

IBM/Lotus
技术丛书

WebSphere

环境下

XML与XSL编程

IBM红皮书

(美) Luis Ennsen 等著 瞿裕忠 张剑锋 陈 峥 王从刚 等译

The XML Files: Using XML and XSL
with IBM WebSphere 3.0



机械工业出版社
China Machine Press

IBM

IBM/Lotus技术丛书

WebSphere环境下 XML与XSL编程

IBM红皮书

(美) Luis Ennser 等著

瞿裕忠 张剑锋 陈峥 王丛刚 等译



机械工业出版社
China Machine Press

XML是SGML的面向Internet应用的一个优化子集，它属于独立系统和处理程序。本书全面地介绍了XML的技术，然后阐述了在IBM WebSphere上应用XML的实用技术，最后还展示了使用上述技术开发的应用示例。全书图文并茂，条理清晰，对于项目经理、系统设计和开发人员都是一本极具参考价值的好书。

本书简体字中文版由国际商业机器中国有限公司授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书的任何部分。

版权所有，侵权必究。

图书在版编目（CIP）数据

WebSphere环境下XML与XSL编程/（美）恩索（Ennsner, L.）等著；瞿裕忠等译. —北京：机械工业出版社，2001.5

（IBM/Lotus技术丛书）

书名原文：The XML Files: Using XML and XSL With IBM WebSphere 3.0

ISBN 7-111-08810-7

I. W… II. ①恩… ②瞿… 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（2001）第11020号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：张鸿斌 曹承志

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2001年5月第1版第1次印刷

787mm×1092mm 1/16·10.75印张

印数：0 001～5 000册

定价：36.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

JSB 11/04

序

IBM，即国际商业机器公司，1914年创建于美国，是世界上最大的信息工业跨国公司，目前拥有全球雇员20多万人，业务遍及150多个国家和地区。

以世界一流的最新技术开发新产品，并以最快的生产速度进入市场是IBM的产品发展战略。IBM拥有综合先进技术与结构的全系列产品，包括新一代基于CMOS的并行企业服务器、首次采用64位RISC技术的AS/400高级系列、基于高性能PowerPC604微处理器的新RS/6000系列以及最具领导地位的电子商务平台WebSphere和优秀的数据库产品DB2 UDB等，在复杂的网络管理、系统管理、密集型事务处理、庞大数据库、强大的可伸缩服务器、系统集成等方面IBM都具有强大的优势。

随着中国改革开放的不断深入，IBM在华业务日益扩大。目前，IBM的各类信息系统已服务于中国金融、冶金、石化、交通、制造、商品流通等许多重要业务领域。

为了推动IBM技术在国内应用水平的提高，IBM中国公司授权国内IT专业图书出版公司华章公司组织翻译出版一批IBM核心技术资料，期望这些图书的出版能为您的工作、学习与研究有所裨益。

IBM中国有限公司软件部

2001.1

译者序

XML是SGML的面向Internet应用的一个优化子集，它属于独立系统和处理程序，这种独立性已经被十几年的SGML应用经验所证实。XML把这种能力扩展到Internet上。因为它的简单性、开放性、可扩展性、灵活性、自描述性等特性，XML在数据和信息管理、数据交换、Web应用、电子商务、应用集成等诸多领域有着重要用途。正因为如此，XML很快地得到了工业界的普遍支持。

本书（原名：The XML Files: Using XML and XSL with IBM WebSphere 3.0）展示了在B2B和B2C环境中应用XML技术的多种不同途径，特别是在IBM WebSphere环境下应用XML的实用技术。全书共11章，前3章介绍XML技术，中间4章阐述在IBM WebSphere上应用XML的实用技术，最后4章展示一个ITSO XML应用示例。书中给出了应用示例的源代码和安装步骤。我们衷心期望这本书能够有益于我国XML及应用系统的研究和开发。

本书第1、4、7章由瞿裕忠翻译，第2、3章由张剑峰翻译，第5、6章由王丛刚翻译，第8、9、10、11章由陈峥翻译，王继承、李文龙、贺晓敏、高红梅、曹承文参加了译文修改工作，全书由瞿裕忠审稿。有关XML技术及其在电子商务中应用的技术资料和应用示例，请访问东南大学电子商务论坛（<http://cse.seu.edu.cn/ebiz/>）。

在翻译过程中，译者参阅了大量的国内外资料。在此，谨向书中提到和参考文献列出的作者表示感谢。本书的翻译工作得到了机械工业出版社和IBM中国有限公司的支持，也得到了东南大学XObjects Group其他成员的大力支持，在此，我们深表谢意！

限于译者的水平及时间，本书尚有许多不足之处，恳请读者指正。

瞿裕忠

2000年12月于东南大学

前 言

可扩展标记语言(XML)非常快速地得到了行业的大力支持。因此,很多有关XML的会议、书籍、Web站点和培训课程如雨后春笋般涌现出来。一大批新的XML开发工具不久将面市。

在本书中,我们试图展示XML技术在B2B和B2C环境中应用的多种不同途径。

本书专为那些对于使用XML及相关技术(如XSL和XSLT)设计与开发Web应用感兴趣的读者而编写。项目经理、系统设计师和开发人员将发现这本书对他们非常有用。

我们以XML技术的概述作为开篇,然后阐述如何在IBM WebSphere中应用XML技术,最后展示一个使用上述技术开发的应用示例。书中还给出了应用示例的源代码和安装步骤。

本书适用于IBM WebSphere Application Server 3.0。

这本红皮书的写作班子

这本红皮书由来自世界各地的专家组创作,这些专家工作在国际技术支持组织(International Technical Support Organization, ITSO)的San Jose中心。

Luis Ennser是ITSO San Jose中心的一位顾问。他从巴西的IME (Instituto Militar de Engenharia, Rio de Janeiro) 取得了机械工程的学士学位。Luis在应用开发和网络技术方面有着15年的经验。他从事IBM全球培训课程的写作与授课,涉及有关XML、WebSphere和Java的所有领域。在加入ITSO之前,Luis作为一个电子商务解决方案设计师在巴西工作。

Christophe Chuvan是一位软件开发及顾问,为Streamlink Pty公司工作,该公司是IBM的一个业务伙伴,它的总部位于澳大利亚的悉尼。他有5年的有关Internet技术的经验,并有3年多的时间专攻电子商务。他从Bond大学取得了计算机科学学士学位,研究方向是网络与通信。他的专长包括电子采购解决方案的咨询与设计,解决方案的平台包括Lotus Notes和Java servlets。

Paul Fremantle是英国的一位电子商务软件设计师,在信息技术及咨询方面有着10年的经验,在Web和Java 方面有着5年的经验。他从Oxford (牛津) 大学的Balliol (贝列尔) 学院取得数学与哲学M.A.学位,从牛津大学的计算实验室获计算机科学硕士学位。他在IBM已经工作了3年。他的专长包括Java、Internet技术与协议、EJB、IBM WebSphere、性能和安全。他已经出版了有关IBM WebSphere 3.0和EJB移植性方面的文章。

Ramani Routray是工作在IBM印度全球服务中心的一位软件工程师。他有2年的面向对象分析与设计,以及Java程序设计的经验。他从Bangalore大学Sri Siddhartha技术学院取得计算机科学与工程学士学位。他的专长包括UNIX、JavaBeans及应用开发、网络技术。

Jouko Ruuskanen是工作在IBM芬兰的一位软件设计师。他有15年的信息技术经验,其中5年主要在电信领域,后10年在面向对象技术与工具方面。他从Helsinki (赫尔辛基) 科技大学取得电子与通信工程的硕士学位。他的专长包括Java、Smalltalk、IBM WebSphere和Internet应用开发。他曾经参与多本IBM 红皮书的写作。

下列人员为本书作出了很大的贡献，特表谢意！ITSO San Jose中心的Andrea Konzett对本书的完成给予了很大的支持。IBM TJ Watson研究中心的Sanjiva Weerawarana提供了有关XSLT的技术指导。位于美国北卡Raleigh 的IBM WebSphere开发组的Elias Bayeh提供了有关WebSphere内XML技术发展方向的信息。IBM Java技术中心的Ted Leung提供了有关XML解析器的帮助。来自EMEA北方软件事业部的Adrian Walmsley审阅了本书并提供了有益的意见。Jane Fremantle为本书起了书名。

目 录

序

译者序

前言

第一部分 XML技术引论

第1章 XML产生的背景	1
1.1 背景	1
1.2 XML——一个通用数据格式	2
1.3 XML的商业作用	2
1.3.1 信息共享	3
1.3.2 XML在单个应用中	3
1.3.3 内容递送	3
1.4 XML的技术作用	4
1.4.1 使用XML的一个示例	4
1.4.2 主要益处	4
1.5 XML的历史	5
1.6 XML的现实用途	6
1.6.1 SABRE和无线标记语言	6
1.6.2 化学标记语言	7
第2章 XML概论	8
2.1 XML概念	8
2.1.1 XML和SGML	8
2.1.2 文档有效性和合规性	8
2.1.3 文档类型定义	9
2.1.4 名字空间	13
2.1.5 DTD与XML Schema	14
2.2 XML Linking	14
2.3 XPath	16
2.4 XPointer	17
2.5 XSL	18
2.5.1 层叠样式表	18
2.5.2 XSL=fo:+XSLT	19
2.5.3 XSL转换	20

2.5.4 XSL和XPath的关系	20
2.5.5 一个XML示例	21
第3章 用Java处理XML	24
3.1 XML应用	24
3.2 SAX	24
3.2.1 SAX类和接口	25
3.2.2 SAX应用示例	26
3.3 DOM	27
3.3.1 DOM层次结构	27
3.3.2 DOM应用示例	28
3.3.3 DOM Level 2	28
3.4 SAX还是DOM	29
3.4.1 SAX的优势和劣势	29
3.4.2 DOM的优势和劣势	29

第二部分 在IBM WebSphere上 应用XML技术

第4章 IBM WAS V3简介	31
4.1 XML Parser for Java	31
4.2 LotusXSL	31
4.3 DTD Catalogs	32
第5章 XML Parser for Java	33
5.1 如何使用DOM分析XML文档	34
5.2 如何使用DOM创建XML文档	37
5.3 读取并查询DTD信息	38
5.4 使用名字空间	38
第6章 LotusXSL	39
6.1 什么是XSLT处理器	39
6.2 XSLT基础	39
6.2.1 处理属性	40
6.2.2 条件处理	41
6.2.3 循环	42
6.2.4 创建属性和元素	43

6.3 LotusXSL API	44
6.3.1 集成解析器和XSL处理器	45
6.3.2 DefaultApplyXSL	45
6.3.3 配置DefaultApplyXSL	45
6.3.4 运行DefaultApplyXSL	46
第7章 WebSphere及XML途径	47
7.1 WebSphere编程模型	47
7.1.1 Servlet	47
7.1.2 JSP	47
7.1.3 Servlet、JSP和JavaBean	47
7.2 在WAS中生成XML	48
7.2.1 println方法	48
7.2.2 DOM途径	49
7.2.3 JSP途径	50
7.2.4 生成XML方法的比较	51
7.3 在WAS中将XSL应用到XML	52
7.3.1 在servlet中使用XSL	52
7.3.2 自动应用XSL	57
7.3.3 XSLT岛	57
7.4 有选择地使用XSL	59
7.4.1 定义可选的Web应用	59
7.4.2 根据浏览器类型应用XSL	60
7.5 方法的比较	60
7.5.1 应用XSL	60
7.5.2 XML/XSL与Servlet/JSP的比较	61

第三部分 ITSO XML应用示例

第8章 XML应用情景	63
8.1 应用概述	63
8.2 数据库体系结构	64
8.3 文档类型定义	65
8.4 应用体系结构	67
第9章 XML应用示例的实现	69
9.1 登录和用户简要表管理	69
9.1.1 用户简要表数据库	69
9.1.2 用户简要表注册servlet	71
9.1.3 登录Servlet	72
9.1.4 用户界面	73

9.2 另一种途径——XSLT Island	78
9.3 搜索引擎	82
9.4 购物篮和订购系统的实现	86
9.4.1 用JavaBean来实现购物篮	86
9.4.2 程序流程	87
9.4.3 orderbasket.jsp文件	88
9.4.4 实现BasketBean	89
9.4.5 XSL样式表——basket.xml	96
9.5 订单跟踪	97
9.5.1 Servlet概述	97
9.5.2 显示订单报告	98
9.5.3 获得单个订单信息	100
9.5.4 生成HTML输出	102
9.6 供应商	107
第10章 DatabaseDom的使用和实现	110
10.1 使用、限制和方法	110
10.1.1 模板文件	110
10.1.2 模板文件元素的定义	112
10.1.3 输出的例子	113
10.1.4 Bean的方法	114
10.1.5 更新数据库	114
10.2 DatabaseDom的实现	115
10.2.1 从数据库中生成XML	115
10.2.2 从XML DOM树更新数据库	116
10.2.3 为XML映射创建字段	116
第11章 结束语	118
11.1 成功之处	118
11.2 取得的经验教训	119
11.3 可进一步探索的领域	120

第四部分 附录

附录A WebSphere安装指南	121
附录B ITSO XML应用示例的安装	146
附录C 数据定义语言	152
附录D 特别注意事项	155
附录E 相关出版物	157
术语表	160
缩写词汇表	162

第一部分 XML技术引论

第1章 XML产生的背景

本章旨在考察可扩展标记语言（XML），我们将概述XML的应用需求、XML的商业作用和技术作用、以及XML的历史，最后给出在实际应用中如何使用XML的示例。

1.1 背景

Internet(因特网)在最近5年中的增长令人惊异。这一增长是由多种因素引起的，包括低成本个人计算机和更好的通信基础设施。然而，Internet增长的最大驱动力在于WWW（World Wide Web，万维网），以及WWW上任何人都可以使用浏览器和电话线来访问的信息、服务和内容。

这一增长与两个技术标准有关，HTML（HyperText Markup Language，超文本标记语言）和HTTP（HyperText Transfer Protocol，超文本传输协议），它们是WWW的基础。它们一起形成了Internet上的通用语言：HTTP提供了通用的通信协议，而HTML以及浏览器提供了通用的用户界面。如果没有这些技术，那么Internet依然是一个有用但并不令人激动的工具，这种状态恐怕要维持20年。与以前的超文本系统不同，HTML是基于文本的，这使得它容易使用；HTML是一个开放的标准，这意味着任何人都能够开发、构建、创建HTML文档和HTML工具。

HTML和HTTP的发明者，Tim Berners-Lee，当然并没有预见到Web的所有用途。HTML自从1992年出现以来已经有了很大的发展，而且许多现代Web站点的发展已经超出了HTML的能力范围——动态数据驱动的Web已经超越了HTML。这并不奇怪，因为HTML原本的设计是为了捕获一种特定的数据——主要是文本文档以及与其他文本文档之间的链接。它起源于CERN（欧洲高能物理实验室）的研究环境，其设计目的是为了共享科学研究信息。

HTML原本的设计是为了描述一个文档的格式。这个格式表示依赖于浏览器。随着Web用途的扩展，越来越多的表示信息被增加到Web页面中，更多的扩充内容被增加到HTML规范中以支持这些特性。

当今的Web包含了许多动态Web站点，它们提供“在线服务”。在Web中，你能够购买几乎任何商品，预定会议席位，检查气象状况，以及其他众多活动。

为了在HTML中表示这些应用所使用的各种数据，Web设计者必须将业务数据和表示信息混合在一个HTML文件中。近期有关CSS（Cascading Style Sheets，层叠样式表）的工作已经改善了这种状况，但即使有了样式表，大多数动态生成的HTML文件仍然是业务数据和表示信息的混合。

对管理大量的复杂动态数据并表示它的期望引发了对通用数据格式的需求——这由XML（eXtensible Markup Language，可扩展标记语言）提供。

1.2 XML——一个通用数据格式

HTML是一个单一的标记语言，为一个特定的应用而设计，而XML实际上是一套标记语言：事实上，你可以用XML定义任何数量的标记语言。这意味着几乎所有的数据类型能很容易地用XML来定义。因此，除了已有的通用通信媒介（Internet）、通用的用户界面（浏览器）和通用的程序设计语言（Java），我们现在有了一个通用数据格式——XML。IBM以及其他许多公司深信这是电子商务（e-business）的坚实基础。

XML是通用的，不仅是从应用范围来看，而且是从其易用性来看：基于文本的本性使得它很容易创建有关工具，它还是一个开放的无需许可证的跨平台的标准，这意味着任何人都能够创建、开发、使用有关XML工具。而XML的通用性还在于它的能力。数据以各种方式被传送并存储在计算机中：最初数据被存储在普通文件中，采用定长或定界的方式，后来被移到数据库中，并通常采用复杂的二进制格式。而XML是一种结构化数据格式，这使得它能用来存储复杂数据，无论是文本的、二进制的或面向对象的。到目前为止，几乎没有那一个数据驱动技术能够一举解决所有这些不同方面的问题——除了XML以外。

XML和HTML的一个简单比较

XML和HTML都源自同一个源头。但是，因为XML还是一项新的技术，Web上大多数数据以HTML格式存储。目前存储在Web上的数据量很难估计：最新的调查可参见OCLC的研究报告，其Web站点如下：

<http://www.oclc.org/oclc/research/projects/webstats/statistics.htm>

该项调查显示Web上至少有3亿个页面。事实上，可用的数据量更高，因为页面记数忽略了这样的一个事实：一个动态生成的页面能作为通向一个大型数据库的网关。

HTML格式的数据带来的问题是这种格式适合于数据的表示，并不适合于计算机使用这些数据。HTML包含一组预先定义好的标记，这些标记的含义以及使用目的是已知的。这使得它成为一个易学易用的语言，但是也使得它很难重用数据。

这就是XML占有优势的地方。正如它的名字所显示的那样，XML是可扩展的，这意味着你可以定义自己的一组标记，并且能够让他人或其他程序认识并理解这些标记。这使得XML比HTML更具灵活性。事实上，因为XML标记能表示数据的逻辑结构，他们能够被解释并能够被不同的应用以各种方式使用。

Web的价值大多在于重用数据。比如，Web成功的因素之一是搜索引擎。它们以HTTP和HTML作为工作基础对Web页面编制目录。但是搜索引擎只能在很有限的信息上工作，因为一个HTML文档只有一小部分可供搜索引擎使用。倘若所搜索的数据存储在一个简单的结构化的可重用的格式中，那么搜索引擎将有何等强大的威力！

1.3 XML的商业作用

迄今为止，XML有三大应用：

- 在计算机系统之间和商业之间共享信息。
- 在单个应用中存储和传送信息。

- 内容递送——递送信息给用户。

XML的早期使用迄今为止主要在前面两个领域——因为XML的作用在这些领域容易实现。

1.3.1 信息共享

在计算机系统之间和商业之间使用XML的作用或许是最大的和最容易实现的：XML允许商业定义数据格式，并且容易构建工具来读数据、写数据、在XML和其他格式之间转换数据。这可使一些商业和工业联盟构建标准的XML数据格式。像EDI（Electronic Data Interchange，电子数据交换）、银行间支付、供应链、贸易和文档管理等领域正在由工业联盟制定基于XML的标准。

通过使用XML，这个标准可以被发布和扩展，并且很容易被新的应用使用。数据转换工具已经存在，它们能够在已有的格式和XML格式之间转换数据，而且新的工具正在出现。XML的链接企业应用（企业应用集成）是许多公司关注的焦点，并且已经为企业消费者降低了成本并增加利润。特别地，许多商业通过在多个已有的系统之上创建单一的逻辑视图来改善客户关系管理（Customer Relationship Management, CRM），而XML是用来创建这样的单一客户视图的一项重要技术。

不仅如此因为XML容易地将结构与内容关联，所以XML能够为特定行业或应用定义子集。比如，XML已经被用来定义银行业的标准的数据格式。同样地，它可以为航空订票系统定制开发一个标准，这样航空公司之间能轻而易举交换信息。

1.3.2 XML在单个应用中

在单个应用中使用XML的作用似乎并不引人注目，但是由于其容易实现，因此我们已经看到了一些应用在内部使用XML。这种方法的好处在于开发的速度和未来的可扩展性。

XML是一种适合于描述现实世界复杂性的强大而灵活的语言。因为XML能够描述复杂数据，这意味着它能成为一个创建新的应用的强大工具。XML的扩展性意味着应用无需太大的变化就能够成长和发展。最后，当应用需要与其它应用连接的时候，XML是一个连接这个应用与其他应用的极好途径。

1.3.3 内容递送

XML对于内容递送有着几点关键优势。主要的优势在于支持不同用户和通道的能力，以及构建更加高效的 ứng dụng 的能力。通道是信息递送机制，比如，数字电视、电话、Web。在将电子商务应用通过各自选择的媒介递送给客户时，支持不同的通道是重要的一步。对此，XML是一个关键技术。

比如，一个客户和一个供应商都需要存取相同的在线产品目录。虽然信息是同一个，他们关注的重点却不同，这与用户的性质有关：客户对查看产品的功能、价格和可用量更感兴趣，而供应商希望能容易地存取产品目录的维持量和库存信息。所有这些信息可以存储在单个XML文档中，并且能被应用以不同的方式来显示。

使用XML递送内容已经受到支持XML浏览器的数量的局限。Microsoft IE 5.0是第一个直接

支持XML的浏览器，在大多数用户准备好使用XML之前，许多Web站点将并不采用XML。正如你将在本书的后面看到的那样，IBM WebSphere允许服务器从XML生成HTML，从而支持标准浏览器的用户。当人们对这个技术更熟悉的时候，XML的这个特别功能将会被更多地开发。

1.4 XML的技术作用

为了观察XML的技术作用，让我们考虑一个假想的示例。

1.4.1 使用XML的一个示例

许多图书馆在Web上提供他们的图书目录。通常，会有一个简单的Web表单，用户要输入一个标题、书名或主题，产生一些搜索结果。如果你要搜索几个图书馆，你将需要访问各个图书馆的Web站点。假如有一个单一的Web页面用来搜索多个图书馆，这将很有用也很方便。当前，要构建这样的系统就需要从各个搜索结果页面中提取有关数据，比如，标题、作者、ISBN等等。但是，各个搜索结果页面的数据格式各不相同，而且这些数据与表示信息混合在一起。要处理多个搜索结果就需要为各个图书馆Web站点进行复杂的编程，以将需要的数据从表示信息中分离开来。

反之，假设有一个单一的格式适用于各个图书馆的搜索结果：这个格式称之为ABCML（A Book Catalog Markup Language，一个图书目录标记语言）。ABCML为标题、作者、ISBN等定义了有关标记，从而使得计算机能轻易地提取这些数据。这时构建这种元目录突然变得容易起来。

ABCML对图书馆将有帮助，因为图书馆能够相互重用软件。同样，当到了一本新书时，出版商能以ABCML格式提供这本书的目录信息，从而减轻图书馆员输入图书目录的工作。

1.4.2 主要益处

因此ABCML将帮助我们构建一个元目录，还帮助图书馆重用已有数据。这些论据也许还不足以说服某个图书馆信息技术主管用ABCML重写在线目录。然而，还有其他一些使用XML的技术优势。除了数据重用，使用XML的好处还包括：将业务数据从显示信息中分离，可扩展性，以及在数据中增加语义信息。

1. 数据重用

我们已经看到数据重用的益处：图书管理员能够重用出版商的数据，因为这些数据采用通用的格式，而且在我们构建元目录的时候能够重用这些数据。当然，在当今世界的计算中有许多允许数据重用的公共文件格式。但这些格式通常是私人拥有的和特定于应用的，而XML却不是这样。

2. 数据和显示的分离

数据和显示的分离能带来什么益处呢？第一，没有这种分离，我们无法实现数据的简单重用。第二，显示的变化较快。比较一下5年前的Web站点和当今的Web站点，它们有着很大的不同。查看一下成功的Web站点，你会发现至少一年一次的重新设计。这不能简单地归结为追求时尚，成功的Web站点分析用户的反馈信息并作出反应，并重新设计站点以达到更具效率更加直观

的目的。让我们考虑图书馆Web站点：Web站点得到了重设计，但是内在的数据依然在那里，因此数据输出从Web站点设计中分离出来很有意义。

还有一个更有说服力的理由来分离数据和显示：普及计算崛起。普及计算发生在计算设备集成到日常使用的设备中：移动电话、电视、打印机、手持计算机。这些设备也许有各不相同的显示技术，需要不同的指令来显示数据。同样的图书馆目录搜索结果应该在移动电话和高分辨率个人计算机上均可显示。

3. 可扩展性

HTML自从出现以来成为一个不断发展的标准，其面临的问题之一是经常被一些公司扩展。浏览器供应商经常给HTML增加一些非标准的扩充。类似地，Web服务器制造商构建“服务端”的HTML扩充，包括NCSA includes、Microsoft Active Server Pages、Java Server Pages和很多其他的扩充。这导致HTML标准的各种变例的混乱，给Web开发者、工具提供商并最终给用户带来困难。

正如其名字所隐含的，XML（eXtensible Markup Language）从一开始就设计为允许扩充。让我们回到图书馆那个示例，当他们首先给图书编写索引时，Web并不存在。图书目录中还没有对Web站点的引用。而现在，许多图书有一个伴随的Web站点，图书馆员也许希望引用它。假如XML被用来开发这个目录，这将很容易地完成。重要的是，使用XML不会使得旧的软件因增加新的信息而崩溃。

4. 语义信息

最后，XML的主要作用是在文档中构建语义信息。语义信息（含义）使得读者能判断一本书是关于褐色（Brown）的书还是由布朗（Brown）写的书。一个基于HTML的Web搜索引擎无法做到这一点，因为它不能区分文档中的标题和作者——在HTML页面中没有足够的语义信息。使用XML，文档包含了自描述的术语，这些术语指示了文档的含义。

5. 其他益处

XML的其他主要益处在于它是人类可读的、基于树状的、容易编程的。随着时间的推移，大量的XML工具将从软件提供商和以XML起家的公司中产生。它是人类可读的，因为它是基于文本和简单标记的机制。这意味着它能被人类所理解，而且只需使用文本编辑器或字处理器就可书写。这一点很重要，程序员在写新的应用时就能更快地解释数据，当然一旦开始运行，除了应用以外没有人读这些数据。XML的基于树状的结构比定长数据格式具有更强大的能力。因为对象也是树状的，XML很适合与面向对象程序设计一起工作。而且，许多人相信Java和XML之间有着无与伦比的亲合力。

最后因为已经有了XML解析器（XML parser）的标准，所以XML容易编程。XML解析器是读XML并允许程序员在XML上编程的子系统。因为XML解析器在新的计算机系统可重用，所以许多程序员开始使用XML，即使他们并不需要前面提到的使用XML带来的益处。有关Java和XML之间的亲合力，以及XML解析器将在第3章“使用Java处理XML”中详细阐述。

1.5 XML的历史

XML的历史其实是另外一个系统SGML（Standard Generalized Markup Language，标准通用

标记语言)的历史。XML实际上只是SGML的一个子集,而SGML已有很多年的历史。事实上SGML可以回溯到60年代后期IBM的一个雇员Charles Goldfarb的工作。Goldfarb当时正在开发一个系统以共享文档,他与另外两个同事Edward Mosher和Raymond Lorie一起,构思了一个称之为GML (Generalized Markup Language, 通用标记语言)的标记语言。(当然,GML其实是代表了Goldfarb、Mosher和Lorie,他们发明了这个语言。)

GML还吸收了William Tunnicliffe和Stanley Rice在构建“通用编码”系统方面的工作,该系统的研发由Graphic Communication Association主办。

GML成为包括Script和Bookmaster再内的众多IBM文档系统的基础。后来GML发展成为SGML,1986年的一个ISO标准。SGML已经有了巨大的成功,但不幸的是主要限于大型企业和政府部门使用。其原因在于SGML需要巨大的投资,因此只有大型组织机构具备足够的资源来实现SGML的优势。有关SGML的详细信息,参阅Charles Goldfarb的著作“The SGML Handbook”。

SGML的最受欢迎的应用也许正是Web: HTML实际上是用SGML定义的,而且是SGML的一个非常有限的应用例子。XML的构想于1996年底提交给W3C (World Wide Web Consortium)。初稿的完整文本可以从<http://www.w3.org/pub/WWW/TR/WD-xml961114.html>中得到。

初稿中包括了十个关键的设计目标如下:

- XML应该适合于在Internet上直接使用。
- XML应该支持广泛的应用。
- XML应该与SGML兼容。
- 处理XML文档的程序应该容易编写。
- XML中可选特性尽可能少,最好没有。
- XML文档应该是人类易读的而且相当清楚。
- XML设计应该能很快地准备好。
- XML的设计应该是形式的和简练的。
- XML文档应该是容易创建的。
- 简洁性的重要性最小。

自从1997年底,XML在W3C的领导下快速地成长,在1999年9月,W3C宣告XML活动已经进入第三阶段:第一阶段构建基础技术,第二阶段创建样式表和名字空间,第三阶段将完成正在进行的工作并且提出一个XML查询标准的新规范。

1.6 XML的现实用途

本节给出2个示例,以展示XML是如何在现实生活中为人们、商业和组织带来益处的。关于更多的示例,参见IBM developerWorks (<http://www.ibm.com/developer/xml>)中的XML部分,或者XML.org 目录站点 (http://www.xml.org/xmlorg_registry/index.shtml)。

1.6.1 SABRE和无线标记语言

SABRE Group是国际旅行服务的主要分销商之一,通过旅行代理商并在Web上提供电子旅行预订。他们使用Java应用将旅行信息转换成XML格式,然后允许全球的移动电话用户从移动

电话来查询、预订、购买旅行服务。XML自动地被转化成WML (Wireless Markup Language, 无线标记语言), 它是一个适合于构建在移动电话上应用的标准。XML对于这个应用的独到之处在于它的可扩展性、开发的速度、构建一个标准的仓库的能力、以及按需转换到特定环境的能力, 在本例中为移动电话。

1.6.2 化学标记语言

化合物和分子是原子的复杂组合。大约有两千万的“已知”分子, 一直到近期, 没有一种可用计算机处理的分子表示方式。两位英国科学家, 诺丁汉大学的Peter Murray Rust教授和伦敦皇家学院的Henry Rezpa 博士, 开发了一种用XML描述分子的标准方法。CML (Chemical Markup Language, 化学标记语言) 预期能为化学工业带来巨大的节约, 而且还有助于化学家之间的交流, 以及有助于其他相关学科, 比如生物学和医学。

在这个应用中使用XML的一个关键优势在于现有的大量的XML工具, 它们能帮助快速而高效地创建CML应用。

第2章 XML概论

前面首先探讨了为什么要开发XML。在接下去的部分里，我们将讨论XML的一些基本概念，如标签（tag）、文档类型定义（Document Type Definition）以及名字空间。

2.1 XML概念

尽管这本书不是要写成为XML参考手册，但我们觉得介绍一些基本概念还是有用的，因为这本书的第二部分需要读者对这些概念有基本的理解。HTML是一个流行的标记语言，很多人对它很熟悉，所以你会注意到这本书里常常有对HTML、XML和它们的共同祖先SGML的比较。虽然对HTML的熟悉会使人们更容易理解XML，但我们还是要把重点放在这两种语言之间的差异上。

2.1.1 XML和SGML

我们已经提到过，XML是SGML的一个子集，SGML比XML更强大但也复杂得多。同SGML一样，XML也是种元语言，它要求文档的作者给出标记（markup）的定义，换句话说，把“标签”和数据联系起来。使用标签的目的是通过标签和文档内容的联系，向应用程序提供关于数据的信息。

因此，一个XML文档一般包括标记（markup）和字符数据（character data）。标记信息通过标签(tag)和其他XML元素来表达隐含在文档内容中的含义；字符数据表示真正的内容。

在下面的例子里，<author>标签指出Joe Bloggs是一个作者的名字：

```
<author>Joe Bloggs</author>
```

请注意，在上面的例子里，虽然我们知道Joe Bloggs是一个作者的名字，但并不是说他一定就是文档的作者。标签后面内容的真实含义要根据这个标签被解释时的上下文来确定，所以要依赖于读这个文档的应用程序。

2.1.2 文档有效性和良规性

XML是一个元语言，这就意味着，作者可以使用XML文档定义一些适用应用程序的标记。但是，XML的主要优势是：它可以被不同的组织所阅读，用不同的方法解释。所以，XML标记集和文档必须遵循少许规则，以确保任何XML应用都能使用它们。

我们前面已经提到过，XML的约束要比HTML强，HTML可以容忍文档里的一些结构上的小失误，如标签没有闭合等。而对XML解析器来说，如果一个文档里有一个开始标签，如<AUTHOR>，而没有对应的的结束标签——在这个例子里是</AUTHOR>，就会拒绝接受。在这一点上，XML同HTML、SGML不一样，对HTML和SGML来说，即使一个结束标签都没有，语