

编程宝典
2002
1

北京希望电子出版社 总策划
com 工作室 创作 黄理 曹林有 张勇等 编写

全面引爆——XSP 网站开发

XSP COCOON XML 核心技术内幕

运用最新的操作系统平台——Windows XP

基于最新的 XML 技术规范——XHTML, WML, XSLT……

最新的基于 Java 的 Server Pages 技术——XSP

讨论最新的 Web 出版框架——Cocoon



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn



96
6.00 元



元



0 元



元



8
0.00 元

TP383-1
H766

编程宝典 2002 1

北京希望电子出版社 总策划
com 工作室 创作 黄理 曹林有 张勇等 编写

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载,
也可到视听部复制

全面引爆——XSP 网站开发

XSP COCOON XML 核心技术内幕

运用最新的操作系统平台——Windows XP

基于最新的 XML 技术规范——XHTML, WML, XSLT……

最新的基于 Java 的 Server Pages 技术——XSP

讨论最新的 Web 出版框架——Cocoon



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

这是一本面向初、中级编程人员的自学指导教程。全面介绍新一代服务器页面开发技术——XSP (eXtensible Server Pages), 以及全新的 Web 出版框架——Cocoon 技术。Server Pages 家族的新成员 XSP 技术, 因其基于 Cocoon 技术使 XSP 程序具有静态内容、程序逻辑、表现样式三层分离的特性, 给编辑、重用和修复代码带来极大便利, 使开发效率极大提升。XSP 正被获得前所未有的认可, 越来越多的迹象表明, 它将带来服务器页面的新时代。

本版由 12 章和 3 个附录组成。主要内容包括: XML, DTD, XSL, XSL FO, Namespaces, Schema, XHTML, SVG 和 WML 等相关技术的介绍; Cocoon 的基础知识; Cocoon 的配置方法; XSP 的基础语法; Cocoon 处理器; XSPUtil 类的用法; XSP 内建对象的用法; XSP 逻辑单的基础知识以及开发流程; 最后给出常用 XSP 逻辑单的用法参考; 附录中提供了关于 XSP/Cocoon/XML 的大量资源列表。

本版内容, 运用最新的操作系统平台——Windows XP, 基于最新的 XML 技术规范——XHTML, WML, XSLT……, 介绍最新的 Server Pages 技术——XSP, 讨论最新的 Web 出版框架——Cocoon。

“用事实讲话”是本教程的最大特色: 用大量实例说明每个概念, 所有相关资源都注明出处, 秉承了专业人员接受新技术的严谨性, 同时体现了专业人员独特的、清晰的学习思路, 全书语言简明扼要, 内容丰富, 包括大小数百个典型范例, 让理论与实现无缝结合。

本版可作为专业网站开发人员和网络编程爱好者的 XSP 技术指导书, 也可作为社会网页设计或编程培训班用教材。

本版 CD 提供本版实例的源代码。

盘 书 系 列 名 : “十五”国家重点电子出版物规划项目 计算机基础知识普及和软件开发系列
编程宝典 2002 (1)

盘 书 名 : 全面引爆XSP网站开发——XSP/Cocoon/XML核心技术内幕

总 策 划 : 北京希望电子出版社

文 本 著 作 者 : com 工作室 创作 黄理 曹林有 张勇 等 编写

C D 制 作 者 : 希望多媒体开发中心

C D 测 试 者 : 希望多媒体测试部

责 任 编 辑 : 栾大成 刘晓融

出版、发 行 者 : 北京希望电子出版社

地 址 : 北京中关村大街 26 号, 100080

网 址 : www.bhp.com.cn

E-mail: lwm@hope.com.cn

电话: 010-62562329, 62541992, 62637101, 62637102, 62633308, 62633309

(图书发行) 010-62613322-215 (门市) 010-62547735 (编辑部)

经 销 : 各地新华书店、软件连锁店

排 版 : 希望图书输出中心 周宇

CD 生 产 者 : 北京中新联光盘有限责任公司

文 本 印 刷 者 : 北京广益印刷有限公司

开 本 / 规 格 : 787 毫米×1092 毫米 16 开本 38.625 印张 889 千字

版 次 / 印 次 : 2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷

印 数 : 0001-5000 册

本 版 号 : ISBN 7-980007-70-0

定 价 : 58.00 元 (本版 CD)

说明: 凡我社产品如有缺页, 可执相关凭证与本社调换。

前 言

1999 年网站开发人员说得最多、用得最多的是 ASP；2000 年，JSP 进入我们的视野；到了 2001 年，XSP 成为新的热点话题，2002 年，XSP 技术将大行其道，一统江湖！？

XSP 是 Server Pages（服务器页面）家族的新成员，它正在获得前所未有的认可，越来越多的迹象表明，它将带来服务器页面的新时代。

XSP 的全称是可扩展服务器页面（eXtensible Server Pages），它是 Cocoon 的核心技术之一。Cocoon 的原意是蚕茧，不过在本书中 Cocoon 是著名的 Apache XML 项目的 7 大技术之一。因为网站的文档内容、表现样式和程序逻辑常常是由不同的角色来提供的，Cocoon 的目标就是将以上 3 个层次完全分离。Cocoon 将网页的静态内容、程序逻辑和表现样式分离成不同的 XML 文件存放，然后采用 XSLT 的转化能力来进行动态粘合。Cocoon 是一个崭新的 Web 出版框架，与别的 Web 出版框架相比，采用 Cocoon 技术将会大大提高 Web 站点的开发速度以及站点的可管理性。

XSP 的优越性

尽管在概念上 XSP 与 JSP 技术有相近之处，但是 XSP 技术仍然有一些突出的优点。

- ✎ 由于 XSP 技术依赖于 Cocoon 技术的实现，所以 XSP 程序自然而然地具有静态内容、程序逻辑、表现样式 3 层分离的特性，开发效率大大加快。
- ✎ XSP 技术的学习曲线十分平坦，不但程序员可以很快地掌握 XSP 技术，就连网站文字编辑也可以轻松地编写 XSP 程序。
- ✎ 编写 XSP 程序就和编写 HTML 页面一样简单。
- ✎ 网络编程人员如果选择了 Cocoon 和 XSP 技术，那么再也不需要在普通的页面中嵌入大量的脚本代码，再也不用为搞不清楚程序代码复杂的嵌套关系而大伤脑筋了。
- ✎ 不需要担心网站编辑把你辛辛苦苦编写出来的代码搞乱，弄得无法运行。
- ✎ 可以同时为采用不同设备上网的用户（包括移动设备用户和桌面用户）提供信息服务，而无需进行重复开发，极大地节约了人力和物力。

XSP 技术采用的脚本语言是 Java，至于 Java 语言的功能如何，还用多说吗？从小小的手机到大型服务器，从 P2P 到企业级分布式型服务器，从 P2P 到企业级分布式以 XSP 程序的功能是十分强大的，选择了 XSP 以及 Cocoon 技术，就相当于获得了数万个 Java 类库的支持。对于 XSP 程序的运行效率更是不用怀疑，因为 XSP 程序和 JSP 程序一样，同样是编译运行的，而且还有 Cocoon 强大的缓存系统的支持，所以 XSP 程序的运行效率十分高。笔者经过测试，XSP 程序的运行效率比 JSP 程序还要稍胜一筹。XSP 既然有这么多优点，我们还有什么理由拒绝呢？

XSP 与 ASP.NET

Microsoft 最近推出了 ASP.NET，同样是用于 Web 开发的最新技术，包含了许多新特

性,那么 XSP 技术与 ASP.NET 技术相比,究竟孰优孰劣呢?应该承认,ASP.NET 技术在 ASP 基础上做了很大改进,然而,ASP.NET 技术并没有“脱胎换骨”的改变,其本质和 ASP 技术并没有区别,它仍然属于第 1 代 Server Pages。ASP.NET 技术不具备上述 XSP 技术所具有的优点。笔者认为,XSP 技术在总体设计、具体实现、体系构架上比 ASP.NET 技术要先进很多,更具有前瞻性,尽管在某些技术细节上不如 ASP.NET 技术。ASP.NET 与 XSP 这两者谁将最终一统江湖?让我们静观事态,拭目以待。

主要内容

本书主要讨论 XSP 和 Cocoon 以及有关的 XML 技术。本书共分 12 章和 3 个附录,每章的内容大致如下所述。

第 1 章 XML 基础,介绍与 XML 以及 DTD 有关的知识。

第 2 章 XML 相关技术基础,介绍 XSL, XSL FO, Namespace 和 Schema。其中 XSL 和 XSL FO 是重点。

第 3 章 XML 技术应用,介绍 SVG, WML, XHTML 三种从 XML 中扩展而来的技术。

第 4 章 XSLT 元素参考,讨论 XSLT 中所有元素的用法参考。

第 5 章 Apache Cocoon 基础知识,主要讲述 Cocoon 1 和 Cocoon 2 的基础知识,还简要讨论了它们的运行机理。

第 6 章 Cocoon 的配置,详细讨论了如何配置 Cocoon 1 和 Cocoon 2,使得它们和主服务器协同工作。

第 7 章 XSP 语法入门,讲述了 XSP 的基本语法,重点是 XSP 元素的用法。

第 8 章 XSP 高级编程,讨论 Cocoon 处理器、XSPUtil 类的用法、XSP 编程实例,重点是 XSP 编程实例。

第 9 章 XSP 内建对象,讨论 XSP 十个内建对象的用法,还有 Cookie 对象的用法。

第 10 章 XSP 逻辑单,主要论述 XSP 逻辑单的原理以及如何开发自己的逻辑单。

第 11 章典型的 XSP 逻辑单(1),介绍 Context Logicsheet, Cookie Logicsheet, Global Logicsheet, Request Logicsheet, Response Logicsheet 和 Session Logicsheet 等逻辑单的用法。

第 12 章典型的 XSP 逻辑单(2),讨论 Util Logicsheet, SQL Logicsheet, ESQL Logicsheet, Mail Logicsheet 和 Log Logicsheet 等逻辑单的用法。

附录 1 是技术术语词汇表,附录 2 是网络学习资源列表,附录 3 是本书参考文献列表。

建议阅读方式

本书前 4 章内容是与 XML 有关的技术,如果读者对书中讲述的内容十分熟悉,那么可以跳过不读。第 5 章的 Apache Cocoon 基础知识不是很重要,读者只要有一个大致的概念就可以了。第 6 章是本书重点,尤其是介绍 Cocoon 1 和 Cocoon 2 配置文件的部分,读者一定要特别留意,至于讨论如何安装 Cocoon 的部分,读者可根据自己使用的做主服务器的软件选择阅读相关的那部分内容,不必通读全章。第 7、第 8、第 9、第 10 章也是本书重点,讲述 Cocoon 和 XSP 技术的核心内容,读者一定要认真阅读。第 7 章是基础,必须

先学习第 7 章，至于随后的 3 章，可以随意安排学习顺序。这 3 章讲述的内容属于不同方面，它们之间没有什么紧密的逻辑关系。本书第 11 和第 12 章属于参考性的资料，读者不必仔细阅读这两章，需要时再查阅即可。

读者应具备的基础

阅读本书需要具有一定编程基础，必须有编写普通 HTML 网页的经历，而且应该对 Java 语言有一定的了解。由于篇幅关系，本书没有介绍 Java 语法，不过与 C/C++ 等语言相比，Java 语言显得更简单一些，对没有学过 Java 语言的读者，建议在阅读本书之前，阅读一本专门介绍 Java 语言的书籍来充充电，我们推荐“*Thinking in Java*”一书。

本书由 com 工作室创作策划，由 com 工作室首席技术官黄理先生领衔执笔编写；本书的出版首先归功于黄理先生，他对于计算机尤其是基于 Java 的互联网技术之敏锐感觉以及深厚的计算机天赋，使得本书得以保持在世界范围的领先地位；他在短时间内的超强度忘我工作，使得国内读者能够与世界同步接触到最新的互联网技术。本书的及时出版也要感谢北京希望电子出版社的领导和编辑，正是出版社的支持使得 com 工作室能够一如既往地向读者推介最新的互联网技术。

由于 XSP 技术发展的十分迅速，加上时间仓促，笔者的水平有限，书中的谬误之处一定很多，在此恳请读者朋友加以指正。

com 工作室

于北京大学

第 1 章 XML 基础

第 1 章开宗明义, 我们首先介绍 XSP 技术的由来、优劣、基础语法等等。由于 XSP 技术和 XML 技术以及众多由 XML 技术延伸的技术关系十分密切, 或说 XSP 技术实际上就是基于 XML 的 Server Pages 技术, 读者如果不具备相当的 XML 基础知识是不可能掌握好 XSP 技术的。所以本章的主要内容就是向读者介绍与 XML 技术有关的基础知识。本书的主旨并非专门讨论 XML 技术, 所以本章对 XML 的介绍只是取其精要, 既不深入, 也不全面。如果读者希望对 XML 技术有更深入的了解, 可以参考其他相关书籍。

1.1 席卷全球的 XML 技术

1.1.1 引入 XML

网络技术发展到今日, 已经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, 单靠抽象的 TCP/IP 协议、金字塔型的 OSI 七层模型、蛛网般的光缆、数以百万计日夜运转的 Unix 和 Windows 主机或者是数十万台冷冰冰的 Cisco 公司的路由器, 以及交换机是远远不够的。因此, HTML——超文本标识语言便应运而生, 并以其简单精炼的语法、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的网民, 互联网也才得以普及发展以致于今日的辉煌。

然而, 电子商务、电子图书、远程教育等全新领域如异军突起, 迅猛发展并逐渐成为互联网世界必不可少且愈发重要的组成部分, 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, 那就是同样的数据能否根据不同用户或者是不同需求而以不同的效果、形式表达、再现出来。

举一个简单的例子。假设我们要在网上拍卖一幅油画, 而在一个 Web 文件中包含了有关这幅油画的全部信息, 包括油画名称、油画简介、作者名称、作者简介、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油画本身的信息, 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油画的资料, 他们有的喜欢作者, 有的喜欢油画的内容, 有的则只喜欢画的色彩等等; 另外一些“穷光蛋”也许只想过过瘾, 看看油画即可; 也许还有一些“别有用心”的人, 他们根本就不关心什么油画、拍卖, 只想让计算机接收到这个文件后能够自动地将其分门别类, 放到自己与之相对应的数据库中(做搜索引擎用)。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 语言所欠缺的恰恰就是这一点, 它的简单性与普遍性原则虽曾使 Web 技术的推广受益匪浅, 但反过来也成为其发展的最大桎梏。尽管人们已付诸很多努力, 不遗余力的对 HTML 语言做改进, 包括各式各样的修改扩充, 如增加表格 (Table)、框架 (Frame)、脚本语言 (Script Language)、样式单技术 (CSS Level 1、CSS Level 2、CSS Level 3) 等等, 但均未能从根本上改变 HTML 语言的局限性。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，成为一股不可遏制的席卷全球的浪潮，这便是 XML——可扩展标识语言（eXtensible Markup Language）。

XML 是互联网联合组织（W3C）定义的一组规范，以便于软件开发人员和内容创作者在网页上组织信息，其目的不仅在于满足不断增长的网络应用需求，同时还希望借此能够确保在通过网络进行交互合作时，具有良好的可靠性与互操作性。

与 HTML 一样，XML 也源自 SGML（全称为 Standard Generalize Markup Language，这是一种老资格的通用标记语言），它保留了 SGML 80% 的功能，使复杂程度降低了 20%。虽然复杂程度下降不多，但是 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，也可以包含具体的描述，从而使 XML 文档的结构可以复杂到任意程度。XML 技术主要有 3 个要素：

- ✎ **Schema 或者 DTD（模式或者文档类型定义）** 规定了 XML 文档的逻辑结构，定义了 XML 文档中的元素、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 的解析程序校验 XML 文档标记的合法性。
- ✎ **XSL（全称为 eXtensible Stylesheet Language，可扩展样式语言）** 用于转换 XML 文档表现样式的语言，它能在客户端使 Web 浏览器改变 XML 文档的表示法，从而不需要再与服务器进行交互通信。
- ✎ **XLL（全称为 eXtensible Link Language，可扩展链接语言）** 虽然 XSL 也能够对 XML 文档产生作用，但 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 4 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。

1.1.2 为什么使用 XML?

XML 给基于 Web 的应用系统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。使用 XML，可以在下列方面得到看得见的好处。

- ✎ **更有意义的搜索** 数据可被 XML 唯一地标识。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。有了 XML，书籍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。
- ✎ **开发灵活的 Web 应用软件** 数据源一旦建立，存放数据的 XML 文档可以被发送到其

他应用软件、对象或者中间层服务器做进一步地处理。或者它可以被直接发送到 PC 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 应用软件的开发提供了所需的技术。

- ✎ **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端数据库和其他应用处来的数据进行集成。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。
- ✎ **多种应用得到的数据** XML 的扩展性和灵活性允许它描述来自不同种类应用软件中的数据，从描述搜集的 Web 页到描述查询数据库所返回的记录等等。同时，由于基于 XML 的数据是自我描述的，所以 XML 数据不需要有内部描述就能被交换和处理。
- ✎ **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，而不仅仅是显示它们。XML 文档对象模式 (DOM) 允许用脚本或其他编程语言处理 XML 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 XML 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基础之上。
- ✎ **数据的多样显示** XML 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，被广泛用来描述使用者接触的界面。HTML 描述数据的外观，而 XML 描述数据本身。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 XML 数据能够以客户端配置、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强大的支持。
- ✎ **粒状更新** 通过使用 XML 技术，数据可以粒状地更新。所谓的粒状更新，实际上就是局部更新。每当一部分数据变化后，不需要重发完整的结构化的数据。只须从服务器发送给客户变化的元素，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。目前的非 XML 技术，只要一条数据变化了，一整页都必须重建，这严重限制了服务器以及客户端的运行性能。XML 也允许动态加进其他数据，比如预测温度、股市行情、外汇报价等。加入的信息能够流入存在的页面，不需要浏览器重新接受一个新页面。
- ✎ **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 页面一样使用 HTTP/1.1 协议进行传送，不需要对现存的网络进行变化。
- ✎ **升级性** 由于 XML 彻底把数据同显示机制分离，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 (XSL 技术可以做到这一点)。这是令人难以置信的强大机制，使得客户端计算机同使用者间的交互作用尽可能地减少了，同时减少了服务器与客户端的数据交换量和浏览器等待服务器响应的的时间。
- ✎ **压缩性** XML 压缩性能很好，因为用于描述数据结构的标记可以重复使用。XML 数据是否要压缩要根据应用来定，还取决于服务器与客户间数据的传递量。XML 能够使用 HTTP/1.1 协议中的压缩标准。
- ✎ **开放的标准** XML 标准是针对 Web 进行过优化的。微软和其他一些公司以及 W3C 中

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。

- ✎ **微软产品的支持** 微软的 IE 4.0 开始支持通用的 XML 解析器 (MSXML)、XML 对象模式 (XML DOM) 和 XML 数据岛 (XML Data Island)。最新的 Internet Explorer 5.0/6.0 和 Office 2000/2002 支持许多添加的 XML 特性和技术。目前, 微软已经把 XML 解析器集成到 Windows 操作系统中, 这使操作系统可以直接对 XML 数据进行处理。
- ✎ **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、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被建立起来, 被应用于某些关键市场, 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, 这使得客户的机会更进一步地扩大。当站点更多地分发数据, 而不仅仅是提供数据浏览时, 特别的机会就产生了。

顾客服务正从电话和传真、上门服务转移到 Web 站点上来, 而且将会由于 XML 的强大功能而受益良多。并且, 由于大多数商业应用软件支持数据的处理和转移, 如购买单、发货单、顾客信息、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 的面向企业内部互联网的标记库, 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。这些标记库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。

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, 包括用来从数据库和存在的使用者界面中产生 XML 数据的工具。开发从数据库中产生 XML 数据的可视化工具也是个很好的机会。浏览 XML 数据的工具可以用 Visual Basic, Java 和 C++ 编写。目前, 需要强大的工具用来在文档中显示丰富、复杂的 XML 数据, 这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 (XSL 正是这样做的)。

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, 可以避免将老化的信息提供给客户。开发用来制定规则、实现规则、基于 XML 的工具是个巨大的机会。我们可以用程序脚本编写一个标准对象模型来过滤新输入的信息、检查储存的信息、创建输出的信息、更新数据库等等。

1.1.3 网络计算的二次革命

当人们刚刚开始学会读书写字的时候, 他们使用简单的文字也能够表达他们的想法。但是随着个人的学习和成长, 还有知识的扩展, 人们的思想越来越复杂, 简单的语言再也不能表达一些细微的差异, 这正表明了人类的进步和发展。

今天, 人们正在使用的网络也正面临相似的情况, 网络的成长速度远远超过了单个个人的发展, 需要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 语言来表达自己的“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。可扩展标记语言 XML (eXtensible Markup Language) 的出现极大地丰富了网络表达的语言集合。

与 HTML 语言一样, XML 也是网络上的一种通用语言。但是 HTML 更像一个易用的阅读器, 简单实用却不精确。XML 则不同, 它的产生就是要通过标记 (Tag) 来精确地表

现信息的各种含义。通过一些自我解释型的标记文本,XML 能够让不同的应用系统理解相同的意义。

有人说,XML 代表了从信息阶段到知识阶段的转移。当网络上的两台主机之间能够用标准通用的通信协议(如 TCP/IP 协议)进行通信时,只能说它们具备了信息传递的能力。至于说传递信息的含义如何解释以及通过何种形式把这些信息再现给人们,以往的语言并不能保证。HTML 用尽量一致的显示方式保证信息的再现,可是假如两台主机传递有关天气预报的信息,能够实现一个人看到天气的图形信息,而另外一个人(如果是盲人)能够听到天气的情况吗?换句话说,能够让主机之间传递一种语言,而让不同的主机用自己的方式表达相同的知识含义吗?

目前,只有 XML 技术能够承担这样的重任。XML 能够有效地表达网络上的各种知识,为信息的交换和网络计算提供新的载体。XML 技术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不同系统之间互操作的光明前景。从这方面看,XML 技术和 Java 技术有相似之处。Java 技术有着非常诱人的初衷,让许多人在这样的一种理想的感召下为想象中的各种系统之间的互操作能力而投入积极的开发中。但是实际上,Java 已经成为一种成功的商业计算工具(例如 J2EE),取得了很大的技术进步。Java 试图从统一计算平台的角度来实现互操作,但是如果没有 XML 的帮助,这可能永远都是一个梦想(现在你应该明白为什么 XML 和 Java 结合得如此紧密了)。

XML 将对网络进入下一个阶段起催化作用。今后几年的网络上,将出现更智能的搜索引擎,不会再出现你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。网络站点采用 XML 技术能够实现高度个性化的模式,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,也要定义一些能够规范描述性标记的模式,称为 Schema),进行自动化的电子商务。一家在北京海淀中关村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(e-book),用可交换的描述方式描述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或者是移动设备在街头的电子书店下载最新的图书,就像我们使用 ATM 取款机一样方便。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,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政府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。而且很幸运, 我们今天有机会站在了这种变化机遇的边缘。当然, 既可以选择参与到推动这种变化的工作中, 也可以袖手旁观, 无所作为。中国的软件和网络产业也都处在这种机遇的边缘, 我们应该选择那种态度和动作呢? 这是一个值得考虑的问题。

1.2 XML 基础

本节的主要目的是向读者介绍 XML 的基础知识。如果读者已经具有一定的 XML 基础, 那么你不妨跳过本节, 直接阅读下面的内容, 这样可以节约你的时间。

1.2.1 XML 介绍

本小节的主题是: XML 是为了描述数据而设计的, 它将重点放在什么是数据上面。HTML 是为了显示数据而设计的, 它将重点放在如何显示数据上面。

应该具有的预备知识

在开始学习 XML 以前, 应该具有下列预备知识。

- ✎ WWW、HTML 以及设计 Web 页面的基础知识。
- ✎ JavaScript 或者是 VBScript 等 Web 脚本编程语言。

如果读者还没有掌握这些知识, 那么我们建议在阅读下面的章节以前, 最好找一些与之相关的书籍来学习一下, 打好基础, 然后再继续本书的学习旅程, 这样才能事半功倍。

什么是 XML?

在上文已经提到过这个问题了, 这里我们再次强调什么是 XML 这个问题。

- ✎ XML 就是 eXtensible Markup Language (可扩展标记语言) 的缩写。
- ✎ XML 是一种十分类似于 HTML 的标记语言。
- ✎ XML 被设计用来描述数据。
- ✎ XML 文档中所使用的标记并非 XML 预定义的, 你必须定义自己的标记。
- ✎ XML 使用 DTD (Document Type Definition, 文档类型定义) 或者是 Schema (模式) 来描述数据。
- ✎ XML 和 DTD 被设计为自描述 (Self-Descriptive) 的语言。

XML 与 HTML 的主要区别

XML 被设计用来携带存放数据。XML 并不是 HTML 的替代品。这一点不容置疑。目前市场上有很多介绍 XML 的书籍在这个观点上都犯了错误。一般的观点是 XML 将会取代 HTML, 这个观点是错误的, 但是却以讹传讹, 流传甚广。XML 之所以不是 HTML 的替代品, 是因为 XML 和 HTML 被设计出来的目标就各不相同。

- ✎ XML 是为了描述数据而设计的, 它将焦点放在什么是数据、如何存放数据上面。

- ✎ HTML 是为了显示数据而设计的，它将焦点放在数据看起来像什么、如何显示数据更好上面。
- ✎ HTML 与显示信息相关，而 XML 则与描述信息相关。

XML 什么也不做？

XML 什么也不做？也许这个标题难以理解，但是 XML 确实是不做任何事情。W3C 组织制定 XML 的标准只是为了组织、存储和发送信息。

下面的例子是 fancy 给 rainbow 的便条，此便条信息使用 XML 的格式存储。

```
<note>
<to>rainbow</to>
<from>fancy</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

这个便条有信息头，还有信息主体，它包含了便条发送者和接受者的信息。但是即使如此，这个 XML 文档仍然不做任何事情，它只是使用 XML 标记包装起来的纯粹信息。有人已经编写出软件来发送、接收并且显示这种格式的信息。这个例子和我们日常生活中接触的手机短信息有点类似之处。

XML 是自由的、可以扩展的

XML 标记并不是 XML 技术自身预定义好的。你必须“发明”自己使用的标记。用于组织 HTML 页面的标记却都是预定义好的。HTML 页面的作者只能够使用 HTML 标准定义好的标记，例如<p>和<h1>。如果在 HTML 页面中使用了 HTML 标准以外的标记，最有可能发生的情况是浏览器会忽略这些标记，不显示任何信息，最坏的情况是浏览器忽然不能正常关闭（笔者已经多次碰到这种情况了）。

XML 允许你定义自己的标记以及 XML 文档的结构。在上面所举的便条的例子中，<to>和<from>等标记都不是在 XML 标准中定义好的，<body>标记虽然在 HTML 标准中有定义，但是这个例子并非 HTML 页面，所以此处的<body>标记和 HTML 页面中的<body>标记完全不同，可以说是全新的标记。这些标记都是此 XML 文档作者自己“发明”的标记。

XML 是 HTML 的补充

XML 不是 HTML 的替代品！理解这一点十分重要。在将来的 Web 开发中，XML 将被用于描述、存放数据，而 HTML 则用于格式化和显示同样的数据。

对于 XML 最好的描述可能是：XML 是一种跨平台的，与软件和硬件无关的，传输信息的工具。

XML 在未来 Web 开发中的作用

XML 无处不在。关于 XML 的标准发展得非常之快，这实在令人感到惊奇。有如此之多的软件开发商已经采用了 XML 标准。

我们坚信，在未来的 Web 开发中，XML 将和 HTML 一般受到重视，它们都是 Web 技术的基础。XML 将成为最普遍的数据操纵和数据传输的工具。

1.2.2 何处使用 XML?

XML 被设计用于存储数据、携带数据和交换数据。它不是为显示数据而设计的。了解这两点很重要。

XML 可以从 HTML 中分离数据

有了 XML，你的数据可以在 HTML 文件之外存储。在不使用 XML 的情况下，HTML 用于显示数据，而数据必须存储在 HTML 文件之内，数据和格式化的标记混合在一起，也许还会有一些客户端运行的脚本。使用了 XML，数据就可以存放在分离的 XML 文档之中。这个方法可以让你集中精力于使用 HTML 做好数据的布局与显示工作，并且确保底层数据的改变不会导致需要对你的 HTML 文件做任何改动。

XML 数据同样可以以“数据岛”的形式存储在 HTML 页面当中（仅有 Microsoft 的 Internet Explorer 浏览器支持这种技术），你仍然可以集中精力于使用 HTML 做好数据的格式化与显示工作。

XML 用于交换数据

使用 XML，我们可以在不兼容的平台之间交换数据。在现实世界中，计算机系统和数据库系统所存储的数据有各种各样的格式。对于开发者而言，最耗时的工作之一就是在遍布网络的系统之间交换数据。将数据转换为 XML 的格式存储可以大大减少交换数据时的复杂性，并且还可以使得这些数据能够被不同类型的程序所读取。

XML 和 B2B

使用 XML，金融信息可以在网络中交换。在不远的将来，我们可以期望看到许多关于 XML 和 B2B（Business To Business）的应用。XML 正在成为遍布网络的商业系统之间交换金融信息所使用的主要语言（例如 ebXML）。许多与 B2B 有关的完全基于 XML 的应用程序正在开发之中。

XML 可以用于共享数据

利用 XML，纯文本文件可以用来共享数据。既然 XML 文档是以纯文本的格式存储，那么 XML 提供了一种与软件和硬件无关的共享数据的方法。这样使得创建能够被不同应用程序读取的数据文件就容易的多了。

同样的道理，我们扩展系统，升级旧系统为新的操作系统，升级服务器、应用程序、采用新的浏览器也就方便多了，容易多了，再也不用担心会产生文件格式不兼容的情况产生了。

XML 可以用于存储数据

利用 XML，纯文本文件可以用于存储数据。大量的数据可以存储在数据库系统中，或者是存储在 XML 文档中，应用程序可以往存储系统中写入数据或者是取回数据。一般的应用程序则可以显示取出的数据。

XML 可以使数据更为有用

采用 XML，你的数据可以被更多的用户利用。既然 XML 是与软件、硬件、应用无关

的，所以可以使你的数据被更多的用户、更多的设备所利用而不仅仅只用于标准的 HTML 浏览器。

别的客户端（如移动设备、WAP 手机）和应用程序可以把你的 XML 文档当作数据源来处理，就像它们对待数据库一样。你的数据可以被各种各样的“阅读机器”处理，这对某些人来说是很方便的，例如盲人或者是残疾人。

XML 可以用于创建新的语言

XML 是 WAP 协议和 WML 语言的母亲。无线标记语言（WML，全称为 Wireless Markup Language）用于标识运行于手持设备上（例如移动电话）的 Internet 程序。WML 采用了 XML 的标准。我们在第 3 章对 WAP 和 WML 还有一番详细的介绍。

关于预见性

如果开发者有足够的预见性，所有将来开发的应用程序都应该使用 XML 的形式来存储数据。将来我们所使用的字处理器、电子制表软件和数据库应该能够以纯文本的格式相互读取数据，而无需经过格式转换的过程。我们现在只能期望 Microsoft 和其他的软件开发商赞同这一个观点。

注意 有读者会问，Word 不是可以读取 Excel 的 xls 文件吗？是的，确实如此，但是 Word 和 Excel 都是 Microsoft 公司的产品，同一个公司的产品之间能够相互交换数据不足为奇。然而，不同公司的字处理器之间呢？比如国产某字处理软件和 Word 2000，该软件可以读取 Word 2000 的 doc 文件，但是必须经过一番格式转换，结果差强人意，经常发生字符丢失、乱码、格式错乱的情况。在没有安装插件的情况下，Word 2000 根本没有办法读取该国产字处理软件所创建的文件。这种情况下，不同公司的不同软件之间交换数据实现起来有相当的困难。就算是同一个公司的产品，它们之间也难以做到无损耗地相互交换数据。如何解决这些问题呢？采用 XML 作为保存数据的文件格式应该是一种很好的解决方法。（不过占据垄断地位的软件生产商多半不会同意这个观点。）

1.2.3 XML 语法

XML 的语法规则非常简单和非常严格，非常容易学会和非常容易使用。正因为如此，编写读取和操作 XML 的软件也是相对容易的事情。

XML 文档的一个例子

XML 文档使用自描述的和简单的语法。请看下面的例子。

程序清单 1.1 (note.xml)

```
<?xml version="1.0" ?>
<note>
  <to>rainbow</to>
  <from>fancy</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

程序清单 1.1 的第 1 行是所谓的 XML 声明——定义此文档所遵循的 XML 标准的版本。在这个例子中遵循的是 XML 标准 1.0 版本。

程序清单 1.1 的第 2 行是此文档的根元素 `note`（就像是说：这篇文档是一个便条）。

程序清单 1.1 的第 3~6 行描述了根元素的 4 个子节点（`to`，`from`，`heading`，和 `body`）

程序清单 1.1 的最后一行是根元素的结束 `</note>`。

读者能够从这个 XML 文档中看出这是一个 fancy 留给 rainbow 的便条吗？难道能不承认 XML 是一个漂亮的自描述语言吗？

程序清单 1.1（`note.xml`）的运行效果如图 1.1 所示，服务器环境为 Tomcat 3.2.3+J2SE SDK 1.3.0+Windows XP Professional，客户端环境为 Internet Explorer 6.0。

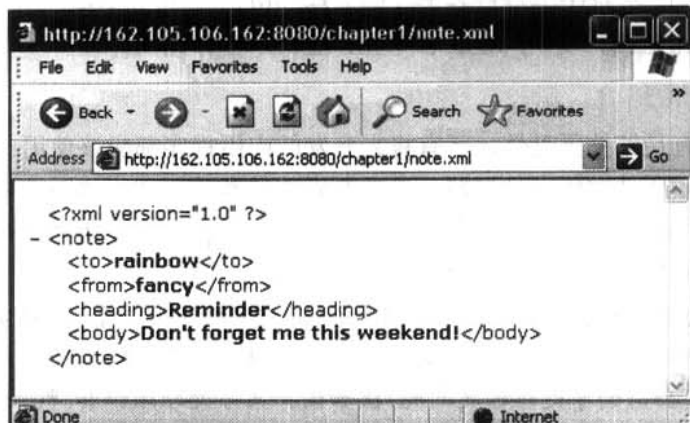


图 1.1 程序清单 1.1（`note.xml`）的运行效果

所有的 XML 元素必须有一个结束标记

在 XML 文档中，忽略元素的结束标记是不符合规定的。在 HTML 文件中，一些元素是可以没有结束标记的。例如下面的代码在 HTML 文件中是完全合法的。

```
<p>This is a paragraph
<p>This is another paragraph
```

但是在 XML 文档中，上述代码必须如下写。

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

注意 读者可能已经注意到了，在程序清单 1.1 中，第 1 行是 XML 声明，它并没有结束标记。这不是错误，因为此 XML 声明并不是 XML 文档的一部分，它不是一个 XML 元素，它也不应该有一个结束标记。

XML 标记是大小写敏感的

XML 和 HTML 不一样，XML 是大小写敏感的语言。在 XML 中，`<Letter>` 和 `<letter>` 是两个不同的标记。因此 XML 文档中，开始标记和结束标记的大小写样式必须保持一致。请看下面的代码。

```
<Message>This is incorrect</message>
<message>This is correct</message>
```

很明显，上面的第 1 行代码是不正确的，第 2 行代码才是正确的。

所有 XML 元素必须适当嵌套

XML 文档中不允许不正确的嵌套，也就是不能够有关系紊乱的嵌套。在 HTML 中，一些元素可以有不正确的嵌套，例如下面的代码可以被浏览器解释。

```
<b><i>This text is bold and italic</b></i>
```

在 XML 中，上述情况是绝对不允许发生的，XML 语法规规范要求正确地嵌套每一个 XML 元素，上面的代码应该如下改写。

```
<b><i>This text is bold and italic</i></b>
```

所有的 XML 文档必须有一个根元素

XML 文档中的第 1 个元素就是根元素。所有的 XML 文档都必须包含一个单独的标记对以定义根元素，所有其他的元素都必须在此标记对中嵌套。XML 文档只能有并且必须有一个根元素。所有的元素都可以拥有本身的子元素，子元素必须正确地嵌套在父元素之中。下面的 XML 代码片段可以对这些规则作出一个形象的解释。

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

属性值必须总是被引号引着

在 XML 中，元素的属性值没有引号引着是不符合规定的。像 HTML 元素一样，XML 元素同样可以拥有属性。XML 元素的属性以名字/值对的形式出现。XML 语法规规范要求 XML 元素属性的值必须使用引号引着。请看下面的两个例子，第 1 个例子是错误的，第 2 个例子是正确的。

程序清单 1.2 (incorr.xml)

```
<?xml version="1.0"?>
<note date=12/11/99>
  <to>rainbow</to>
  <from>fancy</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

程序清单 1.2 (incorr.xml) 的运行效果如图 1.2 所示，服务器环境为 Tomcat 3.2.3+J2SE SDK 1.3.0+Windows XP Professional，客户端环境为 Internet Explorer 6.0。