基于Web Services的软件集成方法的研究与实现

胡继东 杲跃景 刘广钟 (徐州中国矿业大学计算机系 221008)

摘要：本文介绍了Web Services的原理，探讨了利用SOAP协议进行软件集成的优越性，并给出了具体示例。

关键词：Web Services SOAP XML

1 引言

软件设计思想经历了结构化设计、面向对象设计、分布式设计等过程。组件思想出现以后，程序员不需要再编写冗长的代码，而是利用现有的组件及接口进行快速的拼装。远程过程调用(RPC)技术使得位于不同地点的功能模块可以进行相互的调用。本文描述的Web Services即是利用WWW上现有的协议组(HTTP、SOAP等)，通过发布服务、调用服务的方法，进行零支出的软件集成的一种新体系结构。

2 Web Services概述

2.1 概念

Web Services一种封装了的、具有某种功能的、发布到网络上以供其他程序使用的功能集合(即组件)，Web Services是通过SOAP(简单对象访问协议)进行通信，可以本地调用，异地执行，服务提供者和服务请求者之间共用的，仅仅是服务的标准接口，而无需了解两者所处的平台，使用的语言能真正的实现软件透明集成。

2.2 原理

图1描述了Web Services的基本组成及相互操作。

Web Services的三个基本角色为：服务请求者、服务提供者、服务代理者。

Web Services组成之间的操作包括：

发布：服务提供者通过在服务代理者那里注册来配置和发布服务；

查找：服务请求者通过服务代理者找到服务；

绑定：服务请求者绑定服务提供者并使用可用的服务。

服务请求者需要知道的仅仅是它需要什么样的服务，服务提供者仅仅实现各种服务并将标准接口发布出去供服务请求者使用，服务代理者所做的工作是帮助服务请求者查找到服务提供者的服务。

2.3 Web Services实现

服务请求者使用服务提供者服务的过程为：服务请求者查找到(可以通过服务代理者查找，但不是必需的)需要的服务，然后将所需的服务进行绑定。服务请求者调用服务提供者服务的过程是通过SOAP协议实现的(在下文会有详细的介绍)。

2.4 Web Services特点

- 互操作性：任何Web服务之间都可以进行交互。
- 真正的跨平台：可以使用任何语言来编写Web服务。
- 简单性：Web服务使用HTTP和XML进行通信。
- 行业支持：所有主要的供应商都支持SOAP和Web Services，如微软的.NET平台就基于Web服务的。

3 SOAP

SOAP是实现Web Services最重要的协议之一，服务请求者和服务提供者之间的信息交互就是通过SOAP实现的。

3.1 SOAP概念

SOAP即简单对象服务协议，是以XML形

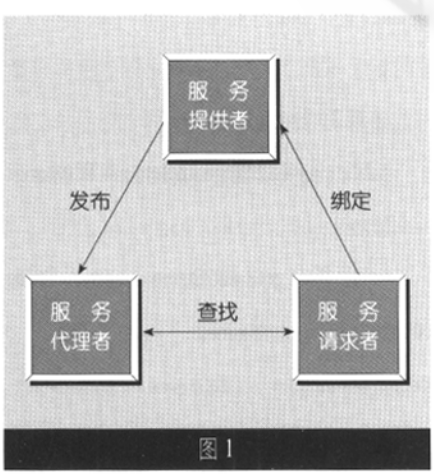
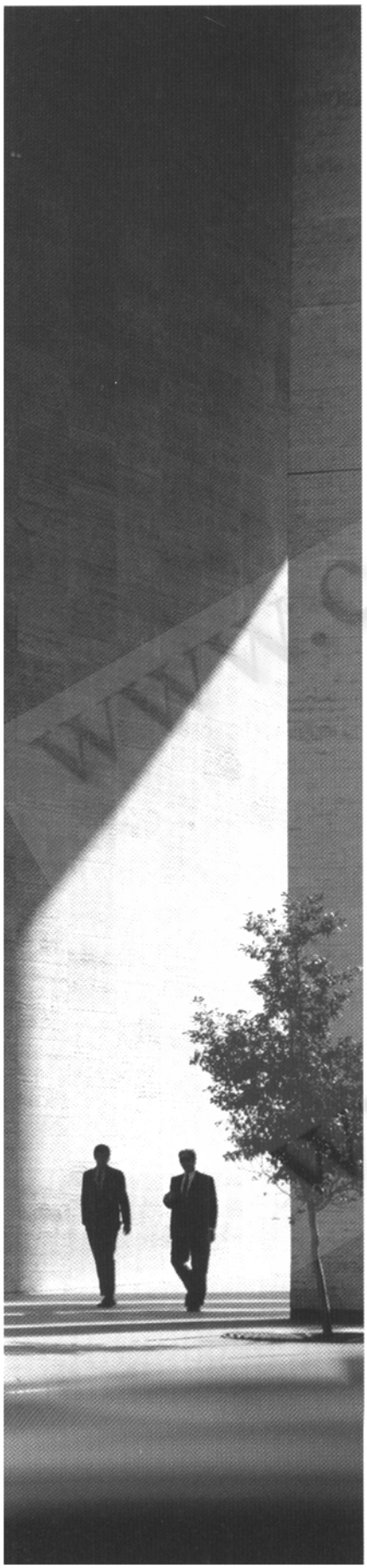


图1



式提供了一个简单、轻量的用于在分散或分布环境中交换结构化和类型信息的机制。

3.2 为什么要用 SOAP

当前的分布式计算现状要求系统能够在多平台、多系统、多种数据格式、多种连接方式上架构，要求这种机制能够满足低成本、高效率、易实现、松散耦合、跨平台，与语言无关，与特定接口无关等需求。SOAP正是为了满足上述要求而产生的。

比如Web Services通信双方可能一个使用window平台，另一个使用Linux平台，但因为他们都能够支持SOAP协议，所以可以实现软件集成。

3.3 SOAP 内容

SOAP是一种文档标准，用于描述服务请求者与服务提供者之间必要的信息。它包括如下四部分：

SOAP 封装(envelop)：用于定义一个描述消息中的内容是什么，是谁发送的，谁应当接受并处理它以及如何处理它们的框架；

SOAP编码规则（encoding rules）：用于表示应用程序需要使用的数据类型；

SOAP RPC 表示(RPC representation)：表示远程过程调用和应答的协定；

SOAP 绑定（binding）：使用底层协议交换信息。

3.4 SOAP 实现

SOAP=HTTP+XML+RPC。SOAP实现的具体过程为：服务请求者发送请求时，不管客户端是什么平台，首先把请求转换成XML格式，SOAP网关可自动执行这个转换。转化成XML格式后，SOAP的远程调用方法名及其他的一些协议标识信息被封装成HTTP请求，然后发送给服务提供者。服务提供者接受请求，在本地执行后再将结果信息转化为XML格式，封装成为HTTP包，返回服务请求者。

4 实例

本文提供的实例实现了一个用于进行汇率转换的 Web Services。

4.1 实现

本实例首先编写一个进行汇率转换的Java类，利用Apache SOAP工具配置成为一个Web Services进行发布，发布的信息包括该服务需要的参数，本例需要的参数为字符串类型的两个需要转换的国家名称（如USA，CHINA），然后通过Java程序进行调用，服务就可以返回汇率转换的结果值。

4.2 环境配置

实现 Web Services 需要的软件环境为：Apache SOAP 、Apache Jakarta Tomcat、Apache Xerces XML Parser，并需要将必要的jar文件称加入系统的CLASSPATH中。

4.3 web services 实现

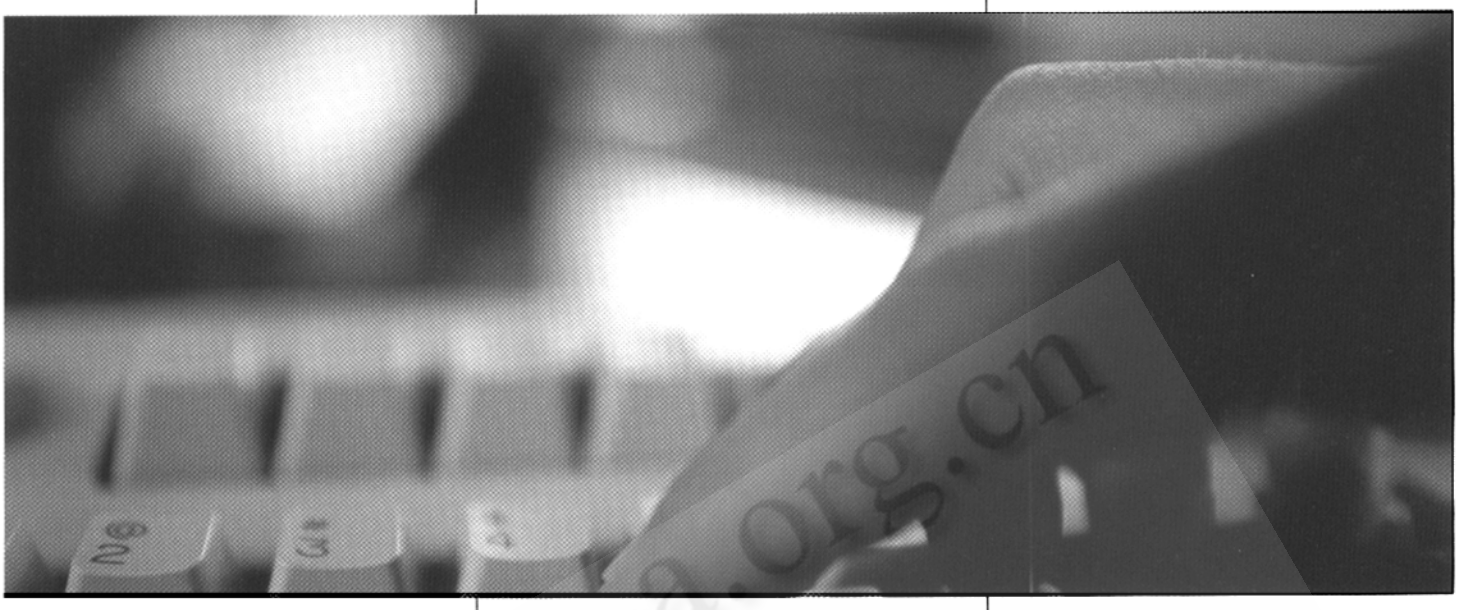
编写实现汇率转换的Java程序，把调用方法名设置为getRate，需要的参数为两个国家的名称（字符串类型）。

清单1：Iexchange.java

```
public interface IExchange
{
float getRate( String country1, String country2 );
}
```

清单2：Exchange.java

```
public class Exchange implements IExchange
{
public float getRate( String country1, String country2 )
    {
    System.out.println( "getRate( "+ coun-
```



try1 + ", " + country2 + ")");

return 8.02F; // always return the same value for now

}

}

将 Exchange.java 编译成为 Exchang.class 文件。

利用soap部署界面进行部署，主要部署的信息为：ID（Web 服务标识）、Scope（服务激活方式）、Methods（服务方法名）、Provider（Web 服务类型）、Provider Class/Static（Java 类名称），这样就在本机实现了一个名为 urn:exchange 的 Web Services。

4.4 客户端实现

客户端的实现主要是初始化对象Call，设置服务所在的位置、服务名称、服务需要的参数，然后进行利用 invoke 方法进行调用。

清单3：Client.java 部分代码。

URL url = new URL("http://localhost:8080/soap/servlet/rpcrouter");

String urn = "urn:exchange";

Call call = new Call(); // prepare the service invocation

call.setTargetObjectURI(urn); //set the property of Call

call.setMethodName("getRate"); //set the method of Call

call.setEncodingStyleURI(Constants.NS-URI-SOAP-ENC);

Vector params = new Vector();

params.addElement(new Parameter ("country1", String.class, "USA", null));

params.addElement(new Parameter ("country2", String.class, "japan", null));

call.setParams(params);

Response response = call.invoke(url, ""); // invoke the service

将 Client.java 编译为 Client.class 文件。

4.5 调用服务

启动 Tomcat 服务器，在命令行下用 Java Client 来执行客户端程序，就可以调用本地名为 urn:exchange 的 Web Services 了。

5 小结

本文实现了一个简单的Web Services，目的在于体现Web Services的基本思想。服务请求者在调用需要的汇率转换服务时，仅仅需要知道要提供需要进行汇率转换的两个国家名称（字符串类型）作为输入参数，然后接收一个符点型的执行结果，不需要了解服务实现的具体细节。

Web Services以其简单性、易实现性在软件集成、组件思想等方面具有其无法比拟的优点和实用价值，在业界得到了如IBM、MicroSoft等公司的大力支持。

注：本文实例是在 windows XP、Jdk1.3、Apache SOAP 2.2、Apache Jakarta Tomcat 3.1、Apache Xerces XML Parser2.0 上实现的。

参考文献

1 Barbara White（美），JavaBeans 开发使用手册，拓文工作室译，机械工业出版社，1998。

2 Joseph L. Weber(美)，Java2 编程详解，卜照斌译，电子工业出版社，1999。

3 World Wide Web Consortium .Extensible Markup Language(XML) 1.0 .http://www.w3.org/tr/。

4 XML,Web Services 专区.http://www-900.ibm.com/developerWorks/cn/xml/index.shtml。