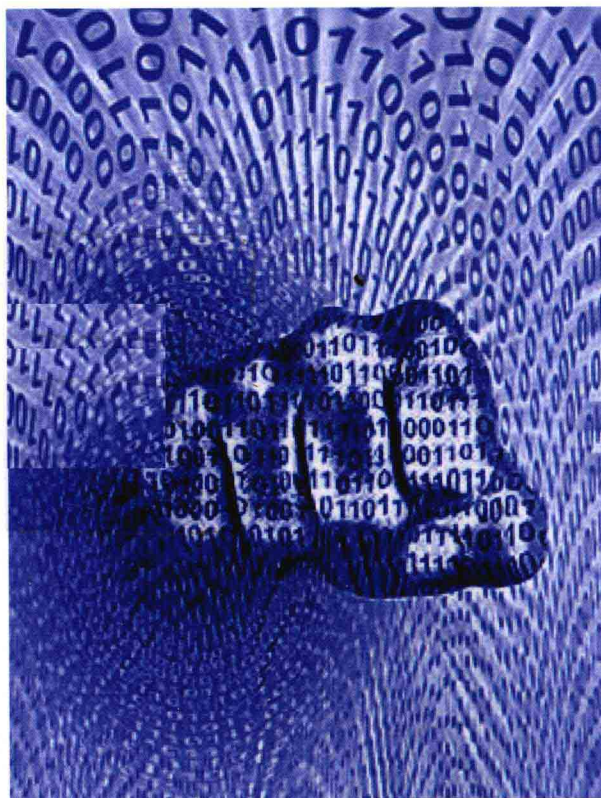


XML编程与应用教程

- ◆ XML基础及语法
- ◆ XML文档类型定义
- ◆ XML模式定义
- ◆ XML和XPath
- ◆ XML DOM编程模型
- ◆ .NET中的XML编程
- ◆ XML数据操作
- ◆ XML编程应用实例



孙更新 肖冰 彭玉忠 编著



清华大学出版社

高等学校计算机应用规划教材 第 1 卷

XML 编程与应用教程

孙更新 肖冰 彭玉忠 编著

清华大学出版社

北 京

内 容 简 介

可扩展标记语言(XML)是一种新的 Web 开发辅助语言,利用它可通过 Internet 进行信息的描述、交换和显示。本书是学习和应用 XML 语言的实用教材,书中详细阐述了 XML 的基本概念、语法规则、文档类型定义、模式定义、级联样式表、可扩展样式表、与数据库的集成、文档对象模型,还介绍了 XML 在 Java 和 .NET 中的编程,最后通过一个综合案例演示了 XML 在实际项目开发中的应用。本书内容由浅入深,在讲解基本概念和基础知识的同时给出了大量实例,便于读者消化吸收所学内容。每章还包括了小结和习题,便于读者巩固所学的知识。

本书可作为高等院校计算机、电子商务以及信息类相关专业的教材,也可供相关研究人员、广大 Web 应用程序开发者和用户参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

XML 编程与应用教程/孙更新 肖冰 彭玉忠 编著. —北京:清华大学出版社,2010.5
(高等学校计算机应用规划教材)

ISBN 978-7-302-22348-1

I. X… II. ①孙… ②肖… ③彭… III. 可扩展语言, XML—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2010)第 050376 号

责任编辑:刘金喜 鲍 芳

装帧设计:孔祥丰

责任校对:胡雁翎

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京顺义振华印刷厂

装 订 者:三河市溧源装订厂

经 销:全国新华书店

开 本:185×260 印 张:21.5 字 数:496 千字

版 次:2010 年 5 月第 1 版 印 次:2010 年 5 月第 1 次印刷

印 数:1~5000

定 价:30.00 元

产品编号:023810-01

前 言

XML 是 Extensible Markup Language 的简写,译为可扩展标记语言。XML 已经在 Web 编程、新型数据库系统、计算机网络应用编程、网络数据交换和跨平台编程中发挥越来越重要的作用。XML 正在成为电子商务运营和数据管理的核心技术。

本书是一本学习 XML 基本概念和基础理论、掌握 XML 开发技术的实用性书籍,较为全面地介绍了 XML 语言及其相关技术,在详细介绍 XML 语言及其相关标准的同时,注重 XML 技术在数据库和编程方面的实际应用,重点突出了 XML 与各种常用编程语言的结合。全书共分为 11 章,具体内容如下。

第 1 章 XML 概述:主要介绍了什么是 XML,以及 XML 产生的背景、XML 的优越性以及 XML 的常见应用等 XML 基础知识。

第 2 章 XML 语法:主要介绍 XML 的语法,重点介绍了 XML 的文档结构,XML 格式的约束规则,如何声明元素和属性,以及怎样使用名称空间等。

第 3 章 XML 文档类型定义:主要介绍确保 XML 文档有效的文档类型定义,重点介绍了 DTD 中元素、属性、实体的声明语法和格式,以及在 XML 文档中使用 DTD 的方式。

第 4 章 XML 模式定义:主要介绍了 XML 模式定义的基本语法知识,重点介绍了模式的文档结构,XML 模式中元素及属性的声明语法和使用 XML 模式的方式。

第 5 章 XML 文件的显示:主要介绍如何使用级联样式表和可扩展样式语言来显示 XML 文档,重点介绍了 XPath、XSL 模板、XSLT 语法元素等。

第 6 章 XML 和 Microsoft SQL Server 的集成:Microsoft SQL Server 2005 中可以应用 XML 技术方便地进行数据处理,本章将介绍 Microsoft SQL Server 2005 对 XML 的支持及其相互转换。

第 7 章 XLink 和 XPointer:主要介绍了 XLink 和 XPointer 的相关概念和具体使用方法。

第 8 章 XML DOM 编程模型:重点介绍文档对象模型结构以及如何使用该模型操作 XML 文档。

第 9 章 Java 中的 XML 编程:详细介绍了 Java 解析和操作 XML 文档的三种不同方式,以及在每种不同的方式下具体的编程方法。

第 10 章 .NET 中的 XML 编程:主要介绍了在 .NET 平台上操作和处理 XML 文件要用到的具体技术及相关对象,重点介绍了 .NET 平台上的 DOM 编程方法,以及与 XML 与 ADO.NET 技术的关系。

第 11 章 基于 XML 的在线相册:通过一个综合案例演示了在实际项目开发中 XML 的应用。

本课程总共有 54 学时,各章学时分配见下表(供参考)。

学时分配建议表

课 程 内 容	学 时 数			
	合 计	讲 授	实 验	机 动
chap01 XML 概述	1	1		
chap02 XML 语法	3	2	1	
chap03 XML 文档类型定义	6	4	2	
chap04 XML 模式定义——XSD	6	4	2	
chap05 XML 文件的显示——CSS 和 XSL	8	6	2	
chap06 XML 和 Microsoft SQL Server 的集成	3	2	1	
chap07 XLink 和 XPointer	6	4	2	
chap08 XML DOM 编程模型	3	2	1	
chap09 Java 中的 XML 编程	4	3	1	
chap10 .NET 中的 XML 编程	6	4	2	
chap11 基于 XML 的在线相册	8	3	3	2
合 计	54	35	17	2

本书将技术阐述与实践应用相结合,强调理论结合实际需要,全书始终以介绍 XML 中已成熟的标准和应用技术为主。书中的应用实例均来自实际开发工作中,读者可以对其稍加修改后即可直接应用到实际开发中。

本书可作为高等院校计算机、电子商务以及信息类相关专业的教材,也可供相关技术人员、广大 Web 应用程序开发者和用户参考。

本书除封面署名作者外,陈秀兰、杨光明、程长友、陈荣、周俊、于海涛、王立华、许小荣、李勇、刘峰、徐浩、李建国和刘湘等同志也参与了本书的编写,在此,对他们表示衷心的感谢。

由于 XML 相关技术标准在不断发布和更新,加之时间仓促和作者水平有限,本书中难免会有纰漏和不足之处,恳请各位专家同仁和读者批评指正。

编 者

2010 年 1 月

目 录

第 1 章 XML 概述	1
1.1 什么是 XML	1
1.2 XML 的产生背景	3
1.2.1 电子数据交换简介	3
1.2.2 XML 的产生及其与 SGML、HTML 的关系	4
1.3 XML 的优越性	6
1.4 XML 应用综述	7
1.5 XML 开发工具	8
1.5.1 Altova XMLSpy 2007 的主要功能	8
1.5.2 Altova XMLSpy 2007 的图形用户界面	12
1.5.3 Altova XMLSpy 2007 的安装	15
1.5.4 Altova XMLSpy 2007 的使用	18
1.6 本章小结	20
1.7 习题	20
第 2 章 XML 语法	22
2.1 XML 文档概述	22
2.2 XML 文档结构	24
2.3 XML 文档规则	25
2.3.1 格式良好的 XML 文档规则	25
2.3.2 格式良好的 XML 文档	29
2.3.3 有效的 XML 文档	29
2.4 XML 声明	30
2.5 XML 文档内容	34
2.5.1 XML 元素	34
2.5.2 XML 属性	36
2.5.3 注释	37
2.5.4 字符引用和实体引用	38
2.6 命名空间	40
2.6.1 命名冲突	40
2.6.2 解决命名冲突的方法	40
2.6.3 命名空间的使用	41

2.7	本章小结	43
2.8	习题	43
第 3 章	XML 文档类型定义——DTD	45
3.1	DTD 概述	45
3.2	DTD 声明	46
3.2.1	内部 DTD	47
3.2.2	外部 DTD	48
3.3	DTD 语法	49
3.3.1	元素声明	49
3.3.2	属性声明	57
3.3.3	实体声明	68
3.4	本章小结	70
3.5	习题	70
第 4 章	XML 模式定义——XSD	72
4.1	XML schema 与 DTD	72
4.2	schema 的文档结构	73
4.3	XML schema 中的数据类型	78
4.3.1	简单数据类型	78
4.3.2	复杂数据类型	84
4.4	模式文件中的元素声明	84
4.4.1	简单元素的声明	84
4.4.2	复杂元素的声明	85
4.4.3	匿名类型的定义	87
4.5	模式文件中的属性声明	88
4.6	在 XML 模式中创建元素和属性组	91
4.6.1	sequence 元素	91
4.6.2	choice 元素	91
4.6.3	group 元素	93
4.6.4	all 元素	94
4.6.5	attributeGroup 元素	95
4.7	模式重用	96
4.7.1	include 元素	96
4.7.2	import 元素	100
4.8	本章小结	103
4.9	习题	103

第 5 章 XML 文件的显示——CSS 和 XSL	107
5.1 级联样式表(CSS)	107
5.1.1 什么是 CSS	108
5.1.2 CSS 的语法	108
5.1.3 样式表与文档的链接	109
5.2 可扩展样式表(XSL)	110
5.2.1 XSLT	112
5.2.2 XPath	112
5.2.3 XSL 文档结构	124
5.2.4 XSL 模板	129
5.2.5 XSLT 的元素语法	134
5.2.6 默认的模板规则	162
5.2.7 多个样式表的合并	164
5.3 本章小结	168
5.4 习题	169
第 6 章 XML 和 Microsoft SQL Server 2005 的集成	172
6.1 Microsoft SQL Server 2005 对 XML 的支持	172
6.2 使用存储在 SQL Server 中的数据生成 XML 文档	173
6.2.1 RAW 模式	173
6.2.2 AUTO 模式	174
6.2.3 EXPLICIT 模式和 PATH 模式	174
6.3 把 XML 数据插入到 SQL Server 数据库的表中	175
6.4 SQL Server 2005 中的 XML 数据类型	179
6.5 本章小结	183
6.6 习题	184
第 7 章 XLink 和 XPointer	186
7.1 XLink	186
7.1.1 链接	186
7.1.2 简单链接	188
7.1.3 扩展链接	189
7.1.4 外联链接	191
7.1.5 扩展链接组	192
7.2 XPointer	194
7.2.1 绝对位置项	196
7.2.2 相对位置项	200
7.2.3 字符串位置项	203

7.2.4	origin 绝对位置项	204
7.3	本章小结	207
7.4	习题	207
第 8 章	XML DOM 编程模型	209
8.1	文档对象模型概述	209
8.2	XML 解析器	210
8.3	DOM 解析树	212
8.4	DOM 模型结构	214
8.4.1	DOMDocument 对象	214
8.4.2	IXMLDOMNode 对象	217
8.4.3	IXMLDOMNodeList 对象	218
8.4.4	IXMLDOMParseError 对象	218
8.5	DOM 编程步骤	219
8.6	本章小结	220
8.7	习题	220
第 9 章	Java 中的 XML 编程	222
9.1	使用 DOM 解析 XML	222
9.1.1	Java DOM 的 API	222
9.1.2	Java DOM 的应用	225
9.2	使用 SAX 解析 XML	232
9.2.1	SAX 中的事件	232
9.2.2	Java SAX 的 API	235
9.2.3	Java SAX 的应用	236
9.3	使用 JDOM 解析 XML	241
9.3.1	JDOM 的 API	241
9.3.2	JDOM 的应用	244
9.4	本章小结	249
9.5	习题	249
第 10 章	.NET 中的 XML 编程	251
10.1	使用流模式处理 XML	251
10.1.1	读取 XML	252
10.1.2	写出 XML	263
10.2	使用 DOM 处理 XML	269
10.2.1	.NET W3C DOM 类简介	269
10.2.2	使用 DOM 加载及保存 XML 数据	271

10.2.3	使用 DOM 浏览 XML	273
10.2.4	创建新节点	285
10.2.5	修改和删除节点	290
10.3	ADO.NET 与 XML	293
10.3.1	ADO.NET 简介	293
10.3.2	XML 与 DataSet 对象的关系	293
10.3.3	使用 DataSet 对象访问 XML	294
10.4	本章小结	306
10.5	习题	306
第 11 章	基于 XML 的在线相册系统	312
11.1	系统功能分析	312
11.2	系统 XML 文件的设计	313
11.3	账户管理模块的设计	316
11.3.1	管理员登录	316
11.3.2	注册账号	318
11.3.3	修改账号信息	321
11.4	相片管理模块的设计	324
11.4.1	上传相片	324
11.4.2	浏览相片	325
11.4.3	相片评论	327
11.5	留言管理模块的设计	330
11.5.1	添加留言	330
11.5.2	删除留言	332
11.6	本章小结	334

第1章 XML 概述

XML 是 Internet 环境中跨平台的、依赖于内容的技术,是当前处理分布式结构信息的选择工具,它可以简化文档信息在 Internet 中的传输。XML 不仅满足 Web 应用开发人员的需要,而且还适用于电子商务、电子政务、数据交换等多个领域。本章将介绍什么是 XML,XML 产生的背景,XML 的优越性和 XML 的应用,以及 XML 开发工具 XMLSpy 2007 的安装与使用。

本章重点:

- XML 的定义
- XML 的应用背景
- XML 开发工具

1.1 什么是 XML

XML(Extensible Markup Language,可扩展标记语言)是一套定义语义标记的规则,这些标记将文档分成许多部件并对这些部件加以标识。它也是元标记语言,可以定义其他与特定领域有关的、语义的、结构化的标记。

可扩展标记语言(XML)是 SGML(Standard Generalized Markup Language,标准通用标记语言)的子集,其目标是允许普通的 SGML 在 Web 上以目前 HTML(HyperText Mark-up Language,超文本标记语言)的方式被服务、接受和处理。XML 的定义方式易于实现,并且可以在 SGML 和 HTML 中进行操作。

XML 由 XML 工作组(原先的 SGML 编辑审查委员会)开发,此工作组由 World Wide Web Consortium(W3C)在 1996 年主持成立。工作组由 Sun Microsystems 的 Jon Bosak 负责,同样由 W3C 组织的 XML SIG(Special Interest Group,原先的 SGML 工作组)积极参与了 XML 工作组的工作。

XML 最初的设计目标如下。

- XML 应该可以直接用于 Internet。
- XML 应该支持大量不同的应用。
- XML 应该与 SGML 兼容。
- 处理 XML 文件的程序应该容易编写。
- XML 中的可选项应无条件地保持最少,理想状况下应该为 0 个。
- XML 文件应该是人们可以直接阅读的,应该是条理清楚的。

- XML 的设计应能快速完成。
- XML 的设计应该是形式化的、简洁的。
- XML 文件应易于创建。
- XML 标记的简洁性是最后考虑的目标。

下面就是一段 XML 示例文档:

```
<myfile>
<title> XML 教程</title>
<author>张三</author>
<email>icewine@tom.com</email>
<date>20090330</date>
</myfile>
```

注意:

- 这段代码仅仅能让读者感性认识 XML, 并不能实现什么具体应用。
- 其中类似<title>、<author>的语句就是自己创建的标记(Tags), 它们和 HTML 标记不一样, 例如这里的<title>是文章标题的意思, 而在 HTML 中<title>指页面标题。

XML 不同于 HTML, 超文本标记的语言定义了一套固定的标记, 用来描述一定数目的元素。例如, HTML 文档包括了格式化、结构和语义的标记。就是 HTML 中的一种格式化标记, 它使其中的内容变为粗体; <TD>也是 HTML 中的一种结构标记, 指明内容是表中的一个单元。XML 是一种元标记语言, 用户可以定义自己需要的标记。这些标记必须根据某些通用的规则来创建, 但是标记的意义, 具有较大的灵活性。例如, 若用户正在处理与学籍有关的事情, 需要描述学生的学号、姓名、年龄、家庭住址等信息, 就必须创建用于每项的标记。新创建的标记可在文档类型定义(Document Type Definition, DTD)或是 XML Schema(XML 模式)中加以描述。而关于如何显示这些标记的内容则由附加在文档上的样式表提供。

例如, 在 HTML 中, 一首歌可能是用定义标题标记<dt>、定义数据标记<dd>、无序的列表标记和列表项标记来描述的。但是事实上这些标记没有一件是与音乐有关的。用 HTML 定义的歌曲如下:

```
<dt>金曲 top1
<dd> 青花瓷
<ul>
<li>词: 方文山
<li>曲: 周杰伦
</ul>
```

而在 XML 中, 同样的数据可能标记为:

```
<SONG>
<TITLE>青花瓷</TITLE>
<COMPOSER>周杰伦</COMPOSER>
```

```
<LYRICIST>方文山</LYRICIST>  
</SONG>
```

在这段代码中没有使用通用的标记如<dt>和,而是使用了具有意义的标记,如<SONG>、<TITLE>、<COMPOSER>等。这种用法具有许多优点,包括源代码易于阅读,使人能够看出代码的含义。

1.2 XML 的产生背景

XML 最初的设计目的是为了电子数据交换,更具体地说是为电子数据交换提供一个统一的标准格式。

1.2.1 电子数据交换简介

EDI(Electronic Data Interchange, 电子数据交换)是一种利用计算机进行商务处理的新方法。EDI 将贸易、运输、保险、银行和海关等行业的信息,用一种国际公认的标准格式,通过计算机通信网络,使各有关部门、公司与企业之间进行数据交换与处理,并完成以贸易为中心的全部业务过程。

EDI 不是用户之间简单的数据交换,EDI 用户需要按照国际通用的消息格式发送信息,接收方也需要按国际统一规定的语法规则,对消息进行处理,并使其他相关系统进行 EDI 综合处理。整个过程都是自动完成,无须人工干预,减少了差错,提高了效率。因此 EDI 又被人们通俗地称为“无纸贸易”。

使用 EDI 的主要优点如下。

- 降低了纸张的消费。
- 减少了许多重复劳动,提高了工作效率。
- 使得贸易双方能够以更迅速、更有效的方式进行贸易,大大简化了订货过程和存货过程,使双方能及时地充分利用各自的人力和物力资源。
- 可以改善贸易双方的关系,厂商可以准确地估计商品的需求量,货运代理商可以简化大量的出口文书工作,商业用户可以提高存货的效率,提高了它们的竞争力。

EDI 是早期计算机网络发展的一个主要目的,而结构化信息的一个主要目的就是使数据交换成为可能。因为如果不同行业中需要交互使用的信息采用统一的模型标识,信息就能方便和高效地进行共享(对于 XML 来说,这个统一的模型就是 XSD)。而 HTML 由于自身的缺陷不能担负起这一任务,那么能不能有一种新的语言来解决这个问题呢?答案就是 XML。

相对于其他应用于 EDI 的结构化信息技术而言,XML 的优势如下。

- 低成本。XML 不需要 VAN(增值网)的高额费用,中小企业也用得起。
- 定制商业规则。XML 允许用户创建自己的商业规则和格式。
- 容易解释。EDI 需要翻译软件来得到贸易伙伴的信息格式,而 XML 通过免费下载

的解析器就可以很容易地解释 XML 文档, 实现这一需求。

- 平台独立。不管是什么平台, 都能进行交易。

1.2.2 XML 的产生及其与 SGML、HTML 的关系

XML 同 HTML 一样, 都来自 SGML(Standard Generalized Markup Language, 标准通用标记语言)。早在 Web 出现之前, SGML 就已存在。正如它的名称所言, SGML 是国际上定义电子文件结构和内容描述的标准, 是一种非常复杂的文档结构, 主要用于具有大量高度结构化数据的防卫区和其他各种工业领域, 便于分类和索引。同 XML 相比, SGML 的功能很强大, 缺点是它不适用于 Web 数据描述, 而且 SGML 软件的价格非常昂贵。SGML 十分庞大, 既不容易学, 又不容易使用, 在计算机上实现也十分困难。鉴于这些因素, Web 的发明者——欧洲核子物理研究中心的研究人员根据当时(1989 年)的计算机技术, 开发了 HTML。

HTML 只使用 SGML 中很少的一部分标记, 例如 HTML 4.0 中只定义了 70 余种标记。为了便于在计算机上实现, HTML 规定的标记是固定的, 即 HTML 语法是不可扩展的。HTML 这种固定的语法使它易学易用, 在计算机上开发 HTML 的浏览器也十分容易。正是由于 HTML 的简单性, 使得基于 HTML 的 Web 应用得到极大的发展。

但是, 随着 Web 应用的不断发展, HTML 的局限性也越来越明显地体现出来。

- HTML 是一种界面技术, 它把数据本身和数据的表现形式混在一起, 这使得分开两者变得相当困难。
- 它有一个固定的标签集。开发者不能用自己的特定应用标签来扩展它。
- 它是“平面型”的。不能使用数据的层次结构来表现数据间, 诸如包含、重要性等关系。
- 它不能把数据传送给客户端, 让客户端自行进行进一步的处理。事实上, HTML 总是在服务器端产生, 客户端只是一个显示器。
- 它只能提供一种固定显示方式给出数据。如果想提供不同的显示方式, 就需要在服务器端重新产生这些数据, 这样才能在客户端显示出另一种显示效果。如果数据和显示形式是分离的, 那么就可以把数据下载到客户端, 然后根据需要随意发送有关这些数据的不同的显示方式。
- 它的可读性不强, 无论是人读还是计算机读都是一样。HTML 也不是很严谨, 有些标签需要匹配开始符和结束符(如<body>和</body>), 但是也有一些只有开始符没有结束符(如<p>和)。HTML 解释器不得不处理这些随机的格式。
- 链路丢失后不能自动纠正。由于许多页面的 URL 地址经常变化, 当浏览这些页面时就会出现“404 URL 地址未找到”的信息。所以用户不得不手工一个个地更改链接相关页面的 URL 地址, 这大大加重了 Web 页面的维护工作量。
- 动态内容需要下载的部件太多。用 HTML 建立的页面目前还不能对其页面的外观属性(例如色彩、字体、背景)等实现更新, 只能重新下载一个新的页面。
- 搜索时间长。由于 HTML 页面没有类似于数据库的结构, 在这样的文档资料中

搜索目标时需要扫描全部页面的所有内容,往往检索出一大堆与主题词无关的内容,这是因为 HTML 无法区分信息与元信息而造成的。而且 HTML 不支持信息嵌套体系结构,因而限制了全文检索功能。

- HTML 缺乏对双字节或多国文字的支持,或者说支持不够。例如,中文信息页面在不同的平台上会出现格式不齐等问题。

尽管 HTML 推出了一个又一个新版本,已经有了脚本、表格、帧等表达功能,但始终不能满足不断增长的需求。因此,有人建议直接使用 SGML 作为 Web 页面语言,这固然能解决使用 HTML 遇到的困难,但是 SGML 太庞大了,用户学习和使用不方便尚且不说,要全面实现 SGML 的浏览器就非常困难。于是自然会想到仅使用 SGML 的子集,使新的语言既方便使用又容易实现。正是在这种形势下,Web 标准化组织 W3C 建议使用一种精简的 SGML 版本,于是 XML 应运而生了。

XML 是一个精简的 SGML 子集,它将 SGML 丰富的功能与 HTML 的易用性结合到 Web 的应用中。XML 保留了 SGML 的可扩展功能,这使 XML 从根本上有别于 HTML。XML 要比 HTML 强大得多,它不再是固定的标记,而是允许定义数量不限的标记来描述文档中的资料,允许嵌套的信息结构。HTML 只是 Web 显示数据的通用方法,而 XML 提供了一个直接处理 Web 数据的通用方法。HTML 着重描述 Web 页面的显示格式,而 XML 着重描述的是 Web 页面的内容。

XML 中包括可扩展样式表语言(Extensible Style Language, XSL)和可扩展链接语言(Extensible Linking Language, XLL)两部分。

XSL 用于将 XML 数据翻译为 HTML 或其他格式的语言。XSL 提供了一种类似 CSS 的设置页面版式的功能,使开发者构造出能表达层结构的 Web 页面来,以区别于 XML 的数据结构。XSL 也能和 HTML 一起构造排版式页面,并用 XSL 可以解释数量不限的标记,使 Web 的版面更丰富多彩,例如动态的文本、跑马式的文字。此外, XSL 还能处理多国文字、双字节的汉字显示,进行网格的各种处理等。

XLL 是 XML 的链接语言,它与 HTML 的链接相似,但功能更强大。XLL 支持可扩展的链接和多方向的链接,打破了 HTML 只支持超级文本概念下最简单链接的限制,能支持独立于地址的域名、双向链路、环路、多个源的集合链接等。另外, XLL 链接不受文档的制约,完全按用户要求来指定和管理。

为了使 XML 易学易用,XML 精简了很多 SGML 中难得用一次的功能。正如几十万汉字中常用的只不过几千,SGML 常用的部分也只占 20%,XML 抛弃了 SGML 中不常用的部分,精简了 80%。这样一来,XML 的语法说明书就只有 30 页,而 SGML 却有 500 页。

XML 的设计中也考虑了易用性,易用性表现在两个方面:一方面,用户编写 Web 页面方便;另一方面,设计人员实现 XML 浏览器也不太困难。

总之,XML 使用一个简单而又灵活的标准格式,为基于 Web 的应用提供了一个描述数据和交换数据的有效手段。但是,XML 并非是用来取代 HTML 的。HTML 着重于如何描述才能将文件显示在浏览器中,XML 与 SGML 相近,着重于如何描述才能将文件以结构化方式表示。就网页显示功能来说,HTML 比 XML 要强,但就文件的应用范畴来

说, XML 比 HTML 要超出很多。

1.3 XML 的优越性

XML 的优点主要表现在以下各方面。

- 更有意义的搜索。

数据可通过 XML 进行唯一的标识。没有 XML, 搜索软件必须了解每个数据库是如何构建的。这实际上是不可能的, 因为每个数据库描述数据的方法都不同。有了 XML, 情况就完全不同了。例如, 学生信息可以按照学号、姓名或其他的标准分类。这样搜索就变得十分方便了。

- 开发灵活的 Web 应用软件。

数据一旦建立, XML 能被发送到其他应用软件、对象或者中间层服务器以做进一步的处理, 它也可以被发送到桌面用浏览器浏览。XML 和 HTML、脚本、公共对象模式一起为灵活的三层 Web 应用软件的开发提供了所需的技术。

- 不同来源数据的集成。

现在搜索多样的不兼容的数据库实际上是不可能的, 而 XML 能够使不同来源、结构化的数据很容易地结合在一起。软件代理商可以在中间层服务器上对从后端数据库和其他应用处发来的数据进行集成。然后, 数据就能被发送到客户或其他服务器上做进一步的集成、处理和分发。

- 多种应用得到的数据。

XML 的扩展性和灵活性允许它描述不同种类应用软件中的数据, 从搜索到的 Web 页到数据记录。同时, 由于基于 XML 的数据是自我描述的, 数据不需要有内部描述就能被交换和处理。

- 本地计算和处理。

XML 格式的数据发送给客户后, 客户可以用应用软件解析数据并对数据进行编辑和处理。使用者可以用不同的方法处理数据, 而不仅仅是显示它。XML 文档对象模式(DOM)允许用脚本或其他编程语言处理数据。此外, 数据计算不需要回到服务器就能进行。可以分离使用者查看数据的界面, 使用简单、灵活、开放的格式, 给 Web 创建功能强大的应用软件, 而这些软件原来只能建立在高端数据库上。

- 数据的多样显示。

数据发到桌面后, 能够用多种方式显示。通过以简单、开放、扩展的方式描述结果化的数据, XML 补充了 HTML, 被广泛地用来描述使用者界面。HTML 描述数据的外观, 而 XML 描述数据本身。由于数据显示与内容分开, XML 定义的数据允许指定不同的显示方式, 使数据更合理地表现出来。本地的数据能够以客户配置、使用者选择或标准规定的其他方式动态地表现出来。CSS 和 XSL 为数据的显示提供了统一的且标准的机制。

- 数据可进行粒状的更新。

通过 XML, 数据可以进行粒状的更新。每当一部分数据变化后, 不需要重发整个结构化的数据。变化的元素必须从服务器发送给客户, 但变化的数据不需要刷新整个使用者的界面就能够显示出来。

- 在 Web 上发布数据。

由于 XML 是一个开放的、基于文本的格式。它可以和 HTML 一样使用 HTTP 进行传送, 不需要对现存的网络进行改变。

- 升级性。

由于 XML 彻底把标识的概念同显示分开了, 处理者能够在结构化的数据中嵌套程序化的描述, 以表明如何显示数据。这是令人难以相信的、强大的机制, 它使得客户计算机同使用者间的交互尽可能地减少了, 同时也减少了服务器的数据交换量和浏览器的响应时间。另外, XML 使个人的数据只能通过更新的布告发生变化, 减少了服务器的工作量, 大大增强了服务器的升级性能。

- 压缩性。

XML 的压缩性能很好, 因为用于描述数据结构的标签可以重复使用。XML 数据是否压缩不仅要根据应用来定, 还取决于服务器与客户间数据的传送量。

1.4 XML 应用综述

如何应用 XML 呢? 在介绍这个内容之前, 有一点必须明确, 设计 XML 的目的是用来存储数据、携带数据和交换数据的, 而不是用来显示数据。

我们可以应用 XML 进行如下工作。

- 使用 XML 从 HTML 文件中分离数据。

在不使用 XML 时, HTML 用于显示数据, 数据必须存储在 HTML 文件之内; 使用了 XML, 数据就可以存放在分离的 XML 文档中, 这种方法可以让程序员集中精力使用 HTML 做好数据的显示和布局, 并确保数据改动时不会导致 HTML 文件的改动。这样可以方便页面的维护。

XML 数据同样可以以“数据岛”的形式存储在 HTML 页面中。此时程序员仍然可以集中精力到 HTML 格式化的使用和数据的显示上。

- XML 用于交换数据。

传统的 EDI 的使用(Electronic Data Interchange, 电子数据交换)标准缺乏灵活性和可扩展性。使用 XML, 并用程序能够理解在交换数据中所表示的商务数据及概念, 并且根据明确的商务规则来进行数据处理。

- 电子商务领域。

XML 的丰富置标信息完全可以描述不同类型的单据, 例如信用证、保险单、索赔单以及各种发票等。结构化的 XML 文档发送至 Web 的数据可以被加密, 并且很容易附加上数字签名。