Microsoft .Net的Web Services及其实现

彭江平 （湖南大学（北校区）信息管理系 410079）

摘要：首先介绍Web Service的编程模型，然后简要概述了Web Service的编程模型中的两个关键要素XML与SOAP，随后给出了在Microsoft .Net环境下，基于XML与SOAP的Web Service的实现方法与步骤，最后，对Web Service的应用前景进行了展望。

关键词：Web Service　Microsoft .Net　XML　SOAP

1 引言

在大多数的情况下，Internet用户就像老式主机的分时终端上的用户一样，他们从一个受保护的资源请求信息，然后等待回应，从正在浏览的Internet站点上接收的信息是预先设计好的、基于HTML的信息。随着Internet的飞速发展及通信基础设施的改善，Internet用户已不再满足于Web站点通过浏览器提供漂亮的用户界面与发布信息，而是期望通过同远程Web站点进行交互式操作，以某种方法访问远程站点提供的计算能力并利用它的服务。这时，Web成为一个能够将组织、应用、服务以及设备迅速联结在一起的、可设计的Web站点；Web不再处于被动的地位，它们变成了可重用的智能化Web Services，即Web服务。

下面首先介绍Web Service的编程模型及该模型中的两个关键要素XML与SOAP，然后给出在Microsoft .Net环境下，基于XML与SOAP的Web Service的实现方法与步骤，最后，对Web Service的应用前景进行展望。

2 Web Service的技术要点

2.1 Web Service的编程模型

创建与使用Web Service的编程模型在许多方面与Com-组件的编程模型相似。COM组件模型的基本目标是使得软件开发者能通过重用各种组件创建新的应用程序，为了实现该目标，COM组件建立了一种定义对象接口的二进制标准，通过这样一种公共接口标准，使得组件的创建是语言中性（或语言无关）的，同时，也建立了COM组件跨进程或跨机器运行的机制，但是，COM组件的编程模型只能局限于特定的平台与对象模型，Web Services的基本目标是使得软件开发者能无缝地集成一类或多类应用或服务组成的集合，并且可运行于多种平台下，集合中的多种应用或服务可以使用不同的对象模型与编程语言实现。与COM编程模型不同的是，Web Services的编程模型是基于得到广泛支持的简单的、开放的标准；替代COM编程模型中的应用之间的二进制的通信方法，Web Services编程模型中使用基于XML的通信标准SOAP；在Web Services的编程模型中，XML不仅是用于表示原始数据的重要方法，而且是信息交互的统一的通信语言。

2.2 XML和SOAP

要实现上面的Web Service编程模型，需要解决的首要问题是数据的编码标准，基于该标准编码的数据或文档能在所有的操作系统平台、所有的应用系统中进行分析与处理。如果没有这样的标准，就正如也许有伟大的思想和信息可以分享，但却不能以一种别人可以理解的方式来表述思想和信息，这样，艰苦工作和想法就只能躺在那里睡大觉。XML，eXtensible Markup Language，可扩展标记语言，在今天的Web服务器中已经几乎无所不在了，数据库将通过XML中的记录集来读写，Web浏览器将接受XML并将其和伴随它的样式表一起显示，许多开发工具，如Visual Studio、Delphi甚至可产生XML代码；几乎所有的计算平台都能分析XML，因此，XML成为Web Service信息编码的事实标准。

实现Web Service编程模型，需要解决的另一个重要问题是数据的通信协议问题。长期以来我们使用超文本传输协议HTTP来提供Web页面以及往来的内容。但当将HTTP或一些其他Internet传输协议同XML结合起来，并指定XML文档自身的格式时，就得到了简单对象访问协议SOAP。在开始构想时，SOAP是被设计为从本地系统向远程系统传递远端方法调用的，基于SOAP的结构与同时代的其他远程调用结构——DCOM、CORBA和RMI等等所不同的是：SOAP协议可以穿越任何防火墙，并且SOAP数据包中包含着以XML编码的数据，易于分析和使用；SOAP具有很好的伸缩性，能同时为非常多的用户服务。SOAP模型最初的构想是使用请求-响应模型，同今天所用的Internet计算模型很相似。随后，SOAP发展到包含了消息模型，两者的不同之处是SOAP在对远端系统上的方法参数进行编码时，有获得结果的特殊目的，它并不请求Web站点提供一个感兴趣的数据表格。比如说，在同样的系统上能调用一个假想称为CalculatePayment()的远程调用，并收到一个个人付款数值，虽然现在也能用一个表单做到这些，但关键是在调用服务和提交表单之间存在着差别，服务调用是功能更强的概念。

3 Microsoft .Net 环境下基于 XML 和 SOAP 的 Web Services 实现

可重用的智能化 Web Services，即 Web 服务，是 Microsoft .Net 将要提供的精华。Microsoft .Net 将远程服务器所提供的计算能力和允许用户交互操作所必需的通信结合在一起。通用语言执行环境(Common Language Runtime，简称 CLR)本身就支持创建及使用 Web Services，其编程模型可升级、可扩展，支持开放的 Internet 标准，比如 HTTP、XML、SOAP、SD，因此，它可以被任何客户端或者具备 Internet 功能的设备访问和使用。为了更快更好地理解 NET 的 ASP Web Service 应用，下面我们举一个例子来说明。

3.1 建立一个基于 ASP.Net 的 Web Service (Calculator.asmx)

要建立一个基于 ASP.Net 的 WEB Service 的项目，首先是创建一个文本文件(.asmx)，在该文件的类中实现相应的服务功能，也就是 Service。在本实例中，写了四个简单的函数分别实现对两个整数的加、减、乘、除四则运算(在这里，选择的语言是 VB，依据 Microsoft .Net 的体系结构可知，除了 V B 外，还可以选择其他的语言，如 VC、C#、Jscript 等)。和普通的语法没有差别，唯一是在函数前面加了一个 WebMethod 的标记。下面是 Calculator.asmx 的完整代码，既可以选择 Visual Studio.NET，也可选择其他简单的文本编辑器完成代码编辑。

```
<%@ WebService Language =" Vb" Class="Calculator" %>
'这个实例演示了简单的 WEB 服务
'计算器有三个 WEB 方法作基本计算'
标记 <WebMethord()>导出方法
option strict off
Imports System.Web.Services
Public Class Calculator: Inherits WebService
Public Function <WebMethod()> Add(Num1 as integer,Num2 as integer) As Integer return Num1 + Num2
End Function
Public Function <WebMethod()> Substract (Num1 as integer,Num2 as integer) As Integer
return Num1 - Num2
End Function
Public Function <WebMethod()> Multiper (Num1 as integer,Num2 as integer) As Integer
return Num1*Num2
End Function
Public Function <WebMethod()> Divide(Num1 as integer,Num2 as integer) As Integer
return Num1 / Num2 End Function
End class
```

3.2 在浏览器中测试 Web Service

将上述.asmx 文件保存在 Web 服务器的适当的目录下，并映射为适当的虚拟目录后，就可在浏览器中测试该 Web 服务。如 http://192.1.2.1/aspnet/Calculator.asmx，就可以看到一个很标准的页面显示出刚刚定义的 Web Service 中的所有方法；而且还提供了可以执行它们的接口。例如，可以利用 Web Service 中提供的加法接口计算 3 + 2，输入参数后，就可得到一个以 XML 形式表示的结果页面。

3.3 使用 Web Service 的 Web 客户的创建

在 Web Service 创建成功后，就可设计一个表单 Web 页面来使用该服务。下面是相应的代码，其中提供了四个输入文本框，第一个文本框用于输入 Web Service 所在的 URL，而第二个与第三个文本框用于两个参数的输入。而下面提供四个按钮，分别调用 Web Service 的四个不同的方法，按钮的 Click 事件中在对控件的属性进行必要的设置后，请求相应的 Web 服务。

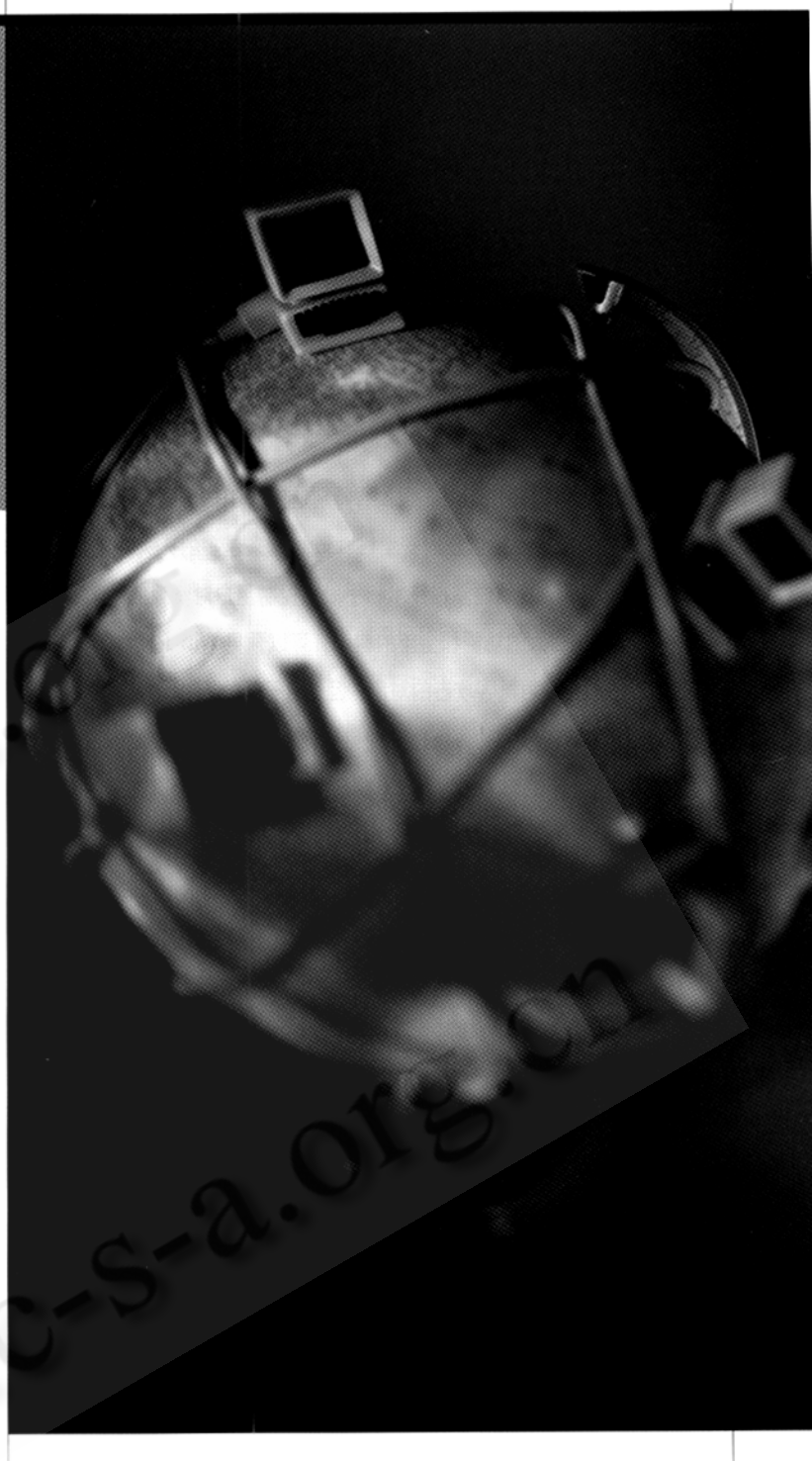
```
<html><h1> Web 服务演示 </h1><body bgcolor = pink>
<form id = frm method=POST >Web服务的URL，例如：http://server/service1.asmx <br><input type = text id = ServiceLocation style="WIDTH: 322px;"><br><br>
参数1: <input type=" text" size=" 5" name=' Num1' \"><br>
参数2 : <input type=" text" size=" 5" name=' Num2' \"><br><br>
<input type = button Value = 加 onclick = "Add()">
<input type = button Value = 减 onclick = "Substract()">
<input type = button Value = 乘 onclick = "Multiper()">
<input type = button Value = 除 onclick = "Divide()">
</form>
<script Language = "vbScript">
Sub Add
frm.action = frm.ServiceLocation.value & "/Add"
frm.submit
End sub
Sub Substract
frm.action = frm.ServiceLocation.value & "/Substract"
frm.submit
End sub
Sub Multiper
frm.action = frm.ServiceLocation.value & "/Multiper"
frm.submit
End sub
Sub Divide
frm.action = frm.ServiceLocation.value & "/Divide"
frm,submit
Snd sub
</script></body></html>
```

3.4 使用 Web Service 的其他客户端的创建

在上一小节中，讨论了在 W eb 中直接使用 Web Service 的方法，由前面的讨论可知，基于 XML 与 SOAP 的 Web Service，不仅能从 Web 游览器中访问，而且能使用其他多种形式的客户端。类似于 Corba 的 IDL，要从各种客户端中使用 Web Service，首先需要产生一个 SDL (Service Describing Language)，生成一个代理类，然后在客户端调用这个类中的方法，也就是相应的服务。

(1) SDL 文件的创建。有三种方式可以产生：手工自己写，使用.Net 工具，由.asmx 文件生成。自动的生成的可以用于观看和测试，但不能保存下来，

Development of the realtime application program in



如果要保存，需要使用一个Framework 的工具 disco.exe (Beta2，在Beta1 中使用 WebServiceUtil.exe)。通过 http://localhost/aspnet/Calculator.asmx?SDL，如果成功会有一个XML显示在IE中，然后在dos提示符下 disco http://localhost/aspnet/Calculator.asmx ?SDL 意思是保留结果，执行成功后可以看到这个 .wsdl 的文件，

(2) 代理类的生成，需要用 Framework SDK 中的另一个工具 wsdl.exe (Beta 2，在 Beta1 中使用 WebServiceUtil.exe) 用它生成一个.cs .vb 或 .js的文件，编译这个文件生成一个DLL，就是所谓的代理类了，然后就可以在 Web 网页，GUI 窗体，web 窗体或控制台程序中调用 Web Service 中提供的服务了，具体的语句：wsdl /out:: Calculator.vb http://localhost/aspnet/Calculator.asmx 成功后你可以看到这个VB文件生成，使用类似下面的命令行编译这个类：

```
vbc /out: Calculator.dll /t:library /r:System.XML.dll /r:System.Web.Services.dll Calculator.vb
```

(3) 代理类的使用，现在可以在各种应用中调用这个代理类组件了，应用很简单，大体都是：

```
dim obj as new Service1
dim retStr as string
dim retIntVal as string
retStr = obj.HelloWorld()
retIntVal = obj.Add (300 , 500)
```

4 结论

基于 Web Services 技术创建的 Internet 站点不再是带宽海洋中的孤岛，相反，可通过合作和互操作能力，使它融入 Internet 之中，Web Services 这种方式不仅灵活而且协议更加标准，底层封装和隐藏了 SOAP 和其他的网络协议，中间靠 XML 来传递数据和信息，开发者只需要专心于 Service 的功能设计上，假设新浪的新闻服务提供这样的 Web Service 接口，则在其他网站中只需要简单的几句就可以有新闻显示了；同样金融服务站点的股票信息，旅游航空公司的旅游和航班信息，等等，都可以以服务的形式供其他网站使用，也许会有一个 Web Service 的门户网站，它也会象 Yahoo 一样成为所有 Web Service 的众所周知的入口，

参 考 文 献

1 Framework SDK beta 2 :http://download.microsoft.com/download/VisualStudioNET/Trial/2.0/W982KMeXP/EN-US/setup.exe

2 Microsoft ASP.NET Premium Edition: http://download.microsoft.com/download/VisualStudioNET/Trial/2/W982KMeXP/EN-US/setup.exe

3 SOAP Toolkit 2.0 SP2:http://download.microsoft.com/download/xml/soap/2.0/W98NT42KMe/EN-US/SoapToolkit20.exe

4 SOAP Extensions with Visual Basic .NET:http://msdn.microsoft.com/msdn-files/026/002/305/VBSoapEx.exe