Developing Applications with Visual Basic and UML

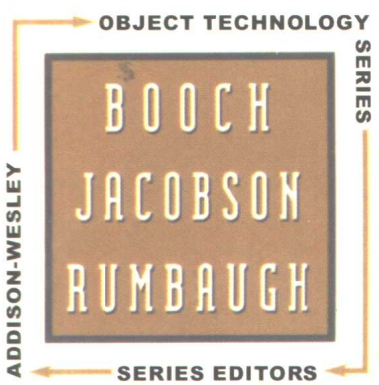
使用 Visual Basic 和 UML 开发应用程序

PAUL R. REED, JR.

李 博 崔慧敏 译

王 菁 审校

Forewords by Grady Booch and Francesco Balena



清华大学出版社

Pearson Education

培生教育出版集团

Addison-Wesley 科海电脑技术丛书

使用 Visual Basic 和 UML 开发应用程序

[美] Paul R. Reed, Jr. 著

李 博 崔慧敏 译

王 菁 审校

清华大学出版社

Pearson Education 培生教育出版集团

(京)新登字 158 号

著作权合同登记号：01-2001-5309

内 容 提 要

本书结合作者在面向对象的客户/服务器系统开发方面的丰富经验，介绍了如何在 Visual Basic 中用 UML 对企业级应用程序进行建模和开发。书中不仅对 UML 基础知识和 Visual Basic 中易被误用的面向对象属性进行了清晰阐释，还围绕一个大型案例研究，引导读者亲历项目开发周期的各个阶段：需求分析、建模、设计原型、创建体系结构到编码，让读者了解 UML 的设计优势，告诉读者如何把 UML 规范转换成 Visual Basic 代码，还教会读者如何结合 DNA，DCOM 和 MTC 技术及 ASP 等工具开发面向对象的 Visual Basic 应用和网络应用。

本书实用性强，适用于任何想成功创建 VB 应用程序并保证其长期运行的程序开发人员。

Developing Applications with Visual Basic and UML

书　名：使用 Visual Basic 和 UML 开发应用程序

作　者：Paul R. Reed, Jr.

译　者：李　博　崔慧敏

审校者：王　菁

出版者：清华大学出版社(北京清华大学校内，邮编 100084)

印刷者：北京市耀华印刷有限公司

发行者：新华书店总店北京科技发行所

开　本：787×1092 1/16　　印张：22.75　　字数：553 千字

版　次：2002 年 6 月第 1 版　　2002 年 6 月第 1 次印刷

印　数：0001～4000

书　号：ISBN 7-302-05465-7/TP・3222

定　价：39.00 元

Grady Booch 序

看到本书的题目，首先你会说“Visual Basic 并不是面向对象的！”是的，准确地讲，Visual Basic 是基于对象，而不是面向对象的一种编程语言（Visual Basic 中没有真正的继承，而在 C++和 Java 中却存在继承）。这就是说，面向对象的原理：抽象、封装和信息隐蔽在 Visual Basic 中依然有效。进一步讲，当你面对大型系统时，将关系分离（与平衡分配责任）的结构原理也是可以运用的。这正是 Paul 写这本书的目的所在。他将向你讲述应用面向对象的概念、使用 Visual Basic 来开发和部署高质量的系统的原则、过程和编程方法。

今天我们看到的 Visual Basic 与 20 世纪 60 年代初期在 Dartmouth 开发出来的最初的 Basic 有很大区别。当 Alan Looper（被认为是 Visual Basic 之父）还在微软的时候，他就开发了一个叫做 Ruby 的程序，这其中已经蕴含了可视化开发的雏形。大约在 1989 年，Ruby 和一个叫做 QuickBasic 的产品结合在一起，这样 Visual Basic 的第一个版本就诞生了。从那时起到现在，Visual Basic 获得了长足的发展。结合比如 Microsoft Transaction Server，Active Server Page 和 Structured Query Language 这些技术，Visual Basic 可以用来创建以 Web 为中心的系统。事实上，许多报告指出，Visual Basic 是当今世界上应用最广泛的编程语言，使用它的开发人员要比使用 COBOL、C++和 Java 的开发人员多。它如此广泛存在的原因可能是因为 Visual Basic（还有与之相关的 VBScript）经常被用作使用微软技术构建多机种系统的粘合剂。

其实，使用 Visual Basic 作为系统开发工具的一个主要原因，就是它能够非常容易地快速构建简单系统。这一点也好（特别是计划中要求开发的速度很快的时候）也不好（当这个系统在它早期的基础上继续发展的时候，它很快就达到一个极点，此时它变得非常的脆弱，并且很难再做更改，所以系统的后续开发容易在这种困境中夭折）。而采用面向对象的方法来开发 Visual Basic 系统，可以减少这种危险。另一方面，使用统一建模语言可以让你建立系统的蓝图，此蓝图可以统一不同投资人的观点，并且在这个项目的整个过程中起指导作用。

在这本书中，Paul 将向你讲述如何在面向对象的前提下应用 Visual Basic，如何用统一建模语言生成系统蓝图。他也将向你讲述如何在 Visual Basic 中应用抽象、封装和信息隐藏这些原则，以及如何构建大规模系统。我特别喜欢 Paul 这本书的原因是它的实用性。他不仅强调了那些原则，还讲述了如何在 Microsoft Transaction Server，Active Server Pages 和 Structured Query Language 这些技术的基础上，创建可以实际运行的系统。

Grady Booch
Chief Scientist
Rational Software Corporation

Francesco Balena 序

不久之前，许多程序员都将 Visual Basic 看作是一种“玩具语言”，或者，最多看作是“像胶水一样的语言”，认为它只适合于将 ActiveX 控件，数据访问对象和组件对象模型这些不同的技术粘合到一起。

实事求是地讲，最初版本的 Visual Basic 可能确实如此。Visual Basic 版本 1 甚至不能处理数据库，Visual Basic 版本 2 只能通过 Open Database Connectivity（ODBC） API 函数来实现对数据库的处理，准确地说，这并不是获取数据最简单的方法。Visual Basic 版本 3 增加了一些有限的数据库功能，但是对于企业级规模的项目，他们仍不能提供足够的支持。

当微软凭借发布 Visual Basic 版本 4 来进入客户端/服务器市场时，一切就发生了变化。Visual Basic 版本 4 是一个有竞争力的、功能强大的产品，使用它可以创建完整的 N 层应用程序。Visual Basic 程序员不仅可以创建组件对象模型组件，他们甚至还可以利用 Remote Automation（Distributed COM 的前身）来尝试编程，并使他们的程序跨网络执行。进行分布式计算从此变得简单。自从那时起，Visual Basic 就开始大踏步的向前发展：Visual Basic 版本 5 包含了一个编译器，并且加入了创建 ActiveX 控件的能力，以及许多和 Internet 有关的特性。Visual Basic 版本 6 增加了对 Active Data Objects，Internet Information Server 和 Dynamic HTML 的更好的支持。所有的这一切，再加上其他的产品如 SQL Server 7，Microsoft Transaction Server 和 Microsoft Message Queue 的支持，你将体会到使用 Visual Basic 可以开发出功能非常强大的应用程序。

然而，在我的印象中，当然这是在我参加过的研讨会和我在欧洲和美国做过的调研，以及从我作品的读者那里得来的反馈信息的基础上得出的结论，就是许多人仍然把这门语言当作是一种快速应用程序开发（RAD）工具。这个错误的想法来自于许多人认为用户界面的好坏是一个应用程序成功的关键。当然这种看法在比较小的应用上还是正确的，有时候在简单的中型数据库应用程序中也还是可以的，但是在大型的客户端/服务器系统中，它就是绝对不行的了。许多雄心勃勃的企业项目在开始的时候就没有一个认真的设计阶段，没有初步的文档，没有风险评估，没有类的建模，而所有这些都应该在写程序第一行代码之前进行。因此当听说有大量的项目没有按时完成，或者开发的应用程序性能很低或者难于维护与扩展的时候，任何人都不会感到意外了。

这就是本书所要解决的问题。我喜欢这本书有很多的原因，我也非常高兴能有机会为这本书写序。我喜欢它的原因之一就是，尽管现在有许多关于统一建模语言和面向对象分析与设计的书，唯有此书是从 Visual Basic 程序员的角度，以一种令人信服的方式来讨论这些问题。Paul 在这两个领域当中都有丰富的经验，因此可以很好地说服那些崇尚纯粹的面向对象编程的人以及只热衷于使用 Visual Basic 的程序员。作为一个例证，你可以阅读第 2 章，来看看对 Visual Basic 拥有面向对象能力的最具说服力的分析。

Paul 很聪明、并且正确的把这本书的重点放在那些创建大型的 N 层 Visual Basic 应用程序时必不可少的内容上，而没有把这本书写成是完整的统一建模语言的参考手册。与此同

时，对于他没有详细说明到的内容他提供了足够详细的信息，这些信息如同精确的参考文献一般。因此，有兴趣的读者可以知道应该再看些什么文献来完善他们的技能。

我相信，Visual Basic 程序员会欣赏在整本书中渗透的实践的气氛。当你看到自己能够将理论同 ActiveX，Active Data Objects，Internet Information Server，Componet Object Model 和 Distributed Internet Architecture 这样的技术结合起来的时候，不是很有趣么？有关在 Windows 应用程序之间处理数据的许多技巧的彻底讨论，是一个非常好的例子，有时候这是你从纯理论到实际的应用这个转换过程中必经的阶段。另外，通过描述结合和不结合 Microsoft Transaction Server 两种方式来写代码，我们看到作者是多么注意那些看上去非常小，但是却能很大程度上决定一个应用程序成功与否的细节。

最后让我用一点建议做我的结束语。不要一页一页匆匆而过，而不去动你的 Visual Basic 代码编辑器。你应该认真阅读所有的内容，理解每一章中提出的概念，并花一些时间来创建你自己的类和数据库，如果可能的话再使用一种建模工具。只有这时，你才会意识到，采用统一建模语言，并进行仔细的系统设计能够在大的项目上节省你几天，几星期甚至几个月的时间。

Francesco Balena
www. vb2themax. com
Editor-in-Chief, Visual Basic Journal, Italy
Contributing Editor, Visual Basic Programmer's Journal
Author, Programming Microsoft Visual Basic 6 (Microsoft Press)

前　言

为什么买这本书？

现在大多数的软件工程项目达不到预期目标，或者不能够按照计划的时间完成。关于这一点，我的看法是大多数的项目开发小组缺乏对开发过程的理解，或者不知道如何根据具体项目的特点，制定具体的计划。另外，大多数的项目在分析和设计时缺少对如何完成各项工作的说明。也就是说，通常的项目没有可追踪性。

一般来说，大多数写 VB 方面书籍的作者都没有从“大”的方面来考虑它。相反的，他们把注意力集中到了小的地方，整页整页充满了各种加载列表框以及调用 Windows API 函数的方法。尽管这一方面也非常必要，但不幸的是几乎没有人谈到过项目计划、软件工程，以及创建企业级 VB 应用程序方法论。这是因为，相比之下，这是一件探索起来或者讲述起来更为复杂的题目。

这本书重点讲述当今对企业级应用程序进行建模和开发的最为强大的方法：1997 年 OMG（对象管理组）采用的为面向对象应用程序建模的“统一建模语言”（UML，Unified Modeling Language）。通过使用 UML，再结合一个好的开发生命周期（在这本书中我介绍了 Synergy 过程模型），VB 项目就可以较为容易地获得成功，同时避免那些可能出现的问题。另外，尽管这本书中讲述的是 VB，但当结合 UML 时，这里所讲述的所有概念都可以应用于其他的编程语言（例如，C＋＋，Java）。

可悲的事实

我开始接触计算机工作是在 1979 年，那时我在 IBM 使用像 IBM 的 IMS（信息管理系统）和后来的 DB2 这样的技术来开发大型 IBM 主机应用程序，现在你们中的大多数人称它们是“过时”程序了。然而，相对于“过时程序”我更喜欢说是“传统系统”或说是“先前的系统”。在这里，我不仅学到了很多东西，使用了一些先进的工具，并且还和一些绝顶聪明的人合作。此外，我懂得了项目计划和创建一个清晰的结构和为目标应用程序进行设计的重要性。当使用一个好的进程创建了清晰的项目小组之间的通信方式时，我可以看到它所带来的巨大好处。更为重要的是，它为成功的完成一个项目提供了一步步的基石。

在 1990 年，我在 OS/2 平台上使用 Smalltalk 开发第一代的客户/服务器应用程序。这是我新事业的开始，这时，我对在客户/服务器环境下创建应用程序所使用的“进程”感到非常吃惊。计划非常的松散，分析和设计报告（就是那些说明我们为什么要创建我们正在创建内容的文档）也都是一样。

在我使用 PowerBuilder 和后来使用 VB 开发软件的时候也一直是这个样子。这样上交的应用程序是可以工作的，但是它们都是非常脆弱的。现在，我觉得许多标记为“客户/服务

器"的应用程序几乎与那时的主机应用程序一样容易过时，更有甚者，在实际应用一两个月之后，它们就成为过时的程序了。产生这种问题的原因不在开发工具上。更准确地说，这是由于缺少一个好的过程模型和一个好的方法，此方法要确保我们所创建的就是用户实际需要的，并确保我们设计的系统不会在第一次改变时就崩溃掉。

慢慢地，我开始在客户/服务器环境下开发应用程序的时候使用我自己的过程和方法，效果非常好。应用程序更有弹性，而且能更好地适应变化，用户一般也都会对完成的项目表示满意。

这本书写出了我结合 UML 创建客户/服务器应用程序的所有经验，我感觉这是当前制作分析和设计应用程序文档的最好方式。我希望你能如同我喜欢写这本书一样来喜欢读这本书。

谁应该读这本书

这本书是为所有想成功创建 VB 应用程序并保证程序能长时间使用的人所编写的。它提供了准确的路标，可以让大家达到下面这些目的：

- 创建合适的项目计划(在附录 E 中深入的表现出来)。
- 高可信度的项目估算，而不是那种以漫不经心的方式作出的估算(在附录 A 中深入的表现出来)
- 使用 UML 支持的模型来了解和描述出应用程序的需求。
- 基于 UML 支持的模型和应用程序需求方的体系结构，创建合适的设计图。
- 使用 Microsoft 的 Distributed Internet Architecture(DNA，分布式因特网应用程序体系结构)策略的强大功能，来创建位置透明的应用程序。
- 使用如 Rational Software 的 Rational Rose，或者 Microsoft 的 Visual Modeler 这样的可视化建模工具来创建和跟踪 UML 生成品，并生成组件代码的框架。(注意：Microsoft 提供的 Visual Modeler 是 Rational Rose 的子集。除了创建动态模型的部分，本书中讲述的 Rational Rose 都可以用 Visual Modeler 来代替。)
- 有效地使用 Microsoft 最新的技术，如 Distributed Component Object Model(DCOM，分布式组件对象模型)，Microsoft Transaction Server(MTS，微软交易服务器)和能够使用 Active Server Page(ASP，动态服务器网页)、VBScript、JavaScript 的因特网。

现在所有创建 VB 应用程序的人都需要这本书。

使用这本书你需要先知道些什么

如果想从这本书中获益，在开始时你需要了解一下你不需要知道什么。

第一，你不需要知道关于 UML 的任何内容。我会展现 UML 中最基本的内容，而且更为重要的是它们是如何和 VB 的内容发生关联的。尽管 UML 是通过 9 类单独的图反映出来的，你可以通过最核心的一组获得最大的收获。

第二，你不需要在面向对象的概念方面有很深的背景。我在正文中会讨论标准的对象结

构并且在附录 C 中回顾这些问题。

第三，你不需要知道 COM 或 DCOM。我在整本书中都大量地使用了它们，并在附录 D 中涉及了一些更为深入的问题。

最后，你不需要对关于 MTS 和 World Wide Web(WWW，万维网)的关键技术有一个很深的理解。在本书中对它们都详细的进行了处理。

本书假定你已经知道 VB 中的基本知识了。VB 编程的新老程序员都可以受益，我没有讲关于 VB 最基础的内容，我假设这些内容你已经会了。如果你没有任何 VB 的经验，不妨也先购买这本书，然后在你学习了一点这种编程语言之后再来看这本书。

这本书还假设你接触过 SQL 和关系数据库这些知识。如果知道一些 ActiveX Data Objects(ADO，ActiveX 数据对象)和 Open Database Connectivity(ODC，开放数据库连接性)的知识会更好。在本书中作为例子模型的项目使用 ADO 而没有使用 ODBC 驱动程序。

这本书的结构

下面是这本书中的各章的内容。

第 1 章 项目的困境

本章中回顾了当前软件开发的状态，以及我对形成这种状态的原因的理解。这里还提到了迭代和增量方式的软件开发，并且对在本书中使用的 Synergy 方法进行了概述。这里还提及到了将在后面章节中深入讨论的 UML 的基本组件。

第 2 章 Visual Basic，面向对象和 UML

本章中讨论了使用 VB 作为开发环境会带来的好处。它通过讲述 VB 怎么实现封装、继承和多态来证明这一点。然后，这里将 UML 映射到各种 VB 程序开发中去。重点包括将 UML 中的类映射到 VB 的类模块；使用例子路径映射到 VB 中的实体、接口，还有控件类型的类；将组件图映射到 VB 的可执行程序和 DLL，以及可选的 MTS。

第 3 章 开始项目

本章中探讨了在这本书中使用到的用例研究——Remulak Productions。这个假想的公司销售乐器，他们需要一个新的接收订单系统。这里介绍了项目计划图以及它的工具，称为事件表，结合它们来快速地确定应用程序特点。进而，本章将事件映射到第一个 UML 模型，用例。

第 4 章 用例

在这一章中讨论了 UML 中的一个主要图，用例。还包含了一个制作用例文档的模板。定义了用例中的活动者和它们的角色。在本章中讨论了用例路径的概念，以及项目最初的执行结构。另外还讨论了如何通过用例对项目进行估算。

第 5 章 类

本章讨论了 UML 中的类图，这是 UML 中最为重要的图表。这里提供了一些确定类的建议和一些定义各种类型的关联的建议。这里还覆盖了商业规则分类，以及怎么将这些规则转换为类的操作和属性。最后，本章中讨论了为了更好地处理所有 UML 内容，怎么来使用可视化建模工具——Rational Rose。

第 6 章 创建一个早期原型

在本章中讨论了每一个用例独特的用户接口需求。它创建了一个早期原型的流和最终的图形原型。最后，它将在创建原型过程中获得的信息映射到 UML 内容中去。

第 7 章 应用程序中的动态元素

本章中讨论了 UML 支持的动态模型，深入地探讨了两个主要图(通常称为交互图)：顺序图和协作图。这些内容与用例中的路径有着直接的联系。其他讨论的动态图包括状态和活动图。

第 8 章 技术概述

这一章中讲述了 DNA 所要求的将逻辑服务分层的重要性。还探讨了针对 Remulak Productions 这个案例中使用的解决方案，包括使用 DCOM，MTS 再加上使用 HTML 表单与 ASP 的互联网技术的分布式解决方案。在 Internet 应用程序部分使用到了两种脚本语言：VBScript 和 JavaScript。

第 9 章 数据持久性：存储对象

这一章中探讨了将类图转换成 Microsoft SQL Server 和 Oracle 都支持的关系设计的步骤。它在类图转换为 RDBMS 时如何处理继承、如何进行设计等方面均提供了权威性的建议。这里还探讨了 VB 如何支持数据部件类，以及将数据翻译分层的重要性，这其中含有大量的与每个类密切相关的 SQL 逻辑。

第 10 章 应用基础结构

在这一章中完成了应用程序各个层的设计。它还展示了在不同层之间进行通信的机制，以及可能的选择。每一个类都属于以下三种类型之一：实体，接口或控制。这是进行设计的基础和提供其他部署策略解决方案的基础。

第 11 章 从 UML 类图生成代码(第 1 部分)

这一章处理使用 Rational Rose 直接从类图中生成 VB 代码的问题。它讨论了为了使这个过程顺利进展所需要的安装步骤，并且讨论了反复再工程，以及保证其实现的过程的重要性。

第 12 章 由 UML 类图生成代码(第 2 部分)

这一章中将在前面那章中生成的 VB 类根的体中填充代码,重点介绍如何通过用例进行单向填充。它还展示了实现一个真正的应用所需要的所有代码,从用户接口到后端。

第 13 章 创建一个分布式应用:DCOM 和 MTS

这一章利用在第 12 和第 13 章中得出的结果来讨论在各种服务器配置情况下展开部署的步骤。它覆盖了要成功使用 DCOM 所需要的所有任务,包括使用部署向导,还有使用各种工具来保证客户端和服务器都能得到正确的安装。

第 14 章 可选接口:Internet

这一章中讲述了现在最热门的话题之一:Internet。这是前面所有章节中所做的努力得到回报的地方。在这一章中,我们创建了一个基于 HMTL 表单的前台的订单入口查询功能。这里使用 ASP 作为浏览器和 VB 应用程序层之间的接口。这一章中探讨了将 VBScript 作为 Web 服务器上最主要的脚本语言,和将 JavaScript 作为浏览器上主要脚本语言的问题。在项目的这个阶段,我们使用 Microsoft 的 Visual InterDev 作为主要的开发工具。

更新和信息

我有幸在很多知名的公司和组织中工作,这些公司不仅在美国,还分布在欧洲、亚洲和南美。在我的这些经历中,我时时会有一些新的关于应用 UML 并使用 VB 以及 C++和 Java 来创建适应性更强的应用程序的想法。你可以访问我的站点,www.jacksonreed.com,从中你可以获得我最新的培训和咨询服务,也可获得本书中全部的源代码。欢迎并且鼓励你同我联系,我的 email 是 prreed@jacksonreed.com。

目 录