

TP 393
C 23

完全探索Microsoft.NET 技术战略

李庆发 / 编著

Microsoft.NET



中国青年出版社
CHINA YOUTH PRESS

目 录

第一篇 .NET 新观念

第 1 章 .NET 概述

- 1.1 信息环境正在革新 4
- 1.2 微软技术演进 4
 - 1.2.1 革命的起点——面向对象
程序设计 5
 - 1.2.2 组件软件革命——ActiveX
与 COM 7
 - 1.2.3 全面网络化——DNA 平
台 8
- 1.3 .NET 概述 10
 - .NET 如何运作 11

第 2 章 XML 概论

- 2.1 文件的“意义” 16
 - 2.1.1 没有意义的文件 16
 - 2.1.2 较为结构的文件——
HTML 16
 - 2.1.3 HTML 文件的缺点 19
- 2.2 SGML、HTML 与 XML 21
 - 2.2.1 标记语言 21
 - 2.2.2 标记语言的标准——
SGML 22
 - 2.2.3 规范文件的 DTD 23
 - 2.2.4 SGML 的优点 24
 - 2.2.5 网页的标记语言——

HTML 24

- 2.3 简单而强大的 XML 25
 - 2.3.1 新需求应势而起 25
 - 2.3.2 二手电脑出售——XML
版 25
 - 2.3.3 功能强大的 XML 26
- 2.4 XML 为什么重要 27
 - 无所不在的 XML 27

第 3 章 SOAP

- 3.1 分布式对象结构演进 30
 - 3.1.1 从一个子程序说起 30
 - 3.1.2 面向对象程序设计 31
 - 3.1.3 标准对象结构 31
 - 3.1.4 把程序分散出去——RPC 32
 - 3.1.5 分布式对象结构 33
- 3.2 分布式对象结构 34
 - 3.2.1 广受学术界喜爱的
CORBA 34
 - 3.2.2 跨平台、跨语言是 CORBA
的特色 35
 - 3.2.3 理想与现实总有差距 35
 - 3.2.4 Windows 平台实作最广的
DCOM 36
 - 3.2.5 简单就是美 36
 - 3.2.6 超级比一比——CORBA 与
DCOM 37

- 3.3 SOAP = HTTP + XML 37
 - 3.3.1 什么是 SOAP 37
 - 3.3.2 DCOM 的服务方式 38
 - 3.3.3 将服务标准化 38

第4章 Web Service

- 4.1 信息之窗——WWW 42
- 4.2 典型的 WWW 服务 42
 - 4.2.1 Windows DNA 架构活化
WWW 服务 42
 - 4.2.2 一般程序运作架构 43
- 4.3 网络上的对象服务——Web
Services 44
 - 4.3.1 .NET 下的 Web Services 44
 - 4.3.2 .NET 结构下的应用程序 45
 - 4.3.3 靠得更近一些 46
- 4.4 Web Services 的安全性 49
 - 4.4.1 存取 Web Services 的保护 49
 - 4.4.2 利用 IIS 的验证机制保护
Web Services 存取 50
 - 4.4.3 保护传递的数据 50

第5章 Common Language Runtime

- 5.1 从虚拟机谈起 54
 - 5.1.1 程序语言和机器语言 54
 - 5.1.2 由程序语言到机器语言 54
 - 5.1.3 机器只认得自己的语言 55
 - 5.1.4 机器不同, 软件版本也
不同 55
 - 5.1.5 虚拟机的诞生 56
- 5.2 现有的虚拟机 58
 - 5.2.1 Java Virtual Machine 58

- 5.2.2 Visual Basic 的 Virtual
Machine 58

5.3 中间语言——Intermediate

- Language (IL) 59

5.3.1 公共的语言——中间语言

(Microsoft Intermediate

Language) 59

5.3.2 跨语言的合作 60

5.3.3 MSIL 的特点 60

5.3.4 中间语言范例 61

5.4 Common Language Runtime

(CLR) 64

5.4.1 Managed Code 64

5.4.2 Common Type System

(CTS) 64

5.4.3 Common Language Speci-

fication (CLS) 64

第二篇 .NET 新开发平台

第6章 XML 与 SOAP 的开发

- 6.1 XML 程序设计 70
 - 6.1.1 存取 XML 文件 70
 - 6.1.2 XML 文件的验证 70
 - 6.1.3 DOM 还是 SAX 73
- 6.2 微软 XML 开发工具 73
 - 6.2.1 MSXML (XML Parser) 73
 - 6.2.2 取得 MSXML 74
 - 6.2.3 如何开始 75
 - 6.2.4 .NET 中的 XML 75
 - 6.2.5 相关信息 75

6.3	SOAP 信息格式	76
6.3.1	SOAP 信息数据包	76
6.3.2	SOAP 的调用与回应	77
6.4	SOAP 开发工具	78
6.4.1	整合 Web Services 开发的 VS .NET	78
6.4.2	SOAP Toolkit	79
6.4.3	将 COM 转换成 Web Services	79

第 7 章 .NET Framework

7.1	.NET 的基础工程——.NET Framework SDK	82
7.1.1	.NET 梦想的实现	82
7.1.2	.NET Framework SDK 的 组成	82
7.2	.NET Framework 对象结构	85
7.2.1	.NET Framework Class Library	85
7.2.2	与程序语言结合的对象 函数库	85
7.2.3	.NET Framework 对象模 型初探	86
7.3	ASP .NET	90
7.3.1	更新的结构	90
7.3.2	结合 .NET Framework	91
7.3.3	性能的提高	92
7.3.4	建置, 不再烦人的建置	92

第 8 章 旧语言、新生命——Visual Studio.NET

8.1	Visual Studio .NET 新结构 与新开发环境	96
8.1.1	整合的开发平台	96
8.1.2	认识 Visual Studio .NET 的新面孔	97
8.1.3	多样化的辅助工具	98
8.1.4	Web Forms 与 Windows Forms	103
8.1.5	简化 Web Services 的开 发	104
8.2	旧语言、新生命——Visual Basic .NET	109
8.2.1	震撼人心的变革	110
8.2.2	细说改头换面的 VB7	110
8.2.3	面向对象时代	114
8.2.4	转换语言	116
8.3	.NET 原生语言——C#	120
8.3.1	告别 Java	120
8.3.2	C 井?	120
8.3.3	C# 的特色	121
8.3.4	与 .NET 结合	121
8.3.5	简单的范例	121
8.4	温和演进的 Visual C++—— Managed Extensions for C++	122
8.4.1	Managed Code 与 Unma- naged Code	122
8.4.2	Managed Extensions for C++ 的功能	123
8.5	惟一不会变的, 就是会一 直在变!	123

第三篇 .NET 新平台

第 9 章 .NET 新操作平台

- 9.1 平台的根基——Windows
 - 2000 Servers 128
 - 9.1.1 .NET 结构的背后支持者 ... 128
 - 9.1.2 Windows .NET——传说
 - 中的 Blackcomb 128
- 9.2 组信息平台——Exchange
 - 2000 Server 129
 - 9.2.1 组信息平台 129
 - 9.2.2 结合 Active Directory 服
 - 务 131
 - 9.2.3 Exchange 的开发资源 131
- 9.3 聪明的数据存储——SQL
 - Server 2000 132
 - 9.3.1 数据仓库 (Data Ware-
 - housing) 132
 - 9.3.2 Data Mining 132
 - 9.3.3 XML 与 Internet 133
 - 9.3.4 轻量级的巨炮——SQL
 - Server 2000 Windows CE
 - Edition 134
- 9.4 让企业沟通——BizTalk
 - Server 2000 134
 - 9.4.1 BizTalk Framework 134
 - 9.4.2 设计商业流程 135
- 9.5 电子商务平台——Commerce
 - Server 2000 136
 - Commerce Server 2000 功能总览 .. 136

- 9.6 知识管理新纪元——Share-
 - Point Portal Server 2001 137
 - SharePoint Portal Server 功能总
 - 览 138

第 10 章 促进企业生产力——其他 .NET Servers

- 10.1 整合原有系统——Host
 - Integration Server 2000 142
 - 整合原有系统 142
- 10.2 Scale Out!!——Application
 - Center 2000 144
 - 10.2.1 Scale Up vs. Scale Out 144
 - 10.2.2 软件 Scale Out 145
 - 10.2.3 Application Center 的
 - 功能特点 145
- 10.3 Internet Security and Accele-
 - ration Server 2000 146
 - 10.3.1 企业的防火墙 147
 - 10.3.2 灵活的防火墙系统 147
 - 10.3.3 管理内部网络资源 147
 - 10.3.4 整合 VPN 148
 - 10.3.5 提高网络效率的 Web
 - Cache 148
 - 10.3.6 定期预载网页内容 148
 - 10.3.7 主动式快取 (Active
 - Caching) 机制 149
 - 10.3.8 与 Windows 2000 整合 ... 149
- 10.4 Mobile Information 2001
 - Server 150
 - 10.4.1 拓展系统平台 150

10.4.2	Outlook Mobile Access	150
10.4.3	Mobile Web Form	151
10.4.4	Mobile Manager	151

第四篇 .NET 企业新印象

第 11 章 新产业、新商机

11.1	畅谈软件产业	156
11.1.1	专业软件公司出现	156
11.1.2	软件分工的初步—— 函数库	156
11.2	组件制造产业	157
	ActiveX Era	157
11.3	应用程序服务供应商 (Ap- plication Service Provider)	158
11.3.1	软件租赁	159
11.3.2	ASP 的智能科技	160
11.3.3	ASP 的优点	161
11.4	.NET 结构下的 Web Services 产业	162
11.4.1	Web Services 产业经营 模式	162
11.4.2	Web Services 产业 vs. 组件产业	163
11.4.3	Web Services 产业 vs. ASP 产业	163
11.4.4	个案实例: Microsoft Passport 服务	164
11.4.5	结束语	165

第 12 章 .NET 系统整合开发

12.1	.NET 开发平台的整合	168
------	--------------	-----

12.1.1	整合的系统	168
12.1.2	开发工具的选用	170
12.1.3	扩大系统规模	171
12.1.4	结合原有系统	171
12.1.5	与合作厂商沟通	171
12.1.6	企业内部客户	171
12.1.7	企业顾客	171
12.1.8	移动设备	172
12.2	案例分析——网上书店的 开发	172
12.2.1	网站的功能	172
12.2.2	有效整合程序及美编 人员	173
12.2.3	结合后端组件服务	174
12.2.4	结合 .NET 平台	175
12.2.5	整合网站设计	178

第 13 章 商业智能

13.1	什么是商业智能	180
13.1.1	传统的商业程序	180
13.1.2	信息孤岛	180
13.1.3	文件管理	180
13.1.4	决策支持	181
13.1.5	现代企业面对的环境	181
13.1.6	商业智能使得组织变 得更具智能	182
13.1.7	商业智能的致能科技	183
13.2	Data Warehousing	183
13.2.1	数据分析程序	184
13.2.2	OLAP	185
13.2.3	Data Mining	185

13.2.4	Microsoft Data Warehouse Alliance	187
13.3	浅谈知识管理	187
13.3.1	知识的管理	187
13.3.2	文件管理中心	188
13.3.3	更多元的文件管理	189
13.3.4	知识管理的未来	189
13.4	打造商业智能环境	190
13.4.1	案例简介	190
13.4.2	目前系统的问题	192
13.4.3	使用 .NET Enterprise Servers 建构商业智能架构	192

第 14 章 拓展 .NET 平台——移动设备解决方案

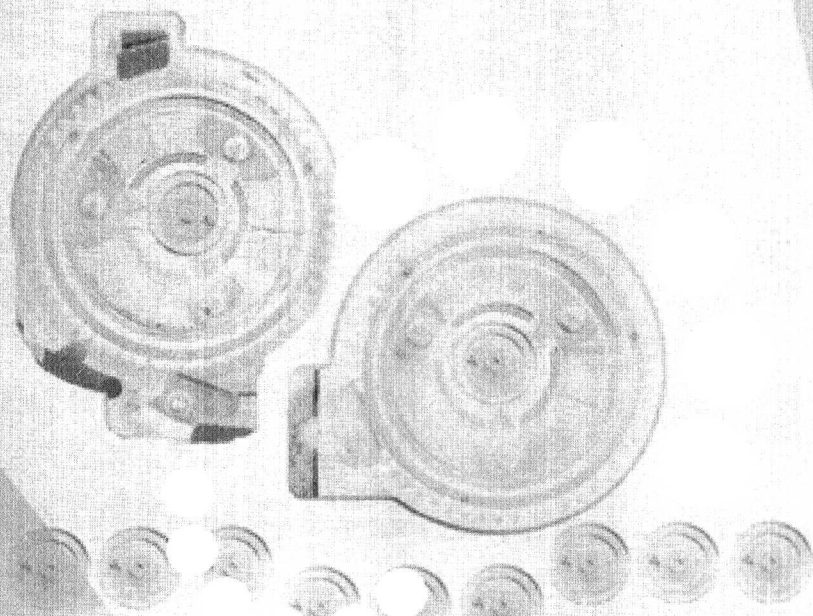
14.1	.NET 移动设备大检阅	196
14.1.1	移动平台的基础——Windows CE 操作系统	196
14.1.2	掌上电脑 (Handheld PC 以及 Handheld PC Pro)	197

14.1.3	掌上型电脑 (Palm-Sized PC)	198
14.1.4	口袋型电脑 (Pocket PC)	198
14.1.5	移动电话 (Mobile Phone)	199
14.2	移动设备应用程序开发	200
14.2.1	eMbedded Visual Tools	200
14.2.2	eMbedded Visual Basic	200
14.2.3	eMbedded Visual C++	203
14.2.4	.NET Mobile Web SDK	204
14.2.5	Mobile Form 范例	205
14.2.6	结束语	206

附录 A 迈向 .NET 之路——.NET 学习地图

A.1	XML	210
A.2	.NET 基本概念	210
A.3	.NET 平台	211
A.4	程序开发相关资源	211
A.5	如何学习 .NET	212

第一篇 .NET 新观念



微软又推出了新技术，名字叫做“.NET”，.NET 是微软下一代操作系统环境的技术总合。它是一个全新的观念，有新的支持平台，也有全新的开发环境。

本篇将向读者做一个全面而浅显的介绍。第一章从微软的技术沿革，到.NET 的初步认识。第二章逐项介绍.NET 的关键技术，如 XML、SOAP、分布式运算等。务求让读者能够了解.NET 的精神与观念。

第1章

.NET 概述

从2000年8月开始，微软的各个研讨会场上，开始出现这样一个影片。影片中的男主角要到外地出差，男主角到了机场的服务台，服务人员给了他一台移动电话。他将自己的SIM卡插入到这台移动电话中，移动电话便自动下载这位男主角的通讯录等个人资料。接下来的旅程中，出了一些意外，男主角便通过这支神奇的电话与自己的保险公司进行视频通信。而保险公司也转介当地的医院，利用他手上的移动电话进行远程医疗。

这看起来似乎是老掉牙的科技生活的影片，虽然从头到尾从未提及到.NET这个字眼，但却突显出了微软对于.NET这个技术的想法以及远景。

2000年8月，微软最大的技术人员研讨会——TechEd 2000揭示了.NET时代的来临。同样地，这次微软还是秉持着一贯的态度，强调新的平台能够降低企业的总拥有成本（TCO，Total Cost of Ownership）、提高企业生产力、获得信息更为方便等。但身为技术前锋的资讯人员，更应该发现另一个不同的世界。

1.1 信息环境正在革新

因特网（Internet）的发展带动真实社会的变化已经不是什么新闻了，1996 年到 1998 年的网络产业也确实推动了世界的经济，更创造了无数的就业机会与梦想。但网络不像石油，网络本身并不具备任何价值，网络的价值在于如何去应用它。有好的应用方式，网络可以比黄金更昂贵。反之，网络终究只不过是一个工具罢了。

因特网带来的新应用有很多，其中有一个技术历史悠久，但一直到最近才又因网络而起——“分布式计算（Distributed Computing）”。分布式计算就是想通过团结就是力量的原理，结合一群电脑来实现极为复杂的计算。靠着分布式计算，电脑可以与其他电脑沟通，彼此共享着计算的信息。而工作也可以分散到网络各地不同的电脑上执行，以加速整体计算的速度。

不但如此，在文件表达方面也有若干进步，在 HTTP 的能力备受质疑，而企业间的文件沟通愈来愈困难的情况下，XML 被制定出来了。XML 具有强大的弹性，可以用来表达文件的相关信息（meta-data）。近年来更成为电子商务的热门话题。

在经过 20 年的极盛时期后，个人电脑也开始走下坡路。取而代之的是种类更多、类型更广的电子产品，如：手机、IA 家电、PDA（个人数字助理）等。这些设备都结合了当前最兴的科技，使得人们取得信息的方式越来越灵活。

在种种技术、平台、产品都迈向同一个目标快速进展的情况下，微软整合了这些平台，于是一个新的平台架构诞生了。

1.2 微软技术演进

在正式进入 .NET 主题之前，先让我们回顾一下微软软件技术的演进。

当微软在以 MS-DOS 主宰着全球个人电脑操作系统市场的时候，微软在软件技术上的成就，并不是极其突出。论操作系统的能力，以 MS-DOS 的 16 位单机操作系统，自然比不上当时 IBM 的 OS/2 或是尚未登上操作系统主流舞台的 Linux。论开发环境，微软除了 Quick Basic 之外，大部分的开发环境仍是 Borland 的天下，Turbo C++ 仍是当时的主流开发平台。

这个态势，在微软推出 Windows 3.0 这个视窗操作系统时更为明显。虽然是自家推出的视窗环境操作系统，但讽刺的是微软并没有为这个操作系统提供任何完全视窗环境下的 IDE（整合式开发环境）开发界面。倒是开发软件供应厂的老大哥——Borland 公司率先提

供了 Borland C++ 这个完全在视窗环境下就可以开发软件的开发工具（注：原来在开发视窗应用程序时，还必须在 MS-DOS 下进行程序撰写及编辑操作。编辑完成后，再拿到视窗环境下执行看看是否运行正常）。但在 20 世纪 80 年代末期，一系列的软件革命使得电脑用户与开发者的使用习惯产生了极大的改变。这一切都得从面向对象开始说起。

1.2.1 革命的起点——面向对象程序设计

对学术界或较资深的程序设计师而言，面向对象（Object-Oriented）早已不是一个新名词了。早在 20 世纪 70 年代，Xerox 的 PARC（Palo Alto Research Center）就已经对面向对象进行过初步的基础研究。到了 1981 年，第一个广受欢迎的面向对象语言——Smalltalk 也开始问世。一直到了 1985 年 C++ 之父——Bjarne Stroustrup 将面向对象概念引入到 C++ 语言之后，面向对象程序设计（Object-Oriented Programming, OOP）便迅速成为程序设计。

究竟什么是面向对象程序设计呢？其实，面向对象的目的，就是要将原来晦涩难懂的程序设计概念拉回到自然、直观的设计方法。以机器来说，在真实世界中，我们所看到的机器，可能是由许多较小的部件组合而成。而这些机器又可能与其他机器有某种关系存在。我们不需要知道每个机器的内部运作，只要知道这个机器它需要哪一类的输入，会产生怎么样的输出结果，我们便可以将这个机器与其他机器结合使用。

面向对象程序设计，就是将这种思考原理应用在程序设计的方法上。OOP 将程序的功能拆成一个一个的“组件”，每个组件都拥有属于自己的属性（property，又叫数据成员——data member）以及方法（method，又称为成员函数——member function）。所谓“属性”，就是这个对象特有的参数，例如：“学生”这种对象，就可能拥有“学号”、“姓名”、“班级”等属性。所谓“方法”，就是这个对象本身可以处理的行为，例如：“出席”、“考试”等，都可能是“学生”这个对象的方法之一。

面向对象的概念有三个，分别是：

● 封装（encapsulation）

封装强调的是对象中数据的隐密性。封装的概念就是，每个对象只能处理自己的属性及变量。对于其他对象，只能通过对方对象提供的方法间接地处理。这种做法的好处在于，因为只有自己能够处理自己的变量，因此简化了将来的排错程序。

● 继承（inheritance）

继承可说是面向对象中最引人好奇的部分了。现在厂商在设计灯具时，已经很少再重新设计灯泡了。软件开发也一样，把程序中具有共同性质的部分独立成一个组件，然后再给其他具有类似性质的对象继承。这样，新的对象不需要每次重写共同部分的程序代码，

便具有共同的组件特性。继承可以简化大型程序的开发。

如图 1-1 所示，以发动机这个机械设备来说，发动机本身具有它自己的属性以及方法。

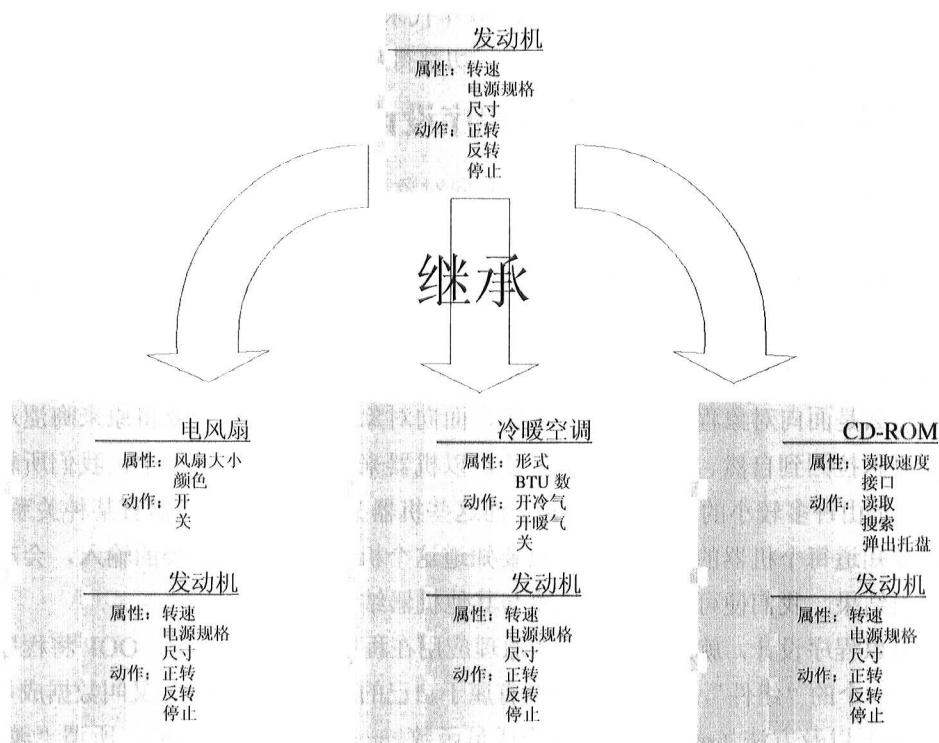


图 1-1

多态 (polymorphism)

多态可说是面向对象思想中最具有挑战性的观念了。多态是让一个继承体系的子对象继承父对象声明的方法。此后我们只要取得这个继承体系的任何一个对象，即使不知道这个对象的确切类，直接调用它的父对象所声明的方法，还是可以得到正确的结果。

以“图形”为例，虽然“圆形”、“方形”、“菱形”都各是不同的对象，但只要它们都是由“图形”这个对象继承而来，而“图形”又有一个叫做“画出来”的方法的话，又因为“圆形”、“方形”、“菱形”等对象都会实作出适合自己的“画出来”方法。因此不论我们取得哪个对象，只要直接调用它“画出来”的方法，自然就会得出正确的结果，程序不必要去知道这个对象到底是什么类型的。

面向对象观念对于传统程序设计的思考方式而言，有着非常大的区别，但这种观念却正适应了当前的电脑应用环境。以视窗这种图形化用户界面 (Graphic User Interface, GUI)

来说,使用对象的概念,可以将画面上看得到的各种组件以对象来开发,并以对象互动的方式来组合出一个大型且复杂的应用程序。因此,面向对象的概念深深地影响着以后所有的软件技术。

1.2.2 组件软件革命——ActiveX 与 COM

故事要从 OLE (Object Linking and Embedding, 对象链接与嵌入) 开始讲起。OLE 是微软为了实现复合式文件 (Compound Document) 的理想而开发出来的技术。所谓复合式文件,就是希望文件不但只是以前印象中的纯文本文件,还要可以在文件中集成其他程序的文件,使得工作表、图表,甚至是多媒体音效等都可以包含在同一份文件中。

顾名思义,OLE 的特色有:

对象链接:

将其他应用程序的文件做一份链接在复合式文件里面。以后只要其他应用程序文件做过任何改变,复合式文件也会跟着改变。例如:Word 中可以链接 Excel 的工作表,以后只要 Excel 工作表的内容有任何改变,Word 文件也会被更新。

对象嵌入:

对象嵌入技术与对象链接技术类似,不同的是,对象嵌入会「复制」一份被嵌入的文件到复合式文件中。也就是说,以后当源文件进行任何更新时,最后的复合文件并不会跟着更新。

虽然 OLE 确实对 Windows 环境用户的使用经验有深远的影响,但真正对未来技术有承前启后意义的应该是 OLE 的下一个版本——OLE2。

OLE2 最大的不同处在于 OLE2 导入了 COM (Component Object Model, 组件对象模型)。COM 是属于操作系统层级的标准,也就是说,COM 是依附在 Windows 的操作系统中的。COM 定义了对象之间的存取方法,例如:对象要怎样让别人知道自己提供了哪些方法可供调用?知道了组件提供的方法之后,要如何取用这些方法?这些都在 COM 里有所规范。

有了 COM 这套标准,以后只要是根据这套标准所开发出来的组件,就可以互相沟通运作了。

此外,这个时期也出现了新的控制组件 (control)——OLE Control (OCX)。由于传统的 VB Control (VBX) 难以开发,且仅适用于特定的开发环境 (Visual Basic),以至于不容易推广。在长远的角度考虑下,COM 上场了。OLE Control 基于 COM 的架构上开发,可适用于微软所有的开发环境,如 Visual C++、Visual Basic 等。其他开发软件厂商、如 Borland 等,也相继开始支持这种新的控制组件。

1995 年到 1996 年，因特网以出人意料的速度发展，同时造就了许多新兴的事业与公司。其中给人印象最深的要数 Netscape 了。Netscape 以开发 WWW 浏览器（Browser）——Netscape Navigator 起家，在两年的时间内迅速窜起，成为因特网上最知名的软件公司。在同时期，SUN 也以 Java 拓展了自己的另一个事业版图。在 SUN 与 Netscape 的相互应合之下，仿佛微软这个软件帝国的霸主地位，正逐渐地受到怀疑及无情的挑战。

1996 年，是微软反败为胜最为关键的一年。这个时候，微软终于开始正视因特网的威力了。在网络用户端方面，微软以免费的方式发行自家的浏览器（说是自家，其实还是并购来的）——Internet Explorer，来打击 Netscape。在网络服务器方面，微软在 Windows NT 4.0 中，免费附加了 IIS（Internet Information Server）2.0，内附有 WWW、FTP 以及 Gopher 等服务器，同样是希望能够以免费的手段来取得市场占有率。但最重要的技术革新，还是 ActiveX。

ActiveX 刚推出的时候，它的名字甚至比它的技术更引人好奇。ActiveX 乍听之下会让人以为是某个新技术的名字。不过事实上，ActiveX 应该更像是一种品牌的名字。就像联合利华（Unilever）销售多芬（Dove）的清洁用品一样。微软生产并推销 ActiveX 品牌的开发工具及技术。

ActiveX 就像是一把大伞，把所有以 COM 为基础的技术都纳入它的旗下。例如：OLE Control 演变成现在的 ActiveX Control，OLE 的 OLE Document（微软的 Compound Document）演变成 ActiveX Document。因此只要是基于 COM 发展出来的技术，就是 ActiveX 的管辖范围。

此外，这个时期的网络平台也是多彩多姿的。Java 的 Java Applet 使得传统的网页除了文字与图形外，更多加了许多互动的功能。按照 SUN 的说法，真的是使网页“活”起来了。而 ActiveX Control 在此时也做了若干修正，微软发行了 ActiveX Template Library（ATL），可以简化 ActiveX Control 并缩小 ActiveX Control 的大小，使得 ActiveX Control 可以轻易地放在网页内容上，让网页俨然成为新一代的复合式文件。

1.2.3 全面网络化——DNA 平台

之前微软在因特网应用上所做的努力，已获得若干成效。Internet Explorer 的占有率，也终于能够与 Netscape 并驾其驱，并且有取而代之的气势。这个时候，微软缺的就是一个全面性的网络整合平台。

1999 年初，微软推出了在 Windows NT 平台上的 Windows NT 4.0 Option Pack，其中包含了 IIS 4.0、MTS（Microsoft Transaction Server）、ASP（Active Server Pages）、信息队列

(MSMQ, Microsoft Message Queue) 等功能。同时, 微软的第一个完全整合 Windows 与因特网的平台——Windows DNA 也诞生了。

Windows DNA 的全名为 Windows Distributed interNet Architecture, 是微软整合了 ActiveX (套句行话, COM)、WWW 技术 (如 VBScript、DHTML 等) 平台的总称。微软利用 IIS 以及 WWW 作为系统的前端来处理用户互动界面, 以 COM 架构作为中间层来处理企业程序, 以 SQL Server 作为后端储存数据的服务器, 以实现三层式 (three-tier) 系统开发的环境, 如图 1-2 所示。

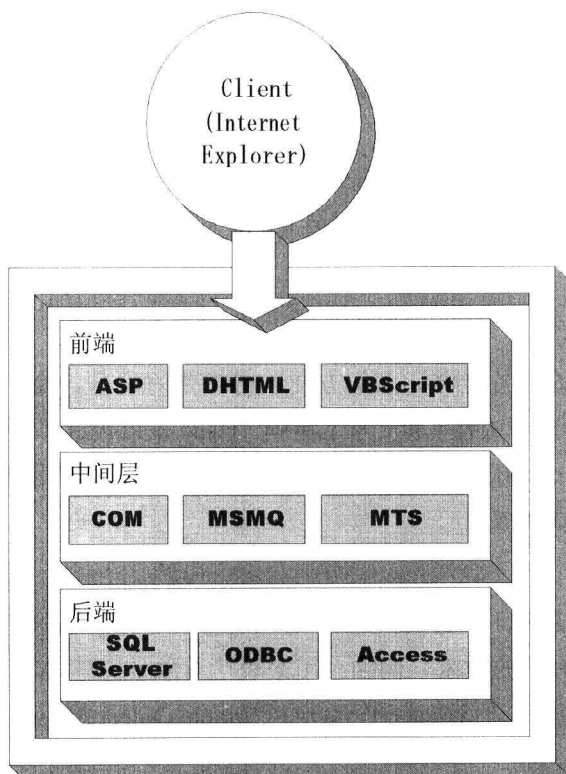


图 1-2

三层式系统开发的好处在于, 系统的界面、系统内部的程序以及企业后端的储存数据库可以分开。也就是说, 当界面、程序或后端数据库有所改变时, 不会影响到其他层的运作, 不会发生以前只要任何一层的程序有更新, 整个系统就必须重新布署的情形。再者, 因为整个系统是建构在网络上的, 因此用户端只需有个 Internet Explorer 就可以在任何地方都可以存取到系统, 以减少最烦人的用户端管理工作。