广西科学院学报
Journal of Guangxi Academy of Sciences
2006,22(3):161～163
Vol. 22,No. 3 August 2006

基于 Web Services 的企业电子商务应用集成技术的设计与实现

Design and Realization of an Enterprise E-Business Application and Services Integration Based on Web Services

杨志和,李业荣
YANG Zhi-he,LI Ye-rong

(广西大学计算机与电子信息学院,广西南宁 530004)
(School of Computer and Electronics Information, Guangxi University, Nanning, Guangxi, 530004,China)

摘要:在分析 Web Services 的体系结构和关键技术的基础上,采用基于 J2EE 的 Web Services 技术,结合标准的 Web 协议(HTTP、SMTP 等)和 XML、SOAP、WSDL、VDDI 等一系列标准,给出一个企业电子商务应用集成技术的设计与实现。

关键词:Web 服务,企业应用集成,J2EE
中图法分类号:TP311 **文献标识码**:A **文章编号**:1002-7378(2006)03-0161-03

Abstract: The architecture and key technologies of Web Services are analyzed. Web Services technology based on J2EE and the standard Web agreement (HTTP SMTP,etc) and a series of standards (XML, SOAP, WSDL, UDDI, etc) are used to design and realize an enterprise e-commerce application integration.

Key words: Web services,EAI,J2EE

在电子商务市场中,要求所有的参与者都采用基于某种语言和平台的模型是不现实的。Web Services 是一种能够被描述并通过网络发布、发现和调用的自包含、自描述、松散耦合的软构件,它结合了面向组件方法和 Web 技术的优势,利用标准网络协议和 XML 数据格式进行通信,具有良好的普适性和灵活性,在 Internet 这个巨大的虚拟计算环境中,任何支持这些标准的系统都可以被动态定位以及与网络上的其它 Web Services 交互,任何客户都可以调用任何服务而无论它们处在何处,突破了传统的分布式计算模型在通信、应用范围等方面的限制,允许企业或个人快速、廉价建立和部署全球性应用[1]。Web Services 以其适用于极端异构的 Internet 环境而受到软件界的一致青睐,正在慢慢渗透入企业中。对大公司的开发者来说,Web 服务已经深入人心。大部分开发者已经在把 Web 服务标准——XML, WSDL,SOAP 和 UDDI 集成到应用软件中去。

本文采用基于 J2EE 的 Web Services,结合标准的 Web 协议(HTTP、SMTP 等)和 XML、SOAP、WSDL、UDDI 等一系列标准给出一个 Web Services 的设计与实现,为企业构造和发布大型的 Web 服务提供参考。

1 Web Services 体系结构

Web Services 结构如图 1 所示。在 Web Services 体系中,所有的应用实体都被抽象成服务,其中包括三个实体和三种操作[2],三个实体分别为:(1)服务提供者(Service Provider)。从商务角度看它是指服务的所有者,从体系结构上看它是指提供服务的平

收稿日期:2005-11-14
修回日期:2006-03-30
作者简介:杨志和(1979-),男,湖北麻城人,硕士研究生,主要从事电子商务研究。

台。(2)服务请求者(Service Requester)。从商务角度看它是指需要请求特定功能的企业,从体系结构上看它是指查找和调用服务的客户端应用程序。(3)服务代理(Service Broker)。它是指用来存储服务描述信息的信息库(Repository)。服务提供方在这里发布他们的服务;服务请求方在这里查找服务,获取服务的绑定信息。三种操作分别为:(1)发布。服务提供者需要首先将服务进行一定描述并发布到注册服务器上。(2)查找。服务请求方根据注册服务器提供的规范接口发出查询请求,以获取绑定服务所需的相关信息。(3)绑定(Binding)。服务请求方通过分析从注册服务器中得到的服务绑定信息,包括服务的访问路径、服务调用的参数、返回结果、传输协议、安全要求等,对自己的系统进行相应配置,进而远程调用服务提供者所提供的服务。

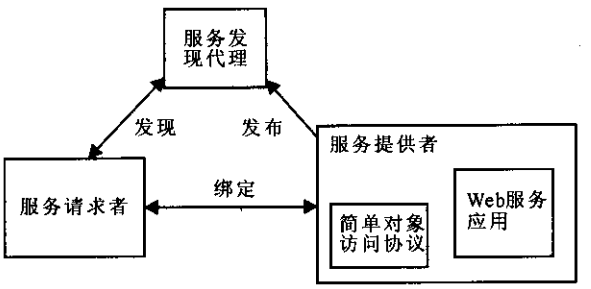

图1 Web Services结构

2 基于Web Services的企业电子商务应用集成的设计与实现

2.1 J2EE基础框架平台介绍[3]

J2EE由一整套技术和说明规范组成,每一种规范都规定了各类J2EE函数的操作方式。J2EE为基于XML的RPC(JAX-RPC)提供了Java API用以支持面向函数/方法的Web Services集成。JAX-RPC采用XML执行远程过程调用(RPC),同时为汇集和去汇集参数、传送和接收过程调用提供API。到目前为止,JAX-RPC的参考实现还依赖于SOAP 1.1和HTTP 1.1两种协议。

定义和实现基于JAX-RPC的Web Services。其Web Services的实现既可以是独立的Java应用程序也可以是Enterprise Java Bean(EJB)。JAX-RPC API可以用来创建基于SOAP的封装器以确定现有Java类或者EJB的WSDL接口。

在服务器端JAX-RPC运行时(runtime)系统上部署Web Services。其部署受到Web Service具体实现的控制。比如说,如果实现方式是EJB,那么其部署就应该在EJB容器内进行。

客户应用程序通过WSDL文档说明端口调用Web Services。对客户应用程序来说,对Web Services的调用应该形如本机方法一样调用。

2.2 企业电子商务应用集成的步骤

开发基于J2EE的Web Services或将原有的J2EE架构改为基于J2EE的Web Services,只需要在开发EJB的基础之上进行Web Services的封装即可。我们使用的开发工具为:在Win2000 Server平台下的Jbuilder7,WSDK3(Web Services Kit for Java,Version 3)和WebLogic7.0.1。具体实现如下:

(1)按照J2EE规范将系统开发完毕,即将系统所需的EJB开发完毕。

(2)启动WebLogic,将开发完毕的EJB进行部署并测试无误。

(3)在Jbuiler7中新建一个SOAP服务器。将其中WebAPP命名为Axis,ToolKit选择为Apache Axis,然后将WebLogic选为其运行时配置(Runtime Configuration)。

(4)启动Axis服务。在Axis的根目录(Root Directory)下将index.html选择用"Web Run"→"Use Web Service Server"启动WebLogic并发布Axis服务。

(5)在Jbuilder7的"Tools"→"Web Service Explorer"中,创建一个新的Axis Server,命名为SmartCard,它的URL为http://harry:7001/Axis/servlet/AxisServlet。

(6)接着点击"Display Services",选择EJBModuleSessionBean(这是在第一步时开发的SessionBean,其作用是把与客户端的接口与抽象表达业务逻辑抽取出来,使客户端通过它来访问EJB容器中的CMP,CMP用来实现数据库访问,执行单纯的数据读取和修改工作,这样可以提高组件的重用性、可维护性和可移植性)。将其WSDL URL改为http://harry:7001/Axis/services/EJBModuleSessionBean?wsdl。在IE中输入上述地址,如果能正确生成一个XML输出,说明EJB已经发布成功。

(7)最后就可以引入WSDL来访问用EJB发布成功的Web Services了。

2.3 基于Web Services的企业电子商务应用集成的实现方法

2.3.1 Web Services的创建

在编写Web服务并提供给外部应用程序通过互联网调用函数时,为使其与一般函数的声明相似,

需要在声明函数时多加<Web Method()> 关键字,让系统知道这个函数可以被其他的程序通过 SOAP 来调用。程序声明与编写如下:

```
<Web Method( )>
    public DataSet Search Book(string msg)
    {
        DataSet myDs;
        try
        {
            Hugo. Book myBook=new Hugo.
Book();
            myDs=myBook. SearchBook(msg);
            return myDs;
        }
        catch(Exception er)
        {
            throw new Exception(er. Message);
        }
    }
```

2.3.2 Web Services 的发布

每一种 Web 服务都需要一个名称空间(namespace)。所谓名称空间就是标识 Web 服务的一种附加方法[4]。如果创建了两个同名的 Web 服务,如果这两个 Web 服务在不同的名称空间内存在,调用就不会混淆。因此,在 Web 服务公开发布之前必须修改默认的名称空间。通常用自己公司的域名作为命名空间。为了发布 Web 服务以便其它人能够使用它,我们在一个可查找的公共 UDDI 目录上登记自己的服务,即注册一个帐号。UDDI 是一种开发的、与供应商无关的标准,可以通过其找到现有的 Web 服务或发布 Web 服务[5]。实际上,Web 服务并没有拷贝到 UDDI 的服务器上,UDDI 的作用是列出现有的服务,指引人们找到服务所在的服务器。在这个意义上说,它是一种真正的信息索引目录而不是存储具体信息的仓库。当然,也可以在企业的机构内引入 UDDI,在自己的企业内部安装 UDDI 服务器。由于条件、资源等问题,我们没有实现 Web 服务发布,只是在本机上进行调试。

2.3.3 Web 服务的调用

服务请求方和服务提供方都应该包含一个 SOAF 消息监听器(SOAP Listener),它专门负责 SOAP 消息的接收与发送。运行时,首先由请求方的应用程序发出服务调用请求,由客户端代理程序将该请求转化成符合 Web 服务调用所要求的格式;然后,由 SOAP 消息监听器将消息以 SOAP 请求的形式传给服务提供方;服务提供方的 SOAP 监听器收到 SOAP 请求后,由 SOAP 路由器(SOAP Router)处理该请求,并将请求转发给能处理该请求的 Web 服务应用程序,由该程序处理并返回相应结果;最后,由 SOAP 消息监听器将处理结果封装成 SOAP 响应的形式返回给客户端;服务请求方收到响应后,由客户端代理程序解析出处理结果并返回给实际的请求程序。

至此,一个 Web 服务的简单应用经过创建、发布、调用就完成了。

3 结束语

目前许多大公司已经开始使用 Web 服务来通过互联网连接公司数据库和其他公司的数据库,特别是用于改进客户服务和供应链。J2EE 作为一种成熟的、广泛应用的企业级应用解决方案和一个开放的规范,其 Web Services 应用领域主要在构造和发布大型企业的 Web 服务,在服务器端汇集 Web 服务,以及需要平台无关性的公司内部网络和外部网 Web 服务客户程序等。同时 J2EE 也在不断的更新和发展,其对 Web Services 的支持也日趋成熟,加上 J2EE 在平台的稳定性、服务器的稳定性以及产品的多样性等方面的优势,在未来很长的时间内,J2EE 都会是企业构建应用系统的重要选择,因此有着广大厂商支持的 J2EE 会在 Web Services 时代发挥越来越重要的作用。

参考文献:

[1] KREGER H. Web Services Conceptual Architecture (WSCA 1.0)[EB/OL]. (2001-05-01). http://www-3. ibm. com/software/solutions/webservices/pdf/WSCA. pdf.

[2] LUBLINSKY B, FARRELL M. Web Services-The Implementation Iceberg[J]. EAI Journal, 2002(6): 79-82.

[3] KUZYK R. Web Services: Standardizing EAI[J]. EAI Journal, 2002(4): 33-35.

[4] AOYAMA M, WEERAWARANA S, MARUYAMA H. Web services engineering: promises and challenges: proceedind 24th International Conference on Software Engineering ICSE[C]. USA, 2002: 28-30.

[5] 李劲. 动态电子商务的 Web 服务[M]. 北京:清华大学出版社, 2002.

(责任编辑:韦廷宗 凌汉恩)