高级.NET开发系列

PEARSON Addison Wesley

Essential .NET

Essential .NET
Volume 1: The Common Language Runtime

.NET本质论
第1卷：公共语言运行库

[美] Don Box, Chris Sells 著
张晓坤 译

- 畅销书《COM本质论》作者Don Box最新作品
- 微软公司CLR首席程序经理大力推荐
- 帮助读者快速完整地理解CLR工作机制

Pearson Education

中国电力出版社
www.infopower.com.cn

高级.NET开发系列

Essential .NET
Volume 1: The Common Language Runtime

.NET本质论
第1卷：公共语言运行库

[美] Don Box, Chris Sells 著
张晓坤 译

中国电力出版社
www.infopower.com.cn

Essential .NET Volume 1: The Common Language Runtime (ISBN 0-201-73411-7)
Don Box, Chris Sells

图书在版编目（CIP）数据
.Net 本质论 第 1 卷：公共语言运行库 /（美）伯克斯，（美）赛欧司编著；张晓坤译.
—北京：中国电力出版社，2004
（高级.NET 开发系列）
ISBN 7-5083-2177-4
Ⅰ.N... Ⅱ.①伯...②赛...③张... Ⅲ.计算机网络－程序设计 Ⅳ.TP393
中国版本图书馆 CIP 数据核字（2004）第 014329 号

丛 书 名：高级.NET开发系列
书　　名：.NET 本质论　第 1 卷：公共语言运行库
编　　著：(美) Don Box , Chris Sells
翻　　译：张晓坤
责任编辑：朱恩从
出版发行：中国电力出版社
　　地址：北京市三里河路6号　　邮政编码：100044
　　电话：（010）88515918　　传　　真：（010）88518169
印　　刷：汇鑫印务有限公司
开　　本：787×1092 1/16　　印　　张：25.5　　字　　数：647 千字
版　　次：2004年4月北京第1版　　2004年4月第1次印刷
定　　价：48.00 元

译者序

第一次看到这本书的时候，就感觉到了这本书的分量。

Don Box（Don Box 的个人网站已经更改，变成了 Blog 的形式；参见 http://www.gotdotnet.com/team/dbox/）在电脑界具有“鬼才”之称，发表过大量的技术文章。他的技术水准无可挑剔。个人觉得，有时候他写的东西很难读懂。但即使是这样，这本书的内容还是值得我们反复阅读的。读过《Essential COM》、《Essential XML》、《Effective COM》、《Creating Components DCOM C++》等书的读者，对于 Don Box 对这些底层技术的精彩描述，想必一定是深有体会的。

有人说，Microsoft 的.NET 技术只是实现了跨语言，而没有做到真正的跨平台，这一点不如 Java。就目前的情形而言似乎是这样。也有人说，CLR（公共语言运行库）作为.NET 的核心部分，尽管 Microsoft 已经将其称为标准，但时下只有 Windows 系列的操作系统上可以跑 CLR。相比之下，JVM（Java 虚拟机）则与众多的操作系统兼容。Microsoft 曾经表示过乐意资助或支持任何交叉平台的.NET，但是对于 GPL， Microsoft 是持谨慎态度的。事实上，在 2001 年有一个小组启动了一个名为“Mono”的计划，致力于实现 Linux/Unix 下的.NET。这个小组目前已经扩充到了 50 多人，他们中有 5 人是 Ximian（Ximian 是一家专门从事为一般计算机用户改进 Linux 操作系统的公司，Ximian 现在已经并入 Novell，详情参见 http://www.ximian.com）的全职雇员，他们正在做着一件极其了不起的事情，你可以在 http://www.go-mono.org 上看到他们工作的最新进展。不过，Microsoft 对于跨平台的应用也有着一套自己的解决之道，那就是.NET Web Services。本书的重点是 CLR，对于.NET Web Services，Don Box 在本书的前言中说准备近期推出本书的姊妹篇，我们将拭目以待。

COM（组件对象模型）技术真的会消失吗？Don Box 曾经谈到了 COM 之后的世界（参见 http://www.csdn.net/develop/Read_Article.asp?Id=12824）。

问：Is COM Dead?

Don Box:Yes:OLE32.DLL。

关于“COM is Dead”的说法，在 2000 年的 PDC 开发者大会上，Microsoft 就曾宣称令人头疼的 COM 编程模型将彻底消失，IUnknown 等接口将不再出现，取而代之的将是 CLR。不过，要知道 COM 是 Windows 的基础，CLR 和 COM 技术之间有着极深的渊源。应该说 COM 的精神并没有消失，而是有了非常大的进步，本书即将揭开这个谜团，并一步一步带领您揭开 CLR 的神秘面纱，在这里 Don Box 将会告诉你，CLR 其实是一个更好的 COM。

CLR 最令开发人员激动的是，它推出了一个比它的前身 COM 更为简洁的开发模型，你不再需要掌握 GUIDs、IUnknown、AddRef、Release 就能进行组件开发！在.NET 中语言的差异似乎不再那么明显，因为所有.NET 语言共享相同的类库；令人厌烦的内存管理，也将被 GC（自动垃圾回收）所取代；在访问代码和资源时，CLR 实施了代码访问安全策略，这种安全性深深地植根于 CLR 之中，这种对操作系统级的安全性的扩展，可以说是一种很大的进步。

本书由 10 章组成，探讨了 CLR 的里里外外，涵盖了基本类型、实例、方法调用和消息、AppDomain、安全，以及 CLR 外部世界。

本书不仅讨论了类型是怎样映射到 CLR 的，还讨论了这些类型在运行时的行为；然后讲述了类型、对象、值之间的联系及它们之间的相互作用；接下来的内容是消息（高级方法，.Net Remoting 的基础）以及方法调用；再有就是关于 AppDomain 的讨论（我理解为 CLR 下的“进程”）；最后，以 CLR 安全模型和 CLR 外部世界结尾。这些技术很艰深，但对于一名渴望深入了解 CLR 底层运作机制的爱好者来说，这绝对是一本让人大呼过瘾的好书，因为其字里行间都是极具价值的技术信息，例如，CLR 中的类型是如何初始化的，is 和 as 操作符（对应于 isinst）与 C 风格的强制类型转换（对应于 castclass）之间的差别，对象在内存中的数据结构和对象的生存期，以及避免程序性能下降的技巧，如在方法调用时如何正确处理多个预处理（pre-process）和后处理（post-process）过程的方法，这些技术对于创建高效的应用程序是很有用的。

严格地说，本书并不是写给初学者的，与 Jeffrey Richter 的《Applied Microsoft® .Net Framework Programming》相比，它更偏向于底层机制，如果你有 C++和 COM 的背景，那么就会更深刻地体会到这一点，因为在书中你会看到很多 C++以及 COM 的痕迹。本书中描述的技术并不限于某种特定

语言，Don Box 试图做到与语言无关，但很多例子都是用 C#写的。作为一本解释 why（相对于那些 How to）的书，这本书和 Don Box 的其他书一样，描述详尽，充满了专家级的指导。在翻译过程中最深的体会是，这不是一本看完了就会放在一边的书，因为你可能需要花很多时间来读这本书才能真正明白 CLR 的思想。当你完成了这项艰难的阅读后，你会发现这是一本值得收藏的经典之作！

为了尽可能地照顾不同技术水平的读者，我增加了多处译注，希望将 Don Box 对.NET 深层次的理解诠释得更清楚。在整个翻译过程中，我总是试着站在读者的角度换位地思考——“这句话这么说，读者会理解吗？”尽管这样，恐怕还是有不尽人意的地方。为了亡羊补牢，以及更好地支持这本书的阅读，我将通过网站 http://www.coreader.com，将本书的勘误整理出来放在网上，同时与大家一起共同探讨技术方面的问题，并提供一些本书相关的源代码。

感谢通常是一篇序变长的原因，因为一本好书的出版，里面必然蕴含了很多人的心血，首先要感谢的当然是 Don Box 和 Chris Sells，他们为我们的书架又增添了一本经典！翻译的过程，是顿悟的过程，也是自我提高的过程。和我以前的翻译工作一样，在本书的翻译过程中得到了许多朋友的支持。谭立平是我近期工作的合作伙伴，他贡献了本书部分章节的翻译与校对，并和我一起编写测试代码，验证作者的理论。卢青锋也是我的合作伙伴之一，他为 http://www.coreader.com 网站的开发做了大量的工作。本书在 china-pub 上引起争议之后（参见 http://www.china-pub.com/computers/common/info.asp?id=14700），我的老朋友汤涛为本书的前四章提出了修改意见；还有我从未见过面的南京大学数学系学生刘未鹏，他对序言、第 1 章等的修改意见是建设性的。同以往一样，我与中国电力出版社的合作非常愉快，我的老朋友高军自然就不用说了。谢工、关敏对这本书的进度与质量提出了严格的要求，并给予了我极大的鼓舞。朱恩从近乎“吹毛求疵”的编辑，毙掉了不少 Bug。此外，与 china-pub 上众多网友进行的诚挚而热烈的讨论使我受益良多，让我更深刻地体会到强烈的责任感。

虽然，我在翻译的过程中已十分小心，但仍然可能会有所疏漏，敬请各位谅解并予以指正。

张晓坤

2004 年元月于北京中关村

序

发生了什么？

在 1998 年，Microsoft 公司在圣地亚哥（San Diego）举办了专业开发人员大会（Professional Developer's Conference，PDC）[1]。COM 的杰出人物 Charlie Kindel 在大会上预言："将不再有 GUID，不再有 HRESULT，不再有 IUnknown。"他和 Mary Kirtland 进一步展示了 CLR 的基本架构，也就是 COM+运行时（COM+ Runtime）。在会议的后半段，Nat Brown 和 David Stutz 演示了使用 Visual Basic 和 Java 进行跨语言的继承。参会者还领到了有关这个非常奇妙演示的 CD，里面包含了该编译器的原始版本。他们可以回到家中，重新进行这个演示。在 2002 年的 2 月，这项技术终于得以正式发布。

Microsoft 平台的演变可以用两个时间进行划分。1993 年 7 月 27 日，Windows NT 3.1 的发布标志着 DOS 时代的终结；2002 年 2 月 13 日，公共语言运行库（Common Language Runtime，CLR），作为.NET Framework 的一部分进行发布，标志着 COM 时代的终结。

.NET Framework[2]是一个用于软件集成的平台。从根本上讲，.NET Framework 提供了两个核心的集成技术：一个是 CLR，它被用来将软件集成到一个单独的操作系统进程中[3]；另一个则是 XML Web Services，它被用

---

1 专业开发人员大会是 Microsoft 专门为那些利用最新的 Internet 科技，使用 C/C++或 Java 编写分布式企业级应用程序，或为 intranet、局域网、广域网，以及 Internet 创建用户定制组件的开发人员而筹备的。你可以在以下 Web 站点上找到有关于 PDC 的更详细内容，其地址为 www.microsoft.com/pdc/。在通常的情况下，参加 PDC 的人数大约为 7 000 名。——译者注

2 .NET Framework 是用于生成、部署和运行应用程序和 XML Web services 的多语言环境。它由三个主要部分组成：公共语言运行库、.NET Framework 类库（统一的编程类库，或者说 API）及 ASP.NET。——译者注

3 CLR 可被视为在运行时管理代码（这些代码必须是以 CLR 为目标的托管代码，它们可由不用的编程语言生成）的代理，它提供核心服务（如内存管理、线程管理和远程处理），强制实施严格的类型安全，以及实施可确保安全性和可靠性的其他形式的代码准确性。事实上，代码管理的概念是运行库的基本原则。而 CLR 本身可由非托管组件承载，例如，Windows 操作系统、IIS、SQL Server 等等，这些都是 CLR 的宿主。——译者注

来在 Internet 范围内进行软件集成[4]。它们都是基于相同的思路，也就是强类型的协定（strongly typed contract）和封装。不过从根本上讲，它们属于不同的技术，可以相互独立地应用。你可以选择 XML Web Services，而不是 CLR（事实上，许多 Web Services 产品就已经这样做了）；当然，你也可以在没有 XML Web Services 的情况下采用 CLR，这样能够更好地利用 CLR 的相关特征，例如，代码访问的安全性，或者高级的内存管理的实用部件。不管怎么说，CLR 和 XML Web Services 都将成为 Microsoft 开发平台的中心，相关开发经验的积累只是时间问题。

CLR 和 XML Web Services 都强调组件之间的强类型协定。这两项技术都需要开发者根据类型定义或者协定描述组件之间的相互关系。在这两种技术中，贯穿在这些协定应用的关键概念是：元数据（metadata）和虚拟化（virtualization）。

CLR 和 XML Web Services 都依赖高保真的（high-fidelity）[5]、无所不在的以及可扩展的元数据，通过这些元数据传达程序员的意图。元数据向将要使用 CLR 组件或者 XML Web Services 的开发人员传达了的基本结构和类型关系。

同样重要的是，无所不在的元数据把组件开发者在编写代码时的意图通知给相应的工具和底层平台。

同组件的完全不透明相比，元数据指引的“洞察力”允许平台为其提供更加丰富的支持[6]。例如，XML 序列化器[7]将捕获元数据中对象到 XML 映射的不同方面。至于 XML 的表示方式，开发人员更倾向于通过声明性元数据

4 XML Web Services 是提供特定功能元素（如应用程序逻辑）的可编程实体，任何数量的、可能是完全不同的系统都可以用常见的 Internet 标准（如 XML 和 HTTP）访问它。XML Web Services 在很大程度上依赖于对 XML 和其他 Internet 标准的广泛接受，而事实上，目前有许多厂商都提供了支持它的平台，例如，IBM 的 WebSphere 和 Sun 的 SunOne。因此，与 CLR 相比，XML Web Services 更强调松耦合、通用的数据格式等。——译者注

5 所谓高保真，是指元数据能够如实地向使用组件或者 Web Services 的开发人员传达构建它们的开发人员的意图，以及向底层平台或者工具如实地反映构建时的意图。——译者注

6 元数据以非特定语言的方式描述在组件代码中定义的每一个类型和成员。使用元数据，平台可以访问加载代码并将其处理为本机指令所需的所有信息。基于这种方式，元数据使自描述文件、公共类型系统和跨语言继承成为可能，因此，不再需要接口定义语言（IDL）文件、头文件或任何外部组件引用方法。——译者注

7 XML 序列化将对象的公共字段和属性或者方法的参数和返回值转换（序列化）为符合特定 XML 架构定义语言（XSD）文档的 XML 流。XML 序列化中最主要的类是 XmlSerializer 类。——译者注

扩展，而不是显式地编写代码[8]。

体现 CLR 和 XML Web Services 协定的第二个关键思路就是虚拟化的概念。这两个技术都强调将语义部分与物理实现的细节分离开来。特别是这两个技术的元数据作用于抽象结构层，而不是依据底层数据的表示和实现技术。当开发者在这个“虚拟”层指定了组件之间的协定，底层平台就能够以最有效的方式传递协定后。例如，通过以某个抽象数据模型表示——Web Services 的协定，通道（plumbing）既可以采用高效的二进制数据表示方式，以提高性能；也可以采用基于文本的 XML 1.0 表示方式，以最大限度地提高协作性[9]。

因为协定是虚拟化的，所以，协定的具体细节可以依据后开发的特性，在运行时绑定。

由于本卷[10]完全集中在 CLR 上，因此，CLR 的工作定义是有序的。从本质上讲，CLR 就是一个加载器，它将你的组件载入到一个操作系统进程中。CLR 代替 COM 的 `CoCreateInstance` 和 Win32 的 `LoadLibrary`，成为代码的主要加载器。

CLR 作为代码的加载器，与前任 COM 和 Win32 相比，它提供了更多的服务。CLR 加载器具有版本感知（version-aware）的特征，它能够提供版本策略和代码存放的灵活配置[11]；CLR 加载器具有安全感知的特征，它是执行安全策略的重要部分；CLR 加载器具有类型感知的特性，对于显式的管理和独立于编程语言的类型创建，它能够提供更丰富的运行环境。总之，CLR 加载器是一种高级的组件技术，将取代 COM，成为 Microsoft 主要的主存内集成策略。

编译器将生成 CLR 的新的文件格式[12]，这样 CLR 就可以懂得许多编程语言。注重编程语言的人将 CLR 看成是为编译器作者提供了关键的生成程

---

8 你可以创建用属性批注的类，或者使用 XML 架构定义工具生成基于现有 XML 架构的类。——译者注

9 这实际上就是.NET 远程处理的框架。对于通道（或者信道），既可以是 HTTP 信道，也可以是 TCP 信道。——译者注

10 动态绑定的协定具体细节。——译者注

11 在.NET 中部署代码，不必涉及注册表（这个过程是带有风险性的）。你只要简单地使用 XCOPY，再修改相应的配置文件就能搞定。——译者注

12 这种文件格式就是改良后的可移植可执行文件（PE 文件），包括三个部分：PE 表头、中间语言和元数据。因此，不管你采用 C#、VB.NET 还是 J#，编译之后的托管代码在 CLR 看来都是一样的。——译者注

序块（key building blocks）技术，而生成程序块降低了编译器实现的复杂度[13]。相比之下，注重系统的人往往将编程语言看成是外在的东西，或者是基于 CLR 底层构件之上的“皮肤”。笔者坚定地倒向后面的阵营。然而，编程语言是必需的透镜，甚至底层系统也要通过它来观察 CLR。正是基于这种观点，本书的示例采用了多种编程语言，理由就是元数据与代码的二进制转储形式（binary dump）[14]才是 CLR 难以理解的地方。

关于本书

我下了很大的功夫来提高本书的可读性，以适合更多的读者。然而，我一贯的写作风格都是简约式的，这使得“Don Box”的书面临着易读性方面的挑战。我的写作经验证明，我不擅长撰写教程或入门级的读物。我能够做好的事情，就是通过书本的形式再现我对世界的看法。所以，你在阅读 Don Box 的书时，需要多读几遍，才能领悟其中的用意。

前一段的意思是本书不适宜作为入门读物。如果你试图通过本书学习.NET Framework 编程，恐怕会让你失望。我推荐你不妨先读一下 Stan Lippman 的《C# Primer》（2002 年由 Addison-Wesley 出版），或者 Jeffery Richter 的《Applied .NET Framework Programming》（2002 年由 Microsoft Press 出版），然后，再来阅读本书。

本系列为两卷：第 1 卷主要阐述公共语言运行库（Common Runtime Language）；第 2 卷将重点讨论 XML Web Services。虽然这两项技术中的许多核心技术都相同，但是，假如将它们纳入一本书中，着实让我为难。

本书是针对 CLR 第 1 版写的。CLR 所用的一些内部技术可能会随着时间而逐步演化，并且事实上可能改动很大。特别是有关虚方法分发的细节就极有可能改变，而这部分内容主要是为了那些想知道 vptr 去向的 COM 开发人员而设的，在本书中也占着相当大的篇幅。也就是说，本书所阐述的基本概念要固定下来，还需要假以时日。

我在整本书的代码中采用了断言，以强化程序的预期状态。在 CLR 中，通过 `System.Diagnotics.Debug.Assert` 执行断言。该方法接收一个布

13 编写编译器时不用考虑各种不同的操作系统，以及特定计算机上的指令系统、内存调度等。——译者注

14 二进制转储形式是指程序编译后的二进制代码。在.NET Framework 中，无论什么编程语言，其编译后生成的二进制代码都是依照相同的标准。——译者注

尔表达式作为参数。如果表达式的运算结果为 false，则断言失败，程序将停止，并发出不同的错误信息。为了提高可读性，书中的所有程序都采用短格式，即使用 Debug.Assert。它假定 System.Diagnostics 命名空间前缀已经被导入。

我认为.NET 并不很在乎具体的编程语言。在我日常基于 CLR 的编程工作中，大约 50%的程序是用 C#写的，40%为 C++，剩下 10%则是 ILASM。本书绝大多数的示例采用 C#编程语言，除了个别概念或技术，需要更为简明的语法表示。虽然有些章节看上去是语言相关的，但实际上没有一章真正是这样的。几乎书中的所有部分都适用于 C++，但出于推广 C#的目的，我还是选择了 C#，从而使本书更容易被读者接受。

本书重点讨论公共语言运行库（CLR），共分为 10 章。

- 第 1 章——CLR 是一个更好的 COM：本章从组件对象模型（Component Object Model，COM）开发人员所面临的问题入手，讲述了 CLR 如何应用虚拟化及无所不在的、可扩展的元数据，解决这些 COM 问题，从而最终取代 COM。

- 第 2 章——组件：从根本上讲，CLR 是 OS 和 COM 加载器的替代品。本章讨论了代码是如何封装和加载的，它们与在 Win32 和 COM 世界中的情形大相径庭。

- 第 3 章——类型基础：组件是代码和构成类型定义的元数据的容器。本章重点讨论了通用类型系统（Common Type System，CTS），包括类型的组成部分，以及类型是如何关联的。这是首次出现源代码块的章节。

- 第 4 章——用类型编程：在编程模型中，CLR 将类型作为第一类概念。本章讨论了 CLR 程序中类型的显式用法，主要是元数据与运行时类型信息的角色。

- 第 5 章——实例：CLR 编程模型是基于类型、对象和值的。第 4 章讨论了类型，本章则重点阐述对象和值。特别是本章概述了这两种实例模型的差别，包括值与对象在内存管理方面的不同。

- 第 6 章——方法：所有组件之间的交互都是在方法调用中呈现出来的。CLR 提供了一系列技术，用于使方法调用成为清楚的行为。本

章将介绍这些技术，从实时编译时的方法初始化开始，直到类型异常时的方法终止。

- 第 7 章——高级方法：CLR 提供了强大的架构用来侦听方法调用。本章剖析了 CLR 侦听的实用部件，以及它对面向方面编程（aspect-orient programming）的支持。这些实用部件是 CLR 比较超前的方面之一[15]。
- 第 8 章——域：CLR 采用 AppDomain（应用程序域）而不是 OS 进程，来划分代码的执行范围。本章从两个方面讨论 AppDomain 的角色，即作为底层 OS 进程的替代品，以及一个 AppDomain 与程序集解析器或加载器之间的交互。熟悉 Java 的读者将发现 Java 加载器与它很相似。
- 第 9 章——安全性：CLR 一个主要优点就是提供安全运行的环境，本章阐述了 CLR 加载器如何支持向代码授予特权，以及这些特权如何被强制实施的。
- 第 10 章——CLR 外部环境：本书前 9 章都是讨论基于 CLR 编程模型的程序设计。最后一章则关注该编程模型对外的方式，以及它如何处理 CLR 外部世界。

致谢

在我的妻子与孩子一如既往的支持下，我顺利地完成了这部书。这里，我要感谢他们在这部书一年多的写作期间所做的一切。

如果没有 Chris Sells 的工作，这部书可能就无法完成。Chris 是我的写作伙伴，并在整个 2001 年的岁月里，无论从精神上，还是专业上，都推动了这部书的撰写进程。尽管 Chris 没有写过一段文字，但他的努力在每一页都卓然可见。

感谢 Microsoft 公司 CLR 小组的 Jim Miller。Jim 慷慨地为本书写了序言，这令我备受鼓舞。不过，更为重要的是，Jim 是 CLI 的 ECMA 规范的首席作者。我认为，这五个部分的 ECMA 规范是关于 CLR 最重要的文献。在.NET Framework SDK 的 Tools Developer's Guide（工具开发者向导）文件夹中，

15 第 7 章介绍的一些内容有些还没有应用到真正的产品中，属于作者的一些推测。——译者注

你可以找到这五个 Word 文档。这些文档揭示了 CLR 的许多内幕。对于 CLR，我相信 ECMA 规范至关重要。

在撰写本书的过程中，我所获得的大多数技术内幕，都是同 DOTNET 邮件列表中的开发人员进行沟通得来的（参见 http://discuss.develop.com）。该列表可以说是活跃于 CLR 编程社区的开发者与架构师的中心。这里特别列出了对我有帮助的人员：Brian Harry、Jim Miller、Dennis Angeline、Steven Pratchner、Brent Rector、John Lam、Mike Woodring、Keith Brown、Peter Drayton、Brad Wilson、Jay Freeman 和 Sam Gentile。

如果没有审稿人，本书的错误肯定更多。这里感谢 Dan Sullivan、Peter Drayton、Mike Woodring、Chris Sells、Stuart Celarier、Jay Freeman、Steve Vinoski、Stan Lippman、Robert Husted、Peter Jones、Mike Giroux、Vishwas Lele、Dan Green、Paul Gunn 和 Fumiaki Yoshimatsu。特别是 Brian Kernighan 对本书早期的审校尤为重要。他的那次审校令我重新评估本书的写作思路，以及如何实现写作意图的方式。如果没有 Brian 的审校，我不能确信自己是否能顺利完成这个过程。谢谢你，Brian。

与以前一样，总有一个“村落”造就了 Don Box。就这部书而言，这个“村落”的成员有：Helga Thomsen、Sandy Deason、Barbara Box、Judith Swerling、David Baum、David Stromberg、Martin Gudgin、Mike Woodring、Fritz Onion、Shannon Ahern Ikeda、Ron Sumida、Tim Ewald 和 Aaron Skonnard。

Don Box
2002 年 2 月
Yarrow Point，WA（华盛顿地区）

前　言

我以前在麻省理工学院（Massachusetts Institute of Technology，MIT）时，曾经为3W协会（World Wide Web Consortium）工作。鉴于COM和CORBA是当时整个网络上最主要的面向对象的框架，于是我们决定同时研究它们。

我打电话给Microsoft公司的代表，希望他能请人为协会成员讲解COM原理。同时我告诉他，我们协会成员的技术水平都很高，比较喜欢刨根问底；我们正致力于制订下一代的Web协议，并且通晓Internet上各种应用协议的细节；此外，我们曾经用C++或Objective C开发过一些非常大型的面向对象程序。

我特别希望他能够推荐出Microsoft公司的专家。然而[1]，得到的答复是“关于COM，没有任何人能阐释得比Don Box更棒！[2]”。这也成了我认识Don的缘由。Don是我所见到的最好的教师：他不仅能够清晰地讲解，很好地辅以实验，其写作能力无可挑剔，而且，他还能够讲解得非常细致透彻，从宏观问题“为什么以那种方式构建这个体系结构？”，到细节问题“那个bit到底是在什么位置?”。就这样，Don教会我（连同其他成千上万的技术人员）什么是COM，为什么要使用COM，COM有什么问题。考虑到我们的实际工作，他还即兴地将COM技术同CORBA技术，以及我们正在设计与构建的Web技术做了令人惊讶的比较。

1　Don Box加盟Microsoft的时间不长，以前他在著名的咨询公司DevelopMentor工作。因此，James S. Miller聘请他时，他还没有成为Microsoft公司的高级架构师。——译者注

2　这句话也是Microsoft公司被誉为“COM guy”的Charlie Kindel对Don Box的评价——参见《Essential COM》（中文版《COM本质论》已由中国电力出版社出版）的序言部分。——译者注

后来，我离开了 MIT 和 3W 协会，加盟了 Microsoft 公司，参与了列为最高机密的名为“COM+服务”的项目。这个项目在我的名片上是这样描述的：“Program Manager，Garbage Collection and Related Rubbish”。所有我们能够公开的（甚至在 Microsoft 公司内部）就是：它带有编程语言和分布式系统，还有一个垃圾回收器。在差不多四年多的时间中，我一直参与并致力于设计和构建现在推出的公共语言运行库（Common Language Runtime），目前仍为这个项目工作。我的主要工作有：IL（中间语言）设计、四种不同的实时编译器（JIT）、元数据、垃圾回收器、执行引擎，以及 ECMA 标准（希望它能尽快成为 ISO 标准）[3]。

我本来自以为能够反过来教 Don 一些东西（我本人的教学水平并不差），但 Don 没有让我如愿以偿。他告诉我读一下这本书，于是我发现，对于我自己的产品，他竟然教了我所不知道的东西！我坚信，你也一定能从中受益匪浅。

Dr. James S. Miller
Microsoft 公司
首席程序经理
公共语言运行库组

3 .NET 已经获得了 ECMA International 和 ISO/IEC 的国际标准认证，它们分别是 **ECMA-334** 标准和 **ISO/IEC 23270** 标准。——译者注

目　录