

高等院校
信息技术应用型
规划教材

XML应用技术

张玉宝 编著

华大学出版社

高等院校
信息技术应用型
规划教材

XML应用技术



清华大学出版社
北京

内 容 简 介

本书共分9章,包括XML概述、XML语法、使用DTD和Schema、使用CSS格式化XML、使用XSLT处理XML文档、使用客户端技术显示XML数据、使用Java技术访问XML文档、XML在AJAX技术中的应用、基于XML技术的博客系统(MyBlog),每章均配有一定数量的例题和习题,从多个方面介绍XML的应用技术。

本书内容丰富、结构合理,由浅入深、由简到繁安排整个教学内容,通过实例说明XML技术的使用,便于读者掌握。本书不仅适合作为高职高专院校及成人高等教育相关专业的教材,也可以供广大XML技术爱好者学习参考之用。

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应用技术/张玉宝编著. --北京:清华大学出版社,2013

高等院校信息技术应用型规划教材

ISBN 978-7-302-32676-2

I. ①X… II. ①张… III. ①可扩展语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2013)第122386号

责任编辑:刘士平

封面设计:傅瑞学

责任校对:袁芳

责任印制:杨艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 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

课 件 下 载: <http://www.tup.com.cn>, 010-62795764

印 刷 者:北京四季青印刷厂

装 订 者:三河市兴旺装订有限公司

经 销:全国新华书店

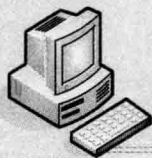
开 本:185mm×260mm 印 张:15.75 字 数:361千字

版 次:2013年8月第1版 印 次:2013年8月第1次印刷

印 数:1~3000

定 价:32.00元

产品编号:053040-01



前言

XML(eXtensible Markup Language)是一种基于文本的结构化信息描述语言,是W3C 结构化通用置标语言(Standard Generic Markup Language, SGML)的子集。它兼备 HTML 超文本置标语言(Hyper Text Markup Language)与 SGML 之长,从而更适合 Web 使用。目前,XML 是实现程序之间、用户之间、计算机与用户之间以及本地数据与网络数据之间结构化信息共享中使用最广泛的格式之一。自 1998 年 XML 标准推出以来,XML 技术受到行业的广泛关注和认同。我们相信,随着 XML 技术标准及其相关技术标准的不断完善与改进,XML 技术在 Web 应用中将占有更加重要的地位。

虽然 XML 技术标准仍然处于不断完善中,但其在数据表示、数据存储、数据显示及数据交换等数据处理技术中已经取得相当广泛的应用。本书着重从 XML 技术中已经成熟并取得广泛应用的标准和技术进行介绍,主要包括 XML 的基本语法、文档格式化描述、结构化数据表示以及 XML 在 Web 中的应用。

本书在编写过程中坚持“实用性、先进性、层次性、易读性”的原则,针对初学者的需求,由浅入深地介绍 XML 技术标准及应用。为便于读者的理解和实际应用,每章安排有大量的应用实例和练习。本书可以作为各类大专院校学生,尤其是高职高专类学生学习 XML 技术的专用教材,也可供广大程序员学习 XML 技术时参考。

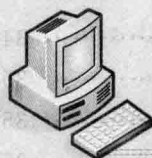
本书第 2、3、6 章由张玉宝编写,第 1、4 章由华铨平编写,第 5 章由张玉宝与华铨平合作编写,第 7、8 章由王海颖编写,第 9 章由徐兵兵编写。全书由张玉宝进行统稿。

由于编者水平有限,书中难免有不足之处,欢迎广大读者批评指正。

编者电子邮件地址为:zbaob@163.com。

编 者

2013 年 4 月



目 录

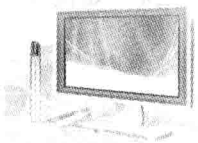
第 1 章 XML 概述	1
1.1 XML 的产生	1
1.1.1 置标语言的定义	1
1.1.2 XML 的产生	3
1.1.3 XML 概述	4
1.2 XML 的发展	6
1.2.1 HTML 的缺点和不足	6
1.2.2 XML 的优点	8
1.3 XML 应用及前景	10
1.3.1 XML 应用概述	11
1.3.2 应用于电子商务	12
1.3.3 网络出版	16
1.3.4 移动通信	20
1.3.5 XML 前景展望	23
小结	23
习题	23
第 2 章 XML 语法	25
2.1 XML 文档结构	25
2.2 文档的声明	26
2.3 处理指令 PI	27
2.4 注释	28
2.5 元素与标记	29
2.5.1 标记的构成	29
2.5.2 标记的命名规则	30
2.5.3 标记的种类	30
2.5.4 元素的嵌套	32
2.6 XML 属性	33
2.6.1 属性的构成	33

2.6.2	属性的命名	34
2.6.3	属性值	35
2.7	特殊字符处理	35
2.7.1	预定义实体的引用	35
2.7.2	CDATA 节	36
2.8	命名空间	37
2.8.1	定义命名空间	37
2.8.2	命名空间和默认命名空间	39
2.8.3	命名空间的作用范围	40
2.8.4	使用命名空间引用 HTML 标记	41
2.9	格式正确的 XML 文档	41
2.10	任务:XML 编辑器 XMLSpy 与使用	42
2.10.1	创建 DTD 文档	42
2.10.2	创建 XML 文档	43
2.10.3	从已有 XML 文档创建 DTD 或模式文件	45
2.10.4	为已有 XML 文档指定 DTD 或模式文件	46
小结		46
习题		47
第 3 章	使用 DTD 和 Schema	48
3.1	DTD 的基本结构	48
3.2	DTD 在 XML 文档中的引用	49
3.2.1	内部 DTD 引用	49
3.2.2	外部 DTD 引用	50
3.3	DTD 中的元素声明	52
3.4	DTD 中的属性声明	60
3.4.1	属性默认值的设定	61
3.4.2	属性的类型	63
3.5	任务:按 DTD 设计学生成绩管理系统中的 XML 文档	68
3.6	XML Schema 简介	70
3.6.1	XML Schema 概述	70
3.6.2	Schema 的使用	70
3.7	XML Schema 的语法	71
3.7.1	模式文件的基本结构	71
3.7.2	简单元素的定义	73
3.7.3	复杂元素的定义	73
3.7.4	Schema 中的内置数据类型	76
3.7.5	Schema 中的自定义数据类型	77

3.7.6 Schema 的引用	81
3.7.7 Schema 中的注释	81
3.8 任务:按学生成绩管理系统中的 XML 编写 Schema	82
小结	83
习题	83
第 4 章 使用 CSS 格式化 XML	84
4.1 CSS 概述	84
4.2 链接 CSS 和 XML 文档	85
4.2.1 使用 xml:stylesheet 处理指令	86
4.2.2 使用 @import 指令	88
4.2.3 CSS 样式表级联顺序	88
4.3 使用 CSS 格式化 XML 文档	89
4.3.1 选择元素	89
4.3.2 在 CSS 样式表中使用注释	92
4.3.3 CSS 中的属性和属性值	92
4.3.4 设置 display 属性	94
4.3.5 设置 whitespace 属性	95
4.3.6 设置字体属性	96
4.3.7 设置 color 属性	97
4.3.8 设置背景属性	98
4.3.9 设置文本属性	100
4.3.10 设置边框属性	102
4.4 任务:设计 CSS 样式显示学生成绩	104
小结	107
习题	107
第 5 章 使用 XSLT 处理 XML 文档	109
5.1 XSLT 基础	109
5.1.1 链接 XSLT 到 XML	109
5.1.2 XML 文档结构树	111
5.1.3 第一个 XSL 样式表	112
5.1.4 在 XML 文档中使用多个样式表	114
5.2 XSLT 模板	115
5.3 XML 文档结构描述——XPath	119
5.3.1 XPath 表达式	119
5.3.2 XPath 函数	124
5.4 XSLT 常用控制元素	127

5.4.1	循环处理元素	127
5.4.2	条件选择元素	128
5.4.3	排序元素	130
5.5	任务:实现对学生成绩的条件显示	131
5.6	任务:实现对学生成绩的排序	132
	小结	133
	习题	134
第 6 章	使用客户端技术显示 XML 数据	135
6.1	数据岛的一般概念	135
6.1.1	数据绑定	136
6.1.2	数据岛的限制	137
6.1.3	数据岛的使用	137
6.2	绑定 XML 元素到 HTML 标记	138
6.2.1	使用单个标记绑定显示 XML 文档	139
6.2.2	使用表格绑定显示 XML 文档	143
6.2.3	使用绑定来显示 XML 文档中元素的属性	148
6.3	使用客户端脚本访问 XML 文档	151
6.3.1	记录集	151
6.3.2	访问记录集中各个字段	153
6.3.3	遍历记录集	154
6.3.4	对记录集进行分页	157
	小结	159
	习题	159
第 7 章	使用 Java 技术访问 XML 文档	160
7.1	Java 技术对 XML 的支持	160
7.1.1	使用 DOM 解析 XML	160
7.1.2	使用 SAX 解析 XML	164
7.2	通过 JSP 编程访问 XML 文档	168
7.2.1	JSP 简介	168
7.2.2	直接处理 XML	172
7.2.3	使用 JavaBean 来处理 XML	174
7.3	任务:使用 JSP 和 XML 实现学生信息管理	178
	小结	187
	习题	187

第 8 章 XML 在 AJAX 技术中的应用	188
8.1 AJAX 技术简介	188
8.2 AJAX 技术的应用	191
8.2.1 XMLHttpRequest 对象	191
8.2.2 简单 AJAX 开发框架	192
8.2.3 使用 AJAX 实现一个简单加法计算器	194
8.2.4 使用 AJAX 实现学号搜索自动提示功能	198
8.3 任务:用 AJAX 技术实现学生成绩的修改	204
小结	211
习题	211
第 9 章 基于 XML 技术的博客系统(MyBlog)	213
9.1 系统分析	213
9.1.1 系统组件的交互关系	214
9.1.2 系统涉及相关技术和工具	214
9.2 系统功能设计	215
9.3 XML 文档结构设计	216
9.3.1 保存状态的 XML 文档	216
9.3.2 定义 XML Schema	218
9.4 用户登录模块实现	221
9.4.1 登录模块 XSLT 样式表实现	221
9.4.2 登录模块控制器实现	223
9.5 博文提交模块实现	225
9.5.1 博文提交模块 XSLT 样式表实现	225
9.5.2 博文提交控制器实现	228
9.6 博文浏览模块实现	230
9.6.1 博文浏览 XSLT 样式表实现	230
9.6.2 博文浏览模块控制器实现	234
9.7 博文管理模块实现	235
9.7.1 博文管理模块 XSLT 样式表实现	235
9.7.2 博文管理模块控制器实现	237
小结	241
习题	241
参考文献	242



第 1 章 XML 概述

本章要点

- 置标语言的定义
- XML 的概念
- HTML 的缺点和不足
- XML 的优点
- XML 的应用及前景

XML(eXtensible Markup Language)是一种可扩展的元置标语言,其设计动机是克服超文本置标语言(Hyper Text Markup Language, HTML)的种种不足,将网络传输的文档规范化,并赋予标记一定的含义,同时保留 HTML 具有的简捷、适于网络传输和浏览的优点。因此,XML 最终以标准通用置标语言(Standard Generalized Markup Language, SGML)子集的形式出现,它集 SGML 和 HTML 的优点于一身,具有易于编辑、便于管理、适于存档、容易查询等诸多优势,已经成为互联网技术发展的又一个亮点。

1.1 XML 的产生

1.1.1 置标语言的定义

在介绍 XML 之前,有几个名词概念需要先介绍一下。“XML、SGML、HTML”中的“ML”(Markup Language)中文含义是“置标语言”。什么是置标语言? XML 与 SGML、HTML 又有什么渊源?

“置标”是指为了处理的目的,在数据中加入的附加信息。“置标语言”是指运用置标方法描述形式语言。其实,“置标”的概念在现实生活中还是比较常见的。例如,在一段书面语言中,为了标注某一语句的重要性,在这条语句下面画上下划线,这就是一种“置标”。这种通过下划线的置标方法是一种图形化的置标。下划线称为置标的标记,如例 1-1 所示。

例 1-1 用图形标记置标

运用置标方法描述的形式语言,就称为置标语言。

用图形标记置标有一个缺点,就是标记的含义不明确,带有二义性。下划线标记既可

以理解为重点强调标记,也可以理解为名词标记。为了更准确地标识置标的含义,直接用文字做标记是一种很有效的办法。针对前面的例 1-1,改用文字作为标记后,如例 1-2 所示。

例 1-2 用文字作为标记

运用置标方法描述的形式语言,就称为<重要>置标语言</重要>。

例 1-2 中的“<重要>”称为起始标记,“</重要>”称为结束标记。这种用文字给出的标记不仅含义明确,而且便于计算机处理。

“置标”在计算机世界中应用甚为广泛。文字编辑器借助置标定义格式与外观;通信程序依靠置标理解线路上所传输信息的语义;数据库通过置标将数据字段与一定的含义相连,并表明字段之间的关系;多媒体应用中,置标则用来标识图像的源数据、声音的源数据。

从形式语言的角度讲,含有标记的书面自然语言,还不能算是置标语言。一般来说,程序设计语言对语言中出现的语句、变量、表达式等都有严格的形式化的定义。置标语言虽然不是一般意义上的程序设计语言,但与程序设计语言一样,是有严格语法定义的形式语言。

讲到置标语言,不能不提及标准通用置标语言(SGML)。SGML 的前身是 IBM 公司为解决公司内部大量文档的交换和存储,于 1969 年发明的通用置标语言 GML (Generalized Markup Language)。经过十几年的完善和改进,由 GML 发展成为 SGML,并在 1986 年被国际标准化组织公布为国际标准——ISO 8879。

SGML 是一个可以定义其他置标语言的元置标语言。通过 SGML 定义出来的置标语言实例有很多,但最知名、最流行的,当属在互联网上描述数据表现的 HTML。HTML 定义了一系列的标记,每个标记表明了数据的一种显示格式。被置标后的文档(即同时包含了纯文本和关于文本显示格式标记的文档)由一个 HTML 处理工具(最常见的是浏览器)进行读取,然后再根据标记所代表的显示规则来加以显示。

下面,让我们通过一个例子来说明 HTML 中的置标是如何发挥作用的。在 HTML 中,标记的含义是要求 HTML 浏览器将一段文本加粗表示,而标记<CENTER>的含义是告诉浏览器将这段文本在一行的中间显示。因此,在浏览器中会将以下带有置标的数据以粗体居中显示。

```
<center><b>浙江纺织服装职业技术学院</b></center>
```

同样,例 1-3 这一段 HTML 代码显示了一个学生的信息列表。

例 1-3 一个学生的信息列表的 HTML 代码

```
<ul> 200120101 </ul>
<li>季慧奇</li>
<li>女</li>
<li>01 信管 1 班</li>
<li>1985-1-2 </li>
```

这段 HTML 置标数据在浏览器中的显示效果如图 1-1 所示。

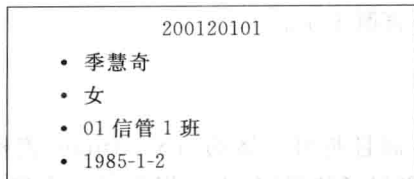


图 1-1 HTML 置标数据显示结果

1.1.2 XML 的产生

了解一下 XML 的发展简史,对 XML 入门者来说,是非常有必要的。

XML 有两个先驱——SGML 和 HTML,这两个语言都是非常成功的置标语言,但是它们都在某些方面存在与生俱来的缺陷。XML 正是为了解决其不足而诞生的。

首先,从 SGML 说起。SGML 的全称是标准通用置标语言,从 20 世纪 80 年代初开始使用。正如 XML 一样,SGML 也可用于创建成千上万的置标语言,为语法置标提供了异常强大的工具,同时具有极好的扩展性,在分类和索引数据中非常有用。目前,SGML 多用于科技文献和政府办公文件中。

但是,SGML 非常复杂,其复杂程度相对于网络上的日常应用来说,简直不可思议。不仅如此,SGML 相关软件非常昂贵。目前,比较便宜的 SGML 软件之一是 Adobe FrameMaker,其标准版本价格为 850 美元,而 Adobe FrameMaker+SGML 是以 1995 美元售出的。还有最关键的一点,几个主要的浏览器厂商都明确拒绝支持 SGML,这无疑是 SGML 在互联网上传播遇到的最大障碍。

相反,HTML 免费、简单,而且获得了广泛的支持。HTML 最初于 1990 年由 CERN 设计,是一种非常简单的 SGML 语言,可以方便普通人使用。正如设计之初所构想的那样,HTML 目前在世界范围内得到了广泛的应用。

正因为如此,1996 年人们开始致力于描述一个新的置标语言,它既具有 SGML 的强大功能和可扩展性,同时又具有 HTML 的简单性。万维网联盟 W3C 决定专门成立一个 SGML 专家小组来从事此项工作,Sun 公司的 Jon Bosak 担任小组的指挥。

事实上,Bosak 和他领导的专家小组对 SGML 所做的贡献就像 Java 研究组对 C++ 做出的贡献一样。SGML 中所有非核心的、未被使用的和含义模糊的部分都被删除,剩下的就成为短小精悍的置标工具——XML。对 XML 的描述只有 26 页,而当初对 SGML 的描述却长达 500 页之多。对 XML 的描述尽管篇幅只是 SGML 的 1/20,但 SGML 中所有的精华都被保留了下来。

这以后,XML 不断发展演化,并且从 CML (Chemistry Markup Language) 和 MathML (Mathematical Markup Language) 中汲取了大量的经验。1997 年春天,可扩展链接语言 XLL (eXtensible Link Language) 草案已被拟定。到了 1997 年夏天,Microsoft (微软公司) 也开始了关于频道描述格式 CDF (Channel Definition Format) 的定义工作,这应该算是 XML 第一次被真正地应用。

最后,XML 于 1998 年修成正果。W3C 于 1998 年 2 月批准了 XML 的 1.0 版本,一

种崭新而大有前途的置标语言诞生了。

1.1.3 XML 概述

XML 不仅是置标语言,而且是可扩展的(Extensible)置标语言。它不像 HTML 那样,提供了一组事先已经定义好了的标记,而是提供了一个标准。利用这个标准,用户可以根据实际需要定义新的置标语言,并为这个置标语言规定其特有的一套标记。因此,准确地说,XML 是一种元置标语言,它允许程序开发人员根据所提供的规则,制订各种各样适合实际问题需要的置标语言。这也正是 XML 语言制订之初的目标所在。

以下是 XML 1.0 标准中描述的制订 XML 的目标。

- (1) XML 应该可以在互联网上直接使用。
- (2) XML 应该支持各种不同的应用方式。
- (3) XML 应该与 SGML 兼容。
- (4) 处理 XML 文档的应用程序应该容易编写。
- (5) XML 中的可选特性的数量应该减到最小,最好减至没有。
- (6) XML 文档应该具有良好的可读性,并且比较清晰。
- (7) 用 XML 设计新的置标语言应该方便快捷。
- (8) XML 设计的置标语言应该正式、简洁。
- (9) XML 文档应该容易编制。
- (10) XML 标记的简洁性并不重要。

让我们来看一个非常简单的例子。如果需要定义一个新的置标语言,以便这个语言定义一些标记来描述学生以及相关信息。这组标记很简单,如例 1-4(ch4-1.xml 为该例对应的文件名,下同)所示,它们的优点是代表了一定的语意。让我们回想一下这些信息在 HTML 中是如何用标记和表示的。与之相比,这种表示方法显然更加清晰、易读。

例 1-4 用 XML 标记置标的文档

```
<?xml version = "1.0" encoding = "UTF - 8"?>
<!DOCTYPE 学生列表 SYSTEM "xuesheng.dtd">
<学生列表>
  <学生>
    <学号> 200120101 </学号>
    <姓名>季慧奇</姓名>
    <性别>女</性别>
    <班级> 01 信管 1 </班级>
    <出生年月> 1985 - 1 - 2 </出生年月>
  </学生>
  <学生>
    <学号> 200120102 </学号>
    <姓名>凌怡</姓名>
    <性别>女</性别>
    <班级> 01 信管 1 </班级>
    <出生年月> 1983 - 6 - 20 </出生年月>
```

```
</学生>  
</学生列表>
```

这一段代码是一个非常简单的 XML 文档(Document)。它看上去它和 HTML 非常相像,但细心的人会发现这里的标记所代表的不再是显示格式,而是对学生信息数据的语义解释。事实上,用 XML 定义的置标语言可以根据标记描述的侧重点不同分为两大类:一类偏重于语意描述,正如例 1-4 所示;还有一类偏重于显示方式的描述,像现在已经推出的扩展文本置标语言 XHTML(eXtensible Hyper Text Markup Language)、可缩放矢量图形 SVG(Scalable Vector Graphic)语言和同步多媒体综合语言 SMIL(Synchronized Multimedia Integration Language)。值得一提的是,这里对于显示方式的描述不仅限于对文本的描述,还可以包括矢量图形、图像和声音。例如,一个形如<强调>的标记在描述文本时可能是要求将文本加粗,而在描述声音时则要求将音量加大。

不过,仅仅将数据置标还不够。为了让别人读懂这些数据,置标语言中的置标标准还需包括:置标的语法和每个置标的含义。

换句话说,如果想让计算机应用程序读懂并能处理这段数据,它还必须知道什么是有效的标记。如何处理一个有效的置标,具体地说,Netscape 浏览器如何知道怎样显示上面的这段 XML 文档? 标记<学号>是什么含义? 它究竟是不是一个合法的标记? 它又应该以什么方式表现? 因此,置标语言必须能够告诉应用程序它所采用的置标的语法,以便应用程序能够对其进行正确的处理。

在 XML 中,置标的语法是通过文档类型定义(Document Type Definition, DTD)或 Schema 来描述的。也就是说,通过 DTD/Schema 来描述什么是有效的标记,从而进一步定义置标语言的结构。第 3 章将详细讨论 DTD/Schema 的定义方法,这里先通过例 1-5(xuesheng.dtd)了解一下关于例 1-4 中用到的 DTD 文件“xuesheng.dtd”中的内容。

例 1-5 DTD 文档

```
<?xml version="1.0" encoding="UTF-8"?>  
<!ELEMENT 学生列表 (学生 *)>  
<!ELEMENT 学生 (学号, 姓名, 性别, 班级, 出生年月)>  
<!ELEMENT 学号 (#PCDATA)>  
<!ELEMENT 姓名 (#PCDATA)>  
<!ELEMENT 性别 (#PCDATA)>  
<!ELEMENT 班级 (#PCDATA)>  
<!ELEMENT 出生年月 (#PCDATA)>
```

除了定义置标的语法外,还需定义置标的含义,以便正确地显示。为了明确各个标记的含义,XML 使用与之相连的样式表(Style Sheet),向应用程序如浏览器,提供如何处理显示的指示说明。样式表的使用将在第 4 章具体描述。这里只需知道,样式表所做的规定可能是以下样式。

(1) 每当看到一个<学号>标记,用一个标记显示它。同样,</学号>转换为一个标记。

(2) 所有的<姓名>、<性别>、<班级>、<出生年月>标记被转换为标记加以显示。同样,</姓名>、</性别>、</班级>、</出生年月>转换为一个

标记。

例 1-5 中,我们使用 HTML 的标记功能来定义该 XML 数据文档的显示格式。但如果 XML 文档不是由浏览器而是由其他应用程序来进行处理,可能会采用其他相应的标记来输出。这样,应用处理程序就需要要综合 DTD、样式表以及 XML 文档数据三方面要素,根据这些数据 and 规定来显示它。

表面上看,以前只要一个 HTML 文档就能把数据和显示方式都包括进去,现在却需要 XML 文档、DTD 和样式表三个文件来共同完成。同时,我们也知道,浏览器只是用来处理一种特定的置标语言(如 HTML)的,而不是用来处理所有置标语言的,这说明我们可能还需要制作或购买一个额外的程序来处理 DTD、样式表和 XML 文档,这显得更加复杂了。的确,对初学者而言,在使用 XML 时确实会感到一些困难。不过,在了解 XML 的诸多优点之后,大家会知道这是非常值得的。

1.2 XML 的发展

1.2.1 HTML 的缺点和不足

HTML 是最早应用于网络信息传输的置标语言,也是近几年互联网上最普及的一种网页制作通用语言。它侧重于主页表现形式的描述,大大丰富了主页的视觉和听觉效果,为推动万维网(World Wide Web, WWW)的发展、推动信息和知识的网络交流发挥了不可取代的作用。但是,HTML 自身的特点使其蕴藏了许多危机。随着它不断发展,这些危机不但没有减弱,反而越来越突出,甚至已然成为 HTML 继续发展应用的障碍。

HTML 制订之初的本意在于根据信息的含义来为它们置标,而没有具体规定其应该如何在浏览器中显示。回忆一下,在 HTML 的早期版本中,<title>代表题目,<h1>代表第一层的大标题,<h2>代表第二层的大标题,、代表强调的文本,<address>代表作者的联系信息。至于这些题目、各层大标题究竟如何显示,应该由浏览器决定,因为 HTML 标准的制订者相信,比起网页的制作人员,浏览器更了解用户的偏好和使用的浏览环境。显然,网页的制作者事先并不知道哪些用户决定不显示图片、有哪些用户喜欢大一些的字体,只有浏览器才能保证为这些特殊用户提供良好的支持。

但是,浏览器的开发者同样也无从了解这些特殊用户的偏好。他们引入自己定义的一些标记和属性,用这些新的标记来专门描述显示格式,如标记、<center>、<bgcolor>等。浏览器厂商还开发了自己的网页制作软件,如 Netscape(网景公司)开发的 Netscape Composer、Microsoft 公司开发的 FrontPage 等。这些所见即所得的网页制作工具自动生成 HTML 文件,而这些 HTML 文件更是忽略了标记的语义信息,而几乎完全将它们作为格式表现的工具。例如,现在关于表格的标记<table>、<tr>、<td>等,不仅可以代表表格中不同行、列的信息,还可能专门用于网页布局。这样一来,HTML 越来越侧重于信息的表现方式,标记中原本就很微弱的对信息含义的描述也被削

弱了。HTML 最终演变为专门用于 Netscape Navigator 和 Microsoft IE 两大浏览器的页面显示语言。

可能读者会觉得,虽然某些特殊癖好用户的要求得不到满足,但毕竟对大多数人而言,浏览页面最基本的问题——显示问题还是解决了。而且,有了这些专门的标记,这个问题可能还解决得不错!其实不然。浏览器生产厂家在激烈的市场竞争中,为了显示自己的独特性,给 HTML 加入了一些特殊的标记,以便为自己的浏览器增加一些特殊的显示效果。日益增多的标记不但使 HTML 越来越庞大,浏览器的开发越来越复杂,还降低了不同浏览器之间的兼容性。例如,你的网页是针对 IE 5 浏览器 800×600 像素屏幕分辨率来制作的,那么在 640×480 像素的屏幕上观看的效果就会大打折扣。如果放到其他浏览器中,显示效果与最初的设计构想甚至会大相径庭。

不仅如此,尽管 HTML 的标记越来越多,其显示力却远远不够。如果希望非常精确地表现一些数据,可能需要一些现在 HTML 中尚不存在的标记。如果你是一个化学家,可能需要表现化学分子式中一些特别的符号,如果你是一个飞机设计师,可能希望能够表现飞机引擎的三维曲线造型。可对于这些,HTML 都爱莫能助。要想满足各行各业对显示的不同要求,显然需要大量的标记,这无疑使日益臃肿的 HTML 雪上加霜。

问题还不止这些,现在 HTML 内部结构的条理性越来越差。程序员写的 HTML 文件,甚至是那些专门的所见即所得工具自动生成的 HTML 文件,可能在语法上会错误百出,但浏览器照样能阅读它。HTML 中的标记可以不满足嵌套关系的层次完整性,如 `<h1><h2></h1></h2>` 可以不配对出现、只有 `<h1>` 而没有 `</h1>`,更不会要求在使用标记二级标题 `<h2></h2>` 的外面一定要保证有一级标题 `<h1></h1>`。乍一看,这仿佛对网页制作者而言是个福音,可浏览器的开发者却不得不把大量的精力耗费在文法错误的容错上,相应的,浏览器的程序也变得复杂,甚至牺牲浏览时的时间效率和空间效率。

另外,HTML 也无助于搜索引擎的开发。因为从 HTML 的标记本身,搜索引擎几乎得不到任何有用的信息。如果需要到网络上找出世界上所有关于 XML 书籍的价钱,搜索引擎将不得不分辨网络上哪些“XML”字段对应的是书名,同时要知道这些书名所对应的价钱。如果我们根据数据库进行搜寻,数据库中的各个字段都有明确的含义,问题比较好解决。但搜索引擎在网络上是根据 HTML 文件来进行搜索的,那些原本条理清晰、层次分明的数据库内容在 HTML 文件中已经被各种各样的标记所拆散,搜索引擎的任务就艰巨多了。

概括起来说,HTML 有以下几个固有弱点。

(1) HTML 是专门为描述网页的表现形式而设计的,疏于对信息语义及其内部结构的描述,不能适应日益增多的信息检索要求和存档要求。

(2) HTML 对表现形式的描述能力,实际上也还是非常不够的。它无法描述矢量图形、科技符号和一些其他的特殊显示效果。

(3) HTML 的标记集日益臃肿,而其松散的语法要求使文档结构混乱而缺乏条理,导致浏览器的设计越来越复杂,降低了浏览的时间效率与空间效率。

1.2.2 XML 的优点

1. XML 良好的可扩展性

在没有 XML 的时候,要想定义一个置标语言并推广利用它非常困难。一方面,如果制订了一个新的语言而期望它能生效,需要把这个标准提交给相关的组织,如:W3C (World Wide Web Consortium,万维网联盟),等待该组织接受并正式公布这个标准,经过几轮的评定、修改、再评定、再修改,等到这个置标语言终于成为一个正式推荐标准,可能已经过了几年的时间。另一方面,为了让这套标记得到广泛应用,制订者必须为它配备浏览工具。这样,就不得不去游说各个浏览器厂商接受并支持新制订的标记,或者索性自己开发一个新的浏览器去与现有的浏览器竞争。无论哪种办法,都需要耗费大量的时间和工作。现在借助 XML 的帮助,制订新的置标语言要简单易行得多了,这也正是 XML 的优势所在。

大家都知道,各个不同的行业可能会有一些独特的要求。例如,化学家需要化学公式中一些特殊符号,建筑家需要设计图纸中某些特殊的标记,音乐家需要音符,这些都需要单独的标记。但是,其他网页设计者一般不会用这些记号,也不需要这些标记。XML 的优点就在于它允许各个组织、个人建立适合他们自己需要的标记库,并且这个标记库可以迅速地投入使用。

不仅如此,随着当今世界越来越多元化,要想定义一套各行各业能够普遍应用的标记既困难,又没有必要。XML 允许各个不同的行业根据自己独特的需要,制订自己的一套标记,同时并不要求所有浏览器都能处理这成千上万个标记,同样也不要求置标语言的制订者制订出一个非常详尽、非常全面的语言,从而适合各个行业、各种领域的应用。比起那些追求大而全的置标语言的做法,这种具体问题具体分析的方法,实际上更有助于置标语言的发展。

实际上,现在许多行业、机构都利用 XML 定义自己的置标语言。比较早而且比较典型的是下面两个实例。

(1) 化学置标语言 CML (Chemistry Markup Language), 由 Peter Murray-Rust 制订。

(2) 数学置标语言 MathML (Mathematical Markup Language) 1.0 版本, W3C 1998/04/07 推荐标准。

2. XML 内容与形式的分离

XML 不仅允许自定义一套标记,而且这些标记不必仅限于对显示格式的描述。XML 允许根据各种不同的规则来制订标记,如根据商业规则、根据数据描述、甚至根据数据关系来制订标记。

仍以前面关于学生列表的文档为例,再对两种不同类型的文档进行一个详细的比较。

在例 1-3 中,文档是用 HTML 语言写的。尽管这也是一个存储、显示数据的可行方法,它的效率和能力却非常有限,至少存在以下三个严重的问题。

(1) 显示方式内嵌于数据之中。在这个例子中,数据是用列表的形式来表示的。如