

“没有人能把 COM 阐释得比 Don Box 更清楚”

—— Charlie Kindel, 微软公司 COM 专家

COM 本质论

Essential COM

DON BOX 著 潘爱民 译

Charlie Kindel 和 Grady Booch 序

- 洞悉 COM 的“为什么”，而不仅限于“如何做”
- 精通 COM 作为程序设计基本素养
- 深入理解分布式 COM

COM本质论

Essential COM

DON BOX 著 潘爱民 译

Charlie Kindel 和 Grady Booch 序

中国电力出版社

内 容 提 要

本书由一流的权威撰写，揭示了 COM（组件对象模型）的本质，旨在帮助开发人员真正理解 COM 的为什么，而不仅限于怎么使用。书中还提供了一个完整的分布式 COM 应用，演示了各种编程技术。是程序员必备的参考书。

本书适合有经验的软件开发人员阅读，也可作高校计算机相关专业软件组件课的教材。

图书在版编目（CIP）数据

COM 本质论 /（美）伯克斯（Box, D.）著；潘爱民译.—北京：
中国电力出版社，2001

ISBN 7-5083-0611-2

I. C… II. ①伯…②潘… III. 软件接口, COM—程序设计 IV.
TP311.52

中国版本图书馆 CIP 数据核字（2001）第 23499 号

北京市版权局著作权登记号 图字 01-2000-3086

本书英文版原名：Essential COM

Published by arrangement with Addison Wesley Longman, Inc.

All rights reserved.

本书中文版由美国培生集团授权出版。版权所有。

中国电力出版社出版、发行

（北京三里河路 6 号 100044 <http://www.infopower.com.cn>）

北京地矿印刷厂印刷

各地新华书店经售

*

2001 年 8 月第一版 2001 年 9 月北京第二次印刷

787 毫米×1092 毫米 16 开本 24 印张 530 千字

定价 49.00 元

版 权 所 有 翻 印 必 究

（本书如有印装质量问题，我社发行部负责退换）

译 序

关于这本书，我最想说的一句话是，它来得太迟了。

自 1993 年 COM 首次发布以来，COM 本身经历了重大的变化，但是它的基本思想和结构一直保持相对稳定，这也正说明了 COM 思想的魅力所在。尽管如此，由于 COM 的神秘本质，并不是每个人都能够很容易地理解 COM 的思想，理解 COM 的魅力。幸运的是，现在有了这本书，它会清楚地告诉你这一切是如何发生的。

本书无疑是 COM 领域著作中最负盛名的一本。两年前我从 MSDN Library 上看到这本书的第三章和第五章，当时的感觉是内容非常零乱，所以对于这本书的盛名非常不解。后来抽时间阅读了 amazon.com 上读者和专家们的评价之后，才意识到这本书对于提高 COM 的理解是多么重要。这次翻译使我有机会仔细地阅读本书的每一个细节，虽然是在原书出版三年之后阅读本书，但是其中的内容现在仍然很有参考价值。

这不是一本供新手学习的书，也不是一本教你如何开发 COM 组件的书，它的目的是帮助你在一定的基础上继续提高对 COM 的理解。如果用它来学习 COM，你可能会失望，因为许多内容需要有一定的经验基础才能看得明白；但是如果用它来加深自己对 COM 的理解，那么这本书再合适不过了，读完这本书之后，COM 对于你会“变得再明白不过”（这也正是《Inside OLE》的作者 Kraig Brockschmidt 和本书的作者 Don Box 在顿悟了 COM 和 OLE 之后的境界）。期望你会有同样的感受。

COM 是跨语言的组件对象模型，但是它与 C++ 和面向对象思想的渊源可谓深矣。本书会向你解释这些内容，但前提条件是你必须具备扎实的 C++ 基础知识，特别要理解 C++ 类的二进制结构以及类型转换机制。反过来，阅读本书也可以加强你在 C++ 和面向对象思想方面的理解。此外，本书还涉及到许多 COM 高级特性，例如线程模型（或者套间模型）、名字对象、连接点等，因此，要想从本书获得最大受益，你也需要有 COM 的基本知识做基础，我建议读者先系统地学习 COM 之后再阅读本书（可以参看《COM 技术内幕》或者《COM 原理与应用》）。

自 1997 年以来 COM 一直是 Microsoft 技术的热点，但是 COM 的热潮似乎随着 Microsoft .net 的推出而冷却了下来。实际上，COM 在 Microsoft Windows 平台上永远有它的一席之地，目前没有任何迹象表明要被其他的结构替换，它将与传统的 SDK API

一起在底层提供各种服务。在新的.net 结构中,COM+继承了 COM 所有的优势,并对 COM 重新做了包装,只是感觉上 COM 与应用层的距离更远一些,因此,本书中讲到的几乎所有内容都适用于未来的应用系统。如果读者对于 Windows 2000 中的 COM 感兴趣的话,可以参阅 David Chappell¹ 撰写的《深入理解 Microsoft Windows 2000 分布式服务》和 David S.Platt 撰写的《深入理解 COM+》。

在开始翻译这本书之前,我有幸得到了台湾著名电脑技术作家侯俊杰先生赠送的繁体版《COM 本质论》(侯俊杰译),这为我的翻译工作提供了很大的帮助。而且我决定采用同样的书名,以保持两个中文版本的一致性。但是本书的叙述风格仍然采用我在其他 COM 书籍中的做法,并且主要术语也保持一致。在此我特别感谢侯先生所做的一切,当然也要感谢 Don Box 为我们写了这么好的一本书。

潘爱民

2001 年 2 月于北京大学燕北园

¹《Understanding ActiveX and OLE》一书的作者,这是另一本重量级的 COM 著作,可惜没有中译本。

Charlie Kindel 序

当我坐下来写这篇序言的时候，我的脑子里有好几个想法：

- Don 的照片会出现在封底吗？如果是这样的话，他的头发会有多长？
- 本书的读者是否知道 Don 有一张非常个性化的车牌，上面写着“IUNKNWN”？
- 一个人在一本书的序言里到底能写些什么呢？

关于到底要在这篇序言里写些什么，我有两个想法。第一，我想写下很久以来就一直想写的、关于 COM 设计思想的内容；第二个想法是，就像 Don 为了让我高兴而请我写这篇序文一样，我也使劲捧一捧他。最后，我决定两样都做。

什么是 COM？为什么要发明 COM？Don 在第 1 章简洁地回答了这些问题。在介绍部分的最后，他这样说：“……本章展示了一个可重用的组件结构，这个结构允许多个独立开发的二进制组件能够动态、有效地组成一个复合系统。”这一章的其他部分带领读者走过 COM 设计者在 1988 年到 1993 年期间经历的过程，COM 最终是在 1993 年首次发布的。

我认为 COM 取得如此长久的成功，要归因于几个方面的因素。第一也是最重要的是实用性，然后是简单性，从而产生灵活性，或称可塑性。

实用性

COM 以一种非常务实的方式来处理软件设计。它不是根据学院派的面向对象编程理论来提供解决方案，而是充分考虑了人的自然因素以及资本的因素。经典面向对象理论中最好的、又最具商业价值的思想被设计小组挑选出来，并且与 Microsoft 内外在以前的软件工程中所获得的有关软件重用的经验结合在一起。

大多数经典的面向对象课本都认为，只要一个系统或者一种语言支持封装性（信息隐藏）、多态性和继承性，那么它就是面向对象的。继承性通常被强调为“实现重用性的主要工具”。但是 COM 的设计者并不同意这种观点。他们认为，这是一种过于简单的看法，实际上应该存在两种继承性。实现继承（implementation inheritance）表示真正

的实现（即行为）被继承；而接口继承（interface inheritance）则表示只有行为的规范被继承。正是后一种继承形式实现了多态性。COM 完全支持这种形式的继承。另一方面，实现继承只不过是重用现有实现的一种简单机制。然而，如果重用是最终目标的话，那么实现继承只是通向目标的简单途径，但不是目标本身。

无论是研究领域还是商业软件开发领域，人们普遍认为，实现继承是一种有效的、功能强大的工具，但是它也会导致基类和派生类之间的过分关联（或称耦合）。由于实现继承往往会造成基类实现细节的“泄漏”，从而违反了基类的封装特性，所以 COM 的设计者感觉到，应该把实现继承局限于组件内部的开发过程中。虽然 COM 不支持跨越组件的实现继承，但是它支持组件内部的实现继承。而且 COM 也完全支持接口继承（事实上，它完全依赖于接口继承）。

COM 设计者并没有特别强调继承本身，继承通常被认为是获得重用性的一条有效途径。在 COM 中，用于构造重用性的基础概念是封装，而不是继承。相反，COM 通过继承建立起功能相似的对象之间的类型关系模型。在封装基础上建立起 COM 的重用模型之后，COM 设计者鼓励使用一种被称为“黑盒子”的重用，这种重用性能够符合预期中的组件市场。这种思想是，客户应该把对象看作不透明的组件，不管其内部是什么，或者它们是如何被实现的。COM 设计者相信，应该设计出一种结构来加强这种思想。为什么要用不同的重用模型来设计一个系统呢？问得好。事实上，这个世界充满了“没有使用黑盒子封装形式”的面向对象系统，甚至这些系统要实现这种黑盒子封装都非常困难。C++就是一个典型例子，在本书的第 1 章，Don 把我在这里所说的想法描述得非常清楚。

下面的等式说明了面向对象程序设计和面向组件程序设计之间的区别：

面向对象程序设计 = 多态性 + （某些）迟绑定 + （某些）封装性 + 继承
Object-Oriented Programming = Polymorphism + (Some) Late Binding + (Some)
Encapsulation + Inheritance

面向组件程序设计 = 多态性 + （完全）迟绑定 + （完全，也是强制性的）封装性 +
接口继承 + 二进制重用性
Component-Oriented Programming = Polymorphism + (Really) Late Binding +
(Real, Enforced) Encapsulation + Interface Inheritance
+ Binary Reuse

对我来说，在任何一种情况下，所有的争论都是非常有趣的。经常出现在 comp.object 和 comp.object.corba 的面向对象纯粹主义者会不屑地指着 COM 说“它并不是真正的面向对象”。你可以用以下两种方法来反驳这种观点：

1. “它当然是面向对象的。只不过你对面向对象的定义是错误的。”

2. “那又怎么样？COM 在商业上获得了巨大的成功，它使成千上万的独立软件开发人员通过插入、集成的手段建立起大型的软件系统。而且他们还赚到了钱，赚到

了不少钱。¹ 他们编写的软件组件在销售、在使用、在重复使用。这不正是任何一种技术的真正要点吗？除此之外，我还可以反驳说唯有 COM 才是真正面向组件的。² 确实如此！”

简单性产生可塑性

malleable 可塑的。形容词。它的意思是：

1. 能够在重击或者重压之下改变形状或者结构，譬如 a malleable metal（可塑金属）。
2. 容易被控制，或者受影响；易控制的（tractable）。
3. 能够调整自身以适应环境的变化；易适应的（adaptable）。譬如，the malleable mind of the pragmatist（实用主义者灵活的头脑）。³

COM 的第一个真正应用是作为 Microsoft 的复合文档结构第二个版本，也就是 OLE（Object Linking & Embedding，对象链接和嵌入）2.0 的底层基础。如果你能够考虑到今天 COM 的众多应用方式，那么你一定能够明白我所说的“COM 具有可塑性”的含义。开发人员利用 COM 为自己的应用提供插件结构、建立大规模商务交易的多层客户/服务器结构、在 Web 页面中提供点击服务、控制和监视制造过程，甚至通过远程控制望远镜阵列以跟踪间谍卫星。

这样的可塑性来自于这样的事实：COM 的设计者使它的模型核心尽可能的简单。可说明这一点的一个证据是 COM 程序设计相对比较“蹩脚”。C 或者 C++ 开发人员必须要处理各种“黏性物质”，包括 GUID（全局唯一标识符）和引用计数。然而所有的复杂性都可以融入到 COM 中，并达到简化的目的。COM 设计师并不把重点放在让这个模型运转起来，他们知道只要他们成功了，那么各种支持工具就会随之而来。最近问世的快速易用 COM 工具使这样的假设得到了证实，这些工具包括 Visual Basic、Visual C++ 中的 COM 支持，以及 ATL（Active Template Library，活动模板库）。当你阅读这篇序文的时候，Microsoft 已经宣布，未来计划通过一个公共的运行时程序库 COM+（所有的工具都可使用）来简化 COM 开发。

大众趣闻

任何一项像 COM 这样被广泛使用的技术一定会有许多趣闻。纯粹出于好玩，这里列出一些趣闻，你可能没有听到过它们，但有一些确实是真的。

¹ Giga（受 Microsoft 的委托）的研究报告表明，在 1997 年基于 COM 的商业组件有 4.1 亿美元的市场。据估计，到 2001 年，这个市场将增加到 28 亿美元。这些数字并没有包括 Microsoft 自己的产品。

² 给读者一个练习：请你说出除了 COM 之外的、可供商业使用的对象系统，要求这个系统提供二进制重用模型、支持稳定的版本控制策略、位置透明性以及编程语言独立性。如果你回答说 CORBA，那么你已经受骗了，只不过你还不知道而已。

³ 摘自《The American Heritage Dictionary of the English Language, Third Edition》（1996 年由 Houghton Mifflin 公司出版）。电子版本由 INSO Corporation 授权；进一步的复制和发行应遵循美国的版权法。

- Microsoft 有许多开发组的人都对 COM 的设计工作做出了贡献，但是 COM 最主要的设计师则是 Bob Atkinson、Tony Williams 和 Craig Wittenberg。他们三人仍然在 Microsoft 工作。
- Bob、Tony 和 Craig 属于一个跨部门的小组，这个小组专门致力于提交各种有关“比尔·盖茨的 IAYF (Information at Your Fingertips, 信息尽在掌握)”远景成真的核心技术。⁴ 虽然这三个人对 COM 未来的威力非常清楚，但是他们却被束缚在“尽快推出 COM 的实用产品 OLE 2.0”上。这也说明了为什么经历如此长时间才形成 COM 文档。抱歉。
- COM 的第一个实现随 1993 年 5 月 OLE 2.0 的发布而问世。
- 根接口（当时并不被称作 IUnknown）有一个方法为 GetClassID。这个方法被移到 IPersist 接口中了，这也正好说明了“保持 COM 模型尽可能简单”的原则。
- 在当时，IUnknown 并没有 AddRef 方法。但是很快 COM 设计者看清楚了，不允许使用者复制接口指针将带来很大的限制。
- IUnknown 中的“Unkown”源自 Tony Williams 在 1988 年 12 月撰写的 Microsoft 内部报告，其标题为“Object Architecture: Dealing with the Unknown – or – Type Safety in a Dynamically Extensible Class Library”。
- 1991 年的前两个月才作出了“使用 RPC 作为进程间远程通信机制”的决定。Bob Atkinson 写了一份备忘录，名为“IAYF Requirements for RPC”，记下了所有的要求，并交给 RPC 小组，该小组后来被改名为“IAYF 小组”。这个小组负责建立起使“比尔·盖茨的 IAYF”远景成真的基础。这个基础就是 COM（只不过当时不是这个叫法）。
- 名字对象（moniker）的功能比你想像的要强大得多。
- 有些人把 COM 称作“Common Object Model”，这是 Mark Ryland 的错。他深感懊悔，并愿意致以最诚挚的歉意。
- “MEOW”并不真的代表 Microsoft Extensible Object Wire。这是 Rick 虚构出来的。
- Windows NT 3.5 包括第一个 32 位 COM 和 OLE 版本。有人意外地在某个（根）头文件中留下了“#pragma optimization off”。真是的！

如果有一本关于 COM、DCOM、OLE 或者 ActiveX 的（英文）书，并且我也读过的话，那么你几乎总是可以在技术评论者一栏中找到我的名字，以此作为该书的信誉保证。我自己也写了许多有关这些技术的文章，并且我也是 COM 规范的主要编辑。我给技术人员和非技术人员作过无数的关于 COM 的报告。很显然，我花了大量的时间和精力来找出表达 COM 的最佳方法。

⁴ 你可以回顾一下比尔·盖茨于 1990 年在 Comdex 大会上关于 IAYF 的演讲。

当我读完这本书的草稿之后，我明白我的所有努力都白费了。没有人能比 Don Box 把 COM 阐释得更清楚。

我希望你能够像我一样享受这本书。

Charlie Kindel
COM 专家，Microsoft 公司
1997 年 9 月

Grady Booch 序

有时候，关于一本书的好处太多了，它就值得说两次。这是为什么 Don 的书有两篇序的原因——它就是这样的一本好书。

如果在 Windows 95 或者 NT 上建立系统，那么你可能躲开 COM。Visual Studio（特别是 Visual Basic）隐藏了 COM 的一些复杂性，但是如果你想真正了解后台所发生的事情，或者充分发掘 COM 的强大功能，那么 Don 的这本书很适合你。

让我特别喜欢这本书的一点是 Don 在说明 COM 时所采用的组织方式，摆在你面前的首先是构建分布式和并发性系统所面临的各种问题，然后向你展示 COM 是如何来处理这些问题的。即使你在读这本书之前对 COM 一无所知，你也会被 COM 中清晰而又简单的概念模型所导引，使你理解问题的症结所在，以及赋予 COM 结构和行为的力量所在。如果你是一位有经验的开发人员，那么你一定特别欣赏 Don 关于“用 COM 来解决常见问题”的各种论述。

对于开发分布式和并发性系统而言，COM 是应用最为广泛的一种对象模型。本书将有助于你利用 COM 成功地建立起这样的系统来。

Grady Booch（译注）

译注：Grady Booch 是世界著名的软件工程大师，UML 语言的创始人之一，现为 Rational 公司首席科学家。

前言

我的工作终于完成了。在终于把被许多人称之为“COM 的口述历史”写成一本书之后，我终于可以休息了。这本书反映了我自己对 Microsoft 于 1993 年展示给程序设计世界的这项恶梦般的技术的理解过程。虽然我没有参加最初的 OLE 专业开发人员研讨会，但是我却感到自己好像一直在从事 COM 有关的工作。四年来一直在与 COM 打交道，使我几乎不能记起在 COM 出现之前的计算时代。然而，我仍然清晰地记得 1994 年初我迁移到 COM 王国时的痛苦。

我几乎花了六个月的时间才对 COM 有所感觉、有所理解。在与 COM 朝夕共处的最初六个月时间里，我能够成功地编写 COM 程序，也总能够解释为什么这些程序能够工作。然而，我对于“为什么 COM 编程模型要被设计成这样”则不能够有整体上的理解。幸运的是，有一天（1994 年 8 月 8 日，在购买了《Inside OLE》一书大约半年之后），我突然有如得到神的启示一般，COM 对我变得再明白不过。这并不是说我理解了所有的接口和 API 函数，而是理解了隐藏在 COM 背后的动机。从这一刻起，“如何把 COM 编程模型应用到每天的编程工作中”对我也变得再明白不过。许多程序开发人员也有类似的体验。当我写这份前言的时候，又有三个 8 月过去了，但是开发人员仍然必须要走过这 6 个月的等待期，才能够成为 COM 社会中具有生产能力的一员。我期望这本书能够缩短这个等待周期，但是我不做任何承诺。

正如这本书所强调的，COM 更像是一门程序设计的学科，而不是一项技术。为此，我并不打算用每一个接口的每一个方法的每一个参数的细节描述来惊吓读者。相反，我尽量提取出 COM 的本质，让 SDK 文档来填补每一章留下的空隙。无论何地，只要有可能，我就尽量指出激发 COM 某一要素的底层张力（动机），而不是提供细节例子来展示如何把每个接口和 API 函数应用到人为构造的例子程序中。我自己的经验表明，只有理解了“为什么（why）”，那么关于“怎么做（how）”的问题自然迎刃而解。相反地，仅仅知道了“怎么做（how）”，则并不能够提供对文档背后内容的足够的洞察力。如果你希望跟上编程模型的演化与发展的话，那么这种洞察力非常重要。

对于建立一个分布式的面向对象系统而言，COM 是一个非常灵活的基础。为了充分发挥 COM 的灵活性，你必须经常超越 SDK 文档、文章或者书籍以外的限制。我个人的建议是，你可以假定所看到的任何事物（包括这本书）可能都是不正确的，或者已经过时了，然后设法形成你自己对编程模型的理解。理解编程模型最可靠的方法是抓住

COM 的基本词汇表，并精通这些词汇的语义。这恐怕只能通过“编写 COM 程序并分析这些程序为什么能够按预定的方式工作”才能够获得。阅读书籍、文章和文档当然有用，但是最终还是要将时间花在对 COM 四个核心概念（接口、类、套间和安全性）的思考上，这样才能使你成为一个有效的开发人员。

为了帮助读者专注于这些核心概念，我尽量包含尽可能多的代码，但是仅限于简单地提供这些源代码，而不作仔细推敲。为了确保 COM 编程技术也能够出现在文字之中，附录 B 包含一个完整的 COM 应用，它也是本书所讨论的众多概念的一个综合例子。而且，本书提供了一些可从网络下载的例子，其中包括一个 COM 工具代码库，这个代码库在我自己的开发工作中非常有用。这个库的某些部分在本书有详细的讨论，但是包含整个库，仅仅是为了演示如何构造标准的 C++ 实现。同时也请注意，每一章出现的许多代码都用到了 C 运行库的宏 `assert` 来强调某些必须满足的前后条件。在软件产品的代码中，这些 `assert` 语句大多应该用更宽大的错误处理代码来取代。

出版书籍的一个缺点是，当这本书到达各地书店时，它往往已经过时了。这本书也不例外。特别是，目前尚未定型的 COM+ 和 Windows NT 5.0（译注 1）必定使本书许多内容不正确或者不完整。我尽力预言 Windows NT 5.0 将会带来的有关模型的变化，然而，在我现在写这份前言的时候，Windows NT 5.0 尚未进入公开的 Beta 测试，所有的信息都有可能发生变化。COM+ 承诺将这个模型进一步发展；然而，我不可能既要把 COM+ 包括进来，同时仍能在今年交出书稿。我鼓励你在拿到了 Windows NT 5.0 和 COM+ 之后，认真地研究它们。

我必须做出一个痛苦的决定，如何在众多的商业库中选出一个库来实现 COM 组件（用 C++ 语言）。在观察了各个 Internet 讨论组上常见的问题之后，我决定放弃 ATL 和 MFC，而把焦点集中在每个开发人员必须要面对的 COM 主题上，不管他们使用了什么样的库都无法回避这些主题。越来越多的开发人员在产生了一大堆 ATL 代码之后，奇怪为什么这些代码不能正常工作。我坚信，一个人不可能在 ATL 和 MFC 的开发过程中学习 COM。这并不是说 ATL 和 MFC 不是开发 COM 组件的有效工具，我的意思是它们不适合用来演示或者学习 COM 编程概念和技术。这对于一本重点集中在 COM 编程模型的书籍来说，ATL 将显得很不合适。幸运的是，大多数开发人员发现，只要理解了 COM，那么短时期内就可以精通 ATL。

最后，每一章开始时的引语使我有机会写出我对书中这一部分的感觉。为了尽可能保持直接的写作风格，我限制这种狂野而跑题的故事在每一章不超过 15 行 C++ 代码。通常，这个引用代码代表了这一章将要面对的概念或者问题在 COM 之前的做法。或许读者根据这些代码可以分析出我根据这些提示写出这一章内容时的心态。

致谢

写作一本书非常艰难，至少对于我是这样的。与我一同受尽煎熬的两个人是我的妻

译注 1：现已改名 Windows 2000。

子 Barbara 和我的儿子 Max（尽管他年纪还很小，但是他喜欢 COM 更甚于其他的对象模型）。我要特别谢谢你们俩，当我努力写作时，能够容忍我的失陪，而且还要容忍我写作时的怪脾气。当本书完成了大部分之后，我的女儿出世了，她很幸运，一出生就有一个好脾气的父亲。同时我也要感谢 DevelopMentor 公司（译注 2）的全体同仁，在我全神贯注于本书写作过程中，你们帮助我完成了本该由我承担的许多事情。

许多关于分布式系统的初步想法早在九十年代初期就形成了，当时我在加州大学 Irvine 分校为 Tatsuya Suda 工作。Tatsuya 教会了我如何写作、如何阅读，如何对付东京不守规矩的火车旅客。非常感谢，也很抱歉。

感谢我的老同事 Doug Schmidt，把我引荐给《C++ Report》杂志的 Stan Lippman（译注 3）。尽管 Stan 拒绝了我的第一篇文章，但我从这里开始了写作生涯。谢谢两位。

感谢 Addison Wesley 公司的 Mike Hendrickson 和 Alan Feuer，是你们启动了这本书的工程。谢谢 Ben Ryan 和 John Wait，感谢你们的耐心。谢谢 Carter Shanklin 坚持到了工程的最后。

感谢《Microsoft Systems Journal》杂志的工作人员，在本书写作期间，你们一再容忍我延期递交文章。特别感谢 Joanne Steinhart、Gretchen Bilson、Dave Edson、Joe Flanigen、Eric Maffei、Michael Longacre、Joshua Trupin、Laura Euler 以及 Joan Levinson。我答应以后再也不会迟交了。

感谢 David Chappell 写了有关 COM 应用最好的一本书。我真诚地建议，每个人都应该买他的书，并且至少读它两遍。

感谢 CORBA 和 Java 的许多民间组织以及爱好者们，你们使我加入到许多 Usenet 新闻组上多年来的白热化论战中。你们的执著使我必须要更好地理解 COM。尽管事实上我仍然感到你们的许多观点有点华而不实，但是我还是尊重你们的愿望。

在过去几年中，Microsoft 有几个人对我的帮助很大，他们直接或者间接地帮助我写这本书。Sara Williams 是我碰到的第一个来自 Microsoft 的 COM 成员，在说明了她自己并不认识 Bill 之后，于是她把我引荐给《Microsoft Systems Journal》的 Gretchen Bilson 和 Eric Maffei，以此作为补偿。Sara 永远是在大屋子里的“福音传教士”，我永远感谢她。Charlie Kindel 为这本书写下了一篇很好的序言，尽管他的工作日程总是排得满满的。Nat Brown 是第一个向我展示“套间（apartment）”的人，他用单词“schwing”无可挽回地玷污了我的词汇表。Kraig Brockschmidt 向我表述了 COM 中某个看起来很精致的方面，其实归根究底只是一个怪异的技巧而已。Dave Reed 把我引见给 Viper，并且每次我到 Redmond 时都认真听取我的想法。Pat Helland 花了 TechEd'97 整整一周的时间来扭转我的想法，并且强迫我重新检查我对 COM 的基本假设。Scott Robinson、Andreas Luther、Markus Horstmann、Mary Kirtland、Rebecca Norlander 以及 Greg Hope 使我走出了黑暗。Ted Hase 帮助我扩展了措辞。Rick Hill 和 Alex Armanasu 在技术前沿为我做了

译注 2: DevelopMentor 公司是 Dom Box 等软件开发大师组建的软件开发咨询公司。

译注 3: Stan Lippman 是名著《C++ Primer》的作者。该书中文版即将由中国电力出版社出版。

大量的支持工作。Microsoft 其他许多人成果也影响到了我的工作，这样的人包括 Tony Williams、Bob Atkinson、Craig Wittenberg、Crispin Goswell、Paul Leach、David Kays、Jim Springfield、Christian Beaumont、Mario Goertzel 和 Michael Montague。

DCOM 邮件列表所讨论的内容是本书最主要的思路来源，也是我灵感的源泉。特别要感谢以下的 DCOM 伙伴：声名狼藉的 Mark Ryland、COM 神童 Mike Nelson、Keith Brown、Tim Ewald、Chris Sells、Saji Abraham、Henk De Koning、Steve Robinson、Anton von Stratten 和 Randy Puttick。

本书的内容主要受到了我在过去几年中在 DevelopMentor 公司教授 COM 课程的影响。同时也得到了学生和其他授课教师的影响。尽管我个人非常希望能够感谢每一位学生（Addison Wesley 限制我的前言不得超过 20 页），不过这里我要感谢现在仍然在 DevelopMentor 的、曾经帮助我提炼出自己对 COM 的理解、并且在“Essential COM”课程中提供了宝贵的反馈信息的各位朋友：Ron Sumida、Fritz Onion、Scott Butler、Owen Tallman、George Shepherd、Ted Pattison、Keith Brown、Tim Ewald 和 Chris Sells。谢谢你们。也要感谢 DevelopMentor 的 Mike Abercrombie 营造了商业领域无法比拟的个人成长环境。

如果不是由于 Terry Kennedy 以及 Software AG 公司的朋友，这本书将可以更早出版。Terry 非常好，他邀请我在休息日帮助在德国的 DCOM/UNIX 方面的工作，我本打算利用这些休息日写作这本书。我无法拒绝 Terry 的邀请（这是我的错，不是 Terry 的错），所以本书晚了一年时间，但是我认为由于在此期间在这个工程上的工作，使这本书更加充实。特别是在与以下人员的协同工作中获得了许多深层的理解：Harald Stidhl、Winnie Froehlich、Volker Denkhaus、Deitmar Gaertner、Jeff Lee、Deiter Kesler、Martin Koch、Blauer Aff、Uli Kaess、Steve Wild 和罪魁祸首 Thomas Volger。

特别要感谢下面的读者，他们在以前的印刷版本中发现了许多错误，他们是：Ted Neff、Dan Moyer、Purush Rudrakshala、Heng de Koneng、Dave Hale、George Reilly、Steve Delassus、Warren Young、Jeff Prosis、Richard Grimes、Barry Klawans、James Bowmer、Stephan Sas、Peter Zaborshi、Christopher L. Akerley、Robert Brooks、Jonathan Pryor、Allen Chambers、Timo Kettunen、Atulx Mohidekar、Chris Hyams、Max Rubinstein、Bradey Honsinger、Sunny Thomas、Gardner von Holt 和 Tony Bervilos。

最后，感谢 Shah Jehan 和可口可乐公司分别为我提供了可口的印度食品和软饮料，使我的工作能不断进行下去。

Don Box
美国加州
Redondo Beach
1997 年 8 月
<http://www.develop.com/dbox>

目 录

译 序

Charlie Kindel 序

Grady Booch 序

前 言

第 1 章 COM 是一个更好的 C++	1
1.1 软件分发和 C++	2
1.2 动态链接和 C++	4
1.3 C++和可移植性	5
1.4 封装性和 C++	7
1.5 把接口从实现中分离出来	10
1.6 抽象基类作为二进制接口	12
1.7 运行时多态性	18
1.8 对象扩展性	20
1.9 资源管理	26
1.10 我们走到哪儿了?	29
第 2 章 接口	30
2.1 再谈接口与实现	30
2.2 IDL	32
2.3 方法和结果	34
2.4 接口和 IDL	36
2.5 IUnknown	39
2.6 资源管理和 IUnknown	44
2.7 类型强制转换和 IUnknown	47
2.8 实现 IUnknown	50
2.9 使用 COM 接口指针	55
2.10 优化 QueryInterface	57

2.11	数据类型.....	60
2.12	IDL 属性和 COM 属性.....	72
2.13	异常.....	73
2.14	我们走到哪儿了?.....	78
第 3 章	类.....	79
3.1	再谈接口与实现.....	79
3.2	类对象.....	81
3.3	激活.....	84
3.4	使用 SCM.....	86
3.5	类和服务器.....	89
3.6	一般化(generalization).....	96
3.7	优化(Optimization).....	99
3.8	再论接口与实现.....	105
3.9	名字对象和组合.....	110
3.10	名字对象和永久性.....	113
3.11	服务器生命周期.....	116
3.12	类和 IDL.....	118
3.13	类模仿(class emulation).....	121
3.14	组件类别.....	123
3.15	我们走到哪儿了?.....	128
第 4 章	对象.....	129
4.1	再谈 IUnknown.....	130
4.2	QueryInterface 是对称的.....	131
4.3	QueryInterface 是可传递的.....	132
4.4	QueryInterface 是自反的.....	134
4.5	对象具有静态类型.....	136
4.6	唯一性和对象实体身份.....	137
4.7	QueryInterface 和 IUnknown.....	138
4.8	多重接口和方法名字.....	141
4.9	动态复合.....	149
4.10	二进制复合.....	155
4.11	包容.....	165
4.12	我们走到哪儿了?.....	166
第 5 章	套间.....	167
5.1	再谈接口和实现.....	167