



南京航空航天大学
研究生系列精品教材

XML知识管理： 概念与应用

黄志球 沈国华 康达周 编著



科学出版社

南京航空航天大学研究生系列精品教材

XML 知识管理：概念与应用

黄志球 沈国华 康达周 编著

科学出版社

北 京

内 容 简 介

本书全面介绍可扩展标记语言(XML)及其相关知识,具体分为三个部分:第1部分介绍XML基础,包括XML的概述、语法、有效性、解析、应用等内容;第2部分介绍Web服务,包括XML与Web的服务描述、服务发现与访问、服务组合、服务安全等内容;第3部分介绍语义Web及知识管理,包括资源描述框架RDF、Web本体语言OWL、面向服务的Web本体语言OWL-S等内容。

本书概念严谨、结构清晰、深入浅出、通俗易懂。通过大量的实例帮助读者掌握必须的基本语法和使用方法。

本书可作为高等学校计算机专业以及相关专业的本科生和研究生的XML、Web服务、语义Web课程教材,也可以作为广大计算机爱好者的自学手册。

图书在版编目(CIP)数据

XML 知识管理:概念与应用 / 黄志球, 沈国华, 康达周编著. —北京:科学出版社, 2015.9

ISBN 978-7-03-045604-5

I. ①X… II. ①黄… ②沈… ③康… III. ①可扩展语言—应用—知识管理—研究 IV. ①TP312 ②G302

中国版本图书馆 CIP 数据核字(2015)第 212038 号

责任编辑:潘斯斯 张丽花 / 责任校对:郭瑞芝

责任印制:徐晓晨 / 封面设计:迷底书装

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

北京京华虎彩印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

2015年10月第 一 版 开本:787×1092 1/16

2015年10月第一次印刷 印张:12 3/4

字数:288 000

定价:45.00 元

(如有印装质量问题,我社负责调换)

前 言

可扩展标记语言(XML)是由 W3C 组织于 1998 年 2 月发布的标准,是在简化了标准通用标记语言 SGML 后形成的一种元标记语言。自从 XML 发布以来一直受到学术界和工业界的广泛关注,并且成为表示结构化数据的标准。

XML 可以用来标记数据、定义数据类型,是一种允许用户对自己的标记语言进行定义的元语言。XML 提供统一的方法来描述和交换独立于应用程序或供应商的结构化数据,是当今处理分布式结构信息并实现数据交换的有效工具。因此,学习 XML 是高等学校计算机专业以及相关专业的学生进行数据类型定义、数据建模的基础。学好 XML 可触类旁通到基于 XML 的其他若干语言,如 Web 服务描述语言 WSDL、服务访问协议 SOAP、Web 本体语言 OWL 等。本书是在作者总结了过去的教学、研究和开发实践的基础上编写而成的,适合作为高等学校计算机专业以及相关专业的本科生和研究生的 XML、Web 服务、语义 Web 课程教材,也可以作为广大计算机爱好者的自学手册。

本书具有如下特点:

- (1) 内容通俗易懂,便于计算机专业以及相关专业的非计算机专业的学生学习。
- (2) 以 XML 为基础,以 Web 为发展脉络,讲述了 XML、Web 服务和语义 Web。
- (3) 给出了很多实例,帮助读者掌握基本语法和使用方法。

本书第 1~6 章由黄志球执笔,第 7~10 章由沈国华执笔,第 11~13 章由康达周执笔。全书由黄志球和沈国华负责统稿。另外,仵志鹏、陆陈、陈光颖、何一凡、张学明、刘银陵、王飞、王思琪、丁泽文、马薇薇、姜家鑫、潘诚、司徒佳、王梓、江东宇、孟云飞等参与了本书编写、图表制作。

限于作者的水平,本书若存在疏漏和不妥之处,恳请专家和广大读者批评指正。本书作者的联系方式: zqhuang@nuaa.edu.cn(黄志球), ghshen@nuaa.edu.cn(沈国华), dzkang@nuaa.edu.cn(康达周)。

作 者

2015 年 5 月

于南京航空航天大学

目 录

前言

第 1 部分 XML 基础

第 1 章 XML 概述	2
1.1 什么是 XML	2
1.2 XML 发展历程及优势	3
1.2.1 XML 的出现及发展	3
1.2.2 XML 与 HTML 的关系	3
1.2.3 XML 的发展前景	6
1.3 XML 的优势	6
1.3.1 良好的可扩展性	6
1.3.2 内容与形式分离	7
1.3.3 遵循严格的语法要求	7
1.3.4 便于不同系统之间的信息传输	8
1.3.5 具有良好的保值性	8
1.4 本章小结	9
第 2 章 XML 语法	10
2.1 XML 文档的序言	10
2.1.1 XML 声明	10
2.1.2 DTD	12
2.1.3 处理指令	12
2.1.4 注释	13
2.2 XML 文档的元素段	13
2.2.1 元素	13
2.2.2 标记	15
2.2.3 属性	17
2.2.4 引用	18
2.2.5 CDATA 段	19
2.3 创建格式良好的 XML 文档	19
2.4 本章小结	21
第 3 章 XML 的有效性	22
3.1 文档类型定义	22
3.1.1 DTD 简介	22
3.1.2 DTD 的语法	23
3.1.3 DTD 的调用	29

3.2	XML Schema	30
3.2.1	XML Schema 的语法	31
3.2.2	XML Schema 间的调用	36
3.3	DTD 与 XML Schema 的比较	37
3.4	本章小结	38
第 4 章	XML 的格式化与转换	39
4.1	CSS	39
4.1.1	CSS 简介	39
4.1.2	CSS 的相关属性	40
4.1.3	CSS 的使用方法	42
4.2	XSL	43
4.2.1	XSL 简介	43
4.2.2	XSLT	44
4.2.3	XPath	46
4.2.4	XSL-FO	48
4.3	本章小结	48
第 5 章	XML 解析	49
5.1	DOM	49
5.1.1	DOM 简介	49
5.1.2	DOM 节点操作	55
5.2	SAX	59
5.2.1	SAX 简介	59
5.2.2	SAX 的常用接口	59
5.2.3	SAX 解析 XML	60
5.3	DOM4J	64
5.3.1	DOM4J 简介	64
5.3.2	DOM4J 使用	64
5.4	本章小结	69
第 6 章	XML 的应用	70
6.1	XML 开发工具	70
6.1.1	XMLspy	70
6.1.2	<oXygen/> XML Editor	71
6.1.3	XMLPad	73
6.1.4	其他开发工具	74
6.1.5	工具的对比	74
6.2	RSS 和 Atom	75
6.2.1	RSS	75
6.2.2	Atom	76

6.3	RSS 和 Atom 的对比	77
6.4	本章小结	77

第 2 部分 Web 服务

第 7 章	XML 与 Web 服务描述	80
7.1	Web 服务概述	80
7.1.1	Web 服务体系结构	80
7.1.2	Web 服务技术架构	81
7.1.3	Web 服务案例	83
7.2	WSDL: Web 服务描述语言	84
7.2.1	WSDL 的主要元素	84
7.2.2	WSDL 结构	88
7.2.3	WSDL 应用实例	88
7.3	本章小结	91
第 8 章	XML 与 Web 服务发现和访问	92
8.1	SOAP	92
8.1.1	SOAP 简介	92
8.1.2	SOAP 消息结构	93
8.1.3	SOAP 通信模型	95
8.2	UDDI	96
8.2.1	UDDI 简介	96
8.2.2	UDDI 数据结构	98
8.2.3	WSDL 到 UDDI 的映射	101
8.3	本章小结	103
第 9 章	XML 与 Web 服务组合	104
9.1	Web 服务组合	104
9.1.1	业务流程管理	104
9.1.2	工作流	105
9.1.3	Web 服务组合流模型	106
9.1.4	Web 服务组合的具体实现	108
9.2	服务编配与编排	111
9.2.1	服务编配与编排比较	111
9.2.2	业务流程执行语言	113
9.2.3	Web 服务编排	116
9.3	实例分析	118
9.4	本章小结	129
第 10 章	XML 与 Web 服务安全	130
10.1	XML 安全性标准	130

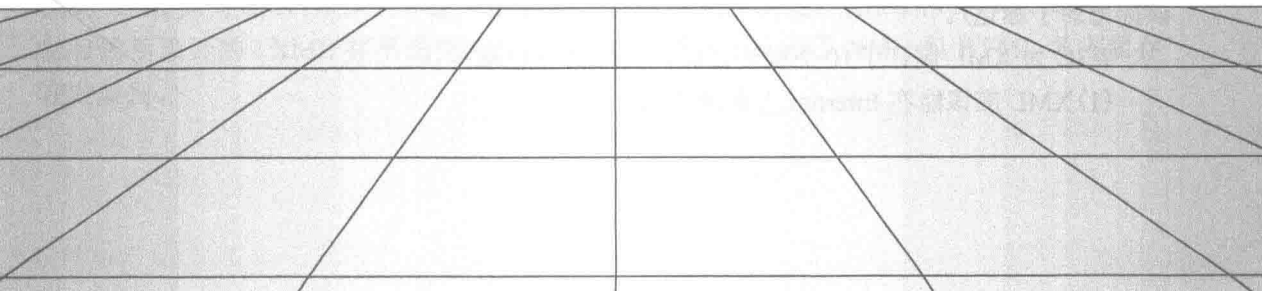
10.1.1	XML 签名	130
10.1.2	XML Encryption	133
10.1.3	XML 加密管理规范	134
10.1.4	安全断言标记语言	136
10.1.5	XML 访问控制标记语言	138
10.2	Web 服务的安全性	140
10.2.1	Web 服务面临的安全性问题	140
10.2.2	Web 服务安全性模型	141
10.2.3	WS-Security	142
10.2.4	Web 服务平台安全性体系结构	144
10.2.5	Web 服务安全性应用	146
10.3	本章小结	147

第 3 部分 语义 Web 及知识管理

第 11 章	资源描述框架 RDF	150
11.1	RDF 简介	150
11.1.1	RDF 的含义	150
11.1.2	RDF 的设计目的	151
11.1.3	RDF 规则	151
11.1.4	RDF 与 XML	152
11.2	RDF 模型	153
11.2.1	RDF 资源和词汇集	153
11.2.2	RDF 图	153
11.2.3	结构化特性与空节点	154
11.3	RDF/XML 语法	155
11.3.1	基本语法	156
11.3.2	简写语法	156
11.3.3	容器	157
11.3.4	集合	158
11.3.5	声明具体化	159
11.4	RDF Schema	160
11.4.1	RDFS 简介	160
11.4.2	类	160
11.4.3	RDF 属性	161
11.4.4	属性约束	162
11.5	本章小结	163
第 12 章	Web 本体语言	164
12.1	本体	164

12.2	OWL 简介	164
12.2.1	RDFS Schema 表达能力的局限性	165
12.2.2	OWL 子语言	165
12.2.3	OWL 本体结构	166
12.2.4	命名空间	166
12.2.5	本体头部	167
12.3	基本元素	167
12.3.1	简单的类和个体	167
12.3.2	简单属性	169
12.3.3	属性特性	170
12.3.4	属性限制	172
12.4	类、属性及个体间的关系	174
12.4.1	类和属性之间的等价关系	174
12.4.2	个体间的关系	174
12.5	复杂类	175
12.5.1	交运算	175
12.5.2	并运算	176
12.5.3	补运算	177
12.5.4	枚举类	177
12.5.5	不相交类	178
12.6	OWL2 标准	179
12.6.1	语法糖	179
12.6.2	属性的新结构	180
12.6.3	扩展的数据类型能力	181
12.7	PROFILE	182
12.7.1	OWL2 EL	182
12.7.2	OWL2 QL	182
12.7.3	OWL2 RL	183
12.8	本章小结	183
第 13 章	服务本体描述语言	184
13.1	OWL-S 简介	184
13.2	服务配置文件	185
13.3	服务模型	186
13.3.1	原子和简单过程	188
13.3.2	复合过程	190
13.3.3	数据流和参数绑定	191
13.4	服务基础	192
13.5	本章小结	193
参考文献		194

第 1 部分 XML 基础



第 1 章 XML 概述

1.1 什么是 XML

XML (eXtensible Markup Language) 是由万维网联盟 (World Wide Web Consortium, W3C) 定义的可扩展标记语言。XML 与 HTML 一样, 都是标准通用标记语言 (Standard Generalized Markup Language, SGML)。XML 允许用户按照 XML 规则自定义标记, 具有可扩展性。XML 文件是由标记及其所标记的内容构成的文本文件, 与 HTML 文件不同的是, 这些标记可自定义, 其目的是使得 XML 文件能够更好地体现数据的结构和含义。W3C 推出 XML 的主要目的是使得 Internet 上的数据相互交流更方便, 让文件内容浅显易懂。以下是一个简单的 XML 文件。

例 1.1

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Book List</title>
</head>
<body>
  <books xmlns="http://www.yinheedu.com/books/xml">
    <book>
      <title>企业级应用数据传技术</title>
      <author>漆美云</author>
    </book>
  </books>
</body>
</html>
```

上述 XML 文件体现了 XML 文件的基本结构, 其基本特点如下:

- (1) XML 文件包含一个声明, 其位置必须在 XML 文件的首行;
- (2) XML 文件中包含若干个标记, 每个标记由开始标签和结束标签构成;
- (3) XML 文件有且仅有一个根标记, 其他标记都必须封装在根标记中, 文件的标记必须形成树形结构;
- (4) 标记的内容定义在开始标签和结束标签之间, 其中可以包含文本或其他标记 (称为该标记的子标记)。

W3C 为 XML 制订了 10 个设计目标, 具体内容包括:

- (1) XML 应该能在 Internet 上直接使用;

- (2) XML 应该广泛地支持不同的应用方式;
- (3) XML 应该与 SGML 兼容;
- (4) 处理 XML 文档的程序应该容易编写;
- (5) XML 可选特性的数目应该无条件地保持最小, 最好是零;
- (6) XML 文件要易读且清晰;
- (7) XML 应该易于设计;
- (8) XML 应设计得正式且简洁;
- (9) XML 文件应该易于创建;
- (10) XML 标签的简洁性应该是最后考虑的目标。

1.2 XML 发展历程及优势

1.2.1 XML 的出现及发展

近年来, 随着 Web 的应用越来越广泛和深入, 人们发现 HTML 语法过于简单且不够严密, 使得它难以表达复杂的形式。尽管 HTML 推出了一个又一个新版本, 并且已经有了脚本(如常用的 VBScript、JavaScript 等)、表格、帧等表达功能, 但始终满足不了不断增长的需求。另外, 由于 HTML 对超级链接支持不足, 多媒体能力较弱, 影响了 HTML 的大规模应用以及用于复杂的多媒体数据处理。另一方面, 由于近年来计算机技术的迅速发展, 各种 Web 浏览器的不断产生, 已经可以实现比当初发明创造 HTML 时复杂得多的 Web 浏览器。所以开发一种新的 Web 页面语言既是必要的, 也是可能的。

有人建议直接使用 SGML 作为 Web 语言, 这固然能解决 HTML 遇到的困难, 但是 SGML 过于庞大, 不利于学习; 同时, 开发一种可以完全实现 SGML 功能的浏览器也比较困难。于是万维网联盟(W3C)建议使用一种精简的 SGML 版本——XML 应运而生了。

W3C 于 1998 年 2 月发布了 XML 的标准。W3C 制定 XML 标准的初衷是定义一种互联网上交换数据的标准。W3C 采取了简化 SGML 的策略, 在 SGML 基础上, 去掉语法定义部分, 适当简化 DTD(Document Type Definition, 文档类型定义)部分, 并增加了部分互联网的特殊成分。因此, XML 也是一种标记语言, 基本上是 SGML 的一个子集。因为 XML 也有 DTD, 所以 XML 可以作为派生其他标记语言的元语言。

1.2.2 XML 与 HTML 的关系

HTML 具有简易且与平台无关等特点, 几乎所有的浏览器都支持 HTML 标记。与 HTML 不同的是, XML 被设计用来传输和存储数据, 而且允许开发者自己定义标记。因此 XML 具有比 HTML 更加强大的功能, 但值得说明的是, XML 和 HTML 是为不同的目的而设计的, XML 并不能完全替代 HTML。表 1.1 给出了 XML 和 HTML 各方面存在的差别。

表 1.1 XML 和 HTML 的对比

比较项目	HTML	XML
是否预设标签	预置大量的标签	否
可扩展性	不具有	具有很好的扩展性
侧重点	如何表现信息	传输和存储数据
语法要求	不要求标记的嵌套	严格要求嵌套和配对,并需要遵循 DTD 或 Schema 定义的语义约束
结构描述	不支持深层的结构描述	对文件的嵌套层次不作任何限制
可读性与可维护性	不易阅读与维护	结构清晰,便于阅读与维护
是否区分大小写	与浏览器相关,大部分浏览器都不区分大小写	严格区分大小写
数据与显示的关系	内容描述与显示混为一体,难以分离	数据逻辑与显示逻辑分离
编辑工具	文本编辑工具,大量的所见即所得编辑工具(如 Dreamweaver 等)	文本编辑工具,大量的 XML 编辑工具(如 XMLSpy 等)
与数据库的关系	没有关系	与关系型数据库的数据表对应,可以进行转换
处理工具	任何浏览器	需要专门的程序

考虑到 HTML 语法不严格在一定程度上影响了网络信息的传输和共享。W3C 吸取了这一经验和教训,对 XML 制定了严格的语法标准。例如,标签都必须有一个开始标签和一个结束标签,所有的标签都必须合理嵌套,即形成树状结构。也就是说,XML 文件必须符合一定的语法规则,只有符合这些规则,XML 文件才可以被 XML 解析器解析,以便利用其中存储的数据。依据对 XML 文档规范的遵守程度,可将 XML 文档分为格式不良好(malformed)的和格式良好(well-formed)的两类,对于格式良好的 XML 文档又可依据其是否使用了 DTD 和 Schema 定义的语义约束分为无效的和有效(valid)的两类。其中格式不良好的 XML 文档是指完全没有遵循 XML 文档基本规则的 XML 文档;有效的 XML 文档不仅需要遵循 XML 文档的基本规则,而且需要使用并遵守 DTD 或 Schema 所定义的语义约束;对于仅仅遵循了 XML 文档的基本规则,却未使用 DTD 或 Schema 定义语义约束的 XML 文档则将其称为格式良好但无效的 XML 文档。

为了检查一个 XML 文件是否格式良好,一个简单的方法就是使用浏览器打开 XML 文件,如果 XML 文件是格式良好的,浏览器将显示 XML 文件的内容,否则将显示错误信息。

例 1.2

```
<?xml version="1.0" encoding="UTF-8"?>
<booklist>
  <book>
    <title>编译原理</title>
    <author>张素琴</author>
    <press>清华大学出版社</press>
    <price>45.00</price>
    <resume>本书介绍编译系统的一般构造原理、基本实现技术和一些自动构造工具。
  </resume>
</book>
<book>
  <title>软件工程</title>
```

```

<author>张海藩</author>
<press>人民邮电出版社</press>
<price>38.00</price>
<resume>本书是软件工程领域的经典教材.....</resume>
</book>
</booklist>

```

这是一个格式良好但无效的文档,图 1.1 给出了该文档 Internet Explorer 中的浏览结果。



图 1.1 格式良好但无效的 XML 文档显示结果

如果将上述文档中最后的</booklist>删除,然后在浏览器中打开该文件,则会在浏览器中出现如图 1.2 所示的错误。

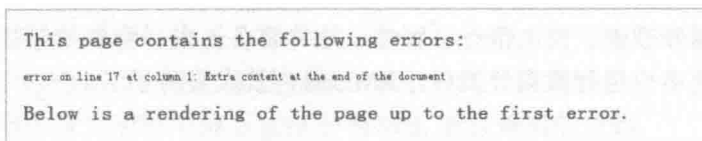


图 1.2 格式不良的 XML 文档显示结果

对于一个格式良好但无效的 XML 文档,进行如下修改即可将其变成一个有效的 XML 文档:①使用 DTD 或 Schema 指定语义约束;②遵守 DTD 或 Schema 所指定的语义约束。下面给出了例 1.2 的 DTD 语义约束。

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE booklist [
<!ELEMENT booklist ((book+))>
<!ELEMENT book ((title, author, press, price, resume))>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT press (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT resume (#PCDATA)>
]>

```

XML 可以很好地描述数据的结构,有效地分离数据的结构与显示,可以作为数据交换

的标准格式，实际上 XML 已经是数据交换领域的行业标准。而 HTML 是用来编写 Web 页面的语言，HTML 同时存储了数据的内容和数据的显示外观，如果只想使用数据而不需要显示，则需要对 HTML 进行专门的处理，例如在 Internet 上广泛使用的搜索引擎，在抓取得到的 Web 页面之后，就需要去除页面包含的标签，保留页面中有用的数据并用于建立索引。另外，HTML 不允许用户自定义标签，目前的 HTML 大约有 100 多个标签。HTML 不是专门用于存储数据的结构，而是主要用于描述数据的显示格式。

1.2.3 XML 的发展前景

XML 自问世以来，一直受到业界的广泛关注。特别是在 1998 年 2 月成为推荐标准之后，许多厂商加强了对它的支持力度，目前已经包含在包括 Microsoft、Oracle 及 IBM 等公司的几乎所有软件之中。

美国微软公司的比尔·盖茨总裁在 Networld+Interop 2000 上做基调讲演时指出：“新一代因特网的关键在于把握 XML”。他在强调该公司的操作系统 Windows 2000 的优越性的同时，还展望了该公司所描绘的未来互联网前景。比尔·盖茨指出“现在的因特网是以 Web 浏览器为中心构成的”。因此，当因特网朝着现实商务中使用的“商务因特网”发展时，Web 浏览器在技术上的限制形成了一大问题。目前必须做的工作就是将 XML 技术导入浏览器。“如果使用 XML，就可以统一多种语言，多种数据格式以及多种表现方式”。因此，微软公司把操作系统和应用软件产品全都与 XML 相对应。XML 将成为具有相互连接特性的因特网标准。

XML 作为表示结构化数据的一个工业标准，为组织、软件开发者、Web 站点和终端使用者提供了许多的有利条件，使得更多的纵向市场数据格式得以建立，并被应用于关键市场，如高级的数据库搜索、网上银行、医疗、法律事务、电子商务等领域。当站点不仅提供数据浏览而是更多地进行数据分发时，XML 语言就大显身手了。

1.3 XML 的优势

XML 作为 W3C 推出的标准，已获得广泛的行业支持，W3C 研究小组确保对工作在多系统和多浏览器上用户间的互用性支持，并不断加强 XML 标准，使其成为一个强大的技术大家族。XML 在采用简单、柔性的标准化格式表达，以及应用间交换数据方面具有革命性的进步。XML 有很多优势，总的来说，XML 不仅提供了直接在数据上工作的通用方法，而且 XML 的优势在于将用户界面和结构化数据相分离，允许不同来源数据的无缝集成和对同一数据的多种处理。从数据描述语言的角度看，XML 是灵活的、可扩展的、格式良好的以及符合指定约束；从数据处理的角度看，它足够简单且易于阅读，几乎和 HTML 一样易于学习，同时又易于被应用程序处理。下面将对 XML 的优势展开详细说明。

1.3.1 良好的可扩展性

在 XML 产生之前，要想定义一个标记语言并推广利用它非常困难。一方面，如果指定了一个新的标记语言并期望它能生效，需要把这个标准提交给相关的组织(如 W3C)，等

待它接受并正式公布这个标准，经过几轮的评定和修改，可能需要几年时间成为一个正式推荐标准；另一方面，为了让这套标签得到广泛应用，制定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定的标签，或者自己开发一个新的浏览器，与现有的浏览器竞争。无论上述的哪个办法，都需要耗费大量的时间和工作。现在借 XML 的帮助，制定新的标记语言要简单易行得多，这也正是 XML 的优势所在。

各个行业会有一些独特的要求。比如说，化学家需要化学公式中的一些特殊符号，建筑家需要设计图样中的某些特殊标记，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使用这些记号，因此不需要这些标签。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签库，并且这个标签库可以迅速投入使用。

不仅如此，随着当今世界越来越多元化，想要定义一套各行各业都能普遍应用的标签既困难，也没有必要。XML 允许各个行业根据自己独特的需要制定自己的一套标签，同时它并不要求所有浏览器都能处理成千上万个标签，同样也不要求标记语言的制定者制定出一个非常详尽、非常全面的语言，从而适合各个行业、各个领域的应用。比起那些追求大而全的标记语言的做法，这种具体问题具体分析的方法实际上更有助于标记语言的发展。现实中，许多行业、机构都利用 XML 定义了自己的标记语言。

1.3.2 内容与形式分离

XML 不仅允许自定义一套标签，而且这些标签不必仅限于对显示格式的描述。XML 允许根据不同的规则来制定标签，如根据商业规则、数据描述，甚至可以根据数据关系来制定标签。

对于 HTML 语言，尽管这也是存储并显示数据的一种可行的方法，但它的效率和能力却非常有限，至少存在以下几个严重的问题：显示方式内嵌于数据之中；在数据中寻找信息非常困难；数据自身的逻辑不得不让位于 HTML 语言规范的逻辑。

与此相比，当使用 XML 数据的表示形式时，以上问题迎刃而解，回顾示例可以看出，现在的标签为要表现的数据赋予了一定的含义。使用 XML 进行数据存储时，数据非常简单明晰，因为它所携带的信息不是显示上的描述，而是语义上的描述。信息的显示方式已经从信息本身抽取出来，单独放在了“样式单”中，这也丰富了显示的样式。这样一来，上面所说的 HTML 的 3 个问题都得到了很好的解决。

1.3.3 遵循严格的语法要求

XML 的标记是程序员自己定义的，标记的定义和使用是否符合语法，需要验证，即 XML 不但要求标记配对、嵌套，而且还要求严格遵守 DTD 或者 XML Schema 的规定。DTD 是一个专门的文件，用来定义和校验 XML 文档中的标记；XML Schema 采用 XML 语法描述，它比 DTD 更有优越性，多个 Schema 可以复合使用 XML 名称空间，可以详细定义元素的内容及属性值的数据类型。

HTML 的语法要求并不严格，浏览器可以显示有语法错误的 HTML 文件。在处理 HTML 文件时，浏览器通常具备一个内置的修改功能去猜测 HTML 文件中漏掉了什么，并试图修

改这个有错误的 HTML 文件。但是，XML 非常注重准确性，一旦语法有什么差错，XML 分析器都会停止对它的进一步处理。就像编译一个程序一样，一个 XML 文档或者被判定为“正确”而被接受，或者被判定为“错误”而不被接受。这是因为 XML 的宗旨在于通过自定义的标签来传递结构化的数据，一个 XML 文档分析器无法像处理一个已有一套固定 DTD 的 HTML 文件那样猜出文件中到底有什么，或者缺什么。

严格的语法要求固然表面上显得烦琐，但一个具有良好语法结构的文档可以提供较好的可读性和可维护性，从长远来看还是大有裨益的。这样大大减轻了 XML 应用程序开发人员的负担，也提高了 XML 处理的时间和空间效率。随着 XML 的自动生成工具和所见即所得的编辑器的广泛使用，XML 的编写者也不必操心 XML 的源代码，更不用去想 XML 的一些琐碎的语法规则。当然，这对于 XML 的开发工具也就提出了较高的要求。

1.3.4 便于不同系统之间的信息传输

在当今计算机世界中，不同企业、不同部门中存在着许多不同的系统。操作系统有 Windows、UNIX 等，数据库管理系统有 DB2、SQL Server、Oracle 等，要想在这些不同的平台、不同的数据库管理系统之间传输信息，不得不使用一些特殊的软件，这样就非常不方便。而不同的显示界面，从工作站、个人计算机到移动终端(如手机、平板电脑等)，使这些信息的个性化显示也变得相当复杂。

有了 XML，各种不同的系统之间可以采用 XML 作为交流媒介。XML 不但简单易读，而且可以标记各种文字、图像，甚至二进制文件，只要有了 XML 处理工具，就可以轻松地读取并利用这些数据，这使得 XML 成为异构系统之间一种理想的数据交换格式。同时，由于 XML 可以很方便地与数据库中的表进行相互转换，使得计算机能够很简易地读取和存储资料，并确保数据结构正确。

1.3.5 具有良好的保值性

XML 的保值性来自它的先驱语言 SGML，SGML 套用十几年历史的国际标准，它最初设计的目标就是要为文档提供 50 年以上的寿命。

我们通过流传至今的大量历史文献知道祖先悠久辉煌的历史，同样，我们的后代也要靠我们留下的文字资料来了解历史。可是现在大部分资料都是电子文档的形式，而且很多没有被打印下来单独存档。若干年后，我们的子孙很可能面对着这些电子文档，苦于没有软件工具能够打开。如果没有 XML，恐怕只有两个办法：要么返璞归真继续使用纸介质，要么不辞辛苦随着软件的更新换代来大规模地转换原有文档到最新格式。SGML 和 XML 不但能够长期作为一种通用的标准，而且很容易向其他格式的文档转化，它们的设计对这一问题给出了圆满的解决方案。

尽管 XML 有诸多优势，但也存在以下不足：

(1) XML 采用树状存储方式，虽然搜索效率极高，但是难以进行插入和修改操作。

(2) XML 的文本表现手法、标记的符号化会导致 XML 数据比二进制表现数据量增加，尤其当数据量很大时，效率就成为很大的问题。