



Fundamentals of Web Applications Using .NET and XML



用 .NET 和 XML 构建 Web 应用程序

ERIC BELL, HAO "HOWARD" FENG
EDWARD L.W. SOONG, DAVID ZHANG 著
SHIJIA "SAM" ZHU

夏 江

译

- 展示 .NET Web 开发的完整过程
- 深入探讨 Web 表单、Web 服务、SOAP 和 XML 技术
- 提供从传统 Web 编程方式向 .NET 开发环境过渡的指导



清华大学出版社

用.NET 和 XML 构建 Web 应用程序

Fundamentals of Web Applications
Using .NET and XML

[美] ERIC BELL

HAO “HOWARD” FENG

EDWARD L. W. SOONG

DAVID ZHANG

SHIJIA “SAM” ZHU

著

夏 江

译

清 华 大 学 出 版 社

北 京

Simplified Chinese edition copyright © 2003 by PEARSON EDUCATION ASIA LIMITED, BEIJING KEHAI TRAINING CENTER TECHNOLOGY LTD. , and TSINGHUA UNIVERSITY PRESS.

FUNDAMENTALS of WEB APPLICATIONS Using .NET and XML by ERIC BELL, HAO "HOWARD" FENG, EDWARD L. W. SOONG, DAVID ZHANG, SHIJIA "SAM" ZHU, Copyright © 2002 ,

EISBN 0-13-041790-4

All rights reserved.

Published by arrangement with the original publisher, Prentice Hall PTR, a Pearson Education Company.

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

本书中文简体字翻译版由北京科海培训中心, 清华大学出版社和 Pearson Education 培生教育出版亚洲有限公司合作出版。

未经出版者书面许可, 不得以任何方式复制或抄袭本书的任何部分。

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。

北京市版权局著作权合同登记号 图字: 01-2003-6247

版权所有, 盗版必究。

图书在版编目 (CIP) 数据

用 .NET 和 XML 构建 Web 应用程序 / (美) 贝尔等著; 夏江译.

—北京: 清华大学出版社, 2003. 8

书名原文: Fundamentals of Web Applications Using .NET and XML

ISBN 7-302-07164-0

I. 用 … II. ①贝…②夏… III. 计算机网络-程序设计 IV.TP393

中国版本图书馆 CIP 数据核字 (2003) 第 077239 号

出 版 者: 清华大学出版社

地 址: 北京清华大学学研大厦

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

邮 编: 100084

社 总 机: 010-62770175

客户服务: 010-62776969

组稿编辑: 成昊

文稿编辑: 何武

封面设计: 杨月静

版式设计: 关静

印 刷 者: 北京市耀华印刷有限公司

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 27.5 字数: 675 千字

版 次: 2003 年 9 月第 1 版 2003 年 9 月第 1 次印刷

书 号: ISBN 7-302-07164-0/TP·5221

印 数: 1~4000

定 价: 43.00 元

内 容 提 要

您想迅速成为微软的.NET 开发人员吗？本书作者采用 step-by-step 的风格，以其前瞻性的产业视角及技术上的专长，借助大量代码示例，向您展示利用.NET 构造 Web 应用程序的全过程，为您铺就一条通往.NET Web 开发的捷径。

书中通过对传统 Web 开发过程和.NET Web 开发过程的全面比较，突出.NET 在创建高级 Web 应用和服务方面的优点，并重点介绍了主要的.NET Web 技术：ASP.NET、ADO.NET、Web 服务和 XML 编程，以及一些先进技术，诸如 SOAP、对象远程部署、系统集成等。最后给出一个实例分析，展示如何利用.NET Web 和 XML 技术创建一种新型的商业模型。

本书面向有经验但尚不熟悉.NET 的程序员，对于具有 VB，C/C++，JScript 和 Java 编程经验的读者，本书也给出了向.NET 技术过渡的实用指导。

前言

一种崭新的、打破商业集成壁垒的计算模型正在悄然形成。在这种模型下，数据可以在因特网上自由交换，应用程序则可以通过因特网无缝地进行协作。基于 XML Web 服务，这一崭新的计算模型将把我们的因特网经济带入一个新纪元。蕴含了一系列新技术的微软 .NET 正是适合于搭建、部署、运营和集成这些 XML Web 服务的平台之一。

作为 Prentice Hall 的 Object Innovations .NET 系列丛书之一，本书面向有经验但尚不熟悉 .NET 的程序员，从实用的角度介绍 .NET Web 技术的基本原理，使他们掌握利用 .NET 编写 Web 应用程序的基础。本书首先回顾现有的 Web 技术；然后介绍 .NET 体系结构；讲述主要的 .NET Web 技术：ASP.NET、ADO.NET、Web 服务和 XML 编程；之后，本书还探讨了一些高级技术，诸如 SOAP、对象远程部署、系统集成以及未来的 Web 技术。最后，通过案例研究展示了如何利用 .NET Web 和 XML 技术创建一种新型的商业模型。

对于具有 VB, C/C++, JScript 和 Java 编程经验，但也许没有 Web 编程经验的读者，本书也给出了向 .NET 技术过渡的实用指导。

组织

本书第一部分首先介绍了 .NET 世界中的 Web 站点（第 1 章），每个读者都应该阅读这一部分，它揭示了由应用和产品到 Web 服务这一崭新概念和实践的模式变化。接下来是对 .NET 框架的介绍（第 2 章），.NET 框架是 .NET 环境中所有应用程序和服务的基本结构。.NET 框架去除了传统软件开发和 Web 开发之间的区别，这是对业界的一大贡献。第 3 章探讨 .NET 框架下的编程环境。在这一章中，你可以找到跨语言互操作的详细介绍和例子。在这一环境中，包括继承和多态等技术在内的面向对象编程打破了语言间的障碍。这一章还将介绍为什么 .NET 框架是开发大规模分布式系统和 Web 应用的最佳选择。

本书的第二部分列举了利用 ASP.NET、ADO.NET、Web 服务和 XML 进行 .NET 编程的主要功能。第 4 章是关于 ASP.NET 的完整综述。ASP.NET 就是基于 .NET 框架的主动服务器页面，对比 ASP，ASP.NET 得益于 Web 表单编程模型以及服务器控件等新功能。通过一个完整的 Web 站点样例，这一章将展示如何搭建和部署 ASP.NET 应用以及如何处理那些真实环境中影响 Web 应用的一些问题，如会话状态以及安全性等。最后，我们简要地介绍一下 ASP.NET 的语法和从 ASP 过渡的途径。第 5 章介绍任何一种软件开发（无论是独立的应用程序或是 Web 应用）都要遇到的任务——访问数据。ADO.NET 是基于 .NET 框架的主动数据对象，提供一种对各种不同的数据源进行一致且可扩展访问的方法。这一功能有助于提高一些设计目标，诸如互操作性、可维护性、可编程性和性能等。第 6 章介绍了 Web 服务这一新的计算模型的核心，包括如何搭建服务，如何由客户端进行访问等等。第 7 章介绍可扩展标记语言（XML）并提供一些样例程序。至此，你应该对使用 .NET 平台进行 Web 应用开发有了相当程度的了解。

本书的第三部分将进一步介绍 .NET 环境中的高级 Web 编程技术，如对象远程部署，简单对象访问协议 (SOAP) 等，同时还将介绍如何和其他平台进行交互。这一部分将向你展示如何基于对象的远程部署构建客户/服务器应用，并进一步构建足够灵活和强大的分布式系统 (第 8 章)。第 9 章是针对那些不想利用 XML，但会直接编写基于 SOAP 的客户端的读者。当然，SOAP 将会帮助你搭建 Windows 和非 Windows 应用间的桥梁，有效地实现跨越整个因特网的应用间通信。是不是微软对开放系统的互操作支持太完美了，以至于你无法相信？第 10 章将向你介绍如何利用 SOAP, UDDI (通用描述、发现和集成) 以及 WSDL (Web 服务描述语言) 等技术实现这一看似不可能的任务。

第 11 章，也就是本书的第四部分，将演示本书的最终目标——使用 .NET 进行 Web 编程。这一章本身是一个案例研究，它将教会你通过实践才能学到的知识。这一章向你展示如何通过因特网使用 .NET 和 XML 以简洁的方式创建基于 Web 的分布式应用。

本书的附录是内容丰富的参考资料。附录 A 是针对没有 Web 编程经验的程序员的 Web 技术入门，它同时也可以为有经验的程序员提供快速参考。附录 B 讨论基于 .NET 框架的 Visual Basic，比较新 VB 与旧 VB 间的差异 (如果了解有关 VB.NET 的更多内容，请参考 “*Introduction to Visual Basic Using .NET and Application Development Using Visual Basic and .NET*”)。事实上，C# 内容 (附录 C) 可以自成一章，但由于本书的主要内容并不是编程语言，所以只将其作为一个附录。(如果了解更多内容，请参考本系列丛书中的 “*Introduction to C# Using .NET and Application Development Using C# and .NET*”。) 你还可以在附录 D 中学习有关 JScript 中的新内容。附录 E 介绍 Visual Studio.NET，微软最流行的 Visual Studio 开发环境的最新版本。新的 Visual Studio 有很多功能，可以使你应用开发更容易更愉快。了解这一环境有助于你的 .NET 编程。

样例程序

学习一种语言的惟一方法就是不断地阅读和编写程序，包括一些相当规模的程序。本书提供许多小程序，它们分别展示了 Web 编程的相关功能，而且非常易于理解。

此外，本书还有一个案例研究，它展示了如何使用 .NET 和 XML 编写 Web 应用。该案例研究对于理解本书的内容尤为重要。

所有样例程序均以自展文件的形式提供。当文件被展开时，在目录 c:\OI\Web 下会创建一个新目录。样例程序分别在 Chap1, Chap2 等目录中。Prentice Hall/Object Innovations .NET 系列丛书中其他书籍的样例程序在 c:\OI 下的其他目录中，所有这些丛书的样例程序均在你所安装的目录下。

Web 站点

Prentice Hall/Object Innovations .NET 系列丛书还有一个 Web 站点：

www.objectinnovations.com/dotnet.htm，该站点提供一个下载本书样例程序的链接。

目 录

第 1 章 .NET 世界的 Web 站点	1
1.1 集成的 Web 站点	1
1.1.1 协作	1
1.1.2 协作的代价	2
1.1.3 体验	2
1.1.4 站点间协作	3
1.1.5 .NET 的解决方案	4
1.2 不同形式的比较	5
1.2.1 传统在线宣传册	5
1.2.2 .NET 在线宣传册	5
1.2.3 传统服务支持网站	6
1.2.4 .NET 的服务支持网站	6
1.2.5 传统在线商店	7
1.2.6 .NET 在线商店	8
1.2.7 传统 Web 应用	9
1.2.8 .NET Web 应用	9
1.2.9 传统服务	9
1.2.10 .NET 服务	10
1.3 体验	10
1.3.1 Web 服务中的体验	10
1.3.2 表层体验	11
1.3.3 向深层体验演化	12
1.3.4 深层体验	12
1.3.5 实现体验	13
1.4 构建和维护 Web 站点	14
1.4.1 页面	15
1.4.2 服务	15
1.4.3 产品	16
1.4.4 利用 .NET 进行构建和维护	16
1.4.5 实现 .NET	17
1.4.6 迁移至 .NET	17
1.5 未回答的问题	18
1.5.1 可用性	19

1.5.2 机密性.....	19
1.5.3 供应.....	20
1.5.4 调解.....	20
1.6 未来 5 年.....	20
小结.....	21
第 2 章 .NET 基础.....	22
2.1 概述.....	22
2.1.1 通用语言运行时.....	22
2.1.2 .NET 框架类库.....	23
2.1.3 应用开发.....	23
2.2 .NET 框架内部.....	24
2.2.1 受控执行进程.....	25
2.2.2 多语言执行环境.....	25
2.2.3 微软中间语言.....	25
2.2.4 JIT 编译.....	26
2.2.5 执行.....	26
2.3 集合.....	27
2.3.1 集合的功能.....	27
2.3.2 集合的优点.....	27
2.3.3 集合的内容.....	28
2.3.4 集合的安全事项.....	30
2.3.5 集合的版本管理.....	30
2.3.6 集合的位置.....	31
2.3.7 比肩执行.....	31
2.4 应用域.....	32
2.4.1 什么是应用域.....	32
2.4.2 应用域和集合.....	33
2.4.3 应用域与线程.....	34
2.5 运行时宿主 (Run-Time Hosts)	34
小结.....	35
第 3 章 .NET 框架的编程环境.....	36
3.1 跨语言互操作性.....	36
3.2 编程环境.....	37
3.2.1 语言支持.....	37
3.2.2 通用类型系统.....	38
3.2.3 元数据系统.....	39
3.2.4 通用语言规范.....	39
3.2.5 调试器.....	39

3.2.6 类	40
3.2.7 类库	41
3.3 能说“Hello!”的控制台程序	42
3.3.1 所需工具	42
3.3.2 所需类和方法	43
3.3.3 C++程序	46
3.3.4 C#程序	48
3.3.5 Visual Basic 程序	49
3.4 说“Hello!”的组件	50
3.4.1 基类	50
3.4.2 C#编写的派生类组件	51
3.4.3 Visual Basic 编写的派生类组件	52
3.4.4 C++编写的派生类组件	52
3.5 组件的客户端程序	53
3.5.1 C#编写的控制台程序	53
3.5.2 Visual Basic 编写的控制台程序	54
3.5.3 C++编写的控制台程序	56
3.5.4 Windows 程序	57
3.5.5 ASP.NET 页面	60
3.6 网络编程	62
3.6.1 请求响应模型	62
3.6.2 TCP 客户	64
3.6.3 套接字编程	66
3.6.4 TCPListener 和服务端编程	67
小结	69
第 4 章 ASP.NET	70
4.1 概述	70
4.2 ASP.NET 特性	72
4.2.1 ASP.NET 与 ASP 代码的对比	72
4.2.2 ASP.NET 和通用语言运行时	75
4.2.3 ASP.NET 的其他特征	77
4.3 Web 表单	78
4.3.1 Web 表单和页面类	79
4.3.2 ASP.NET 页面处理	80
4.3.3 Page 指令和跟踪	87
4.3.4 HttpRequest 和 HttpResponse 类	89
4.4 服务器端控件	94
4.4.1 HTML 控件	95
4.4.2 Validation 控件	99

4.4.3 Web Form 控件	103
4.4.4 Rich 控件	106
4.5 ASP.NET Web 应用程序	107
4.5.1 用 Visual Studio.NET 建立 TAUM 网站	108
4.5.2 ASP.NET 状态基本组成	117
4.5.3 ASP.NET 下的配置	123
4.5.4 ASP.NET 下的应用程序安全性	127
4.6 从 ASP 移植到 ASP.NET	133
4.6.1 移植或不移植	133
4.6.2 发生改变的部分	134
4.6.3 最好通过实践来为移植做准备	137
小结	137
第 5 章 ADO.NET	138
5.1 概述	138
5.1.1 ADO.NET 设计目的	138
5.1.2 ADO.NET 体系结构	139
5.2 简单的例子	140
5.3 ADO.NET 数据提供者	144
5.3.1 SQL Server .NET Data Provider	145
5.3.2 OLE DB .NET Data Provider	145
5.3.3 选择一个 .NET 数据提供者	146
5.3.4 通用模型	146
5.4 使用 .NET 数据提供者访问数据	149
5.4.1 Connection	149
5.4.2 Command	150
5.4.3 DataReader	151
5.4.4 取得单值	152
5.4.5 多个结果集	152
5.4.6 非查询 SQL 语句	153
5.4.7 存储过程和函数	154
5.4.8 事务	157
5.5 DataSet 和 DataAdapter	159
5.5.1 DataSet 构造块	159
5.5.2 从数据库组装 DataSet	160
5.5.3 定义一个新的 DataTable	162
5.5.4 操纵 DataTable	164
5.5.5 DataTable 之间的关系	173
5.5.6 从 DataSet 更新数据库	175
5.6 XML 与 ADO.NET 的集成	179

5.6.1 XML 与 DataSet 之间的数据交换.....	179
5.6.2 DataSet 的模式和 XML	182
5.6.3 Typed DataSet	183
小结	184
第 6 章 Web 服务	185
6.1 定义 Web 服务	185
6.1.1 C#编写的 Hello 服务	185
6.1.2 Visual Basic 编写的 Hello 服务	187
6.1.3 JScript 编写的 Hello 服务	190
6.2 Web 服务客户端	193
6.2.1 为 Web 服务作代理	193
6.2.2 作为 Web 服务客户端的控制台程序	195
6.2.3 作为 Web 服务客户端的 ASP.NET 页面	196
6.3 Web 服务的异步调用	197
6.3.1 同步模式与异步模式	197
6.3.2 异步调用标准方法	197
6.3.3 Web 服务的捷径	201
小结	202
第 7 章 用 .NET 框架进行 XML 编程	203
7.1 访问 XML	203
7.1.1 树——XML DOM——随机层次访问模型	204
7.1.2 指针——读写器——顺序访问模型（单向）	212
7.1.3 检查	221
7.1.4 写 XML 数据	228
7.2 使用关系数据：XmlDataDocument 和 DataSet	228
7.2.1 DataSet 和模式：DataSet.ReadSchema	228
7.2.2 将 XML 映射到表格：DataSet.Tables	232
7.2.3 行中的数据记录：DataSet.Tables[.Rows	234
7.3 转换 XML	235
7.3.1 使用 XML 样式表：Xml.Xsl.XslTransform	235
7.3.2 转换成 XHTML 或者其他 XML	236
7.4 服务 XML	239
7.4.1 在 Web 服务器上进行 XML 服务	240
7.4.2 数据表示	241
7.4.3 数据交换	241
7.4.4 Web 服务和对象远程部署	241
小结	241

第 8 章 对象远程部署	242
8.1 概念	242
8.1.1 应用间通信	242
8.1.2 对象远程部署的构造块	242
8.1.3 服务器对象	243
8.1.4 通道	243
8.1.5 格式化器	243
8.1.6 已知对象的注册	244
8.1.7 远程部署配置	244
8.1.8 激活	244
8.2 HTTP 通道上的对象远程部署	245
8.2.1 已知对象注册服务器	245
8.2.2 使用 Activator.GetObject 的客户端	247
8.2.3 远程注册服务器	248
8.2.4 远程注册客户端	250
8.2.5 客户端的 ASP.NET 页面	251
8.3 TCP 通道上的对象远程部署	252
8.3.1 已知对象注册服务器	253
8.3.2 使用 Activator.GetObject 的客户端	254
8.3.3 远程注册服务器	255
8.3.4 远程注册客户端	255
8.3.5 客户端的 ASP.NET 页面	255
8.4 远程方法的异步调用	255
8.5 部署未实现的服务	258
8.5.1 使用基类	258
8.5.2 使用接口	261
小结	263
第 9 章 SOAP 客户端与 XML	264
9.1 SOAP 概念	264
9.1.1 什么是 SOAP	265
9.1.2 SOAP 消息交换模型与 XML	266
9.1.3 SOAP 封装	266
9.1.4 SOAP 编码	267
9.1.5 SOAP 出错处理	269
9.1.6 在 HTTP 中使用 SOAP 和为 RPC 使用 SOAP	271
9.2 SOAP 的优势	272
9.2.1 SOAP 与 DCOM	273
9.2.2 SOAP 与 CORBA	274
9.2.3 SOAP 与 RMI-IIOP	274

9.2.4 SOAP 的局限性	275
9.2.5 结论	275
9.3 在.NET 下构建简单 SOAP 客户端程序	276
9.3.1 使用 SOAP 客户端程序访问 Web 服务	276
9.3.2 使用 SOAP 客户端的.NET 远程部署	288
小结	291
第 10 章 .NET 平台和其他平台的互操作	292
10.1 WSDL 和.NET	292
10.1.1 WSDL 定义	293
10.1.2 操作和端口类型	296
10.1.3 绑定	297
10.1.4 端口和服务	299
10.2 与.NET 互操作的例子	300
10.2.1 从 Apache SOAP 客户端访问.NET Web 服务	300
10.2.2 从.NET SOAP 客户端访问 Apache Web 服务	309
10.3 通用描述、发现及集成	315
10.3.1 什么是 UDDI	315
10.3.2 使用 SOAP, UDDI 和 WSDL 连接企业	316
10.3.3 UDDI 发展现状	316
小结	317
第 11 章 案例研究：分布式 Web 应用	318
11.1 TAU 商业模型	318
11.2 TAU.NET 系统设计目标	319
11.2.1 子系统	319
11.2.2 信息交换	319
11.2.3 远程服务	319
11.2.4 TAU.NET 节点	320
11.3 体系结构	321
11.3.1 概述	321
11.3.2 数据交换模式	322
11.3.3 TAU.NET 节点接口	328
11.4 TAU.NET 节点适配器	330
11.4.1 TAU.NET 节点适配器链接子系统数据库	330
11.4.2 TAU.NET 节点适配器为网页提供服务	332
小结	334
附录 A Web 编程基础	335
A.1 经典 Web 技术	335
A.2 因特网编程测试环境	343

A.3 微软的 Web 技术	348
A.4 ASP 和 COM	359
小结	362
附录 B VB.NET 的新功能	363
B.1 更强的面向对象特征	363
B.2 更加模块化	372
B.3 更加正式, 减少随意性	375
B.4 更安全, 更强大的性能提高	379
小结	381
附录 C C++或 Java 程序员的 C#	382
C.1 C++程序员的 C#	382
C.2 Java 程序员的 C#	387
小结	395
附录 D JScript.NET 的新功能	396
D.1 为什么使用 JS.NET	396
D.2 编译的 JScript	396
D.3 JS.NET 的两种用法	397
D.4 Visual Studio.NET 中的 JScript 一览	397
D.5 面向对象功能	399
D.6 性能的提高	405
D.7 打包与部署 (EXE, DLL 和打包)	406
D.8 调试	408
D.9 编译器	409
小结	412
附录 E Visual Studio.NET	413
E.1 Visual Studio.NET 概述	413
E.2 工具条	416
E.3 创建控制台应用程序	416
E.4 使用 VS.NET 的文本编辑器	418
E.5 项目配置	419
E.6 调试	422
小结	426

第 1 章 .NET 世界的 Web 站点

如果你想学习如何利用 .NET 构建 Web 站点、控制远程应用，而且想通过例子来了解有关平台的内容，那么，本书就是为你而准备的。.NET 目前正以迅猛的势头涌向你身边的计算机，将来，不可避免地，它会出现在每一台 Windows 计算机中。

为了帮助你开始使用 .NET 来构建 Web 站点，本书会对相关技术进行全面的介绍。但在你开始正式学习之前，本章首先向你展示“.NET 世界”发生了哪些变化，以及为什么会发生这些变化。本章将讨论：

- .NET 的分布式特性
- 目前的 Web 与 .NET Web 间的比较
- 类似于“体验”这样的新概念

Web 站点可以而且将会以与今天不同的方式构建。在开始学习之前，先做一些指导也许会对你有所帮助。如果你想直接了解技术，可以进入第 2 章。

1.1 集成的 Web 站点

在 .NET 出现以前，Web 站点大多是独立操作的。但某些网站不同，通过将其他 Web 站点集成到自己的网站上，它们形成了合作关系。这种现象在门户网站中最为常见，比如 Yahoo (www.yahoo.com) 和 Snap (www.snap.com)，这里不仅有天气预报还有新闻栏目。为了丰富门户的内容，这些门户网站获取许可可以使用或拥有这些服务，也就是成为它们自己的站点。这种行为可以称为集成或协作，或称之为协同工作。这种“分布式计算”具有很多优点，可以将网站的不同部分集成到另一个网站中，使它们一起运作，将不同的页面缝制在一起，而不是简单地给出页面的超链接。让网站在一起工作并且实现这些优点，正是 .NET 的初衷。

但这种类型的协作比想象的要棘手得多。除了将两个或多个网站合在一起的技术细节外，还要考虑很多商业细节。必须筹备并提供服务，而且门户网站必须以特定方式推出这些服务。最后，其表现必须是无缝的。

对于 Web 开发人员而言，将不同网站的页面或部分页面集成在一起并不像日常工作那么简单，但大多数开发者都可以花一定精力做到这一点。不过，集成两个或多个网站并没有一定的框架，也没有标准，没有流行的工具集，更没有一本书籍完整地介绍这一主题。集成多个 Web 站点至今仍是需要自己动手解决的问题。

1.1.1 协作

首先的问题是就协作达成一个可行的定义。字典的定义很简单，“与他人一起工作”。

联网计算机通常都被认为可以使计算机在一起工作，但联网本身并不能达到这样的目标。联网为计算机共享数据提供了途径——这也是它能做的全部工作。在一起工作也就意味着每一台计算机都要为整个会话作出一定贡献，主动地完成它们各自的任務。

协作需要各方主动在一起工作，这意味着协作者以一种分布式的方式一起处理任务。每一台计算机按照所执行的任务提供其他计算机所需的数据。这种计算机间的协作是分布式计算的一种形式。关于分布式计算的理论和实践已有很多书籍，而且超出本书所讨论的范围。通过分布作业在若干计算机上构建 Web 站点或 Web 应用是分布式计算的一种形式。在.NET 出现之前，计算机信息技术界一直缺乏促进这种形式的有效机制。.NET 并不涉及分布式计算的所有形式，但它却非常适合于分布式计算中的一种关键形式——将多个应用有机地缝合在一起构成一个应用。这种分布式计算就是.NET 中的分布式计算，我们将其称为计算机间的协作。

现在我们对协作有了一个可行的定义，但为什么需要这种协作呢？原因就是它具有三点优势。首先，同时使用两台或多台计算机解决问题是更好地使用计算机的方法。这样，就可以充分利用很多计算机的空闲时间了。

其次，决定如何分解一个问题来进行分布式处理是很困难的。在计算机间分解一个问题需要花费一些工作，这样的过程尚不现实。.NET 的方法是重新描述问题，它提供了一个框架，该框架在问题某部分的解决方案出现之前就刺激它们的产生。.NET 用另外一种方式确定如何分解问题。不是空想问题并分解它，而是想象你需要什么，寻找各个解决部分并构成最终的解决方案。这是一种自里向外的思考方式。

第三，外包你的网站的某些部分意味着其他人会为那些部分的维护而操心。这是简单的经营之道。

1.1.2 协作的代价

这种协作的代价是采用.NET。这意味着你将依托这个平台，而且学习使用支持它的工具和服务器。就构成.NET 的所有产品而言，学习任务很艰巨。类库非常庞大，编程语言也是新发明的或为.NET 进行了修改，而且还有新工具需要学习和使用，并且需要集成新的服务器。本书讨论所有这些问题¹。关于服务器，这里仅介绍如何使用它们而不是如何管理它们。比如，ASP.NET 是因特网信息服务器（IIS）和 SQL 服务器（微软的工业级的数据库服务器）所支持的技术，所以本书关于 ASP.NET 的章节将使你能够启动并运行它，而无需考虑管理 IIS 的细节。尽管这种依托是可以预期的，但需要说明的是，由于.NET 的成功取决于人们是否接受这种变化，微软正在努力吸引开发人员使用这个平台，并正在改进.NET 使其更适用于在公司部署，但人力的代价始终都是存在的。

1.1.3 体验

由微软勾勒的“体验(experience)”一词正逐步成为人们的口头禅。许多文章都暗示.NET

1. 关于类库的讨论参见第 2 章和第 3 章。一个类库是一个预先编写好的代码的集合，你必须使用它们来利用.NET。关于 C#，受控 C++，VB.NET 和 JScript.NET，参见附录 B，C 和 D。

将预示着一个软件易用性的新纪元，它将丰富人们的日常生活，带给他们新的体验。

同时，计算机并不能隐藏你必须了解如何操作它们的事实。像朝圣者朝拜圣殿一样，你必须坐在计算机旁，知道如何操作它，并从中获得你所需要的内容。通常，如果有足够的程序，一台计算机可以运行成千上万个程序，你必须一一学习如何操作它们才能从中获益。这种学习内容非常广泛，从不同软件包相似操作的不同命令到管理硬盘所需的操作。计算机需要用户屈从于它们的需求。

人们逐渐需要计算机做更多的工作。在浏览器中保存 Web 页面就是这样的一种需求。首先，你必须选择菜单项来保存页面，其次，你要为页面选择位置而且可能还需要更改页面的名字。这是大多数软件的一种常见行为模式。一旦熟悉了这些，你就接受了这种重复行为并且可以不假思索地执行它。如果要在硬盘上为 Web 页面寻找存放的位置，你可能仅仅需要点击几下鼠标或敲击几下键盘。当你保存若干个文档时，才遇上真正的麻烦，你不得不在每次保存页面时在硬盘中寻找同一个文件夹，程序无法记住你存放上一个文档的位置。这是一次相当难堪的体验。

作为 Web 用户的你对体验的理解非常重要。体验一词——直接观察事件——现已成为术语。微软使用体验来表示“你对事件感觉如何”，这种感觉会因人而异。.NET 的体验对用户与软件交互的方式非常关注。同一软件包由于它的外观、工作原理、操作速度以及其他更多因素的不同，也会带来完全不同的体验。

让个人计算机更易于使用一直是一种推动力量，让使用 Web 站点的细节更不易于察觉的工作也从未中断。最终，我们希望完全不用考虑技术细节。对技术考虑越少，效率就越高。这就是为什么计算机新手往往觉得不自在，甚至是害怕的原因。计算机带给新手们前所未有的体验。计算机新手会花费大量的时间来将他们的需求转换成计算机的需求。这需要大量的工作且是一个容易出错的过程。

用户期望良好的体验。如果你不得不苦思冥想如何使用某个网站，那么你很可能不会使用这个网站。与某个网站进行交互必须相对比较简单。能够提供良好的体验正是 .NET 浪潮的一部分，尽管可能还未发现对此有所支持。至今，.NET 平台尚没有工具、API 或者支持（或实施）体验的编程机制，也没有很多的文献可供参考，但这个词却在很多不同的地方出现，而且会越来越普遍。支持创建、维护和实施体验的工具最终会出现。

1.1.4 站点间协作

为了让 Web 站点间可以协作，必须有适当的标准机制。那些不能彼此协作的设备将使我们的生活变得一团糟，因为没有机制可以使它们在一起工作。一种成功的技术如果能为他人所用，那么它才有生命力，而且，如果有标准可以遵循的话，则会更好。

超文本标记语言（HTML）就是这样的机制，它可以让全世界的每个人都参与 Web 页面的编写。同样，其底层的超文本传输协议（HTTP），也就是 Web 服务器用来通信的协议，也是如此。HTML 主要用来描述如何表达数据，但不提供操作这些数据的方法。比如，货币对你和我可能意味着不同的内容，或者，你我可能按照不同的方式书写日期。渐渐地，像 HTML 一样流行的可扩展标记语言（XML）粉墨登场了。XML 是一种描述数据的通用方法。