GOTOP

北京科海培训中心

DNA & Web数据库应用与剖析

YINGYOUGYUPOUXI

李世杰　编著
刘　云　审校

科学出版社

GOTOP 北京科海培训中心

DNA & Web 数据库应用与剖析

李世杰 编著

刘 云 审校

科学出版社

1999

著作权合同登记号:01-1999-2094

内 容 简 介

数据库在 Web 上的应用与 Microsoft DNA 技术是开发分布式网络应用系统的基础。

本书以循序渐进的方式重点介绍了 DNA 与 Web 数据库结合的技术,内容包括:ASP 的重要概念及其用法示例;DNA 体系结构、功能与技术剖析;ODBC,ADO 结合 Web 数据库的操作重点;MTS 事务处理系统,MSMQ 消息队列系统的开发,并以开发一个多层次的应用系统为实例解释重点,以便读者达到最好的学习效果。

无论你是数据库新手还是网络应用程序高手,本书将是你学习 Web 数据库与 Windows DNA 技术的最佳伴侣!

版 权 声 明

图书在版编目(CIP)数据

DNA & Web 数据库应用与剖析/李世杰编著.

—北京:科学出版社,2000.1

ISBN 7-03-007867-5

Ⅰ.D… Ⅱ.李… Ⅲ.网络-数据库 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(1999)第 73142 号

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

北京市朝阳区科普印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

2000 年 1 月第 一 版 开本:787×1092 1/16

2000 年 1 月第一次印刷 印张:13

印数:1—5 000 字数:316 000

定价:20.00 元

出 版 说 明

大陆与台湾有关术语对照如下表：

台湾用词	大陆用词
物件	对象
元件	组件
伺服器	服务器
交易	事务
架构	体系结构
整合	集成
资料	数据
资讯	信息
讯息	消息
程式	程序
运作	运行
搜寻、质询、窥视	查询、窥视
指标	光标
分页	分批
引数	参数
汇出	导出
佇列	队列
方式	模式
闸道	网关
介面	接口
执行绪	线程
范本	实例
环节	层次
记事本	剪贴板
回应	响应
型态	类型
唯读	只读
位元组	字节
预设	默认
公用识别码	公用密钥
认证	确认

台湾用词	大陆用词
识别字	标识符
拼凑逻辑	哈希算法
布林值	布尔值
愈期时间	超时时间
隐密	保密
记忆体	内存
非同步	异步
事件监测员	事件通知器
取回	检索
开启	打开
侦错	调试
呼叫者	调用者
资料库	数据库
错误例外处理	异常处理
栏位	字段
子程序	存储过程
巢状	嵌套
字元	字符
二元码	二进制
回传	返回
快取	缓存
化名	别名
住址	地址
绕径	寻径
站台	站点
依存用户端	从属用户端
封包	数据包
专案	工程

注：若需要购置本书光盘的读者，请与科海联系：电话 62562449。

目　录

第 1 章　Web 数据库与微软 DNA 技术概述 …… (1)

1.1　什么是 Windows DNA 体系结构 …… (1)
1.1.1　Windows DNA 可以解决的问题 …… (1)
1.1.2　Windows DNA 包含的核心组件 …… (2)
1.1.3　Windows DNA 体系结构的优点 …… (2)
1.1.4　使用 Windows DNA 体系结构的系统有什么特色 …… (2)
1.1.5　系统开发者如何实现 Windows DNA 体系结构 …… (2)
1.1.6　Windows DNA 体系结构集成 COM 技术 …… (3)
1.1.7　Windows DNA 支持的客户端平台 …… (3)
1.1.8　Windows DNA 支持的开发工具 …… (3)
1.1.9　Windows DNA 支持的应用服务器 …… (3)
1.1.10　Windows DNA 支持的后端应用服务器或资源 …… (3)
1.1.11　如何集成 Java 于 Windows DNA 体系结构中 …… (4)
1.2　DNA 系统的结构 …… (4)
1.2.1　两层式主从体系结构 …… (4)
1.2.2　DNA 体系结构 …… (5)
1.3　什么是 COM …… (7)
1.3.1　COM 的发展 …… (7)
1.3.2　COM 是什么 …… (7)
1.3.3　Automation …… (8)
1.3.4　Dual Interface …… (8)
1.3.5　Registry …… (8)
1.4　小　结 …… (8)
1.4.1　准备工作 …… (9)
1.5　思考问题 …… (9)

第 2 章　轻松掌握 Web 数据库技术 …… (10)

2.1　IIS 4.0 揭开了 Web Computing 的序幕 …… (10)
2.1.1　Web Computing 带来的强大效益 …… (10)
2.1.2　Active Server Pages 的使用环境 …… (11)
2.1.3　回顾超文本的历史 …… (11)
2.1.4　Active Server Pages 的运行方式与优点 …… (12)
2.1.5　Active Server Page 概述 …… (13)
2.1.6　6 大对象的简单介绍 …… (13)
2.2　设计 ASP 应用程序的 10 个要点 …… (14)
2.2.1　Server Script& Client Script …… (14)

2.2.2 Get&Post&Form …… (15)
2.2.3 Request …… (15)
2.2.4 Redirect&Buffer …… (16)
2.2.5 ContentType …… (17)
2.2.6 Cookies …… (18)
2.2.7 身份验证 …… (19)
2.2.8 Session&Application&Global.asa …… (20)
2.2.9 CreateObject …… (23)
2.2.10 ASP Function …… (23)
2.3 ODBC 与 SQL …… (24)
2.3.1 ODBC …… (24)
2.3.2 SQL 语言 …… (26)
2.4 关于 ActiveX Data Objects 组件模块的使用 …… (28)
2.4.1 利用 ADO 开发数据库应用程序的优点与特色 …… (29)
2.4.2 ADO 结构及对象 …… (29)
2.4.3 Recordset 对象 …… (31)
2.4.4 Connection 对象 …… (34)
2.4.5 Command 对象 …… (35)
2.5 掌握 Web 数据库 …… (37)
2.5.1 读取数据库表单的最后一项记录 …… (37)
2.5.2 判断有无数据库输出 …… (38)
2.5.3 使用(%)进行查询 …… (38)
2.5.4 返回的记录项数 …… (38)
2.5.5 Timeout …… (39)
2.5.6 数据库打开方式 …… (39)
2.5.7 光标向前或向后移动 …… (40)
2.5.8 分页输出 …… (41)
2.5.9 输出图片数据库 …… (42)
2.5.10 建立自己的 ASP DB Function …… (43)
2.5.11 参数型 Command …… (54)
2.5.12 Stored Procedure …… (57)
2.6 思考问题 …… (63)

第 3 章 MTS 事务服务器 …… (64)

3.1 前言 …… (64)
3.2 两层式主从体系结构 …… (64)
3.3 微软事务服务器为三层式体系结构提供强有力支持 …… (65)
3.4 微软事务服务器的作用 …… (67)
3.4.1 组件的事务处理 …… (67)
3.4.2 事务监控 …… (67)
3.4.3 对象代理 …… (68)
3.4.4 资源回收 …… (69)
3.4.5 系统安全 …… (70)

3.4.6 系统管理 …… (73)
3.4.7 组件开发与客户端程序开发 …… (73)
3.5 MTS 重点实例 …… (73)
3.5.1 开发 MTS 组件或系统的要点 …… (74)
3.5.2 开发 MTS 组件 …… (74)
3.5.3 使用 ASP 启动事务处理 …… (80)
3.6 MTS 最佳范例:银行转账系统 …… (81)
3.6.1 开发 COM 事务组件 …… (82)
3.6.2 创建实验数据库 …… (87)
3.6.3 开发独立客户端应用程序界面 …… (89)
3.6.4 用 ASP 封装 DLL 组件 …… (91)
3.6.5 在 MTS 中创建 JFBank 套件 …… (92)
3.6.6 导出 JFBank 套件 …… (94)
3.6.7 让远程客户使用 MTS 组件 …… (94)
3.7 负载平衡 …… (100)
3.8 MTS 效率最佳化 …… (101)
3.9 开发以 ASP 为基础的事务系统 …… (102)
3.9.1 交易网页:ASPBankTransfer.asp …… (103)
3.10 小结 …… (106)
3.11 思考问题 …… (106)

第 4 章 MSMQ 消息队列服务器 …… (107)

4.1 MSMQ 具备哪些功能 …… (107)
4.1.1 无连接消息缓冲传递 …… (107)
4.1.2 消息处理的优先权控制 …… (107)
4.1.3 保证一次送达 …… (107)
4.1.4 动态队列与智能路由 …… (107)
4.1.5 安全设定与管理 …… (107)
4.1.6 完美又方便的开发环境 …… (108)
4.2 MSMQ 网络拓扑与站点 …… (108)
4.2.1 MSMQ 企业 …… (108)
4.3 关于 MQIS …… (110)
4.4 队列 …… (110)
4.4.1 公用队列 …… (111)
4.4.2 私有队列 …… (111)
4.4.3 日志队列 …… (111)
4.4.4 Dead-Letter 队列 …… (111)
4.4.5 异常 Dead-Letter 队列 …… (111)
4.4.6 管理员队列 …… (111)
4.4.7 报告队列 …… (111)
4.4.8 系统队列 …… (112)
4.5 消息传递 …… (112)
4.6 MSMQ 实例 …… (113)

4.6.1 创建队列 …… (118)
4.6.2 获取公用队列 …… (119)
4.6.3 创建并打开一新的队列 …… (119)
4.6.4 创建一个消息并且把它送到目的队列 …… (120)
4.6.5 送出一消息并要求"确认"响应 …… (121)
4.6.6 送出一消息并要求"消息"响应 …… (123)
4.6.7 发送私有消息 …… (125)
4.6.8 使用内部事务发送消息 …… (125)
4.6.9 使用 MTS 外部事务发送消息 …… (126)
4.6.10 以同步方式读取或查找队列消息 …… (127)
4.6.11 以异步方式读取或查找队列消息 …… (127)
4.6.12 重设与更新队列管理属性 …… (128)
4.7 MSMQ 最佳范例——股票交易系统 …… (129)
4.7.1 配置 MSMQ 网络与服务器 …… (130)
4.7.2 编写股票交易系统服务器端程序 StockServer …… (135)
4.7.3 编写股票交易系统客户端程序 StockClient …… (145)
4.7.4 股市交易中心数据库 StockMarketCenter …… (151)
4.7.5 股票效果测试 …… (152)
4.8 其他内容 …… (155)
4.8.1 可复原传递 …… (155)
4.8.2 安全验证 …… (156)
4.8.3 消息路由 …… (156)
4.9 思考问题 …… (157)

附录 A MSMQ Object Library …… (159)

附录 B MTS Object Library …… (171)

附录 C ActiveX Data Object 2.0 Library …… (174)

附赠 MCSE～IIS 4.0 考前 10 小时 …… (189)

第 1 章　Web 数据库与微软 DNA 技术概述

你知道吗？在未来的信息世界里，除了数据就是数据库，因此数据库管理与应用就成为十分重要的工作，而 Web 的普及产生了跨平台客户端界面浏览器，数据库在 Web 上的应用也就相应地重要起来，这些都是本书中所要介绍的重点。

Windows Distributed interNet Application（简称为 Windows DNA 或 DNA）体系结构，是针对微软在 Windows 平台上如何发展一强健、稳固、扩充性大的分布式网络应用程序系统而推出的方案：DNA 为将来先进分布式系统刻画出每个层次（Tier）位置及重要性，这包括了数据后端、商业逻辑、前端界面...等，此外，DNA 方案亦可以集成多方面在过去推出的重要技术，例如 COM（Component Object Model），配合新的组件服务器，例如 Microsoft Transaction Server（MTS）、Microsoft Message Queue（MSMQ），形成应用程序事务（Transaction）与无连接缓冲通信（Connectionless Communication）的多层次主从式应用系统（N-tiers Client-Server Application）。

信息的洪流就如同洪水猛兽一般，看清楚方向往里头钻才能适应时代的发展，学会 Web 数据库与 Microsoft DNA 技术已成为未来开发分布式应用系统的基础工作，也是致富的捷径：通过本书你可以学会 Web 数据库与 DNA 结合的所有技术，包括：

- **前端服务**：HTML，Scripting，Components，Win32 API
- **应用程序资源**：IIS（ASP），MSMQ，MTS，COM，DCOM
- **数据库统一使用**：ADO，OLE DB，SQL，ODBC
- **系统资源**：安全、管理、网络通信

结合以上技术，开发多层次（N-tiers）应用系统为本书的实例解说重点。笔者深知学习新技术的难度与麻烦，为了使读者达到最好的学习效果，本书将以循序渐进的方式介绍整个结构，而所需的全部背景知识将在第 1 章中介绍。不管你是数据库新手还是网络应用程序老手，本书将是你学习 Web 数据库与 Windows DNA 技术的最佳伴侣！

1.1　什么是 Windows DNA 体系结构

Windows Distributed interNet Application（简称为 Windows DNA 或 DNA）体系结构，是微软在 Windows 平台上针对如何发展一强健、稳固、扩充性大的分布式网络应用程序所推出的方案（Methodology）：DNA 将为未来先进的分布系统刻画出每个层次的位置及重要性，这包括了数据库后端、商业逻辑、前端界面...等，此外，DNA 方案亦可以集成多方面在过去推出的重要技术，例如 COM，配合新的组件服务器，例如 MTS，MSMQ，形成应用程序事务与无连接缓冲通信的多层次主从式应用系统。

1.1.1　Windows DNA 可以解决的问题

Windows DNA 为现有的任何机器及资源提供了一个可以轻松容易地推行分布式网络

应用系统的方案，其中包括 Web 的集成运用：由于 DNA 方案建立在 Windows 平台上，利用现有的丰富资源，系统开发者可以集中精力做系统的逻辑开发，而不是费力地与系统体系结构打交道。另外，运用微软所推出的子组件工具(MMC，MTS，MSMQ...)，就可以轻松地管理每一个系统层，这包括系统安全、资源再利用以及通信上的问题。

1.1.2 Windows DNA 包含的核心组件

DNA 可以运用 COM 技术并充分利用现有的资源，再加上微软开发的专用于 DNA 的子组件工具，这样，Windows DNA 的核心组件资源十分丰富，它包括：

- 前端服务：HTML，DHTML，Scripting，Components，Win32 API，标准浏览器
- 应用程序资源 IIS(Internet Information Server)，MSMQ，MTS，COM，ASP(Active Server Pages)，DCOM，Microsoft Java Virtual Machine
- 数据库统一使用：ADO，OLE DB，ODBC
- 系统资源：安全、管理、网络通信

1.1.3 Windows DNA 体系结构的优点

- Windows DNA 提供了各种平台开发与集成的能力，且系统开发者可以运用并提供多种多样的中间层资源(Middle-Tier)，这些中间层资源包括了异步消息队列(asynchronous message queuing)、事务操作(transaction)、COM 组件(Component)、数据库存取(Database Access)与 Web 集成。
- 建立在熟悉的 Windows 平台上，让系统开发者可以更快、更容易的开发系统。
- 支持各种各样的程序语言与开发工具，这让系统开发者可以选择自己最擅长的工具开发 Windows DNA 系统。
- Windows DNA 提供了与现行企业应用服务器和子系统的高度兼容性，这使得企业对过去的投资不会被浪费。

1.1.4 使用 Windows DNA 体系结构的系统有什么特色

- DNA 可以帮助你清楚又有效地规划多层次主从式应用系统。
- DNA 让客户端透明：也就是说无论后端的应用系统如何运行，前端的用户都不需要了解。
- DNA 体系结构支持事务处理：所谓“事务”就是整体处理的一致性，即一事务除了成功外就是失败，不存在第三种情况。
- DNA 体系结构可以增强数据事务的容错与缓冲能力(fault tolerant)。
- 使用 DNA 可以构造十分完美的分布式系统。

1.1.5 系统开发者如何实现 Windows DNA 体系结构

- 将整个系统分为三个部分：前端(Presentation)客户界面、商业逻辑(Business logic)、层与后端数据库(Data)。
- 前端可以选择合适的窗口对象与技术开发用户的操作界面。
- 在中端商业逻辑层中，可以自行开发 COM 对象并集成到 Windows NT 系统。

· 在后端数据库的存取上，我们使用 OLE DB 来处理。

1.1.6　Windows DNA 体系结构集成 COM 技术

Windows DNA 体系结构的最核心技术即是集成 COM 到 Web 与主从式应用系统之中，它为分布式系统定义了每一层所需要的服务资源与兼容性，这包括了 DLL 组件、Dynamic HTML、Web 浏览服务与服务器(Server)、脚本(Script)，事务(Transactions)、消息队列(message queuing)、安全与系统管理、数据库与数据存取、用户界面，这些可用的数据都必须符合 COM 标准。

1.1.7　Windows DNA 支持的客户端平台

Windows DNA 体系结构支持所有符合 W3C HTML 标准的客户端环境，另外，Microsoft 的 COM 技术已经逐渐实现在 UNIX、MVS 与 Macintosh 平台上，并推出 Internet Explorer 让 UNIX 与 Macintosh 可以读取丰富的 HTML 多媒体文件。就像一般的 Internet 应用程序，Windows DNA 的前端亦支持 Win32 用户界面，这意味着系统开发人员可以自行建立多媒体、音响数据流或数据存取界面。

1.1.8　Windows DNA 支持的开发工具

由于 Windows 体系结构可以在 Windows 平台上切实地与 COM 技术进行集成，因此现在市面上所有的 RAD(Rapid Application Development)开发工具皆可用来开发 Windows DNA 系统，例如，Microsoft Visual Basic，Microsoft Visual J++，Borland 的 Delphi，PowerSoft 的 PowerBuilder；另外，像 Microsoft InetDev 标准的 HTML 开发工具与 Microsoft FrontPage 管理与应用工具也可以与 Windows DNA 结合用于开发。

1.1.9　Windows DNA 支持的应用服务器

Windows DNA 广泛地支持各种应用服务器，这些服务器包括了高性能的 Web 服务器(Microsoft Internet Information Server)、可扩充与继承的对象技术(COM，DCOM)、强健又稳固的分布式事务管理器(Microsoft Transaction Server)、面向消息及消息队列的中层(Middle-Tier)应用工具(Microsoft Message Queue Server)，以及其他安装在 Windows NT 平台上的各种应用服务与协议，例如安全管理与群组账号，TCP/IP 通信协议等，因此只要是 Windows 平台的客户端都可以直接使用 Windows DNA 的系统资源。

1.1.10　Windows DNA 支持的后端应用服务器或资源

Microsoft 的 BackOffice 平台上的所有资源皆可与 Windows DNA 集成，例如数据库系统(Microsoft SQL Server)、电子邮件与群组系统(Microsoft Exchange Server)、主机统一连接与管理工具(Microsoft SNA Server)、应用程序管理工具(Microsoft Systems Management Server)、Internet 与企业网络建设与管理工具(Microsoft Site Server)。

另外，Windows DNA 也可以集成其他厂商的后端应用服务资源，例如，Oracle 的数据库系统、Lotus 的 Notes 群件系统，以及其他的开放式协议与 WOSA(Windows Open Services Architecture)体系结构技术，例如通用数据存取模式(Universal Data Access model，这包括

ODBC 与 OLE DB 技术),即可以存取各种各样的企业数据库资源。

1.1.11 如何集成 Java 于 Windows DNA 体系结构中

今天,熟悉 Java 语言的系统开发者可以利用它来自动开发 COM 组件,并集成至 Windows DNA 体系结构中的任一层(Any Tier),微软为此提供了与 Windows 平台相集成的高性能 Java 虚拟机(Virtual Machine),因此可与 Internet Explorer 或 Microsoft Internet Information Server 做出高度集成,另外,Windows DNA 体系结构也可与开放式标准的 CORBA 对象技术结合。

1.2 DNA 系统的结构

在开始使用与了解 DNA 三层式体系结构时,让我们回顾一下传统的主从式两层体系结构。

1.2.1 两层式主从体系结构

主从体系结构是用于 Internet 系统设计的最自然的方法,它以组件为基础使得前端客户端界面、后端服务器区分的很清楚,这样也大大地降低了子系统、组件、服务器复杂的设计机制。运用主从式体系结构的设计模式,程序员可以把整个系统划分为一部分、一部分的组件或子系统(这也比较符合人类的思考模式),并分别运用在前后端。典型的主从式系统(见图 1-1)仅区分为前端客户端系统(Front-End Client Application)与后端服务器系统(Server Application),两者所起的作用如下:

- 前端客户端系统:包含了客户端使用界面(任何程序语言皆可以开发)与程序逻辑或商业规则运算。
- 后端服务器系统:系统所需的数据(例如文件、数据库等)与部分程序逻辑。

两层式主从体系结构典型的运作方式是前端客户端与后端服务器系统直接通信,客户端直接向服务端发出数据请求,服务端则运用本身的数据处理机制处理请求,并将数据返回客户端应用程序,客户端接收后再经由逻辑运算或商业规则(例如,银行贷款的计算规则)加以处理并显示给用户。

大致上看来,两层式体系结构系统似乎运行很顺利,事实上不然,它隐藏了以下几个缺点:

- 逻辑运算与商业规则缺乏灵活性与扩充性。
- 服务器数据处理过分依赖数据库系统。
- 系统无法有效地扩充与提供良好的分布处理效率甚至数据事务的容错与缓冲能力。

逻辑运算与商业规则由前端客户系统处理,这潜在地使系统低效及缺乏灵活性,系统设计人员也因此必须为不同的用户(有些权限较高、有些较低)开发不同的客户端界面,由于部分的关键安全检查位于客户端界面,安全上也可能造成漏洞;在服务器端数据库的开发要完全依赖数据库产品所提供的开发类模块,这可能造成开发者学习上以及跨数据库应用上的困难;另外,当程序逻辑或商业规则改变(例如银行某一贷款的计算公式完全改变)则必须重

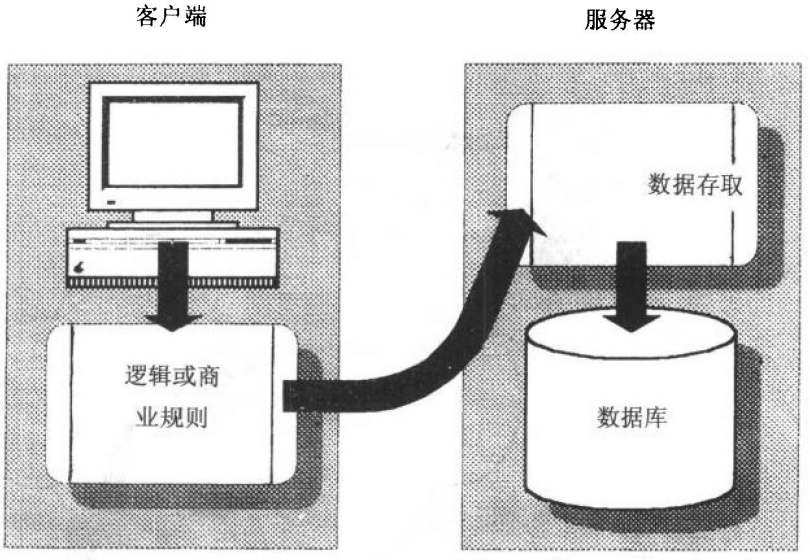

图 1-1　传统的两层式主从体系结构

新开发前端客户界面程序、再重新安装在每一个客户端上，不用想也知道这种做法不灵活，正因为如此，分布式的系统(亦即系统各部分在不同的电脑上执行)根本不可能在两层式体系结构中实现！

1.2.2　DNA 体系结构

Microsoft 推出的 DNA 体系结构中，将分布式网络系统区分为三层，分别是：

- 前端用户(Front-end User Service tier)
- 中端商业逻辑(Middle Bussiness Service tier)
- 后端数据储存(Back-End Data Service tier)

与传统两层式主从式体系结构最明显的不同之处，在前端的数据库处理商业逻辑与后端的数据库处理原则已被封装为标准的 COM 组件，并依附于中端商业逻辑处理环节，借助 Microsoft 开发的 Microsoft Transaction Server 使该逻辑组件更强健、稳定性更高，借助 Microsoft Message Queue Server 让数据库处理具备容错与缓冲能力，借助 Windows NT 本身群组与账号管理模块，让组件的使用达到安全管理目的。

将商业逻辑与数据库处理原则封装为标准的 COM 组件，并放在中间层，如此产生的差异性有很多，对后端数据处理可以通过标准的 OLE-DB 与 ODBC 访问数据库，而不再是过分依赖数据库厂商提供的开发环境，对于前端用户界面，除了可提供标准 Win32 界面外(与 DCOM 结合)，亦可以运用标准的 Web 浏览平台(这时候中端采用 Active Server Pages 封装 COM 组件)，以增进应用程序用户学习的亲切感与熟悉度。

图 1-2 刻划出典型的三层式 Web-DNA 应用体系结构，前端为标准 Web 浏览器，用于显示 Web Server 输出的网页内容，该输出网页可能包含了 HTML、Script(JavaScript，VBScript)，DHTML，Java Applet 或 ActiveX Control，动态的网页则由 Web Server (IIS 4.0)的 Active Server Pages 产生，其中的商业逻辑组件则由 ActiveX DLL 组件包裹，该商业逻辑组

件包含了对数据库的处理方法，该数据库可能是本地的，也可能是远程的。

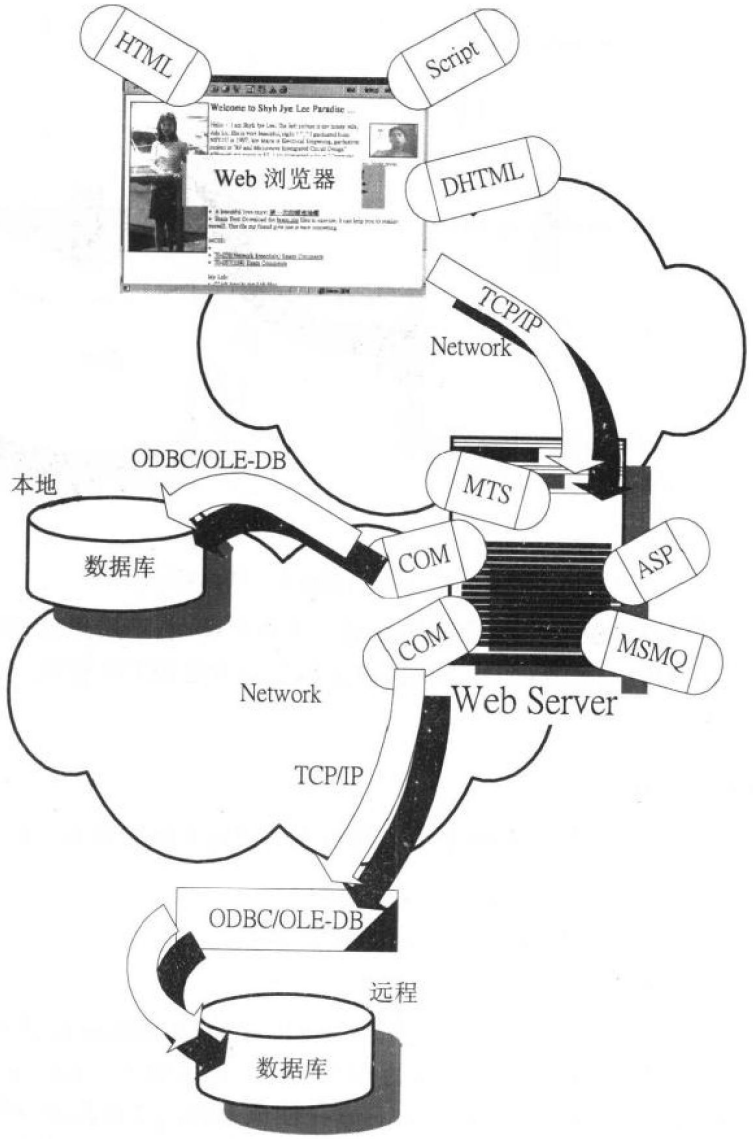

图 1-2 三层式 Web-DNA 应用体系结构

有些系统数据处理必须具备“事务”的能力，所谓“事务(Transaction)”就是应用系统动作的一致性，即是事务只有成功与失败，不存在第三种例外情况，比如银行转账，因为我们希望提款与加款的银行操作必须一致，而不发生只提款或不加款的第三种情况，通过 MTS 可以轻松实现这种功能。

网络与系统时常存在不稳定因素，例如，由于网络临时的断线或高峰时刻的阻塞而造成系统之间无法通信，甚至数据库服务器死机。为了让分布式应用程序系统具备容错处理的能力，能够让不稳定因素恢复正常后能继续运行，你可以集成 MSMQ 至系统中达到目的。

提示 在 Windows NT5 与 NT4 Option Pack 中，所有对数据库(SQL Server，Ac-

cess...)的通信均通过 OLE-DB,当你在 ODBC 设定一数据库源名称后,就可以通过 OLE-DB 与数据库源做处理。

1.3　什么是 COM

前一节已清楚地介绍过,微软事务服务器(Microsoft Transaction Server, MTS)提供了一组件式的主从式体系结构执行环境,这些组件就是 COM 对象,它主要负责数据的存取、程序逻辑或商业规则的封装操作。

因篇幅所限,在此不对 COM 作全面讲解,本节只讲 COM 的基础内容。

1.3.1　COM 的发展

自从 Windows 操作系统流行以来,"剪贴板(Clipboard)"首先解决了不同应用程序间的通信问题(由剪贴板作为数据交换中心,进行复制、粘贴的操作),但是剪贴板传递的都是"死"数据,应用程序开发者得自行编写、解析数据格式的代码,于是动态数据交换(Dynamic Data Exchange,DDE)的通信协定应运而生,它可以让应用程序之间自动获取彼此的最新数据,但是,解决彼此之间的"数据格式"转换问题仍然是程序员沉重的负担。

对象的链接与嵌入(Object Linking and Embedded,OLE)的诞生把原来应用程序数据的交换提高到"对象交换",这样应用程序之间不但获得数据也同样获取彼此的应用程序对象,并且可以直接使用彼此的数据内容,但是 OLE 的复杂程度不亚于 DDE,因此,许多程序员仍望而却步,直到 OLE 的出现才解决了所有的问题。

OLE 使用了新的机制,使得程序员可以很自然的把别人的程序当作自己的一部分来使用,这就解决了应用程序彼此通信的问题,让程序员拥有无限宽广的想象空间,而 OLE 的一切技术基础都是建立在用面向对象的概念使大家可以共享程序与重用组件(Component Object Model,COM)的体系结构之上,它提供了一组件彼此间通信的机制,且对象可以由不同的程序语言来开发,因此未来的软件开发不再是一行一行的编码,而是像堆积木一般将不同的组件组合起来。

随着 Internet 的发展,微软在 1996 年推出 ActiveX 技术用来取代庞大的 OLE 以应用于 Internet,但它仍建立在 COM 的基础之上(COM 提供了低层对象之间通信的机制,而 ActiveX 与 OLE 是建立在 COM 上的服务),ActiveX 与 OLE 最大的不同在于,OLE 针对的是桌面上应用软件和文件之间的集成,而 ActiveX 则以提供进一步的网络应用与用户交互为主;而我们使用 MTS 的最终目标是 ActiveX 的 Server DLL 组件开发。

1.3.2　COM 是什么

本书不准备写太多关于 COM 的细节(市面上关于 COM 的好书很多),而着重在实际的开发,现在讲述作程序开发所需的 COM 知识(目的是希望没有 COM 或 OO 经验的程序写作者也可以轻易地开发企业 DLL 组件)。

如何开发一个 COM 组件?开发一个 COM 组件就像开发一个对象类(Class),该类拥有属性(Property)与方法(Method)集合(Collection)作为它提供的服务与外部通信的接口(Interface)或通道,类在执行时期被称作实例(Instance);对象就像一部电视机,28 寸屏幕是它

拥有的属性，利用选台器的按键提供给用户切换频道的方法，这样比喻是为了便于读者对对象(Object)这一概念的理解。

提供给外部应用程序使用的 COM 对象，我们称之为 COM Server，在 ActiveX 领域里则称为 ActiveX Server，该对象称为 ActiveX Server DLL 组件，而一个 ActiveX Server 组件内可以存在多个类(Class)，每个类亦拥有多个属性与方法。

使用 Microsoft Visual Basic 6.0 来开发 ActiveX Server DLL 组件是最简单的，因为 VB 隐藏所有的复杂机制，组件开发者可以很轻松地设计出想要的 Dual Interface 企业组件。只要打开 ActiveX DLL 工程，并设计组件所需的方法(Method)与属性(Property)，即可立刻编译成 DLL 文件，在编译的同时 VB 会自动将组件注册在系统注册表(Registry)中。

在此，还要介绍几个概念，分别为 Automation，Dual Interface 与 Registry。

1.3.3 Automation

要说明 Automation 得从早期绑定与滞后绑定两个概念着手；在使用 VB 开发应用程序时，系统会引用所需的对象供程序员使用，因此 VB 的开发环境可以即时列出该对象内所拥有的属性、方法或相关信息，若应用程序编译完成后该组件被更新(或增强)，应用程序得重新使用 VB 编译才可以拥有该组件更新后的功能，这种在开发时期引用对象的方式称作早期绑定。

滞后绑定则是在应用程序执行时期(run-time)才通过公用界面，IDispatch 动态引用所需的对象，这种做法特别适合网络的环境或 Script 应用程序环境(例如 Active Server Pages)，这种支持 IDispatch 公用界面的 COM 对象就称作 Automation Server。

1.3.4 Dual Interface

一般拥有自定义 Custom Interface 的 COM 对象效率高，适合用于开发高效率应用程序，拥有公共接口 IDispatch 的 COM 对象适用及应用的范围较广，例如网络或 Script 解释环境，至于同样拥有该两种特性的 COM 对象接口则称作 Dual Interface，使用 VB 6.0 则可以轻易地建立 Dual Interface 的 COM 对象。

1.3.5 Registry

COM 对象经编译后在系统内可以按两种机制来识别，分别是 CLSID 与 ProgID，ProgID 是 128 位的 GUID(Globally Unique IDentifier)数值，该数值在系统内是唯一存在的，一般编译环境会自动产生。但是这一长串 128 位数字很难记住，于是引入了 CLSID 以方便组件用户引用(换个角度来看，ProgID 就像身份证号的一长串，CLSID 就是姓名)，CLSID 一般以下面的形式存在：

公司或作者名称·组件名称·版本号

例如，JFCorp. ADODatabase. 1，而 MTS 则是以 ProgID 识别组件。

1.4 小 结

本章对 DNA 体系结构作了清楚地介绍，概述了 DNA 的总体结构，下面三章为本书的