

# 技术专题: Web Service

## Web Service简介

撰文 / Yasser Shohoud

你可能早就听说过Web service了,你也可能已经对Web service有一些概念了。一时间,好像所有的计算机期刊、书籍和网站都开始提及Web service。然而,当前大多数对Web service的介绍都没能清楚的说明Web service到底是什么。他们只是鼓吹Web service是多么好,简直就像是在做广告。在本文中,我就将讲清楚Web service到底是什么。

**关键词: Web service 分布式应用程序 浏览器**

### 分布式应用程序和浏览器

研究一下当前的应用程序开发,你会发现一个绝对的倾向:人们开始偏爱基于浏览器的瘦客户应用程序。这当然不是因为瘦客户能够提供更好的用户界面,而是因为它能够避免花在桌面应用程序发布上的高成本。发布桌面应用程序的成本很高,一半是因为应用程序安装和配置的问题,另一半是因为客户和服务器之间通信的问题。

传统的Windows胖客户应用程序使用DCOM来与服务器进行通信和调用远程对象。配置好DCOM使其在一个大型的网络中正常工作将是一个极富挑战性的工作,同时也是许多IT工程师的噩梦。事实上,许多IT工程师宁愿忍受浏览器所带来的功能限制,也不愿在局域网上去运行一个DCOM。在我看来,结果就是一个发布容易,但开发难度大而且用户界面极其受限的应用程序。极端的说,就是你花了更多的资金和时间,却开发出从用户看来功能更弱的应用程序。不信?问问你的会计师对新的基于浏览器的会计软件有什么想法:绝大多数商用程序用户希望使用更加友好的Windows用户界面。

关于客户端与服务器的通信问题,一个完美的解决方法是使用HTTP协议来通信。这是因为任何运行Web浏览器的机器都在使用HTTP协议。同时,当前许多防火墙也配置为只允许HTTP连接。

许多商用程序还面临另一个问题,那就是与其他程序的互操作性。如果所有的应用程序都是使用COM或.NET语言写的,并且都运行在Windows平台上,那就天下太平了。然而,事实上大多数商业数据仍然在大型主机上以非关系文件(VSAM)的形式存放,并由COBOL语言编写的大型机程序访问。而且,目前还有很多商用程序继续在使用C++、Java、Visual Basic和其他各种各样的语言编写。现在,除了最简单的程序之外,所有的应用程序都需要与运行在其他异构平台上的应用程序集成并进行数据交换。这样的任务通常都是由特殊的方法,(如文件传输和分析,消息队列,还有仅适用于某些情况的API,如IBM的“高级程序到程序交流(APPC)”等)来完成的。在以前,没有一个应用程序通信标准是独立于平台、组件模型和编程语言的只有通过Web Service、客户端和服务

才能够自由的用HTTP进行通信,不论两个程序的平台和编程语言是什么。

### 什么是Web Service

对这个问题,我们至少有两种答案。从表面上看,Web service就是一个应用程序,它向外界提供一个能够通过Web进行调用的API。这就是说,你能够用编程的方法通过Web来调用这个应用程序。我们把调用这个Web service的应用程序叫做客户。例如,你想创建一个Web service,它的作用是返回当前的天气情况。那么你可已建立一个ASP页面,它接受邮政编码作为查询字符串,然后返回一个由逗号隔开的字符串,包含了当前的气温和天气。要调用这个ASP页面,客户端需要发送下面的这个HTTP GET请求:

```
http://host.company.com/weather.asp?zipcode=20171
```

返回的数据就应该是这样:

```
21.0
```

这个ASP页面就应该可以算是Web service了。因为它基于HTTP GET请求,提供了一个可以通过Web调用的API。当然,Web service还有更多的东西。

下面是对Web service更精确的解释:Web services是建立可互操作的分布式应用程序的新平台。作为一个Windows程序员,你可能已经用COM或DCOM建立过基于组件的分布式应用程序。COM是一个非常好的组件技术,但是我们也很容易举出COM并不能满足要求的情况。

Web service平台是一套标准,它定义了应用程序如何在Web上实现互操作性。你可以用任何你喜欢的语言,在任何你喜欢的平台上写Web service。只要我们可以通过Web service标准对这些服务进行查询和访问。

### 新平台

Web service平台需要一套协议来实现分布式应用程序的创建。任何平台都有它的数据表示方法和类型系统。要实现互操作性,Web service平台必须提供一套标准的类型系统,用于沟通不同平台、编程语言和组件模型中的不同类型系统。在传统的分布式系统中,基于界面(interface)的平台提供了一些方法来

描述界面、方法和参数(译注:如COM和COBAR中的IDL语言)。同样的,Web service平台也必须提供一种标准来描述Web service,让客户可以得到足够的信息来调用这个Web service。最后,我们还必须有一种方法来对这个Web service进行远程调用。这种方法实际是一种远程过程调用协议(RPC)。为了达到互操作性,这种RPC协议还必须与平台和编程语言无关。下面几个小节就简要介绍了组成Web service平台的这三个技术。

## XML和XSD

可扩展的标记语言(XML)是Web service平台中表示数据的基本格式。除了易于建立和易于分析外,XML主要的优点在于它既是平台无关的,又是厂商无关的。无关性是比技术优越性更重要的;软件厂商是不会选择一个由竞争对手所发明的技术的。

XML解决了数据表示的问题,但它没有定义一套标准的数据类型,更没有说怎么去扩展这套数据类型。例如,整型数到底代表什么?16位,32位,还是64位?这些细节对实现互操作性都是很重要的。W3C制定的XML Schema(XSD)就是专门解决这个问题的一套标准。它定义了一套标准的数据类型,并给出了一种语言来扩展这套数据类型。Web service平台就是用XSD来作为其数据类型系统的。当你用某种语言(如VB.NET或C#)来构造一个Web service时,为了符合Web service标准,所有你使用的数据类型都必须被转换为XSD类型。你用的工具可能已经自动帮你完成了这个转换,但你很可能会根据你的需要修改一下转换过程。

## SOAP

Web service建好以后,你或者其他入就会去调用它。简单对象访问协议(SOAP)提供了标准的RPC方法来调用Web service。实际上,SOAP在这里有点用词不当;它意味着下面的Web service是以对象的方式表示的,但事实并不一定如此:你完全可以把你的Web service写成一系列的C函数,并仍然使用SOAP进行调用。SOAP规范定义了SOAP消息的格式,以及怎样通过HTTP协议来使用SOAP。SOAP也是基于XML和XSD的,XML是SOAP的数据编码方式。第三章我们会讨论SOAP,并结识SOAP消息的各种元素。

## WSDL

你会怎样向别人介绍你的Web service有什么功能,以及每个函数调用时的参数呢?你可能会自己写一套文档,你甚至可能会口头告诉需要使用你的Web service的人。这些非正式的方法至少都有一个严重的问题:当程序员坐到电脑前,想要使用你的Web service的时候,他们的工具(如Visual Studio)无法给他们提供任何帮助,因为这些工具根本就不了解你的Web service。解决方法是:用机器能阅读的方式提供一个正式的描述文档。Web service描述语言(WSDL)就是这样一个基于XML的语言,用于描述Web service及其函数、参数和返回值。因为是基于XML的,所以WSDL既是机器可阅读的,又是人可阅读的,这将是很大的好处。一些最新的开发工具既能根据你的Web service生成WSDL文档,又能导入WSDL文档,生成调用相应Web service的代码。

(接59页)

假设你有一个COM组件,暴露出一个GetTemperature方法:

```
Public Function GetTemperature(ByVal zipcode As String,
    ByVal celsius As Boolean) As Single
```

要把这个组件变成一个Web service,你可以使用WSDL向导。

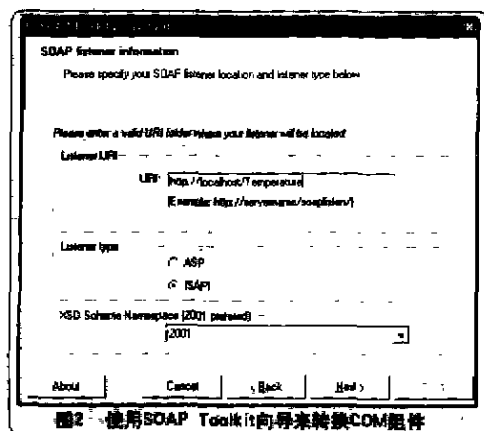


图2 使用SOAP Toolkit向导来转换COM组件

给出你要转换的组件后,向导会要你选择你想暴露出的方法,指出生成的Web service所在的URL(如http://localhost/Temperature/),以及你希望用ASP还是ISAPI做你的请求处理器(如图2)。然后向导还会问你生成的WSDL和ASP文件应该放在那个目录下。

现在该调用这个Web service了。方法是在VB或其他任何可以使用COM的语言里调用SoapClient组件。下面这段代码演示了怎样调用Web service中的GetTemperature方法:

```
Dim soap As MSSOAPLib.SoapClient
Set soap = New MSSOAPLib.SoapClient
soap.msoapinit "http://localhost/Temperature/Temperature.wsdl"
MsgBox ("气温是: " & soap.GetTemperature("20171", False))
```

首先调用msoapinit,把WSDL文档的URL传给SoapClient。WSDL文档的URL就是你在WSDL向导中给出的URL加上(Service名字.wsdl)。一旦初始化完成,SoapClient就得到了Web service的所有方法,你就可以直接调用这些方法了。

论文降重、修改、代写请扫码



免费论文查重，传递门 >> <http://free.paperyy.com>

阅读此文的还阅读了：

- [1. 基于RFID的可跟踪追溯平台研究与设计](#)
- [2. 兰州石化公司乙烯原料运输车辆管理系统的GPS接口实现](#)
- [3. 基于Asp.net的web service调用](#)
- [4. 异构系统间电力调度数据交换的设计与实现](#)
- [5. 基于RFID的可跟踪追溯平台研究与设计](#)
- [6. 基于BizTalk的异构系统集成实现](#)
- [7. 基于.Net的综合电子商务网站的设计与实现](#)
- [8. 透过奥运看数据——电视包装中实时数据的开发与应用](#)
- [9. 如何用Axis2开发Web Service服务](#)
- [10. 结合DEA和AHP的合作伙伴选择系统](#)