,陈强璋等译.
—北京:机械工业出版社,1998. 3
书文原文:Web Developer′s Guide to Java Beans
ISBN 7-111-06271-X

Ⅰ. 怎... Ⅱ. ①费...②陈... Ⅲ. 网络软件—软件开发Ⅳ. TP311. 52

中国版本图书馆 CIP 数据核字(98)第 06202 号

出版人:马九荣　(北京市百万庄南街 1 号　邮政编码 100037)
责任编辑:科培　　责任校对:成昊
朝阳科普印刷厂印刷・新华书店北京发行所发行
1998 年 3 月第 1 版・1998 年 3 月第 1 次印刷
787mm×1092mm 1/16・16. 5 印张・401 千字
00001—5000 册
定　价:30. 00 元

译者序

软件组件结构(SCI)是继过程模型和面向对象模型之后软件工程领域中又一新的逻辑模型。在该模型下,应用程序由组件、框架和对象总线构成。利用SCI技术实现最大程度的代码重用,能将开发应用程序的工作量降低到最低限度,使应用程序开发人员把精力集中在有别于其他应用程序的变化部分,而这一变化部分往往只占开发工作量的很小一部分。建立在组件基础上的软件开发技术为软件的开发、维护和升级提供了一个很好的模型,它将获得日益广泛的应用,特别是在客户机/服务器的应用技术中。

本书详细介绍了SCI的基本概念、Java Beans技术的各个细节和应用。鉴于Java语言和可重用软件技术被软件开发人员广泛接受,这本书的重要性是不言而喻的。

本书第1部分给出了SCI的总体描述,并详细叙述了SCI的三个基本概念:框架、组件和对象总线。在介绍了SCI概念的两个实现(OpenDoc/CORBA和ActiveX/DCOM)之后,引入了Java Beans技术,它是SCI的又一实现,为以后在这方面的深入讨论奠定了基础。

本书第2部分介绍了Java Beans技术原理,并通过一个很简单的HelloWorld应用程序揭示了使用框架结构的优越性:框架产生了你的应用程序所需的全部代码。在这样一个例子之后,又在该部分详细介绍了Java Beans技术的全部要点:属性概念和它在应用程序中的地位、事件和JDK 1.1建立在授权基础上的事件模型化技术,扩充AWT组件中的事件处理,以及对象串行化技术的基本点和它在对象持久性上的应用。JDK 1.1中的核心为重查应用程序接口及其具体应用,以及如何提供了Java Beans组件的显式信息也在本部分进行了讨论。本部分还介绍了如何提供属性的定制编辑器。在这部分最后对Java Beans的这一机制作了综述。

本书第3部分建立在第2部分的基础上,专门讨论了Java Beans的应用程序,这些应用程序实际上是"超Java Beans组件"。读者对这部分内容和代码的研究将会受益匪浅,并能最终写出自己的Java Beans应用程序。

本书第1章到第5章由陈强璋翻译,第6章由王俊文翻译,第7章到第10章由徐国定翻译,第11章到第15章由吴洪来翻译。由徐国定负责全书校核。在本书稿件的编排和数据录入方面,王心园先生给予了很大的支持,在此表示感谢。

目 录

第1部分 软件组件结构的总体描述

第 2 部分 Java Beans 基础

第 3 部分 Java Beans 的应用程序

第 1 部分　软件组件结构的总体描述

Java Beans 是用于开发基于组件软件的一种基本结构。在软件组件结构(SCI:Software Component Infrastructure)中有三个基本的概念:框架、组件和对象总线。框架表示对问题的部分解决,它是让用户集成组件的架构。组件的集成构成软件应用,通常它针对一个特殊问题。组件是软件的基本量子(单元)。组件既应当足够小,以便于维护;又应当足够大,以使之具有功能,可以被打包和使用。对象总线是一种机制,它使得组件和框架能调用分布式环境中的另一组件或框架的服务。

上述基于 SCI 应用的特性描述,听起来可能是相当革命的。实际上,它是渐进性的,代表了软件从基于主机的同类环境向面向对象的分布式异类环境发展中正常的一步。

本书的第 1 部分致力于建立一个 Java Beans 的参考框架。第 1 章涉及 SCI。虽然有一点新东西加入到 Java 编程环境,但 Java Beans 在软件组件的可重用方面并不是一个新概念。还有一些基于组件的软件开发架构,最著名的有 OpenDoc/CORBA 和 ActiveX(OLE)/DCOM。理解这些环境可以有助于阐明 Java Beans 的基本概念。第 2 章和第 3 章提供了对这些技术的概述。第 4 章给出了 Java Beans 的架构。

第1章 软件组件结构

- 通过组件修改软件
- 用框架把握软件结构
- 用对象总线连接事务

软件组件结构(SCI)是软件工程继过程性模型和面向对象模型之后的下一代逻辑模型。面向对象技术,自这一概念提出30年以来,已经证明在解决现实世界中的问题方面是有用的。500家大公司曾报道过把面向对象的原理运用到至关重要的应用程序中的成功实例。

但是,随着更多面向对象解决方法的开发和使用,很显然的是,单单采用面向对象技术难以与如今软件应用日益增长的复杂性相抗衡。面向对象技术提供了通过信息隐藏和对象抽象而建立自我运作实体的有价值方法。对象技术的前景是通过重用类库中的对象达到代码重用。通过重用类库中已有的类,可以提高程序员的生产能力,这一点是确定无疑的。然而,对象只构成应用程序的一部分,它完全不能把握应用程序的结构(控制流)。实际情况是,大量应用程序,特别是同一领域中的应用程序,分享相似的结构。不同的程序员使用不同的技术去把握和实现这些相似的结构。结果,这些结构并没有经过通常的面向对象技术而被重用。

本章提供了对SCI有深度的概述,定义了组件和框架,并提出了基于框架、组件和对象总线的另一种软件应用程序观点。最后,探讨了对象总线并解释了它在分布式环境下的用法。

重要概念 面向对象技术、对象、类和类库,本来并不具备提供代码高度重用的能力,因为它们完全不能把握应用程序的结构。SCI提供了最高层次的代码重用。

1.1 模 型

软件组件结构是软件开发的最新模型。通过对象总线,这种模型使跨越不同操作系统、硬件架构、编程语言和编译程序的异类环境上的应用程序开发成为可能。同时,它通过引入框架使软件工程从过程化编程向类库再前进了一步。框架将代码重用的概念,通过为组件提供结构,又向前推进了一步。组件是可以重用的现成的软件部件,它可以从软件经销商处买到。组件被安插到框架,框架和组件通过对象总线与别的框架和组件连接。组件、框架和对象总线组成了软件应用程序。

一个应用程序的不同组件能存在于不同的操作系统之中,也能以不同语言实现。用过的代码可以包裹在组件中,和系统的其余部件一起使用。在新环境中实现新的组件会立即利用到软件和硬件方面的新技术。新的组件通过对象总线与已有的组件交互运行。

应用程序的一个组件可以独立于别的组件而加以改写不需要重新编译和重新插用整个应用程序。易于升级和维护足以应付如今软件应用日益增加的复杂性。易于插用使软件经销商能迅速向顾客推出新的功能。

基于组件的软件降低了市场进入的障碍，因为软件的最小部件是组件而不是整个应用程序。软件经销商以最少的人员支持也能设计和实现缩小包装的组件，并将之作为现成的软件销售。

重要概念　软件组件结构提供了在分布式环境下的可交互操作性。它实现最大程度的代码重用，并为软件维护和升级提供了一个较好的方法。SCI 方便了使用并降低了市场进入的障碍。

我们可以根据组件、框架和对象总线来配制应用程序。这种应用程序的配制方法是根本不同于导致整体型应用程序的软件工程的传统方法。图 1.1 说明了我们对应用程序的概念模式。

在图 1.1 中，应用程序是以填上灰色阴影的整个区域表示。这个区域包含整个框架、框架里面的组件总成、由对象总线提供的安全服务、用户接口水平功能和对象总线本身的一个部分。

那被包括在内的对象总线部分并不是指该总线里面的某个特殊软件成分：该总线的全部功能对于可交互操作性都是需要的。但我们可以把网格区域看作正在运行应用程序不同部分的硬件平台联成网络所需的全部软件。用户接口功能是提供给对象总线用户的有附加值的功能。这些功能本身可以看作是建立在组件和框架顶部的应用程序。所以，我们对应用程序构成概念还可以推广到包含一个以上的框架。安全服务是总线提供的核心功能。这种服务本身也是一个组件。水平和垂直功能以及总线服务的细节包含在以下各节之中。网格区域的框架提供该应用程序的结构，框架内的组件总成表示了建立一个应用程序所需的开发努力(在框架一节将给出一个组件总成的精确定义)。这种努力对于这个特定应用程序是独特的，表示了整个应用程序的变化部分。其余的网格部分表示应用程序的不变部分。如果我们认识到 SCI 可以使应用的不变部分达到 80%而变化部分只有 20%时，就可以体会到 SCI 功能的显著。这里给出的这些概念将在以下各节得到充实。

重要概念　一个应用程序可以由多个框架、组件和一个对象总线配制而成。这种构成使得软件经销商把注意力集中到应用程序的有附加值的功能上去，而 SCI 只提供一般性的结构和功能。

1.2　组　件

基于主机的应用程序的整体性质已不能应付软件应用程序日益增加的复杂性。传统的客户机/服务器方法一直尝试把一个应用程序分为客户机一方和服务器一方，以此来降低应用程序的复杂性。但是，这种模型最终只是将应用程序以两大块来代替一整块。传统的客户机/服务器技术确实成功地提供了具有相当确定的接口(服务)，并允许客户机访问其服务的服务器应用程序。但是，它们既不规定也不实施任何进一步将客户机和服务器划分成更小的部分的技术。对这种整体性的客户机和服务器，软件经销商在维护和升级时，代价通常是极

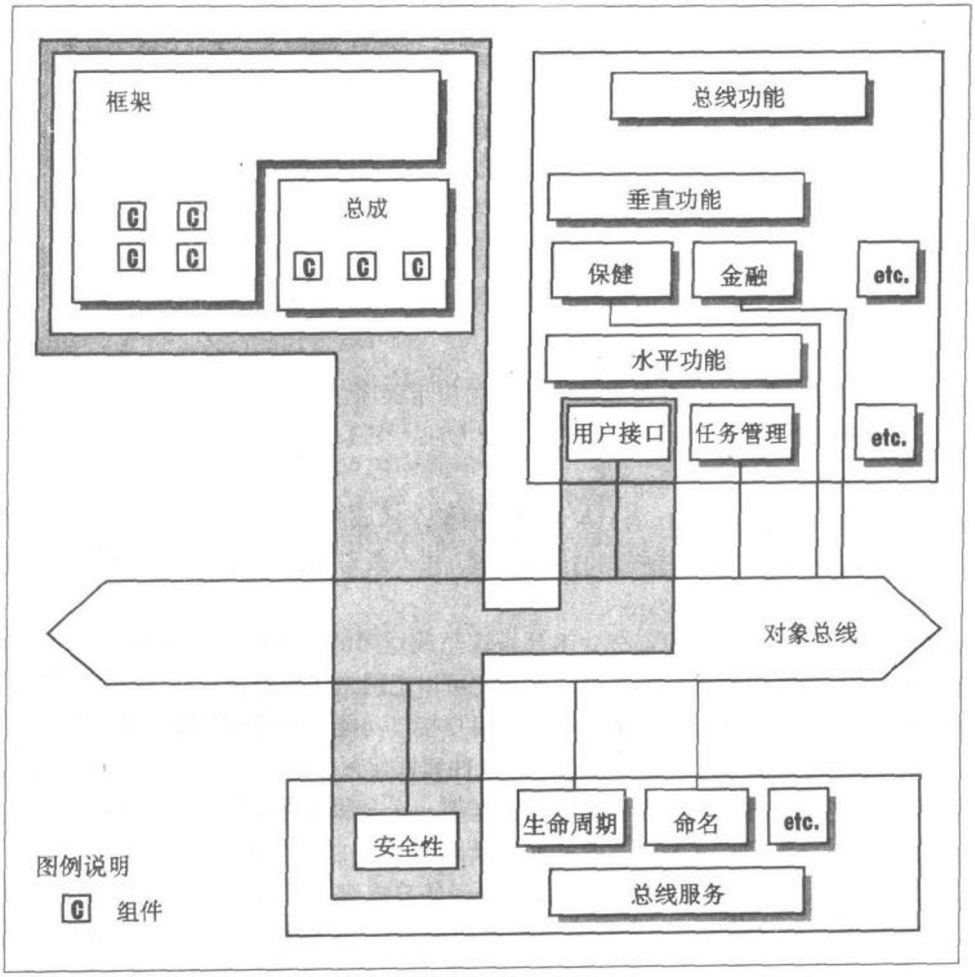

Figure 1.1 The formulation of an application in terms of components,

图1.1 按照组件、框架和对象总线构成的一个应用程序

其昂贵的。

典型的客户机/服务器技术的另一固有问题是缺乏即插即用的功能。编写应用程序的客户机一方必须具有服务器那一方的特定知识。只有这样的客户机才能与服务器交互运行。让独立开发的应用程序去调用现有服务器的服务是行不通的。

基于组件的软件开发是客户机/服务器计算的新方向，它试图克服传统的客户机/服务器技术的缺点。据估计，到1996年底全部开发者中66%将用组件开发软件(见Strategic Focus 1995)。这一节的余下部分将详细讨论组件。

重要概念 传统的客户机/服务器应用程序不能应付软件应用日益增加的复杂性。它们不能加入到即插即用的环境之中。

1.2.1 什么是组件

我们把组件定义为软件的基本量子(单元)。这种单元应当足够大,以使组件是有功能的,并能被打包和插用。但是,这种单元又应当足够小,以便于它的维护和升级。

重要概念 组件是软件的基本量子。它既是足够大,因而是有功能的、可插用的,但又是足够小,因而是可维护的。

一个组件有如下属性:

- 可插用——组件是现成的可打包的软件部件,它可以从经销商处购得。
- 可维护——组件是足够小的,因而易于维护和升级。
- 有功能——组件是足够大的,使它以所希望的有用方式动作。
- 有特点——组件是设计好用来执行很特殊任务的,它代表了整个应用程序中的一个部件。这个部件可以是细颗粒的、中等颗粒的或粗颗粒的。
- 没有与别的组件直接通信的路径——组件不直接与别的组件通信。不直接通信对于组件的可维护和可升级是必不可少的。此外,它允许它们用在分布式环境。注意,这一属性并不意味着一个组件不能请求另一组件的服务。相反,需要强调的是,请求服务的实际机制是由另一实体(对象总线或框架)来实现的。
- 自我可描述——组件能向系统的其余部分描述它提供的服务。这种描述通常是由通过说明性的、与实现无关的接口定义语言提供的。
- 目标框架——组件通常是为一特殊的框架所建立的。这种组件不可能在其他框架中运行。例如,Java Beans 组件不能直接用于 ActiveX 环境。但是,对象覆盖(object wrapper)可以用于这种跨框架的功能。
- 目标对象总线——直接与某一对象总线接口的组件,它也许不能与另一对象总线一起使用。要用网关来提供两个不同对象总线之间的可交互操作性。

上述组件特征提供了将一个应用程序分成若干组件的机制。每个组件提供了一个相当专门的功能,它也向框架的其余部分描述自己,以便别的组件能访问它的功能。这种描述是经过说明性语言实现的,它本质上是把组件的接口与实现分离开来。这种分离对组件在分布式环境中运行是至关重要的。

由于组件与组件之间不能直接发生交互作用,一个很自然要问的问题是:组件是如何相互协同配合以形成一个应用程序的呢?其回答是通过使用框架或对象总线。

组件对分布式计算是极其有用的。它们对软件开发者来说具有巨大的价值。但是,由于其专门性,它们对最终用户可能没有那样的价值。最终用户通常需要涉及的组件是直接映射到真实世界中的实体,如飞机、汽车和宾馆那样的组件。这些真实世界组件称为事务组件。

1.3 框 架

人们具有极大兴趣应用面向对象技术去解决真实世界的问题。类库在软件工业中正在激增。自对象技术概念出现经过 30 多年,它已经找到了进入生产系统和至关重要应用程序的道路。对象技术真正实现了信息隐藏和数据抽象的承诺。但是在提供高度代码重用机制

方面它还没有那么成功。

确实，类库提供了细颗粒的对象，它们能被各种应用程序分享。但是，这些细粒对象只占全部应用程序的一部分。它们完全不能把握将这些对象拼装在一起所需的逻辑联系。图1.2、图1.3和图1.4提供了基于过程、基于对象和基于框架的应用程序的高层表示。

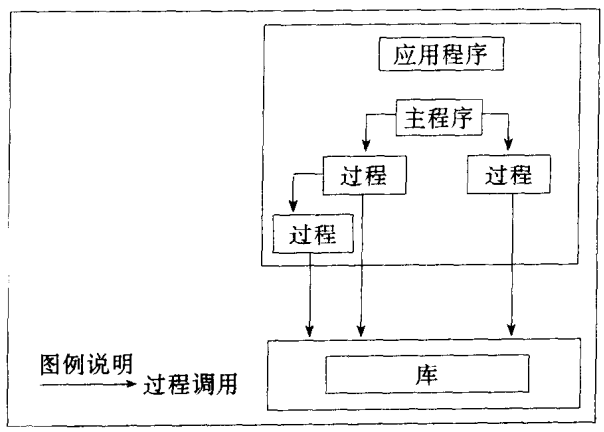

图 1.2　基于过程的应用程序

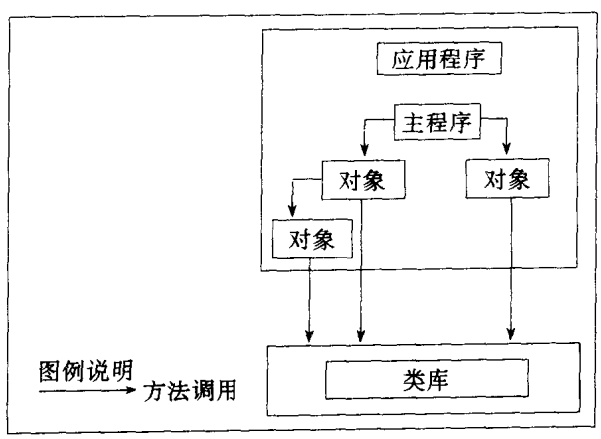

图 1.3　基于对象的应用程序

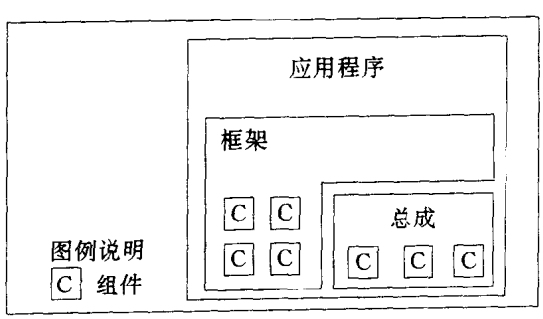

图 1.4　基于框架的应用程序