

<<分布式对象技术及其应用>>

图书基本信息

书名：<<分布式对象技术及其应用>>

13位ISBN编号：9787302180586

10位ISBN编号：730218058X

出版时间：2008-11

出版时间：孟宪福 清华大学出版社 (2008-11出版)

作者：孟宪福

页数：188

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>

<<分布式对象技术及其应用>>

前言

分布式对象技术是在面向对象技术的基础上发展起来的，它要解决的主要问题是位于不同进程中的对象之间的调用问题。

在中间件系统、Web服务以及SOA（Service Oriented Architecture，面向服务架构）的研究与开发等许多重要领域，分布式对象技术都发挥着不可替代的作用。

本书共由8章组成，按照循序渐进的原则，从理论到实践逐步介绍分布式对象技术的典型代表——CORBA的基本概念与程序设计规则，主要包括CORBA的组成与处理流程、IDL接口定义语言、CORBA客户端程序设计与服务器程序设计以及动态接口等内容。

特别是，为了使读者能够尽快运用分布式对象技术来解决实际问题，本书利用两章的篇幅详细地介绍了基于CORBA的程序设计实例和基于Java RMI的程序设计实例，所给出的几个例子都是具有代表性的并具有实用价值的，通过对这些实例的学习，能够使读者进一步掌握分布式对象程序设计的要点，并能达到举一反三的目的。

所给出的程序都是在实际的环境下调试完成的，以使读者能够尽快掌握分布式对象这门应用技术。

本书详细地介绍了分布式对象的基本内容，从理论到实践完整而系统地介绍了CORBA分布式对象系统设计规范和程序设计过程。

本书的前6章主要是从理论的角度来介绍分布式对象系统的设计过程，而本书的后2章则从应用的角度来介绍分布式对象系统的实现过程。

第1章简要介绍了Java语言的基本内容，是为阅读后续章节中的程序做准备的，所介绍的内容也仅局限在其他章节中需要使用的基本内容，包括基本语句、类的定义和接口等。

第2章介绍了分布式对象的基本概念以及CORBA的基本构成和处理过程，这一章的内容将为准确理解后续章节的内容打下基础。

第3章主要介绍了分布式对象系统的开发流程、IDL语言以及从IDL到Java语言的映射。

IDL语言是基于CORBA的程序设计基础，而语言映射则为实现客户端和服务器提供了必需代码，这些代码包括Stub类、Skeleton类、Holder类和Helper等。

第4章介绍了CORBA客户端程序设计过程，主要包括ORB的初始化、分布式对象引用的获取以及分布式方法的调用等。

第5章介绍了CORBA服务器程序设计过程，主要包括BOA与POA的基本内容、分布式对象实现以及服务器进程实现等。

第6章介绍了动态接口技术，主要包括 any类型的处理、接口仓库、动态启动接口DII和动态骨架接口DSI等。

第7章介绍了几个典型的CORBA实例，通过对这些实例的学习，能够准确了解CORBA应用系统的完整实现过程。

第8章介绍了Java RMI远程对象技术，其中详细说明了基于回调技术的分布式对象系统设计过程。

作者认为，分布式对象作为一门应用技术，要想学好它，除了掌握基本理论之外，还必须加强实践环节。

读者可以边学习边上机，刚开始时可以在给定的环境下调试本书中的例题，待学习一段时间之后，就可以调试自己编写的程序了。

只有这样，才能加快学习进度，提高学习效率，真正掌握这门应用技术。

由于作者水平有限，经验不足，书中一定有不少缺点和错误，敬请有关老师、计算机工作者和广大读者批评指正。

<<分布式对象技术及其应用>>

内容概要

分布式对象技术是在面向对象技术的基础上发展起来的，它要解决的主要问题是位于不同进程中的对象之间的调用问题。

在中间件系统、Web服务以及SOA等需要多程序协作的许多领域，分布式对象技术都发挥着重要作用。

本书分8章，按照循序渐进的原则，从理论到实践逐步介绍分布式对象技术的典型代表CORBA和Java RMI的基本概念与程序设计规则。

特别是，为了使读者能够尽快运用分布式对象技术来解决实际问题，在本书的最后两章完整地给出了基于CORBA和Java RMI的多个应用实例及其程序开发过程。

本书是作者根据多年的教学经验和实践体会编写而成的，在内容编排上尽量体现易学的特点，在文字叙述上力求条理清晰、简洁，便于读者阅读。

本书可以作为大专院校计算机专业研究生或高年级本科生的教材，也可以作为非计算机专业学生或软件开发人员的参考书或自学用书。

<<分布式对象技术及其应用>>

书籍目录

第1章 Java语言基础1.1 Java语言的特点及其程序开发过程1.1.1 Java语言的特点1.1.2 Java程序的开发过程1.2 数据类型、变量、运算符和基本语句1.2.1 数据类型1.2.2 变量与常量1.2.3 运算符1.2.4 运算符的优先级1.2.5 数组1.2.6 字符串1.2.7 基本数据类型包装类1.2.8 基本语句1.3 类1.3.1 对象的生成与引用1.3.2 this与super1.3.3 类的定义1.3.4 成员变量的定义1.3.5 方法的定义1.3.6 构造方法的定义与使用1.3.7 static块1.3.8 对象的释放1.4 接口与异常处理1.4.1 接口1.4.2 异常处理1.4.3 包1.4.4 命令行参数1.5 多态性的实现1.6 委托处理与功能继承第2章 分布式对象与CORBA2.1 CORBA与OMG2.2 CORBA的发展历程2.3 分布式对象的定义与特点2.3.1 分布式对象的定义2.3.2 分布式对象系统的透明性2.3.3 分布式对象系统的复杂性2.4 CORBA系统的基本构成2.5 CORBA分布式对象环境2.6 分布式对象系统的处理过程第3章 分布式对象系统设计与IDL定义3.1 分布式对象系统的开发流程3.2 基于CORBA分布式对象系统设计3.3 IDL接口定义语言3.3.1 IDL的作用3.3.2 数据类型3.3.3 类型定义3.3.4 常量定义3.3.5 异常定义3.3.6 属性定义3.3.7 操作定义3.3.8 接口定义3.3.9 模块定义3.3.10 预处理器3.4 从IDL到Java的映射3.4.1 接口定义的映射3.4.2 实现引用传递的Holder类3.4.3 提供各种实用功能的Helper类3.4.4 其他IDL定义的映射3.4.5 IDL映射后的使用第4章 CORBA客户端程序设计4.1 问题描述与IDL定义4.2 CORBA客户端的组成4.3 ORB的初始化4.4 ORB接口的功能4.5 分布式对象引用的获取4.5.1 利用文件的方法获取对象引用4.5.2 利用Binding服务的方法获取对象引用4.5.3 利用命名服务的方法获取对象引用4.5.4 利用factory对象的方法获取对象引用4.6 Stub类的构造4.6.1 代理对象的概念4.6.2 分布式对象引用与本地对象引用的区别4.6.3 Stub类的构造4.7 org.omg.CORBA.Object接口4.8 分布式对象方法的启动4.9 JavaApplet中的CORBA客户端结构4.9.1 JavaApplet中的CORBA客户端结构4.9.2 ORB的初始化4.9.3 分布式对象引用的获取4.9.4 在HTML文件中使用Applet第5章 CORBA服务器程序设计5.1 CORBA服务器的构造5.2 对象适配器的作用5.2.1 对象适配器的作用5.2.2 BOA与POA5.2.3 伪对象5.3 BOA的功能5.4 分布式对象实现5.4.1 Skeleton继承方式5.4.2 Skeleton类的构造5.4.3 Tie机制方式5.5 分布式对象的生成5.6 分布式对象的登录5.7 接收请求开始5.8 POA基础5.8.1 POA中的CORBA对象与Servant的关系5.8.2 POA与策略5.8.3 POA的生成5.8.4 POA策略简介5.8.5 POA管理器5.8.6 Servant管理器5.8.7 默认Servant5.8.8 基于POA的服务器程序设计第6章 动态接口6.1 通用伪接口的定义6.1.1 TypeCode接口6.1.2 NamedValue接口6.1.3 NVList接口6.2 Any类型数据的处理6.2.1 Any的功能与数据构造6.2.2 Any类型的Java映射6.2.3 Any对象的生成6.2.4 Any对基本类型数据的存取6.2.5 Any对用户定义类型数据的存取6.2.6 DynAny接口6.3 接口仓库6.3.1 接口仓库的构造6.3.2 接口仓库的接口6.3.3 对接口仓库的访问6.3.4 仓库ID6.4 动态启动接口DII6.4.1 DII程序设计过程6.4.2 Request对象6.4.3 动态启动调用请求6.4.4 返回值的取出6.5 动态骨架接口DSI6.5.1 Dynamic Implementation类6.5.2 ServerRequest接口第7章 CORBA实例7.1 JavaIDL及其应用系统开发过程7.2 环境配置7.3 CORBA实例1：一般属性和操作的定义与使用7.3.1 问题描述与IDL接口定义7.3.2 IDL到Java语言的映射7.3.3 服务器端的Java语言程序设计7.3.4 客户端的Java语言程序设计7.3.5 Java类的编译7.3.6 启动orbd7.3.7 服务器端程序的执行7.3.8 客户端程序的执行7.4 CORBA实例2：本地方法与Holder类的使用7.4.1 问题描述与IDL接口定义7.4.2 IDL到Java语言的映射7.4.3 服务器端的Java语言程序设计7.4.4 客户端的Java语言程序设计7.4.5 Java类的编译7.4.6 启动orbd7.4.7 服务器端程序的执行7.4.8 客户端程序的执行7.5 CORBA实例3：Factory对象的定义与使用7.5.1 问题描述与IDL接口定义7.5.2 服务器程序设计7.5.3 客户端程序设计7.5.4 语言映射、编译与运行7.6 CORBA实例4：利用文件方式获取分布式对象引用的程序实现过程7.6.1 IDL接口定义7.6.2 服务器程序设计7.6.3 客户端程序设计7.6.4 语言映射、编译与运行7.7 简便的程序调试方法第8章 JavaRMI技术8.1 JavaRMI远程对象调用过程8.2 远程对象8.2.1 远程接口8.2.2 远程接口的实现类8.2.3 远程对象的生成8.3 Stub与Skeleton8.4 启动RMI注册器8.5 RMI程序设计过程8.5.1 远程接口的定义8.5.2 服务器程序的实现8.5.3 客户端程序的实现8.5.4 类文件的编译8.5.5 启动RMIregistry8.5.6 运行服务器程序8.5.7 运行客户端程序8.6 基于回调技术的RMI程序设计8.6.1 服务器的远程接口8.6.2 服务器的远程接口的实现类8.6.3 客户端的远程接口8.6.4 客户端的远程接口的实现类8.6.5 异常类的定义8.6.6 Applet程序与HTML文件的定义8.6.7 定义java.policy文件8.6.8 编译与运行参考文献

<<分布式对象技术及其应用>>

章节摘录

Java语言基础Java语言是目前应用最广泛的面向对象程序设计语言之一，它具有面向对象、与平台无关、安全、稳定和多线程等优良特性。

Java语言不仅可以用来开发大型的应用程序，而且特别适合于包括Internet应用等网络程序的开发。由于本书是以Java语言为基础来描述分布式对象技术的，因此，本章将对后续章节中需要使用的Java语言的基本内容进行简单的介绍。

1.1 Java语言的特点及其程序开发过程Java语言的魅力主要体现在以下三个方面：(1)不管使用何种机器环境，只要有Java运行环境，Java的程序就可以执行。

(2)Java是一种拥有图形用户接口(GUI)和图像处理能力的新型的面向对象程序设计语言。

(3)Java语言程序可以作为web页面的一部分来使用，这不仅体现在能使web页面具有动态性的特点，而且体现在能够将Java语言程序从一台机器上快速下载到另一台机器上并运行这一强有力的功能上。

下面将从程序设计语言方面来介绍Java语言的特点，同时简要介绍一下Java语言程序的开发过程。

1.1.1 Java语言的特点从程序设计语言的角度来看，Java语言主要有如下一些主要特点。

1. 与C++语言相似Java语言是不具有C++语言中的结构体、联合(共用体)、指针、预处理器等功能的非常简单的程序设计语言，如果熟悉c++语言的话，就可以比较容易地学会Java语言。

<<分布式对象技术及其应用>>

编辑推荐

《分布式对象技术及其应用》是作者根据多年的教学经验和实践体会编写而成的，在内容编排上尽量体现易学的特点，在文字叙述上力求条理清晰、简洁，便于读者阅读。

《分布式对象技术及其应用》可以作为大专院校计算机专业研究生或高年级本科生的教材，也可以作为非计算机专业学生或软件开发人员的参考书或自学用书。

<<分布式对象技术及其应用>>

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:<http://www.tushu007.com>