

Designed for
Microsoft®
Windows NT®
Windows 98



CD-ROM
Included



Microsoft® 程序设计系列

FOR
ENTERPRISE
DEVELOPERS

Designing Component- Based Applications

基于组件的应用程序设计

内附
配套光盘

使用 Microsoft®
Windows® DNA
构建企业解决方案

北京大学出版社

Microsoft Press

7.12
1.58

基于组件的 应用程序设计

[美] Mary Kirtland

北京博彦科技发展有限公司 译

北 京 大 学 出 版 社

· 北 京 ·

著作权合同登记 图字: 01-1999-1922 号

Designing Component-Based Application

Mary Kirtland

本书版权为 Mary Kirtland 所有, 1998。(Copyright © 1998 by Mary Kirtland. All rights reserved.)

本书中文版由美国 Microsoft 出版社授权北京大学出版社独家出版, 1999。

本书封面贴有北京大学出版社的激光防伪标签, 无标签者不得销售。

版权所有, 翻印必究。

图书在版编目(CIP)数据

基于组件的应用程序设计/[美]Mary Kirtland 著; 北京博彦科技发展有限公司译. —北京: 北京大学出版社, 1999.10

ISBN 7-301-04185-3

I. 基… II. ①M… ②北… III. 程序设计 IV. TP311.11

中国版本图书馆 CIP 数据核字(1999)第 31640 号

书 名: 基于组件的应用程序设计

著作责任者: Mary Kirtland 著 北京博彦科技发展有限公司 译

责任编辑: 邱淑清 熊 勇

标准书号: ISBN 7-301-04185-3/TP·461

出版者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

网 址: <http://cbs.pku.edu.cn/cbs.htm>

电 话: 出版部 62752015 发行部 62754140 总编室 62752032

电子信箱: zpup@pup.pku.edu.cn

排 版 者: 北京博彦科技发展有限公司

印 刷 者: 北京大学印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 24.5 印张 518 千字

1999 年 10 月第一版 1999 年 10 月第一次印刷

定 价: 64.00 元(含光盘)

译者序

本书是根据美国 Mary Kirtland 著, Microsoft Press(微软出版社)出版的“Designing Component-Based Applications”一书翻译的。它介绍了一种新的、基于组件的编程概念。列举了当前可用的和将来可用的 Microsoft 技术,描述了怎样设计、实现、测试、调试和部署基于 COM 和 MTS 的三层式应用程序。本书的焦点集中在三层式应用程序的中间层,即“业务逻辑层”。贯穿全书,使用了一个示例应用程序来说明各种概念和技巧。

书中的建议来自 Microsoft 的 3-2-1 程序的经验。其中“3-2-1”表示三层式(3)应用程序,需要两个(2)开发人员工作一个月(1)的时间。

本书的主要读者群是那些全日制专业开发人员。某些内容对系统管理员也很有参考价值。

随本书配售的光盘中包括了本书原著的一个电子版本。只需从中单击一下鼠标,就可以找到任何特定的信息。

参加本书翻译的有黄建江、黄林红、韩贵玲、朱仰太、常斌、黄太成、乔娜、任民、李莉、严德、刘霄田、何琴、罗纯夫、戈遂炎、贾鲁剑和柳锐。黄太成审校了全部译稿。

译文中的术语尽量采用已为 Microsoft 所认可的。但对少量仍需商榷的术语,我们也根据原文和所描述对象的实际含义,确定一种我们认为最合适的译法。

尽管我们的确在煞费苦心追求尽善尽美,疏漏之处仍在所难免,恳请读者批评指正。

北京博彦科技发展有限公司

1999年9月于北京

致谢

首先,我要感谢我在 Microsoft 的几位经理,他们为我编写本书提供了大力支持。他们当中有: James Utzschneider、Dave Cameron 和 Hugh Teegan。我还要感谢那些为我分担了其责任的人们,正是由于他们的帮忙,我才得以集中注意力编写此书。他们当中有: Jeff Wickman、Cathy Brennan、Marc Levy 和 Jocelyn Garner。

我也要感谢 COM 组的成员,他们帮助我阐明了 COM、MTS 和 COM+ 的各个方面的技术。我要特别提到他们当中的 Marc、Jocelyn、Tracey Trewin、Pat Helland、Robert Barnes、Gerald Hinson、Dick Dievendoff、Scott Robinson 和 Markus Horstmann。Visual Studio 组的 Tracey Trewin 和 Sharon Chen 负责本书所引用的“Island Hopper”示例应用程序。我感谢他们两人开发了这一示例,并不惜时间和精力解释了他们所采用的各种办法的理由。Robert 和他领导的工作组开发了 MTS Performance 工具包,本书第十三章“性能验证”就是基于这一工具包编写的。STEP Technologies 的 Bob Beauchemin 和 Shaun McAravey 开发了 MTS 培训教程,该教程为本书提供了最初的编写思路和大纲。Gary Voth、Charlie Kindel、Peter Houston、Angela Mills、David Lazar 和 Kamaljit Bath 编写了好几种白皮书,我通过这些白皮书学习了我的主要专业领域之外的很多技术。

很多人认真阅读了本书的初稿,并提供了颇有价值的反馈意见,从而使本书有了不少改进。他们是 Jocelyn Garner、Vincent Sacksteder、Mike McKeown、Ralph Squillace、Tim Ewald、Bryan Lamos、Mario Benuzzi、Chris Sells、Govind Kanshi、Paul Stafford 和 Mario Raccagni (如果我漏掉了某个人,我表示歉意。您的评述意见一定是进到了我的电子邮件收件箱,但因为我的不慎而把它删除了……)。我要特别感谢 David Chappell,正当我为我的书稿水平深感自卑时,他的充满鼓励的话语来到了我的耳边。

最后,我要感谢 Microsoft 出版社的每一个成员,他们为我付出了艰辛的劳动,帮助把我的那时松散的不成熟的资料变成了一本真正的有价值的书,他们是 Eric Stroo、Wendy Zucker、Jennifer Harris 和 Jack Beaudry。

Mary Kirtland

引言

在 20 世纪 90 年代后期,低价位计算机得到了广泛应用,以及在全球范围内访问信息也变得现实可行,这就点燃了对一种新型应用程序的需要之火。这种新型应用程序就是高分布式应用程序。高分布式应用程序被全球范围内极其广泛的人使用。他们通过非持久性或低速链接连接到许多应用程序和数据服务器。这些特征使得传统的应用程序体系结构已不能再满足应用程序的需求。

Microsoft Windows 的 Distributed interNet Applications(DNA: 分布式网间应用程序)体系结构代表了 Microsoft 用以创建高分布式应用程序的方法。DNA 应用程序采用的是一种逻辑上三层的、基于组件的体系结构。Microsoft 的系统服务提供了这种类型的应用程序所需要的基础结构。而关键的基础结构服务则是 Component Object Model(COM: 组件对象模型)和 Microsoft Transaction Server(MTS: Microsoft 事务服务器)。组件对象模型提供了组件交互作用的基本机制,而 Microsoft 事务服务器则为构造不同规模的服务器应用程序提供了工作环境。

本书的目的在于帮助您构造第一个基于 COM 和 MTS 的分布式的三层式应用程序。它不试图解决在编写企业级的应用程序时将会遇到的每一个问题。它也不试图定义一种对每一个应用程序开发群体都有效的方法。但它的确能给您一些实际可行的忠告,帮助您设计、开发、测试和部署基于 Microsoft Windows 的 DNA 技术的三层式应用程序。本书的侧重点主要集中在三层中的中间层(商业逻辑层)上。正是在这一层,您将使用 COM 和 MTS。

本书中的建议都是根据从 Microsoft 的“3-2-1 编程”中得到的经验给出的。“3-2-1”代表三层式应用程序,两个开发人员,一个月的时间。也就是说,它代表了这样一个工程项目:为开发一个三层式应用程序,需要两个开发人员工作一个月的时间。3-2-1 规划是一种 MTS 快速入门的方法。有关这方面的详细信息,请参见本书附录。

本书为哪些人而写?

本书的主要读者群是那些全日制专业开发人员,他们应熟悉一种面向对象的程序设计

语言,比如 Microsoft C++、Java 或 Microsoft Visual Basic。如果他们对 Internet 和一般的数据库概念有一定的了解(这并不是必要的),那对他们了解本书的内容会是有帮助的。下面涉及的一些主题对系统管理员也是有吸引力的。

本书各章节中的某些部分假定您已经有了一定的使用 Visual Basic 或 Visual C++ 编程的经验。我还假定不需太多的屏幕图示来说明怎样安装一个应用程序或使用一种开发环境,您就能够懂得怎样使用您的各种工具附带的文档。我很厌倦为了说明如何选择—一个菜单选项,而不厌其烦地罗列屏幕图示和操作步骤,我希望您也和我一样。

内容简介

本书分为三个部分。第一部分和最后一部分提供了关于当前可用的和将来可用的 Microsoft 技术的一般性说明。中间部分描述了怎样设计、实现、测试、调试和部署一个基于 COM 和 MTS 的三层式应用程序。

第一部分 技术概览

这一部分提供了一个概要说明,说明的对象包括: Windows DNA,三层式应用程序体系结构,以及用来创建基于 Windows DNA 解决方案的多项技术——包括 COM、数据访问、MTS 和 Active Server Page(ASP: 活动服务器页)。我们还将考察一下其他的可行的技术,比如 COM Transaction Integrator(COMTI: COM 事务集成器)和 Microsoft Message Queue Server(MSMQ: Microsoft 消息队列服务器),它们可与 MTS 集成。

第二部分 构造应用程序

这一部分向您展示了一种途径,以它来建立三层业务解决方案的体系结构,并将其模型化。其中讨论了用来将数据业务对象和各种类型的客户模型化的工具、技巧和思考过程,还考察了在设计过程中必须想到的实际问题,如安全性考虑和性能需求。

接下来考察怎样实现三层式应用程序。其中的侧重点在于构造商业对象和数据对象。我们还将考察怎样利用 ActiveX 控件、分布式 COM(DCOM)、ASP 和 Dynamic Hypertext Markup Language(DHTML: 动态超文本标记语言)为 Internet 和企业内联网编写客户。

最后,这一部分还考察了怎样利用 Microsoft Visual Basic 和 Microsoft Visual Studio 调试程序,以进行交互式程序调试,并介绍了针对在应用程序设计过程中所建立的性能需求,检查三层式解决方案的技巧,也介绍了部署三层式解决方案的技巧。

第三部分 MTS 的未来

这一部分说明了怎样用现有技术(比如 COM、MSMQ 和 COMTI)来扩充您的应用程序,使

之与其他平台集成。您也将大致了解 COM 和 MTS 未来的方向,即所谓的COM+。

您所需要的工具

为了利用本书描述的技术构造和部署一个应用程序,您需要安装下面这些产品:

- ◆ Microsoft Windows NT 4.0, Workstation 或 Server。您可以利用 Workstation 来测试您的应用程序,但应该利用 Server 在企业内部部署您的应用程序。
- ◆ Microsoft Windows NT 4.0 Option Pack (含在随本书配售的光盘中),它包含 Microsoft Windows NT 4.0 Service Pack 3、Microsoft Transaction Server 2.0、Microsoft Internet Information Server 4.0、Microsoft Message Queue Server、Microsoft Internet Explorer 4.01,以及一个关于 Microsoft SQL Server 6.5 的评估许可。
- ◆ Microsoft SQL Server 6.5 或 Oracle 数据库。在本书发稿时,它们是 Microsoft Transaction Server 2.0 所支持的两种数据库。关于对其他数据库的支持情况,请查阅 Microsoft COM 的 Web 网址 <http://www.microsoft.com/com/>。您可以利用 Microsoft Windows NT 4.0 Option Pack 中的 SQL Server 评估复制进行开发和测试。
- ◆ 支持创建 COM 对象的开发工具。本书描述的很多技巧都是用 Microsoft 的开发工具中的术语来说明的。在采用其他销售商的工具时,也可能有同样的技巧可供采用。

当然,为利用分布式调试和部署技术,在您的 Windows NT 机器上,至少应该连接一台客户机。这样的客户机可以是一台 Windows 95, Windows 98 或 Windows NT 机器。

如果您想探索运行在非 Windows 平台上的 COM 组件或资源的互操作性,您还将需要附加的软件和机器。对于直接支持与其他平台或资源间的互操作性的 Microsoft 产品,在它们的产品文档中包含有您所需要的关于其他特定产品的信息。在 <http://www.microsoft.com/com/resource/cominop.htm>, 您可找到关于其他 Microsoft 互操作性的信息,包括 Microsoft 所支持的 COM 实现方法。

示例程序代码

贯穿全书,我们使用的都是“Island Hopper”示例应用程序,以说明各种概念和技巧。“Island Hopper”是 Microsoft 的 COM 组和 Visual Studio 开发组开发的,用于说明对三层式应用程序怎样才算好的设计、开发和部署技术。在 Microsoft 的 Web 网址上,您也许可以找到更新版本的“Island Hopper”,但这里使用的“Island Hopper”版本包含在随本书配售的光盘中,其中也提供了为了运行或构造这一版本的“Island Hopper”,需如何配置机器的说明。

随本书配售的光盘中包括有本书原著的一个电子版本,其中内容已被编译为

HTML Help(帮助)文件。本书的联机版本提供了全文本搜索功能,以及在全书中交叉引用超级链接的功能。这样,只需单击一下鼠标,您就可以找到任何特定的信息,比如一个过程或一个定义。

目录

译者序	1
致谢	3
引言	5

第一部分 技术概览

第一章 Windows DNA	1
1.1 商业问题,结构化的解决方案	1
1.1.1 应用程序体系结构	2
1.1.2 组件	6
1.1.3 三层模型启用器	6
1.2 Windows DNA 的体系结构	7
1.2.1 组件对象模型	8
1.2.2 表示层	9
1.2.3 商业层	9
1.2.4 数据访问层	11
小结	12

第二章 组件对象模型	13
2.1 为什么要采用 COM?	13
2.2 COM 编程模型	14
2.2.1 对象	15
2.2.2 接口	16
2.2.3 类	23
2.2.4 组件	24

2.3 分布式 COM	30
2.3.1 COM 的安全性	30
2.3.2 重新访问注册	32
2.3.3 远程激活和调度	36
2.4 Automation(自动化)	39
2.4.1 IDispatch 接口	39
2.4.2 类型库	41
2.4.3 双重接口	42
小结	44
第三章 数据访问基础	45
3.1 为什么采用 MDAC?	45
3.2 访问数据	46
3.2.1 ODBC	46
3.2.2 万能存储	47
3.2.3 万能数据访问	48
3.3 OLE DB	49
3.4 ActiveX 数据对象	52
3.5 远程数据服务	53
小结	55
第四章 Microsoft Transaction Server	57
4.1 为什么要使用 MTS?	57
4.2 MTS 的多种面孔	59
4.2.1 作为 ORB 的 MTS	59
4.2.2 作为 COM 对象代理的 MTS	61
4.2.3 作为 TP-Monitor 的 MTS	62
4.3 应用程序服务器编程模型	68
4.3.1 易用性	69
4.3.2 通过资源共享实现伸缩性	69
4.3.3 规则 1: 调用 SetComplete	70
4.3.4 规则 2: 获取和独占接口指针	73
4.3.5 规则 3: 使用和释放资源	74
4.3.6 规则 4: 使用基于角色的安全性	74
4.3.7 规则 5: 使用事务	78

小结	81
第五章 Active Server Pages	83
5.1 为什么要使用 ASP	83
5.1.1 Automation 组件	85
5.1.2 ASP 编程模型	86
5.2 三层式应用程序体系结构	86
5.3 使用 MTS 进行集成	86
小结	88
第六章 扩展 MTS 的功能	89
6.1 UNIX 上的 COM	89
6.2 XA 的交互操作性	90
6.3 COM 事务集成器	91
6.3.1 COM+ 体系结构	92
6.3.2 创建一个 COM+ 自动化组件	93
6.4 Microsoft Message Queue Server 消息队列服务器	93
6.4.1 消息队列	94
6.4.2 使用 MSMQ	95
小结	97

第二部分 构造应用程序

第七章 应用程序的定义和模型化	99
7.1 创建一个应用程序定义	100
7.1.1 定义商业问题	100
7.1.2 定义功能规范	101
7.1.3 建立应用程序需求	103
7.2 应用程序的模型化	108
7.2.1 持久数据模型化	108
7.2.2 设计类	112
7.2.3 数据对象模型化	115
7.2.4 商业对象模型化	121
7.2.5 表示层模型化	126
7.3 定义物理结构	127

7.3.1 将类分组为组件	127
7.3.2 将组件分组为程序包和进程	127
7.3.3 将程序包和进程指定到机器	128
7.3.4 建立物理结构文档	128
小结	130
第八章 构造数据对象	133
8.1 从设计到实现	133
8.2 为 MTS 实现组件	133
8.3 数据对象技术	136
8.3.1 利用 ADO 编程	137
8.4 实现数据对象	142
8.4.1 在 Visual Basic 中实现 db_CategoryC	143
8.4.2 在 Visual Basic 中实现 db_CustomerPasswordC	161
8.4.3 快速考察 db_CustomerC	171
8.4.4 用 Visual C++ 实现数据对象	175
小结	190
第九章 构造商业对象	191
9.1 从设计到实现	191
9.2 为 MTS 实现组件	192
9.2.1 构造功能	192
9.2.2 管理状态	197
9.2.3 处理错误	201
9.2.4 安全地访问您的组件	203
9.3 构造商业对象	204
9.3.1 在 Visual Basic 中完成实现 bus_PaymentC	205
9.3.2 在 C++ 中实现 bus_InvoiceC	212
9.3.3 利用 SPM 实现 util_TakeANumber	228
小结	233
第十章 包装组件	235
10.1 程序包	235
10.2 设计程序包	237
10.2.1 激活	237

10.2.2 共享资源	238
10.2.3 故障隔离	238
10.2.4 安全性隔离	238
10.2.5 “Island Hopper”程序包设计	239
10.3 构造程序包	240
10.3.1 创建程序包	241
10.3.2 向程序包中添加组件	242
10.3.3 设置激活属性	243
10.3.4 配置安全性	244
10.3.5 设置事务属性	249
10.3.6 输出程序包	250
小结	252
第十一章 构造表示层	255
11.1 从设计转到实现	255
11.1.1 表示层应用程序的类型	255
11.1.2 选择一种表示层体系结构	256
11.2 实现表示层	261
11.2.1 使用商业对象	261
11.2.2 使用 RDS 访问远程对象	264
11.2.3 使用数据绑定	265
11.2.4 使用 ASP	266
11.2.5 使用客户端组件	268
11.3 实现“Island Hopper”应用程序的表示层	268
11.3.1 设计决策	268
11.3.2 实现 Classifieds 应用程序	269
11.3.3 实现 Internet 客户	276
小结	285
第十二章 调试和排除故障	287
12.1 组件级测试	287
12.2 本地测试	289
12.2.1 调试 MTS 组件	290
12.2.2 追踪和确认	294
12.2.3 将错误消息解码	295

12.2.4 检查并排除数据库方面的问题	296
12.2.5 用于排除故障的 MTS 工具	296
12.2.6 Windows NT 事件查看器	300
12.3 测试分布式应用程序	301
小结	302
第十三章 性能验证	303
13.1 为什么要进行性能验证?	303
13.2 性能验证的工作过程	304
13.2.1 定义性能需求	304
13.2.2 评估性能	306
13.2.3 找到并消除瓶颈	308
13.3 常见瓶颈	312
13.3.1 SQL Server 瓶颈	312
13.3.2 数据访问瓶颈	312
13.3.3 事务瓶颈	313
13.3.4 其他瓶颈	314
小结	314
第十四章 部署分布式应用程序	317
14.1 选择拓扑结构	317
14.1.1 集成到公司现有的应用程序中	317
14.1.2 满足性能需要	318
14.1.3 常用部署配置	318
14.1.4 容错功能	325
14.2 部署过程	325
14.2.1 安装预先建立好的程序包	326
14.2.2 创建客户安装程序	327
14.2.3 最终确定程序包设置	328
14.2.4 安装客户程序	330
小结	330

第三部分 MTS 的未来

第十五章 扩展应用程序	333
15.1 利用 COMTI 开发组件	333
15.1.1 使用 Component Builder	334
15.1.2 COMTI 支持的事务处理程序	336
15.2 利用 MSMQ 开发应用程序	339
15.2.1 MSMQ 应用程序模型	339
15.2.2 编制 MSMQ 应用程序	341
15.2.3 配置 MSMQ 应用程序组件	346
小结	346
第十六章 COM+	347
16.1 发展中的 COM 和 MTS	347
16.1.1 一种统一的体系结构	347
16.1.2 一种统一的编程模型	348
16.1.3 一种统一的管理模型	350
16.2 针对企业应用程序的新服务	350
16.2.1 对象池	351
16.2.2 对象构造器	352
16.2.3 补偿资源管理器	352
16.2.4 安全性强化	353
16.2.5 队列组件	354
16.2.6 事件	356
16.2.7 内存中的数据库	358
16.2.8 动态负荷平衡	360
小结	361
附录 关于 3-2-1 程序	363
参考书目	365

第一部分 技术概览

内容提要 在第一部分中,我们要考察分布式应用程序开发人员所面对的业务问题和技术方面的挑战,并介绍 Microsoft 解决这类问题的方法,即采用 Distributed interNet Applications(DNA: 分布式网间应用程序)。我们要考察 Windows DNA 所推荐的三层的、基于组件的体系结构,以及用来创建 DNA 解决方案的技术。这样的解决方案包括 Component Object Model (COM: 组件对象模型)、Universal Data Access (UDA: 万能数据访问)、Microsoft Transaction Server(MTS: Microsoft 事务服务器)和 Active Server Pages(ASP: 活动服务器页)。

第一章

Windows DNA

当今的商业环境是极其活跃的,并极富竞争性。多家公司组建成联合公司,或合并为新的公司。为了降低成本和掌握竞争的主动权,各公司都在缩小商业过程的规模和重新设计商业过程。要求管理机制和客户都不断地进行变化。这种变化在一个多国企业中可能因地域不同而产生差异。商业是快速变化的,永远不会停息,而且是在全世界范围内发生的。

1.1 商业问题,结构化的解决方案

在这种活跃的商业环境中,分布式计算机系统起着重要作用。现代企业集团的各个方面,比如市场跟踪、加工制作、财务状况、客户支持,以及您可能想象到的几乎所有其他方面,都依赖于这种企业集团的信息技术(IT)产业。IT 系统对每一个机构方面的运作提供支持。这些系统必须协同工作,才能提供在整个企业集团范围内密切配合的信息