

<<Web Services技术、架构>>

图书基本信息

书名：<<Web Services技术、架构和应用>>

13位ISBN编号：9787505382626

10位ISBN编号：7505382624

出版时间：2003-1

出版时间：电子工业出版社

作者：柴晓路

页数：659

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Web Services技术、架构>>

内容概要

本书的内容涵盖了Web Services的各种关键技术、Web Services的整体体系架构和应用体系架构，以及Web Services应用的设计和开发。

本书以Web Services技术系列为主线，逐一详细分析解释包括Web Services的各种核心技术，包括XML、Namespace、XML Schema、SOAP、WSDL、UDDI、WS-Inspection、WS-Security、WS-Routing等。在分析阐述的同时，使用了大量的应用设计案例和基于Microsoft.NET的开发实例，以使读者能够更形象地理解这些技术和技术的设计思路、设计原则和相应的实现方法。

本书适用于对计算机技术的基础和架构已经有了一定的了解，对WEB应用的架构有一定了解，同时又对当今正在发生的计算机技术的新一代技术与应用革命——Web Services浪潮的本质有着浓厚的兴趣的读者。

作者简介

柴晓路，上海华和得易信息技术有限公司CIO。

2000年获复旦大学计算机科学硕士学位，同年加入上海华和得易信息技术有限公司。

专长于Web Services架构、Web Services系列技术以及基于XML的系统集成和数据交换技术。

目前是WS-I工作组成员（国际Web Services互操作组织）、UDDI顾问组成员，参与UDDI2.0版和3.0版的开发。

2001年创办UDDI-China中国UDDI技术联盟，该联盟承担UDDI.org的所有英文规范的中文版开发。

目前还担任IBM developerWorks专栏作家、CSDN名家专栏作家等。

著有学术论文和技术文章共约50余篇。

梁宇奇，Intel（中国）公司软件工程师。

2002年获复旦大学计算机科学硕士学位。

专长于Web Service系列技术、Web Service开发、数据挖掘技术等。

2001年成为UDDI-China中国UDDI技术联盟成员，该联盟承担UDDI.org的所有英文规范的中文版开发

。

<<Web Services技术、架构>>

书籍目录

第1章 Web Services, 第三次革命 11.1 Web Services的兴起 11.2 Web Services规范 21.3 Web Services技术平台 41.4 市场趋势 61.5 本书的内容安排 8第2章 Web Services带来了什么 102.1 什么是Web Services 102.1.1 Web Services的概念 102.1.2 Web对象 112.1.3 Web Services体系架构模型 122.1.4 Web Services协议栈 142.1.5 扩展Web Services协议栈 162.1.6 Web Services的类别 172.2 Web Services与应用集成 182.2.1 什么是企业应用集成 182.2.2 企业应用集成EAI的类型 192.2.3 商业需求驱动Web Services 212.2.4 Web Services和EAI 242.2.5 Web Services与B2Bi 292.3 J2EE与.NET, 对抗与整合 302.3.1 J2EE与.NET概述 302.3.2 J2EE与.NET的比较 312.3.3 整合J2EE和.NET 382.3.4 发展方向 392.3.5 J2EE与.NET总结 402.4 小结 41第3章 XML技术简述 423.1 了解XML 433.2 XML语法简述 463.2.1 XML元素 463.2.2 XML文档 493.2.3 元素, 组成XML文档的基本实体 503.2.4 字符数据 543.2.5 使用属性修饰元素 543.2.6 字符引用和实体引用 563.2.7 一些其他特性 573.2.8 格式正规的文档 583.3 XML命名空间 593.3.1 什么是命名空间 603.3.2 定义和声明命名空间 603.4 XML相关技术 643.4.1 DOM 643.4.2 SAX 653.4.3 XSL/XSLT/XPath 653.4.4 XLink/XPointer 663.4.5 XQuery 673.5 小结 68第4章 XML Schema建模 694.1 XML Schema入门 694.1.1 购买订单模式文档 714.1.2 复合类型定义, 元素和属性声明 734.1.3 使用简单类型 774.1.4 元素内容 844.1.5 模式注释 874.1.6 构造内容模型 884.1.7 属性组 904.1.8 空值 924.2 案例研究——认证考试系统数据建模 934.2.1 Application 944.2.2 CourseSession 994.2.3 ExamSession 1034.3 模式的命名空间 1054.3.1 目标命名空间和未限定的局部元素/属性 1054.3.2 带有命名空间限定的局部元素和属性 1084.3.3 全局声明与局部声明 1114.3.4 未声明的目标命名空间 1124.4 模式文档的重用和扩展 1134.4.1 由多个文档组成的模式文档 1134.4.2 通过扩展来派生类型 1184.4.3 在实例文档中使用派生类型 1194.4.4 通过约束来派生复合类型 1204.4.5 重新定义类型 1224.4.6 置换组 1234.4.7 抽象元素和类型 1254.4.8 控制对派生类型的创建和使用 1264.5 级模式约束 1284.5.1 描述惟一性 1314.5.2 定义键和键的引用 1324.5.3 XML Schema约束与XML 1.0的ID属性 1334.5.4 导入类型 1334.5.5 any元素和any属性 1364.5.6 schemaLocation 1404.5.7 一致性 1414.6 小结 142第5章 开发工具简介 1435.1 Visual Studio.NET 1435.1.1 Visual Studio.NET简介 1435.1.2 .NET Framework 1435.2 C#语言 1455.2.1 C#语言特征 1465.2.2 C#程序结构 1465.2.3 方法介绍 1475.3 数据访问技术ADO.NET 1525.4 Web Services设计基础 1535.4.1 第一个Web Service – “Hello World” 1535.4.2 Web Services与ADO.NET结合 1555.5 小结 158第6章 调用Web Services 1506 SOAP的起源 1536.1.1 CORBA 1536.1.2 COM+ 1556.1.3 EJB 1576.1.4 抹平平台差异的SOAP 159 SOAP概览 1606.2.1 需求、目标与约定 1616.2.2 SOAP消息示例 1626.2.3 基本概念 1646.2.4 消息交换模型 1666.2.5 SOAP和XML的关系 1756.3 SOAP消息结构 1756.3.1 编码风格定义 1766.3.2 信封版模型 1766.3.3 SOAP信封的模式定义 1806.3.4 SOAP Header 1846.3.5 SOAP Body 1896.3.6 SOAP Fault 1976.4 SOAP编码规则简述 2026.5 SOAP消息编程 2036.5.1 定制SOAP客户端发送的SOAP消息格式 2036.5.2 定制Web Services期望的SOAP消息格式 2116.5.3 利用XML Serialization定制SOAP消息 2126.6 在SOAP中使用URI 2136.7 小结 214第7章 SOAP技术进阶 2157.1 SOAP编码规则 2157.1.1 SOAP简单类型编码规则 2157.1.2 简单类型 2187.1.3 多态存取 2227.1.4 复合类型 2227.2 SOAP的RPC表示 2387.2.1 RPC和SOAP Body 2387.2.2 将应用程序定义的名称映射成为XML名称 2397.2.3 RPC和SOAP Header 2407.2.4 RPC错误 2407.2.5 SOAP RPC表示示例 2407.3 SOAP协议绑定框架 2457.3.1 绑定框架概述 2457.3.2 绑定框架的目标 2467.3.3 绑定框架 2467.3.4 消息交换模式 2477.3.5 与应用相关的协议实施绑定 2487.3.6 描述特性和绑定 2487.4 单一请求响应传输消息交换模式 2507.4.1 传输消息交换模式概述 2507.4.2 消息交换模式名称 2517.4.3 单一请求响应简介 2517.4.4 正式定义 2527.4.5 错误处理 2547.4.6 经由SOAP中介的操作 2547.5 SOAP HTTP绑定 2557.5.1 单一请求响应交换 2557.5.2 SOAPAction 2617.6 SOAP Header编程 2647.6.1 如何构建SOAP处理程序 2647.6.2 自定义SOAP Header类 2667.6.3 在Web Services中处理SOAP Header 2677.6.4 在Web Services的客户端处理SOAP Header 2697.6.5 控制SOAP Header接收者 2707.6.6 Web Services期待的可选SOAP Header 2717.6.7 处理未知SOAP Header 2727.6.8 处理mustUnderstand要求的SOAP Header 2747.6.9 错误处理 2767.7 SOAP扩展编程 2777.7.1 自定义一个SOAP Extension类 2787.7.2 保存代表SOAP消息的Stream引用类型 2787.7.3

<<Web Services技术、架构>>

初始化SOAP Extension特殊数据 2797.7.4 处理SOAP消息 2807.7.5 SOAP Extension方法的调用次序 2817.7.6 配置SOAP Extension与Web Services接口的结合 2827.7.7 完整的SOAP Extension实例 2837.8
 SOAP应用模式 2867.8.1 基本消息交换模式 2867.8.2 运用加密机制的安全消息交换模式 2957.8.3 包含消息中介的消息交换模式 2987.8.4 高级消息交换模式 3007.8.5 面向处理和性能的消息交换模式 3087.9
 基于SOAP Header的Web Services规范 3147.9.1 WS-Security 3157.9.2 WS-License 3177.9.3 WS-Routing 3197.9.4 WS-Referral 3217.10 小结 323第8章 Web Services接口描述 3258.1 概述 326
 WSDL文档结构 3328.2.1 WSDL文档框架 3338.2.2 应用实例：产品目录服务 3378.2.3 类型定义 3418.2.4 消息定义 3448.2.5 端口类型定义 3478.2.6 绑定定义 3508.2.7 服务访问点定义 3518.2.8 任务定义 3528.2.9 WSDL模式定义 3538.3 多种形式的绑定 3628.3.1 SOAP绑定 3628.3.2 HTTP GET和HTTP POST绑定 3728.3.3 MIME绑定 3758.4 VS.NET中的WSDL支持 3818.4.1 WSDL工具 3818.4.2 .NET Framework中的WSDL支持 3828.4.3 动态代理类生成 3918.5 Web Services界面图示工具 3968.6
 小结 397第9章 Web Services注册与发现 3999.1 引言 3999.1.1 商业需求驱动 3999.1.2 UDDI计划 4019.1.3 UDDI规范 4029.1.4 UDDI数据模型和信息描述 4039.1.5 UDDI注册中心 4059.1.6 UDDI API 4069.1.7 UDDI注册中心运作模式 4069.1.8 分类法 4079.1.9 安全性 4089.1.10 全球化考虑 409
 UDDI数据结构 4109.2.1 概述 4109.2.2 businessEntity结构 4139.2.3 businessService结构 4179.2.4 bindingTemplate结构 4199.2.5 bindingTemplate和服务调用 4239.2.6 tModel结构 4279.2.7 publisherAssertion结构 4319.3 UDDI注册服务API 4329.3.1 SOAP的使用细节 4339.3.2 安全认证API 4359.3.3 查询API 4429.3.4 发布API 4579.4 分布式Web Services发现 4719.4.1 概述 4729.4.2 WS-Inspection文档结构 4729.4.3 WS-Inspection与UDDI的关系 4769.5 小结 480第10章 UDDI编程与UDDI高级特性 51710.1 UDDI SDK编程 51710.1.1 UDDI SDK简介 51710.1.2 UDDI SDK使用 52010.1.3 查询UDDI节点内的注册信息 52210.1.4 从UDDI注册中心查询详细信息 52810.1.5 向UDDI注册中心发布信息 53410.1.6 在UDDI注册中心中删除信息 54110.1.7 错误处理 54310.1.8 UDDI注册的一般方法 54510.1.9 Microsoft UDDI注册中心 54810.2 UDDI应用案例 54910.2.1 登录UDDI注册中心 55010.2.2 发布UDDI高技术公司的商业信息 55110.3 核心元数据体系tModel与服务注册 56110.3.1 使用标识符 56110.3.2 使用分类法 56410.3.3 第三方分类法和标识系统 56710.3.4 关系和发布者断言 57110.3.5 UDDI类型系统 57510.4 小结 584第11章 Agility电子商务解决方案 5851
 电子商务应用集成需求 58511.1.1 新的电子商务发展 58511.1.2 新的挑战 58611.1.3 电子商务运作模式的革新 58611.2 Agility电子商务解决方案 58611.2.1 传统的解决方案 58611.2.2 基于Web Services的电子商务应用集成模式 58711.2.3 Agility电子商务解决方案 58911.2.4 案例 59111.2.5 Agility电子商务解决方案设计概述 59211.3 Web Services提供者的设计 59311.3.1 内部应用系统设计 59311.3.2 Web Services设计 59411.3.3 Web Services的部署 59911.3.4 Web Services注册模块设计 59911.4 Web Services客户端设计 60011.4.1 UDDI查找 60011.4.2 Web Services集成 60611.5 小结 613第12章 Web Services案例研究 61412.1 软件反馈跟踪平台 61412.1.1 应用背景简述 61412.1.2 系统构架概述 61512.1.3 系统间的互 61512.1.4 为什么使用Web Services解决方案 61612.1.5 交互界面设计 61612.1.6 Web Services实现 62112.1.7 遗留的问题 62512.2 认证考试申请服务 62512.2.1 应用背景简述 62512.2.2 应用模式演示 62512.2.3 服务的可用性和连接的持久性 62712.2.4 服务的延伸 62912.3 Web Services事务性应用 62912.3.1 应用背景简述 62912.3.2 解决方案 63012.3.3 数据库事务的XML表示 63212.3.4 事务性保证 63512.3.5 相关的技术发展 64412.4 统一身份认证服务 64412.4.1 应用背景简述 64412.4.2 解决方案 64512.4.3 使用Web Services架构 65012.4.4 相关的工作 65712.5 小结 658第13章 Web Services, 时我待 65913.1 当前Web Services技术的局限性 65913.2 业界在Web Services领域的努力 66013.3 Web Services带来的机遇 663

<<Web Services技术、架构>>

媒体关注与评论

本书以Web Services技术系列为主线，逐一详细分析解释包括Web Services的各种核心技术，包括XML， Namespace， XML Schema， SOAP， WSDL， UDDI， WS-Inspection， WS-Security， WS-Routing等。在分析阐述的同时，使用了大量的应用设计案例和开发实例，以使读者能够更形象地理解这些技术和技术的设计思路、设计原则和相应的实现方法。具体的，本书使用的实现平台是Microsoft.NET。

编辑推荐

本书以Web Services技术系列为主线，逐一详细分析解释包括Web Services的各种核心技术，包括XML, Namespace, XML Schema, SOAP, WSDL, UDDI, WS-Inspection, WS-Security, WS-Routing等。在分析阐述的同时，使用了大量的应用设计案例和开发实例，以使读者能够更形象地理解这些技术和技术的设计思路、设计原则和相应的实现方法。具体的，本书使用的实现平台是Microsoft.NET。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>