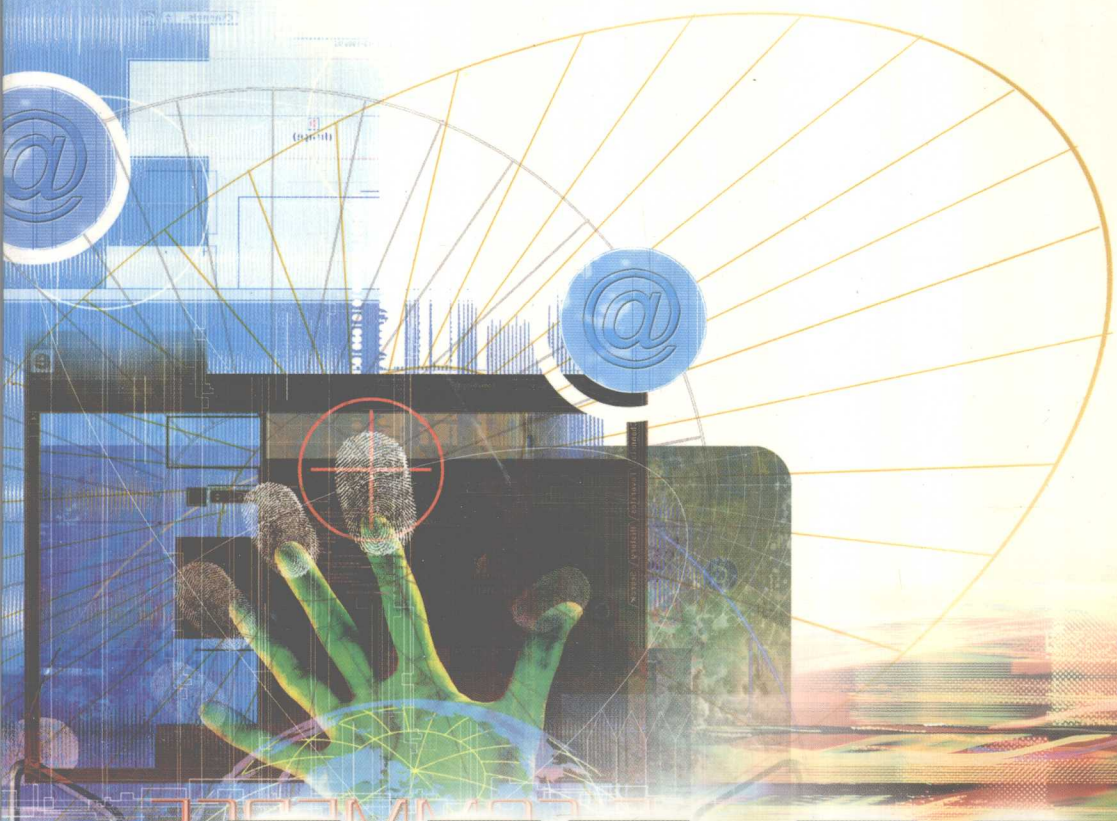
网站建设系列丛书

编著
风火轮小组



XML

从入门到精通

大恒电子出版社

网站建设系列丛书

MXL 从入门到精通

编者　　风火轮小组

大恒电子出版社

2002 年 5 月

内容提要

XML 称为为可扩展的标记语言。相对与其他语言而言，XML 语言功能强大，使用简单方便。许多利用 HTML 难以解决的任务变得简单，使只利用 HTML 不可能完成的任务得以完成；XML 语言的优越性首先在于它良好的的可扩展性能，允许我们自定义标记，开发特定领域的应用程序；其次是它 XML 文档具有自描述性，这使它很容易被各种应用程序理解和处理，因此 XML 被称为网络上的 ASCII 码。

本书从 XML 的基础知识、XML 文档的处理、XML 的应用几个方面对 XML 语言的精髓进行了详尽的阐述，本书既适合初次接触 XML 的读者作为入门教材，也可作为有一定 XML 编程基础的读者作为参考书籍。

网站建设系列丛书

XML 从入门到精通

著译：风火轮小组
责任编辑：李振华
出版：大恒电子出版社
经销：各地新华书店软件连锁店
CD 生产：北京维宝光盘有限公司
文本印刷：北京北七家印刷厂
开本/规格：787×1092　1/16 开　22.2 印章　338 千字
版次/印次：2002 年 5 月第一版　2002 年 5 月第一次印刷
印数：0001-5000 册
本版号：　ISBN 7-900092-79-X/TP.79
定价：50 元（1CD 含配套书）
说明：凡光盘配套图书若有自然损坏、缺页、脱页，本社负责调换。

前　言

XML 是 eXtensible Markup Language 的缩写，意为可扩展的标记语言)。XML 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，即定义了用于定义其他与特定领域有关的、语义的、结构化的标记语言的句法语言。

XML 来自与 SGML，它是 SGML 的一个子集。HTML 也是 SGML 的子集，人们往往拿 HTML 和 XML 来比较。XML 使许多只利用 HTML 难以解决的任务变得简单，使只利用 HTML 不可能完成的任务得以完成。因为 XML 是可扩展的，开发人员喜爱 XML 有许多原因。到底是哪个更令人感兴趣，取决于每个人的需要。但有一点是肯定的，一旦用上 XML，就可发现，它正是解决许多令人感到棘手的问题的有力工具。

使用 XML 给我们带来了很多无可比拟的优势。XML 文档具有可扩展性，允许我们自定义标记，这样就可以涉及与特定领域有关的标记语言，开发特定领域的应用。XML 文档具有自描述性，这使它很容易被各种应用程序理解和处理。XML 被称为网络上的 ASCII 码，称为各种异构应用之间交换数据的标准。XML 实现了跨平台的数据，而 Java 实现了跨平台的程序设计，两者的结合将是一个大趋势。

本书分为四个部分。首先向读者介绍了 XML 的基础知识，这部分内容几乎涵盖了 XML 基础的各个方面。在充分了解 XML 之后，我们介绍了 XML 文档的处理，这部分内容是写给编程人员的。第三部分介绍了 XML 的应用，这部分介绍了当前与 XML 相关的内容，比如 XML 在 WebService、电子商务和数据库等等中的具体应用。相信读者在阅读本书之后，即能够充分掌握 XML 技术本身，同时也能了解 XML 的高层应用。

尽管我们多次的校对，认真的核查，但由于时间有限，本书中一定留有各种错误，敬请读者批评指正。

编者

2002 年 5 月

XML 循序渐进

第一部分　XML 基础

第一章　XML 简介

1.1　什么是 XML？……1
1.1.1 置标语言……1
1.1.2 SGML 简介……1
1.1.3 HTML 的困境……2
1.2　XML 将给我们带来什么……3
1.2.1 XML 的优势……3
1.2.2 XML 的特点……4
1.2.3 XML 的应用……4
1.3　XML 相关技术概览……6
1.3.1 XML DTD……6
1.3.2 XML Schema……6
1.3.3 名字空间……6
1.3.4 XSL 与 XSLT……7
1.3.5 Xlink 与 Xpointer……8
1.3.6 Xpath……8
1.4　XML 标准体系结构……9
1.4.1 XML 体系结构……9
1.4.2 XML 基础标准……9
1.4.3 XML 应用标准……10
1.5 关于 XML 的十种观点……10

第二章 创建一个 XML 文档

2.1 第一个 XML 文档……12
2.2 XML 文档的基本组成元素……14
2.2.1 头部……14
2.2.2 内容……15
2.3 XML 文档书写规范……18
2.3.1 格式良好（Well-formed）……18
2.3.2 几个需要注意的问题……19

第三章 DTD 详解

3.1 概述……24
3.2 DTD 文档的基本结构……25
3.2.1 内部 DTD……25
3.2.2 外部 DTD……27
3.2.3 公用 DTD……29

3.3 理解元素……30
3.3.1 元素类型声明……30
3.3.2 定义元素及其子元素……33
3.3.3 有顺序的子元素……34
3.3.4 重复元素……35
3.3.5 成组元素……36
3.3.6 OR 或……36
3.3.7 可选子元素……37
3.3.8 混合内容……38
3.3.9 空元素……39
3.4 DTD 中的属性……39
3.4.1 定义有效的元素属性……39
3.4.2 属性缺省值……40
3.4.3 属性类型……40
3.5 DTD 中的实体……46
3.5.1 什么是实体？……46
3.5.2 内部通用实体……47
3.5.3 定义内部通用实体引用……47
3.5.4 在 DTD 中使用通用实体引用……48
3.5.5 预定义通用实体引用……49
3.5.6 外部通用实体……50
3.5.7 内部参数实体……51
3.5.8 外部参数实体……53
3.5.9 根据片段创建文档……58
3.5.10 结构完整的文档中的实体和 DTD……67
第四章 XML Schema
4.1 Schema 简介……74
4.1.1 Schema 的产生……74
4.1.2 Schema 的发展历程……75
4.2 Schema 详解……77
4.2.1 Schema 的简单实例……77
4.2.2 Schema 的文件结构……79
4.2.3 用 Schema 定义元素及其内容……80
4.2.4 用 Schema 定义元素属性……82
4.2.5 Schema 中的自定义数据类型……83
4.2.6 Schema 中的注释……85
4.3 Schema 与 DTD 的比较……86
4.3.1 目前形势……86
4.3.2 丰富的类型……87

4.3.3 出现约束……88
4.3.4 枚举……89
4.3.5 DTD 适用的场合……90
第五章 名字空间
5.1 什么是名字空间……91
5.2 XML 中的名字空间……91
5.2.1 命名冲突……91
5.2.2 用名字空间解决命名冲突……91
5.2.3 使用命名空间……93
5.2.4 命名空间与 DTD……96
第六章 XSL 和 XSLT
6.1 XML 文档的显示……99
6.2 CSS 简介……107
6.2.1 CSS 初步……100
6.2.2 深入 CSS 样式表……100
6.2.3 创建 CSS 样式表……104
6.2.4 用 CSS 格式化 XML 文档……104
6.3 XSL 与 XSLT……107
6.3.1 概述……107
6.3.2 XSLT 的概念……108
6.3.3 XSLT 的实例……110
6.3.4 XSLT 的语法……113
6.3.5 XPath 详解……117

第二部分 XML 文档的处理

第七章 XML 文档处理基础
7.1 概述……121
7.2 DOM 简介……121
7.2.1 概述……121
7.2.2 创建一个 Document 对象……123
7.2.3 加载 XML 文档……124
7.2.4 遍历 XML 文档……125
7.2.5 修改 XML 文档……126
7.2.6 Document 接口简介……127
7.2.7 Node 接口……132
7.2.8 NodeList 接口简介……139
7.2.9 NodeMap 接口简介……139
7.3 SAX 简介……141
7.4 DOM 与 SAX 的对比……144

7.5 JDOM 简介 ……145
7.5.1JDOM 概述……145
7.5.2 有关于 XML 的背景知识……145
7.5.3 在 JDOM 中操作一个 XML 文档……146
7.6 XML 解析器 ……147
7.6.1 概述……148
7.6.2 Microsoft 解析器……149

第三部分 XML 应用

第八章 XML 应用标准简介

8.1 XHTML……157
8.2 MathML……158
8.3 SVG ……162
8.4 HDML 和 WML……168
8.5 OEB ……169
8.6 SIML ……169

第九章 Web Service 与 XML

9.1 WebService 概述……173
9.2 XML 在 WebService 体系中的重要作用……179
9.2.1 WebService 的角色……179
9.2.2 两种重要技术 ……180
9.2.3 Web services 体系……180
9.3 SOAP 协议……181
9.3.1 概述 ……181
9.3.2 SOAP 消息交换模型……184
9.3.3 与 XML 的关系……184
9.3.4 SOAP Envelope……185
9.3.5 SOAP 编码……189
9.3.6 在 HTTP 中使用 SOAP……204
9.3.7 通过 SOAP 使用 RPC……208
9.4 WSDL 详解 ……217
9.4.1 为什么使用 WSDL?……217
9.4.2 WSDL 文档结构 ……217
9.4.3 WSDL 文件示例 ……219
9.4.4 Namespace……230
9.4.5 SOAP 消息……231
9.4.6 WSDL 的 Types 栏和 Messages 栏中的 XML Schema……233
9.4.7＜portType＞和＜operation＞元素 ……242
9.4.8＜binding＞和＜operation＞元素……244

9.4.9＜service＞和＜port＞元素 ……247
9.4.10 总结……248
第十章 电子商务与 XML
10.1 概述……249
10.2 BIZTALK 简介……251
第十一章 XML 与数据库
11.1 简介 ……254
11.2 为什么使用数据库？ ……254
11.3 数据和文档的对比……254
11.3.1 数据为主的文件……255
11.3.2 以文档为主的文件……257
11.3.3 数据、文件和数据库……257
11.4 存储和获取数据……258
11.4.1 转录数据……258
11.4.2 将文档结构映射成数据库结构……259
11.4.3 数据类型，空值（Null），字符集设置和其他所有的类似集……261
11.4.4 从数据库的结构生成 DTDs 和逆反过程……263
11.5 存储和获取文件（Documents）……264
11.5.1 内容管理系统和关系数据库……264
11.6 可利用的软件……266
11.6.1 中间件……266
11.7 XML 与数据库之间的转换……269
11.7.1 必要性……270
11.7.2 需要涉及的问题……270
11.7.3 一般的解决方法……272
11.7.4 特殊的数据库例子以及 XML-DBMS……275

第一部分　XML 基础

第一章　XML 简介

1.1　什么是 XML?

XML 的全名是 eXtenxible Markup Language(可以延伸或扩展的标记语言),它的语法类似 HTML,都是用标签来描述数据。HTML 的标签是固定的,我们只能使用、不能修改;XML 则不同，它没有预先定义好的标签可以使用，而是依据设计上的需要，自行定义标签。所以在电子商务的网络时代，用 XML 来建立数据，再由 HTML 来显示，将是设计网页的新方向。

1.1.1 置标语言

XML 是一种置标语言。什么是置标语言? 我们日常书写的语言,被称为书面自然语言。如果在书面自然语言中为了标识某些信息,而加入一些标记,这种书面自然语言就可被称为置标语言(Markup Language)。比如在一段书面语言中,为了说明某一句话的重要,在这句话下面画上底划线。但是,我们在这里解释的置标语言,实际上是一种为了计算机处理而设计的置标语言,其中所用到的标记,往往使用代表一定含义的文字或数字表示。通常的做法是,根据需要,先定义一套助意的标记,然后将这套标记添加到书面语言中去,使书面语言变成置标语言。XML 同样依赖于描述一定规则的标签和能够读懂这些标签的应用处理工具来发挥它的强大功能。谈到置标语言，我们自然会想到 SGML。

1.1.2 SGML 简介

SGML(Standard Generalized Markup Language—标准的通用置标语言)是一种置标语言，实际上它是一种通用的文档结构描述置标语言，主要用来定义文献模型的逻辑和物理类结构。SGML 是 ISO 组织于 1986 年发布的 ISO 8879 国际标准。

一个 SGML 语言程序，要由三部分组成，即语法定义、文件类型定义(简称 DTD--Document Type Definition)和文件实例。语法定义，定义了文件类型定义和文件实例的语法结构；文件类型定义，定义了文件实例的结构和组成结构的元素类型。文件实例是 SGML 语言程序的主体部分。

SGML 是一种用标记来描述文档资料的通用语言，它包含了一系列的文档类型定义（简称 DTD），DTD 中定义了标记的含义，因而 SGML 的语法是可以扩展的。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。于是人们抽取 SGML 的一个微小子集，提出了 HTML 语言。

1.1.3 HTML 的困境

如今 HTML 的应用可谓如日中天，有网络的地方就有 HTML，没有 HTML 的网络是难以想象的。Internet 上不计其数的 WEB 站点，以 HTML 为基础借助于其它如 Applet、ASP、JSP 等技术，为我们展现了一个五彩缤纷的网络世界。

但是 HTML 只是一种表达的技术.它并不一定能揭示 HTML 标记中所揭示的含义.举一个最简单的例子,<h1>Green</h1>这句话在网络浏览器中有特定的表现,但是 HTML 却并没有告诉我们它倒底是什么.Green 只是一个英文单词罢了.HTML 只是告诉计算机如何显示这个单词的大小，却无法描述它的语义，不能告诉我们 Green 具体的内容.Green 在不同的环境之下可能会有不同的意义.可能是一种颜色，还可能是一个姓氏。

HTML 标记的集合是固定的，用户不能新增有意义的能供他人使用的标记.网络浏览器是一个应用平台,以 HTML 作为数据标准。为了产生更具表现力的效果，人们就不断向 HTML 中扩充新的标签。

尽管 HTML 的标签越来越多，其显示力却还远远不够。如果你希望非常精确地表现一些你自己的数据，可能你需要一些现在在 HTML 中尚不存在的标签。比方说，你是一个化学家，你可能需要表现化学分子式中的一些特别的符号。又比方说，你是一个飞机设计师，你希望能够表现飞机的动力引擎。可对于这些，HTML 都望尘莫及。要想满足各行各业对显示的不同要求，显然需要大量的标签，这无疑给当今日益臃肿的 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。你写的 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，不过没关系，浏览器照样能读它。HTML 中的文件可以不具有嵌套关系，比如<h1><h2></h1></h2>，也可以不配对出现，只有<h1>而没有</h1>，更不会要求你在使用标签<h2></h2>的外面一定要保证有<h1></h1>，（在语意上难道不该先有一级标题，再有二级标题吗？）。乍一看，这仿佛对网页制作者而言是个福音，可对浏览器的开发者就是件头痛的事了，他们不得不把大量的精力耗费在文法错误的包容上。相应的，浏览器的程序也要加大，甚至牺牲浏览时的时间效率和空间效率。

另外,还更有一批对 HTML 无可奈何的人,那就是搜索引擎的开发者。因为从 HTML 的标签本身，他们几乎得不到任何有用的信息。如果你要到网上去找出世界上所有关于 XML 的书籍的价钱，天啊，搜索引擎要被你忙坏了。它要分辨网络上哪些“XML”字段对应的是书名，又要知道这些书名所对应的价钱。可能你会说，在我们图书馆的网页中，这不是已经办到了吗？问题就在这里，图书馆是根据内部的数据库来进行搜寻的，数据库中的各个字段都有着明确的含义。

但搜索引擎在网上是根据 HTML 文件来进行搜索的，那些原本条理清晰、层次分明的数据库的内容在 HTML 文件中早就被各种各样的标签搞得混乱不堪，而搜索引擎则不得不在这些混乱的内容中大海捞针。正是在这样的背景下，人们对 SGML 的复杂性和 HTML 很弱的表达能力进行了折中，提出了 XML。图 1-1 显示了置标语言的发展历史。

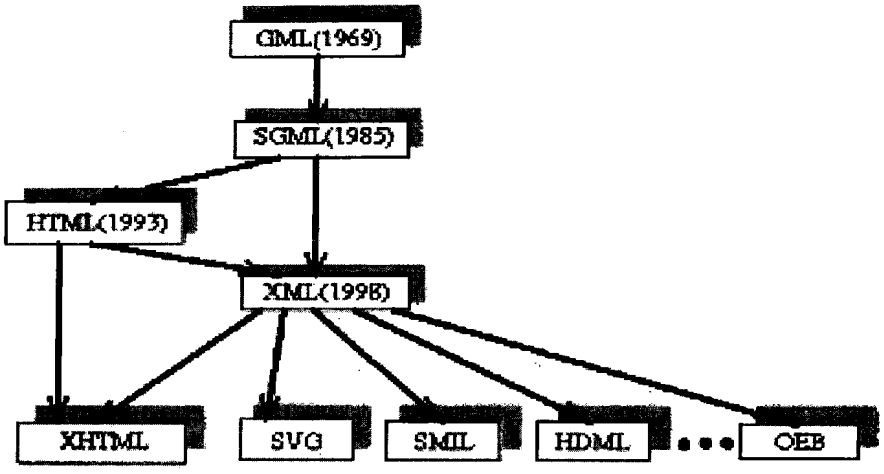


图 1-1 置标语言发展历史图

图中可以看出，随着 XML 的出现，出现了 XHTML、SVG、SMIL 等等置标语言，这些语言是基于 XML 的，它们被人们用于特定的领域，比如 SVG 用于描述矢量图形。

1.2 XML 将给我们带来什么

1.2.1 XML 的优势

XML 开发者源于 SGML 的设计和应用者. 他们已经在 SGML 上投入了大量精力,但他们却发现 SGML 并没有完全发挥它的作用.他们当然有其充分的理由.我们可以列举以下几个重要方面给大家.在这些方面 SGML 带来的影响可以说是一场革命.

XML 仿佛充当着自由宣言的角色，它打破了标记定义的垄断，将网上世界变为一个更加自由民主的世界。

不知你是否清楚在没有 XML 的时候，要想定义一个置标语言并推广利用它是何等困难。一方面，如果你制定了一个新的语言而期望它能生效，你需要把这个标准提交给相关的组织，例如 W3C，等待它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，等到你的置标语言终于熬到成为一个正式推荐标准，可能几年的时间都已匆匆而过了。另一方面，为了让你的这套标记得到广泛应用，你必须为它配备浏览工具。这样，你就不得不去游说各个浏览器厂商接收并支持你的标记，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，无论哪个办法，都令人望而却步！

当然，别以为 XML 的主要目的真的仅仅是为了提供一种祥和的气氛，体现新时代的自由平等的主旋律，它在网络应用中有着确确实实的作用。大家都知道，各个不同的行业可能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，建筑设

计图纸中需要某些特制的标记，音乐家需要音符，这些都需要单独的标记。但是，其它网页设计者则用不着这些记号，也不需要这些标记。XML 好就好在它允许各个组织、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，要想定义一套适合各行各业、能够普遍应用的标记既困难，也没有必要。XML 允许各个不同的行业根据自己独特的需要制定自己的一套标记，但它并不强迫所有浏览器都能处理这些成千上万个千奇百怪的标记，同样也不要求置标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。

1.2.2 XML 的特点

可扩展性：XML 让使用者创建和使用他们自己的标记而不是 HTML 的有限词汇表。可扩展性是至关重要的，企业可以用 XML 为电子商务和供应链集成等应用定义自己的标记语言，甚至特定的行业一起来定义该领域的特殊的标记语言，作为该领域信息共享与数据交换的基础。

灵活性：HTML 很难发展，因为它是格式、超文本和图形用户界面语义的混合，要同时发展这些混合在一起的功能是很困难的。而 XML 提供了一种结构化的数据表示方式，使得用户界面分离于结构化数据。

自描述性：XML 文档通常包含一个文档类型声明，因而 XML 文档是自描述的；不仅人能读懂 XML 文档，计算机也能处理。XML 文档中的数据可以被任何能够对 XML 数据进行解析的应用所提取、分析、处理，并以所需格式显示。XML 表示数据的方式真正做到了独立于应用系统，并且这些数据能重用。所以 XML 适合开放的信息管理。因为它的自描述性，文档里的数据可以由 XML 使能的应用来创建、查询和更新，跟处理传统的关系型数据库、面向对象数据库里的数据类似。XML 甚至还能用来表示那些以前不被看作文档但是对传统的数据库来说又过于复杂而难以处理的数据。所以，XML 文档被看作是文档的数据库化和数据的文档化。

1.2.3 XML 的应用

最初XML的目标是让各种结构的文件都作为统一的网络文件的一部分在网上传输.过去这些文件是 HTML 实现的. HTML 允许指定明确的元素类型说明, 比如特定的商品标号、文档标识或是可测量的数值.和 HTML 相比,XML 允许客户自定义他们的文件元素集合,同时也可以指示这些元素在屏幕上如何按指定的要求表现.

在早期,为了解决怎样在固定的目标之间传输数据,XML 被定义为一种自然的编码形式.在一些方案被考虑之后,一种被称为 RDF (资源描述框架)的方案倍受青睐.RDF 为 XML 提供了元数据编码定义.这就像是一个公用的翻译器,为不同的固定目标之间的数据提供翻译.

、XML 将支持更加专业的数据语言,比如说 OSD(开放软件描述).OSD 是由 Microsoft

和 Marimba 提出的一种新的格式描述语言.在这种格式下,软件在网上能时时检查, 时时刷新版本. 不是等用户自己更新,或由是软件提供商提供类似的服务.当 OSD 镶嵌于 XML 支持的 CDF(频道定义格式)中时, OSD 更能使支持频道的桌面自动地更新.

正如前面提到的,XML 的一个主要目标市场是电子商务.传统 EDI(电子数据交换)机制依靠不同商业之间的强大计算机系统来实现压缩的信息传输.每一条信息在传输使用,提供给用户之前都必须编码.电子商务在网上运作时用户端每填完一个 HTML 的表格之后,都要把表格返还给初始的服务器处理.产品交易,谈判签约,后勤管理,税收报表等等这一些活动的数据处理都集中在了一端.可以预测到,有了 XLL 所链接的行为控制机构和 XSL 所提供的客户端评价功能,将来的数据可以从屏幕上抓取,有必要的话可在客户端处理,在处理数据时,传输给相关用户而不必要改换数据格式.一个类似的协议是 OTP(开放网络贸易协议).它的草稿最初是于 1998 年 1 月发布的.这个协议的制定是为了满足在网上.消费者和销售者之间交易时消息的传输.它同时也允许第三方,比如说供货商,市场评估机构, 消费者保护机构等,来参与使用.

XML 的应用弥补了许多 HTML 的缺陷，我们把它在网上的应用总结为四点:

- 当网络客户必须在不同的数据库之间传递信息时的应用.
- 当需要把大部分从网络服务器载下的数据在用户端处理时的应用.
- 当相同的数据对于不同的用户需要有不同的界面时的应用.
- 当网络情报供货商要把发现的信息精心裁减, 并发送给不同的个人用户时的应用.

表 1-1 详细列出了 XML 与 HTML 的异同点:

比较内容	HTML	XML
可扩展性	不具有扩展性	是源置标语言，可用于定义新的置标语言
侧重点	侧重于如何表现信息	侧重于如何结构化地描述信息
语法要求	不要求标记的嵌套、配对等，不要求标记之间具有一定的顺序	严格要求嵌套、配对，和遵循 DTD 的树形结构
可读性及可维护性	难于阅读、维护	结构清晰，便于阅读、维护
数据和显示的关系	内容描述与显示方式整合为一体	内容描述与显示方式相分离
保值性	不具有保值性	具有保值性
编辑及浏览工具	已有大量的编辑、浏览工具	编辑、浏览工具尚不成熟

表 1-1 XML 与 HTML 的比较

1.3 XML 相关技术概览

为了让读者能够从总体上对 XML 设计到的技术有一个印象，这一部分主要介绍一下 XML 涉及到的主要技术点。

1.3.1 XML DTD

在介绍 DTD 之前，我们先看一下 XML 文档的有效性。简单说来，一个 XML 文档是有效的是指这个 XML 文档是符合使用者需要的文档。有效性的问题将在第二章中详细介绍。保证 XML 有效性的两种手段就是 DTD 和 Schema。

DTD 实际上可以看作一个或多个 XML 文件的模板，这些 XML 文件中的元素、元素的属性、元素的排列方式/顺序、元素能够包含的内容等，都必须符合 DTD 中的定义。XML 文件中的元素，即我们所创建的标记，是根据我们应用的实际情况来创建的。想要创建一份完整性高、适应性广的 DTD 是非常困难的，因为各行各业都有他们自己的行业特点，所以 DTD 通常是以某种应用领域为定义的范围，如：医学、建筑、工商、行政。DTD 定义的元素含盖范围越广泛，那么就越复杂。

DTD 可以是一个完全独立的文件，也可以在 XML 文件中直接设定。所以，DTD 分为外部 DTD（在 XML 文件中调用另外已经编辑好的 DTD）和内部 DTD（在 XML 文件中直接设定 DTD）两种。比如，有几十家相互联系的、合作伙伴关系的公司、厂商，他们相互之间的交换电子文档都是用 XML 文档。那么我们可以将这些 XML 文档的 DTD 放在某个地方，让所有交换的 XML 文档都使用此 DTD，这是最方便的做法，同时也适用于公司内部的 XML 文件使用。

1.3.2 XML Schema

Schema 是一种描述信息结构的模型.它是借用数据库中一种描述相关表格内容的机制.它为一类文件树立了一个模式.这个模式规范了文件中 tag(标签)和文本可能的组合形式。和 DTD 一样，XML Schema 也是用于文档有效性的验证。但是 Schema 是完全基于 XML 的，也就是说 Schema 文件也是符合 XML 规范的。与 DTD 相比，XML Schema 是一种内容“开放”的模型，可扩展、功能强，而 DTD 是内容“封闭”的模型，可扩展性差。并且它支持丰富的数据类型，完全能够满足网络应用特别是电子商务的需求，而 DTD 不支持元素的数据类型，对于属性的类型定义也很有限。此外，XML Schema 支持名称空间机制，可以针对不同情况分别对整个 XML 文档或者是文档局部进行验证，而 DTD 缺乏这种灵活性。

1.3.3 名字空间

XML 的命名空间（Namespace）提供了一种简单的方式，用来解决多 DTD 的 XML 文档中元素名、属性名相冲突的问题。由于 XML 标准越来越丰富，命名空间也变得越来越重要。XML 的命名空间（Namespace）是指用 URI 标识的名字的聚集（collection），在 XML 文档中被用作元素类型和属性的名字。XML 的命名空间同计算科学中传统的命名空间不一样，不管从数学上还是从其内部结构上看，XML 命名空间都不是一个集合（set）。根据 XML 的数据模型，一个 XML 文档包含一棵由元素组成的树。每一个元素有一个元素类型名（标签名）和一些属性；每一个属性包括一个名字和一个值。应用程序一般根据元素类型名和元素的属性确定如何处理元素。XML 1.0 没有命名空间，元素类型名和属性名是由一些规定范围内的字符（NameChar）组成的无结构字符串，类似编程语言里的标志符。这类名字可以称作本地名（local name）。

这种方式在 Web 这样的分布环境里是有问题的。一个 XML 文档也许会用 part 元素描述书的章节，另一个 XML 文档也许会用 part 元素描述汽车的部件。XML 应用程序需要文档之外的附加的信息才可能确定怎样处理一个 part 元素。XML Namespace Recommendation 试图通过扩展数据模型，用 URI 限制元素类型名和属性名，来解决命名冲突的问题。比如，描述书本章节的 part 元素和描述汽车部件的 part 元素就用不同的 URI 修饰。这种本地名和修饰性的 URI 的组合可以称作全局名（universal name）。URI 在其中仅仅是利用其唯一性帮助应用程序区分名字，而不是要向应用程序提供一个资源，所以，URI 可以真地标识了一个资源，也可以没有对应的资源。

1.3.4 XSL 与 XSLT

XML 的一个最重要的特性是把内容和显示格式分开。这样做带来了很大的好处，可以让不同的用户按照各自希望的格式显示同一 XML 文档的数据内容。这也就意味着 XML 文档本身并没有关于格式方面的信息。为 XML 文档提供格式信息的是样式表，适用于 XML 文档的样式表语言有 XSL 和 CSS2 语言。CSS2 语言既可以用于 HTML 文档，也可以用于 XML 文档；而 XSL 是专为 XML 设计的样式表语言，并采用 XML 语法，目前处于 W3C 的推荐标准阶段。XSL 的优势在于它可以用于转换，当然 XSL 也可以把 XML 文档转换为 HTML 格式。而且同一个样式表可以用于多个具有相似源树结构的文档。显示的媒介不仅限于 Web 浏览器，还可以是印在纸上的书和报告等等。

处理 XSL 样式表的是 XSL 样式表处理器，样式表处理器接受一个 XML 文档或数据，以及 XSL 样式表，输出特定样式的显示，其显示格式根据 XSL 样式表确定。这个处理过程分两步进行，首先，从 XML 源树构建一棵结果树，然后，翻译结果树，产生作用于显示器或纸或其它媒介的显示。第一步被称为树转换；第二步称为格式化。XSL 中完成树转换功能的部分又被称为在 XSLT（XSL Transformation）。虽然 XSLT 被定义为 XSL 的一部分，但是它能够独立使用，其输出并不限于 XSL-FO 的形式，也并不限于表现数据的目的。可以把 XSLT 看作处理文档的脚本语言，是文档处理 API 之外的另一种文档处理

手段。

1.3.5 Xlink 与 XPointer

只要将 XML 张贴到 Internet 上，用户当然希望能够对此文档寻址并且可以将这些文档链接起来。标准的 HTML 链接标记可用在 XML 文档中，而且 HTML 文档也可与 XML 文档加以链接。例如，下面的 HTML 代码将链接指向了一个 XML 文档：

```
<a href="http://www.abc.com/abc/xml/abc.xml">
abc
</a>
```

如果用户跟随着链接，浏览器能否显示这个文档，依赖于该浏览器处理 XML 文件的能力。目前大多数浏览器还不能很好地处理 XML 文档。

然而，XML 利用 XLink 来与文档链接，用 XPointer 来确定文档个别部分的位置，就可以有更多的功能。

XLink 使任意元素成为链接，而不只是 A 元素。进一步说，链接可以是双向的、多向的或是指向多个镜像的站点，并选择这些站点中最近的一个。XLink 利用普通的 URL 来标识它链接的站点。

XPointer 能使链接不仅指向特定位置处的特定文档，而且还可指向特定文档的特定部分。XPointer 可以引用文档中的特定的元素，如第一个、第二个或是第十七个特定的元素。XPointer 提供了文档间连接的非常强大的功能，而这些文档不必有包括附加标记的目的文档，正因为如此，其中的个别部分才可以被链接。

进一步说，与 HTML 的锚（anchor）不同，XPointer 不只是引用文档中的一点。XPointer 可以指向一个范围或是一个区域。因而 XPointer 可以用来选择文档的特定部分，或许这样一来，就可以将这部分复制或是将其装入其他程序。

1.3.6 XPath

XPath 是用于描述如何识别、选择、匹配 XML 文件中的各个构成元件，包括元素、属性、文字内容等。该标准最初是从 XSL 标准中分离出来的，但由于其定义的是 XML 中一种常用的功能，为了 XML 标准本身的一致性，该标准不再仅仅为 XSL 标准服务，当需要进行 XML 文档内部元素定位时都采用它所规定的方法。其中 XPointer 就充分地利用了其内容，并在它基础上进行扩展。

XPath 是在 1999 年 11 月 16 日和 XSLT 一起成为正式标准的。XPath 是用作 XSLT 和 XPointer 的对 XML 文档各部分进行定位的语言。它给 XSLT 和 XPointer(XML 文件内部链接语言)提供一个共同、整合的定位语法，用来定位 XML 文件中各个部位。XPath 除了提供一套定位语法之外，还包括一些函数，它们提供基本的数字运算、布尔运算和字符串处理功能。

XPath 使用一个紧凑的、非 XML 的语法来方便实现 XPath 在 XML 属性值中的使用，它基于 XML 文档的逻辑结构，在该结构中进行导航。除了用于定位，XPath 自身还有一