



精通 ASP.NET 2.0 + XML + CSS

网络开发 混合编程



汪孝宜 徐宏杰 胡海璐 等编著
飞思科技产品研发中心 监制

Web

ASP .NET 2.0 · XML ·
CSS · XSL · XSLT ·
SQL Server 2005 ·
AJAX · 整合应用



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



精通 ASP.NET 2.0 + XML + CSS

汪孝宜 徐宏杰 胡海璐 等编著
飞思科技产品研发中心 监制

网络开发 混合编程



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以最新的 ASP.NET 2.0 为基础,详细阐述了当前网络开发的经典架构 ASP.NET 2.0+XML+CSS 的各个知识点,以及 SQL Server 2005 的相关知识。全书各知识点均配以实例,按照基础知识、实战实例和综合实例的顺序,循序渐进、由浅入深地进行讲解。本书由各具特色的、不同深度的案例表达每一个关键的技术要点,同时给出多个综合的大型应用案例,全部案例源自作者多年实际 Web 开发的项目积累。

本书分为 9 章。第 1~4 章主要从基础知识入手,简要介绍:XML、CSS、XSL 和 XSLT 的基础知识,以及在 ASP.NET 2.0 中使用 XML 和 CSS 的方法。第 5~7 章主要从实战出发介绍网页数据绑定技术和实例,从应用领域、学习目的出发讲解基于 ASP.NET 2.0+XML+CSS+SQL Server 2005 技术的留言板、博客等网络常用的实例。第 8~9 章比较深入地介绍了 AJAX 的基础知识,在 ASP.NET 2.0 中的成熟架构 AJAX.NET 和 Atlas,然后结合一些常用的实例,例如进度条、投票等,介绍了在 ASP.NET 2.0 中应用 AJAX 的方法。最后一章是一个实际的软件工程开发应用实例。书中实例源文件请到 <http://www.fecit.com.cn> 的“下载专区”中下载。

本书适合于使用 ASP.NET 开发 Web 程序的技术人员,也适合具备一定 ASP.NET 基础知识的读者学习和提高,当然也可以作为很好的入门教材。

未经许可,不得以任何方式复制或抄袭本书的部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

精通 ASP.NET 2.0+XML+CSS 网络开发混合编程 / 汪孝宜等编著. —北京:电子工业出版社, 2007.1

(网站开发专家)

ISBN 7-121-03541-3

I. 精... II. 汪... III. ①主页制作—程序设计②可扩展语言, XML—程序设计 IV. ①TP393.092②TP312

中国版本图书馆 CIP 数据核字 (2006) 第 141029 号

责任编辑:王树伟

印 刷:北京天宇星印刷厂

装 订:三河市鹏成印业有限公司

出版发行:电子工业出版社

北京海淀区万寿路 173 信箱 邮编:100036

开 本:850×1168 1/16 印张:21 字数:571.2 千字

印 次:2007 年 1 月第 1 次印刷

印 数:6 000 册 定价:33.80 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系电话:(010) 68279077;邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

本书特色

本书以最新的 ASP.NET 2.0 为基础,详细阐述了当前网络开发的经典架构 ASP.NET+XML+CSS 的各个知识点,另外还附带介绍了 SQL Server 2005 的相关知识。细分起来本书具备以下特点:

- 全书各知识点均配以实例,按照基础知识、实战实例和综合实例的顺序,循序渐进、由浅入深地进行讲解。
- 详细介绍了 AJAX 的基础知识,以及在 ASP.NET 中应用 AJAX 的方法和技巧。
- 详细介绍了使用 ASP.NET+XML+CSS+SQL Server 2005 开发留言板的过程。
- 详细介绍了使用 ASP.NET+XML+CSS+SQL Server 2005 开发博客的过程。
- 应用软件工程的思想,使用 ASP.NET+XML+CSS 开发电子资料管理系统的过程。

本书内容安排

本书各章节的内容安排如下。

第 1 章	XML 快速入门	详细介绍了 XML 的基础知识
第 2 章	在 ASP.NET 中使用 XML	详细介绍了在 ASP.NET 中操作 XML 的方法
第 3 章	CSS 快速入门	详细介绍了 CSS 的基础知识
第 4 章	XSL 和 XSLT 快速入门	详细介绍了 XSL 和 XSLT 的基础知识,结合通信录实例,介绍了使用 XSLT 格式化 XML 的方法
第 5 章	如何在 ASP.NET 中进行 XML 实战	介绍了 ASP.NET 2.0 特色之一:XML 菜单。介绍了 XML 导航菜单和目录树的制作方法,并结合 XSLT 和 CSS 介绍如何实现网站的个性化:变脸
第 6 章	ASP.NET+XML+CSS+ SQL Server 2005 综合应用:留言板	详细介绍了使用 ASP.NET+XML+CSS+ SQL Server 2005 开发留言板的过程
第 7 章	ASP.NET+CSS+XML 从留言板升级到博客	详细介绍了使用 ASP.NET+XML+CSS+ SQL Server 2005 开发博客的过程
第 8 章	高级应用——AJAX 技术在 ASP.NET 中的应用	介绍了 AJAX 的基础知识,以及在 ASP.NET 中的成熟架构 AJAX.NET 和 Atlas,详细介绍了应用 AJAX 实现进度条、投票、联动下拉列表等的过程
第 9 章	ASP.NET+SQL Server 2005+XML+CSS 电子资料管理系统	应用软件工程的思想,完整地介绍了一个实际 Web 应用:电子资料管理系统的开发过程

你适合看本书吗

本书适合于使用 ASP.NET 开发 Web 应用程序的技术人员及对相关技术感兴趣的读者,同时也非常适合具备一定 ASP.NET 知识的读者学习和提高。

虽然本书并不是一本 ASP.NET 的教程或参考书,但读者只需要具备 ASP.NET 的基本知识,就可

以学习本书的案例。在本书的编写过程中，我们充分地考虑了初学者可能会遇到的问题，因此，在书中穿插讲述了许多应用的小技巧。

本书技术支持

如果读者在阅读本书的过程中遇到任何问题，可以访问我们的技术支持网站 www.hbook.cn，我们将在 5 个工作日内安排专人解决读者所遇到的每个问题。

致谢

本书由电子工业出版社飞思科技产品研发中心策划，由指针工作室开发团队编写测试所有系统，参加本书编写的有汪孝宜、徐宏杰、胡海璐、王景春等。另外王隽和胡俦在程序测试和书稿校对中做了很多的工作，在此对他们的辛勤工作表示感谢。

在本书的编写过程中，我们力求精益求精，但难免存在一些错误和不足之处，敬请广大读者批评指正。

编著者

联系方式

咨询电话：(010) 68134545 88254160

电子邮件：support@fecit.com.cn

服务网址：<http://www.fecit.com.cn> <http://www.fecit.net>

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

第 1 章 XML 快速入门	1
1.1 初识 XML	1
1.1.1 XML 定义	1
1.1.2 XML 与 HTML、SGML 之间的关系	2
1.1.3 XML 的优点	3
1.2 Web 开发模式: ASP.NET+XML+CSS 开发	4
1.2.1 传统的 Web 开发模式	4
1.2.2 基于 XML 的 Web 体系结构	4
1.2.3 应用前景	5
1.3 XML 基本知识点	6
1.3.1 XML 语法	6
1.3.2 XML 元素	8
1.3.3 XML 属性	8
1.3.4 形式完备的和合法有效的 XML 文档	9
1.3.5 XML 的命名空间	10
1.4 第一个 XML 实例: 我的藏书阁	11
1.5 XML 如何适合 .NET	17
1.5.1 符合标准	18
1.5.2 与 ADO.NET 集成	18
1.5.3 XML 设计器	19
1.6 搭建开发环境	21
1.6.1 开发 ASP.NET+XML+CSS 的 Web 环境	21
1.6.2 开发 .NET 应用下 IIS 的安装配置	22
1.6.3 ASP.NET 应用的部署	24
1.6.4 XML 解析器的安装	24
1.7 小结	25
第 2 章 在 ASP.NET 中使用 XML	27
2.1 ASP.NET 命名空间简介	27
2.2 先分析一个案例: XML 的通信录	29
2.3 在 XML 中使用 DOM 模型	31
2.3.1 XmlNode 类: 节点	31
2.3.2 XmlDocument 类: 文档	32
2.4 在 ASP.NET 中读取 XML	36
2.4.1 XmlTextReader 类: 读取信息	36
2.4.2 XmlNodeReader 类: 读取	40
2.5 在 ASP.NET 中写入 XML 文档	41
2.5.1 XmlTextWriter: 写入	41
2.5.2 利用 DataSet 读取/写入较大的数据块	45

2.6 小结	46
第3章 CSS 快速入门	47
3.1 什么是 CSS 与为什么用 CSS	47
3.2 从这个实例开始使用 CSS: CSS 样式单	47
3.2.1 注释	48
3.2.2 选择符	49
3.2.3 优先权	51
3.2.4 属性	51
3.2.5 单位	59
3.3 使用 CSS 实例: 加上 CSS 的 XML 通信录	60
3.3.1 实例任务与学习目标	60
3.3.2 设计思想和设计方案	61
3.3.3 编写通信录列表页的 CSS 文件	61
3.3.4 在通信录列表页应用 CSS 样式	64
3.4 CSS 使用技巧汇聚	66
3.5 小结	70
第4章 XSL 和 XSLT 快速入门	71
4.1 XSL 和 XSLT 基本概念	71
4.1.1 XSL 基本概念	71
4.1.2 XSLT 基本概念	71
4.2 XSL 格式化对象	72
4.2.1 XSL 知识	72
4.2.2 格式化对象及其属性	73
4.2.3 XPath	76
4.3 XSLT 基础	81
4.3.1 简单实例: “Hello World”	82
4.3.2 XSLT 的元素语法	84
4.4 在 ASP.NET 中使用 XSL	89
4.5 使用 XSL+XML 显示通信录实例	92
4.6 小结	94
第5章 如何在 ASP.NET 中进行 XML 实战	95
5.1 XML 菜单技术	95
5.1.1 XML 菜单先睹为快: 浏览	95
5.1.2 XML 菜单动脑: 设计	95
5.1.3 XML 菜单动手: 开发	97
5.1.4 XML 菜单在开发环境中测试运行	101
5.2 用 XML 实现树型文件列表	102
5.2.1 文件列表一览	102

5.2.2	树型文件列表: 设计思路	102
5.2.3	树型文件列表: 开发详解	103
5.2.4	树型文件列表在开发环境中测试运行	106
5.3	大变脸之一: 我的页面我做主	106
5.3.1	大变脸: 构思	106
5.3.2	大变脸: 开工准备	109
5.3.3	大变脸: 画多个脸谱	109
5.3.4	在开发环境中测试运行脸谱	114
5.4	大变脸之二: 我的布局我做主	114
5.4.1	大变脸: 构思	114
5.4.2	大变脸: 开工准备	114
5.4.3	大变脸: 画脸谱	115
5.4.4	在开发环境中测试运行脸谱	117
5.5	小结	118
第 6 章	ASP.NET+XML+CSS+SQL Server 综合应用: 留言板	119
6.1	浏览换脸的留言板	119
6.1.1	留言板的架构	120
6.1.2	留言板的制作和组成	120
6.2	留言板需求分析	121
6.2.1	留言板的功能需求	121
6.2.2	留言板的分页需求	121
6.2.3	留言板的流程图	121
6.3	留言板的详细设计	122
6.3.1	SQL Server 2005 中的数据类型	122
6.3.2	根据留言板需求进行数据库设计	126
6.3.3	对留言板的业务逻辑层进行设计	130
6.3.4	前台页面表示层	131
6.3.5	留言板后台业务逻辑层的设计	132
6.3.6	留言板后台留言回复表示层的设计	132
6.4	在 Visual Studio 中进行留言板开发详解	133
6.4.1	建立网站	133
6.4.2	建立 DAL 类库	135
6.4.3	留言板的功能实现	152
6.5	换脸配置: XML 与 CSS	175
6.6	在开发环境中测试留言板	177
6.7	小结	178

第 7 章 ASP.NET+CSS+XML 从留言板升级到博客	179
7.1 从留言板到博客的功能	179
7.2 博客的需求	179
7.3 博客的设计	180
7.4 博客的开发	184
7.4.1 博客首页图片功能的开发	184
7.4.2 content.aspx 页面的开发	195
7.4.3 photoAlbum.aspx 页面的开发	197
7.4.4 Default.aspx 页面的开发	200
7.4.5 ManageMain.aspx 页面的开发	201
7.4.6 Desktop.aspx 页面的开发	201
7.4.7 BlogList.aspx 页面的开发	201
7.4.8 addblog.aspx 页面的开发	205
7.4.9 photolist.aspx 页面的开发	206
7.4.10 Addphoto.aspx 页面的开发	207
7.4.11 link.aspx 页面的开发	208
7.4.12 UserPassword.aspx 页面的开发	213
7.5 博客的 RSS	215
7.5.1 RSS 如何工作	216
7.5.2 RSS 新闻阅读器的特点和优点	216
7.5.3 RSS 的制作	216
7.6 博客的测试运行	217
7.7 小结	218
第 8 章 高级应用——AJAX 技术在 ASP.NET 中的应用	221
8.1 AJAX 技术基础	221
8.1.1 AJAX 基本原理	221
8.1.2 AJAX 框架：一步步实现 AJAX 技术	224
8.2 在 ASP.NET 中的成熟架构——Atlas	227
8.3 在 ASP.NET 中的成熟架构——AJAX.NET	234
8.3.1 AJAX.NET 简介	234
8.3.2 AJAX.NET 软件包的安装	235
8.3.3 AJAX.NET 框架实例	236
8.4 在 ASP.NET 中应用 AJAX 实战	239
8.4.1 实例一：计算字符串的长度	239
8.4.2 实例二：简易投票系统	242
8.4.3 实例三：可以自动补全的文本框	246
8.4.4 实例四：动态填充的下拉列表	253

8.4.5 实例五: 进度条	256
8.4.6 实例六: 自由编辑器	260
8.4.7 实例七: 联动的选择列表	263
8.5 小结	266
第9章 ASP.NET+SQL Server 2005+ XML+CSS 电子资料管理系统	267
9.1 特色突破	267
9.2 系统概述	268
9.2.1 系统功能与应用背景	268
9.2.2 系统预览	268
9.3 系统设计与架构	269
9.3.1 系统设计思想	269
9.3.2 系统架构	270
9.3.3 系统结构设计	270
9.4 数据库的设计、生成与配置	271
9.4.1 数据库需求分析	271
9.4.2 数据库概念结构设计	271
9.4.3 数据库逻辑结构设计	273
9.4.4 数据库的生成与配置	274
9.4.5 数据库的连接	274
9.5 公用模块的编写	274
9.5.1 系统配置模块	274
9.5.2 通用数据访问模块	276
9.5.3 字符串操作模块	290
9.5.4 日期操作模块	292
9.5.5 文件操作模块	293
9.5.6 JavaScript 文件	294
9.5.7 CSS 样式表	295
9.6 用户界面设计	295
9.6.1 登录模块设计	295
9.6.2 首页设计	295
9.6.3 动态树型菜单显示设计	296
9.6.4 资料查询界面模块设计	298
9.6.5 资料列表显示模块设计	300
9.6.6 资料下载模块设计	301
9.7 管理员界面设计	303
9.7.1 用户管理模块设计	303
9.7.2 资料类型管理模块设计	303
9.7.3 添加资料模块设计	304

9.7.4	资料管理模块设计	307
9.7.5	树型菜单维护模块设计	308
9.7.6	数据备份模块设计	316
9.7.7	数据恢复模块设计	320
9.7.8	退出和重登录模块设计	322
9.8	开发技巧与难点点拨	322
9.8.1	#Region 指令的使用	322
9.8.2	防止 SQL 注入	323
9.9	系统的编译和发布	324
9.10	小结	324
7.4.1	Design.aspx 页面的开发	201
7.4.2	Blog.aspx 页面的开发	202
7.4.3	addBlog.aspx 页面的开发	205
7.4.4	showBlog.aspx 页面的开发	206
7.4.5	AdminBlog.aspx 页面的开发	207
7.4.6	Link.aspx 页面的开发	208
7.4.7	UseCommand.aspx 页面的开发	211
7.5	RSS	212
7.5.1	RSS 的工作原理	212
7.5.2	RSS 新闻阅读器的工作	216
7.5.3	RSS 的部署	216
7.6	搜索引擎	217
7.7	小结	218
第8章	高级应用	221
8.1	1.1X 技术	221
8.2	1.1X 技术	221
8.3	1.1X 技术	221
8.4	1.1X 技术	221
8.5	1.1X 技术	221
8.6	1.1X 技术	221
8.7	1.1X 技术	221
8.8	1.1X 技术	221
8.9	1.1X 技术	221
8.10	1.1X 技术	221
8.11	1.1X 技术	221
8.12	1.1X 技术	221
8.13	1.1X 技术	221
8.14	1.1X 技术	221
8.15	1.1X 技术	221
8.16	1.1X 技术	221
8.17	1.1X 技术	221
8.18	1.1X 技术	221
8.19	1.1X 技术	221
8.20	1.1X 技术	221
8.21	1.1X 技术	221
8.22	1.1X 技术	221
8.23	1.1X 技术	221
8.24	1.1X 技术	221
8.25	1.1X 技术	221
8.26	1.1X 技术	221
8.27	1.1X 技术	221
8.28	1.1X 技术	221
8.29	1.1X 技术	221
8.30	1.1X 技术	221
8.31	1.1X 技术	221
8.32	1.1X 技术	221
8.33	1.1X 技术	221
8.34	1.1X 技术	221
8.35	1.1X 技术	221
8.36	1.1X 技术	221
8.37	1.1X 技术	221
8.38	1.1X 技术	221
8.39	1.1X 技术	221
8.40	1.1X 技术	221
8.41	1.1X 技术	221
8.42	1.1X 技术	221
8.43	1.1X 技术	221
8.44	1.1X 技术	221
8.45	1.1X 技术	221
8.46	1.1X 技术	221
8.47	1.1X 技术	221
8.48	1.1X 技术	221
8.49	1.1X 技术	221
8.50	1.1X 技术	221
8.51	1.1X 技术	221
8.52	1.1X 技术	221
8.53	1.1X 技术	221
8.54	1.1X 技术	221
8.55	1.1X 技术	221
8.56	1.1X 技术	221
8.57	1.1X 技术	221
8.58	1.1X 技术	221
8.59	1.1X 技术	221
8.60	1.1X 技术	221
8.61	1.1X 技术	221
8.62	1.1X 技术	221
8.63	1.1X 技术	221
8.64	1.1X 技术	221
8.65	1.1X 技术	221
8.66	1.1X 技术	221
8.67	1.1X 技术	221
8.68	1.1X 技术	221
8.69	1.1X 技术	221
8.70	1.1X 技术	221
8.71	1.1X 技术	221
8.72	1.1X 技术	221
8.73	1.1X 技术	221
8.74	1.1X 技术	221
8.75	1.1X 技术	221
8.76	1.1X 技术	221
8.77	1.1X 技术	221
8.78	1.1X 技术	221
8.79	1.1X 技术	221
8.80	1.1X 技术	221
8.81	1.1X 技术	221
8.82	1.1X 技术	221
8.83	1.1X 技术	221
8.84	1.1X 技术	221
8.85	1.1X 技术	221
8.86	1.1X 技术	221
8.87	1.1X 技术	221
8.88	1.1X 技术	221
8.89	1.1X 技术	221
8.90	1.1X 技术	221
8.91	1.1X 技术	221
8.92	1.1X 技术	221
8.93	1.1X 技术	221
8.94	1.1X 技术	221
8.95	1.1X 技术	221
8.96	1.1X 技术	221
8.97	1.1X 技术	221
8.98	1.1X 技术	221
8.99	1.1X 技术	221
8.100	1.1X 技术	221

第 1 章 XML 快速入门

近年来, XML 由于适于异构应用系统间的数据共享, 优越的数据存储机制, 强大的数据检索能力等优点而被广泛应用。在实际的应用系统中, 合理广泛地应用 XML 能给解决的实际应用难题提供极大的便利, 因此学习、理解、掌握 XML 的使用是很有必要的, 本章内容意在让读者通过本章的学习, 能快速掌握 XML 的优点、用途和使用方法。

1.1 初识 XML

可扩展标记语言 XML (eXtensible Markup Language) 是一门新兴的面向 Internet 应用的标记语言, 是为在 Web 上使用而进行优化了的 SGML (标准通用标记语言) 的子集。它是由 WWW 联合会 (W3C) 于 1998 年 2 月制定的一种通用语言规范。该标准确保了结构化数据的统一性和相对于应用或供应商的独立性。

作为一种可用来制定具体应用语言的元语言, XML 既具有强大的描述能力, 又具有适合网络应用的简捷性。作为对 SGML 语言标准的一种改良, XML 具有适于异构应用系统间的数据共享, 可以进行数据检索和提供多语种支持等优点。而这些优点均与 XML 的数据存储机制紧密相关。

XML 是一种提供数据描述格式的标记语言。该语言使跨越多个平台进行更准确的内容声明和获得更有意义的搜索结果变得更加容易。此外, XML 还实现了数据表示与内容的分离。例如, 在 HTML 中, 使用标记来告诉浏览器将数据显示为粗体或斜体; 而在 XML 中, 标记只用于描述数据, 例如城市名、温度和大气压等。在 XML 中, 使用样式表 (例如, 可扩展样式表语言 (XSL) 和层叠样式表 (CSS)) 在浏览器中显示数据。XML 使数据内容与数据表示及处理分离开来, 通过应用不同的样式表和应用程序, 使用户能够根据需要显示和处理数据。

在 .NET 领域中, XML 是传输数据的默认格式, 也是 Visual Studio.NET 和 .NET 框架众多功能的核心。因此理解其基本知识至关重要。

1.1.1 XML 定义

XML 是一种界定文本数据的简便而标准的方法, 曾经被人称为 “Web 上的 ASCII 码”。就好像程序员可以使用自己喜爱的编程语言来创建任意一种数据结构, 然后同其他人在其他计算平台上使用其他语言来共享一样。XML 的标记用来说明程序员所描述的概念, 而属性则用来控制其结构。然后, 程序员可以定义自己所设计出来的语法并同其他人共享。

例程 1-1 是一个简单的 XML 文件的例子。

例程 1-1 XML 文件示例

```
<联系人>
<姓名>张三</姓名>
  <ID>001</ID>
  <公司>A 公司</公司>
  <EMAIL>0000@XXX.com</EMAIL>
  <电话>(010)00000008</电话>
  <地址>
```

```

<街道>五街 1234 号</街道>
<城市>北京市</城市>
<省份>北京</省份>
</地址>
</联系人>

```

XML 的数据描述机制意味着其将成为一种在 Internet 上共享信息的强大途径, 因为:

- XML 是开放的, 能够在不同的用户和程序之间交换数据, 而不论其平台如何。
- 自描述的特性使其对于 B2B 和企业内部网解决方案来说是一种有效的选择。
- 无需事先协调, 即可在程序之间共享数据。后面将看到, XML 的机制使程序能够找出一类 XML 文档的结构。

为了使用 XML 文档, W3C 为 XML 标准化了一套应用程序编程接口, 这样即可轻松地编制读写 XML 的程序, 同时, 开发者团体还设计了一套特殊的, 免费赠送的, 基于事件的替代 API。此外, XML 在设计时已准备支持非欧洲语言和进行国际化。同 HTML 4.01 一样, XML 基于的是在 ISO/IEC 10 646 字符集标准中定义的通用字符集 (UCS)。可以说, 所有促使 HTML 得以流行的特性均出现在了 XML 当中。

但是, XML 并非 HTML 的直接替代品。在 XML 推荐标准 (WWW 协会的对等标准) 中, 并没有发现任何与可视化表现形式有关的内容。与注重数据及其表达方式的 HTML 不同, XML 只关心数据本身。

虽然 XML 本身是数据, 但 XML 的研究者并没有忘记如何表示数据的问题。与依靠扩充代码部分的传统数据表示方法不同, XML 样式的表现技术是由数据驱动的。从最简单的到极端复杂的, 无所不能。但是, 不论采用什么技术, XML 的样式化均通过样式表来实现。在其中, 设计者会格式化样式并决定何时应用样式的规则。

1.1.2 XML 与 HTML、SGML 之间的关系

HTML 是 SGML 的一个实例, 其 DTD 作为标准被固定下来, 因此, HTML 不能作为定义其他标记语言的元语言。XML 是 SGML 的一个子集, 严格而言, XML 也还是 SGML。与 HTML 不同的是, XML 有 DTD, 因而也可以像 SGML 那样作为元语言来定义其他文件系统, 或称其他标记语言。如果把标记语言分为元标记语言和实例标记语言, SGML 和 XML 都是元标记语言, 而 HTML 和由 XML 派生的 XHTML 都是实例标记语言。

XML 提供了一种描述结构化数据的方法。与主要用于控制数据的显示和外观的 HTML 标记不同, XML 标记用于定义数据本身的结构和数据类型。XML 使用一组标记来描述数据元素。每个元素封装的数据可能十分简单也可能十分复杂。用户可以定义一组无限制的 XML 标记。例如, 可以定义一些 XML 标记来声明订单中的数据, 如价格、税收、发货地址和账单等。

XML 来源于 SGML, 是 SGML 的一种简化。为实现这一点, XML 重新定义了一些 SGML 的内部值和参数, 并删去了大量繁杂的、不常用的、使编程复杂的特征。例如, 在标准 SGML 文档中, 所有用到的标记、元素和元素的属性都必须由文件类型定义 (DTD) 来说明。但是这种限制在 XML 的应用中并不总是需要的, 所以, XML 文档被设计成既可以带有 DTD, 也可以没有 DTD 文件。由于 XML 舍弃了 SGML 的复杂性, 因此编写处理 XML 的应用程序会很容易。

XML 是一种简单的、与平台无关并被广泛采用的标准, 相对于 HTML 来说其优点是将用户界面与结构化数据分隔开来。这种数据与显示的分离使得集成来自不同源的数据成为可能。客户信息、订单、研究结果、账单付款、病历、目录数据及其他信息均可转换为 XML。

以下是理解 XML 与 HTML 之间关系的几个要点。

- XML 不是要替换 HTML，实际上 XML 可以作为 HTML 的补充，是 HTML 的高层扩展。XML 和 HTML 的目标不同。HTML 的设计目标是显示数据并集中于数据外观，面向文本和信息发布，允许混乱；而 XML 的设计目标是描述数据并集中于数据的内容，面向数据和数据处理，要求工整（well-formed）合法（Valid）。用户可用 XML 创建自己的 HTML。
- 与 HTML 相似，XML 不进行任何操作。虽然 XML 标记可用于描述订单之类的项的结构，但不包含可用于发送或处理该订单及确保该订单交货的任何代码。其他人必须编写代码来实际对 XML 格式的数据执行操作。
- 与 HTML 不同，XML 标记由架构或文档的作者定义，并且是无限制的。HTML 标记则是预定义的，只能使用当前 HTML 标准所支持的标记。

1.1.3 XML 的优点

XML 使用标记来标记内容以传输信息。标记用于界定内容，而语法允许用户自行定义任意复杂度的结构。

而这一切都使用普通的文本，而不是二进制的格式。对于跨平台的大量数据交换和存储而言，XML 是一个非常理想的解决方案。每种操作系统（只要不是嵌入式的）都能够以某种方式处理文本。可以预见 XML 将会成为描述各类数据内容的流行标记语言。HTML 能够在很短的时间内流行起来的最重要原因就是每一种浏览器都能处理它。

XML 具有强大的生命力，就在于其可以根据需要任意地扩展。同时，由于该扩展机制是标准化的，所以可以自动地将这类扩展信息传递给任何读取数据的实体——编程者或机器。

1. 适于异构应用系统间的数据共享

XML 的灵活性和扩展性使其可以对不同应用，甚至是差异很大的应用系统间的数据进行描述，尤其是对于那些专用于记录数据的应用。另外，XML 具有自我描述的特性，结果是数据可以在不同的应用系统间进行交换与处理，而不必要求相应的应用程序是针对该数据定制的。

2. 强大的数据检索能力

XML 属于元标记语言。用户只要在 XML 的文档类型定义文件中定义一系列有意义的标记，即可按照任意的条件进行查询和检索，甚至实现计算机自动检索。而相应的检索引擎可以是通用的而不必局限于具体的应用。

3. XML 数据存储机制

XML 有 3 种数据源：一是 XML 纯文本文档；二是关系型数据库；三是来源于其他各种应用数据；如邮件、目录清单或商务报告等。其中，第 1 种是最基本的也是最为简单的。将数据存储于文件中，最大的优点在于可以直接方便地读取，或者以样式信息在浏览器中显示，或者通过 DOM 接口编程同其他应用相连。第 2 种数据源是对第 1 种的扩展，目的是便于开发各种动态应用，优点则在于通过数据库系统对数据进行管理，然后再利用服务器端应用（如 ASP、JSP、Servlet）进行动态存取。这种方式最适合于三层结构的应用开发。

在存储信息时，XML 相对于其他格式的优越性如下。

- XML 格式是基于文本的，更容易读，更便于记录，有时也更便于调试。
- XML 文档可使用已为 HTML 建立的很多基础结构，包括 HTTP 协议和某些浏览器。HTTP 允许穿过防火墙传输 XML。
- XML 分析已有完善的定义，且已广泛应用，使得在各种环境中从 XML 文档检索信息成为可能。

- 应用程序可依赖 XML 分析器进行某些结构验证及数据类型检查（当使用架构时）。
- XML 建立在 Unicode 基础上，使得创建国际化文档更容易。

注意：XML 并不是万能的，其文档往往比所替换的二进制格式详细，占用更多的网络带宽和存储空间，或需要更多的处理器时间进行压缩。XML 分析可能比分析高度优化的二进制格式慢，并且可能需要更多内存。不过，这些问题可以通过良好的应用程序设计来避免。

1.2 Web 开发模式：ASP.NET+XML+CSS 开发

1.2.1 传统的 Web 开发模式

传统的 Web 开发模式多采用三层架构设计，它包括：表现层、数据访问层（含业务逻辑）和数据层。

- 表现层为客户端提供应用程序的访问，在 ASP.NET Web 应用中通常为带有 CSS、JavaScript 的 Aspx 页面和 Htm 页面。
- 数据访问层为数据层提供数据服务，为表现层提供带有业务逻辑的封装数据，在 ASP.NET Web 应用中通常为 ADO.NET。
- 数据层采用传统数据库实现，通常为 SQL Server、Oracle、MySql 或 Access 等商业数据库应用。

传统的 Web 开发模式结构如图 1-1 所示。

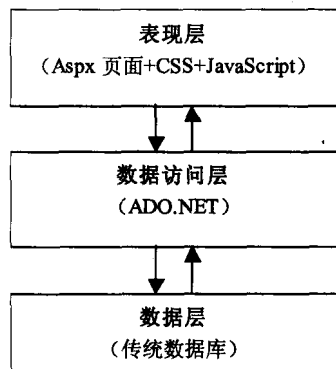


图 1-1 传统 Web 开发模式结构图

传统的 Web 开发模式采用数据库来存储数据，存在以下问题。

- 不同的数据库系统使用不同的格式存储数据，在不同的系统间进行数据交换极为复杂。
- 存储在数据库系统中的数据需要额外的应用支持才能应用于 Web 应用。

1.2.2 基于 XML 的 Web 体系结构

基于 XML 的 Web 应用从逻辑上可分成 4 个层次，包括：表现层、数据层、逻辑处理层和 XML 解析层，如图 1-2 所示。

- 表现层采用 XML 技术结合 CSS 和 JavaScript 实现。基于结构化的特性，XML 可方便有效地表示结构化的数据，因此，可采用 XML 来表示数据。CSS 可储存并控制元素的显示样式。
- XML 解析层主要包括 XML 解析器，其可读入 XML 文档或数据，将 XML 转换成便于应用程序处理的格式。由于 XML 是基于文本的，具有平台中立性，各种语言均有相应

的 XML 解析器。VS.NET 2005 自带的解析器为 MSXML 6.0。

- 逻辑处理层为 ASP.NET 页面，页面从后端数据层中或 XML 格式存储的数据中获取数据，加工后再通过 XML 解析层，以 XML、XSL、XSLT 的形式返回客户端，而实际上客户端的请求格式也是 XML 显示的。在该模式中，ASP.NET 页面充当了逻辑控制、数据处理的角色。
- 数据层采用传统数据库或 XML 数据实现。XML 是层次结构的，可以方便地对非关系型数据进行编码，其可以看成是数据库中的数据区，DTD 或者 Schemas 可以看成是数据库模式设计，XQL 可以看成是数据库查询语言，SAX 或 DOM 可以看成是数据库处理工具。因此，在某些应用中，XML 可作为整个 Web 应用的数据源，代替传统的关系型数据库的部分功能。

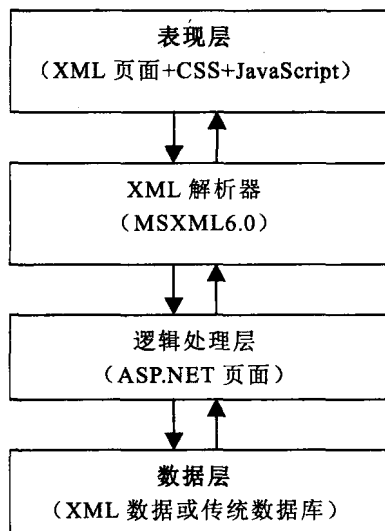


图 1-2 基于 XML 的 Web 开发模式结构图

注意：虽然 XML 和数据库具有类似的特性，但是 XML 不会取代传统的数据库，因为 XML 与数据库管理系统相比，其本身不具有安全性、事务处理、数据完整性、触发器、多用户处理机制等特性，而且在数据处理的效率上也低于数据库管理系统。

XML 和数据库系统是相辅相成的关系，正如 SQL Server 2005 一样，将 XML 集成到传统的关系数据库系统中，数据的存储和管理仍使用数据库管理系统，但是提供基于 XML 的数据访问方式。

XML 特性决定了其可使用简单的协议控制松散的、开放的资源集合。基于 XML 的 Web 应用架构，具有跨平台、开放、易扩充、可维护性强的优点。该模式可提高开发效率，节省开发时间。

1.2.3 应用前景

目前越来越多的设备都开始支持 XML。这使得 Web 开发商可以使用 XML 来建立丰富的多层 Web 应用。

1. 集成不同数据源

XML 文档可以用来描述包含在不同应用中的数据，从 Web 页面到数据库记录等，Web 应用的中间层服务程序将这些用 XML 表示的数据组合起来，提交给客户端或下一步的应用。XML 还提供了客户端包含机制，可以将多个来源的数据集成在一个文档内显示。

XML 对于大型和复杂的文档是理想的，因为数据是结构化的。这不仅使用户可以指定一个定义了文档中的元素的词汇表，而且还可以指定元素之间的关系。例如，如果要销售客户的地址一起放在 Web 页面上，就需要有每个客户的电话号码和电子邮件地址。如果向数据库中输入数据，可确保没有漏下的字段，当没有数据输入时还可提供一个默认值。XML 也提供客户端的包括机制，可以根据多种来源集成数据并将其作为一个文档来显示。数据还可以立刻进行重新排列。数据的各个部分可以根据用户的操作显示或隐藏。当处理大型的信息仓库时，比如关系型数据库是极为有用的。

XML 文档可以包括与其他文档和资源的超链接。这些链接是根据 XLink 规范创建的。XLink 用 URI（理论上）或是用 URL（实际上）标识出链接的文档。一个 XLink 可进一步指定其所链接文档的个别部分。该个别部分是通过 XPointer 来寻址的。如果打算由人来阅读 XML 文档，那么样式单就提供个别元素格式化的指令（并不是所有的 XML 文档都如此）。样式单可用几种样式语言中的任意一种来编写。CSS 和 XSL 是两种最常用的样式语言，虽然也存在其他基于 XSL 的样式语言，如 DSSSL（Document Style Semantics and Specification Language，文档样式语义和规格语言）。

2. 本地计算

XML 数据传输到客户端后，客户端可以利用 XML 分析器对数据进行解析和操作，以完成系统所需功能。从而合理地分配客户端和服务器的负荷。例如，常规的客户/服务器方式为了获得数据排序或更换显示格式，必须向服务器发出申请；而使用 XML 则可以数据库记录直接传输到客户端，再进行排序，实现数据的直接处理，避免了向服务器申请“查询—返回”结果这样的双向“旅程”，同时，在客户机设备上也无需配置数据库。还可以对设备上的 XML 文件进行修改并将结果返回服务器。

3. 数据的多种显示

由于 XML 将内容与显示分离。XML 只描述数据的结构和语义，显示外观是通过 CSS 或 XSL 进行描述。因此只要简单地在显示时配置不同的 CSS 或 XSL 文件即可实现多种显示效果。

1.3 XML 基本知识点

1.3.1 XML 语法

XML 文档和 HTML 的代码类似，也是用标识来标识内容。一个合法的 XML 文档包括以下语法要素。

- 序言。表示 XML 数据的开始，说明字符编码方法并提供 XML 分析器和应用程序的其他信息。序言包括一个可选的 XML 声明、零个或多个注释、处理指令和空格字符。
- XML 声明。所有的 XML 文档都可以（并应该）使用一个 XML 声明来开始。XML 声明说明有一个 XML 文档并提供版本号。
- 注释。用于增强文档可读性的可选实体。可以在 `<!--` 和 `-->` 字符之间键入“--”之外的字符。
- 处理指令。将指令（例如，如何处理元素或其内容）传递给应用程序。
- 属性。提供用于在元素中包含其他信息的另一种方法。在通常情况下，将要显示的大多数信息存储在元素的上下文中。然后，将元素的各种特性（不一定要显示）存储在一个属性中。