

实 战 .NET

——使用 C# 开发面向人的软件

Ronan Sorensen

George Shepherd

John Roberts

Russ Williams

著

英 宇 林 琪 侯战友 译

清 华 大 学 出 版 社

北 京

Simplified Chinese edition copyright © 2003 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Applied .NET: Developing People-Oriented Software Using C#

by Ronan Sorensen, George Shepherd, John Roberts, Russ Williams, Copyright © 2002

EISBN: 0-201-73828-7

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macau).

本书中文简体翻译版由 Addison-Wesley 授权给清华大学出版社在中国境内（不包括中国香港、澳门特别行政区）出版发行。

北京市版权局著作权合同登记号 图字 01-2002-4513 号

版权所有，翻印必究。

本书封面贴有 Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

图书在版编目(CIP)数据

实战.NET——使用 C# 开发面向人的软件 / (美) 索伦森等著；英宇，林琪，侯战友译. —北京：清华大学出版社，2003.7

书名原文：Applied .NET: Developing People-Oriented Software Using C#

ISBN 7-302-06431-8

I. 实... II. ①索... ②英... ③林... ④侯... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 018535 号

出版者：清华大学出版社（北京清华大学学研大厦，邮编：100084）

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

<http://www.tup.com.cn>

责任编辑：冯志强

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：24 字数：537 千字

版次：2003 年 7 月第 1 版 2003 年 7 月第 1 次印刷

书号：ISBN 7-302-06431-8/TP · 4846

印数：0001~4000

定价：42.00 元

前 言

就像在海上积聚能量的风暴一样，新的计算时代的第一波浪潮已经开始撞击软件开发的“海岸”。这场风暴背后的力量已酝酿多时，随着它的登陆，有点猝不及防的软件业必须振作起来，并为在这场风暴中生存做好准备。

尽管采用上面的比喻来刻画目前计算机产业的状态有点夸张，但是这种比喻绝对是准确的。因为庞大的开发人员团体，不管他们是集中在一起的，还是分布合作的，还从来没有像现在这样面对如此众多的可用技术。就像一个站在海岸边的人，正面临着一场即将到来的风暴，作为一名开发人员，同样也能意识到软件产业正在经历一场变革。毫无疑问，你一定想努力领会这场变革的全部意义，并为此做出最佳的准备。希望本书能使你对这场变革获得一定程度的理解，帮助你在新的开发时代中生存下来，并能追赶潮流。本书提供的信息有助于读者理解 .NET 技术，它讲述了如何将所有这些技术融合在一起，迅速建立下一代的解决方案。本书涉及到的那些开发思想所使用的都是 C#，并通过几个 .NET 应用的开发来阐述。

《实战 .NET》(*Applied .NET*)从面向人的角度来观察促进软件开发变革的新力量，并为建立高效的 Internet 软件提供了一套开发原则。我们使用面向人(*people oriented*)这一术语来刻画即将到来的新的软件浪潮，因为该术语抓住了激发 .NET 风暴的原动力。该术语的起源要回溯到几年前 Ronan 于 1998 年所写的一本书——《Microsoft Windows NT Internet 开发技术内幕》。该书的第一部分介绍了崭新的面向人的编程模式，以及这类软件所具有的概念。第二部分阐述如何采用早期的技术来开发符合这些原则的系统。

.NET 技术为实现那本书中提出的目标和思想迈出了如此重要的一步，因而我们选择《使用 C#开发面向人的软件》作为本书副标题就是很自然的事情了。希望通过这个标题能够表达出如何采用 .NET 建立一种新的、令人激动的软件的看法。因此，尽管本书使用了 .NET 技术，但是建立这种应用的目标是为了创建更卓越的软件，这些软件被正式称作“面向人的软件”(people-oriented software)。

在《Microsoft Windows NT Internet 开发技术内幕》出版以来的这段时期，它所提出的思想已经趋于成熟，并成为本书作者所进行的各种讨论的焦点。其中一些讨论往往使问题更白热化，但是最后我们一致认为本书列出的原则都是正确的原则，未来的软件肯定将是面向人的。当 Microsoft 最近宣布了它的 HailStorm (冰雹) 计划后，这一观点的论据显得

更加充分。对面向人的观点产生的任何怀疑都随之消失殆尽。一个崭新的时代正在到来，人们不必再面向计算机使用软件，软件将面向人们的实际生活。软件会运行在日常生活中的许多设备上，所有这些设备都会以不可思议的方式连接起来。在它们之中，核心主题是如何让软件嵌入社会，并面向于使用它们的人。

从面向人的角度出发，.NET 是通向最终目标的工具而不是最终的目标。显然除本书之外，会有更多的书详细介绍 .NET 的其他方面，并且为你提供有关方面的更大帮助。本书的重点则是在理论和实践之间寻求一个平衡点，它不但会介绍如何应用 .NET，而且还展示了这种应用能为你带来什么。我们认为《实战 .NET》包含的观点和原则给我们带来的好处远不止这些。本书会实实在在地向你展示如何应用所学的方法来建立真实的 .NET 应用——实践出真知。

致 谢

作者要集体感谢以下人员：

- 感谢 Addison-Wesley 小组：Kristin Erickson, Curt Johnson, Chris Kief, Chanda Leary-Coutu, Marilyn Rash, Cathy Comer, Dianne Wood, Karin Hansen 和 Mark Bergeron 等出版人员的建议，尤其要感谢我们敢作敢为的编辑 Stephane Thomas。
- 感谢 Plural 全体同事的鼓励，特别要感谢 James Watkins 和 Connie Hughes，他们为本书的改进提供了帮助。
- 感谢 Miki Bell，他是本书的美工。
- 感谢 Microsoft 公司的 Sanjay Parthasarathy, Nelson Rossa, Connie Sullivan 和 Rodney Miller 提供的帮助和建议。
- 感谢 Rob Howard, John McGuire, Greg Hack, Daryl Richter, Don Browning, Maxim Loukianov 和 Christophe Nasarre 对本书做了技术审校。

此外，以下是每位作者分别的致谢。

首先，并且是最重要的，要感谢我的妻子 Irene 和三个女儿 Mary、Catherine 和 Sophia 对我写作这本书的支持。在这段时间里，她们知道我希望做什么，并且始终支持我的想法，鼓励我将这些想法写出来，因此我要特别感谢她们。我还要感谢我在爱尔兰、美国和意大利的亲戚，感谢他们给予的信任、希望和爱。最后，要感谢我的合作伙伴 George、John 和 Russ 感谢他们的友谊以及让我有机会与他们合作编写此书。

R.S.

我要特别感谢我的家人 Sandy Daston 和 Ted Shepherd，在我辛苦写作本书时给予我的支持和问候。还要感谢 DevelopMento 为开发人员提供了大量练习和思考的场所。感谢 Patrick Shepherd 所作的大量宣传，他为现代软件筑起了另一面围栏（Remond 所景仰的人是 San Jose，而我则对 Remond 仰慕不已）。最后，要特别感谢 Ronan Sorensen, John Roberts 和 Russ Williams，没有他们的努力，本书是不可能完成的。

G.S.

我想感谢 Pete Nash, Mike Cabrera 和 Jason Cuplin 给予的帮助和支持。感谢本书的合

作者，多年来与他们相识和共同工作使我得到了特别的快乐。感谢 Ronan Sorensen 在最近 4 年为我创造了许多很好的软件开发和出版的机会，当然包括这一次写作机会。感谢 Russ Williams 不断给予我的鼓励和灵感。感谢 George Shepherd 在 20 世纪 90 年代就鼓励我进行写作。与你们一起工作是我从事软件行业以来的重要部分。最后，我要特别感谢我的妻子 Sue，我的孩子 Daniel，Luke 以及 Michael 给予我的支持和理解。

J.R.

我首先要感谢我的家人在我写作本书期间做出的牺牲。因为我抽不出时间与他们共享闲暇时光，因而错过了棒球比赛，无法出席家庭聚会，并且带给他们疲惫的身心 and 心烦意乱的情绪，希望能得到他们的谅解。我要感谢我的女儿 Ryan，Chase 和 McKenzie 对我的理解，尤其是我的妻子 Gina 对我的大力支持，她在关键的时刻给予了我特别需要的鼓励。我有这样的妻子和孩子，是上帝给我的恩赐，我将永远感激不尽。我还要感谢本书的合著者，他们都曾与我共过事，我非常钦佩他们。能与他们合作编著本书是我的荣幸，在此感谢他们对提高本书的质量所做的贡献。尤其要感谢 Ronan，他做出了核心的贡献。正是他的辛勤工作、领导能力、经验、洞察力和潜质造就了你现在拥有的这本书。最后，我要感谢我的父母在这些年来对我的充分信任。在与我父亲的一次偶然交谈中，他讲述了自己对程序设计的偏爱，为我种下了从事软件开发的“种子”。这件事不仅是我职业选择的开始，而且还教育我与孩子们的任何交流都不能当成是小事。爸爸和妈妈，我非常地爱你们。

R.W.

尽管我们已经为避免本书出错付出了极大的努力，但是在印刷后难免会存在一些缺陷。Web 站点 <http://www.people-oriented.net> 会提供报告错误的详细联系方式，而且也会列出所有的勘误或更新列表。

目 录

第 1 章 面向人的软件	1
1.1 面向人的范例	2
1.1.1 通用化	4
1.1.2 协作	4
1.1.3 转换	4
1.2 .NET 方法.....	5
1.2.1 .NET 和通用化.....	5
1.2.2 .NET 和协作.....	7
1.2.3 .NET 和转换.....	10
1.3 小结	11
第 2 章 应用面向人的软件	12
2.1 面向人的设计	12
2.2 概念运用: InternetBaton 应用	15
2.2.1 InternetBaton 应用特征	15
2.2.2 通用化设计: 挖掘运行时	17
2.2.3 协作设计: 指挥乐队	53
2.2.4 转换设计: 语言学家的快慰	64
2.3 小结	71
第 3 章 C#简介	73
3.1 个性化时代	73
3.2 什么是 C#.....	74
3.3 C#有何特殊之处.....	75
3.3.1 当前的看法	75
3.3.2 面向人的角度	78
3.4 语言简介	80

3.4.1	基本概念	80
3.4.2	类型	98
3.4.3	类	107
3.4.4	接口	124
3.4.5	结构	127
3.4.6	枚举	128
3.4.7	属性	129
3.4.8	异常	130
3.5	小结	133
第 4 章	应用 C#	134
4.1	ManagedSynergy 应用	134
4.1.1	目标	134
4.1.2	功能	135
4.1.3	设计	136
4.1.4	实现	138
4.2	小结	174
第 5 章	公共语言运行时	175
5.1	窗口和组件	176
5.1.1	静态库	177
5.1.2	动态链接库	178
5.1.3	COM 的解决之道	183
5.2	公共语言运行时入门	195
5.2.1	普遍类型系统	195
5.2.2	类型是根本	196
5.2.3	公共类型系统	196
5.2.4	公共语言规范	199
5.2.5	装箱	200
5.2.6	类型如何映射到 C#	200
5.2.7	程序集	201
5.2.8	.NET 版本化	204
5.2.9	公共语言运行时中的生命期	204
5.2.10	中间语言和即时编译	205

5.2.11	.NET 垃圾回收.....	205
5.2.12	线程和公共语言运行时	207
5.2.13	应用域	207
5.2.14	互操作性	208
5.3	小结	209
第 6 章	应用运行时	210
6.1	建立程序集和应用	210
6.1.1	命令行	210
6.1.2	makefile 文件	212
6.2	使用 Visual Studio.NET 建立项目	212
6.3	检查清单	213
6.4	部署和版本化	217
6.4.1	全局缓存	220
6.4.2	加载程序集和版本化	221
6.4.3	有关配置文件的更多内容	222
6.5	垃圾回收	223
6.5.1	效果	223
6.5.2	确定性终止化	223
6.6	线程和公共语言运行时	225
6.6.1	创建线程	225
6.6.2	同步	227
6.6.3	方法级锁定	232
6.7	互操作性	233
6.7.1	平台调用	233
6.7.2	与 COM 的互操作	234
6.8	Windows 窗体	240
6.8.1	窗体类	241
6.8.2	处理事件	243
6.8.3	图形与输出	244
6.9	小结	244

第 7 章	ASP.NET 详述	246
7.1	连接性问题	246
7.2	传统 ASP 与 ASP.NET	248
7.3	弱化 ISAPI	248
7.4	ASP.NET: 公共语言运行时的成员	249
7.4.1	System.Web.UI.Page 类	249
7.4.2	System.Web.UI.Page 基础	256
7.4.3	ASP.NET 连接对象模型	257
7.4.4	混合 ASP.NET 和 C#	258
7.4.5	ASP.NET 配置文件	260
7.5	Web 表单	262
7.6	定制服务器端控件	262
7.6.1	扩展浏览器	263
7.6.2	服务器端表现	263
7.6.3	控件的生命周期	264
7.6.4	使用定制服务器端控件的原因	265
7.7	Web 服务和 ASP.NET	265
7.7.1	Web 方法和 ASP.NET	266
7.7.2	服务描述语言和 ASP.NET	266
7.7.3	调用 Web 方法	267
7.8	优化 ASP.NET 缓存	267
7.8.1	输出缓存	267
7.8.2	数据缓存	267
7.9	管理会话状态	268
7.10	小结	269
第 8 章	应用 ASP.NET	270
8.1	用户界面控件和 Web	270
8.1.1	HTML 控件	275
8.1.2	Web 控件	276
8.2	Web 表单和 Visual Studio.NET	279
8.3	Web 应用的状态管理	291
8.3.1	应用状态	291

8.3.2	会话状态	293
8.3.3	会话配置	293
8.4	缓存	295
8.4.1	输出缓存	295
8.4.2	数据缓存	296
8.5	HTTP 处理程序	300
8.6	小结	301
第 9 章	.NET 企业服务器	302
9.1	.NET 企业服务器和面向人的软件	302
9.1.1	通用化	302
9.1.2	协作	302
9.1.3	转换	302
9.2	使之协同工作	303
9.3	解决方案的关键点	304
9.4	.NET 企业服务器和.NET	305
9.5	XML 的角色	305
9.5.1	当前互操作性的基础	305
9.5.2	XML 基础知识	308
9.5.3	处理模型	313
9.6	SOAP 简介	314
9.6.1	描述和用途	314
9.6.2	定义	315
9.6.3	Microsoft 实现	317
9.7	BizTalk Server 核心:解决 EAI 问题及未来	318
9.7.1	BizTalk Orchestration	319
9.7.2	BizTalk Messaging	325
9.7.3	可扩展性框架: BizTalk Hook	330
9.7.4	BizTalk 开发工具	333
9.7.5	BizTalk 管理工具	334
9.7.6	BizTalk Messaging 对象模型	336
9.7.7	解决的问题	338
9.8	Commerce Server 核心	342

9.8.1	Commerce Server 体系结构	342
9.8.2	持续改进周期	343
9.8.3	业务处理流程	344
9.8.4	简表系统	345
9.8.5	目标系统	345
9.8.6	产品目录系统	346
9.8.7	业务分析系统	346
9.8.8	解决方案网站	347
9.9	供应商支持工具包	348
9.10	集成点	349
9.10.1	Internet 信息服务器到 BizTalk Server Orchestration	349
9.10.2	Internet 信息服务器到 BizTalk Server Messaging	349
9.10.3	BizTalk Server 到 Commerce Server	352
9.11	小结	354
第 10 章	应用 .NET 企业服务器：外部销售商的订单履行	356
10.1	订单处理流程	357
10.2	业务过程定义	358
10.3	端口实现	360
10.4	企业到消费者(B2C)网站的集成	367
10.4.1	订单转换为外部销售商格式	367
10.4.2	交付给外部销售商的 BizTalk Server	367
10.4.3	商业网站状态更新和消费者通知	368
10.4.4	外部销售商 BizTalk 处理	369
10.5	小结	370

第 1 章 面向人的软件

Internet 将软件引入到了大众中间。在这个世界上，一般的人都可以通过使用软件来相互联系，这在历史上尚属首次。随着 Internet 互连性扩展到了电视、广播、电话、个人数字助理(personal digital assistant, PDA)技术以及汽车等领域，这一趋势必将延续下去。除此之外，人们的日常生活也逐步成为软件所关注的重点，在此要么直接通过基于 Web 用户界面而实现的人类交互来完成，要么则是间接通过旨在满足人们需要的企业对企业(bussiness-to-business, B2B)的沟通来做到。一方面借助于软件使得互连性日渐增强，再加之软件更加侧重于以人为本，这就带来了软件设计的革新。

以往的软件所关注的是对某些事物操作建模，由此产生了面向对象思想的兴起。尽管在目前，“人”仍然可视作为一个面向对象世界中的对象集合，但是这种方法也许并不实用，而且很可能失败。利用面向对象设计，我们无法对这个社会中的各种动态交互和影响提供一种模糊的方法来建立模型。社会性的交互涉及到诸如自由度的使用、多种文化的引用、移动性、不可预见性以及地理位置等一系列问题，而以上也只是列出了相关问题中的一小部分。简单地说，仅利用一个对象模型的抽象是不足以充分表示一个社会的。人类的真实世界与事物世界大相径庭，正如哲学家 Karl Wojtyla（更为人熟知的是 John Paul II 主教）数年前所指出的：

我们所生活的这个世界由许多对象所组成……作为一个对象，人即为“某人”。这样就使其区别于这个可见世界中的其他实体，而一般来讲，一个对象往往仅作为“某物”。根据这一基本的区别，可以揭示出人类世界与事物世界之间所存在的鸿沟。¹

由于人是类型迥异的对象，其行为就成为了软件开发的核心所在，由此就呈现出一种全新的编程方式，其目标为以一种更为专用的方法来处理人们之间复杂而动态的交互性。

.NET 平台即为这种编程方式的一个早期体现，它是面向人的。要充分了解 .NET 的诸多功能，很有必要对这种编程方式的各个元素加以理解。在人们从 C 语言的面向过程编程转向 C++ 语言的面向对象编程的那段时期，很容易产生以下的错误，即在对于设计用来解决问题的范例转换不甚了解之前，就试图草率地采用这一新工具。有些人错误地将 C++ 视

¹ Karol Wojtyla (John Paul II 主教), 《Love and Responsibility (爱与责任)》。译者: H.T.Willemts (New York: William Collins Sons London 和 Farrar, Straus 及 Giroux, 1981), 21。在《爱与责任》一书中所提出的伦理准则可以普遍应用在 Internet 所产生影响的社会性方面。

为 C 的一种更好版本，而没有将它作为编写软件的一个完全不同的方法。同样地，当前的 .NET 也可能被错误地认为只是一种建立 Web 网站的更好的办法，而不是认为它能为面向下一代 Internet 提供支持技术。为了避免这种误解的产生，这一章余下的内容将对用于构建下一代 Internet 的面向人的编程范例加以探究，并讨论如何利用 .NET 平台实现这种范例。

Internet 将演化成什么，.NET 将如何提供帮助呢？Internet 的成功与人具有社会性这一事实有着不可分割的联系。我们很快就接纳了促进沟通的各种革新，从报纸、广播和电视的发展可见一斑。Internet 所提出的主要需求是人们需要参与到一个群体中——这是一个在线的群体，在规模上体现出全球性。这一全球性的在线群体很自然地划分了大量更小的群体，这些小群体则以更专有的方式面向特定的团体。作为利用诸如 .NET 等技术而发展的下一代 Internet，它将通过经济、社会和文化交互与人做到更为全面也更加无缝的连接。

建立一个足以表示社会的在线群体是一项复杂的任务。尽管在速度和无线方面的显著进步对真正普及 Internet 是必要的，但工程师们面临的最大挑战却是要解决缺乏软件方法论这一问题。需要有更复杂且更强大的软件工具和技术来满足现有的大量工程任务。一旦有了合适的方法论和工具，一个全球性的在线群体即可从数以百万计的各自独立的人们中间脱颖而出。如若没有适当的工具、标准和方法论，那么进展将相当缓慢，而且会走很多弯路。

1.1 面向人的范例

要将万维网（World Wide Web, WWW）转换为一个更具有全球性的互连群体，所需的正确的软件方法论是什么呢？《Microsoft Windows NT Internet 开发技术内幕》²一书的第一部分介绍的一种面向人的编程范例可以解决这个问题。面向人的范例所强调的是通过 Internet 以更直接的方式在人们之间建立联系，另外该范例还强调将软件嵌入到社会运作之中，从而使在线群体得以产生。不同于软件行业以往所出现的诸如面向过程和面向对象编程等范例转换，面向人的编程并不关注于新的编程语言（如 Java）的创建。相反，面向人的编程所强调的是充分利用诸如 Windows Server 等现代操作系统所提供的丰富服务。

² Roman Sorensen, 《Inside Microsoft Windows NT Internet Development》（《Microsoft Windows NT Internet 开发技术内幕》）（Redmond, WA: Microsoft Press, 1998），5，未经授权不得使用。

1999 年 12 月，Pearson PTR、Slashdot.org、Netscape 公司的 DevEdge Online 以及 Doctor Dobb's Journal 向读者做出调查，要求就当今对计算机技术影响最大的书进行提名，也就是那些置于计算机旁并被翻得卷角了的书，而且它们常常可用于解决新的难题，这些书要经得起时间的考验，此外其价值将延续数年。《Microsoft Windows NT Internet 开发技术内幕》一书在 20 世纪的最佳计算机图书竞赛中位居第四。

就社会本身的行为方式而言, Internet 革命代表了一次重大的变革, 由此对于软件的目标来讲, 也势必带来巨大的转变, 这是一种范例的转变, 即将重点从原来强调单个计算任务的技术调整为强调社会交互、文化表述以及信息交换的技术。基本上, 面向 Internet 设计的软件将负责建立一个全球性的群体。它将强调改善一般的生存环境, 使得人们能更有效地完成日常活动。因此将这种新的范例称为面向人的编程(people oriented programming)一点也不为过。尽管 ActiveX 和 Java 为 Internet 的发展做出了卓越的贡献, 它们本身却不足以满足 Internet 时代的需要。为了快速地建立可靠且可扩展的分布式软件解决方案, 我们需要利用 ActiveX 和 Java, 并将它们嵌入到支持 Internet 的系统中。这正是 Windows NT 服务器平台技术所提供的。Windows NT 和 Microsoft Windows 分布式互联网应用体系结构(Microsoft Windows Distributed interNet Applications Architecture, Windows DNA)就提供了用于实现面向人的编程的工具。³

这一指导原则最初在 Windows DNA 中实现, 它在 Microsoft 的下一代 .NET 平台中得到了更全面的表述。.NET 平台包含了 Windows DNA 的丰富的系统服务, 另外还对其作了扩展从而得以创建面向人的 Web 服务, 而这些服务能够以一种精心打造并具个性化的方式应用于 Internet 上。不过 Windows DNA 的重点在于满足类似 Internet 互连性、事务、异步编程、容错性、安全性和可扩展性等基本需求。.NET 平台所解决的是面向人的 Web 服务需求。这些 Web 服务使人们将软件更加无缝地集成到他们的生活中去。例如, 人们将能够以某种标准的方式查看他人的约会日历, 或者是将其客户与供应商的业务处理加以集成。将重点放在人上面, 在最近几年这一点对于软件行业的重要性日渐提高, 而正因如此 .NET 平台应运而生。1999 年 3 月 29 日, Microsoft 宣布其公司已做彻底改造。Microsoft 力图通过卓越的软件提供一个全新而且应用更为广泛的方式来使人的能力得到增强, 而不论何时、何地及在何种设备之上。Bill Gates, Microsoft 的主席和首席软件设计师, 对于该公司更加以人为本的观点做出如下解释:

我们最初提出的“每家每户以及每个桌面上都有一台计算机”的观念仍然相当有意义。预视未来, 这一观念将得到更大的发展。我们将看到一个新的世界, 在这个世界中, 人们可以使用任何计算设备在任何时间、任何地点完成所需做的任何工作。PC 仍将在这一未来世界中起到核心作用, 但同时会有超乎想象的大量数字设备加入进来访问 Internet 的强大功能。我们希望为人们提供这种功能和互连性, 以及选择如何在生活中进行计算的能力。⁴

.NET 平台将有助于创建更加面向人的软件, 这是因为面向人的编程范例直接支持它所体现的 3 个概念的实现。这 3 个概念是 (1) 通用化、(2) 协作和 (3) 转换。

³ Roman Sorensen, 《Inside Microsoft Windows NT Internet Development》, 10.

⁴ Microsoft press release(《Microsoft Announces Reorganization》, 1999): <http://www.microsoft.com/PressPass/features/1999/03-29reorg.asp>.

1.1.1 通用化

通用化(Universalization)是一种开发模型,其依赖于复杂的通用运行时的能力,以实现普遍认可的 Internet 标准。

通用化模型依赖提供系统服务的运行时,这些系统服务可广泛应用于解决复杂的软件工程任务。尽管诸如 C 或 C++ 等编程语言也有着强大的功能,而且也提供运行时,但面向人的编程所提供的运行时在相应功能上则大为胜出。在此并没有把重点放在类似于封装、多态或继承等用于编写可重用代码的编程技术上,面向人的编程所强调的是对普遍存在的运行时所提供的服务进行重用,从而使代码创建工作最小化,同时代码编写也可以有所侧重,即更直接地为构建在线群体而开发面向人的 Web 服务。

编程的重点已由编程语言的内在功能转向了运行时的内在功能。Windows Server 即为实现通用化 Internet 标准的运行时的典型例子,它在其组件对象模型(component object model, COM+)服务层中提供了极为丰富的特性。PocketPC 也是一个实现了通用化 Internet 标准的运行时例子,不过其服务有所缩减,从而适应于其所运行的设备。.NET 运行时则更充分地体现了这一模型,对此将在本章稍后进行解释。

1.1.2 协作

协作(Collaboration)是一种合作模型,其中面向人的 Web 服务要相互合作以提供改进的服务。

协作模型有助于更为复杂的软件能够跨越组织界线实现集成。面向人的 Web 服务可以是任何在 Internet 上提供了可编程接口的应用,这与图形用户接口有所不同,前者的目的是为了开发人员能够构建一个在线群体。在零售业中,此类 Web 服务的例子包括产品和附件目录、结账和支付处理服务以及配送和交付服务等。通过程序将这些 Web 服务捆绑在一起,全球的软件开发人员就可以联合起来创建一个大型“场所”,在此数百万的企业和消费者的需求都可得到识别和匹配。

要完成这一“壮举”,对软件工程来说可是相当艰难的挑战。在此需要利用目录来识别可用的 Web 服务,还需描述开发人员如何使之与其他服务集成。人们需要就 Web 服务描述约定达成共识以实现相同类型的服务,从而避免在集成多个服务提供商时复杂性过大。测试与故障诊断需要为编程人员提供简单易行的方法,从而在开发和操作阶段可以与多个 Web 服务提供商协作。

1.1.3 转换

转换(Translation)是一个互操作性模型,它所解决的是异构平台之间及不同服务描述约

定之间功能的转换。对于在一个存在颇多变化的环境中创建一个虚拟的联合体，转换模型提供了相应的方法。构建一个在线群体需要为数以千计的 Web 服务之间的相互通信提供无缝的方法。这绝不是一件容易的事，因为存在着数百万的软件开发人员使用着各种技术在独立地建立各类服务。Internet 是一个异构的基础设施，它可运行多种不同的操作系统，并可使用不兼容的组件协议，如分布式组件对象模型(distributed component object model, DCOM)、通用对象请求代理结构(Common Object Request Broker Architecture, CORBA)以及远程方法调用(remote method invocation, RMI)等。

将所有这些系统都迁移到一种通用技术并不可行。与此不同，面向人的编程所强调的是转换技术，从而使不同系统能够基于广泛应用的 Internet 标准实现通信，如超文本传输协议(Hypertext Transfer Protocol, HTTP)和可扩展标记语言(eXtensible Markup Language, XML)等。面向人的 Web 服务可以利用任何编程语言在任何平台上构建。在内部，服务可以使用专用协议来获得最大的可扩展性，而外部接口则可以在专用接口和普遍存在的 Internet 标准之间来回转换。转换模型还有另一个同等重要目标。多个 Web 服务提供者很可能提供同一类型的服务，但却使用不同的描述约定来指出如何通过程序来集成此类服务。软件开发人员若要试图适应所有这些不同的服务描述约定，这实在是一项相当复杂的任务。转换模型需要利用有关的工具和技术来实现这种方法，从而将所有这些描述约定都映像为消费系统能够理解的一个通用格式。

1.2 .NET 方法

为帮助软件工程人员建立面向人的系统，从而做到通用化、协作和转换性，.NET 平台在这方面可谓颇有建树的最早版本。尽管也可以使用另外的一些计算技术（如 Sun ONE）而不是 .NET 来实现这些原则，而且将来还将出现一些其他新技术，但由以下分析你可以了解到使用 .NET 平台的原因，下面这些分析的目的就是要提供一个示例，从而指出 .NET 如何有利于面向人的软件开发。本书余下部分则对此进行更为详尽的分析。

1.2.1 NET 和通用化

相较于当前的 Internet，未来的 Internet 将更为普及，同时功能也更加强大，由此人们可以通过自然的接口彼此交互。Internet 将成为一个高带宽的全球性网络，它可以传输数据、声音和视频，并且能够连接遍布于世界各地的计算机、电话、广播、电视、PDA 和汽车等。

无线互连将相当快捷，同时价格也将可以承受，它很有可能取代有线电视而成为 Internet 访问的最主要方式。新型 Internet 设备将纷至沓来，它们可以把诸如电冰箱、门、窗、空调器以及安全系统等家用设施连入到 Internet 中。到时候，可随意使用的 Internet 设