



[返回总目录](#)

目 录

第一章 什么是 XML 语言.....	4
1.1 XML 的产生.....	4
1.2 XML 的基本概念.....	8
1.3 XML 的结构.....	11
1.4 关于 XML 的常见问题.....	14
1.5 小 结	18
第二章 新一代网络编程语言 XML.....	19
2.1 HTML 现状	19
2.2 XML 与 HTML 的比较.....	20
2.3 XML 与 SGML 的比较.....	20
2.4 XML 的优势及其特点.....	20
2.5 XML 的应用.....	23
2.6 小 结	24
第三章 XML 的应用与发展前景.....	25
3.1 XML 行业的未来.....	25
3.2 XML 带给网络应用的技术革命	25

3.3 XML 能够带给我们什么	26
3.4 XML 在各个领域中的运用	28
3.5 小 结	31
第四章 XML 工具	32
4.1 XML 开发工具	32
4.2 XML 浏览器	34
4.3 小 结	35

第一部分 XML 基础

第一章 什么是 XML 语言

第二章 新一代网络编程语言 XML

第三章 XML 的发展前景

第四章 XML 工具

第一章 什么是 XML 语言

本章将向读者介绍 XML 的产生背景、基本概念以及 XML 文档的结构，引导读者进入 XML 的神奇世界。同时还就一些关于 XML 的常见问题作了分析和回答。

本章包括以下内容：

- XML 的产生
- XML 的基本概念
- XML 的结构
- 关于 XML 的常见问题

1.1 XML 的产生

人类思想的交流和智慧的继承与语言文字标记是分不开的。当人类历史上文字符号产生的时候，就不只是一个单一的符号记录，而是由许许多多的记号组成的字符集。当文字产生时，人们在骨头、石头和木器上做标记，后来产生了纸和笔当成作记号的工具。当人们用笔和纸记录语言文字时，由于各种不同的应用需求产生了各种各样的文档格式，如书信、法律文件、各种公文、通知等等。不同的文档必须遵守各自的格式要求，文档中不同的地方有不同的标记来规定文档的格式，使不同的人阅读文档时不会把各种文档格式混淆。进入计算机时代，人们用字处理软件来完成文档的存储、格式编排和出版等工作，这些工具软件一般都是“所见即所得”的工具，掩盖了格式编排任务的复杂性。具体怎样进行格式编排、什么指令规定了标题的大小和字体，用户是不知道的。由于软件的这种标记过程注重的是视觉效果，常常依靠于计算机屏幕和打印机性能。SGML、HTML、XML 都是标志语言，利用各种标记使文档规范化格式化，以便于信息的整理与交流。这些标志语言的发展经历了一个发展过程。首先产生了 SGML，正是由于 SGML 的应用产生了 WEB 和 HTML 语言。由于 HTML 语言满足不了日益复杂的 WEB 信息交流的需要，因此，发展到了 XML 语言。

1.1.1 SGML 的产生

六十年代末，IBM 公司的 Charles Goldfarb、Ed Mosher 和 Ray Lorie 为了解决由不同专用格式创建的法律文件不能在公司不同部门间相互移植的问题，提出了任何可互操作系统的三个主要要求：

- 文件处理程序需要支持一个公共文档格式。
- 这个公共格式对特定的领域是专用的。
- 为了达到高度可靠性文档格式必须遵守特殊规则。

这个文档格式编排系统称为通用标记语言（Generalized Markup Language, GML）。在以后的八年时间中，Goldfarb 继续研究 Generalized Markup Language，并担任了一个国际开发人员组织的技术领导，终于建立了所谓的标准通用标记语言（Standard Generalized Markup Language, SGML），而且国际标准化组织 ISO 于 1986 年最终批准采用 SGML。

SGML 提供描述文档和创建新的一致性衡量准则所必需的公共框架。几乎所有处理文档的语言，都可以至少部分追溯到 SGML。诚然，如果不实行这种描述标准，Web 技术就不存在。SGML 可以用下列三条来描述其特点：

- 不提倡一种特殊的文档结构。
- 不存在必须使用的有限标注集。
- 不限制创建新文档的标准的潜力。

SGML 避免了其它文档格式共同的表示特征，而集中于信息的结构。它不提倡一种特殊的文档结构，而是要求定制数据。只通过结构而不是表现形式约束数据，就能以任何方式迅速编辑、检索、分类、转换或汇集数据，并传递给任何媒体、打印机或屏幕等。正是由于 SGML 的产生导致了超文本标志语言 HTML 的诞生。

1.1.2 Web 和 HTML 语言的诞生

Tim Berners-Lee 是设在瑞士日内瓦的欧洲离子物理实验室的一位英国研究人员。1989 年，他对该研究中心的大量研究资料的不可移植和不可兼容深感失望，并构想出这样一个世界：在其中存取数据是一项简单的任务，以一种一致的方式进行，不考虑使用什么终端和程序，包含任何地点的任何计算机应能只是用一个简单的和常用的程序存取数据。他意识到传统的线性思考不能达到这种跨越系统表达信息的要求，于是导致了超级链接概念的产生。随着 Berners-Lee 建立的超文本标志语言（Hypertext Markup Language, HTML）的开发成功，形成了今天的 Web 信息存储和交流技术。

1.1.3 从 HTML 到 XML

今天，网络的成长速度远远超过了单个人的发展，所以需要比 HTML 更完整的语言来表达网络内容的内涵。人们刚刚看到和学会使用 HTML 来表达自己的“超链接”式的想法，就发现这种语言已经远远不能满足网络潜在应用能力的需要。“扩展标记语言”（Extensible Markup Language, XML）的出现把网络表达的语言集合推进了一大步。HTML 只是一种表达的技术，它并不一定能揭示 HTML 标记中的含义。举一个最简单的例子：<h2>Pine</h2>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到底是什么。Pine 只是一个英文单词罢了，它在不同的环境之下可能会有不同的意义。是一个计算机公司，一棵松树，还是一个人？HTML 并没有告诉我们 Pine 的具体内容。HTML 中有一个大问题就是它的标记的集合是固定的，用户不能新增有意义的可以让其他人使用的标记。网络浏览器是一个应用平台，以 HTML 作为数据标准，网络上的应用程序依赖于服务器上的 CGI 脚本或 ASP 程序等来处理网页上的数据。这样的结构是非常清晰的，但 HTML 却在服务器一端要处理的数据量过大。这使得网络速度低下，效率不高。

SGML 是 HTML 技术的前身。它是文件和文件中信息的构成主体。SGML 与 HTML 不同，它允许用户扩展标记集合，允许用户建立一定的规则。SGML 所产生的标记集合是用来描述信息段特征的，而 HTML 仅仅只是一个标记集合，所以我们可以说，实际上 HTML 只是 SGML 的一个子集。

XML 开发者源于 SGML 的设计和应用者。他们在 SGML 上投入了大量精力，但是却发现 SGML 并没有完全发挥作用。从下面几个方面就可看出。

1. 对 EDI 的支持

EDI 就是电子数据交换，它是网络发展的一个主要目的市场。结构化信息的一个主要目的就要使数据交换成为可能。不同的行业都制定了本行业统一的模型，就像是不同的国家有着不同的语言，这便于本国文化的交流。不同的行业内部信息用统一的模型标识，就能够使信息方便、高效地共享。这样一个统一的模型就是文件类型定义（Document Type Definition, DTD）。当然 DTD 已经落伍了，它正被 XML 的模式（Schema）所替代。很明显的，网络是一个理想的电子数据的集散地。很显然，网络环境下 HTML 是一种有缺陷的数据形式，因为 HTML 不能完全表示不同行业中所需的不同的模型和它的语义。能不能有一种新的语言来解决这个问题呢？答案就是 XML。

2. 对 Java 技术的支持

Java 技术是本世纪最重要的技术发展之一。Java 使浏览器工作时就像在通用的应用平台上，而平台与平台之间却是独立的。但固定的标志集合和 HTML 语义上的贫乏使得 Java 的应用受到了极大的限制。正如前面提到的，在 HTML 中不同的语义无法表现，因此数据元中丰富的信息得不到一种统一的表示，而 XML 却能完全胜任这份工作。HTML 页面要依赖网络服务器上的 CGI 脚本来表现几乎每一个编程函数，这显然使服务器的工作量过大。有了 XML 和 Java 技术，更多的应用软件处理起来将不会占用多少网络通信量。这使得网络更加快捷，客户可以同时应用多个应用软件。XML 真正使 Java 有了用武之地。

3. 信息独立于平台之间

SGML 作为 HTML 和 XML 的前身技术，一直是一种平台之间互相独立的信息技术。这便于指定信息语义的结构。当企业正忙于开发各种各样的信息格式时（比如微软的 RTF，Adobe 的 PostScript 和 MIF，以及 WordPerfect，Lotus，Borland 等公司的格式），SGML 已先人一步，建立了一套严格一致的、独立于平台之间表达信息的格式。但在 80 年代，正当 SGML 悄然兴起时，绝大多数工业计算机的开发者都把目光集中在新的计算机平台上。人们并没有意识到多种私有信息格式可能带来的麻烦。一直到了 90 年代，网络技术崛起以及 Internet 的迅速流行之后，人们才清醒过来，试图寻找一种解决办法。

4. 网络上的 SGML

SGML 已不适用于网络社会的需要。“如何使 SGML 能成功地运用于网络”这一问题已自然而然地提上了议程。1996 年 8 月，GCA（图形通信协会）在西雅图召集 SGML 开发者们举行了一次会议。会议由 Sun Microsystems 公司的 Jon Bosak 主持。论题集中于两大方向：

(1) 在软件应用中 HTML 不是一种理想的信息表现格式。讨论的结果是有必要把 SGML 应用于网络。

(2) SGML 的某些方面已经落伍了，并已成为了它广泛传播的障碍。讨论的结果是有必要考虑怎样修补 SGML。

既然 SGML 有着多重语义的标记集合，它早就应出现在网络上了。然而，在 1996 年 8 月，在网络上人们热衷的却并不是 SGML，而是 HTML 和它的固定的标记集合。SGML 的开发者们立刻制定了一个紧急修改 SGML 标准的方案。因为 SGML 是一个严格而完整的系

统，方便软件应用并不是它的首要任务。所以在 SGML 中有许多语法语义标准。它们既不方便而且消耗昂贵。它们必须被修改或是删除。SGML 开发者们首先做的工作就是得到一个 SGML 中可移给网络的、非关键的结构信息的清单。他们可以基于这个清单对 SGML 进行修改。

5. XML 的诞生

早在西雅图会议之前，Bosak 和一些 SGML 结构信息专家就已向 W3C 提出了“网络上的 SGML”计划。W3C 支持并赞助了他们的努力，工作于 1996 年 7 月正式开始启动。工作的早期，有较大的阻力。因为也存在反对 SGML 的人。一些制定 XML 标准的 W3C 代表甚至声称“网络上的 SGML”是不可能实现的。工作组（原称“SGML 编辑审议委员会”）并未退缩。他们打算让 SGML 以全新的面目出现在网上，给 SGML 以全新的面貌，因此将其命名为“可扩展标识语言（eXtensible Markup Language）”，即 XML。

工作组制定了一个雄心勃勃的计划来展示 XML 的特色，计划的实施分三部分：

- (1) XML 的句法。
- (2) XLL（可扩展链接语言）：XML 的语义链接。
- (3) XSL（可扩展类型语言）：XML 的表现。

XML1.0 版本标准由 W3C 于 1998 年 1 月 10 日正式批准公布，XML 1.0 第二版于 2000 年 10 月 6 日发布（XML 1.0 第二版不是 XML 的一个新版本，它只是为了方便读者，并入了第一版勘误表中指出的错误和修改）。XLL 和 XSL 的工作还处于进一步修订和完善的阶段。

6. 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。过去这些文件是用 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 允许客户定义他们自己的文件元素集合，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出来。在早期，为了解决怎样在固定的目标之间传输数据元，XML 定义为一种自然的编码形式。在一些方案被考虑之后，一种称为 RDF（资源描述框架）的方案受到了欢迎。RDF 为 XML 提供了数据元编码定义，这就象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，软件在网上能实时检查，实时刷新版本，而不是等用户自己更新，或由软件提供商提供类似的服务。当 OSD 内嵌于 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 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，都要把表格发还给初始的服务器处理。产品交易、谈判签约、后勤管理、税收报表等等活动的数据处理都集中在了服务器一端。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），

它的草案最初是于 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信息的传输，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：

- (1) 当网络客户必须在不同的数据库之间传递信息时的应用。
- (2) 当需要把大部分从网络服务器下载的数据在用户端处理时的应用。
- (3) 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。
- (4) 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。

1.1.4 XML 的设计目标

(1) XML 可在因特网上直接使用：这是最重要的一点，设计 XML 的出发点，就是针对 Internet 的迅速发展以及 HTML 的缺点，而寻求一种新的解决方案。

(2) XML 将支持各种各样的应用程序。

(3) XML 应与 SGML 兼容：对于 XML 和 HTML 前身的 SGML，必须提供向前兼容性。

(4) 编写处理 XML 文档的程序将十分方便。

(5) XML 中可任选的特征数目极少，理想情况为零。

(6) XML 文档应当清晰易读。

(7) XML 设计应能迅速完成。

(8) XML 设计应条理清晰、简明扼要。

(9) XML 文档应易于创建。

(10) XML 标记可以是复杂的标记。

1.2 XML 的基本概念

1.2.1 元素

元素就是文档的组件，元素可由其他元素、其他类型数据等组成。

XML 元素间的关系：由于 XML 元素必须规定根元素，所以根元素以下的元素都为子元素，相互之间为兄弟元素关系。为了数据存储方便，应确保子元素内容与根元素相关，并且各个兄弟元素间最好也有逻辑上的关联。

XML 元素内容：XML 文件是用于储存数据的，因此它一定要有内容，而它的内容则是在各个元素之中进行声明。在这里，内容可以是空白内容，也可以是简单的文本或单一数据，也可以是复杂的多个数据列，甚至还可是其他的分类子元素。

XML 元素命名规则：同 Java、C 等命名规则类似，可以是英文字母或中文作为元素名（也可以是其他语言，但要首先确保数据读取软件可以支持该语言），可以用数字和字母作为名字开头，需要注意的是名称中不能出现空格！可以用下划线“_”取代空格的功能来进行详细命名，而且在名称中严禁出现“：”号。

1.2.2 属性

属性是包含关于元素的额外信息的元素部分。

那么，属性值可以做什么呢？XML 中的属性值可以被用来为元素添加额外的说明信息。其实我们很早以前就已经接触过属性值，而且频率非常得高，那就是出现在 HTML 中的属性值。如：

```
<font size="7" color="red"></font>
```

标记中 size 和 color 就是 font 的属性。在 XML 中，我们也可以应用属性值到元素，但不要太频繁，毕竟 XML 是用来储存和发送数据信息的，因此它的可扩展性就显得非常重要。我们可能随时需要向 XML 文件中添加数据，虽然使用属性值可以方便地为元素添加额外的信息说明，但是这样做非常不利于日后的维护和更新。更何况，这些事情使用元素来做才是正确的选择。频繁的使用属性值很难维护和更新。而且，使用属性值还有一系列的弱点：

- (1) 属性值不能包含多重数值（元素可以）。
- (2) 属性值难于扩展（考虑到长远打算）。
- (3) 属性值不能用于描述结构内容（子元素则可以）。
- (4) 属性值很难通过 DTD 来进行一个测试。

1.2.3 文档类型定义

文档类型定义（Document Type Definition, DTD）是一套关于标记符的语法规则。它告诉我们可以文档中使用哪些标记符，它们应该按什么次序出现，哪些标记符可以出现于其它标记符中，哪些标记符有属性等等。DTD 原来是为使用 SGML 开发的，它可以是 XML 文档的一部分，但是它通常是一份单独的文档或者是一系列文档。因为 XML 本身不是一种语言，而是定义语言的一个系统。它没有象 HTML 一样拥有一个通用的 DTD，相反，想使用 XML 进行数据交换的行业或组织可以定义它们自己的 DTD。如果一个组织想用 XML 来标识仅在内部使用的文档，它可以创建自己私有的 DTD。

1.2.4 XML 模式

一些供应商，包括微软，已经提议替代 DTD 的一个方法，称为模式（Schema）。他们已经将其以 XML 数据提交给了 W3C。就像 DTD，schema 提供了文档的规则，并指出用什么标记符，标记符的属性之间的联系等等。但是，不同于 DTD，schema 可以定义数据类型。例如，DTD 可能有一个标记符<PRICE>，而标记符之间的内容可以是数字或字符串。Schema 可以规定只输入数字。这个方法显然有其优点，唯一的问题是它将成为 DTD 规范，还是 XML 的一个扩展。

1.2.5 XML 名字空间

在 XML 中，用户可以自己定义标记和元素。但我们可以很容易想到，如果把多个 XML 文件合并为一个时，就很可能出现冲突。名字空间（Namespaces）就是为了解决这个问题的。对于 XML，Namespace 严格的定义是：Namespace 是用 URI 加以区别的、在 XML 文件的元素和属性中出现的所有名称的集合。有了 Namespace，用户就可以保证在他的文件中使用的名称是独一无二的。在没有 Namespace 的 XML1.0 文件中，元素和属性中出现的名称无异于一族没有结构的字符。我们称它们为 local names（本地名称）。本地名称在网络上是不合

适的。可以想象，网上会有成千上万的人使用同一个名称，而它们却代表了不同的意义。Namespace 通过 URI 区别同名的标识，那么我们完全可以相信不会出现冲突了，因为 URI 是独立的。Namespace 的声明要用到前缀 xmlns。声明的 Namespace 位于指定的 URI，同样的，它也有个名字，我们称这个名字为：Namespace 名，Namespace 名必须独有而且一致。

1.2.6 层叠样式表 CSS 与可扩展样式语言 XSL

在 XML 中内容与表现形式是分开的。这就使得不同的用户可以根据他们自己的需要来定义数据的表现形式。在一个 XML 的源文件中并没有关于表现形式的信息。这一点我们可以从下面的例子中看出：

例 1.1 在 HTML 中的一段代码：

```
<H1>test document1</H1>
<H2>test document2</H2>
<H3>test document3</H3>
<H4>test document4</H4>
<p>test document5</p>
```

这是一段典型的 HTML 代码。在每一个标签中都含有表现形式的信息，而关于标签之间联系和结构的信息却没有，这事实上约束了 HTML 的发展。

例 1.2 在 XML 中的一段代码：

```
<Car Register>
<Car>
  <Registration Number>ABC123 </Registration Number>
  <Make>Saab 9000</Make>
  <Model>1995 </Model>
  <Owner>
    <Name>jone </Name>
    <Address>address</Address>
    <Zip code> Zip code </Zip code>
    <City> City </City>
  </Owner>
</Car>
</Car Register>
```

在 XML 文件之中，标签的信息是非常“纯”的，它没有表现部分，所以有人又称 XML 为传输知识的语言。那么 XML 文件是怎样进行表现的呢？XML 文件的所有表现信息多发放在了样式表（stylesheet）文件当中。样式表文件全权负责 XML 源文件的表现形式。所以说如果一个 XML 源文件对应不同的样式表文件就会有不同的表现形式。有了样式表文件，我们可以对文件表现型的大小、颜色、空白做特定的规定。CSS（Cascading Style Sheets，层叠样式表）和 XSL（XML Style Language，可扩展类型语言）是两种样式表语言。它们是互相补充、各有特色的。CSS 可以展现 HTML 和 XML 文件，而 XSL 可以展现 XML 和转型语言（transformation）。

1.2.7 XML 链接

当人们想到 Web 时，可能就会想到引人入胜的超链接。但是 HTML 只能向一个方向进

行，从用户所在的页面进入用户想去的页面。没有办法链接到页面的特殊部分，也无法轻易地创建外部页面之间的链接。XML 的链接语言就能做到许多新功能的链接。三个 W3C 工作草案文件规范了 XML 的链接和寻址机制，即 Xpath 语言、XLink 语言和 XPointer 语言。

Xpath: 主要目的是进行部分（而不是整个）XML 文档的实际寻址。

XLink: 利用 XML 句法创建结构，来描述目前的 HTML 的单向链接以及更复杂的多端和多类型的链接。XLink 的重要部分是定义两个或多个数据对象或对象部分之间的关系。

XPointer: 建立在 Xpath 的基础上，以支持到 XML 文档的内部结构中的寻址。因此，可以使用 XML 标记链接到另一个文档的特殊部分，而不必提供 ID 引用。

1.2.8 文档对象模型

XML 文档对象模型（Document Object Model, DOM）结构将 XML 文档的内容实现为一个对象模型。W3C 的 DOM Level 1 说明定义了 DOM 结构如何实现属性、方法、事件等等。微软的 DOM 实现完全支持 W3C 标准，并且还有许多使程序更容易访问 XML 文件的新增特点。

要使用 DOM，需要创建一个 XML 解析器的实例。微软公司在 Msxml.dll 中创建了一系列标准的 COM 接口来使创建实例变得可能。Msxml.dll 中包含了类型库和可应用的代码，用户可以使用其处理 XML 文件。如果用户使用可实现脚本的客户端，比如 VBScript 和 IE，就可以使用 CreateObject 方法来得到解析器的实例。

```
Set objParser = CreateObject("Microsoft.XMLDOM")
```

如果使用 ASP，则使用 Server.CreateObject 方法。

```
Set objParser = Server.CreateObject("Microsoft.XMLDOM")
```

如果用户在使用 VB，则可以创建一个对 MSXML 类型库的引用，这样就可以访问 DOM。要在 VB6.0 中使用 MSXML，操作如下：

(1) 打开 Project References 项。

(2) 从 COM 对象中选择 Microsoft XML, version 2.0，如果找不到此项，则需要重新安装获取它。

这样我们就可以创建一个解析器的实例了：

```
Dim xDoc As MSXML.DOMDocument Set xDoc = New MSXML.DOMDocument
```

用户可以通过两个途径获取 Msxml.dll：

- 安装 IE5.0，MSXML 解析器是其中集成的部件。
- 或者到相关网站上下载。

一旦建立了对类型库的引用，就可以执行解析，调入文档；用户从而可以对 XML 文档进行表现和处理。

1.3 XML 的结构

XML 语言，XML 名字空间和 DOM 是 W3C 建议的，这是 W3C 发展过程中决定性的一步。由于它们已经成为正式的规范，开发人员能够用 XML 的格式标记和交换数据。XML 在三层架构上为数据处理提供了很好的方法。

使用可升级的三层模型，XML 可以从存在的数据中产生出来。使用 XML 结构化的数据可以从商业规范和表现形式中分离出来。数据的集成、发送、处理和显示是下面过程中的每一个步骤：

(1) 数据结构，名字空间。XML 名字空间允许开发人员在可识别的情况下定义元素的名称，以避免同名元素间产生冲突。在一个文档中使用的元素，比如购买单，可以在不同的 schemas 中被定义。名字空间保证元素名称不会产生冲突，并且阐明了各个元素的来源，但是不能决定如何处理元素。解析器必须知道每个元素的意义和如何处理它们。来源于不同名字空间中的标记可以混合在一起，这是从不同来源过来的数据所必须具备的。有了名字空间，元素既可以存在于相同的以 XML 为基础的文档中，也可以存在于不同的 schemas 中，说明唯一的语义。例如，在书店的购买单上，一个 title 元素可以包含一个书名，另一个 title 元素可以包含作者名。

W3C 已经发布了 XML 的名字空间，允许元素服务于 URI。即使不同的作者选择同样的元素名称，也不会辨识不清。随着任何人都能发布自己的主页或者浏览他人的主页，名字空间的功能允许使用者定义个人的术语字典或者使用公布的公用名字空间。

```
<orders xmlns:person="http://www.schemas.org/people"
        xmlns:dsig="http://dsig.org">
  <order>
    <sold-to>
      <person:name>
        <person:last-name>Jordan</person:last-name>
        <person:first-name>michael</person:first-name>
      </person:name>
    </sold-to>
    <sold-on>2000-10-01</sold-on>
    <dsig:digital-signature>123456789</dsig:digital-signature>
  </order>
</orders>
```

这段编码告诉读者如果一个元素是以 dsig 打头，它的意义是由 <http://www.dsig.org> 的名字空间所定义的。同样，以 person 打头的元素的意义是由 <http://www.schemas.org/people> 的名字空间所定义的。名字空间保证元素名称不会冲突，也阐明了元素是由谁定义的。它并不给出如何处理元素的指令。读者仍然需要知道元素的意义，并且决定如何处理它们。名字空间只是针对元素名称。处理人员能够定义元素的数据类型和内容的格式，可以使用数据类型名字空间的 dt 属性来达到这一目的。

```
<sold-on dt:dt="date" xmlns:dt="urn:schemas-microsoft-com:datatypes">
2000-09-03</sold-on>
```

在这里，data 说明 sold-on 元素的内容是按照标准格式的，这种格式是由数据类型名字空间说明的。有了元素名称，处理人员可以设计他们自己的数据类型，也能使用共享的类型。微软同 W3C 一起定义了一套标准类型，并且已经在 IE5 中提供部分支持 XML Schema 的第一份数据类型清单。

(2) 数据发送、处理。由于 XML 是开放的、基于文本的格式，它可以通过 HTTP 像 HTML 一样传送。桌面上的数据可以用 DOM 处理。代理商将支持 XML 更新功能，使得中

间层或数据服务器上数据的变化可以传递给客户，反之亦然。因此，代理商能够从客户端得到更新的数据，并把数据传送到储存服务器上。

(3) 解析 XML。IE5 中的 XML 解析器能够读入一串 XML 数据，经过处理，产生一棵结构树，并且使用 DOM 把所有数据元素作为对象。解析器用 CSS 或 XSL 样式表显示数据，或者用脚本把数据进行进一步的处理，或者把数据移交给另外的应用软件或对象进行进一步的处理。DOM 用扩展方式支持名字空间、数据类型、查询和 XSL 转化。

(4) 使用文档对象模式 (DOM) 处理和编辑数据。DOM 实际上是一个应用编程接口，用来定义一种标准方法。通过这种方法，开发人员能够处理 XML 结构树的元素。对象模式控制着使用者如何同结构树交流，并且把所有树的元素作为对象暴露出来。

(5) 用 HTML 显示 XML 数据。XML 文档自身不能决定如何显示信息，而 HTML 是一个理想的显示语言。举个例子来说，网上书店的店员可以访问主页寻找订单。在后台，个人数据记录是用 XML 表示的。但是，在前端，它们是用 HTML 表示的。为了构造这个主页，Web 服务器和 Web 浏览器都需要把 XML 数据记录转变为用 HTML 来表示。

数据捆绑和样式表可以用来把 XML 数据组织成形象化的表达形式，并加上交互功能。数据捆绑是动态 HTML (DHTML) 的一个方面，它把单独的数据从信息源 (例如 XML 文档) 移动到 HTML 显示上来，允许把 HTML 作为显示 XML 数据的模块。微软把 XML 数据源对象 (XML DSO) 作为 IE5 的一部分。XML DSO 能够在 XML 数据岛基础上被调用。

XSL 能够进一步加强这一过程。一个 XSL 样式表包括如何从 XML 文档中取出信息以及如何把它转变为另一种格式的指令。XML 转变为另一种格式，比如 HTML，采用的是一种公布了的方法，这比采用脚本编写简单而且容易理解。另外，XSL 把 XML 作为它的语法，使 XML 的编写者不用去掌握另外的标识语言。

CSS 仍然被应用于结构简单的 XML 数据，并且也很有用。但是，CSS 不提供与数据源结构不同的数据显示结构。使用 XSL，可以产生与原来的 XML 数据结构完全不同的表达结构。XSL 提供内容和表现形式的语义和结构独立性。

(6) 增加 HTML。HTML 页增加语义信息并不容易。很多程序曾经试图用一些非标准的方法来解决这一问题，比如在 HTML 注释中隐藏数据。但是，这样的注释是很难使用的，对象模式并不能理解它们。

为了解决这一问题，W3C 定义了一个格式，用来把基于 XML 的数据放到 HTML 页中。通过使用数据岛 (data islands)，扩展 HTML 允许很大范围的应用软件使用 HTML 作为主要文档和显示格式，并且使用这些文档中内含的 XML 保存数据。

一个 HTML 页包含有关这一页主题的特殊数据。例如，如果这一页显示一位作者最近一部小说的广告，这一页也包括有关这本书的 XML 数据，比如 ISBN 序号、出版者或者是价格。这些信息显不显示并不重要，重要的是这些信息作为数据可被获得和理解。

(7) 转换和查询 XML。随着 XML 作为在 Web 上交换数据的一种标准方式的出现，不可避免地种种需要就会产生，比如查询 XML、制作压缩数据、对数据分类和过滤以及转换 XML 语法。XSL 和 XSL 模式语言提供了满足这些需要的一种方法。XSL 模式是简明的语法用来识别 XML 文档的节点，建立在节点类型、名称、内容和与树中其他节点相关的前后联系的基础上。

XSL 提供了一种语法，使 XSL 模式查询的结果与模板有关，使 XML 源文档中的数据

具体化。XML 语法可以输出，以供分类和过滤，或者把一个模式中的数据转化到另一个模式中去。

(8) 设置字符和编码。XML 中的信息都是用统一的字符编码 (Unicode) 标准编写的，包括元素的内容和名称。因此 XML 支持所有的国际字符的表现形式。

统一的字符编码标准可以直接转换为 16 位字符，但更通常是把它转换为方便使用的或者是简化的某种语言的编码。XML 支持广泛的编码，但要求在一个文档中使用同一种编码。

(9) 空格符。HTML 在多数情况下忽略空格符，而 XML 是针对数据的，因此通过 `xml:space` 属性可以保留空格。例如，下面两种情况是不同的：

```
<title xml:space="preserve"><composer>Tchaikovsky</composer>'s  
    First Piano Concerto</title>
```

```
<title xml:space="preserve">  
    <composer>Tchaikovsky</composer>'s  
    First  
    Piano Concerto  
</title>
```

在 IE5 中 `xml:space="default"` 这一取值在标记间加入了一些装饰用的空格符。

1.4 关于 XML 的常见问题

1.4.1 XML 是什么

XML 全名是 Extensible Markup Language，中文意为可扩展标记语言，是用于标记电子文件使其具有结构性的标记语言。XML 是根据 SGML (Standard Generalized Markup Language) 所制定出来的，所以 XML 可看成是 SGML 的一部份子集，且其设计目的是能广泛地应用在 WWW 上。

1.4.2 XML 的最终标准已经完成了吗

XML 的最终标准还没有完成。XML 是个 W3C 的推荐标准，制订该标准的成员大多来自业界的大公司。使用 XML 无需支付技术许可，而且同平台无关，并且支持良好，所以用户可以使用任何喜欢的工具。“推荐”这个词听起来像个建议，在行话中这实际上就是一个正式标准，而“草案”这个词则表明该标准仍然有待修改，所以基于草案的工具，有可能在标准的后续版本中进行相应修改，这是用户在规划 XML 的大项目时，需要考虑的一点。目前，用户可以依赖的 W3C 的推荐标准包括：XML 1.0、Namespaces、Associating Style sheets 和 XSLT/Xpath。

1.4.3 XML 仅仅适用于网络吗

尽管 XML 技术适用于网络，但却不仅限于网络。对与那些开发数据库项目的开发者来说，XML 也是个功能强大的工具，尤其是当用户打算同其它数据库开发商交换信息时。XML 数据同所定义的语法和每个元素保持一致，共享在一个团体内部定义的词汇表（例如，这些

团体包括化学研究机构，或者计算机部件制造商，或者是企业的客户供应链）。一旦不同的厂家达成协议，数据之间的交换将不成问题。对于结构化的文档来说，除了具有解析和显示数据库信息的能力之外，还可以很容易地构造和使用搜索引擎，以及很容易地采用其他数据管理工具进行管理。XML 可以对同一数据进行重组，以便应用于不同的场合，例如，网站发布，数据库事物处理，无线装置或者打印手册。几乎每一个数据库厂家都支持 XML，或者即将加入对 XML 的支持，大多数信息交换方面的应用软件（包括网站内容管理、目录服务、Lotus Notes 或者中间件服务器）提供了一定程度的 XML 互换性。XML 技术也不仅仅适用于大企业内部的使用，例如 IBM 公司正在研制的语音 XML 项目，可以让用户通过手机访问网站。

除此之外，我们将发现不少 XML 工具，包括分析标识和文件结构的解析器、XML 编写工具、以及 XML 处理器，这些工具可以把主机的数据转换为 XML 格式的数据，用于数据交换，可以在 Apache 网络服务器上加入对 XML 的支持，或者用于不同的系统之间的通信。

1.4.4 XML 是为了取代 HTML 而发展的吗

XML 与 HTML 被制定设计的目标是不一样的，HTML 是为网页设计而制定的，所以 HTML 着重在如何将文件资料显示出来，而 XML 是要制定一种高弹性与扩展性的标记语言，使其能广泛应用在 WWW 上的电子文件，主要着重在如何将文件结构化以方便资料的交换。网页资料当然也是一种电子文件，所以 XML 的应用范围包含到网页，但这并不代表 XML 就要取代 HTML。HTML 在网页设计上依然会扮演一个重要的角色，非 XML 短时间能取代的。

1.4.5 XML 文件中可使用中文吗

不管是 XML 文件本身的内容或是元素的名称都可以使用中文，但重点不在 XML 文件中是否可以使用中文，而是解析器是否能接受中文。

1.4.6 微软领导着 XML 的发展方向吗

尽管微软在 XML 领域很活跃，微软也确实参与制订了 XML 标准及其相关技术，但是由于有很多其他组织和机构的参与，所以说微软领导着 XML 的发展方向是不正确的。微软很早就参与了 XML 标准的制订工作，W3C 的中的大多数委员会都有微软的参与，微软基于 XML 工作草案推出了其软件工具，比如 IE5。同时，微软也对 XML 和 XSL 标准作了些扩展，不幸的是，如果这些扩展未被接纳为标准，微软也不会把这些扩展功能从其软件工具中去掉，这样就使得一些 XML 用户产生混淆。例如，不太标准的 MSXSL 使得一些软件开发商感到迷惑，因为微软的工具同 W3C 制订的标准所规定的不一致。而如果用户忽略微软所制订的这些扩展功能，使用符合 W3C 标准的文档，则会很容易学会 XML 和 XSL。

1.4.7 什么是 DOM

XML 在 Web 的真正实力在于它是如何与文档对象模型 (Document Object Model, DOM) 交互的。DOM 是定义访问文档中数据的机制的界面。使用 DOM，程序员可以用一个标准的方法写动态的内容。换句话说，他们可以使用它来使浏览器文档树中的一部分特定内容按

照一定的方式表现。例如，产生一个小效果，当用户将鼠标移至文字上时，这些文字变成蓝色。Netscape Navigator 和 Microsoft Internet Explorer 都有他们各自的 DOM，但是他们都称将在其下一版本的浏览器中支持 W3C 的标准 DOM。

1.4.8 SGML、HTML 和 XML 有什么联系

SGML 是在文字处理应用中表达数据的一个方法。它已经出现十多年了，XML 和 HTML 都是从 SGML 发展而来的文档形式，可以看作是 SGML 的子集。因此，它们都有一些共同点，如相似的语法和带括弧的标记符的使用。但是 HTML 是 SGML 的一个应用，而 XML 是 SGML 的一个子集。基本上 HTML 不能用来定义新的应用，而 XML 可以。例如，资源描述格式 (Resource Description Format, RDF) 和频道描述格式 (Channel Description Format, CDF) 都是使用 XML 定义的应用。事实上，XML 和 SGML 是兼容的，XML 文档可以通过任何 SGML 制作或浏览工具阅读。但是，XML 没有 SGML 那么复杂，它是设计用于有限带宽的网络的，如 Internet。XML 规范的合作者 Tim Bray 说，XML 的设计出发点是取 SGML 的优点，去除复杂的部分，使其保持轻巧，可以在 Web 上工作。HTML、SGML 和 XML 将继续用于其合适的地方，它们中的任何一个不会使其他一个废弃。HTML 仍是在 Web 上快速发布数据的最简单的方法，包括大部分短期的数据，如会议议程或广告宣传册。如果数据会长期使用，并且需要更多的一些结构，Web 建造者将愿意使用 XML。不同于 HTML 和 XML，SGML 可能永远不会在 Internet 上被广泛接受，因为它从来没有为某个网络协议的需求而设计或优化过。对于高端的、复杂结构的发布应用，SGML 将继续适用。

1.4.9 如何实现 XML

XML 将以几种不同的方式应用。一个是在人机之间交换数据，如从 Web 服务器至用户的浏览器。另外一个是在不同的应用之间交换数据，或者是机器之间交换数据。

在这些情况下，用户都可能需要三层架构：后台数据库，针对数据的处理逻辑的中间层服务器，以及数据进一步显示和处理的客户端。数据库可以从多个数据来源接收信息，可能已经是 XML 格式的数据。中间层然后聚拢数据并在最终的表现层上公布。

现在，网页已经可以用这种方法传送——CNET 的 NEWS.COM 从一个数据库中发表数据，但是要获得一页的新的视图，如 NEWS.COM 的新的“打印机友好”选项，服务器必须产生一个新的页面。一份适当格式化的 XML 文档将允许客户端的应用为不同的媒体修改文档的表现形式，比如为打印机。

1.4.10 什么是构造良好和有效的文件

基本上有两类相关的 XML 文档：构造良好的和有效的。

(1) 构造良好的 XML。文档遵守 XML 语法的一般规则，这些规则比 HTML 和 SGML 的更为严格。XML 的字符数据决不会吊在那里，没有某种结束标识符，或者是像 <MYTAG></MYTAG> 成对出现的结束标识符，或者是一个特别的在右尖括弧前带有一个斜杠的空元素标记，比如 <MYTAG/>；XML 标识总是以左尖括弧或“&”开始；元素类型和属性名称是大小写区分的；属性需要引号等等。

(2) 有效的 XML。文档遵守某个特定的文档类型定义 (DTD)。确认 XML 文档正确性的工作主要由制作发布工具承担，而 XML 浏览器为读取 XML 文档，只需要检查其

构造的良好性。这样，制作工具中的解析器得要检查构造良好性和有效性，而浏览器仅要考虑寻找已经构造良好的 XML。

1.4.11 如何在浏览器中阅读 XML

阅读 XML 文档的工具一般称为 XML 解析器，虽然其更正式的名称是 XML 处理器。XML 处理器将数据传送到应用软件，以做制作、发布、查询或显示。XML 不给应用软件提供应用程序接口，它只是把数据传给应用软件。XML 处理器不解析非构造良好的数据。Netscape 和 Microsoft 都已经将 XML 解析器包含在其浏览器中，或正计划将其包含到浏览器中。

1.4.12 RDF 和 XML 有何联系

如果 XML 是说语言的能力，XML 应用软件则是特定的语言。资源描述格式 (RDF) 是这样的一个应用软件：使用 XML 语法的数据建模语言。

RDF 是描述和访问数据的一个方法。这意味着 RDF 是关于数据的数据，或者说元数据。在 Web 中，这些元数据将被用于建立标准的站点地图，更精确的搜索结果，以及分层次的主题索引。RDF 也支持智能书签，当被索引的网页变化时，书签随之发生变化。如果用户跟踪内容定期更新的站点，比如 CNET 的 NEWS.COM，将很有用。对于网站建造者，建立可被搜索引擎引用的网站内容的元数据并不困难。我们很快就会有商业化的软件，来自动产生给定站点的 RDF 文件。

1.4.13 Netscape 浏览器中如何实现 XML

Netscape 将在 Communicator/Navigator 5.0 中以一个代号为 Aurora 的交付部件来支持 XML 元数据。Aurora 利用 RDF 来获得 Netscape 所说的“桌面信息全面集成”。

Aurora 在网络、桌面和数据库之间查找和管理信息。它将在桌面上以“窗口”菜单的界面出现，会聚集指向当前项目、研究主题、或日常活动等资源的指针。RDF 使 Aurora 的导航条指向不同数据类型（文字处理文档、表格数据、电子邮件消息、数据库内容）的本地文件，也指向 Internet 或 Intranet 服务器上的资源（搜索和查询的结果、书签链接等等）。

一个读取 RDF 的 XML 解析器将是 Netscape 5.0 版浏览器的一部分，并且在最终产品交付之前会出现在某个 beta 版本中。除了这个初始的 RDF 实现，Netscape 正计划将一个通用的 XML 解析器包含在浏览器中，而其浏览器可以和其他的 XML 应用软件一起工作，比如 hakespeare markup（一个早期的 XML 应用软件）、Chemical Markup Language (CML)、以及 MathML（一个正成为 W3C 推荐的数学标识语言）。

1.4.14 Microsoft 浏览器中如何实现 XML

微软的 Internet Explorer 4.0 是第一个实现 XML 的 Web 浏览器。微软提供了一对 XML 处理器：浏览器所携带的用 C++ 写的解析器，和一个 Web 建造者可以从中下载和加入他们自己的应用程序的 Java 解析器的源代码。Java 解析器是一个有效的解析器，就是说它根据一个文档类型定义或模式来进行检查。为了提高性能，浏览器所带的 C++ 版的解析器是一个非常有效的解析器。据微软的产品经理 Steve Sklepowich 称，这两个解析器都是“通用的”，因为它们不依靠特定的 XML 应用，如 Channel Definition Format。由于 XML 数据和其表现分离，在一个 Web 浏览器本身实际显示 XML 的能力需要样式表，例如 XSL (Extensible Style

Language)。同时，微软使用了所谓的 XML 数据资源对象，或叫 XML DSO。它应用了动态 HTML 的数据捆绑能力，将一端的 XML 数据和另一端的 HTML 数据相链接。IE 4.0 访问 XML 文档，从中查询数据，然后作为 HTML 显示出来。

微软也使用了 XML 对象模型 (Object Model, OM) 来让开发者与浏览器中的 XML 数据进行交互。它的实现是通过将 HTML 作为基于 DOM 的对象显现，尽管 HTML 和 DOM 并非直接兼容。DOM 让脚本和程序访问结构化的 XML 数据。

Sklepowich 说，虽然目前在微软，实现 XML 的重心在浏览器部分，但 XML 将最终出现在“任何 HTML 已经出现的地方”。盖茨已经公开宣布微软 Office 未来的版本将支持 XML，而且公司也计划支持电子邮件包和制作 XML 工具的标准。

1.4.15 XML 如何改进超链接

XML 超链接比基本的 HTML 风格的超链接多了一些新的特性，包括无需手写许多 JavaScript 代码就能创建智能型的链接。而且在 XML，链接本身成为了对象，可以像其他对象一样被管理。原来的链接规范——XLL，或 XML 链接语言，正被分为两个不同的规范：XPointer 和 XLink。

XPointer: 在 HTML，要链接到一个页面的中间，页面作者必须在那儿加上定位标识符。使用 XPointer，你可以“取址到”（而不是“链接到”）其他人的文本的任何部分。显而易见，这样将有助于工作于法律文件，科学和学术论文，甚至 W3C 规范！

XLink: 当用户点击一个 HTML 超链接时，当前的网页被连接到的文件替代。XLink 可以使 Web 建立者给链接增加行为。例如，现在，必须用一些 JavaScript，使在链接处弹出一个独立的窗口，但是 XLink 让 Web 建立者对链接进行编码来执行一系列动作，包括弹出一个链接选择的菜单。

另一个应用可以是弹出一个对话框，可能是一个提醒用户他们正要更新数据库的警告。链接弹出菜单可能需要用户点击一个框来表示在进一步处理前他们接受警告。而目前，实现这样的功能要写许多的脚本代码。

XML 也让 Web 建立者创建类似 Web 环工作的 Extended Link，Web 环是通过“下一个/前一个”行进来导航用户选择出来的关于相同主题的网站组。对于弹出菜单太长的相关链接站点表，Web 建立者可以创建一个链接表，这个表对应不同的站点、页面时会有变化。用户可以点击一个图标来自动转移到环中的下一个成员。目前，实现这样的功能需要 CGI scripts，而 Extended Links 提供了一个标准的、非私有的建立资源间联系的方法。

仍然有很多的问题需要解决，特别是在行为策略方面。必须有一种方法来协调以下三方面的关系：文档作者对链接所建议的行为，用户所喜好的显示链接信息的方式，以及是否和何时忽略用户的意愿的策略。

1.5 小 结

通过本章的介绍，读者对 XML 将有一个初步的认识和了解，这是一种令人兴奋的具有明显特性的崭新技术，它的出现，使整个网络应用都呈现出新的活力，每个人都将从 XML 技术中获益良多。

第二章 新一代网络编程语言 XML

本章对 XML、HTML 和 SGML 进行了综合的比较，从各个方面突出了 XML 的特点和优势，概括性地提出了 XML 的应用类型。

本章包括以下内容：

- HTML 现状
- XML 与 HTML 的比较
- XML 与 SGML 的比较
- XML 的优势及其特点
- XML 的应用

2.1 HTML 现状

网络技术发展到今日，的确已渗透到社会生活的每一个角落。而令网络如此平易近人，为大众所接受，单靠 TCP/IP 是远远不够的，由此，HTML——超文本标识语言便应运而生，并以其简单精练的语法、极易掌握的通用性与易学性，使 Web 网页可以亲近于每一个普通人，互联网也才得以普及发展以至今日的辉煌。

然而，电子商务、电子图书、远程教育等全新领域异军突起，迅猛发展并逐渐成为互联网世界必不可少且非常重要的组成部分。随之而来的是 Web 文件的复杂化、多样化、智能化，于是高容量、高信息量、高效率便成为网络信息传输技术发展的追求。与此同时还有另一种需求变得愈发广泛而迫切，那便是同样的数据能否根据不同用户不同需求而以不同的效果和形式表达、再现出来。举一个简单的例子来说，假设我们要在网上拍卖一辆“奔驰”汽车，而在一个 Web 文件中包含了有关这辆“奔驰”汽车的全部信息，包括汽车简介、生产厂商、性能简介、拍卖行信息、拍卖底价、各竞价方名称、各竞价方出价、最高价等等。而在客户端有各种用户，如拍卖行的经纪人，他并不需要任何有关汽车本身的信息，而只需看到一个包含所有竞价方及其竞价的统计列表；而那些竞价方则希望看到自己喜欢的有关汽车的资料，他们有的喜欢汽车外形，有的喜欢汽车的内部装饰，有的则只关心汽车的行驶里程等等；另外一些人也许只想看看“奔驰”汽车即可；也许还有一些人，他们根本就不关心什么“奔驰”汽车、拍卖，只想让计算机接收到这个文件后能够自动地将其分类，并放到自己创建的与之相对应的数据库中。这样一来，这个 Web 文件必须具有极高的复杂性与灵活性，以应付如此复杂多变的具体应用。HTML 所欠缺的恰恰就是这一点，它的简单与普遍性原则虽然曾使 Web 的推广受益匪浅，但反过来也成为其发展的最大的桎梏，尽管人们已付出很多努力，包括各式各样的修改扩充，如增加表格、框架、脚本语言等等，但均未能从根本上改变 HTML 的局限性。于是人们开始全力重新构造 Web，并相继诞生了许多新的技术，但由于种种原因大都没能普及而生存下来。唯有一项，不仅在使用中得到各方肯定，更在短短两年内迅速崛起，并得到 Microsoft、IBM 等各大公司的全力支持，竞相开发，成为一股不可遏抑的席卷全球的浪潮，这便是 XML——可扩展标识语言。

2.2 XML 与 HTML 的比较

与 HTML 一样，XML 也源自 SGML，它保留了 SGML 80% 的功能，使复杂程度降低了 20%。尽管如此，XML 却有着 HTML 语言所欠缺的巨大的伸缩性与灵活性。XML 不再像 HTML 一样有着一成不变的格式。XML 实际上是一种定义语言，即使用者可以定义无穷无尽的标记来描述文件中的任何数据元素，从而突破了 HTML 固定标记集合的约束，使文件的内容更丰富更复杂并组成一个完整的信息体系。

XML 语言可以让信息提供者根据需要，自行定义标记及属性名，也可以包含描述法，从而使 XML 文件的结构可以复杂到任意程度。XML 主要有三个要素：模式（Schema）、可扩展样式语言（eXtensible Stylesheet Language, XSL）和可扩展链接语言（eXtensible Link Language, XLL）。模式规定了 XML 文件的逻辑结构，定义了 XML 文件中的元素、元素的属性以及元素和元素的属性之间的关系，它可以帮助 XML 的分析程序校验 XML 文件标记的合法性；XSL 是用于规定 XML 文档样式的语言，它能在客户端使 Web 浏览器改变文档的表示法，从而不需要再与服务器进行交互通信；XLL 将进一步扩展目前 Web 上已有的简单链接。

良好的数据存储格式、可扩展性、高度结构化、便于网络传输是 XML 主要的四大特点，决定了其卓越的性能表现。由于 XML 能针对特定的应用定义自己的标记语言，这一特征使得 XML 可以在电子商务、政府文档、报表、司法、出版、联合、CAD/CAM、保险机构、厂商和中介组织信息交换等领域中一展身手，根据不同的系统、厂商提供各具特色的独立解决方案。

2.3 XML 与 SGML 的比较

SGML 是一个创建语言的语言，利用 SGML 能创建标注集。例如，航空运输协会利用 SGML 创建一个飞机维护文档标记集；汽车工程师协会使用它创建一个汽车服务手册标注集。XML 是 SGML 的一个子集，在很多方面就像是一个琐碎的 Web 版本。

XML 与 SGML 的差别是实际上使得 XML 更加适合 Web 的分布式环境。实际的差别相当程度上是技术方面，与标注形成和缩写方式、字符和实际引用、注释的结构以及其他限界符有关。

2.4 XML 的优势及其特点

XML 给基于 Web 的应用软件赋予了强大的功能和灵活性，因此它给开发者和用户带来了许多好处。

2.4.1 更有意义的搜索

数据可被 XML 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都是不同的。而有了 XML，搜索就将变得

十分容易。例如对于图书信息的搜索，图书的相应信息可被 XML 唯一进行标识，这样，书就可以很容易以标准的方式按照作者、标题、ISBN 序号或其他的标准分类，搜索书就变得十分方便。

2.4.2 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做进一步的处理。或者它可以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 应用软件的开提供了所需的技术。

2.4.3 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。XML 能够使不同来源的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的服务器上对从后台数据库和其他应用得来的数据进行集成。然后，数据就能被发送到客户或其他服务器做进一步的集合、处理和分发。

2.4.4 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，从描述搜集的 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。

2.4.5 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。数据计算不需要回到服务器就能进行。分离使用者观看数据的界面，使用简单灵活开放的格式，可以给 Web 创建功能强大的应用软件，这些软件原来只能建立在高端数据库上。

2.4.6 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 描述数据本身的结构。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、使用者选择或其他标准决定的方式动态地表现出来。CSS 和 XSL 为数据的显示提供了公布的机制。

2.4.7 粒状的更新

通过 XML，数据可以粒状地更新。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，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。目前，只要一条数据变化了，整个 Web 页都必须重建。这严重限制了服务器的升级性能。XML 也允许加进其他数据，比如预测的温度。加入的信息能够流入已存在的页面，不需要浏览器发一个新的页面。

2.4.8 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变化。

2.4.9 升级性

由于 XML 彻底把标识的概念同显示分开，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。这是一种强大的机制，使得客户计算机同使用者间的交互作用尽可能地减少了，同时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。

2.4.10 压缩性

XML 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 数据是否要压缩要根据应用来决定，同时还取决于服务器与客户间数据的传递量。XML 能够使用 HTTP1.1 中的压缩标准。

2.4.11 开放的标准

XML 基于的标准是为 Web 进行过优化的。微软和其他一些公司以及 W3C 的工作组正致力于确保 XML 的互用性，以及为开发人员、处理人员和不同系统和浏览器的使用者提供支持，并进一步发展 XML 的标准。

XML 包括一套相关的标准：

- 可扩展标识语言 (XML) 标准，这是 W3C 正式批准的。这意味着这个标准是稳定的，完全可用于 Web 和工具的开发。
- XML 名字空间标准，用来描述名字空间的句法，支持能识别名字空间的 XML 解析器。
- 文档对象模式 (DOM) 标准，这为给结构化的数据编写脚本提供了标准，这样开发人员就能够同计算机在基于 XML 的数据上进行交互作用。
- 可扩展类型语言 (XSL) 标准，这是一个工作草案。XSL 有两个模块：XSL 转换语言和 XSL 格式化对象。转换语言可用来转换 XML 以满足显示要求。由于 XSL 的两部分是模块，转换语言能够独立的用来进行多用途的转换，包括把 XML 转换成结构完整的 HTML。CSS 可应用于结构简单的 XML 数据，但不能以不同于信息如何传递来的方式显示信息。
- 可扩展链接语言 (XLL) 标准和 XML 指针语言 (XPointer) 标准是当前的工作草案。XLL 提供类似与 HTML 的链接，但功能更强大。例如，链接可以是多方向的，可以存在于对象上而不仅仅是页面上。IE5 内在不支持 XLL。

2.4.12 微软产品的支持

微软 IE4.0 支持通用的 XML 解析器、XML 对象模式和 XML 数据源对象。IE5 和 Office 2000 支持许多添加的 XML 特性和技术。

2.4.13 新的机会

作为表示结构化数据的一个工业标准，XML 为组织、软件开发者、Web 站点和终端使用者提供了许多有利条件。更多的纵向市场数据格式建立起来，被应用于关键市场诸如高级的数据库搜索、网上银行、医疗、法律事务、电子商务和其他领域，这使得机会更进一步地扩大。当站点更多地进行分发数据，而不仅仅是提供数据浏览时，新的机会就产生了。

由于当前电子商务的迅猛发展，客户服务正在逐渐转移到 Web 站点上来，而且将会由于 XML 的强大功能得到更多的好处。并且，由于大多数商业应用软件包括数据的处理和转移，如购买单、发货单、客户信息、合同、图纸等等，XML 将会改革终端用户在 Internet 上的行为，许多商业应用将能实现。另外，使用基于 XML 的面向企业内部互连网的词汇库，Web 站点上的信息，无论是储存在文档中还是数据库中，都可以被标识。这些词汇也能够对那些需要在客户和供应商之间交换信息的中小型企业提供帮助。

一个重要的未开发的市场是开发使终端用户很容易建立自己的 Web 站点的工具，包括用来从数据库信息和存在的使用者界面中产生 XML 数据的工具。另外，标准模式可以开发用来描述数据，可以使用规划、图表、Excel 或其他电子数据表的功能。开发公布的用来描述从数据库中产生的 XML 的可视化工具是个很好的机会。观看 XML 数据的工具可以用 Visual Basic, Java 和 C++ 编写。

XML 需要强大的新工具用来在文档中显示丰富而复杂的 XML 数据，可以在分层的动态变化的数据上映射用户友好的显示层来实现这一目的。XML 数据的布局图包括数据透视表等。

Web 站点可以提供股票报价、新文章或实时的交易数据。通过制定信息老化的规则，信息超载可以避免。开发用户用来制定规则、服务器和客户软件用来实现规则的基于 XML 的工具有着广阔的发展前景，可以用脚本编写一个标准对象模式用来过滤进来的信息、检查储存的信息、创建输出的信息、进入数据库等等。

2.5 XML 的应用

总的说来的 XML 的应用可分为四类：

(1) 应用于客户需要与不同的数据源进行交互。数据可能来自不同的数据库，它们都有各自不同的复杂格式。但客户与这些数据库间只通过一种标准语言进行交互，那就是 XML。由于 XML 的自定义性及可扩展性，它足以表达各种类型的数据。客户收到数据后可以进行处理，也可以在不同数据库间进行传递。总之，在这类应用中，XML 解决了数据的统一接口问题。但是，与其他的数据传递标准不同的是，XML 并没有定义数据文件中数据出现的具体规范，而是在数据中附加 tag 来表达数据的逻辑结构和含义。这使 XML 成为一种程序能自动理解的规范。

(2) 应用于将大量运算负荷分布在客户端，即客户可根据自己的需求选择和制作不同的应用程序以处理数据，而服务器只须发出同一个 XML 文件。仍以上例为论，如按传统的“客户/服务器”工作方式，客户向服务器发出不同的请求，服务器分别予以响应，这不仅加重服务器本身的负荷，而且网络管理者还须事先调查各种不同的用户需求以做出相应不同的程序，但假如用户的需求繁杂而多变，则仍然将所有业务逻辑集中在服务器端是不合适的，

因为服务器端的编程人员可能来不及满足众多的应用需求，也来不及跟上需求的变化，双方都很被动。应用 XML 则将处理数据的主动权交给了客户，服务器所作的只是尽可能完善、准确地将数据封装进 XML 文件中，这样用户可以根据自己的需求有选择地获取数据。XML 的自解释性使客户端在收到数据的同时也能理解数据的逻辑结构与含义，从而使广泛、通用的分布式计算成为可能。

(3) 应用于将同一数据以不同的面貌展现给不同的用户。这一应用也可在上例中体现出来。这就好象是同样的一份材料，不同的厨师却可以使用这些相同的材料做出不同的菜肴，而且各有各的妙处。这一应用适应当前网络用户界面个性化、风格化的发展趋势。

(4) 应用于网络代理对所取得的信息进行编辑、增减以适应个人用户的需要。有些客户取得数据并不是为了直接使用而是为了根据需要组织自己的数据库。比方说，教育部建立一个庞大的题库，考试时将题库中的题目取出若干组成试卷，再将试卷封装进 XML 文件，接下来，在各个学校让其通过一个过滤器，滤掉所有的答案，再发送到各个考生面前，未经过滤的内容则可直接送到老师手中，当然考试过后还可以再传送一份答案汇编。此外，XML 文件中还可以包含诸如难度系数、往年错误率等其他相关信息，这样只需几个小程序，同一个 XML 文件便可变成多个文件传送到不同的用户手中。

综合以上四种不同类型的应用，我们可以总结出，XML 其实就是一种数据公有，各取所需的标准。具体说来便是数据制作者并不考虑日后这些数据具体会有哪些用途，只是尽量全面地考虑今后有可能会被用到的信息，并将其完整、规范地制作成 XML 文件，服务商则不会被拘束于特定的脚本语言、制作工具及传输引擎的之内，而是提供一种标准化、可独立销售、有级别操作的领域，在那里不同的制作及传输工具将各显神通，一决雌雄，从而极大地满足客户的需求，成为“最信得过”的服务商。

2.6 小 结

XML 的诸多优点及先进性令其刚一出现，便备受网络开发者的青睐。微软、IBM 对其异常重视，大有得 XML 者得天下之势。尤其是微软公司行动最为迅速，在 XML1.0 刚一发布，微软便迫不及待地宣布，微软的 Office 2000、Windows 2000 都将完全采用 XML 格式（现已上市并集成了 XML），而 IE4.0、IE5.0 则早已声称可支持 XML 了（实际上是部分支持）。商业公司如此，国外的研究机构更是对其宠爱有加，如 W3C（万维网联盟）、www.gca.org、www.irt.org 等等，各类 XML 站点如雨后春笋、层出不穷。W3C 正在号召全世界的开发人员共同对 XML 进行定义和开发，XML 正走着与 Linux 相同的发展道路，但它生逢其时，势必将比 Linux 更加成功。

第三章 XML 的应用与发展前景

XML 本身的特性和优点加上业界的鼎力支持,使得它具有良好的应用和发展前景。XML 将给网络应用带来全新的感受,给我们带来更为便利和快捷的工作、生活、信息交换方式以及全新的观念,我们将在各个领域感受到 XML 的卓越性能和“不俗”表现。

本章包括以下内容:

- XML 行业的未来
- XML 带给网络应用的技术革命
- XML 能够带给我们什么
- XML 在各个领域中的运用

3.1 XML 行业的未来

在不久的将来,XML 工作组的许多草案会成为 W3C 推荐的标准。关于草案的最新消息,请访问 <http://www.w3.org/xml> 和 <http://www.oasis-open.org/cove>。XML 将被普遍采用,这是无可置疑的。除了积极参与扩展标准并在这些扩展基础上开发新产品的人以外,XML 将不再是每个人的热门话题。几乎所有的产品都将能够把文档保存为 XML。XML 行业将有一个辉煌的未来。

3.2 XML 带给网络应用的技术革命

XML 将给网络进入下一个阶段带来根本性的催化作用。今后几年的网络上,将出现更智能的搜索引擎。用户想搜索一个关键词的时候,出现一大堆毫不相干的搜索结果的情况将不会再出现。网络站点采用 XML 技术能够实现高度的个性化模式,让用户以自己定义的方式获取有用的信息。

在商业上,不同的商业系统之间采用相同的词汇(在 XML 中,也要实现定义一些能够规范描述性标识的模式,称为 Schema)进行自动化的电子商务。一家在中国北京的网络商店能够随时自动更新来自全国几百家供应商的商品目录和价格信息,并根据商品类别为网络用户提供最直观的“购物环境”。另外一种应用于金融机构中的可交换数据格式 OFX(采用 XML 标准制定),更让商家之间交换金融信息,如电子支付、财务分析、咨询文件等,成为自动化的工作。

在科研教育中,XML 能够描述从前人们只能从图形符号上才能够理解的复杂公式,让科学家和普通的学习者都能够从网络上直接交换具有明确意义的学科知识。一种标准的可交换的网络教学课程规范(IMS)也正在成为远程教育和“虚拟学校”的信息表达格式。

出版行业中,目前正在开发中的“电子书”标准,用可交换的描述方式描述今天书籍中的内容和版权信息。今后的“图书”可能都借助一种方便的手持设备在街头“电子书店”

下载最新的图书。绵延千年的纸质图书和传统出版业将面临巨大的挑战。

此外，今后政府部门之间传递的公文和统计数据也将采用基于 XML 的标准格式。信息能够有方向地在各种职能部门之间传递和“过滤”，让政府工作效率大大提高。

XML 给网络上的各种应用带来第二次革命性的变化（HTML 算得上第一次革命），也会创造出更多的新型应用机会。

3.3 XML 能够带给我们什么

通过前面的介绍，我们现在已经对 XML 的技术意义有了一定的了解，但是它能够给我们的工作和生活带来一些什么实际性的改变？或者说可以让我们在哪些方面享受到这项令人激动的技术带来的好处？这也许是每一个读者（也包括笔者自己）除了技术因素之外所最为关心的问题。其实，这也是每一项新技术出现并进入应用领域时所遇到的相同的疑问，技术必须为人服务，这样才能拥有广阔的发展空间。

XML 使许多 HTML 难以解决的任务变得十分简单，使许多 HTML 不能完成的任务得以完成。XML 是可扩展的，它可以适应每个人的不同需求，它允许不同的专业（如：物理、数学、文学等等）开发与自己的特定领域相关的标记语言；可以使特定领域的开发人员向本领域外的人分发文档和资料；使 Web 页面上数学家书写公式与音乐家创作音乐不会互相干扰。XML 采用简单而标准的数据格式和规范，便于数据的存储、读取、处理，比如：20 年后的开发人员阅读 XML 技术文档时不会因为几个标记的损坏和错误而无法浏览和理解。XML 作为一种标准的数据格式，也可以用来在不同的应用中交换数据，当前的 OFX（Open Financial Exchange，开放财务交换）就是一个具体的实例，它让商家之间自动地交换金融信息（如电子支付、财务分析、咨询文件等），数据可以在基于不同格式的处理程序间来回地交换。同样地，电子出版商只需要向读者提供内容，而读者不再拘泥于专有的格式，可以用自己喜欢的界面和显示方式来进行阅读。

XML 的发展是属于未来的，多元化的服务、友好的界面、丰富的数据共享等。XML 需要强大的新工具用来在文档中显示丰富复杂的 XML 数据，今后我们可以在分层的动态变化的数据上映射用户友好的显示层来实现这一目的。用户将会从 XML 的强大功能中受益很多，比如电子商务，由于大多数商业应用软件包括数据的处理和转移，如订货单、发货清单、顾客信息、合同资料、图纸等等，XML 将会改变用户在 Internet 上的行为，许多商业应用将能实现。另外，通过使用基于 XML 的面向企业内部互连网的词汇库，Web 站点上的信息，无论是储存在文档中还是数据库中，都可以被标识。这些词汇也能够对那些需要在顾客和供应商之间交换信息的中小型企业提供帮助。

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 站点和用户提供了许多有利条件。这些有利条件使更多的纵向市场数据格式建立了起来，并被应用于关键市场，诸如高级的数据库搜索、网上银行、医疗、法律事务、电子商务和其他领域，这使得机会更进一步地扩大。站点将更多地分发数据，而不仅仅是提供数据浏览。

XML 为信息的交换和计算提供新的载体。XML 相对于网络计算的作用，完全可以与计算机起步阶段 ASCII 码的作用相提并论。XML 也可以说是网络信息的标准代码，它表示

的不是符号信息，而是知识化的块状内容。这种标准语言虽然不是程序设计语言，但是它代表的却是下一代网络上相互操作的光明前景。说到这里，很自然地让人想起了人们“当年”对 Java 的狂热。确实，Java 有着非常诱人的初衷，让许多人能够在这样的一种理想的感召下为想象中的各种系统之间的相互操作能力而投入积极的开发中。但是实际上，Java 既没有成为人们想象的成功商业计算工具，也没有实质上的技术进步。Java 试图从统一计算平台的角度来实现相互操作，但是这可能永远都是一个梦想。真正能够相互操作的，只能是标准和通用的可扩展的标识语言——XML。

XML 将给网络进入下一个阶段带来根本性的催化作用。网络站点采用 XML 技术能够实现高度的个性化模式，让用户以自己定义的方式获取有用的信息。

(1) 在商业上，不同的商业系统之间采用相同的词汇进行自动化的电子商务。一家在上海的网络商店能够随时自动更新来自全国几百家商品供应商的产品目录和价格信息，并根据商品类别为用户提供最直观和个性化的“购物环境”。

(2) 在科研教育中，XML 能够描述从前人们只能从图形符号上才能够理解的复杂公式，让科学家和普通的学习者都能够从网络上直接交换具有明确意义的学科知识。如化学标记语言 CML 和数学标记语言 MML，它们提供对传统的科学数据的管理和检索，使得复杂的公式和分子模型等可以在 Web 上传输。由于 XML 是与平台无关的，所以又避免了不同的平台所会引起的二进制格式不兼容的问题。

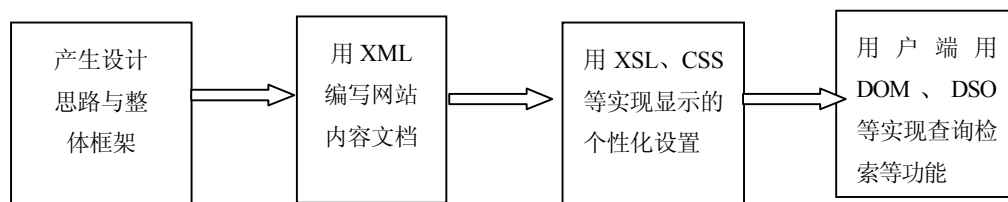
(3) 文学方面，比如所有的古典戏剧可以“翻译”成 XML 文档，用 XML 标记来区分曲牌、台词、旁白、各幕标题等，这样就能够分析检索出某角色的台词等相关信息，为不同需求的读者和演员生成不同的文档和剧本。

(4) 音乐方面，Connection Factory 公司创建了用于乐谱的 XML 应用 MusicML，它包括音符、节拍、五线谱等，可以在 Web 上显示乐谱，并能够将数据在不同的程序中转移，符合了作曲家们的需要。

(5) 出版行业中，目前正在开发“电子书籍”标准。它可以用可交换的描述方式描述今天书籍中的内容和版权信息。今后的“图书”可能都借助一种方便的手持设备在街头“电子书店”中下载最新的图书。延续数个世纪的纸质图书和传统出版业将面临巨大的挑战。

此外，在软件分发、矢量图形绘制、WAP 信息服务的提供等各方面 XML 都有着非常重要的应用，这在后面的章节中我们将会具体地讨论。

下面我们以图例来阐述使用 XML 建设网站的简单流程。



当然，这只是一个最基本和最简单的设计流程，网站开发不会如此简单，否则网站开发技术人员岂不是全都失业了（Just a joke）。我们这里只是用它来说明一下 XML 应用与网站开发的结合而已。

在网站的开发中，XML 文档在浏览器中显示机制的变化，有可能改变许多站点下面的

体系结构。XML 语法本身和 XLink 定义将驱动这些在结构和设计上的不同变化。XML 语法允许产生内容、显示和脚本彼此分离的 Web，这与 HTML 明显不同。XML 趋于把内容与从 HTML 中获取的一些要素分离。层次样式表对许多设计者最有吸引力的方面是它的中心化样式信息，避免了创建和更新 HTML 页面的许多重复性的工作。样式表是图形设计者说明书的自动版本，为了使信息进入文档，提供顺畅的路径，而不用不断地进行人工干预。CSS 对应用程序开发也有一些优点，为那些可以应用到元素的提供格式化结构，而不管操作目标的元素类型。

XML 的出现可能促使许多网站开发者把代码从内容中分离，而 XLink 对站点结构的长期影响可能更大。XLink 使开发者可以用过去越过限制传统文件结构的方法来分解和合并文档。分段和链接给开发者提供考虑他们文档的新方法，可以参考文档的部分，并把它们嵌入到其它文档中创建新文档。分段和链接有潜在的使当前工作的数据库驱动应用程序失去作用的功能，开发者通过重新使用数据，包括它在多文档的数据，而不用编写专用的应用程序。例如，目录可以在一个大文档中存放它的所有信息，除了记录项目，不需要做任何事情。目录的页面可以参考带有 XPointer 的目录重新找回以属性和内容为基础的信息块。作为结果，模板可以链接到所有待售项，所有的体育用品、客户需要的特殊装卡、甚至在目录中所有合理的请求的项目。所有这样的信息可以只用项目所需的部分适当传输。

XML 给 Web 提供真正的潜力，它将成为信息交换的工具，或成为管理文档的工具。虽然现有的一些 XML 应用在 Web 上不太成熟，但是它不会总是如此。XML 是从 HTML 模型中变化产生的，它有一些地方似乎让人感到很熟悉。HTML 开发者现在可以专注于创建他们自己的标记系统和样式表，不需要符合严格单一的规则。服务器开发者可以努力使他们的 Web 看起来象希望的那样，利用 XML、样式表和 Java 允许的灵活性，聚在一起形成新的数据交换标准，使描述、读取、处理和共享数据这些方面发生根本性的变化。

XML 描绘着未来的网络，也描绘着我们未来的生活。XML 给网络上的各种应用创造出更多的新型应用机会，同时改变着我们的生活。XML 继续发展下去，将会很快找到合适的体系结构，使 Web 最终能像它许诺的那样，成为方便、友好、便利和交互式的信息通路。我们有机会面对这种技术的变革，就应该参与到推动这种变革的工作中去，为软件和网络产业增加新的生机与活力。

3.4 XML 在各个领域中的运用

以下是 XML 在各个领域中的运用例子，这里只是罗列了其中的一部分，目的是为了读者更好地理解 XML 的真正作用。

- BSML(Bioinformatic Sequence ML), BSML 是一个提供单一文档接口的 XML 运用程序，它试图使得任何一个项目的信息描述一体化，用于 network/Internet 上的数据交换和获取。有关它的标准可以在以下网址得到：
<http://visualgenomics.com/>
- CDF(Channel Definition Format), CDF 是 Microsoft 开发的一种基于 XML 文件格式的描述频道信息的语言。CDF 是 PUSH 技术的一种引擎。有关它的标准可以在以下

网址得到。

<http://www.microsoft.com/xml/>

- CDIF(CASE Data Interchange Format), CDIF 是一种团体合作时, 例如工具销售者、用户、系统综合人在共享和使用不同版本的工具模型(例如 CASE 工具)时, 如何协调和综合这些工具间的差异所制定的 XML 标准。有关它的标准可以在以下网址得到:
<http://www.cdif.org/>
- GedML(Genealogical Data in XML), GedML 是 XML 运用在系谱编码中的一种 DTD, 它是建立在一种广为传播的系谱文档信息交换对象 GEDCOM 基础上的。有关它的标准可以在以下网址得到:
<http://home.iclWeb.com/icl2/mhkay/gedml.html>
- gXML(Guideline XML), gXML 是一种被 EDI 软件公司支持的文件结构, 用于公开电子商务指导方针地交换。有关它的标准可以在以下网址得到:
<http://www.commercedesk.com/>
- ICE(Information Content and Exchange), ICE 是一种设计成方便推动在 Web 的各个网站之间进行内容管理(例如传统的印刷业)和交换的 XML 运用。有关它的标准可以在以下网址得到:
<http://www.w3.org/TR/1998/NOTE-ice-19981026>
- IMS(Educom Instructional Management Systems Project), IMS 是一种特定的支持在线教育标识元语言, 它起源于广泛协作、会议请求讨论组和与开发研究相关群组。有关它的标准可以在以下网址得到:
<http://www.imsproject.org/>
- JSML(Java Speech ML), JSML 是一种运用于 Java 语音 API 系统的 XML 标识语言, 它用文本形式来注释语音。有关它的标准可以在以下网址得到:
<http://java.sun.com/products/java-media/speech/forDevelopers/JSML/>
- MathML(Mathematical Markup Language), 是一种在 Web 上描述数学符号的 XML 方法, 类似与使用 HTML 描述普通文本。有关它的标准可以在以下网址得到:
<http://www.w3.org/TR/1998/PR-math-19980224>
- OSD(Open Software Description Format), OSD 是由 Microsoft 和 Marimba 开发的一种基于 XML 的软件自动发布系统。OSD 使用唯一的 XML 标志来描述软件包。有关它的标准可以在以下网址得到:
<http://www.w3.org/TR/NOTE-OSD.html>
- PGML(Precision Graphics ML), PGML(紧密图形标识语言)是一种用来描述 2D 平面简单的矢量图和用户自己需要的精度图形语言。PGML 使用 PostScript 语言和便携文件格式(Portable Document Format, PDF)。它同样也提供满足 Web 运用程序的特征。有关它的标准可以在以下网址得到:
<http://www.w3.org/TR/1998/NOTE-PGML>
- PIF-XML(Process Interchange Format), PIF-XML 是一种描述处理指令格式的 XML 语言。有关它的标准可以在以下网址得到:

<http://www.xmls.com/pif/>

- SMIL(Synchronized Multimedia Integration Language) , SMIL 是一种用来实现同步表达多媒体对象的语言。有关它的标准可以在以下网址得到:
<http://www.w3.org/Press/1998/SMIL-REC>
- TMX(Translation Memory eXchange) , 设计 TMX 格式的目的是为了提供一种标准的方法, 来描述转化储存器中的数据, 这些数据一般都是使用在工具之间的信息交换。有关它的标准可以在以下网址得到:
<http://www.lisa.org/>
- VoXML, VoXML 是一种用来在 Web 运用程序中传输和交换与声音有关的接口 XML 语言。有关它的标准可以在以下网址得到:
<http://www.voxml.com/>
- WebDAV(Distributed Authoring and Versioning), WebDAV 是一个用来提高公共远程编辑操作的 XML 标准草案。它提供了一套机制来避免佣金上的纠纷, 增强了非 HTML 数据类型之间的连接管理, 提供了简单属性值元数据的工具, 提供了建立和读取数据的容器等等。有关它的标准可以在以下网址得到:
<http://www.ics.uci.edu/~ejw/authoring/>
- WIDL(Web Interface Definition Language), WIDL 是一种元数据 XML 语法, 用来实现定义运用程序编写中 Web 数据和服务的接口。有关它的标准可以在以下网址得到:
<http://www.w3.org/Submission/1997/15/Overview.html>
- XBEL(XML Bookmark Exchange Language) , XBEL 是一种 Internet “书签” 交换格式。它是 XML 在群组列表领域的运用。有关它的标准可以在以下网址得到:
<http://www.python.org/topics/xml/dtds>
- Xcatalog, Xcatalog 是一种基于 SGML/开放目录上的系统, 用来把 XML 公众标识符翻译成 XML 系统表示符(统一的资源标识符)。有关它的标准可以在以下网址得到:
<http://www.ccil.org/~cowan/XML/XCatalog.html>
- XFRML(Extensible Financial Reporting ML) , XFRML 是设计用来金融团体交换和分析大量的金融报告的 XML 标准, 目前还正在设计中。有关它的标准可以在以下网址得到:
<http://www.oasis-open.org/cover/xfrml.html>
- XHTML(Extensible HyperText ML), XHTML 可以简单地理解为最新的 HTML 版本。有关它的标准可以在以下网址得到:
<http://www.w3.org/TR/xhtml1/>
- XLF(Extensible Log Format), XLF 是基于 XML 的 Web 服务器日志文件格式。有关它的标准可以在以下网址得到:
<http://www.docuverse.com/xf/>
- XMOP(XML Metadata Object Persistence) , XMOP 是设计用来在不同的对象技术间协同合作的 XML 协议, 例如 Java, Microsoft COM 和 CORBA。它意味着对象可以

在不同的对象系统（COM 和 Java）或者不同的 Java VM（Microsoft 的 VM 和 Sun 的 VM）之间传输。有关它的标准可以在以下网址得到：
<http://jabr.ne.mediaone.net/documents/xmop.htm>

3.5 小 结

Web 应用的飞速发展给了 XML 一个广阔的空间，XML 正在成为 Web 和 Internet 的基础，它提供给用户功能更为强大的服务，带给开发人员更为便利的开发环境，同时也为其自身创造了更多的令人惊异的机会。

第四章 XML 工具

“公欲其善事，必先利其器”，进行 XML 的设计、开发和浏览都需要强大的工具来支持。虽然可以用最简单的文本编辑器来编写 XML 文档，但低下的效率和烦琐的规范标记使得 XML 文档对于我们来说不是一种应用新技术的乐趣，因此，相应的 XML 工具就有了“大显身手”的舞台。

本章包括以下内容：

- XML 开发工具
- XML 浏览器

4.1 XML 开发工具

XML 开发工具与日俱增，帮助开发人员在各种各样的场合制作 XML 文件。这里介绍较少几种常见的开发工具。主要是编辑工具和分析工具，读者可以访问以下网址以了解最新工具的发展动态：[HTTP://WWW.XML.COM](http://www.xml.com)。该网站是由 Seybold Publications 公司设立，主要是密切跟踪业界最新动态，包括最新工具的信息。

4.1.1 编辑器

编辑器允许输入标记并把文件保存到 XML 文件，由于功能的不同编辑器分为简单的和复杂的两种。有的像记事本一样简单，有的功能复杂，因为它能够针对 DTD 确认 XML 文档的有效性。

1. HomeSite 4

在 HTML 开发圈内，Allaire 公司的 HomeSite 是一个较为成熟的编辑器。HomeSite 具有很多独特的特征，具有强大的搜索和替换能力，同时具有句法突出、CSS 编辑组件、实况预览等许多功能。

2. Xmetal 1

Xmetal 1 是市面上最全面的编辑器，也是一流的编辑器之一。其原因是该程序的功能相当完备，支持文档确认和 DOM。其特征包括：

- (1) 针对 DTD 确认 XML 文档。
- (2) 拥有一套规则设置开发工具，可以协助用户利用各种规则创建各种 DTD，它们以 Xmetal 专有的格式存储，与用户要确认的 XML 文档匹配。
- (3) 支持 DOM，它包括 W3C 正式推荐的规范中包含的每个 DOM 界面的全面的资料。
- (4) 支持使用 JavaScript 和 VBScript 编写脚本。
- (5) 支持 COM。

3. XML Notepad

XML Notepad 是微软的一个简单的文本编辑器，它可针对在 IE4 中运行的 XML DOM 确认 XML 文档。该对象模型与 W3C 定义的 DOM 不同。IE4 运行在这样一个模型上：它的基础是在 DOM，但却对该模型做了大量的改动。

4. XML Authority 1

XML Authority 1 特别擅长模式和 DTD 的开发。如果你已经开始制作 XML 文档，或者有许多希望转成 XML 文档但又没有 DTD 或模式从事此事的遗留文件，则可以使用该编辑器提供一个极为有用的工具。

5. Stylus

Transformis 的 Stylus 是一个 XSL 样式表编辑器，提供结果数的实时预览。在学习如何开发 XSL 文档时，看见结果树的能力是很重要的。该编辑器是一个早期版本，但很稳定。

4.1.2 分析器

分析器是专门分析 XML 文档的程序，经常被合并到 Mozilla 之类的更大程序中。它们通常作为独立的程序使用，在这些情形中，往往运行于命令行界面，因为设计它们时，XML 还处在发展初期，变数较多。对于程序员来说，这些是很有用的资源，因为它们常常提供对类库的存取，这时用户可以围绕其设计自己的软件。

1. Expat

Expat 分析器以前称为 xmltok，它使用 C 语言编写的分析引擎，用来运行 Mozilla XML 分析器。它是 James Clark 对 XML 界所做的诸多贡献之一。程序可在 <http://www.jclark.com/xml/expat.html> 网址上下载。

2. XP 和 XT

XP 是一个基于 Java 的 XML 分析器，XT 是一个 XSL 处理器。二者都是用 Java 语言编写的，都能用在 Java 程序中，应用范围很广。

3. SAX (Simple API for XML)

它是一个采用基于事件的方法进行 XML 分析的 API。W3C 的 DOM 也是能用于 XML 的 API，但它以 XML 文档数为基础，而 SAX 则以事件为基础。也就是说，在 XML 文档上，它不是由节点到节点地报告，而是从事件到事件（文档的开始到元素标注的开始到文本字符）。许多 XML 分析器都用它的界面作为自己分析例程的一部分。可以在 <http://www.megginson.com/SAX/> 网址下载。从技术上看，SAX 本身并不是一个分析器，而是一个可供其它欲构建分析器的人在其分析软件中使用的 API。

4. 其它分析器

在因特网上还有其他几个分析器。其中一些在 XML 标准还只是 W3C 工作草案的时候就已经开发出来了，还有一些则仍未定型。这些分析器的名字及其地址如下：

- SAX for Python，来自 Lars Marius Garshol (<http://www.stud.ifi.uio.no/~larsga/>)

download/python/xml/saxlib.html)，这是 SAX 的 Python 版。

- SXP, 来自 Silfide (<http://www.loria.fr/projets/Xsilfide/EN/sxp/>)，一个客户机/服务器 XML 分布系统。
- Projet X, 来自太阳微系统公司，它是一个符合 XML 1 和 SAX 并支持 DOM (规范 1) 的基于 Java 的分析器 (<http://developer.java.sun.com/developer/products/xml/>)。
- XML Parser for Java, 来自 Oracle 公司 (<http://technet.oracle.com/>)。
- Lark, 来自 Tim Bray (<http://www.texruality.com/lark/>)。
- XML Parser, 作者 Dan Connolly (<http://www.w3.org/XML/9704/hacking>)，这是一个早期的 XML 分析程序。
- XTL, 来自 Vivid Creations (<http://www.vivid-creations.com>)，是一个处理 XML 的 COM 对象。
- SAXON, 来自 Michael Kay (<http://home.iclWeb.com/icl2/mhkay/saxon.html>)，具有多功能界面，包括支持 XSL 和 XSL 的某些扩展。

4.2 XML 浏览器

4.2.1 InDelv 浏览器

InDelv 公司的 InDelv 浏览器是用 Java 编写的，它的唯一用途是浏览已经转换为 XSL 格式化对象的 XML 文档。该浏览器的早期测试版不支持 DOM 或脚本设计，支持 X-Link，它的大多数例子都是以 X-Link 联接机制和父样式表为基础。可以在 <http://www.indelv.com> 网址下载 InDelv。

4.2.2 Internet Explorer

IE5.0 广泛支持 XML，但并不完善。尽管还不完美，但它是一个了解 XML 的免费工具，支持下述 XML 实现：

- 利用缺省 XSL 样式表直接浏览 XML。
- 支持 XSL 的早期工作草案。
- 支持模式和扩展数据类型。
- 支持基于 XML 的客户端数据联接。

4.2.3 Netscape Navigator

Netscape Navigator 4.0 不支持 XML，但是下一代将全面支持 XML。Mozilla 代码将是下一个 Netscape Communicator 的核心 API。其体系结构在下列几个方面将与 XML 有关：

- RDF，它提供对遍及 Internet 的数据描述的访问。
- James Clark 的 XML 分析引擎，任何 Netscape 模块都能访问它。
- W3C 为 XML 定义的 DOM，任何插件和 JavaScript 都能访问它。
- XUL (XML-based User Interface Language) 是基于 XML 的标记语言，使用 HTML4.0 和 CSS 描述用户界面。

- 大量的 JavaScript 核心文件，与 DOM 和 XUL 文件一起完成用户界面。

4.3 小 结

各种 XML 工具的使用将使我们充分享受到开发和应用 XML 的乐趣,同时,众多的 XML 工具大大地简化了我们在文档编写、分析、浏览等过程中的工作,这样我们就可以把全部的精神投入到文档结构设计、个性化显示等与实际开发联系更为密切的部分中去。