

CORBA体系在网络聊天系统中的实现

CORBA architecture of the system in the network chat

(山东省烟台大学)马晓敏 赵 蓬
MA XIAOMIN ZHAO PENG

摘要:CORBA(Common Object Request Broker Architecture)是分布计算机技术的发展结果, CORBA 技术在理论和技术上扩展了客户/服务器的模式。本文在分析 CORBA 体系架构的基础上,提出了基于 CORBA 体系架构的网络聊天系统的实现,并给出部分关键代码。

关键词:CORBA; 中间件; IDL 接口; 网络聊天

中图分类号:TP312

文献标识码:A

Abstract:CORBA, Common Object Request Broker Architecture is the development results of Distribution of Computer Technology, CORBA technology expanded Client / Server model from the theoretical and technical. CORBA systems in the framework of this analysis, on the basis of the proposed framework base on CORBA system network chat system to achieve, and given some critical code.

Key words:CORBA, Middleware, IDL interface, network chat

现今分布式系统领域中最重大的成就之一就是创建的与编程语言无关的、标准的编程系统——CORBA(Common Object Request Broker Architecture),通用对象请求体系架构。

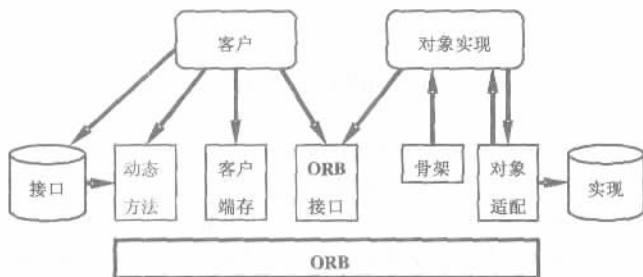


图1 CORBA 体系架构

CORBA 并不是一门编程语言,而是一个中间件平台。它代表的是分布式系统编程中的基础设施,不能直接用来编程。这种体系结构为开发大型的分布式异构应用软件系统提供了独立于硬件平台、独立于编程语言、独立于操作系统和独立于网络传送协议的平台。它定义了对象之间通过对象请求代理的发送请求以及接收响应的机制,保证了在异构系统中对象之间的互操作性。具体体现在用接口定义语言 IDL (Interface Definition Language)进行对象建模,以及采用 IIOP(Internet Inter-ORB Protocol),即互联网 ORB 互通协议进行协同用户和服务器之间的通信。CORBA 将面向对象技术与分布式计算相结合,提供了软总线机制。CORBA 技术的核心作用就是为异构环境下的分布式系统中分布在不同节点上的对象提供一个对象总线以及相应的总线服务。分布式系统中各分布对象只要按 CORBA 规定的接口方法接上软总线,便可以安全方便地实现对象间的互操作性。

1 CORBA 体系架构

如图1所示,CORBA 不是一个软件,而是一种软件体系架

构,由不同组件组成。

架构的核心是对象请求代理(ORB),它负责接收一个 CORBA 对象或者客户对另一个 CORBA 对象(服务器)提供的服务请求,负责定位并找到服务器对象,转发方法调用给它,并将调用的结果传回给客户对象。对于客户对象来说,确定服务器对象的过程是透明的。客户对象并不直接寻址 ORB,在客户端有一个客户存根对象,它提供和服务器对象一样的接口,由它负责和 ORB 通信。

CORBA 的各种标准操作都可以通过 ORB 来直接调用,包括启动和初始化 ORB 的操作。接口库包含了所有登记了的对象的接口说明,包括它们的方法、属性等。

实现库提供了名字服务,用来查询哪些类在服务器端得到了支持,哪些类能够实现实例化并可以做引用管理。一旦一个客户获得了另外一个对象的引用,它就可以调用这个对象所提供的操作。

2 IDL

接口定义语言 IDL 是用来将对象和它们的方法以及属性联合在一起的说明语言。IDL 是一种定义接口的语言,与实际的编程语言无关。这个特性使得 IDL 技术在 CORBA 体系架构中充当了核心“角色”。IDL 使客户和服务器端采用不同的编程语言开发成为了可能,如图2所示。

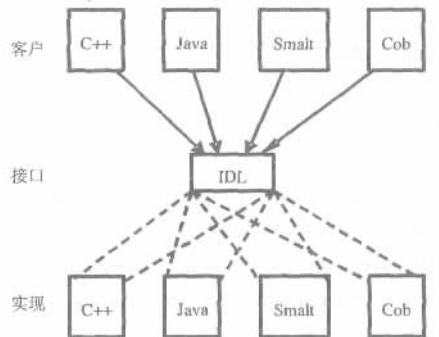


图2 IDL 实现结构

马晓敏: 硕士

基金项目:山东省青年基金(JS0528)

IDL接口定义好后,就可以由IDL编译器来处理,产生所需要的输出。

3 IDL—Java 映射

由于CORBA采用IDL语言来规定,在实际应用里必须将其映射成实际编程语言的接口,本文中是映射到Java语言。

4 基于CORBA架构的聊天系统实现

CORBA是一个典型的C/S基础设施,因此,我们的聊天系统也采用了C/S架构。在服务器端创建一个ChatServer对象,提供了Login()和Logout()操作。其他独立的ChatClient对象可以登录到这个对象或从这个对象中注销。和其对应服务的IDL接口定义如下:

```
Module Chat {
    typedef string Name;
    struct Message {
        Name name;
        String message;
    };
    exception UnknownName {};
    exception Reject { string reason ;;};
    interface ChatClient {
        void receiveEnter(in Name name,in ChatClient client);
        void receiveExit(in Name name);
        void receiveMessage(in Message message);
    }
    interface ChatServer {
        const short MaxClients=50;
        readonly attribute short numofClient;
        void login(in Name name,in ChatClient client)raises(Reject);
        void logout(in Name name);
        void send(in Message message);
    };
};
```

登录时,客户必须指定名字和一个针对自己对象实例的引用。登录以后,客户可以使用send()方法来发送消息,这条消息由包含客户名字和字符串message组成。每个客户在其他客户登录和注销时都会得到通知,并且还可以接收其他客户发送的消息。

这样一个IDL接口定义好后,由IDL编译器产生如下输出:

(1)客户存根类。用于将程序语言的结构映射成IDL语言中的结构。部分代码如下:

```
public class _st_ChatClient extends org.omg.CORBA.portable.
ObjectImpl
implements Chat.ChatClient {
    .....
    public void receiveMessage(Chat.Message message) {
        try{
            org.omg.CORBA.portable.OutputStream_output=
            this._request("receiveMessage",true);
            Chat.MessageHelper.writer(_output,message);
            org.omg.CORBA.portable.InputStream_input =this._invoke
            (_output,null);
```

```
} catch (org.omg.CORBA.TRANSIENT_exception){
    receiveMessage(message);
} } }
```

(2)服务器骨架类。负责解包从客户存根类对象发来的信息并把它传给被调用的服务器对象。关键代码如下:

```
abstract public class _ChatServerImplBase extends org.omg.
CORBA.portable.Skeleton
implements Chat.ChatServer{
    .....
    public static boolean _execute (Chat.ChatServer _self,int
_method_id,
    org.omg.CORBA.portable.InputStream _input,org.omg.CORBA.
portable.OutputStream _output){ switch(_method_id){
        case 0:
            try {
                java.lang.String name;
                name=_input.read_string();
                Chat ChatClient client;
                client=Chat.ChatClientHelper.read(_input);
                _self.login(name,client);
            } catch (Chat.Reject _exception){
                Chat.RejectHelper.write(_output,_exception);
                return true;
            }
        case 1:
            .....
            throws new org.omg.CORBA.MARSHAL();
        } }
```

(3)创建对象的实现代码框架,即具体的实现代码。

通过以上客户存根类和服务器骨架类,就可以创建具体的聊天系统实现代码,为了能够采用CORBA架构的方法来启动Chat Server,对于每个服务都需要一些类似的步骤,写入服务的main()方法。由于篇幅关系,本文中不再将具体代码列出。

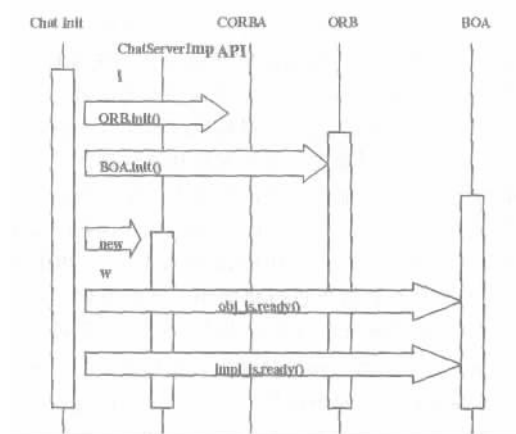


图3 说明了客户在服务器端的登记过程。

首先,调用静态方法ORB.init(),该方法启动并初始化ORB,返回一个ORB引用。负责激活对象实例的对象适配器BOA(Basic Object Adapter)也必须初始化。接下来,创建ChatServer实例并把它登记到ORB中,调用BOA的impl.is_ready()方法后,就可以等待对这个方法的调用了。(下转第23页)

个 Microsoft.Web.Services.Security.UsernameToken 作为输入参数,该方法返回指定用户的密码。可以使用任何机制保存有效的用户名/密码对,然后提供了一个实现 IPasswordProvider 接口的类,让 WSE 访问该特定密码存储机制,通过用户名/密码来验证一个用户。



图 2 年度评估界面

* Web 服务访问控制。为了保证 Web 服务调用的合法性,需要对这些服务的访问进行控制,为此建立了基于角色的 Web 服务访问控制表如图 3 所示。

	Web 服务 1	Web 服务 2	Web 服务 n
角色 1	Yes	Yes		No
角色 2	Yes	No		Yes
.....				
角色 n	No	Yes		No

图 3 基于角色的 Web 服务访问控制表

对于某一个服务请求,将根据其提供的用户/密码来验证此用户是否合法,如果合法的,则查询服务访问控制表看其所属的角色是否有权访问此服务。

* SOAP 消息安全性设计。本系统利用 WSE 对 SOAP 报文使用对称密钥进行部分加密。首先由客户端给服务发送一个简单的 Web 服务请求,该请求将返回一个由三元 DES 对称加密算法(使用了一个共享密钥和一个初始向量,IV)加密过的 XML 文档,当客户端收到了加密后的应答信息后,SecurityInputFilter 类将调用一个在客户端的解密密钥提供程序,来访问客户端上相同的共享密钥,以此来对报文进行解密,利用此方法我们对诸如评分计算结果等类型的消息进行了加密,保证了该类数据的私密性。

5 结语

“基于 Web 服务的海事系统执法人员绩效评估信息系统”的开发得到了交通部海事局的资助,并且已经在宁波海事局,南通海事局,北仑、三江口海事处试运行了两年,运行结果表明,本系统不仅提供了一种科学、客观、公正、快捷的绩效评估方法,同时由于使用了 Web 服务这种分布式计算和处理技术,也有效地解决了与海事系统内或跨系统的其他应用系统的集成,交互以及可重构问题,从而提供了一种科学化和规范化的管理模式。

本论文的创新点在于研究了基于 Web Service 的管理信息系统,提出了其设计思路和体系结构,同时理论联系实践,将其

成功应用于海事系统执法人员绩效评估中,并就 Web 服务设计粒度等关键技术进行了讨论。

参考文献

[1]麻昌德,易高翔.使用 ASP.NET/ Web 服务实现新闻发布系统[J],2004,21(10):42-43
[2]W3C. Web Services Architecture Requirements [EB/OL]. <http://www.w3.org>, 2002-09-21.
[3]Ramesh Nagappan,Robert Skoczylas ,Rima Patel Sriganesh Developing Java Web Service[M].John Wiley & Sons ,Inc. 2003: 36-37
[4]Dan Foody. 调整合适的 Web 服务粒度。 <http://searchweb.services.techtarget.com.cn/42/2103042.shtml>
[5]徐忻,吴介.Web 服务结构模型的研究与实现[J]微计算机信息,2006 5-3: 103-105
作者简介:高曙(1967-),女,安徽芜湖人,武汉理工大学计算机学院副教授,硕士生导师,主要研究方向:分布式计算。
Biography:Gao Shu (1967-),Female, Wuhu, Anhui Province, School of Computer, Wuhan University of Technology, Associate professor, Tutor of M.Sc student, majored in distributed computing.
(430063 湖北武汉 武汉理工大学)高曙
通讯地址:(430063 湖北武汉 武汉理工大学余家头校区计算机学院)高曙

(收稿日期:2007.6.03)(修稿日期:2007.7.05)

(上接第 280 页)

当客户端调用该服务器时,将该对象引用采用文本方式输出,客户端根据这些信息来指定服务器的对象引用。

本文作者创新点:本文在分析 CORBA 体系架构的基础上,提出了基于 CORBA 体系架构的网络聊天系统的实现,并给出部分关键代码。

参考文献:

[1]孙冬梅,陈小安,罗天洪等.基于 CORBA 和 Java 的协同设计通信平台[J].重庆大学学报,2006.29(2).12
[2]王艳,薛涛,张书杰.基于 CORBA 的 JAVA 消息服务中间件的设计与优化[J].计算机信息技术.2006.2.7
[3]MarKo Boger(德),《Java 与分布式系统》曹学军.北京:机械工业出版社,2003.
[4]姜国庆,黄永忠,牛向华.网络环境下数据传输模型的研究,微计算机信息,2005,10-3: 103-105
作者简介:马晓敏(1972-),女(汉),黑龙江省木兰县人,讲师,硕士,主要研究领域为网络程序设计。
Biography:Ma Xiaomin (1972-), Female (Han) Lecturer in College of Computer, Yantai University,, Research interesting: Networks Programme Designing.
(264005 山东省 山东省烟台大学计算机学院)马晓敏
(264005 山东 山东省烟台大学经管学院)赵蓬
(College of Computer, Yantai University, Shan dong, 264005) Ma Xiao-min
(College of Economic Management, Yantai University, Shan dong)Zhao Peng
通讯地址:(264005 山东 山东省烟台市烟台大学计算机学院)马晓敏

(收稿日期:2007.6.03)(修稿日期:2007.7.05)