

XML 高级网络应用

但正刚 李 顺 等 编著

清 华 大 学 出 版 社

(京)新登字 158 号

内 容 简 介

本书在读者对 XML 技术有所了解的基础上,进一步讲述 XML 高级应用知识。本书主要内容涉及 XML 模式、模式的几种方法、信息模型的建立及实现、XML 的 DOM 接口/SAX 接口/数据库技术、无线应用协议和 XML 在电子商务开发中的应用,以及 XML 与 ASP.NET 的联系及其应用。通过本书的学习,Web 站点的开发者能够把自己的站点提高一个层次,而编程者和软件系统程序员能够实现使 XML 与他们的系统相配合,并掌握使用 XML 来解决应用程序集成问题的方法。

本书适用于有一定 XML 基础的设计人员以及希望使用 XML 构建应用程序和系统的读者。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

XML 高级网络应用/但正刚等编著. —北京:清华大学出版社, 2002.8

ISBN 7-302-05583-1

I. X... II. 但... III. 可扩充语言, XML - 程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 041131 号

出 版 者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑:胡先福

印 刷 者:

发 行 者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:25 字数:607 千字

版 次:2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷

书 号:ISBN 7-302-05583-1/TP·3298

印 数:0001~5000

定 价:36.00 元

前 言

可扩展标记语言 (XML) 是在 Internet 上储存和传送信息方面最有发展前途的程序语言。虽然目前 HTML 语言仍然是建立网页最常使用的程序语言,但是它储存信息的能力却有很大的限制。比较而言,XML 就具有比较大的弹性,它允许程序员使用任何虚拟形态的信息,从简单的单笔数据直到复杂的数据库;一个 XML 文件通常与样式表或传统的 HTML 网页连接在一起,因此 XML 文件可以轻易地在网页浏览器中显示出来;XML 文件以高效率的结构和卷标来储存其所包含的信息,因此浏览器可以使用较灵活的方式来寻找、选取、排序、筛选、排列和处理信息;XML 提供了一种解决快速增加的网站数据量和信息复杂度问题的理想的解决方法;等等。这些都是其他网络语言所不具有的优点,也正是这些优点使得 XML 在网络开发中起到了越来越大的作用。

本书在读者对 XML 技术有所了解的基础上,进一步讲述 XML 高级应用知识。全书共 12 章,分为上、下两篇,上篇(1~6 章)为 XML 提高篇,下篇(7~12 章)为 XML 工业应用篇。第 1 章是 XML 模式的提高,主要包括 DTD 和 Schema 机制的介绍;第 2 章主要介绍 XML 语言中一个十分重要的概念——名域;第 3 章介绍了数据建模的概念及其在 XML 中的应用;第 4 章和第 5 章侧重 SAX 接口和 DOM 接口的高级使用,因为考虑到在 Windows 独霸天下的今天,各个软件都充斥着接口、模块、对象等诸如此类的东西,这种编程趋势使得我们必须对新语言的接口问题产生高度的重视;第 6 章介绍 XML 文档的存储、数据岛的概念以及 XML 与数据库的关系。从第 7 章开始,本书转入到对 XML 实际应用的介绍,主要是 XML 语言在几个典型领域的应用,如电子商务、无线应用协议、SOAP 协议、多媒体 SMIL 等。对设计者来说,一种语言最终的用处就是在实际的应用上,当然,XML 的实际应用远远超出本书中所介绍的这几个方面。本书最后一章介绍了 ASP.NET 的基本概念、XML 在 ASP.NET 平台中的重要作用以及用 .NET 实现 XML 的 DOM 接口和 SAX 接口的相关知识。

也许读者觉得本书介绍的内容实在太多了,实际上,本书仅仅着重于将 XML 作为一种开放技术的应用工具来介绍,提供了 XML 模式、数据建模以及数据交换等技术的研究。本书面向有一定 XML 基础的设计人员,因此无论在内容上还是形式上都比较专业,大多涉及 XML 实际应用的内容,例如 XML 在 DOM 接口、SAX 接口和数据库方面的应用以及 XML 的一些实际应用扩展。尤其是 XML 的扩展,本书介绍了最新的 WML 语言、XML 在 .NET 平台中的重要角色以及 XML 在电子商务中的应用包括 BizTalk 方面的研究。相信读完本书后,读者对于 XML 的各种专业技术会有比较全面的了解。尽管如此,本书在写作上仍然注意语言的浅显易懂,并且尽量避免过多地介绍 XML 基础概念和基础语法,而是直接从 XML 的高级应用方面入手,为有一定 XML 基础的读者节约时间。

除封面署名外,参与本书编写、整理、校对等工作的还有王少华、方美玲、王冰、钱

其胜、韩卫东、吴嘉慧、林清松、徐岭、袁东、董建平、黄海英、付清源、曹立清、王昕、古彦平、汪静、王少波、李济民等人。由于水平和经验所限，书中不足和纰漏之处在所难免，恳请广大读者批评指正。

作 者

2002 年 6 月

目 录

上篇 XML 提高篇

第 1 章 XML 模式提高	3
1.1 DTD 和 Schema 回顾.....	4
1.1.1 DTD 机制.....	4
1.1.2 XML Schema.....	5
1.1.3 DTD 和 Schema 的比较.....	7
1.2 DTD 语法	8
1.2.1 DTD 的一般结构	8
1.2.2 DTD 元素定义.....	12
1.2.3 DTD 与实体.....	13
1.2.4 DTD 元素属性.....	19
1.2.5 内部 DTD 和外部 DTD	25
1.3 DTD 进阶	28
1.3.1 DTD 的约束和有效性检查	28
1.3.2 DTD 的编写实例	29
1.3.3 DTD 的说明	30
1.3.4 DTD 的命名空间问题	32
1.4 XML Schema 语法	33
1.4.1 Schema 的一般结构.....	33
1.4.2 Schema 的元素定义.....	33
1.4.3 Schema 的属性声明.....	38
1.4.4 XML Schema 中的实体声明以及注释	40
1.5 DTD 和 XML Schema 实例.....	41
1.6 小结	43
第 2 章 名域.....	44
2.1 名域	45
2.1.1 名域的产生	45
2.1.2 名域概述	49
2.1.3 表示名域	49
2.2 名域的定义和声明.....	52
2.2.1 声明一个名域	52

2.2.2	限定名	53
2.2.3	范围	53
2.3	名域的范畴和应用	54
2.3.1	名域的范畴	54
2.3.2	名域应用	57
2.4	Schema 中的名域	62
2.5	小结	70
第 3 章	数据建模与 XML 应用	71
3.1	数据建模	72
3.1.1	UML 方法	72
3.1.2	动态模型和静态模型	75
3.2	信息建模	78
3.2.1	静态建模	78
3.2.2	动态建模	81
3.3	设计 XML 文档	82
3.3.1	XML 文档的概念	83
3.3.2	将信息模型映射到 XML	87
3.4	模式语言和表示法	93
3.4.1	模式的作用	93
3.4.2	将 DTD 作为模式	94
3.4.3	根据信息模型创建 DTD	96
3.4.4	XML 模式	99
3.5	小结	100
第 4 章	DOM 进阶	101
4.1	XML 文档的加载和遍历	102
4.1.1	DOM 回顾	102
4.1.2	使用 DOM 创建节点树	102
4.1.3	DOM 创建和加载文档应用	104
4.1.4	DOM 遍历文档实例	107
4.2	使用 DOM 对文档进行操作	123
4.2.1	添加元素	123
4.2.2	删除元素	125
4.2.3	改变元素	126
4.2.4	处理错误	126
4.3	DOM 接口介绍	128
4.3.1	Document 接口	128

4.3.2	Node 接口	129
4.3.3	NodeList 接口	132
4.3.4	NamedNodeMap 接口	132
4.4	DOM 应用实例	133
4.4.1	简单的客户端实例	134
4.4.2	复杂的 DOM 编程实例	141
4.4.3	基于 DOM 的投票系统	148
4.5	DOM 和 XML 的未来	153
4.5.1	W3C 的工作	153
4.5.2	DOM 的应用	153
4.5.3	数据库、模式和 DOM	154
4.6	小结	154
第 5 章	SAX 进阶	155
5.1	SAX 接口	156
5.1.1	SAX 概述	156
5.1.2	DOM 和 SAX	159
5.2	SAX 的接口和类介绍	160
5.2.1	SAX 的接口	160
5.2.2	SAX 的类	164
5.3	SAX 应用实例	167
5.3.1	SAX 处理元素属性	167
5.3.2	SAX 的解析器处理 XML 文档	170
5.3.3	基于事件解析的 SAX 模型分析	171
5.4	小结	175
第 6 章	XML 与数据	176
6.1	存储 XML 文档	177
6.1.1	采用文件系统存储的局限性	177
6.1.2	数据库存储	178
6.1.3	关系数据库介绍	182
6.1.4	Microsoft SQL Server 数据库对 XML 的支持	185
6.2	数据交换	187
6.2.1	数据传送	187
6.2.2	用 DOM 方法创建 XML 文档	193
6.3	数据岛概念	195
6.3.1	数据岛对象	196
6.3.2	节点的操作	198

6.3.3	其他 XML 文档的显示方法.....	201
6.3.4	XML 数据岛和 DSO 的使用.....	204
6.3.5	DSO 结合数据绑定-Client 端的分页显示	206
6.3.6	XML 数据岛和 XSL 应用-Client 端的查询/排序	206
6.3.7	XML 数据岛中数据的编辑和发送.....	208
6.3.8	动态改变 XSL 以实现按要求查询/排序	209
6.3.9	数据岛实例分析	210
6.4	XML 模式与数据库.....	213
6.4.1	数据库到 XML 模式	213
6.4.2	应用规则	215
6.5	XSL 样式表概述.....	219
6.6	网上音乐预定系统.....	222
6.6.1	页面显示音乐	223
6.6.2	处理客户信息	227
6.6.3	保存用户信息	231
6.6.4	后续优化工作	235
6.7	小结	236

下篇 XML 工业应用篇

第 7 章	同步多媒体集成语言 SMIL.....	239
7.1	SMIL 简介	240
7.1.1	SMIL 概述.....	240
7.1.2	SMIL 结构.....	241
7.1.3	SMIL 示例.....	241
7.1.4	SMIL 的支持工具.....	243
7.1.5	SMIL 前景.....	243
7.2	SMIL 的元素	243
7.2.1	文件头元素	243
7.2.2	基本布局元素	245
7.2.3	文件体元素	248
7.2.4	超链接元素	256
7.3	SMIL 实例	260
7.3.1	“幻灯片”——使用行内时序	260
7.3.2	“增长的列表”——使用 CSS 时序	260
7.3.3	“正方形”——使用时间表	261
7.4	小结	262

第 8 章 WAP 基础及 Server 配置	263
8.1 WAP.....	264
8.1.1 WAP 概述	264
8.1.2 WAP 网络实例	265
8.1.3 WAP 上网过程	266
8.2 WAP Server 的建立	266
8.2.1 预备知识	266
8.2.2 从 Microsoft IIS5.0 到 WAP Server.....	267
8.2.3 从 PWS 到 WAP Server.....	270
8.2.4 从 Apache 到 WAP Server.....	270
8.3 WML 的软件使用	271
8.3.1 Infinite WAP Server	271
8.3.2 Nokia WAP Toolkit.....	273
8.4 小结	276
 第 9 章 WML 语言及 WAP 开发	277
9.1 WML.....	278
9.1.1 WML 基本概念.....	278
9.1.2 WML 与 XML.....	278
9.1.3 WML 文件结构.....	280
9.1.4 WML 的显示.....	281
9.2 WML 语法规则.....	281
9.2.1 字符集	282
9.2.2 大小写敏感	282
9.2.3 WML 文档的基本框架.....	282
9.2.4 WML 文档中的保留符号.....	283
9.2.5 标签的属性	283
9.2.6 引用变量	284
9.3 WML 中的标签.....	284
9.3.1 页面与卡片	285
9.3.2 用户输入	288
9.3.3 文本格式	290
9.3.4 图片	292
9.3.5 锚	293
9.3.6 事件	294
9.3.7 任务	295
9.3.8 定时	296
9.3.9 变量	296

9.4	WML 的实例分析	297
9.4.1	显示文本	297
9.4.2	显示图片	298
9.4.3	显示表单	298
9.4.4	处理事件	301
9.4.5	CGI 编程	304
9.4.6	单页的含有图片的 WML	306
9.4.7	多页的 WML 的定位	307
9.5	小结	309
第 10 章	SOAP 协议规范	310
10.1	SOAP 简介	311
10.1.1	设计目标	311
10.1.2	SOAP 消息举例	312
10.1.3	SOAP 消息交换模型	313
10.1.4	与 XML 的关系	313
10.2	SOAP 封装	314
10.2.1	封装	314
10.2.2	SOAP 头	315
10.2.3	SOAP 体	316
10.2.4	SOAP 错误	317
10.3	SOAP 编码	318
10.3.1	XML 中的编码类型规则	319
10.3.2	简单类型	321
10.3.3	多态 accessor	323
10.3.4	复合类型	323
10.3.5	默认值	332
10.3.6	SOAP root 属性	332
10.4	在 HTTP 中使用 SOAP	333
10.4.1	SOAP HTTP 请求	333
10.4.2	SOAP HTTP 应答	333
10.4.3	HTTP 扩展框架	334
10.4.4	SOAP HTTP 举例	334
10.5	在 RPC 中使用 SOAP	335
10.5.1	SOAP 与防火墙	335
10.5.2	RPC 和 SOAP 体	337
10.5.3	RPC 和 SOAP 头	337
10.6	小结	337

第 11 章 XML 与电子商务	339
11.1 电子商务基础.....	340
11.1.1 电子商务	340
11.1.2 电子商务的三种业务模式	341
11.1.3 电子商务是现代工业社会高度发展的结果.....	342
11.1.4 在电子商务中使用 XML 的理由.....	342
11.2 XML BizTalk 框架.....	343
11.2.1 BizTalk 概述	344
11.2.2 符合 BizTalk 框架的文件结构.....	344
11.2.3 BizTalk 框架的优点	345
11.2.4 BizTalk 大纲标准库	346
11.2.5 BizTalk 框架结构原则.....	346
11.2.6 深入讨论 BizTalk 框架.....	347
11.3 BizTalk 的工作、部署及开发工具	351
11.3.1 BizTalk 如何工作	351
11.3.2 BizTalk 架构	352
11.3.3 部署 BizTalk	353
11.3.4 构建于 SQL Server 2000 之上	354
11.3.5 BizTalk Jumpstart 工具包简介	354
11.4 BizTalk 的使用.....	355
11.5 小结	359
 第 12 章 XML 与 ASP.NET.....	360
12.1 .NET 及 ASP.NET 基础.....	361
12.1.1 .NET 战略的目标.....	361
12.1.2 .NET Framework 的层次结构	361
12.1.3 .NET Framework 的组成部分	362
12.1.4 什么是 ASP.NET?	365
12.1.5 ASP.NET 的新特性	365
12.1.6 ASP.NET 与 XML 的联系	366
12.2 .NET 中的 DOM 对象模型.....	367
12.2.1 .NET 的 DOM 实现	367
12.2.2 .NET DOM 对象的主要类	368
12.3 .NET Framework 中的 SAX	377
12.3.1 .NET Framework 中的 SAX 对象	377

12.3.2	使用 XmlReader 访问数据	377
12.3.3	结合 ASP.NET 运用 SAX	380
12.4	Web 控件使用 XML Web Service.....	382
12.5	在 ASP.NET 中利用 XML 绑定数据.....	386
12.6	小结	387

上篇XML 提高篇

第1章

XML 模式提高

知识点

- DTD 和 Schema 以及两者比较
- DTD 语法
- DTD 进阶
- XML Schema 语法
- 编写 DTD 和 XML Schema 文件

导读：

在实际应用中,只保证一个 XML 文档是一个结构良好的 XML 文档是不够的,必须还要保证 XML 文档的有效性。DTD 和 XML Schema 就是为了保证 XML 文档的有效性而制定的。本章将以对比的形式来讲述 DTD 和 XML Schema 的联系和区别,然后详尽介绍它们的语法和结构,在本章的最后部分将介绍如何在实际应用中编写 DTD 和 XML Schema。

1.1 DTD 和 Schema 回顾

1.1.1 DTD 机制

XML 提供一种被称为文件类型声明的独立于应用程序之外的数据交换机制,用于定义对逻辑结构的约束,和支持预定义存储单元的使用。如果一个 XML 文件有相应的文件类型声明并且它遵循其中的约束,则称它是有效的。XML 文件类型声明包含或指向标记声明,标记声明提供某一类文件的文法。这种文法被称为文件类型定义(Document Type Definition, DTD)。

XML 是可扩展的,但它必须受到一定的语法限制。因为在不同的领域中,XML 数据在含义、数据类型以及数据关联上必须有不同的限制。让我们来看一个火车时刻表的记录片断,这个记录描述了火车的车次、车种、起点站、终点站、发车时间和到站时间等信息:

```
<火车时刻表>
<火车>
  <车次>5017</车次>
  <车种>普快</车种>
  <起点站>杭州</起点站>
  <终点站>温州</终点站>
  <终点站>福州</终点站>
  <发车时间>8:29</发车时间>
  <到站时间>16:30</到站时间>
</火车>
</火车时刻表>
```

可以看出,以上记录并不合法,因为它包含了两条终点站的记录,而我们知道一列火车是不会有二个终点站的。

再看一个记录片断:

```
<火车时刻表>
<火车>
  <车次>5017</车次>
  <车种>好快啊</车种>
  <起点站>杭州</起点站>
  <终点站>温州</终点站>
  <发车时间>8:29</发车时间>
```

```
<到站时间>16:30</到站时间>
</火车>
</火车时刻表>
```

同样,我们现在知道车种只有 3 个选择:普快、快速和特快,而在这个记录中车种为“好快啊”,所以以上记录也是不合法的。

通过制定 DTD 文档,XML 的数据格式就会得到约束,使数据交换更加有章可循。在发送数据之前可以验证其合法性,收到的数据在进一步处理之前也可以保证它符合 DTD 结构内容的规定。

至于如何编写 DTD 和 XML Schema 文档,我们将在后面详细介绍。

1.1.2 XML Schema

目前 XML 文档的内容描述和约束是由 DTD 完成的。DTD 中明显还深深刻着 SGML 的烙印。与 XML 的巨大革命和成功相比,DTD 尚没有实质性的飞跃。关键的一点是,DTD 有自己的特殊语法,它虽然可以用以规定限制 XML,但其本身并不是 XML。我们必须认真学习 DTD 才能熟练地创建和编写 XML 文档;而且我们必须有两套不兼容的解析器:一套用来分析 XML,如判断其结构是否良好;另一套用来分析 DTD,再用分析的结果去检查 XML 文档,判断 XML 是否有效。DTD 只提供了非常有限的几种数据类型。DTD 中规定的文档内容都是字符数据,连整型、浮点型、数据型、布尔型这样简单的数据类型都不提供,更毋谈更复杂的数据类型了。DTD 不支持名域(Namespace)机制,虽然 DTD 可以为不同的 XML 所引用,但一个 XML 文档只能有一个相对应的 DTD。不同的 DTD 并不能同时对同一个 XML 文档进行规定,哪怕只是对其中的部分元素。也就是说传统的 DTD 机制使得 XML 的继承性和重用性受到限制。DTD 有扩展的机制,但这个机制太复杂而且很脆弱。DTD 扩展机制的最大毛病在于不能清楚地表达相互之间也的关系。两个有着完全相同内容的元素怎么做也不能互相联系。同样地,被定义为参数实体的不同属性之间也不能建立任何联系。另外 DTD 中的内容模型是不开放的,它不能随意扩充内容,否则将无法被解析。以上的这些缺点事实上已经在束缚 XML 的发展,重新制定符合 XML 的大纲机制已经是大势所趋,这样的新机制就是 XML Schema。

Schema 标准的发展经历了较长的过程。早在 XML1.0 标准尚未确定之前,W3C 就已经对 XML Schema 开始进行研究。1998 年 1 月 5 日,W3C 接受 XML Data 为 Note。1999 年 2 月 15 日,又提出了“XML Schema 需求报告”。不久后的 1999 年 5 月 6 日,“XMLSchema 工作草案”被接受为工作草案。同年的 11 月 5 日,该草案被修改,但还是工作草案。该草案分为两个部分:结构(Structure)和数据类型(Datatypes)。

虽然没有确定 XML Schema 为最终标准是出于慎重考虑。不过经过多次的讨论和完善,目前的工作草案已经基本成熟。

Microsoft 十分积极地参与了草案的修改,在 XML-Data 的基础之上提出了一个 XML Schema 的子集,没有静等 XML Schema 标准确定就开发了支持该子集的解析器和浏览器 Internet Explorer 5.0。Microsoft 的 XML-Data 出乎意料地受到了业界的认可。人们觉得在正

式标准出来之前，XML-Data 是不错的“准标准”或过渡。后来 Microsoft 对 XML-Data 进行了精减，产生了 XML-Data Reduces Version，即现在常用的 XDR。

最新的 XML Schema 于 2000 年的 10 月 17 日被作为候选推荐发布，它结合 XDR 和 DDML(文档定义标记语言)的思想对 XML Schema 做出了重新的设计。

XML Schema 结构规定了 XML Schema 的定义语言。它提供了规定 XML 文档结构和内容约束的机制，在 XML 1.0 DTD 的基础上更迈进了一步。这个规范和 XML Schema 的第二部分“数据类型”共同对 XML 文档进行定义。一个 XML Schema 包含类型定义和元素声明等部件，用来衡量格式良好的元素信息的有效性。进一步还可以规定这些元素项目及其子节点，比如属性和元素的默认值、元素类型和属性表等。

1. XML Schema 的新特点

基于 DTD 到 Schema 的变化，XML 大纲提供了一系列的新改进，这些改进大大弥补了 DTD 的不足：

- ❑ 丰富的数据类型。XML 大纲支持的数据类型包括：数字型、布尔型、整型、日期时间、URI、十进制数等等。而且它还支持由这些简单的类型生成更复杂的类型。
- ❑ 可以由用户自定义数据类型，称为 Archetype(原型)。比方说，可以先定义 PostalAddress 这个数据类型，然后定义 ShippingAddress 和 BillingAddress 为这个类型的两个元素。
- ❑ 属性分组。属性的应用范围是多种多样的，有的是所有元素都有的，有的是专门为图形元素设定的。
- ❑ 原型可以更新。这是最重要的特色。DTD 定义的内容模式是封闭的，而 XML 大纲定义的内容模式是开放的，是可以更新的。
- ❑ Namespace(名域)的支持。自从 XML 引入了名域，有效性的检查就变得复杂了。在 XML 完全开发之前，这个问题很难有完满的解决。

2. XML Schema 的整体结构

XML Schema 结构规定了 XML Schema 的定义语言。它提供了规定 XML 文档结构和内容约束的机制，在 XML 1.0 DTD 的基础上更迈进了一步。这个机制和 XML Schema 的第二部分“数据类型”共同对 XML 文档进行定义。

一个 XML Schema 由类型定义和元素声明等部件来衡量格式良好的元素信息的有效性。进一步还可以规定这些元素项目及其子节点，比如属性和元素的默认值、元素类型和属性表等。从抽象数据模式上看，XML Schema 的通用组成项是 Schema 组件(Schema component)。一个 XML Schema 是许多 Schema 组件的集合。共有 12 类组件，它们可以分为 3 组。最基本的一组组件有：

- ❑ 简单类型定义
- ❑ 复杂类型定义
- ❑ 属性声明

□ 元素声明

次要一些的组件有：

□ 属性组定义

□ 标识限定定义

□ 模型组定义

□ 记法声明

以上两组的组件都有名称，可以单独地出现在 Schema 的内容中，而且可以被单独访问。最后，helper 组件提供了其他组件的细小部分。它们不能在 Schema 中单独出现而且不能独立访问，这些组件包括：

□ 注释模型组

□ 粒子

□ 通配符

1.1.3 DTD 和 Schema 的比较

1. DTD 的缺点

DTD 的缺点在前面已经介绍过了，这里再总结如下：

- DTD 只支持自己的特殊语法。它虽然可以用以规定 XML 的限制，但其本身并不是 XML。我们必须认真学习 DTD 才能熟练地创建和编写 XML 文档；而且我们必须有两套不兼容的解析器：一套用来分析 XML，如判断其结构是否良好；另一套用来分析 DTD，再用分析的结果去检查 XML 文档，判断 XML 是否有效。
- DTD 只提供了非常有限的几种数据类型。DTD 中规定的文档内容都是字符数据，连整型、浮点型、数据型、布尔型这样简单的数据类型都不提供，更毋谈更复杂的数据类型了。
- DTD 不支持名域(Namespace)机制，虽然 DTD 可以为不同的 XML 所引用，但一个 XML 文档只能有一个相对应的 DTD。不同的 DTD 并不能同时对同一个 XML 文档进行规定，哪怕只是对其中的部分元素。也就是说传统的 DTD 机制使得 XML 的继承性和重用性受到限制。
- DTD 的扩展机制太复杂而且很脆弱。DTD 扩展机制的最大毛病在于不能清楚地表达相互之间的关系。两个有着完全相同内容的元素怎么做也不能互相联系。同样地，被定义为参数实体的不同属性之间也不能建立任何联系。
- 另外 DTD 中的内容模型是不开放的，它不能随意扩充内容，否则将无法被解析。

纵然 DTD 有着诸多的缺点，但是由于 DTD 发展比较成熟而且有了广泛的工具支持，所以在短期内 DTD 还有一定优势及存在的理由。

2. Schema 的优点

XML Schema 是一种描述信息结构的模型。它是用来定义 XML 文件的文本结构、数