

CORBA 与 Web Services 应用集成网关的研究与实现^{*}

罗 真 周 刚 康建初

(北京航空航天大学软件开发环境国家重点实验室 北京100083)

摘 要 针对 CORBA 应用在 Internet 范围内进行数据共享和应用集成的局限性,研究了 CORBA 与 Web Services 应用集成的方式,设计实现了 CORBA/Web Services 应用集成网关,深入探讨并初步解决了应用集成中的接口转换、数据类型转化、CORBA 对象调用等关键问题。

关键词 CORBA, Web Services, 企业应用集成, 分布式计算, SOAP, WSDL, IDL

A Gateway for CORBA/Web Services Application Integration: Research and Implementation

LUO Zhen ZHOU Gang KANG Jian-Chu

(Beijing University of Aeronautics & Astronautics, Beijing 100083)

Abstract To solve the problem of data sharing and application integration for those CORBA-based application, this paper investigates the method for CORBA/Web Services application integration, designs a CORBA/Web Services application integration gateway, thoroughly discusses three key problems within the application integration process which include interface translation rules, data type transformation method and CORBA invocation strategy.

Keywords CORBA, Web services, EAI, Distributed computing, SOAP, WSDL, IDL

1 引言

CORBA(Common Object Request Broker Architecture)是OMG(Object Management Group)组织提出的分布式对象应用规范,已经在电信、金融等领域中得到了广泛应用^[1]。然而 CORBA 在基于 Internet 的分布式应用上仍存在一定的局限性,主要表现在以下几个方面:(1)通讯协议 IIOP 使用非标准端口,一般难以穿越防火墙;(2)客户端都必须安装 ORB 环境,不易于扩展;(3)CORBA 与 J2EE、DCOM 等分布式应用系统之间难于实现互操作,因而基于 CORBA 的应用在 Internet 环境下难以进行数据共享和应用集成,容易形成信息孤岛。

近年来新兴起的 Web Services 技术为实现基于 Internet 的分布式应用和资源共享提供了解决方案。Web Services 是“分布于 Internet 之上,可以通过 Internet 标准协议进行访问和使用的,具有松散耦合特性的可重用软件组件”^[2]。它采用面向服务的体系结构,使用 HTTP 等通用 Internet 协议和 XML 编码进行消息传输。XML 的自描述性和可扩展性以及 HTTP 使用的广泛性和对防火墙的透明性,使得基于 Web Services 能够轻松实现异构系统的应用集成。

Web Services 在互操作性等方面的优点使其在企业应用集成、电子商务等领域得到了越来越广泛的应用,实现 CORBA 和 Web Services 的应用集成也变得越来越重要,它一方面使 CORBA 应用可以进行跨越防火墙的通讯,另一方面使 CORBA 能与 J2EE、DCOM 等系统基于 Web Services 接口进行互操作,因而能在最大限度保护现有投资基础上实现企业异构系统的集成。

本文研究了 CORBA 和 Web Services 应用集成的方法,

设计实现了 CORBA/Web Services 应用集成网关。在网关中通过定义接口转换规则实现了将 CORBA 应用发布成 Web Services 服务;使用类型二元组解决了数据类型转化二义性的问题;使用 CORBA 命名服务和动态调用机制保证对 CORBA 对象得正确定位和调用。

2 CORBA 与 Web Services 应用集成方式

CORBA 与 Web Services 应用集成的方式主要有两种:一种是 Web Services 客户端访问 CORBA 服务器,即服务器端使用 Web Services 接口封装 CORBA 对象的操作,对客户屏蔽实现细节。另一种集成方式是 CORBA 客户端访问 Web Services 服务器,即使用 CORBA 接口封装 Web Services 服务,客户端以访问 CORBA 对象的方式访问一个或多个 Web Services 服务。

企业应用集成通常通过封装遗留的应用程序对外提供服务,所以第一种集成方式更为通用。本文中主要讨论的是 CORBA 与 Web Services 第一种应用集成方式,其工作原理如图1所示,主要包括服务部署和服务运行两个过程。服务部署是指将 CORBA 对象的接口描述转换成 Web Services 的接口描述,同时在网关中注册 CORBA /Web Services 服务映射信息。在此过程中,Web Services 的接口描述使客户端知道服务器端提供服务接口,服务映射信息将作为服务运行时协议转换的依据。服务运行是指网关完成 Web Services 的通讯协议 SOAP 和 CORBA 通讯协议 IIOP 之间的协议转化,将 SOAP 请求中的参数转换成 CORBA 对象调用需要的参数,定位和调用 CORBA 对象,同时将结果封装成 SOAP 消息返回。协议转换过程中需要使用 CORBA /Web Services 服务映射信息,以完成参数校验、对象定位的功能。

^{*} 本课题得到国家973重点基础研究规划项目“基于先进网络的大型网络管理示范系统”资助(批准号: G1999032709);得到国家863高技术研究发展计划项目“面向电信运营管理的中间件平台”资助(批准号:2002AA113090)。

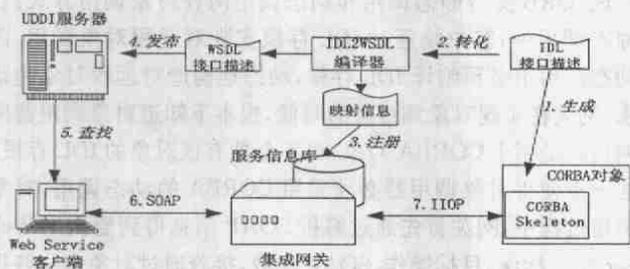


图1 CORBA/Web Services 应用集成网关

3 网关系统结构

在CORBA与Web Services进行应用集成过程中,应用集成网关的主要功能是注册CORBA对象并对外发布Web Services接口,在服务调用时完成SOAP协议和IIOP协议之间的协议转化。应用集成网关系统结构如图2所示。其中,网关前端从Web服务器接受SOAP消息,后端通过ORB总线连接CORBA服务器,SOAP消息处理被分为了三个层次:传输层、服务层、调用层。其中传输层主要功能是屏蔽不同传输协议的差异,使后续的消息处理过程对传输协议透明;服务层定义了安全、日志等系统服务;调用层完成SOAP消息解析和对CORBA对象的调用。

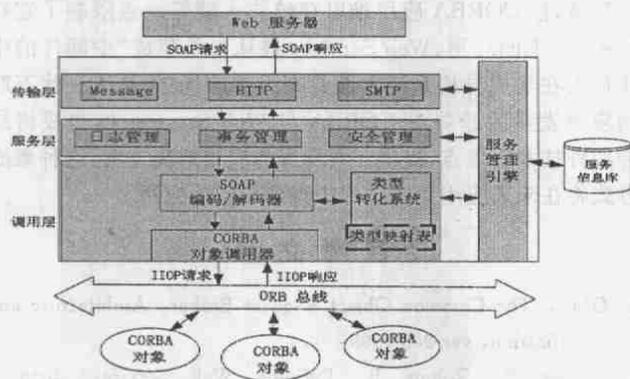


图2 CORBA/Web Services 应用集成网关系统结构

网关对服务调用的处理过程如下:首先,服务管理引擎在系统初始化时从服务信息库读入配置文件,建立Web Services服务对象池,并创建该服务在传输层和服务层的信息处理对象。当客户端SOAP请求到达时,传输层从Web服务器获得与传输协议相关的数据包,从中提取SOAP请求体;服务管理引擎根据SOAP请求中的Web Service标识获得服务对象,并在服务层进行安全和事务处理;接着SOAP编码/解码器解析出SOAP请求中的目标操作和参数,并调用类型转化系统完成SOAP参数到CORBA参数的类型转换。最后CORBA调用器根据目标操作和转换后的CORBA参数,通过ORB总线定位和调用CORBA对象。调用结果被返回到SOAP编码/解码器,封装成SOAP响应返回给客户端,从而完成了一次调用过程。

4 网关关键问题

应用集成网关作用相当于从Web Services到CORBA的单向桥接系统,其中涉及的关键问题主要有:

1. 接口转换规则:接口转换规则定义了CORBA接口和Web Services接口之间的映射关系,同时该规则保证客户端根据Web Services接口生成的SOAP调用消息能在协议转

换时正确地映射成IDL中定义的操作。

2. 数据类型转化:IDL提供了丰富的数据类型,SOAP使用XML Schema规范提供的数据类型相对要简单得多。服务运行时刻网关需要将SOAP请求中的WSDL数据类型转换为IDL类型,并将IIOP应答消息中的IDL类型封装为WSDL类型,数据类型转化机制保证类型转化的正确性。

3. CORBA对象调用:网关对客户端提供Web Services访问接口,相当于SOAP服务器,但对于运行CORBA对象的服务器来说,网关的作用却相当于发起调用的客户端。CORBA对象调用机制保证网关根据SOAP消息中的服务标识、参数等内容能正确调用相应的CORBA对象。

4.1 接口转换规则

CORBA接口定义语言IDL采用了与C++语言类似的语法,主要包括类接口定义、操作定义、类型定义等几个部分。Web Services的接口定义语言WSDL采用了基于XML的语法,其接口定义内容包括服务接口定义和服务实现定义两部分,其中服务接口定义描述服务提供的操作及操作调用时的输入输出的消息,服务实现定义描述服务的协议绑定、地址绑定等实现信息,二者结合构成了一个完整的服务描述。在接口转换过程中,IDL接口定义与WSDL定义中的服务接口定义部分具有映射关系,因此可以按照转换规则进行相应转换,而WSDL中的服务实现定义与网关的具体实现相关,必须通过用户输入获得。接口转换的主要规则如下:

1. 类接口(Interface)定义:类接口在IDL定义中通常做为操作的集合,相应转换为WSDL定义中的起同样作用的端口类型(PortType)元素,类接口中包含的操作转换为对应端口类型中的操作。值得注意的是,IDL定义中提供了类接口的继承机制,而WSDL不支持继承关系定义,本文解决方法是在WSDL中将父类接口中定义的操作复制到子类接口中,从而使子类接口也具备了父类接口全部的操作,而父类接口中的类型定义不进行复制,因为在WSDL中父类接口的名称空间对子类接口来说仍然是可用的,子类接口可以通过父类接口的名称空间直接引用父类接口中定义的类型。

2. 操作定义:IDL中的操作直接转换成WSDL中的操作,IDL操作的In和Inout参数构成WSDL操作中的输入消息,IDL操作的out和inout参数和返回值构成WSDL操作中的输出消息,IDL操作中定义的异常构成WSDL操作的出错消息。

表1 IDL/WSDL 类型映射表

	IDL 类型	WSDL 类型	IDL 类型	WSDL 类型
简单类型	Any	anyType	Long	Int
	boolean	boolean	long long	Long
	Char	string	short	Short
	wchar	string	string	String
	double	double	wstring	String
	float	float	unsigned short	unsignedShort
	octet	unsignedByte	unsigned long	unsignedInt
复杂类型 (struct 类型为例)	struct PersonalData{ string Name; short Age; }		{ complexType ame = " PersonalData" (element name = " Name" type = " xsd:string" /> (element name = " Age" type = " xsd:short" /> /</complexType	

3. 类型定义:IDL简单类型定义可以直接转换成WSDL

简单类型定义,IDL 复杂类型定义可以转换为 WSDL 复合类型定义。类型定义的映射关系表如表1所示。需要注意的是:IDL 对象引用类型可以转换为 WSDL 字符串类型,Web Services 客户端可以将对象引用当成普通的字符串标识,而网关能识别对象引用并能将它实例化为具体的 CORBA 对象。

4.2 数据类型转化

数据类型转化是指网关在处理服务调用时进行的 SOAP 参数和 CORBA 参数之间的数据类型相互转化,其基本要求是类型转化准确,否则服务调用就会产生参数错误。然而受 WSDL 中数据类型限制,接口转换规则定义的类型映射关系中在一对多的映射关系。例如 WSDL 中的 String 类型对应 IDL 中的 Char,String 等多个类型,因此网关进行数据类型转化时存在二义性。

分析 SOAP 调用机制可以发现,服务器通常需要先从前 Web Services 接口定义中查询到相应参数类型,然后再做类型转化,因为 SOAP 消息为了尽量减少网络传输负载,消息体中通常只提供参数的值而没有直接提供的数据类型描述。因此可以在服务器端存储一定的类型描述信息,辅助完成参数验证和类型转化的功能。

本文通过在网关中存储类型二元组描述的方法解决数据类型转化二义性问题。首先网关进行 CORBA 对象注册时,使用 (IDL 类型,WSDL 类型)的类型二元组描述 CORBA 对象操作中的参数的数据类型,同时将信息存储在服务器端。(IDL 类型,WSDL 类型)二元组共同决定一对类型编码器和类型解码器,完成参数值在两种类型之间的转化,因而消除了二义性的问题。当网关处理 SOAP 消息调用时,首先通过查询类型映射信息获得某个特定参数的类型二元组,继而生成类型编码/解码器,完成参数值在 WSDL 和 IDL 两种类型之间的转换。

4.3 CORBA 对象调用

CORBA 对象调用是指根据 SOAP 消息中解析出的客户请求的方法名和参数,去调用计算任务的实际承载者,通常包括对象定位和对象调用两个过程。

CORBA 对象定位即获得 CORBA 对象引用。客户端使用 SOAP 消息中包含的 Web Services 标识表明要访问的目标服务,而不知道对象实现细节,因此网关必须维护 Web Services 标识和 CORBA 对象引用之间的映射关系。通常可以使用一张表存储两者的直接绑定关系,但这种方法使客户端和服务端耦合紧密,可扩展性差。本文采用了 CORBA 提供的命名服务解决上述问题。在服务部署阶段,网关记录服务的 Web Services 标识和实现该服务的 CORBA 对象逻辑名的绑定;服务调用时,网关先根据 SOAP 消息中的 Web Services 标识查询到 CORBA 对象逻辑名,然后运行命名服务解析该对象逻辑名,从而得到目标对象的引用。网关维护了一个 (Web Services 标识,CORBA 对象逻辑名,CORBA 对象引用)三者的映射关系,使服务器端的 CORBA 对象实现对客户端透明。

CORBA 对象调用可以利用 ORB 的提供的调用接口来

完成。ORB 支持静态调用和动态调用两种对象调用方式:在静态调用中,客户端通过 IDL 存根实现对远程对象调用,而动态调用不必预编译 IDL 存根,动态地构造对远程对象的请求。网关在实现对象调用器的时候,根本不知道对象调用器所调用的是哪个 CORBA 对象,更不会拥有该对象的 IDL 存根。这一点使得对象调用器必须采用 CORBA 的动态调用。对象调用过程中,网关首先通过解析 SOAP 消息得到要调用 Web Services 标识、目标操作、SOAP 参数,接着通过对象定位获得目标对象引用,通过类型转化获得 CORBA 参数,最后将目标对象引用、操作、CORBA 参数动态构造造成 IIOP 请求,发送到 CORBA 服务器,从而完成对象调用。

结束语 本文实现了一个 CORBA/Web Services 应用集成网关。考虑到异构系统互操作的需要,网关使用 JAVA 语言进行开发,开发环境如下,操作系统:Window98(客户端),Window 2000 Server(服务器端);Web 服务器:Tomcat4.0.3;开发工具:Jbuilder7, Xerces 解析器。网关目前支持 SOAP 协议1.1版本,WSDL 协议1.1版本,CORBA 2.3标准。在 Windows 平台的使用.Net Application Center 对网关进行的测试表明:对于 CORBA 对象提供的简单计算服务,网关平均每秒处理请求数216个,服务平均响应时间为5.33ms,系统具有较高的执行效率。

目前,CORBA 已经广泛的应用于电信、保险、银行等多个领域,但 CORBA 应用难以穿越防火墙等缺点限制了它在 Internet 上的应用。Web Services 被认为是集成“中间件的中间件”,在集成异构系统方面具有天然的优势,是下一代互联网应用发展的趋势^[6]。CORBA 与 Web Services 的集成将结合两种技术的优点,提供一种良好的应用解决方案,这种集成方式将在未来应用集成过程中得到广泛的应用。

参考文献

- 1 OMG. The Common Object Request Broker: Architecture and Specification, ver 2.3 1999
- 2 Sleeper B, Robins B. Defining Web Services. <http://www.stencilgroup.com/>
- 3 OMG. CORBA Web Services: OMG TC Document orbos/2001-06-07
- 4 OMG. CORBA to WSDL/SOAP Interworking Specification: OMG document number prc/2003-01-14
- 5 W3C. SOAP Version1.2. <http://www.w3.org/TR/soap12/>
- 6 Gokhale A, Kumaraud B, Sahuguet. Reinventing the Wheel? CORBA vs. Web Services. In: Eleventh Intl. Conf. on World Wide Web (WWW2002), Honolulu, Hawaii, May 2002
- 7 侯孟书,赫俊民等. XML 与 CORBA 的集成. 电子科技大学学报, 2002,31(5):530~533
- 8 Martin D,等著. XML 高级编程. 李喆译. 北京:机械工业出版社,2000
- 9 Michi H, Steve V 著. 基于 C++CORBA 高级编程. 徐金梧译. 北京:清华大学出版社,2000
- 10 R.otte, Patrick P, Roy M 著. CORBA 教程. 李师贤译. 北京:清华大学出版社,2000