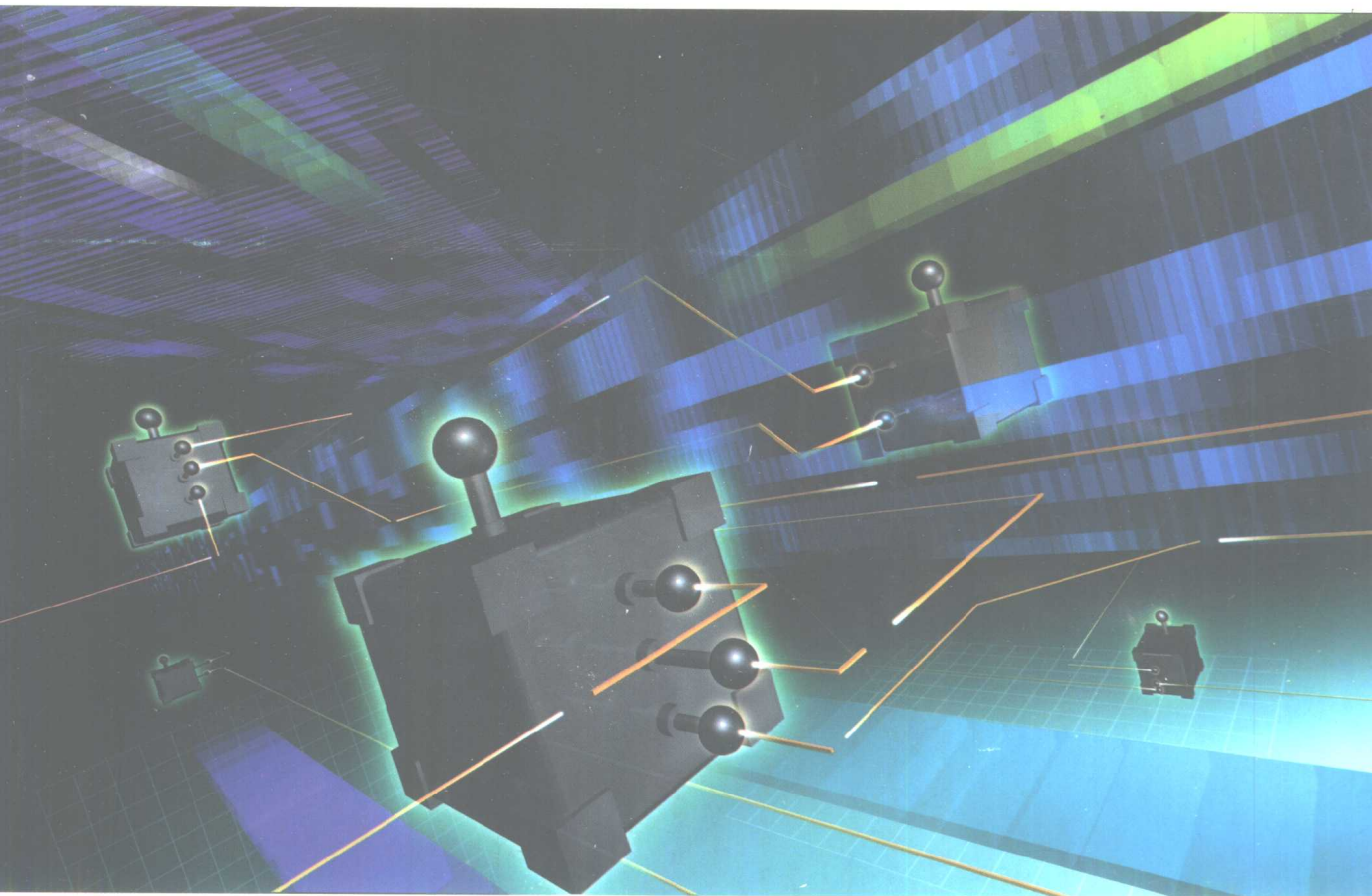


开发人员技术丛书

Microsoft® Press



深入理解 COM+

Microsoft技术
用于企业开发的体系结构



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



[美] David S. Platt 著
潘爱民 译

深入理解 COM+

[美] David S. Platt 著

潘爱民 译

清华大学出版社

(京)新登字 158 号

深入理解 COM +
Understanding COM +
David S. Platt

Copyright ©1999 by Microsoft Corporation.

Original English Language Edition Copyright©1999 by Microsoft Corporation.
Published by arrangement with the original publisher, Microsoft Press,
a division of Microsoft Corporation, Redmond, Washington, U.S.A.

本书中文版由 Microsoft Press 授权清华大学出版社出版。

北京市版权局著作权合同登记号 图字 01-98-0007 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: 深入理解 COM +

译 者: 潘爱民

出 版 者: 清华大学出版社(北京清华大学校内,邮编:100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 丁朝欣

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×960 1/16 印张: 11 字数: 239 千字

版 次: 2000年4月第1版 2000年6月第2次印刷

书 号: ISBN 7-302-00836-1/TP·308

印 数: 5001~10000

定 价: 25.00 元

原著前言

COM 是一项位于底层的有效技术。如果没有一个基于组件的工具或者应用把 COM 带入到我们的开发工作中并实现一个有用方案的话,那么 COM 纯粹是一种摆设,既让人厌烦又缺乏效率。我对大量的基于 COM 的工具或者应用的考察结果是,许多开发人员、用户和管理员从来没有听说过 COM。COM 这种神秘的本质的结果是,许多人对那么多应用中包含 COM 感到非常惊讶。

1992 年,Tony Williams 和 Bob Atkinson(COM 的两位早期设计者)定义了一个简单的结构,从而使得 Microsoft 公司内部和外部的各种组织一次又一次地用它来处理各种各样的问题。复合文档或者对象链接和嵌入? COM。脚本语言支持和自动化? COM。ActiveX 控件和身份鉴定技术(Authenticode)? COM。统一数据访问? COM。Microsoft 事务服务系统(MTS)和 Microsoft 分布式事务协调器(MS DTC)? COM。Microsoft SNA 服务器事务集成器? COM。您看到这种情景了吗?

COM 的灵活性表现在它能够以一致的方式解决广阔的需求,从而奠定了它作为最成功的对象模型的地位。关于面向对象程序设计与组件软件设计哪个更先进的争论也终于结束了。COM 为基于组件的开发方式提供了简单、一致的模型,实践已经证明了这一点。今天,有三百万开发人员使用各种 COM 工具建立 COM 应用,并且有超过 2 亿的 Windows 用户每天在使用这些应用。

这里有必要简单介绍一下 COM 在整个软件领域的地位。Bill Gates 在他最近的新作“Business @ the Speed of Thought”(Warner Books, 1999)中对 CEO 们讲了这样一段话:“在未来的 10 年里,商务应用的变化将比过去 50 年间的变化还要大”;“如何获取、管理和使用信息将会决定您的输赢”。然后,他循序渐进地讲述了实现一个使信息流动起来的数字神经系统的每一个步骤,讨论了 Microsoft Windows DNA 作为应用结构所处的地位。一大堆非常前卫的东西。您可能会问,是什么东西把 Windows DNA 绑到一起了呢?猜一猜。还是 COM。

回到我们的主题上来,COM + 1.0 对 COM 具有非常重要的意义,它也为 COM 增加了许多新的、更为先进的特性。COM + 1.0 在设计的时候,已经充分考虑并增强了用户原先在 COM、MTS 和其他基于 COM 应用上的投资效益。从更深一层来讲,COM + 1.0 中的“+”代表三个基本设计思想。

第一,由于 COM + 1.0 集成了 COM 和 MTS 这两种原先相互分离的技术,所以它既是一个更好的 COM,也是一个更好的 MTS。环境、类属性和截取的概念已经被植入系统内部,所以各种功能强大的服务在实现的时候表现得更为清晰、独立。各种服务之间可以无

缝结合,增强了性能和可靠性。以前有些问题——比如在调试的时候非正常终止——现在都不成问题了。

第二,现有的服务比如事务、安全性、可管理性和代理进程等,不仅由于内部高级的截取结构而变得更好,而且它们的特征更为丰富,在深度和广度上都有所增强。我并不打算在这里列出这些特征;但是我可以这样说,那些已经使用这些服务的 COM 应用可以瞬间变得更加强大、更为灵活。

最后,COM + 1.0 新加入了一些基本的服务,比如用于非连接方式的队列组件、针对松散连接的发布-订阅事件的事件服务、用于高速事务一致数据缓存的内存数据(IMDB),以及用于建立可扩展应用群集系统的组件负载平衡和对象池机制。我们希望这些“新的和增强的”服务使得开发、配置和使用 COM + 应用更加容易。

继续向前看,我们几乎看不到终点。鼓舞人心的是,我们的前进方向是正确的,但我们仍有许多工作要做。COM + 及其相关的技术必须要继续发展,以满足不断变化的市场需求。一些重要的因素,包括从桌面应用和客户-服务器应用向多层 Web 计算的转变,以及简单性、可靠性和应用计算所必需的其他因素,都需要我们不断调整策略。

本书中,David Platt 将用一种友好、清新的风格带着您学习一遍 COM + 1.0,他采用了一致的、自上而下的方法介绍了目标、概念和代码。David 已经从他的“自我提高”中汲取了广泛的经验,在此,他给我们提供了许多关于这种方法的缺点的轶事,以鼓励我们采纳他提出的“我们继续前进,让 COM + 帮助我们”的建议。David 一方面自己学习,同时也在 COM + 产品开发组的支持下对 COM + 进行深入研究。如果管理员和开发人员想理解和实现一个用 COM + 作为基本结构的大型应用,那么这是一本必读的书。

Greg Hope

设计师

Microsoft COM + 开发组

译者前言

在 Windows 平台上有过开发经历的程序员几乎都听说过 COM 和 OLE,也一定听说过 COM+,但了解 COM 和 COM+ 的人不会很多,因为关于这方面的书籍比较少。虽然 COM 已经诞生六年多了,但 COM 技术在国内尚未广为流传,目前大多数的程序员还停留在各种开发工具所提供的封装层面上,往往无法领略底层的细节,所以也不能完全控制自己所开发的产品,我想这也是国内的中间件产品比较少的原因吧。可喜的是,这种情况正在好转,本书就是一个很好的例证。这是第一本全面介绍 COM+ 的专题书籍,而且这本书的出版非常及时,它可以使我们在 COM+ 发布之前就充分了解 COM+ 的全貌,包括 COM+ 的各种特性以及 COM+ 所针对的各种应用模式。

本书向读者全面展示了 COM+ 的所有关键特性,尤为可贵的是,作者并不是简单地按照指导编程的模式介绍这些内容,而是从 COM+ 的各项服务所蕴含的思想出发,深入剖析了这些服务的内在动机以及它们所提供的有效解决方案。本书后五章向读者展示了 COM+ 的五个基本服务,每一章讨论一个系统服务,虽然篇幅不大,但是由于作者采用了“问题背景、解决方案、简单例子、细节讨论”的叙述方式,思路非常清晰,特别适合于那些想要理解 COM+ 本质的应用开发人员。每一章也提供了相应的实例(读者可以从本书的 Web 站点上得到所有例子的源代码),这些实例可以为应用开发人员提供基本的引导,使他们感觉到 COM+ 并不是一个虚的体系结构,而是一个实实在在的产品,是一个即将进入到我们的应用系统中的基础平台。

COM+ 和 COM 的关系很容易让人误解,其实 COM+ 并不是 COM 的更新产品,COM+ 的发布绝不意味着 COM 的结束。反过来,COM+ 和 COM 将共存于操作系统之上,但是 COM+ 更着眼于应用层,它的目标是通过一系列系统服务为企业应用提供通用的基础设施;而 COM 仍然是一个组件对象模型,它为组件对象提供了基本的规范以及相应实现上的支持(这些支持在 COM+ 的版本中有所增强)。所以我们在本书中将会看到许多地方混合使用“COM”和“COM+”两个名称。如果读者能够从层次角度对 COM+ 和 COM 进行区分的话,相信就不会再产生误解。也正因为 COM+ 和 COM 的层次关系以及职能分工、互补关系,所以我建议读者在阅读本书之前能够有一点基本的 COM 知识,至少应该对 COM 组件、COM 对象的含义以及 COM 对象与客户程序之间的调用方式有所了解。目前国内介绍 COM 的书籍并不多,读者可以参考我在不久前写的《COM 原理与应用》(清华大学出版社)或者由清华大学出版社引进 Microsoft Press 版权出版的《COM 技术内幕》。如果有可能的话,读者可以阅读有关 MTS 的资料,目前国内介绍 MTS 的资料并不多,但是 MTS 的许多概念对 COM+ 有直接的指导作用。

在翻译本书的过程中,我曾经在亚马逊网上书店(www.amazon.com)为本书写了一段评论,之后有些人问我,在 COM + 正式发布之前写成的这本书,是否值得买?书中的内容会不会过期?我想读者不必存有这些顾虑,因为作者在书中所写的内容直接来自于 Microsoft 的 COM + 开发组,而且作者也在书中指出了那些可能存在的变化,但这些变化无碍于读者对 COM + 的理解。

本书作者使用了大量的比喻来解释 COM + 的概念,而且 COM + 提出的新概念也确实很多,这两方面的原因使得我在翻译过程中遇到了不少困难,由于译者水平有限,难免有不妥之处,敬请读者谅解。

潘爱民

1999年9月

于北京大学燕北园

致 谢

生命中的每一件事情都是有联系的。你该从何处开始对那些曾经帮助过你的人表示感谢呢？比如，在写作本书过程中帮助过我的那些人——Ben Ryan 是 Microsoft 出版社的选题编辑 (acquisitions editor)，去年秋天我在 WinDev 认识他，他建议我写这本书；Jeff Prosize 是那次我认识 Ben 的外出旅行活动的组织者；Kim Crouse 在三年前阻止我去 WinDev，才促成了我这次的 WinDev 之行。我是该从他们中的哪一位开始谈起呢？还是从第一次向 Kim Crouse 提起我名字的那个人开始谈起？Johnson 女士是我的中学英语教师，是她告诉我一个正确的动词相当于一打形容词，我该从她开始谈起吗？或者我该从我的祖父 Shmuel Platovsky 在 1895 年从俄罗斯迁移到美国来的时候谈起？

我将从 Microsoft 出版社的有关人员说起。Ben Ryan 起着重要的作用，最初我打算写一本基于程序代码的书，类似于我以前写的三本书。开始时他也同意我的想法，但后来他跟我说：“嘿，写一本关于 COM + 上层概念的书，就像 David Chappell 写的关于 ActiveX 概念的书，怎么样？”。下一位应该感谢的是 Mary Barnard，他是这本书的项目编辑 (project editor)，在此之前我还从来没有真正跟编辑合作过。以前我写书全靠自己，从非技术性的校对人员那里我能够得到的帮助很少，顶多只是像“C++ 例子程序包含太多的分号”之类的意见。说实在的，我很担心经过编辑修改过的书是否还能保留我的原义（不管您是喜欢还是不喜欢，毕竟那是我的特色）。幸好可怕的事情没有发生。我认为我的表达能力和文采还是很不错的，至少对于一个技术人员来说足够了。乍一看到初稿中许多地方被编辑改过了，我感觉很是恼火。当我开始阅读修改稿之后，我开始对自己说：“等一等，这正是我想要说的，只是修改过后显得更好。”有时候，我们确实要很费一番心思才能得到正确的短语用法。我认为与编辑合作会有更好的效果，至少从正确性上讲是这样的。我把时间都化在考虑如何用最好的比喻来表达某个技术要点上，而不是主谓一致。如果我强烈坚持我的观点，她就会尊重我的意见。（全书中只有一个例外：我想为书中几个本该是商标但实际上还不是商标的应用名字加一个商标符号；您可以自己猜一猜这是哪几个应用。我认为我现在终于理解了术语“挣脱枷锁”的含义。）Jack Beaudry 作为技术编辑做了很多工作，对例子应用做了测试，并且回答了一些诸如“如何使 ProgID 变成大写？”这样的问题。John Pierce 是本书的管理编辑 (managing editor)，他在 Mary 之后阅读了本书，并提出了一些意见使它变成现在的面貌。在此谨向出版社的计算机工作人员、美术编辑及校对人员致敬。谢谢你们大家，你们的努力使这本书变得更好，这是我所不能及的。我们组成了一个很好的团队，我很乐意有机会再次合作。

能够写是一回事，知道该写些什么是另一回事。在写这本书之前，如果我仅靠自己的

能力就能够领悟 COM + 的话,那么 COM + 将很快就会过时。我从原来的出版社转到 Microsoft 出版社的主要原因在于,我可以有机会与实际设计 COM + 的程序员们坐在一起,听他们给我介绍 COM +。对我来说,这是无价之宝。COM + 开发组的每个人都非常出色,即使在时间安排极为紧张的情况下,他们也会抽出时间给我介绍 COM + 的细节,而且往往重复多次,直到我真正理解为止。在这里第一个要提的是 Mary Kirtland。我有许多问题都直接从她那里得到答案,并且她也为我追踪其他一些她不很清楚的问题。她安排我在 Microsoft 进行为期两天的强化训练,让我向每一个技术人员学习,以使我获取关于 COM + 的每一个细节知识。她为我提供了 COM + 的预发布软件,以便我测试例子程序并把屏幕图截下来。Greg Hope 为我表述了环境的概念以及在 COM + 中环境对对象的意义。他的论断“只有当大家都不知道自己在使用 COM 的时候,COM 才是成功的”至今仍在影响着我。Dick Dievendorf 和 George Chung 放下工作专门陪着我,一直到我掌握了队列组件的基本结构为止。Joe Long 和 Scott Robinson 为我表述了环境的中心思想以及外部联系,Steve Jamieson 为我讲述了补偿资源管理器,Joyce Etheridge 介绍了 COM + 与其他事务系统交互过程中的内部和外部细节。Markus Horstmann 阐述了 COM + 的目录和管理特性(以及为什么要用术语 application 代替 package)。Shawn Woods 介绍了最新修订的事件服务,Senthil Kumar Natarajan 提供了一些关键的示例代码,并在 Don McCrady 和 Radu Palanca 的协助下解释了代码的工作原理。Marc Levy 为我演示了 COM + 安全系统,他对此是如此的熟练,以至于我希望他永远也不要成为网络攻击者。Amit Mital 介绍了 COM + 的负载平衡机制。Henry Lau 和 Jocelyn Garner 介绍了内存数据库系统,而我的科研助手 Paul Holley 则为这一章做了许多很有价值的幕后工作。Richard Ersek、Chris Whitaker、Robert Barnes、Tarun Anand、Emily Kruglick、Phil Garrett 以及 Wolfgang Manousek 帮助 Mary Kirtland 为我找到了许多问题的答案。

最重要的是,我必须感谢我的妻子 Linda。没有她这么长时间来对我的支持,我永远也不可能写成这本书,也不可能做许多其他的事情。

David S. Platt
www.rollthunder.com
Ipswich, Massachusetts
1999.5

目 录

第 1 章 什么是 COM+	1
1.1 企业——最终的应用领域	1
1.2 企业发展的挑战	2
1.3 COM + 基本结构	3
1.4 我的第一个企业系统	6
1.5 关于本书	8
1.6 警告：预发布的软件	9
1.7 继续学习	10
第 2 章 COM+ 系统结构	11
2.1 业务问题背景	11
2.2 解决方案	12
2.3 最简单的 COM + 例子	15
2.4 COM + 组件	21
2.5 COM + 目录	23
2.6 COM + 环境	28
2.7 安全性	30
2.7.1 鉴定	31
2.7.2 授权	33
2.8 服务进程身份	37
2.9 线程套间	40
2.9.1 标准套间	41
2.9.2 同步	45
第 3 章 事务	49
3.1 业务问题背景	49
3.2 方案结构	50
3.3 最简单的事务例子	51
3.4 事务的特征	59

3.5	指定组件使用事务	61
3.6	提交或放弃事务	65
3.7	事务中的及时激活	68
3.8	事务与对象状态	70
3.9	事务与对象池	72
3.10	补偿事务	76
3.11	补偿资源管理器	76
第 4 章	队列组件	83
4.1	业务问题背景	83
4.2	方案结构	85
4.3	最简单的 QC 例子	88
4.4	队列组件的设计	94
4.4.1	接口设计	94
4.4.2	接收输出	96
4.4.3	队列组件中事务的使用	99
4.4.4	队列组件的安全性	103
4.4.5	使用队列名字对象	104
第 5 章	事件	109
5.1	事件问题背景	109
5.2	方案结构	112
5.3	最简单的事件例子程序	115
5.4	关于订阅信息的进一步讨论	120
5.5	关于激发事件的进一步讨论	123
5.6	订阅者对入调用的参数过滤	124
5.7	发布者对出调用的过滤	125
5.8	事件与队列组件	129
5.9	事件与事务	132
5.10	事件安全性	133
第 6 章	内存数据库	135
6.1	数据缓存问题背景	135
6.2	解决方案	136
6.3	最简单的 IMDB 例子程序	138

6.4	IMDB 的事务操作	143
6.5	IMDB 安全性	145
6.6	IMDB 设计考虑	146
6.7	IMDB 的 OLE DB 实现细节	148
6.8	在超大内存模型中使用 IMDB	148
第 7 章	负载均衡	151
7.1	负载均衡问题背景	151
7.2	方案结构	152
7.3	最简单的负载均衡例子程序	154
7.4	负载均衡组件的设计	157
7.5	负载均衡客户的设计	158
7.6	处理路由机失败	158
7.7	将来的发展方向	159

第 1 章 什么是 COM +

1.1 企业——最终的应用领域

个人计算机已经发展到了这样一个阶段：几乎任何个人希望能够完成的事情，计算机都可以为我们去做。记录熟人的联系方式？当然可以。写流水备忘录？非常容易，用一些样式表和向导工具就可以保证自动记录所有的事情，而不必担心会有遗漏。玩单人纸牌游戏？当然可以，而且不断会有新的变化，但是您必须要留意这些变化。不会再有人去开发字处理软件了，因为它已经很完善了。在过去的两年中，MFC (Microsoft Foundation Classes) 没有增加新的特性，这也说明现在的桌面系统已经非常成熟了，所以，加入到桌面环境中的新特性将少而又少。那么在 PC 软件市场中下一个继续保持快速发展的领域在哪里呢？

还记得 Microsoft Windows 的第一、二个版本有将近 5 年的时间不能很好地进入市场吗？实际上 Microsoft 已经把这些软件送到用户手中了，但仍然没有人愿意使用，原因就在于这些软件没有提供足够的有用、方便、省时的特性。后来许多人有了自己的台式 386 PC 机，他们都把这些计算机用作快速 AT 机，CPU 芯片的性能并不能真正发挥出来，直到这个时候 Windows 才开始正式进入软件市场。Windows 3.x 则充分利用了这些未被挖掘出来的资源，从而获得效果极佳的 MS-DOS 多任务环境，在此环境中我们可以玩单人纸牌游戏，而 DOS 下很多其他的东西渐渐地成了历史。整个现代 PC 环境就从这时候开始发展起来了。

在计算机领域，一个类似的革命正在进行之中。Internet 已经迅速席卷了整个世界，差不多每一台 PC 机都在某个时刻出于某种目的连接到 Internet 上。它们或者通过调制解调器连接到 Internet 上收发电子邮件，或者通过专门的高速连接下载彩色电影。然而，即使现在大多数的 PC 机都以某种方式连接到网络中，但它们通常并不在应用层上进行交互。它们使用网络的目的在于发送可供阅读的电子邮件，共享一台打印机，或一个硬盘。因此，PC 机仍然是一座座孤岛，相互之间通过塞在瓶子里的消息联系起来，只不过这些瓶子传递消息的速度很快。

桌面计算机的发展渐趋成熟——相对而言，现在已经很少再有新的关键特征可以加入到桌面型计算机中了。

虽然现在大多数的 PC 机都以某种方式连接到网络中，但它们通常并不在应用层上进行交互。

应用发展的下一个领域是分布式系统,或称企业系统。其中,位于不同机器上的不同程序之间可以进行交互。应用系统开发人员现在只是刚刚开始开发这个领域,而 COM + 正是其核心所在。

企业应用的基本定义包含三个关键点——企业应用是大型的、分布式的,并且是企业的**关键业务**。

企业应用的基本定义包含三个关键点——企业应用是大型的、分布式的,并且是企业的**关键业务**。对于本书,我只坚持前面两点:一个企业应用是大型的系统,至少比单个桌面应用的规模要大;同时也是分布式的,在同一时刻它运行在多台计算机上。企业应用通常也是企业的**关键业务**,因为这些应用的成本往往非常高,一般没有人能够支付得起开发这类应用的费用,除非企业没它不能运作,比如说,航空订票系统。然而,就在我写这些话、您读这些话的时候,这种形势正在发生变化。COM + 预先提供的企业应用基础设施使得企业应用的开发比以前容易很多,因而也比以前节省很多开支,我希望能够早日看到更多这样的企业应用。我认为,一个大型的、分布式的、多人参与的、基于角色的游戏和一个大型的、分布式的、用于北美地区航空管制的应用都属于企业应用,不过后者属于**关键业务**而前者不是(除非您钱多得没地方化了)。最后我再次说明,对于本书目标而言,企业应用仅仅是一个大型的分布式应用。

1.2 企业发展的挑战

以前,开发企业应用是一项长期的、痛苦的、昂贵的、而又不容易成功的任务。

企业应用并不是新生事物。银行、航空公司、保险公司及医院等大型组织中的程序员们已经为企业应用奋斗很多年了。开发企业应用历来是一项长期的、痛苦的、昂贵的、但又不容易成功的任务。究其原因,并不是因为企业应用的业务逻辑非常复杂,真正阻碍应用发展的是,开发人员必须从头写所有的基础设施,从最基本的业务逻辑规则到整个企业应用层。举例来说,一个企业应用要求高度的安全性,而安全性对于桌面应用则是不必要的。企业应用的开发人员必须编写(以及测试、调试、配置、技术支持,并且当雇员被调动工作时,还要负责维护)所有的安全性系统代码,包括如何鉴定用户身份,如何判定一个用户是否允许做这些操作或其他操作,以及管理员用来设置或者删除用户安全许可的管理工具。企业应用还包含许多其他基础设施的挑战,比如资源的池管理、对共享资源的序列化访问、在多服务器和多数据库基础上的原子事务的同步处理,以及把应用系统配置到多个客户机上。编写这些基础代码既耗时,又耗钱(由于不是一次性投入,所以开销显得相对比较低),而且这些代码难以调试,其正确性也不易验证。

这种企业应用的基础设施与银行、医院、航空订票业务或任何一个企业应用的业务逻辑无关。但是,一个企业应用的基础设施与其他企业应用的基础设施非常类似——银行应用的用户身份鉴定系统与医院应用的用户身份鉴定系统非常相似,航空订票应用的线程资源池管理器与超级市场应用的线程资源池管理器完全一致,等等。如果有人一次性编写了所有的企业应用基础设施,那岂不太棒了?一旦写出来之后,我们就可以继承^①我们所关心的部分,而把精力集中在应用本身的业务逻辑上。然后,把这样的基础设施集成到我们所购买的操作系统中。COM + 正是提供了这样的解决方案。

一个企业应用的基础设施与其他企业应用的基础设施非常类似。

1.3 COM+ 基本结构

COM + 其实是一个高级的 COM 运行环境,它预先提供了许多面向企业应用设计者的通用基础设施解决方案,特别是三层结构模型的中间业务层上的基础设施模块。如果您把 COM + 理解为“Microsoft 企业应用预制功能工具包”,那么您也就领会了 COM + 模型的本质。

COM + 其实是一个高级的 COM 运行环境,它预先提供了许多面向企业应用设计者的通用基础设施解决方案。

COM + 也是 Windows 2000 操作系统的一部分。Windows 2000 到底何时正式发布,现在还难以确定。但是可以肯定地说,那必定是在本书被放到书店的书架上之后了。

COM + 从两个方面对 COM 进行了改进。首先,它是 MTS (Microsoft Transaction Server) 的第三个版本,它被集成到原先的 COM 中,并且对 COM 作了全面的更新,形成了一个无缝的整体。MTS 最早发布于 1997 年的春季,它是 Microsoft 把企业应用基础设施做成正式产品的第一个尝试。正如我们将要在第 2 章所讨论的那样,MTS 过去建立在 COM 基础上,形成一个独立封装层。Microsoft 的 MTS 和 COM 开发小组于 1997 年秋季正式合并,COM + 也是他们的第一个集成产品。COM + 对 MTS 的改进和更新表现在以下这些服务上:

COM + 是 MTS 的第三个版本,它被集成到原先的 COM 中,并且对 COM 作了全面的更新,形成了一个无缝的整体。

- 事务服务。事务是保证分布式系统数据完整性的一种机制,特别是在通信失败或者机器故障的情况下。COM + 的事务服务为 COM 对象提供了一种用来创建、使用事务、或者表决 (vote) 事务结果的简单方法。我们将在第 3 章讨论这些服务。

^① 此处“继承”的含义非局限于 OO(面向对象)领域中的概念,而是指更通用的继承概念。

- 安全服务。从本质上讲,所有的企业应用都关心安全问题。它们必须保证只有授权的用户才被允许执行重要的操作。使用 Windows NT 自身提供的安全机制编写安全性代码是一件很可怕的事情,以前是这样,以后仍然会是这样。COM + 的安全服务提供了一种很容易的方法,使我们可以用管理手段指定安全性,而不必编写任何代码。当管理机制难以满足复杂情况的需要时,这些服务也提供了一种用来编写安全代码逻辑的更为简便的方法。安全服务将在第 2 章作进一步的讨论。
- 同步服务。分布式系统的设计人员必须要保护他们的组件程序的关键部分,以便允许多线程并发访问。写这些代码既费时,又费钱,而且很难做到正确无误。COM + 的同步服务提供了一种很容易的方法,使我们可以用管理手段指定组件的同步要求,而不必编写任何代码。当管理机制难以满足复杂情况的需要时,这些服务也提供了一种用来编写同步代码逻辑的更为简便的方法。同步服务也将在第 2 章作进一步的讨论。

COM + 也包含了 4 个新的运行时刻服务,这些服务专门针对企业应用,使得解决某些问题更为容易。

其次,COM + 也包含了 4 个新的运行时刻服务,这些服务专门针对企业应用,使得解决某些问题更为容易,下面逐一说明:

- 队列组件。队列组件是指一种通信机制,它允许 COM 客户对 COM 对象的调用即使当对象所在的服务器并不能通过网络到达的时候也可以进行,比如,在客户机与服务器断开连接操作的过程中。所有的调用被一个系统工具记录下来,然后通过异步协议传递到服务器,当服务器又回到正常运行状态时再通过另一个系统工具将其调用到服务器端的 COM 对象。这种机制使得编写企业应用更加省事,因为不管是客户程序还是服务程序,它们都不需要关心对方的生存周期。我们可以把队列组件当作 COM 调用的电子邮件或者电话应答机。关于队列组件将在第 4 章进行讲述。
- 事件服务。发布者(publisher)是指企业应用中提供信息变化情况的程序,比如股票价格的变化;订阅者(subscriber)是指企业应用中接收这些变化信息的程序。COM + 的事件服务为订阅者提供了一条简单的途径可以进行登记,以便接收来自发布者的变化通知;它也为发布者提供了简单的途径可以确定订阅者的位置并调用到这些订阅者。第 5 章对事件服务进行讨论。
- 内存数据库。内存数据库(IMDB, In - Memory Database)也是一种

服务,它为中间层机器上的后台数据表提供了自动的内存缓存功能。它可以极大地加快对数据表访问的速度,通常这些数据表以读访问为主,而不是以写为主,比如邮购目录。COM + 的 IMDB 服务将在第 6 章进行讲述。

- 负载均衡。在企业应用中,客户程序的负载往往会使得单个服务器超过其承受能力。COM + 的负载均衡服务提供了一种自动机制可以把对象的创建请求分布到一组服务器池中,从而把负载分散开。负载均衡机制将在第 7 章进行讨论。

这就是 COM + 的全貌,那么哪些不属于 COM + 的特性呢?首先,COM + 不是一个根本概念的转变,不像从 MS-DOS 到 Windows,或者 COM 首次出现的时候对软件世界的影响那么大。我们仍然可以使用标准的开发环境编写 COM 类,比如使用 Visual Basic 或者 Visual C++ ;不同的是,这些类现在比以前可以从操作系统获得更多的支持,它们运行起来更快,编写也更为容易。客户程序仍然使用 CreateObject 或者 CoCreateInstance 函数创建 COM 对象,在函数和对象之间包含更多由操作系统提供的预制功能。所以说,COM + 也算是一种概念上的转变,但它更着眼于细微之处。

从某种意义上讲,COM + 实际上是 COM 在 Windows 2000 中的发布版本,只是增加了一些新的特性以便能够解决新的问题。只不过,这些新特性的数量比以前任何时候都要大得多,使 COM 变得更“酷”。如果仍简单地把它叫做 COM,那么我们就无法区分以前的版本和现在的版本,而且我觉得这种区分还是很有必要的。我可以把它叫做“W2KV0C”,虽然取“COM 在 Windows 2000 中的发布版本”的首字母缩写有可能使某些情绪低落的 COM 爱好者振奋起来,从而在寂寞的周末之夜过得更好一些,但那样写起来太长,也不太好听。Microsoft 可以把它叫做“活动服务(Active Service)”,或者其他更长更含糊不清的市场术语。至少“COM +”这个名字只增加了一个字母,而且当我在键盘上输入“COM”时我已经习惯于按下“Shift”键了。

COM + 的这个版本跟 Mary Kirtland 在 1997 年第 11、12 月的 MSJ(Microsoft Systems Journal)杂志上发表的关于 COM + 的内容大相径庭。正如她在 1998 年秋季的 PDC 会议上所说:“那些文章已经过时了……请大家忘了我去年说过的关于 COM + 的所有内容”。后来 Richard Nixon 在 Watergate 期间的新闻秘书(press secretary)Ron Ziegler 也说:“以前的声明现在不再有效了”。那些文章是建立在早期原型系统的基础上的,通常,在从原型到产品的转变过程中,计划也会跟着变。我认为您应该更理性地

COM + 不是一个根本概念的转变。

从某种意义上讲,COM + 实际上是 COM 在 Windows 2000 中的发布版本,只是增加了一些新的特性以便能够解决新的问题。