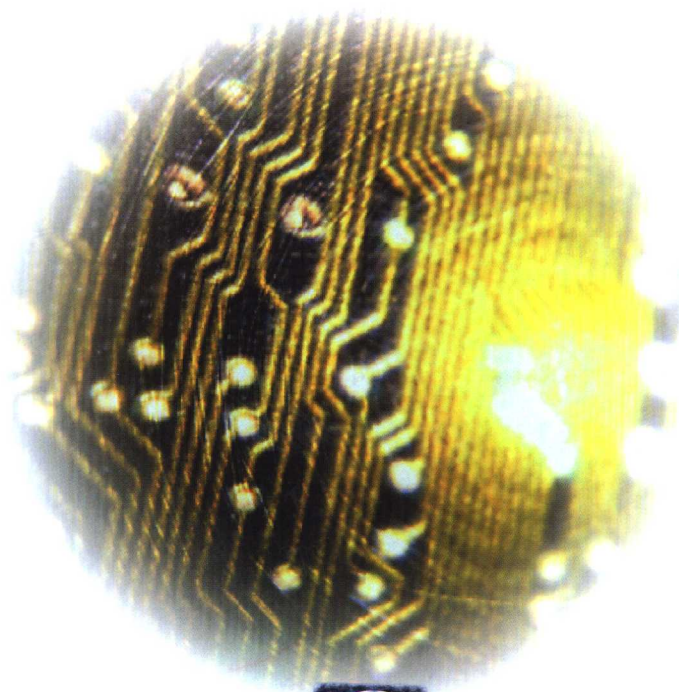


Java Web Services Programming

Java

Web 服务编程

(美) Rashim Mogha 著
V. V. Preetham 译
刘 凌



清华大学出版社

Java Web 服务编程

(美) Rashim Mogha 著
V. V. Preetham
刘 凌 译

清华大学出版社

北 京

EISBN:0-7645-4952-9

Rashim Mogha, V. V. Preetham

Java Web Services Programming

Copyright © 2002 by Wiley Publishing, Inc.

Original English language edition Published by Wiley publishing, Inc.

All Rights Reserved.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾地区)出版、发行。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,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

北京市版权局著作权合同登记号 图字:01-2002-6320

图书在版编目(CIP)数据

Java Web 服务编程/(美)莫哈(Mogha, R.), (美)普林西姆(Preetham, V. V.)著;刘凌译.

—北京:清华大学出版社,2003

书名原文:Java Web Services Programming

ISBN 7-302-07342-2

I. J… II. ①莫… ②普… ③刘… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 089129 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:曹康

文稿编辑:陈宗斌

封面设计:康博

版式设计:康博

印 刷 者:清华大学印刷厂

装 订 者:三河市化甲屯小学装订二厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:26.25 字数:672 千字

版 次:2003 年 11 月第 1 版 2003 年 11 月第 1 次印刷

书 号:ISBN 7-302-07342-2/TP·5332

印 数:1~4000

定 价:52.00 元

前 言

在现今的商务方案中，兼并和收购看起来是一种潮流。这一情况就产生了对能够满足企业业务需求的分布式 Web 应用程序的需要。过去，分布式技术行业的很多人都致力于提供一个有效的框架来把那些由基于组件应用程序构成的完全不同的系统集成起来。但是，这样的系统太脆弱了，很难变更或太昂贵，以致难以集成。这些系统是非常巨大的单片集成电路代码基，修改和变更都非常困难。其体系结构是固定的，而且可能是来自单一的供应商。这样的系统有 CORBA、分布式 SmallTalk 和 Java RMI。要用在跨越 Internet 的动态电子商务框架中，这些系统的效率是不够的。CORBA 和 RMI 的解决方案需要严格遵从所有者的标准或是严格遵守数据结构和完全不同的系统间的共享语境。由于这些约束，业界想出了一个更好的解决方案——一种提供松散耦合的体系结构，可以重用并且适合更改和扩展。Web 服务可能是最能满足这些需求的技术。

事实上，Web 服务是不同服务的最佳协作，由松散耦合的独立系统组成，也就是由基于组件的模块式应用程序框架组成。Web 服务为我们提供了一组操作并且通过采用标准接口使这组操作可用，这些标准接口是由企业提供的服务，它们被充分描述并且发布到服务目录中。分布式客户程序可以很容易定位这样的服务并且采用基于 XML 的标准化格式通过 Web 进行远程调用。

更具体地讲，Web 服务是一组组成技术堆栈的几种相关联的技术。注册和声明一项 Web 服务要用到下列协议：

- SOAP
- WSDL
- WSFL
- UDDI
- ebXML

Java 平台以其灵活性、可扩展性、可靠性和一次编写，随处运行的特点成为企业用于开发可扩展 Web 服务的事实上的标准。500 多万注册到 Sun 公司网址的下载已经表明将 Java 平台建立起作为 Web 开发者优先选择的编程语言。为了进一步减轻开发者的工作，Sun Microsystems 已经引入它自己的 Java Web 服务开发包。

Web 服务开发包是一套 Sun Microsystems 和其他 Java 合作伙伴开发的应用程序编程接口 (Application Programming Interface, API) 和体系结构。这些 API 和体系结构为重要的 Web 服务特征提供了 Java 支持。本书就是介绍为 Web 开发者提供创建 Web 服务的深入知识。本书讨论了 Web 服务技术堆栈(包括 UDDI、SOAP 和 WSDL)，Java Web 服务体系结构和 XML 的 Java API 的完整程序组。本书适用于所有想学习创建 Java Web 服务的人员，同时还是已经创建 Web 服务并且想要用 Web 服务包在其工作中提高效率的开发者的理想选择。

本书结构

本书的基本目标就是培训开发者掌握创建 Java Web 服务所必需的知识。本书由 Web 服务的基础开始,在读者充分了解 Web 服务的概念后,再讨论 Web 服务技术堆栈和 Java Web 服务的体系结构。

本书分为如下 4 个部分:

第 1 部分 Web 服务简介

第 1 部分给出了 Web 服务的概述。讨论了对 Web 服务的需求,深入详细地讨论了面向服务的体系结构(service-oriented architecture, SOA),并描述了 SOA 如何适应于当前的电子商务模式。这一部分还讨论了 Web 服务模式与业界现有其他模式相比的优点。

第 2 部分 Web 服务技术堆栈

这一部分讨论了 Web 服务技术堆栈各种各样的组件,如 XML、WSDL、SOAP 和 UDDI。这一部分详细讨论了如何实现这些 Web 服务技术堆栈组件,并且给出了作为帮助的示例。

第 3 部分 Java Web 服务的体系结构

第 2 部分由 Java Server Pages(JSP)和 servlet 的简介开始,然后讨论了 Web 服务中 J2EE 的角色。在全面介绍了 Java Web 服务体系结构的基础知识后,讨论了下列用于访问 Web 服务的 API:

- XML 处理的 Java API(Java API for XML Processing, JAXP)
- XML 绑定的 Java API(Java API for XML Binding, JAXB)
- XML 通信的 Java API(Java API for XML Messaging, JAXM)
- 基于 XML 的 RPC 的 Java API(Java API for XML-based RPC, JAX-RPC)
- XML 注册的 Java API(Java API for XML Registries, JAXR)

第 4 部分 附录

这一部分包括供读者快速参考的附录和一个关于 Java WSDP 的附录。Java WSDP 附录阐述了如何使用 WSDP 创建 Web 应用程序。

辅助 Web 网址

本书提供了一个辅助 Web 站点,从中可以下载各个章节的代码。所有代码列表都以 zip 文件形式放在 www.wiley.com/extras 的 Java Web Services Programming 链接下。如果当前您没有 Winzip,可以从 www.winzip.com 下载一个测试版本。

致谢

在此对所有在本书成书过程中做出贡献的 NIIT 和 Wiley 出版公司的职员表示感谢。我要特别感谢 NIIT 公司的项目经理 Anita Sastry, 没有她一贯的支持, 就没有这本书。同样也要特别感谢 Simanta 的帮助。还要感谢 Shweta、Shilpa 和所有在我最需要的时候给我坚定支持的亲爱的朋友们。我还要感谢我的合作者 V. V. Preetham, 感谢他的合作。

我还要特别感谢组稿编辑 Christopher K. Webb 和项目编辑 Sharon Nash 给我这个机会让我编写这本书。也要特别感谢技术编辑 David Wall 和高级文稿编辑 Bill Barton 有价值的录入工作和一贯支持。

——Rashim Mogha

我非常高兴有机会感谢所有参与创作这本书的每个人: NIIT 公司的项目经理 Anita Sastry、制图小组和技术编辑们。Wiley 出版公司的组稿编辑 Christopher K. Webb、项目编辑 Sharon Nash、技术编辑 David Wall 和高级文稿编辑 Bill Barton, 感谢他们的支持和鼓励。

我还要借此机会感谢所有直接或是间接对编写本书做出贡献的人们。

——V. V. Preetham

关于作者

Rashim Mogha 是微软认证的解决方案开发人员(Microsoft Certified Solution Developer, MCSD)。Rashim 曾从事多种技术工作, 如 ACT! 2000、SQL Server、Goldmine、A+认证、Cisco 和 Forte for Java。她的工作涉及教师引导培训课程的设计、开发、测试和实现。她的首要职责包括培训开发经理和项目管理人, 进行教学评审、技术评审和保证 ISO 的一致性。她喜欢写作。

V. V. Preetham 为 J2EE 规范和技术的组织提供咨询服务, 例如, 关于 WebLogic 的 EJB、Java、Tomcat 和 XML。他在运用多种 Java API、Java 2 技术、BEA WebLogic Server 和 J2EE 核心模式方面有着广泛的经验。

目 录

第 1 部分 Web 服务简介

第 1 章 Web 服务的体系结构	1
1.1 Web 服务概述	1
1.2 面向服务的体系结构	2
1.2.1 服务的抽象	2
1.2.2 服务的封装	3
1.2.3 服务的模块化	4
1.2.4 服务的多态性	4
1.3 服务的角色	5
1.3.1 服务提供商	5
1.3.2 服务代理商	6
1.3.3 服务请求者	7
1.4 技术堆栈	7
1.5 体系结构的过程	9
1.5.1 描述	10
1.5.2 发现	11
1.5.3 调用	13
1.6 价值链	13
1.6.1 标准体	15
1.6.2 框架提供商	15
1.6.3 工具和产品供应商	15
1.6.4 Web 服务开发者	15
1.6.5 Web 服务销售者	15
1.6.6 Web 服务提供商	15
1.6.7 Web 服务消费者	15
1.7 小结	16

第 2 部分 Web 服务的技术堆栈

第 2 章 XML	17
2.1 XML 简介	17
2.2 XML 基础	18

2.2.1 合适的 XML 文档——规则	20
2.2.2 XML 文档的组件	23
2.3 XML 命名空间	33
2.4 小结	37
第 3 章 WSDL	38
3.1 WSDL 的起源	38
3.2 WSDL 简介	40
3.2.1 WSDL 的结构	40
3.2.2 WSDL 的示例	41
3.3 服务抽象的定义	43
3.3.1 WSDL 的语法	43
3.3.2 WSDL 的元素	45
3.3.3 绑定类型	54
3.3.4 MIME 绑定	58
3.4 小结	59
第 4 章 SOAP	60
4.1 SOAP 简介	60
4.1.1 SOAP 的设计目标	61
4.1.2 符号约定	62
4.1.3 SOAP 的示例	62
4.2 SOAP 的消息交换模式	63
4.3 SOAP 消息	64
4.3.1 SOAP 的信封	65
4.3.2 SOAP 的标题	65
4.3.3 SOAP 的主体	67
4.3.4 SOAP 的故障	68
4.4 SOAP 的属性	70
4.4.1 soapenv:encodingStyle 属性	70
4.4.2 soapenv:mustUnderstand 属性	70
4.4.3 soapenv:actor 属性	70
4.5 SOAP 的编码	71
4.5.1 简单类型	71
4.5.2 多态性存取器	72
4.5.3 复合类型	72
4.6 HTTP 上的 SOAP	73
4.6.1 HTTP 请求	73
4.6.2 SOAPAction	74

4.6.3 HTTP 响应	74
4.7 SOAP-RPC	75
4.8 SOAP 绑定示例	76
4.9 SOAP 消息的处理	80
4.10 运用 Apache SOAP 工具包实现 SOAP	81
4.11 小结	88
第 5 章 UDDI	89
5.1 UDDI 是什么	89
5.2 为什么 UDDI 很重要	92
5.3 UDDI 的技术框架	93
5.4 UDDI 数据结构	94
5.4.1 关于企业的信息	94
5.4.2 关于服务的信息	95
5.4.3 关于绑定的信息	96
5.4.4 关于服务规范的信息	98
5.4.5 发布者断言	99
5.5 UDDI 程序员的 API	99
5.5.1 体系结构和设计	100
5.5.2 API 参考	100
5.5.3 查询 API	101
5.5.4 发布 API	103
5.6 UDDI 的最佳实践	107
5.7 UDDI 工作区	108
5.8 协同使用 UDDI 和 WSDL——一个实例	109
5.9 小结	111
 第 3 部分 Java Web 服务的体系结构	
第 6 章 JavaServer Pages 和 Java servlet 简介	112
6.1 系统体系结构概述	112
6.1.1 两层模型	112
6.1.2 三层模型	113
6.1.3 n 层模型	114
6.2 servlet 是什么	115
6.2.1 servlet 和其他可选技术	116
6.2.2 servlet 的工作方式	117
6.2.3 Javax.servlet 包	117
6.2.4 servlet 的生命周期	119

6.2.5	部署 servlet	119
6.2.6	创建、部署和测试 servlet	120
6.3	会话跟踪	128
6.3.1	servlet 中保持会话跟踪的技术	128
6.3.2	Javax.servlet.http.Cookies 类	130
6.4	Java Server Pages 概述	130
6.4.1	JSP 组件	132
6.4.2	运用 JavaBean 组件实现 JSP	134
6.4.3	JSP 自定义标记	137
6.5	小结	140
第 7 章	J2EE 和 Web 服务	141
7.1	J2EE 体系结构简介	141
7.2	Web 组件	144
7.2.1	servlet	144
7.2.2	Java Server Pages	151
7.3	J2EE 和 Web 服务	153
7.3.1	SunONE 概述	154
7.3.2	DART	154
7.4	小结	155
第 8 章	JAXP	156
8.1	JAXP 简介	156
8.2	SAX API	157
8.2.1	设置 JWSDP	159
8.2.2	XML 文档	159
8.2.3	分析和显示	160
8.2.4	内容处理程序	163
8.3	DOM API	166
8.4	小结	179
第 9 章	JAXB	180
9.1	JAXB 简介	180
9.1.1	文档类型定义	181
9.1.2	绑定模式	182
9.1.3	JAXB 的优点	189
9.1.4	JAXB 的局限性	190
9.1.5	JAXB 的组件	190
9.1.6	JAXB 运行时库	190

9.2 解编、验证和编组	196
9.2.1 解编	197
9.2.2 验证	197
9.2.3 编组	197
9.3 用 JAXB 编程	198
9.4 小结	215
第 10 章 JAXM	216
10.1 JAXM 简介	216
10.1.1 通信类型	218
10.1.2 互操作性	220
10.1.3 SOAP 打包	221
10.2 J2EE 通信	222
10.3 JAXM 体系结构	223
10.3.1 JAXM 客户程序	223
10.3.2 JAXM 消息配置文件	224
10.4 JAXM 编程模型	225
10.4.1 javax.xml.messaging 程序包	225
10.4.2 javax.xml.soap 程序包	227
10.5 JAXM 示例	229
10.6 小结	239
第 11 章 JAX-RPC	240
11.1 JAX-RPC 概述	240
11.1.1 服务端点定义	241
11.1.2 服务的实现	242
11.1.3 服务的部署	242
11.1.4 服务的调用	242
11.1.5 服务的描述	243
11.2 JAX-RPC 映射	244
11.2.1 XML 到 Java 的映射	244
11.2.2 WSDL 到 Java 的映射	246
11.3 JAX-RPC 编程模型	248
11.3.1 客户编程模型	248
11.3.2 简单的回送示例	249
11.4 小结	261
第 12 章 JAXR	262
12.1 JAXR 概述	262

12.1.1	XML 注册库	263
12.1.2	性能配置文件	265
12.2	JAXR 编程模型	265
12.2.1	JAXR API	267
12.2.2	注册库信息模型	269
12.2.3	ConnectionFactory 和连接对象	271
12.2.4	LifecycleManager 和 BusinessLifecycleManager 接口	272
12.2.5	BusinessQueryManager 和 DeclarativeQueryManager 接口	276
12.3	JAXR 示例	278
12.4	小结	284
第 13 章	JSTL	285
13.1	JSTL 概述	285
13.1.1	JSP 自定义标记	286
13.1.2	JSTL 的目标	288
13.1.3	多重 TLD	289
13.2	表达式语言支持	290
13.2.1	EL 支持标记	291
13.2.2	EL 流程控制	293
13.2.3	EL 与 URL 相关的方式	295
13.3	i18n 支持	297
13.3.1	i18n 地区	297
13.3.2	i18n 消息模式化	298
13.3.3	i18n 数字和日期格式化	300
13.4	数据库支持	304
13.4.1	<sql:query>标记	304
13.4.2	<sql:update>标记	305
13.4.3	<sql:transaction>标记	305
13.4.4	<sql:driver>标记	306
13.4.5	<sql:param>标记	307
13.5	XML 支持	307
13.5.1	XML 核心标记	308
13.5.2	XML 流程控制	309
13.5.3	XML 转换支持	311
13.6	小结	313

第 4 部分 附 录

附录 A WSDL 基础	314
附录 B SOAP 基础	321
附录 C UDDI 2.0 版本程序员 的 API 基础	326
附录 D XML 入门	342
附录 E Java WSDP	387

第1部分 Web服务简介

第1章 Web服务的体系结构

本章主要内容:

- Web 服务概述
- 面向服务的体系结构
- 服务的角色
- 技术堆栈
- 体系结构的过程
- 价值链

在当前的商务方案中,电子商务是个时髦的词。电子商务的概念是在 Web 服务的推动下兴起的。Web 服务是通过 Web 定义、发布和访问的完整模块式应用程序。本章简述了 Web 服务概念,还讨论了面向服务的体系结构(service-oriented architecture, SOA)并且描述了 SOA 如何与当前的电子商务模式相配合,进而讨论了 Web 服务模式与行业中现有其他模式相比的优势。

1.1 Web 服务概述

在今天这个充满竞争的时代,全球化看起来应该是成功的诀窍。几乎所有的企业都转向将电子商务作为其全球化市场标志的手段。然而,随着企业需求越来越复杂性,IT 行业很快就认识到为企业提供电子商务解决方案并不容易。尽管有很多技术是可行的,例如远程方法调用(Remote Method Invocation, RMI)和公用对象请求代理程序体系结构(Common Object Request Broker Architecture, CORBA),但是还没有适合用于动态电子商务解决方案的。这些技术都需要与后端数据结构保持紧密的一致性,而且改变时容易受到损害。因此就需要一种能够集成松散耦合的、异种系统的技术。Web 服务就是针对行业中这一需求应运而生的。

Web 服务本身就是由松散耦合的异种系统构成的。而这些异种系统又是由基于组件的应用程序和采用基于可扩展标记语言(eXtensible markup language, XML)的标准化信息格式构成。

为了理解 Web 服务模式的适用性,我们还需要了解其体系结构,下面是 Web 服务模式的基本组成部分:

- 面向服务的体系结构
- 服务的角色

- 技术堆栈
- 体系结构的过程
- 价值链

本章下面的部分将会深入介绍这些组成部分的细节。

1.2 面向服务的体系结构

面向服务的体系结构(SOA)构成了 Web 服务模式的基础。SOA 是从面向对象模式发展而来的，在面向对象模式中，可以将所有的实体看作对象。类似地，在 SOA 中，也可以将所有的实体看作服务。可以使用对象来创建服务，对象不过是可以重用的软件组件。SOA 提供了一种编程模式，使驻留在任意网络中的服务都可以被其他服务发布、定位和调用。

SOA 定义了服务模式，这也是从对象模式派生而来的。服务模式具有如下特征：

- 服务的抽象
- 服务的封装
- 服务的模块化
- 服务的多态性

下面将进一步讨论这些特征的具体细节。

1.2.1 服务的抽象

在对象模式中，抽象定义了对应的特征。与之类似，在服务模式中，服务的抽象定义了服务的特征。

研究如下的例子：假设有一个包括结账任务在内的电子购物应用程序。这个任务包括两个子任务：分别是访问信用卡验证服务和商品交付的运输服务。信用卡验证服务给出消息，指出信用卡是否成功地验证。这一服务可以分成若干子过程，分别是：

- 授权访问客户系统
- 验证依据
- 数据库查询和查找
- 验证信用卡
- 生成应用程序异常
- 返回逻辑输出

这些子过程需要如下输入信息：

- 持卡者的姓名
- 持卡者的地址
- 信用卡号
- 信用卡有效日期
- 需要确认的金额

正如您注意到的这个例子是用一个用例来表示，如图 1-1 所示。可以在开发应用程序中采用一个用例来模块化应用程序并为之提供一个结构。

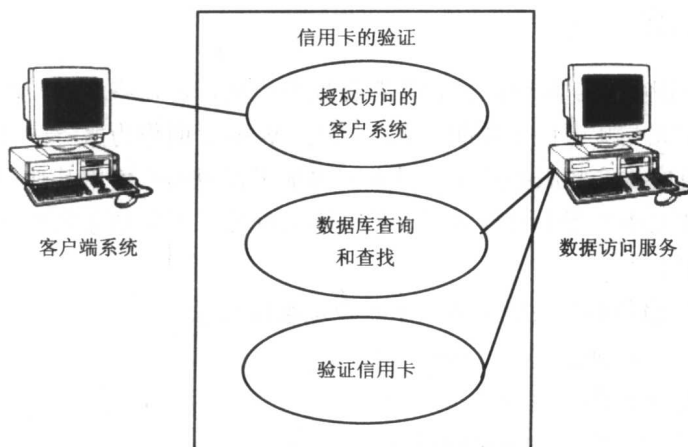


图 1-1 信用卡验证服务的用例

系统边界定义了系统为满足程序的目标所要做的事。信用卡验证服务的系统边界包含下列用例：

- 授权访问客户系统：这一用例的目的是确保客户在验证他们的客户 ID 之后可以登录至服务。主要行动者是客户系统。在本用例中，客户系统登录到授权给客户 ID 的服务中。服务设置客户 ID 允许访问的具体资源上下文环境。这种许可是以策略文件为基础的
- 数据库查询和查找：这一用例的目的是确认客户的特征。主要行动者是数据访问服务。在本用例中，服务根据来自数据库的客户 ID 获得与客户相关的信息。服务还要核对客户 ID 的身份并确认客户的特征。验证依据的子过程也在这一用例中执行
- 验证信用卡：这一用例确认客户的信用详情。主要行动者是数据访问服务。在本用例中，服务确认与信用卡相关的输入细节。为此，服务要访问外部信用数据库来核对信用卡。在核对之后，服务会授权事务处理。事务处理完毕后，服务将事务处理记录在数据库中。服务还可以记录服务使用的模式

在验证信用用例之后，就是更为普遍的子过程，例如，生成应用程序异常和返回逻辑输出。

注意，这种描述定义了信用卡验证服务的特征。服务的这种特征，根据其基本原理可以称为服务的抽象。

在定义信用卡验证服务的特征后，就可以惟一地这一服务与其他服务相区别。这样也就可以像定义服务的操作特征一样定义服务的抽象。

1.2.2 服务的封装

在定义服务的特征后，有必要对用户隐藏服务的基本原理过程。对服务的用户隐藏过程称为服务的封装。就像在对象模式中，抽象和封装互为补充一样，在服务模式中，服务的抽象和服务的封装也是互为补充的。

服务的封装对用户和实现者均有益处。用户得益于使用这种服务的简单，不需要考虑服务抽象的内部处理。而且用户也不需要关心结构细节，可以只为服务抽象提供即插即用逻辑。实现者从这种模式的易扩展性受益，可以用改进的版本代替执行的现行版本而不会影响用户。

1.2.3 服务的模块化

服务的模块化将服务分割成松散耦合的服务抽象模块，可以将这些模块组合创建新的可重用的服务库。我们可以回顾一下前面的内容，放置在 Web 上面作为服务的任何应用程序均由封装的多重服务抽象组成。这些服务抽象可以在其内部工作流程中使用其他服务抽象。例如，购车服务抽象可以在其工作流程中用到信用卡验证服务抽象。服务抽象的相互依赖需要一个服务库。这种库的好处在于：

- 服务库可以将服务抽象按照服务的基本原理编成目录
- 服务库为重用服务抽象提供一种公共机制
- 服务库将服务抽象的使用分类
- 服务库可以作为所有服务抽象的容器

模块化的优势在于将应用程序放置在包、模块或是库中，而这些已经存在于面向对象领域中。

服务模块化背后的基本思想是提供服务抽象模块，而这些服务抽象模块也是由其他服务抽象所构成。尽管这些服务抽象均是捆绑在给定的模式中，但是它们彼此之间并没有严格的语义约束。当然，在一个模块中的服务抽象与模块中其他服务抽象相联系时还是有结合性的。

为了理解服务的模块化，可参考 1.2.1 小节的结账任务模块实例。结账模块由信用验证服务和发货服务构成，这两种服务均是服务抽象。它们之间是松散耦合的，但又是彼此相互关联的。

1.2.4 服务的多态性

当前的行业中，多态性并不是新概念，在对象模式中已经得到广泛应用来实现可扩展性。服务的多态性通过使用服务抽象的惟一通用定义的服务接口提供调用不同的服务行为的可行性。服务接口展示了 Web 服务的多种服务方法。服务的多态性使服务的动态运行时绑定成为可能。这一特征使一个服务可以在运行时动态地使用另一个服务。

为了理解服务的多态性，来看一下发布服务的例子。假定有一个具体的服务抽象——发布框架的 Web 服务，在下面的多态方式中将要用到这一服务抽象。

设想这种服务抽象定义了通用的服务接口，这种接口采用了公共数据集作为输入，这样就可以按照适当的方式发布数据。在给定这种通用数据接口后，服务提供商就可以定义一种具体的服务模块来转换数据，例如，将数据转换成 Adobe PDF 格式显示。作为客户，假定需要使用服务以 Adobe 格式发布数据。在这种情况下，可以运用通用服务接口的一种具体实现在客户端应用程序中添加适当的代码，以便以 PDF 文件形式发布数据。假设稍后需要在现有的 PDF 发布标准上作一些改动，使之成为更加适应业界的无线标记语言(Wireless Markup Language, WML)或类似格式。这时，不需要打乱客户端的代码，因为代码是围绕通用服务接口建立的；而只需要在应用程序上面添加一些适当的代码，就能够使用这种通用服务接口作为具体实现，以 WML 格式发布数据。

所有服务提供商需要做的就是为同一服务接口提供不同的实现。在这种情况下，封装良好的服务抽象可以提供我们所需要的可扩展性。

这一提供通用服务接口的松散耦合服务抽象的过程就称为服务的多态性。