

《SOA核心技术及应用》

图书基本信息

书名：《SOA核心技术及应用》

13位ISBN编号：9787121061738

10位ISBN编号：7121061732

出版时间：2008-5

作者：王紫瑶,南俊杰,段紫辉,钱海春,陈荻玲,李冬

页数：586

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《SOA核心技术及应用》

内容概要

本书立足于技术底层，试图指引读者理解构建SOA的基础编程模式。这一技术模式也并非耳熟能详的Web服务，而是新兴的SOA技术标准——服务组件架构（Service Component Architecture, SCA）、服务数据对象（Service Data Object, SDO）和业务流程执行语言（Business Process Execution Language, BPEL）。这三个技术分别解决了服务构建、数据模型和流程编排领域的问题，因而被称为“SOA编程模式的铁三角”。

本书首先和读者一起回顾一下SOA的基本理念和实施SOA的技术需求，进而简要介绍SCA、SDO和BPEL技术。此后的章节根据这三个主要的技术方向分为不同的模块，分别介绍相应技术的起源、理念、规范和实例，力图使读者全方位地理解这些技术，并能够进行基本的开发实践。..

本书的读者对象是参与SOA项目实施的系统架构设计师、服务开发人员和测试人员，或者是相关专业的在校学生，以及任何希望了解SOA的广大技术人员。...

书籍目录

第1章 SOA的技术决策

- 1.1 公司IT部门面临的困境
- 1.2 决策者的决策——部署SOA
 - 1.2.1 什么是SOA
 - 1.2.2 SOA实施的主要困难
- 1.3 SOA的技术抉择
 - 1.3.1 相关技术概览
 - 1.3.2 服务
 - 1.3.3 数据和消息模型
 - 1.3.4 服务编排和流程整合
- 1.4 本章小结

第2章 为什么需要SCA

- 2.1 SCA的起源
 - 2.1.1 Web服务技术的兴起
 - 2.1.2 Web服务调用框架的任务
 - 2.1.3 SCA的提出
- 2.2 WPS中SCA实现简介
 - 2.2.1 服务组件 (Service Component)
 - 2.2.2 导入 (Import)、导出 (Export) 和绑定 (Binding)
 - 2.2.3 同步调用和异步调用
 - 2.2.4 SCA的静态调用和动态调用
 - 2.2.5 接口映射 (Interface Map)
 - 2.2.6 服务模块 (Module)
- 2.3 为什么需要SCA
- 2.4 本章小结

第3章 SCA基本概念

- 3.1 SCA编程模型
- 3.2 SCA组件 (component)
 - 3.2.1 组件的实现 (implementation)
 - 3.2.2 组件中的接口
 - 3.2.3 组件中的绑定
 - 3.2.4 组件的服务 (service)
 - 3.2.5 组件的引用 (reference)
 - 3.2.6 组件的属性 (property)
 - 3.2.7 组件中的服务质量
- 3.3 SCA构件 (composite)
 - 3.3.1 构件的属性 (property)
 - 3.3.2 构件的服务 (service)
 - 3.3.3 构件的引用 (reference)
 - 3.3.4 构件的连接 (wire)
 - 3.3.5 将构件用做组件的实现
- 3.4 打包 (package) 和部署 (deployment)
 - 3.4.1 SCA域 (domain)
 - 3.4.2 Contribution
- 3.5 SCA中的常见绑定
 - 3.5.1 SCA绑定
 - 3.5.2 Web服务绑定

- 3.5.3 JMS绑定
- 3.6 策略框架 (policy framework)
 - 3.6.1 框架模型
 - 3.6.2 策略意向和策略集的应用
 - 3.6.3 角色和分工
 - 3.6.4 安全策略
 - 3.6.5 可靠性策略
- 3.7 扩展模型 (extension model)
- 3.8 本章小结
- 第4章 Tuscany SCA简介
 - 4.1 Tuscany与Apache开源
 - 4.2 Tuscany与Apache Maven
 - 4.3 Maven简介
 - 4.3.1 Maven及其所解决的问题
 - 4.3.2 Maven的安装
 - 4.3.3 命令行的使用
 - 4.3.4 Maven与Eclipse的结合使用
 - 4.3.5 从源代码构建Tuscany SCA Java
 - 4.4 Maven原型 (Archetype) 与Tuscany SCA开发
 - 4.4.1 Maven原型 (Archetype) 简介
 - 4.4.2 创建Maven原型
 - 4.4.3 Tuscany SCA的安装
 - 4.5 本章小结
- 第5章 SCA实践起步
 - 5.1 SCA初接触
 - 5.2 SCA项目的创建
 - 5.3 定义一个SCA组件
 - 5.4 SCA组件的单元测试
 - 5.4.1 JUnit与单元测试 ”
 - 5.4.2 SCA单元测试的编写
 - 5.4.3 单元测试的运行
 - 5.5 接口与实现
 - 5.5.1 接口的重用
 - 5.5.2 接口与实现的分离
 - 5.5.3 业务逻辑与技术的分离
 - 5.6 实现一个Web服务
 - 5.6.1 使用Web服务绑定
 - 5.6.2 Java接口到WSDL接口的转化
 - 5.6.3 Web服务测试用例
 - 5.6.4 部署Web服务到Tomcat服务器
 - 5.7 在一个SCA中使用Web服务
 - 5.7.1 引用的定义
 - 5.7.2 在实现中使用引用
 - 5.7.3 引用使用的测试用例
 - 5.7.4 测试驱动的学习模式
 - 5.8 本章小结
- 第6章 使用SCA构建SOA应用
 - 6.1 SCA服务整合的特点
 - 6.1.1 对服务整合技术的要求

- 6.1.2 SCA服务整合的特点
- 6.1.3 组件、构件和SCA域
- 6.1.4 两个主要的核心问题
- 6.2 SCA绑定
 - 6.2.1 默认服务绑定
 - 6.2.2 显式地获取服务
 - 6.2.3 使用引用获取服务
 - 6.2.4 服务的命名约定
 - 6.2.5 引用的使用与多接口
 - 6.2.6 服务的无状态实现
- 6.3 服务实现的不同作用域
 - 6.3.1 EJB会话Bean的无状态和有状态
 - 6.3.2 一个错误的计数器实现
 - 6.3.3 实现实例创建和销毁的时机
 - 6.3.4 构件作用域
 - 6.3.5 请求作用域
 - 6.3.6 会话作用域
- 6.4 可会话服务
 - 6.4.1 可会话的例子
 - 6.4.2 会话的标识
 - 6.4.3 可会话服务的生命周期
 - 6.4.4 会话的属性
 - 6.4.5 服务的显式引用
- 6.5 异步调用与可会话编程
 - 6.5.1 单向异步
 - 6.5.2 延迟响应
 - 6.5.3 组件上下文和请求上下文
 - 6.5.4 请求回调 (Callback)
- 6.6 不同类型的绑定
 - 6.6.1 web服务绑定
 - 6.6.2 企业会话Bean绑定
 - 6.6.3 JMS绑定
- 6.7 组件的不同类型的实现
 - 6.7.1 Java实现
 - 6.7.2 SCA对脚本语言的支持
 - 6.7.3 使用构件作为组件实现
- 6.8 SCA的服务接口
 - 6.8.1 WSDL接口
 - 6.8.2 Java接口
- 6.9 SCA其他语言的实现
 - 6.9.1 SCA的C++实现
 - 6.9.2 SCA的PHP实现
 - 6.9.3 SCA不同语言实现小结
- 6.10 本章小结
- 第7章 SCA与其他技术
 - 7.1 SCA与Spring
 - 7.1.1 Spring的新主张
 - 7.1.2 从Spring到Tuscany
 - 7.1.3 在SCA里使用Spring

- 7.2 SCA与OSGi
 - 7.2.1 OSGi的成功
 - 7.2.2 OSGi与SCA的异同
 - 7.2.3 SCA和OSGi的关系
- 7.3 SCA与SDO
 - 7.3.1 SDO简介
 - 7.3.2 SCA里使用SDO
 - 7.3.3 数据整合和服务整合的目的
- 7.4 本章小结
- 第8章 什么是SDO
 - 8.1 产生背景
 - 8.1.1 SDO的历史
 - 8.1.2 产品支持
 - 8.1.3 基本结构
 - 8.2 设计思想
 - 8.2.1 为什么需要SDO
 - 8.2.2 设计需求
 - 8.3 SDO、EJB和Hibernate
 - 8.3.1 SDO和EJB
 - 8.3.2 SDO和Hibernate
 - 8.4 Data Access Service (DAS)
 - 8.4.1 DAS的功能
 - 8.4.2 关系数据库数据访问服务 (RDBDAS)
 - 8.5 本章小结
- 第9章 SDO的基本概念
 - 9.1 数据图和数据对象
 - 9.1.1 数据图 (DataGraph)
 - 9.1.2 数据对象 (DataObject) 概念和特性
 - 9.1.3 数据对象的接口定义
 - 9.1.4 XPath表达式
 - 9.2 变更摘要
 - 9.2.1 内容和范围
 - 9.2.2 序列化
 - 9.2.3 接口定义
 - 9.3 类型、属性和序列
 - 9.3.1 类型 (Type)
 - 9.3.2 类型和属性的模型
 - 9.3.3 属性 (Property)
 - 9.3.4 序列 (Sequence)
 - 9.4 Helper (帮助) 类
 - 9.4.1 DataFactory
 - 9.4.2 TypeHelper
 - 9.4.3 CopyHelper
 - 9.4.4 EqualityHelper
 - 9.4.5 XMLHelper
 - 9.4.6 XMLDocument
 - 9.4.7 XSDHelper
 - 9.4.8 DataHelper
 - 9.4.9 HelperContext和HelperProvider

9.5 序列化

9.5.1 Java序列化

9.5.2 XML序列化

9.6 本章小结

第10章 SDO和XSD的关系

10.1 XSD基本概念

10.1.1 XML概念

10.1.2 XSD介绍

10.2 XSD和SDO的映射

10.3 本章小结

第11章 在Tuscany上使用SDO

11.1 搭建Tuscany环境

11.2 定义SDO模型

11.2.1 静态模型的定义

11.2.2 动态模型的定义

11.3 创建和使用SDO实例

11.4 SDO和SCA的集成使用

11.5 本章小结

第12章 SDO的深入实践

12.1 ChangeSummary

12.1.1 应用场景描述

12.1.2 使用介绍

12.1.3 总结

12.2 序列化服务

12.2.1 应用场景描述

12.2.2 使用介绍

12.2.3 总结

12.3 比较和复制服务

12.3.1 应用场景描述

12.3.2 使用介绍

12.3.3 总结

12.4 本章小结

第13章 业务对象介绍

13.1 WPS简介

13.2 WID简介

13.3 业务对象

13.3.1 业务对象介绍

13.3.2 业务对象（BO）模型的开发

13.3.3 总结

13.4 本章小结

第14章 BPEL概述

14.1 BPEL及其发展历程

14.2 BPEL相关技术

14.3 初识BPEL

14.4 BPEL引擎

14.5 BPEL与SOA

14.6 BPEL与业务过程管理

14.7 本章小结

第15章 BPEL详解

- 15.1 BPEL总体结构
- 15.2 BPEL中的交互
- 15.3 BPEL中的数据
- 15.4 消息关联
- 15.5 BPEL活动
 - 15.5.1 BPEL基本活动 (BaActivities)
 - 15.5.2 BPEL结构化活动
- 15.6 作用域 (Scope) 的概念
- 15.7 错误处理与补偿 (Fault Handling and Compensation)
- 15.8 事件处理 (Event Handling)
- 15.9 BPEL的扩展机制
- 15.10 本章小结
- 第16章 IBM业务过程产品简介
 - 16.1 业务过程管理及其产品概述
 - 16.2 WebSphere Process Server (WPS)
 - 16.3 WebSphere Integration Developer (WID)
 - 16.4 本章小结
- 第17章 实例分析
 - 17.1 背景介绍
 - 17.1.1 业务需求
 - 17.1.2 技术架构考虑
 - 17.2 方案设计
 - 17.3 开发过程
 - 17.3.1 创建库和模块
 - 17.3.2 创建业务对象
 - 17.3.3 创建服务接口
 - 17.3.4 创建业务状态机
 - 17.3.5 创建业务流程和人员任务
 - 17.3.6 送货服务系统的开发
 - 17.3.7 银行业务系统的开发
 - 17.3.8 创建调解流程
 - 17.3.9 Web应用程序的开发
 - 17.3.10 整合服务
 - 17.4 系统调试
 - 17.4.1 功能测试
 - 17.4.2 系统测试
 - 17.5 监控管理
 - 17.5.1 流程的监控管理
 - 17.5.2 通用事件的监控管理
 - 17.5.3 失败事件的监控管理
 - 17.6 本章小结
- 附录A 数据对象的接口
- 附录B Externalizable Delegator
- 附录C 数据图序列化的XSD
- 附录D XSD和SDO的映射表格
- 附录E BPEL的Schema定义
- 技术术语表
- 参考文献

《SOA核心技术及应用》

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:www.tushu000.com