



开源魅力

面向Web开源技术整合开发与实战应用

马洪江 周相兵 余堃 等编著

- ✓ 结构完善、体系清晰
- ✓ 知识面广、内容翔实
- ✓ 原理、技术与实战完美结合
- ✓ 经验、体验整合的结晶

清华大学出版社





开源魅力

面向Web开源技术整合开发与实战应用

马洪江 周相兵 余堃 等编著

- ✓ 结构完善、体系清晰
- ✓ 知识面广、内容翔实
- ✓ 原理、技术与实战完善结合
- ✓ 经验、体验整合的结晶

清华大学出版社
北 京

内 容 简 介

本书不仅是一本 J2EE 入门图书,还详细地介绍了面向开源软件的构架原理、分析设计方法、开发方法、开发技术和众多当前流行的开源框架。重点分析介绍了 SSH(i) (Spring、Struts、Hibernate (和 iBatis)、A2J (Axis/CXF、WSDL2OWL、Jena) 和页面处理 (AJAX、Direct Web Remoting、Portlet (Jetspeed、Liferay) 等开源软件集成方法、基本应用方法和案例。书中也详细分析介绍了面向开源软件的软件开发模式,即用软件工程知识和软件开发方法将基于 Web 的开源软件集成在一起,并使这些开源软件松散耦合地组织在了一起。书中最后配备了两个实用性强的案例来进一步分析面向开源软件的软件开发方法。这两个案例都是以本书中介绍的开源软件为基础,以便读者更快速地掌握 Java EE (J2EE) 应用开发技术。本书配套的光盘内容包括了两个案例的源代码,也可作为读者掌握这一门软件开发方法的指导具体案例。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。
版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目 (CIP) 数据

开源魅力:面向 Web 开源技术整合开发与实战应用 / 马洪江,周相兵,余堃等编著. —北京:清华大学出版社,2013.4

ISBN 978-7-302-30926-0

I. ①开… II. ①马… ②周… ③余… III. ①软件开发 IV. ①TP311.52

中国版本图书馆 CIP 数据核字 (2012) 第 286736 号

责任编辑:夏兆彦
封面设计:柳晓春
责任校对:徐俊伟
责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:清华大学印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:190mm×260mm 印 张:44.5 字 数:1111 千字

版 次:2013 年 4 月第 1 版 印 次:2013 年 4 月第 1 次印刷

印 数:1~4000

定 价:79.00 元

前 言

近年来面向 Web 开发的开源件已在各行各业的软件领域得到了广泛的应用，这就使得面向开源软件的软件开发逐渐形成了一种新的软件开发方法。但怎样使其成为一种软件开发体系和方法是目前摆在众多软件研发人员面前的难题，因此编者根据多年的 Java EE (J2EE) 开发经验、对开源软件的深入研究成果，以及国内开源软件的相关文献和 developerWorks 知识来解决这个问题：本书全面介绍面向 Web 的开源件原理方法、技术构成、集成方法、开源件技术、应用体系和面向服务的软件技术；同时，将现阶段流行的、基于 Web 的开源件整合成一个中间件；并以该中间件为基础，举例了 Web 服务和语义 Web 服务的开发方法，并开发了一个科研统计分析系统。全书在撰写过程特别注重可读性、实用性、专业性、学术性、可操作性，并力求知识面广、难易合理搭配，从而让读者不用翻阅太多的资料就可以从事相关软件研究与开发工作。因此，本书首先从原理、方法、技术层面进行全面分析研究，然后从具体的开源件进行分析研究，最后总结面向开源件框架的轻量级软件开发方法和流程。

全书内容共分 9 章，分别如下：

- 第 1 章 开源软件发展的概况
- 第 2 章 面向开源软件的软件构架原理
- 第 3 章 面向开源软件的分析设计方法
- 第 4 章 面向开源软件的软件开发方法
- 第 5 章 面向开源软件的软件开发技术
- 第 6 章 面向开源软件的软件开发开源框架
- 第 7 章 多开源软件框架整合方法
- 第 8 章 SAJP-M 轻量级开源中间件整合实现
- 第 9 章 用 SAJP-M 设计实现科研绩效系统

本书的特色主要表现在：

1. 开源软件方法集成、开源软件技术集成

开源软件方法集成就是根据开源软件的发展概况，以软件工程思想知识来组织当前流行的、基于 Web 的开源软件，从而形成一种面向开源软件的软件开发方法。开源软件技术集成就是通过基于 XML 的配置文件来集成当前流行的、基于 Web 的开源软件，使其成为一种面向开源软件的软件开发中间件模式。

2. 经验丰富、针对性强、技术价值高、技术总结区分比较

编者有着丰富的 Java EE (J2EE) 软件开发经验，又从事软件开发相关课程教学，还有着长期从事开源软件技术研究工作的经历。因此，编者明白读者需要什么样的书籍，即需要理论联系技术，技术联系应用的体系化的书籍，也需要书中对同一类知识、技术进行对比分析。

3. 内容翔实、实用性强、层次逻辑分明、难易结合

本书介绍了面向开源软件的发展历程，详细分析了面向开源软件的软件构架原理、方法和技术，以及具体的应用案例；针对不同的开源软件列出了具体的应用方法和集成原理。严格按照 Java EE (J2EE) 及相关规范进行本书的知识布局，使得只要读者参考本书内容，就完全可以身临其境地感受企业实际的开发。

本书第 1、2 章由马洪江、周相兵编写，第 3 章由周相兵编写，第 4 章由马洪江、余堃编写，第 5 章由周相兵、马洪江编写，第 6、7、8、9 章由马洪江、周相兵、余堃共同完成编写，全书由马洪江统稿。

本书既可以作为 Java EE (J2EE) 初学者的书籍，也可以作为 Java EE (J2EE) 应用开发者的提高指导，还可以作为高年级的实践指导用书。

限于编者的水平和时间，在本书中难免存在错误，希望读者给予指导并与我们进行交流，以便本书的修订，使之更能符合读者的要求。

本书在成书过程中，得到了 IBM 软件设计师谢成锦、徐海的大力支持，也得到了成都理工大学苗放教授和电子科技大学国家级计算机实验教学示范中心有关人员的支持，在这里一并感谢他们。

最后，我们衷心感谢清华大学出版社的编辑为此书出版付出的努力和辛勤劳动。

编 者

2012 年 1 月

目 录

第 0 章 概论	1
第 1 章 开源软件发展的概况	7
1.1 开源软件的定义	7
1.1.1 自由软件定义	7
1.1.2 开源软件定义	9
1.1.3 中国开源软件推进联盟定义	12
1.2 开源软件状况	13
1.2.1 开源软件发展状况	14
1.2.2 开源软件应用状况	19
1.3 开源软件分类	23
1.4 开源软件的优点	26
1.5 开源软件的特点	27
1.5.1 开源软件的成本	28
1.5.2 开源软件的成熟度测评概况	30
1.5.3 开源软件的选择策略概述	32
1.5.4 开源软件的管理机制	34
1.5.5 开源软件与知识产权	35
1.6 最有价值的开源软件	40
小结	42
参考文献	43
第 2 章 面向开源软件的软件架构原理	44
2.1 软件构架概述及基本方法	44
2.1.1 软件构架的特点	47
2.1.2 软件构架的质量评估	48
2.1.3 软件架构“4+1”视图模型	54
2.1.4 软件构架师	59
2.1.5 案例分析——档案管理系统	62
2.2 基本的软件构架方法	66
2.2.1 软件体系结构论述	67
2.2.2 软件层次结构	97
2.2.3 软件中间件构架方法	106

2.2.4	轻量级的软件构架方法	107
2.3	可信软件的构架方法	116
2.3.1	可信软件概述	116
2.3.2	可信软件基本原理	119
2.3.3	可信软件构造所满足的基本条件	122
2.3.4	可信软件演化	123
2.3.5	可信软件度量	125
2.3.6	可信软件技术	126
2.3.7	可信研究进展	128
2.4	协同软件构架方法	130
2.4.1	协同软件概述	131
2.4.2	协同软件原理	136
2.4.3	协同软件模式	137
2.4.4	协同软件角色	145
2.4.5	协同软件的工作流技术	146
2.5	开源软件的软件开发构架模式	147
	小结	150
	参考文献	150

第 3 章	面向开源软件的分析设计方法	154
3.1	开源软件分析设计方法概述	154
3.2	基本的软件分析设计方法	155
3.2.1	面向对象设计方法	156
3.2.2	面向构件设计方法	166
3.2.3	UML 建模方法	173
3.3	面向服务计算的软件分析设计方法	202
3.3.1	面向服务的分析设计方法概述	202
3.3.2	面向服务体系结构的设计方法	204
3.3.3	面向服务流程的建模方法	220
3.3.4	面向企业服务总线的方法	239
3.3.5	面向服务体系架构建模语言 (SoaML)	250
3.4	面向服务的软件语义化的软件分析设计方法	258
3.4.1	面向服务的软件语义化概述	260
3.4.2	OWL-S 与 WSDL、TMDM 的特征关系	263
3.4.3	OWL 与 Web 服务、主题图的关系	269
3.4.4	面向服务软件语义化基础	270
3.4.5	面向服务的软件语义化方法	274
3.4.6	面向服务的软件语义化研究进展	280

3.4.7 面向服务的软件语义的软件分析设计方法	284
小结	285
参考文献	285
第4章 面向开源软件的软件开发方法	289
4.1 面向开源软件的软件开发特点	289
4.1.1 软件体系架构选择原则	289
4.1.2 面向开源软件的软件开发的代码原则	289
4.1.3 开源软件选择方法分析	290
4.2 面向开源软件的软件开发方法	293
4.2.1 开发模型分析	293
4.2.2 开发需求分析	294
4.2.3 开发分析设计方法	294
4.2.4 开发实现流程	295
4.2.5 测试方法	295
4.3 面向开源软件的软件开发标准探索	296
4.3.1 软件可信性	296
4.3.2 软件质量	298
4.3.3 软件复用	299
4.3.4 软件再生	299
4.3.5 软件自动化	300
4.3.6 软件验证与确认	300
小结	301
参考文献	301
第5章 面向开源软件的软件开发技术	302
5.1 概述	302
5.2 常用的开发及平台语言	302
5.2.1 PHP	302
5.2.2 Perl	314
5.2.3 Flex	314
5.2.4 Harmony	321
5.2.5 JSP	324
5.2.6 Android	327
5.3 常用的开发环境	330
5.3.1 Eclipse	330
5.3.2 CVS	333
5.3.3 NetBeans	334

5.3.4	Apache Ant	334
5.3.5	JUnit	335
5.4	常用的支持服务器软件	336
5.4.1	Tomcat	336
5.4.2	Geronimo	343
5.4.3	Jboss	346
5.4.4	Jetty	346
5.4.5	Derby	349
5.5	Web 2.0 技术	350
5.5.1	Web 2.0 实现的相关技术	351
5.5.2	Web 2.0 用户界面定制工具	353
5.5.3	Web 2.0 页面处理技术	355
5.5.4	RSS 技术	357
5.6	面向服务的软件开技术	360
5.6.1	Web 服务技术	360
5.6.2	语义 Web 服务技术	387
5.6.3	RESTful Web 服务技术	388
5.6.4	SOA 技术	391
5.6.5	BPEL4WS	403
5.7	语义描述语言	404
5.7.1	RDF	404
5.7.2	OWL-S	411
5.7.3	WSMO	418
5.8	数据库访问技术	424
5.8.1	ODBC	424
5.8.2	JDBC	426
5.8.3	ADO.NET	431
5.8.4	pureXML	432
小结		434
第 6 章	面向开源软件的软件开发开源框架	435
6.1	概述	435
6.2	DWR	436
6.2.1	AJAX 基本应用方法	436
6.2.2	DWR 应用方法	448
6.3	Portlet	451
6.3.1	容器	454
6.3.2	页面处理	455

6.3.3	Jetspeed	461
6.4	iweb SNS	468
6.5	Struts	469
6.5.1	MVC	469
6.5.2	Struts 应用方法	474
6.6	Spring	483
6.6.1	Spring 框架介绍	483
6.6.2	AOP	490
6.6.3	IoC	495
6.6.4	Spring3 在构建 RESTful Web Services 的方法	501
6.7	数据持久化框架	504
6.7.1	Hibernate	504
6.7.2	Hibernate 应用方法	513
6.7.3	iBatis 应用方法	517
6.8	A2JT	522
6.8.1	A2JT 介绍	522
6.8.2	Web 服务框架: Axis、CXF	523
6.8.3	服务功能语义转换: WSDL2OWL-S	536
6.8.4	语义推理: Jena	536
6.8.5	本体编辑工具: Protégé	547
6.8.6	WSMO 编辑工具: WSMO Studio	549
6.8.7	SOA 框架: Tuscany	550
6.9	数据处理框架	562
6.9.1	开源搜索框架 Lucene	563
6.9.2	多源数据抽取框架	575
	小结	580
第 7 章	多开源软件框架整合方法	581
7.1	概述	581
7.2	PP: 面向 AJAX 的 DWR 与 Jetspeed 整合	581
7.2.1	配置 web.xml 格式	581
7.2.2	配置 dwr.xml 格式	582
7.2.3	配置 portlet.xml 格式	582
7.3	SSH 整合	585
7.3.1	概述	585
7.3.2	Struts 与 Spring 整合	585
7.3.3	Struts 与 PP 整合	593
7.3.4	Spring 与 Hibernate 整合	594

第 0 章 概 论

自从 WWW(World Wide Web, 简称为 Web) 在 20 世纪 90 年代初正式由万维网联盟[World Wide Web Consortium, W3C, 其发明者是美国 Timothy John Berners-Lee (简称为 Berners-Lee, 现任 W3C 主席), 他早在 20 世纪 80 年代初就给出了建议和原型。]发布以来, 使各行各业的信息化建设和发展逐渐发生了本质性的变化, 这主要表现在:

- ① 使信息化建设真正走向网络化发展;
- ② 通过 Web 浏览器给用户操作带来了便捷性;
- ③ 信息发布与获取更加简单、容易;
- ④ 改变了传统的信息化建设模式, 加快了信息化建设速度;
- ⑤ 促进了网络技术快速、飞猛的发展, 特别是软件技术的发展, 使得了软件技术向多元化、多样化方向发展;
- ⑥ 催生了基于 Web 的新生网络应用, 如论坛 (BBS)、社交网络 (SNS)、博客 (BLOG)、微博 (Micro Blog)、电子商务及与电子相关的新兴产业等;
- ⑦ 促使信息化系统的网络结构发生了转变, 由最初的客户/服务 (Client/Server, C/S) 向单一的浏览器/服务器 (Browser/Server, B/S) 模式转变, 以及向多层次、多结构 (如点对点 (Peer-to-Peer, P2P) 结构等) 的 B/S 的分布式系统转变, 改变了对传统网络信息系统的认识;
- ⑧ 促进对信息化的认识从最初的以业务为中心的到当前以需求为中心方向转变;
- ⑨ 促使了信息化的网络计算模式发生了变革, 由最初基本网格计算模式向并行计算、服务计算、网格计算、云计算和绿色计算方向发展, 以及向多级别、多结构、多集群、多端、多层次、多需求等综合性的分布式计算;
- ⑩ 使得网络信息化应用向自由和共享角度发展, 特别对软件技术影响起到了非常重要的作用, 最明显的就是催生开源软件 (Open Source Software, OSS) 的发展与进步。

随着 Web 技术的发展, 促进面向 Web 的开源软件的快速发展, 也使得开源软件在软件研发过程中占有重要的地位; 使得出现了一大批可用度高、质量好、健壮性稳定的开源软件, 以及在各行各业信息化建设中起到关键作用的大量优秀的各类型的开源软件, 最具有代表的就是以 Sun 公司 (现已被 ORACLE 收购) 的 Java 为基础语言的开源软件阵营, 她目前在信息化建设领域占很大的比重, 最为成熟就是 J2EE (Java 2 Platform Enterprise Edition) 为框架的结构, 大多数的开源软件都是以 J2EE 中的 JDK (Java Development Kit) 为基础进行扩展、深化和开发的; 又特别是大多数基于 Web 技术开源软件的都是以 J2EE 为基础的, 包括正在发展的面向服务和面向智能分析设计的信息化系统, 其中许多相关开源软件已陆续问世。

开源软件的发展离不开 W3C、结构化资讯标准促进组织 (Organization for the Advancement of Structured Information Standards, ORSIS) 组织所提供的相关标准和技术的支持, 这两个非营利组织是目前基于 Web 的开源软件的重要支撑, 它为开源软件的应用、兼容和扩展提供了统一的、规范的说明和实践方法; 同时, 开源软件的持续和健康发展也离不开

这两个组织继续修改、更新和推出适应 Web 技术发展的标准和规范。

1. WWW 发展概况

而 WWW (Web) 的核心部分由统一资源标识符 (Uniform Resource Identifier, URI)、超文本传送协议 (HyperText Transfer Protocol, HTTP)、超文本置标语言 (Hypertext Markup Language, THML; 习称超文本标记语言) 这三个基本标准组成。

1) URI、URL、URN

URI 是一个用于标识某一互联网资源名称的字符串, 并允许用户对网络中的资源通过协议进行交互操作, 由存放资源的主机名、片段标志符和相结 URI 组成。即 URL 是标识一个互联网资源, 并指定对其进行操作或取得该资源的方法的 URI,

URL 是因特网上标准的资源的地址, 可以认为 URL 是 URI 的一个变种, 或是命名机制的一个子集。URI 是确定一个因特网资源, 而 URL 不但确定一个资源, 还表示这个资源在哪里, 因此, 采用 URL 可以用一种统一的格式来描述文件、服务器的地址和目录等资源。URL 由协议、主机 IP 地址和主机资源的具体地址三部分组成。

URN 即统一资源名称 (Uniform Resource Name, URN) 如同一个人的名称, URL 就代表一个人所在的地址, URN 则定义某事物的身份, 不依赖资源所处的位置, 且有可能减少失效连接的个数; 而 URL 提供查找该事物的方法。因此, URL 和 URN 有着互补的作用。

2) HTTP、HTTPS

HTTPS 即超文本传输协议安全 (Hypertext Transfer Protocol over Secure Socket Layer, HTTPS) 是 HTTP 和安全套接层协议层 (Secure Socket Layer, SSL)/传输层安全 (Transport Layer Security, TLS) 的组合, 用以提供通信加密及对网络服务器身份的鉴定, 它常用于因特网 (Internet) 上的交易支付和企业信息系统中敏感信息通信。即在 HTTP 层下加 SSL/TLS 层, 因此, HTTPS 的安全基础是 SSL, 但对上层协议一无所知, 所以 SSL 服务器只能为一个 IP 地址/端口组合提供一个证书^①。HTTPS 和 SSL 支持使用 X.509 (是公钥证书、证书吊销清单、属性证书和证书路径验证算法等证书标准) 数字认证, 也可以认为 HTTPS 是建立一种安全信息通道, 以及另一种确认网站的真实性。而 HTTP 与 HTTPS 的区别主要有:

① HTTP 的 URL 表示为 "http://", 使用的默认端口是 80; HTTPS 的 URL 表示为 "https://", 使用的默认端口是 443。

② HTTP 是明文传输, 因此不安全, 攻击者容易通过 “监听” 和 “中间人攻击” 等手段获取网站账户和敏感信息等。HTTPS 具有安全性的 SSL 加密传输协议, 被设计为可防止 HTTP 所带来的攻击, 并认为是安全可靠的; 但 HTTPS 并不能防止站点被网络蜘蛛抓取; 在某些情形中, 被加密资源的 URL 可仅通过截获请求和响应的大小推得^②。

③ HTTPS 中的 TLS 有简单和交互两种策略, 交互策略更为安全, 但需要用户在他们的浏览器中安装一个 CA (Certificate Authority) 来进行认证, 而且需要交费。

④ HTTP 属于简单连接、无状态的; 但 HTTPS 协议是由 SSL/TLS+HTTP 协议构建的可进行加密传输、身份认证的网络协议, 因此 HTTPS 比 HTTP 协议更安全。

① Apache FAQ: Why can't I use SSL with name-based/non-IP-based virtual hosts? http://httpd.apache.org/docs/2.0/ssl/ssl_faq.html#vhosts

② Pusep, Stanislaw. The Pirate Bay un-SSL. 2008-7-31. <http://sysd.org/stas/node/220>

但 HTTPS 与安全超文本传输协议 (Secure Hypertext Transfer Protocol, S-HTTP) 是有区别的, S-HTTP 是一种面向安全信息通信的协议, S-HTTP 定义于 RFC 2260^①中。在语法上, S-HTTP 报文与 HTTP 相同, 可以与 HTTP 结合使用, 能与 HTTP 信息模型共存并易于与 HTTP 应用程序相整合, 是一个 HTTPS URI Scheme 的可选方案。因此, 在市场上, HTTPS 远胜于 S-HTTP。

3) HTML、SHTML

当前使用的 SHTML (Server Side Include HyperText Markup Language) 是通过 SSI (Server Side Include) 扩展 HTML 文件名, 通常称为服务端嵌入, 是为 Web 服务提供一套可以直接嵌入到 HTML 中的命令, 当浏览器端访问这些 SHTML 文件时, 服务器端会把这些 SHTML 文件进行读取和解释, 并把 SHTML 文件中包含的 SSI 指令解释出来, 因此也可以认为 SHTML 是服务器动态产生的 HTML, 它的程序格式为 `<!--#指令名称="指令参数">`, 也可以认为是一种类似于 ASP (Active Server Page) 的基于服务器的网页制作技术 (相信熟悉 ASP 的读者, 就非常容易理解和掌握 SHTML 的应用方法)。通常情况下, 使用 SHTML 时, 需要明确如下 SSI 基本指令格式和意义:

① 将浏览器能识别的文档内容直接插入到动态文档中 `<#include>`, 例: `<!--#include file="test.html"-->`。

② 显示服务器端环境变量 `<#echo>`, 例: `<!--#echo var="REMOTE_ADDR"-->`, 表示显示获取浏览器用户的 IP 地址。

③ 直接在服务器上执行各种程序 `<#exec>`, 例: `<!--#exec cgi="/cgi-bin/test.cgi"-->`, 表示将执行 CGI 程序 test; `<!--#exec cmd="dir /b"-->`, 表示将会显示当前目录下文件列表。其中 `"#exec cgi="` 用来专门执行 CGI 程序, `"#exec cmd="` 用来执行 shell 命令。

④ 设置 SSI 信息显示格式 `<#config>`, 高级 SSI `<XSSI>` 可设置变量和使用 if 条件语句。其中 XSSI (Extended SSI) 是一组高级 SSI 指令, 内置于 Apache 1.2 或更高版本的 mod-include 模块中, 一般可以使用的指令有 `#printenv`、`#set`、`#if`; 其中:

`#printenv` 表示显示当前存在于 Web 服务器环境中的所有环境变量, 例: 直接使用 `<!--#printenv-->` 就可以了。

`#set` 表示给变量赋值, 以用于 if 语句, 例: `<!--#set var="test" value="测试"-->`。

`#if` 表示创建可以改变数据的页面, 这些数据根据使用 if 语句时计算的要求予以显示 (最重要的目的之一就是用来生成动态的 HTML)。例:

```
<!--#if expr="$test=\"this is a test\""-->
这是一个测试;
<!--#elif expr="$test=\"this is not a test\" -->
这不是一个测试;
<!--#else-->
是一个测试吗?
<!--#endif"-->
```

① E. Rescorla, A. Schiffman. The Secure HyperText Transfer Protocol (RFC 2660), <http://tools.ietf.org/html/rfc2660>

⑤ `<#flastmod file>`显示 Web 文档的 file 所指文件最近更新日期，表示把当前目录下的网页页面插入到当前页面，例：`<!--#flastmod file="test.html"-->`，其中 file 表示指定页面相对于 test.html 的位置。

⑥ `<##fsize file>`显示 Web 文档的 file 所指文件的长度，表示将当前目录下网页页面的文件大小入到当前页面，例：`<!--#flastmod file="test.html"-->`。

Web 服务器在处理网页的同时处理 SSI 指令，当 Web 服务器遇到 SSI 指令时，直接将包含文件的内容插入 HTML 网页。当在使用 SSI 将内容发送到浏览器之前，可以使用 SSI 指令将文本、图形或应用程序信息包含到网页中，所以对 Web 网站结构固定且需要维护时，把一些有限的 HTML 模块放在 Web 服务器上，然后把需要更新的网站信息传到服务器就可以方便、简单、轻松地实现网页更新，自动生成网页，这样会有效提高大型网站管理效率。

而 SHTML 与 HTML 的区别有以下两点：

① SHTML 是一种 Web 服务器 API (Application Program Interface)，而 HMTL 不是。

② SHTML 是 Web 服务器动态生成的 HTML，是一种用 SSI 技术的超文本书件格式；而 HTML 是一种静态页面格式，没有在服务端执行脚本。

4) THML、XML

可扩展置标语言 (Extensible Markup Language, XML) 是一种跨平台的、依赖于内容、处理半结构化的一种处理置标语言，与 HMTL 同属于标准通用置标语言 (Standard Generalized Markup Language, SGML)，被认为 XML 是对 HTML 一种扩展，这是由于：

① HTML 不能解决所有解释数据的问题、难以扩充、不具备弹性、易读性不好、处理效能不佳。

② XML 通常用来传送及携带数据信息，不用来表现或展示数据，HTML 语言则用来表现数据，XML 也有严格的语法结构。

③ XML 扩展性比 HTML 强，XML 的语法比 HTML 严格，XML 与 HTML 可以互补。

同时，XML 在传送、携带数据信息时，具备严格的语法判定、认识能力，而且具备处理这些数据信息的能力，如数据索引、排序、查找、相关一致性检查等，这些工具包括 XPath、Xquery 等，而 XML 编辑器主要有 XML Notepad、XML Spy、Xeena 等。XML 的文档对象模型 (Document Object Model, DOM) 将 XML 文档作为一个树形结构，而树叶被定义为节点来组织这些信息，因此具有良好的层次结构和树结构的特性，并通过文档类型定义 (Document Type Definition, DTD) 和 Schema (XML 模式) 进行解析；利用可扩展样式表转换语言 (EXtensible Stylesheet Language Transformation, XSLT) 来操纵 XML 数据，他将 XML 文档转换为不同 XML 结构的文档，甚至还可以转换为非 XML 文档。因此，XML 相比 HTML 具备以下优势：

① XML 具备跨平台、跨语言、跨结构的能力，即可以搜索、查找不同平台、不同结构、不同语言的数据信息。

② XML 可以用于存储数据，使结构化的数据表示为半结构化、非结构化数据，这样便于不同的数据源或不同请求访问。

③ XML 可以用于共享数据，由于 XML 是以纯文本的形式存储数据，因此与软件、硬件的状态无关，使得不同的需求源可以使用 XML。

④ XML 可以用于交换、利用数据，由于 XML 自由的文本格式，可以作为数据交换的中

间过渡介质,使不同的数据格式间相互转换,从而提高数据的使用能力。

⑤ XML 用于多语言、多平台融合和部署配置,利用 XML 可以方便的将要使用的多种开发语言,支持平台融合在一起进行使用,并能将所开发的应用程序进行部署。

⑥ XML 可以从 HTML 中分离数据,确保数据改动时不会导致 HTML 文件也需要改动,因此大大降低了网页维护难度。

⑦ 通过 XML 可以与电子商务、各类企业系统、各类商业系统、各类信息化集成系统、电子政务等进行数据信息通信,从而提高各系统使用效率。

⑧ XML 可以作为创建其他新语言,如 XML 就是 WAP 和 WML 语言的基础。

2. Web 技术概况

Web 技术是实现网页页面的技术,当然也包括前面所述的 HTTP、HTTPS、SHTML、XML 等技术。Web 是一种典型的分布式应用结构,因此每一次数据信息交换都要涉及客户端和服务端。当前,实现 Web 的技术包含了许多种类型,如 HTTP、XHTML、CSS、JavaScript、DOM、Java servlet、XML 及相关技术、JSP、SOAP、Web 服务、JAX-RPC、CGI、ASP、SQL、ASPX、PHP 和 ColdFusion 技术等。其中:

① 有包括 W3C 标准的 HTML、XHTML、CSS、XML 等;

② 客户端的 HTML 语言、XML 语言、Java Applets、脚本程序、CSS、DHTML、XHTML、插件技术以及 VRML 技术等;

③ 服务端的 CGI、PHP、ASP、ASP.NET、Servlet 和 JSP 技术等。

3. 语义 Web 技术概况

语义 Web (Semantic Web) 在 2001 年由 Berners-Lee、Hendler 和 Lassila 三人正式提出(早在 1998 年, Berners-Lee 就提出了语义 Web 的概念),它是对 Web 的扩展,是计算机业和互联网业对网络下一阶段发展所作出的术语化定义;它的核心是通过给 WWW 上的文档添加能够被计算机所理解的语义,从而使整个互联网成为一个通用的信息交换媒介;它的关键技术是资源描述框架 (Resource Description Framework, RDF)、XML、本体 (Ontology),如图 0 所示。

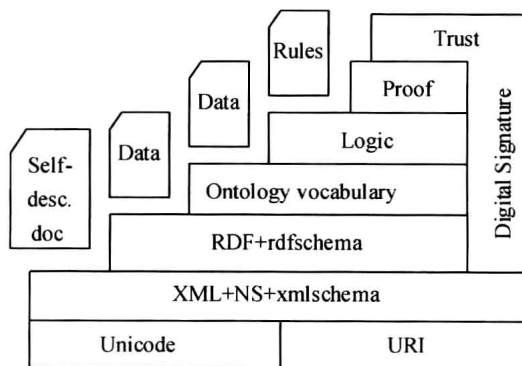


图 0 语义 Web 结构图 (In W3C)

与此同时, Berners-Lee 在 2000 年也正式提出了七层语义 Web 层次结构(又称语义 Web 堆栈),并指出语义 Web 是建立在 XML 之上的,从而使 Web 成为语法与语义两重标准的智能化网成为可能,并能使计算机能判断和识别数据信息。其主要思想是 Web 与语义, Web 主

要包括资源静态映射的 URI、创立可导航的空间、通信交互的共享空间和自描述文档，语义主要包括机器可以认识、希望得到什么信息、信息特征间怎样转换和表达方式。在语义 Web 技术上主要体现在以下几个方面：

① 当前多种 WWW 上的技术都能适应用语义 Web 中，这为语义 Web 发展奠定了基础；

② 在语义 Web 中，需要得到基础的国际资源标识符(Internationalized Resource Identifier, IRI)、Unicode、XML 和 XML 命名空间等的支持。其中，IRI 作为 URI 的泛化形式，提供对语义网资源加以唯一标识的手段；Unicode 表示采用多种语言来表现和处理文字，XML 命名空间是作为语义 Web 连接的访问方式。

③ 当前支持语义 Web 的技术包括 RDF、RDF Schema (RDFS)、网络本体语言 (Web Ontology Language, OWL)、SPARQL、规则交换格式 (Rule Interchange Format, RIF) 等。其中，RDFS 是 RDF 的基础词汇表；OWL 是对 RDF 的扩展，用来描述、声明 RDF 的语义构造，是由描述逻辑 (Description Logic) 为基础的，因此，具备传递性、判定性等特征，且具有推理的能力，为丰富语义 Web 提供前所未有的帮助；SPARQL 是一种 RDF 查询语言，用于查询任何基于 RDF 的数据。

④ 而对于语义 Web 的用户界面、信任、验证和加密等，还处于发展中。

综上，对面向 Web 的一些基础知识进行了概述、比较，为面向 Web 开源技术整合与实践应用提供一定的基本知识一览，为快速进行本书学习提供预备知识。

第 1 章 开源软件发展的概况

当前，开源软件的应用已经扩展到诸多行业，对推动软件业迅速发展起到了不可估量的作用，这也体现了软件的自由和共享宗旨；软件开发者只需要根据开源软件的开发社区可以方便获得应有的疑惑、问题，方便改善自己的编程风格，以及根据版权持有人在软件协议的规定之下保留一部分权利并允许用户学习、修改、增进提高这款软件的质量、软件的健壮性和可用性；当然对于开源软件所组成的源代码开放，并不一定是开源软件，它只是一种共享形式，一种可以任意获取的计算机软件。

1.1 开源软件的定义

在介绍开源软件定义之前，需要回顾一下开源软件的基础自由软件，也就是说，开源软件的产生是由自由软件衍生而来。同时，也简要介绍中国开源软件推进联盟。图 1-1 所示是自由软件与非自由软件的分类结构图。

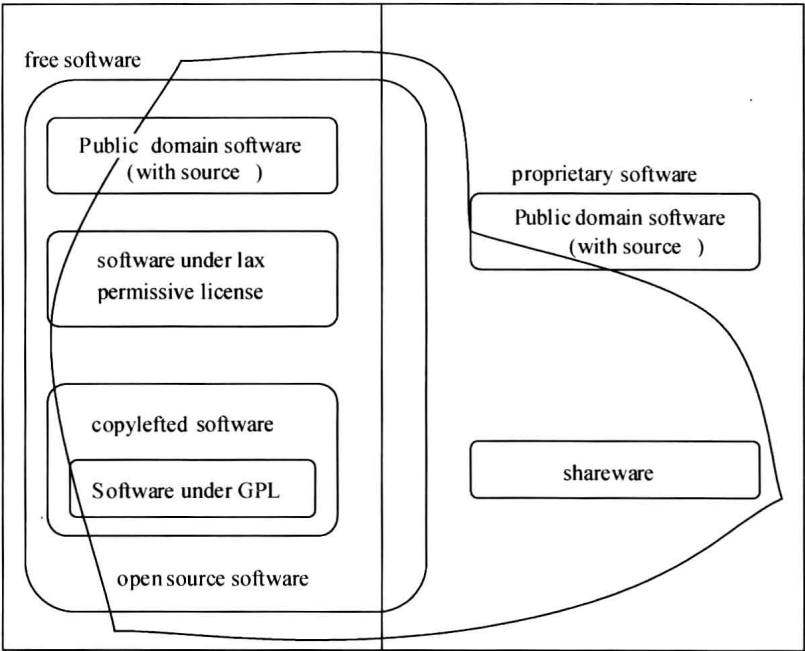


图 1-1 自由软件与非自由软件分类结构图
(来源: <http://www.gnu.org/philosophy/categories.html>)

1.1.1 自由软件定义

自由软件（Free Software）是由自由软件基金会（Free Software Foundation, FSF）定义