

文章编号: 1005—0523(2006)05—0098—04

EJB 和 CORBA 互操作的研究及实现

易娟玲, 李正凡

(华东交通大学 信息工程学院, 江西 南昌 330013)

摘要: 简要介绍了 EJB 和 CORBA 规范, 剖析了 EJB 和 CORBA 的互操作技术及其映射规范, 详细描述了实现两者互操作的过程. 为方便读者了解 EJB 和 CORBA 互操作技术, 文章最后给出了一个实例.

关键词: EJB; CORBA; 互操作

中图分类号: TP335

文献标识码: A

1 引言

EJB 和 CORBA 是目前十分流行的分布式对象技术, 两者的应用很广泛, 且具有各自的优势. EJB (Enterprise JavaBeans) 是 SUN 公司提出的一种基于组件的开发模型, 其主要目标是由 EJB 容器负责为其中的 EJB 组件提供各类对象服务, 从而使 EJB 组件自身可集中精力处理与企业应用密切相关的业务逻辑^[1]. 一个 EJB 组件主要由三部分构成: Remote 接口和 EJB 对象、Home 接口和 Home 对象以及 EJB 组件的实现类. Remote 接口描述了该组件执行的业务逻辑, Home 接口简单地定义了建立、删除、和寻找 EJB 对象的方法. 客户端应用程序通过 Home 对象来定位、创建、删除 EJB 类的实例, 通过 EJB 对象来调用实例中的业务逻辑.

CORBA (Common Object Request Broker Architecture) 是 OMG (Object Management Group) 定义的分布式对象标准, 也是为解决分布式异构环境的异构性、互操作性、可移植性和可重用性而提出的接口标准和规范^[1]. CORBA 标准主要分为三部分: 接口定义语言 (IDL)、对象请求代理 ORB (Object Request Broker) 以及 ORB 之间的互操作协议 IIOP (Internet Inter-ORB Protocols). CORBA 的核心是 ORB, 它负责数据的打包、传送、解包以及服务对象的定位, 遵

守 IIOP 协议. 接口定义语言 IDL 根据对象可执行的操作定义对象的类型, 它是独立于编程语言、下层网络和具体实现的数据类型描述语言. ORB 产品通常会提供一个 IDL 编译器, IDL 经编译后会为每一对象接口创建相应的桩与框架. 客户端通过桩来激活远程服务对象提供的服务, 服务器端有对象适配器 OA 根据用户的请求选用相应的框架来完成服务对象实现的激活. CORBA 的实质就是客户仅要知道服务器对象发布的接口, 而不需要知道具体的实现细节.

作为两种主流的分布式对象技术, EJB 和 CORBA 在某些方面互相竞争, 同时它们又彼此补充, 可以将两者结合起来开发分布式系统. 对于 EJB, 可以利用 CORBA 跨平台、多语言的优势提供对遗留系统的封装; 对于 CORBA, 可以利用 EJB 简化服务器端的开发, 缩短开发周期, 并保持 CORBA 跨平台的特性. 因此, EJB 和 CORBA 技术相结合, 是一种很有应用前景的开发模式, 深入研究它们之间的互操作具有很重要的意义.

2 EJB 和 CORBA 的互操作技术

EJB 和 CORBA 之间的互操作就是分别由这两种方式实现的服务能够互通互用^[2], 具体要求体现在:

收稿日期: 2006—06—24

作者简介: 易娟玲 (1981—), 女, 江西宜春人, 华东交通大学在读研究生, 研究方向为中间件技术.

1) 任何语言编写的 CORBA 客户端都可以访问 EJB 中的 Bean 提供的服务。

2) 任何语言编写的 CORBA 服务器端的代码都可以访问 EJB 中的 Bean 提供的服务。

3) EJB 中的 Bean 及客户端可以访问任何语言实现的 CORBA 对象提供的服务。

主要讨论如何实现 1)、2) 的要求, 由于 CORBA 中不存在绝对的服务器端和客户端, 所以 1)、2) 的要求实际是一样的。笔者将从通信协议映射和服务映射两个方面讨论 EJB 组件和 CORBA 客户端(包括 Java 语言编写的和非 Java 语言编写的)之间的互操作。

2.1 通信协议映射

EJB 规范要求 EJB 组件及其容器使用 RMI (Remote Method Invocation) 技术, 实现分布式对象之间的方法调用, CORBA 也提供对远程分布式对象的透明访问。CORBA 和 RMI 远程调用都是通过调用各自的 API, 底层使用 IIOP 或 JRMP (Java Remote Method Protocol) 协议来实现的。IIOP 是基于 TCP/IP 的传输协议, 且 CORBA 不支持 JRMP, 所以应该选择 IIOP 作为两者的桥梁。将 JRMP 转换成 IIOP 后, RMI 对象可以应用 IIOP 协议和 CORBA 对象进行通信, 这个解决方案就是 RMI-IIOP。

在 RMI-IIOP 出现以前, 只有 RMI 和 CORBA 两种选择来进行分布式程序设计。RMI-IIOP 兼有 RMI 的简单和 CORBA 多语言性(兼容性), 克服了 RMI 只能用于 Java 的缺点和 CORBA 的复杂性(程序员可以不用掌握 IDL)的缺点, 使得程序员能更方便的编写分布式程序设计。RMI-IIOP 采用 CORBA 的 IIOP 协议, 使得 RMI-IIOP 与 CORBA 相兼容, 也就是说, 不是基于 Java 开发的客户端也可以通过 CORBA 和 EJB 进行通信。要实现这个, 必须使用至少符合 J2EE1.3 的应用服务器, 以及符合 CORBA2.3.1 或更高版本的 ORB^[3]。

2.2 服务映射

为了更好的实现 EJB 和 CORBA 之间的互操作, SUN 公司制定了 EJB 到 CORBA 的映射规范。该规范将 EJB 到 CORBA 的映射分为下面四个部分^[4]:

1) 分布映射

分布映射定义了 EJB 和 CORBA 对象之间的关系, 并规定了 EJB 规范中 Java RMI 的远程接口、类和 OMG IDL 之间的映射, 包括 EJB 中 Home 和 Object 接口到 IDL 的映射、值对象到 IDL 的映射、Java 系统异常到 CORBA 系统异常间的映射。这个规定使得

CORBA 分布对象可以访问本来不具有 CORBA IDL 的 EJB, 从而实现了 CORBA 和 EJB 之间的互操作。

2) 命名映射

命名映射规定了基于 CORBA 的 EJB 如何通过 OMG 的 COS (CORBA object service) 命名服务来定位 EJBHome 对象。JNDI (Java Naming and Directory Interface) 提供到 CORBA 命名服务的接口, CORBA 客户机既可以通过 JNDI 调用间接访问基础命名服务, 也可以通过 COS 命名 API 直接访问该服务。

3) 事务映射

EJB 事务支持依赖于 CORBA Transaction Service, 即 OTS (Object Transaction Service)。JTS (Java Transaction Service) 代表 OTS 的 Java 绑定, 它是语言中立的。基于 CORBA 的 EJB 容器必须识别 CORBA 客户机通过 OTS 接口发出的事务边界, 以及 EJB 应用程序通过 JTA (Java Transaction API) 接口发出的事务。

4) 安全映射

EJB 规范的安全方面的内容主要是访问权限控制, 要求 EJB 服务端对客户端进行安全认证。CORBA 规定了一些方法来定义表示符, 包括简单的 IIOP 方式、可靠的 IIOP 方式及 IIOP OVER SSL (SSL 是加密套接字协议)。

根据 EJB 到 CORBA 的映射规范, IDL 接口可以从 EJB 远程 Java 接口生成, Java 命名服务 JNDI 映射成 CORBA 的 COS 命名服务, 这样 EJB Home 对象就可以出现在 CORBA 命名服务中了。CORBA 客户就可以直接在 CORBA 命名服务中查询 EJB Home 对象, 并把 EJB 对象用作 CORBA 对象, 不需要其它额外工作。

3 互操作实现的过程

前面提到的 EJB 到 CORBA 的映射规范中, 后三个映射的实现对于开发者来说比较难, 可以通过应用服务器无缝实现。但这些实现对于开发者来说是不可见的, 因此开发者最感兴趣的问题是分布映射, 即 EJB 接口到 CORBA 接口的映射。EJB 是用 Java 来定义它的所有接口的, 而 CORBA 使用 IDL 语言来定义客户与调用对象之间的接口, 为了实现 CORBA 客户端对 EJB 组件的调用, 即实现 CORBA 客户端和 EJB 组件的互操作, 必须将 Java 接口转换成 IDL 接口定义, 也就是完成 Java-IDL 的映射。要开发一个能访问 EJB 组件的 CORBA 客户端, 具体步骤如

下^[5]:

1) 从 EJB 远程接口生成 IDL
为从 Java 接口生成 IDL,可以使用任何支持 Java 语言到 OMG IDL 映射规范的工具,比如 JDK 自带的 `mic` 编译器、VisiBroker 提供的 `java2idl` 工具^[6]等,其它的应用服务器也提供类似的工具.

2) 简化生成的 IDL
生成的 IDL 比较复杂,要实现所有生成的接口和值类型是很费时间的,但我们并不需要所有的生成的接口,有些方法我们可能不会用到.所以,我们可以根据需求简化生成的 IDL 接口文件,从而节省工作量.

3) 编译 IDL 接口,生成相应编程语言的桩及框架

接下来,我们需要将简化后的 IDL 接口映射到客户端的编程语言,这种映射可以通过支持 IDL 到客户端编程语言映射规范的编译器完成,比如 VisiBroker for Java 提供的编译器 `idl2java` 可将 IDL 映射到 Java 语言^[6].因为我们只是开发客户端,所以不需要服务端的框架文件.

4) 确定如何使用 JNDI 作为 CORBA 的命名服务

在具体实现客户端之前,需要特别注意的是,EJB 采用 JNDI 命名服务,这和 CORBA 的 COS 命名服务是不同的.而发布 EJB 时,已经将 Bean 的信息经注册到应用服务器的 JNDI 上,所以 CORBA 客户端必须访问 JNDI 命名服务.有些应用服务器提供了一个跟 CORBA 命名服务相兼容的接口去访问 JNDI,比如 `WebLogic`、`JBoss` 等,因此我们可以得到对象的 IOR(interoperable object reference),通过 IOR 直接连接到 JNDI 上.

5) 实现 CORBA 客户端
现在,我们就可以采用某种语言开发一个 CORBA 客户端,该客户端可以调用 EJB 中定义的方法.

6) 运行 CORBA 客户端
CORBA 客户端实现后,我们就可以运行客户端,步骤为:首先启动 EJB 使用的应用服务器;接着部署 EJB 到应用服务器上;最后启动客户端程序.

4 实例

在一个基于 EJB 技术的物流信息系统的基础上,用 Java 语言开发了 CORBA 客户端,实现了 EJB 和 CORBA 的互操作.该系统包括基础设置模块、接

单模块、分单模块和调度模块四大模块,系统模型如图 1 所示.

由于篇幅所限,下面将以接单模块的部分代码为例说明 EJB 接口如何映射成 IDL 语言.

```
package dissertation.jiedanmokuai.ejb.facade;  
import java.rmi.*;  
import javax.ejb.*;  
.....  
public interface JieDanMoKuaiFed  
extends EJBObject {  
    public boolean addKeHuGuanLi ( JD-KeHuGuanLi  
vo) throws RemoteException;  
    public boolean delKeHuGuanLi (KeHuGuanLiEntPK  
pk) throws RemoteException;  
    .....  
}
```

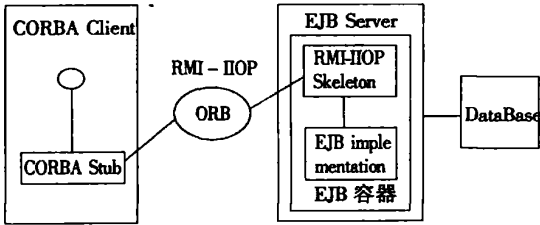


图 1 系统模型图

相应的 IDL 定义如下:

```
model dissertation {  
    model jiedanmokuai {  
        model ejb {  
            model facade  
            interface JieDanMoKuaiFed: :: javax::ejb:: E-  
JBObject {  
                boolean addKeHuGuanLi(in :: javax::ejb:: EJBOb-  
ject arg0);  
                boolean delKeHuGuanLi(in :: javax::ejb:: EJBOb-  
ject arg0);  
            };  
        };  
    };  
};
```

5 小结

分布式对象技术是目前很流行的技术,通过解决 EJB 和 CORBA 的互操作问题,可以实现 CORBA 客户对 EJB 的访问,并能充分利用已有的 EJB 或 CORBA 资源,提高软件的可重用性.

参考文献:

[1] 李文军,周晓聪,李师贤.分布式对象技术[M].北京:机械工业出版社,2004.

[2] 赵银花,侯迪,齐勇.EJB 和 CORBA 互操作的实现[J].计算机工程与应用,2003,32:148—150.
[3] 李慧,陶培基,李敏.CORBA、EJB 及其互操作技术[J].计算机工程与科学,2005,27(8):95—97.
[4] Sun Microsystems.Enterprise JavaBeans TM to CORBA Mapping[R].1999—10.
[5] Seven steps to build a VisiBroker C++ CORBA Client for an EJB Server[EB/OL].<http://www.borland.com/devsupport/appserver/faq/ejbcpp/ejb-cpp.html>.
[6] Borland/Inprise.VisiBroker for Java 4.0;Programmer's Guide and Reference[R].2000.

Research and Implementation of Inter-operation between EJB and CORBA

YI Juan-ling, LI Zheng-fan

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: It simply introduces EJB and CORBA specifications, explains the EJB and CORBA inter-operation techniques and the EJB to CORBA mapping, and then describes the implementation procedure of inter-operation between the two. In the end, the paper gives a instance for better understanding of the EJB and CORBA inter-operation techniques.
Key words: EJB; CORBA; inter-operation

(上接第 93 页)

Modeling of Stray Capacitor in Inductors

YUAN Yi-sheng

(School of Electronical and Electrical Engineering East China Jiaotong University, Nanchang 330013, China)

Abstract: An analytic approach is applied in modeling stray capacitor in inductors. The stray capacitor of basic part of inductors is deduced, and the complete capacitor model is built up based on the approach. This model is applied in analyzing several inductors. In the end, the comparison with measurement and predicted result verify the model.
Key words: analytic approach; stray capacitor; modeling