XML 原理及应用

袁　俊　王增武　廖德钦　编著

电子科技大学出版社

内 容 简 介

本书在讨论 XML 基本原理和结构的基础上，运用实例描述了 DTD、XML Schema、NameSpac、CSS、XSL、Xpath、XLink、Xpointer 等相关协议内容及其规范的使用方法；阐明 DOM 和 SAX 两类编程接口以及 XML 数据库访问技术。通览全书，读者对 XML 的综合应用会有较全面的认识与把握。本书内容安排紧凑、循序渐进，讨论风格深入浅出，既是大学计算机及其应用、电子商务、网络营销等专业网页设计的教材，又是信息产业和 IT 技术中蓝领和白领的必备读物。

图书在版编目（CIP）数据

XML 原理及应用 / 袁俊，王增武，廖德钦编著. —成都：电子科技大学出版社，2004. 8
ISBN 7 – 81094 – 625 – 0

Ⅰ. X… Ⅱ. ①袁… ②王… ③廖… Ⅲ. 可扩充语言，XML－程序设计－高等学校－教材 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2004）第 088914 号

XML 原理及应用

袁 俊 王增武 廖德钦 编著

出 版： 电子科技大学出版社(成都建设北路二段四号 邮编：610054)
责任编辑： 谢应成
发 行： 新华书店
印 刷： 四川省地矿局测绘队印刷厂
开 本： 787×1092 1/16 印张 21 字数 511 千字
版 次： 2004 年 8 月第一版
印 次： 2004 年 8 月第一次印刷
书 号： ISBN 7 – 81094 – 625 – 0/TP・372
定 价： 32.00 元

前　言

《XML 原理及应用》是作者跟踪 IT 技术的发展前沿而潜心研究的重要成果。它将帮助运用 HTML 和工具软件的人们，成功地掌握 XML 文档的设计与应用，开发与编写网页；帮助人们正确地掌握和运用 XML 的技术。

21 世纪初，基于 XML 的应用已经深入到网络、无线技术、软件开发与数据库等多个领域。XML 是 SUN 企业平台的基石，也是微软的 Web 服务和下一代操作系统的基础。目前，所有软件厂商和 IT 企业都宣布对 XML 的支持，XML 已经成为现代信息处理的重要技术。

XML 既是一种跨平台的数据描述语言，又是一种完全可移植的数据格式，同时，它和诸如文档对象模型等相关规范都是跨语言的，因而在实际操作中，程序设计人员不用放弃所熟悉的语言环境，即可开发出新环境下的网络和分布应用。

XML 还跨越发布网页的限制，最有力地影响着商业交互方面的沟通能力。这种沟通既可在 Web 上实现，又可直接以 Internet 为基础。与此同时，XML 的应用已使电子商务、电子数据交换发生着深刻的变化，因为，它可提供通用的语言来传输订单、报价和查询各种商业文档；按照特定的商业术语，使远程计算机之间直接交换文档并相互理解其文档含义；它正在使 Web 成为发展世界经济和贸易的平台和枢纽。

XML 是纯文本的，可阅读的，并且实现简单，既可以自由扩展，不限制任何自然语言对标记进行定义，又可用最基本的文本编辑软件和维护文档，得到廉价或免费的 XML 的处理工具。

飞速发展的信息技术的事实愈来愈生动地证明，XML 的种种特性，已使它正在成为网络应用、电子商务、网页设计、项目开发、软件设计、数据库管理、多媒体应用、电子文档等多领域不可或缺的基本技术。为提升学习质量，提高工作效率，驰骋在网络经济无垠的大舞台上，我们应该成功地掌握 XML 的原理及应用，因为它是生活在网络时代里的人们所必备的重要技能。

黄宗捷

甲申年仲夏于网络商学院

目　录

第一章 XML 基础

本章要点：

- XML 基础知识
- HTML 的基本标记及应用
- XML 的应用

1.1 XML 的发展历史

XML 是可扩展标记语言(EXtensible Markup Language)的简称，它的先驱是 SGML 和 HTML，它们都是很成功的标记语言，但是在某些方面，这两种语言也存在着自身的一些缺陷，XML 正是为了解决它们的不足而产生。

XML 通过允许编程者重用称为 XML 处理器(Processor)或解析器(Parser)的代码片段而节约开支。

理解 XML 核心思想的最简单方法就是追溯其源头——SGML(标准通用标记语言)。

1.1.1 SGML 语言

SGML 是在深厚的文本处理系统背景上发展起来的。文本处理是计算机科学的一个分支学科，旨在创建可以自动化部分文档生成和发布流程的计算机系统。文本处理软件包括简单的文字处理、高级新闻项目数据库、超文本文档表示系统以及其他发布工具。

自动化文本处理的第一次浪潮是计算机排版。用户只需键入文档并描述其格式化方式。计算机就会根据所描述的文本和格式输出文档。

我们把包含实际文档数据及所需格式描述等混合内容的文件格式称为重现(Rendition)。一些发展史上重要的重现表示法包括 troff、RTF(Rich Text Format)和 LaTeX。

排版系统可以把这种重现转换成人类可以理解的某种形式——即显示(Presentation)。传统的显示介质是纸，但最终会是电子显示器。

排版系统加速了文档发布过程，并发展成为目前所谓的桌面发布系统。Microsoft Word 和 Adobe PageMaker 之类的新型程序仍然处理重现，但它们向用户提供操作重现的更友好界面重现（其中带有格式化代码的文件）的用户界面设计得看上去类似于显示（纸质成品）。我们把这种发布方式称为 WYSIWYG（What You See Is What You Get——所见即所得）。由于重现仅仅是描述表示，因此用户界面理应反映最终产品。

1．格式标记

所见即所得之前的排版表示法称为格式标记（Formatting Markup）。下面类比说明：假

定作者向排版员提交了一份手稿以备出版，如果该手稿没有任何格式，甚至没有段落和字体区分，而只是单个“ 标记 ”和如何排版的书面指令。作者可以书写非常准确的排版指令说明：“ 把这个单词移动两英寸，用粗体。移动位于其旁边的下一个单词。移动位于其下面的下一个单词，用粗体。从这里另起一行。” 等等。

格式标记与此非常类似。我们采用称为标注（Tag）或代码（Code）的指令“ 环绕 ” 文本（是用标注还是用代码指令根据具体格式标记语言而定）。

2．通用标记

大约在 20 世纪 60 年代后期，人们开始希望从其文档中获取更多东西。特别值得一提的是，IBM 要求一个名叫 Charles Goldfarb 的年轻研究人员构建一个系统，用来存储、查找、管理和发布法律文档。

Goldfarb 发现，IBM 内部的很多系统相互之间无法通信。它们不能交换信息。它们各自使用互不相同的命令语言，彼此无法读取对方的文件，就像我们无法在 Word 中加载 WordPerfect 文件（或相反）一样。与现在的情况类似，当时的问题就在于它们都采用不同的信息表示（有时也称为文件格式）。

20 纪 60 年代后期，Goldfarb 和其他两位 IBM 研究人员 Ed Mosher 及 Ray Lorie 开始着手解决这一问题。这个三人研究小组认识到，这种语言理应需要文档三个基本特征：

（1）公共数据表示：标记。

（2）不同的计算机程序和系统理应能够读写表示相同的信息。

（3）标记应可扩展。

必须交换各种各样的信息类型。这种标记语言必须具有足够的扩展能力以便支持所有信息类型。

文档类型需要规则。

1986 年，国际标准组织（ISO）最终采纳该语言为标准通用语言（SGML）。

SGML 提供一套复杂的系统来对文档进行标记操作，使得文档外观独立于特定的应用软件。SGML 语言庞大、功能强、选项多、使用于需要有严格文档标准的操作。

同时，SGML 也可用于创建更多的标记语言，为其语法提供强大的工具，而且该有很好的扩展性。

20 世纪 80 年代，SGML 较多的用于科技文献和政府的办公文件中。

但是，SGML 强大功能的背后是它的复杂性，所以不适合在 Web 中快速简便的发布。

1.1.2 HTML 语言

HTML（超文本标记语言）是一种的特殊的 SGML 文档类型。HTML 最初于 1990 年由 CERN 设计，它是非常简单的 SGML 语言，由于它的学习和实现非常容易，而且还免费提供源代码，因此，HTML 很快得到 Web 浏览器厂商的支持。后来，W3C（万维网联盟）承担了 HTML 的开发和标准化任务，从 1993 年的 HTML1.0 标准到现在的 HTML4.0 标准，经历着不断的完善和改进。

但是，HTML 最大的缺点就是过于简单。在 Web 发展的早期，所有的文档都是基于文本格式的，只有标题、项目列表和指向其他文档的超链接等。随着应用的发展，有了对多媒体技术和页面设计的要求，这就开始了 HTML 的痛苦历程。随着后来出现的闪烁文本、表

格、帧等技术的出现，HTML 越来越不能应付了。

多年来，微软公司添加了很多适用于 IE 的标记，而网景公司则添加了只用于 Navigator 的标记，然而作为网页创作者的用户则无法添加属于自己的标记，这一切都归咎于 HTML 的不可扩展性。

1.1.3 XML 语言

1996 年，人们开始致力于研究一种新的标记语言，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。为此，W3C 专门成立了一个专家小组来负责此项工作，并由 Sun 公司的 Jon Bosak 担任小组的指挥。

该小组将 SGML 中所有非核心的、未被使用的和含义模糊的部分删除掉，剩下的就是短小精悍的可扩展标记语言 XML。XML 保留了 SGML 语言 80%的功能，而其复杂程度则降低到原来的 20%。

1997 年春，XML 草案拟订，后几经修改、完善，W3C 于 1998 年 2 月 10 日批准 XML1.0 版本，从此一个全新的、大有前途的标记语言诞生了。

1998 年，在 XML 协调小组的指导下，XML 的设计工作分成 5 个新的工作组：XML Schema 工作组、XML Fragment 工作组、XML Linking 工作组（XLink 和 XPointer）、XML 信息集（Information Set）工作组和 XML 语法工作组。这 5 个工作组与 W3C 的 XSL 和 DOM 工作组一起，为 XML 技术的发展和完善竭尽所能。

1.1.4 XML 的发展前景

XML 的发展是属于未来的，先前所做的努力只是为将来它能更好地为我们服务。网络的未来属于商业，而那时的商业操作已经不再只是简单的网页广告，或是有限信息的反馈和狭窄的交易空间，取而代之的是多元化的服务、友好的界面、丰富的数据共享。XML 需要更加强大的新工具的支持，以便在文档中使用丰富的、复杂的数据。

XML 作为表示结构化数据的一个工业标准，为组织者、软件开发者、Web 站点和终端使用者提供许多有利的条件。这些条件使更多的纵向市场的数据格式建立起来，并被利用到关键市场，诸如高级的数据库搜索、网上银行、医疗、法律事业、电子商务和其他领域，这使得机会更进一步的扩大。当站点更多的进行分发数据，而不仅仅是提供数据浏览时，特别的机会就产生了。另外一个有待开发的市场是建立 Web 站点的工具，包括用来从数据库信息和使用者界面中产生 XML 数据的工具。

1.2 XML 的相关技术

XML 并不是在真空中操作的，如果将 XML 应用于多种数据格式的操作，就需要与相关的技术进行合作，其中包括 HTML、CSS、URL、XLink、XPath、XPointer、Unicode 字集等。

1.2.1 元素：逻辑结构

描述究竟如何表示文档之前，我们的脑海里必须拥有一个文档如何构成的模型。大多数

文档可以分解成组件（章节和文章）。这些组件又可以分解成组件（标题、段落、图形等）。依次分解，直到分解成文本数据本身——单词和句子。

结果，每个文档都可以这样看待，尽管对此模型的适宜性各自有别。实际上，可以这样看待所有信息，附加说明也一样。

在 XML 中，这些组件称为元素（Element）。每个元素表示一种文档逻辑组件。元素既可以包含其他元素，也可以包含通常会当作文档文本的词和句子。XML 把此文本称为文档的字符数据(Character Data)。

文档的层次结构如图 1-1 所示。

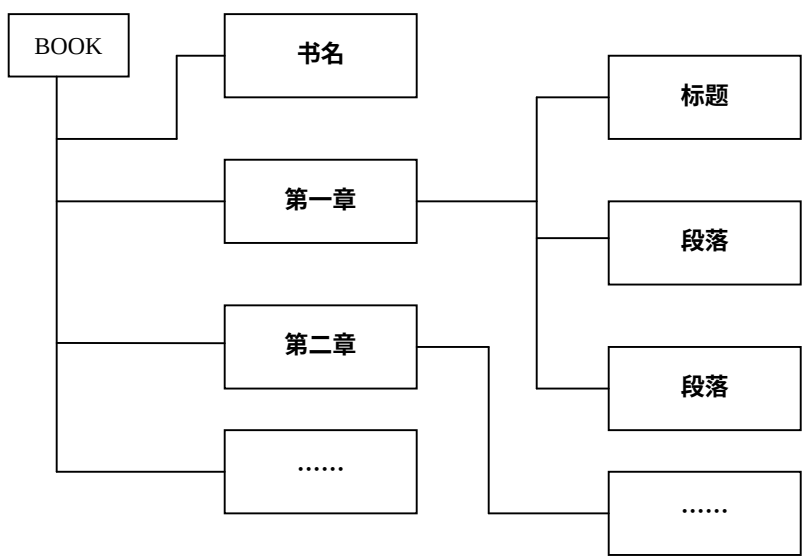

图 1-1　文档的层次结构图

标记语言专业人员将该视图称为文档的树结构（Tree Structure）。包含所有其他元素（如 BOOK）的元素叫做根元素(Root Element)。根元素也叫文档元素（Document Element），因为它在其内部拥有整个逻辑文档。

根元素中包含的元素称为其子元素。它们可以包含自己的子元素。如果子元素又包含子元素，我们将把它们称为枝（Branch），如果不包含子元素，则称为叶（Leave）。

元素还可以附带称为属性的附加信息。属性描述元素特征。

1.2.2　标记

XML 文本是字符数据与标记组合的统称，并不是指单个字符数据。字符数据+标记=文本。标记通过名叫分隔符（Delimiter）的专用字符与字符数据相区别。不太严格地讲，小于号（<）与大于号（>）之间的文本或与号（&）和分号（;）之间的文本就是标记。这四个符号是最常用的分隔符。

例题　标记的使用

```
<?xml version="1.0" encoding="gb2312"?>
<!DOCTYPE Q-AND-A SYSTEM "http://www.q.and.a.com/faq.dtd">
```

```
<Q-AND-A>
<QUESTION>I'm have trouble loading a ?</QUESTION>
<ANSWER>Why don't you use XML?</ANSWER>
<QUESTION>What's XML?</QUESTION>
<ANSWER>It's a long story,but there is a book I can … </ANSWER>
</Q-AND-A>
```

以小于号开始、大于号结束的标记称为标注（Tag）。

1.2.3 级联样式单（CSS）

级联样式单（Cascading Style Sheets，CSS）最初是为 HTML 服务的，它可以定义字号、字体、段落缩进、段落对齐、列表格式、表格和表项格式、超级链接样式等属性，这些属性可以施加到个别元素上。如 CSS 允许 HTML 文档来指定所有的 H1（一级标题）元素格式化为隶书，32 磅，红色（h1{font-family:"隶书"; color:#FF0000; size:32px}）。

单独的样式可以施加到大多数 HTML 标记上，它能够覆盖浏览器的缺省设置，多个样式单可以施加到一个文档上，也可以施加到单个元素上，因为样式是根据特定的一套规则级联起来的。

向 XML 施加 CSS 规则也很容易，只要改变施加规则于其上的标记名就可以了。Mozilla 5.0 直接支持 CSS 样式单与 XML 的结合，但此时浏览器时常会发生崩溃。

1.2.4 可扩展的样式语言（XSL）

XSL（Extensible Style Language，XSL）可扩展的样式语言是更先进的专门用于 XML 文档的样式单语言，它本身就是结构完整的 XML 文档。

XSL 文档包括一系列的适用于特定 XML 元素样式的规则，它读取 XML 文档并将其读入的内容与样式单中的模式进行比较，当在 XML 文档中识别出 XSL 样式单中的模式时，对应的规则将输出某些文本的组合。

与 CSS 不同，输出的文本比较任意，也不局限于输入的文本加上格式化信息。CSS 只能改变特定元素的格式，是以元素为基础的。而 XSL 可以重新排列元素并对元素进行重新排序。XSL 可以隐藏一些元素而显示另外一些元素。

CSS 的优越性在于具有广泛的浏览器的支持，但 XSL 更为灵活和强大，而且 XSL 样式单的 XML 文档可以很容易的转换成 CSS 样式单的 HTML 文档。

1.2.5 URL 和 URI

Web 上地址的基本形式是 URI，它代表统一资源标识符。有两种形式：

（1）URL 目前 URI 的最普遍的形式就是无处不在的 URL 或“统一资源定位器”。其基本形式是：协议名：//主机名/数据路径。如 http://jwc.cuit.edu.cn/JXGL/Default. asp?。

（2）URN URL 的一种更新形式，统一资源名称（Uniform Resource Name）不依赖于位置，并且有可能减少失效链接的个数。

虽然 URL 得到人们的广泛理解，但 XML 规范使用的则是更为通用的 URN。理论上说 URN 可以找出镜像文档的最近的文档，或是找出已经从一个站点移动到另一个站点的文档。

目前 URN 还处在进一步的研究之中。

1.2.6 XLink 和 XPointer

在 Web 页面上，用户都希望能够从一个地方出发，将多个文档连接起来。在 HTML 中使用超链接可以完成这些任务，如：成都信息工程学院，将这个超链接代码放在 HTML 文档内的，在进行页面浏览的时候，我们就可以单击“成都信息工程学院”，快速地进入到学院的首页页面。

在 XML 中，利用 XLink 来与文档相链接，用 XPointer 来确定文档个别部分的位置。XLink 可以是任意元素成为链接，而不是一个元素。也就是说 XLink 可以使得链接是双向的、多向的或者是指向多个镜像站点，但是在选择的时候，系统自动的选择最近的一个。XLink 利用普通的 URL 来表示它链接的站点。

XPointer 能使链接不仅指向特定的位置，而且还可以指向特定文档的特定部分。它可以引用文档中特定的元素，如第一个、第三个、第十个等特定的元素。XPointer 提供非常强大的文档之间的链接功能，而文档不必有包括附加标记的目的文档。

XPointer 与 HTML 中的锚点（anchor）类似，但它们也有很大的不同。XPointer 不只是引用文档中的一个点，同时可以指向一个范围或一个区域，因而 XPointer 可以用来选择文档的特定部分。

1.2.7 Unicode 字符集

文本由字符构成，如果要表示文本，则必须表示组成文本的字符。因此，必须确定位和字节层次的字符表示方式，这叫做字符编码（Character Encoding），同时还必须确定文档中可以使用的字符，这就是字符集（Character Set）。

通行的 7 位 ASCII 字符集合只有 26 个大写字母、26 个小写字母、10 个数字、一些标点符号和其他一些符号共 128 个符号。ASCII 既是字符集，又是字符编码方式，它定义可用的字符集，以及如何用位和字节进行字符编码。

XML 使用的字符集是 Unicode，一种增强型的 ASCII。Unicode 字符集囊括了源于世界各地的数千个字符。不过 Unicode 的前 128 个字符与 ASCII 一致，而且还有 Unicode 的字符编码方式，即与 7 位 ASCII 一致的 UTF-8 编码。每个 ASCII 文档可以自动转换成 Unicode 文档，因此，我们就可以用标准文本编辑器来创建 XML 文档。

1.2.8 名域空间

允许任何人都可以像 XML 那样选择名称时就会出现一个问题。即，不同地方的不同人会在不经意间选择相同名称。这使得处理来自多个独立来源的文档的系统非常难以构建，因为某发布者可以使用元素类型名称 PAK 指示段落（Paragraph），而某军事词汇表可以使用名称相同的元素类型指示空降兵（Paratroper），某数学家则可能使用该名称的元素类型标记反论（Paradox）。

如前所述，词汇表就是一个元素类型集合。XML 没有可以允许计算机解决这种同音异义问题的正式“词汇表”构造。

因此，即使使用全名作为元素类型也无法避免名称冲突。在两种不同的电子商务框架中，

术语 Purchase_order 所指内容可能差别不大。对于计算机程序而言，这种轻微误解与更严重的错误一样危险。如果 XML 系统误译了某个元素类型，则它将会启动错误的处理，指出错误元素处在特定位置，否则只会乱成一团。

然而，混用且匹配不同组织在不同地方创建的元素类型变得越来越必要：如果需要在技术报告中包含数学公式，可以使用 MathML 词汇表时，重新创建元素类型毫无意义。问题在于，如何才能区分自己文档中的哪些名称来自 MathML？

存在一个处理此问题的标准。该标准称为“XML 名域空间（Namespaces）”。名或空间就是这样的区域：在此区域之中，名称在其使用时具有相同含义。词汇表是名域空间，因而也是文档。通过创建可以前缀到元素类型名称的词汇表诨名，此标准允许读者在文档中混用词汇表。

例如，读者可能会选择使用“met：”加在源自气象信息词汇表的名称之前。因此，某文档可能包含“met：temperature，met：humidity”之类的元素。它也可以包含源自另一个不同词汇表的“health：temperature”。对于计算机而言，“met：temperature”和“health：temperature”显然属于不同的名称。

当然，这种解决办法也会产生自己的问题：如何才能确信不同 XML 设计者会使用 met 指示同一词汇表？简短且中肯的回答是：不必如此。

原因是，在此文档之内，met（一个诨名而已）与此词汇表的一个明确标识符相关。此标识符基于一个 URL 之上。其他文档可以使用不同前缀指示此词汇表，或者使用“met”前缀到不同词汇表。虽然，实际上词汇表开发者推荐前缀，而人们也倾向于使用推荐前缀。

1.2.9 XML 路径语言(XPath)

XPath 于 1999 年 11 月 16 日和 XSLT 一起成为正式标准，用做 XSLT 和 XPointer 对 XML 文档各部分间进行定位的语言。它提供 XSLT 和 XPointer 一个共同的、整合的定位语法，用来定位 XML 文档中各个部分和选择文档中的构成部件（元素、属性、内容等）。此外，还提供一些函数，如基本的数字运算函数、布尔运算函数、字符串处理函数等。

XPath 利用一个紧凑的、非 XML 的语法用来实现在 URI 和 XML 属性值中使用，它基于 XML 文档的逻辑结构，并在结构中进行导航。除了用于定位，XPath 自身还有一个子集能够用于匹配，即能验证一个节点是否匹配某个模式，并且用表格列出了各种匹配的方式。

XPath 把一个 XML 文档看成一个树或节点的模型。节点的类型有多种，包括元素节点、属性节点和文本节点。它定义计算每一个类型节点的字串值方式，并完全支持 XML 名域。

XPath 的核心语法构架是表达式，表达式经过计算后的值是一个对象，对象有四种基本类型：

（1）节点集合

（2）布尔类型

（3）数字类型

（4）字符串类型

XSLT 和 XPointer 分别规定 XPath 的表达式将在什么情况下出现，这些上下文的关系包括：

（1）节点。

（2）一对正整数，表明位置个大小。

（3）一套变量绑定（binding）集合。变量绑定是指从标量名称到变量值的映射。

（4）函数库。函数库中的每个函数都有 0 个或多个参数，并返回一个结果。

（5）规定表达式范围的名域声明。

XPath 表达式经常出现在 XML 属性中，而且在这里它遵循 XML1.0 的规范。

1.2.10 定义文档类型 DTD

元素类型集合称为词汇表（Vacobulary），每一种文档类型当然它有一个词汇表，或者可能不止一个。

文档类型定义（Document Type Definition，DTD）或模式定义（Schema Definition）包括元素类型、属性、实体及表示法的一系列定义。

DTD 或模式定义声明哪些元素类型在文档内部是合法的，以及它们在何种地方是合法的。文档可以在其文档类型声明（Document Type Declaration，通常缩写成 DOCTYPE）中声明符合某特定 DTD。还存在一种通过专用属性访问模式的机制。

DTD 和模式是组织标准化的强大工具，正如表单、模板和样式向导一样。极严格的 DTD 在特定位置只允许一种元素类型。更灵活的 DTD 类似于样式向导。

DTD 不仅对于组织标准化极其重要，而且对于软件可靠处理文档也同样重要。

通过全权负责输入有效性的检查，DTD 和模式简化了软件的建构过程。这可以与 XML 处理器担起解析之责相比拟。

DTD 和模式也充任信息创建者与使用者之间的某种协议。一旦拥有模式或 DTD，就可以将其用做系统之内文档有效性的极其正式且客观的定义。

下面例中展示了 DTD 的声明：

例题 DTD 的声明

```
<!element Q-AND-A(QUESTION,ANSWER)>
<!--This allows:question,answer,question,answer…-->
<!element QUESTION(#PCDATA)>
<!--QUESTION ARE JUST MADE UP OF TEXTUAL DATA-->
<!element ANSWER(#PCDATA)>
<!--ANSWER ARE JUST MADE  UP OF TEXTUAL DATA-->
```

文档类型或模式是一种概念。文档类型定义（DTD）或模式定义是此概念的表达式。

有些 XML 文档没有模式定义或文档类型声明。这并不是指它们不符合某种文档类型，而只表明，它们没有声明符合某种正式定义的文档类型定义。

如果准备用于 XML 文档，文档还必须拥有某种通过元素、属性等表示的结构。为文档创建样式单时，需要根据此结构拥有特定元素，根据元素类型名称了解特定含义，并根据元素而出现在特定位置。

1.2.11 XML Schema

Schema 标准的发展经历了较长的的过程。早在 XML1.0 标准尚未确定之前，W3C 就已