SOAP技术及其应用

李俭兵　何登平
（重庆信科设计有限公司　重庆400065）

The Technology and Application of SOAP

LI Jian-Bing　HE Deng-Ping
(Chongqing Information Technology Designing, CO.,LTD Chongqing400065)

Abstract The Simple Object Access Protocol (SOAP) facilitates interoperability among a wide range of programs and platforms, making existing applications accessible to a broader range of users. SOAP combines the proven Web technology of HTTP with the flexibility and extensibility of XML. This paper takes you on a comprehensive tour of Object RPC technology to help you understand the foundations of SOAP and the ways it overcomes many of the limitations of existing technologies, including DCOM and CORBA. This is followed by a detailed introduction of the SOAP development with Microsoft SOAP ToolKit.

Keywords SOAP,RPC,XML,HTTP,ROPE

1. 引言

在八十年代，两个主要的RPC协议是Sun RPC和DCE RPC。最流行的Sun RPC应用是大多数UNIX系统所使用的Network File System (NFS)。最流行的DCE RPC应用则是Windows NT，它采用DCE RPC协议来实现许多系统服务。这两个协议被证明适用于很大范围的应用。但是，在八十年代末期，面向对象技术的风靡使软件界沉迷于在面向对象语言和基于RPC的通信之间建立一个纽带。在九十年代产生的对象RPC (ORPC)协议正是试图把面向对象和网络协议联系起来。ORPC和RPC协议的主要不同是ORPC代码化了从通信终端到语言级对象的映射。目前两个主要的OPRC协议是DCOM和CORBA的Internet Inter-ORB Protocol (IIOP)或更一般的General Inter-ORB Protocol (GIOP)。对DCOM和CORBA/IIOP来说，最令人难以忍受的一点是它们不能在Internet上发挥作用。更糟的是，如果防火墙或代理服务器分隔开了客户和服务器的机器，任何IIOP或DCOM包要通过的可能性是很低的，主要是因为防火墙一般会根据协议的端口号对来访的请求数据进行控制，而大多数的分布式对象协议通常使用动态生成的端口号，因此实际上会造成同防火墙的冲突。为了克服这一问题，一种全新然而也是非常直观的解决方案——SOAP诞生了。

SOAP后面的指导理念是“它是第一个没有发明任何新技术的技术”。SOAP采用了已经广泛使用的两个协议：HTTP和XML，HTTP用于实现SOAP的RPC风格的传输，而XML是它的编码模式，SOAP简单地用XML来编码HTTP的传输内容。SOAP的核心是在HTTP消息体的请求和应答数据中引入XML结构，不过依然以HTTP作为数据的载体，通过POST命令发送数据，利用GET命令接收消息。SOAP最初由微软提出，后经IBM及Lotus参与修改，于2000年5月初提交W3C组织，并受到众多知名厂商的支持。

SOAP为Web服务器注入了新的生机和动力，但从根本上看，这其中XML功不可没。HTTP是一个相当有用的RPC协议，它提供了IIOP或DCOM在组帧、连接管理以及序列化对象应用等方面大部分功能的支持。(而且URLs与IORs和OBJREFs在功能上令人惊叹地接近)。HTTP所缺少的是用单一的标准格式来表达一个RPC调用中的参数。这正是XML的用武之地。象NDR和CDR，XML是一个与平台无关的中性的数据表达协议。XML允许数据被序列化成一个可以传递的形式，使得它容易地在任何平台上被解码。

XML有以下不同于NDR和CDR的特点：①有大量XML编码和解码软件存在于每个编程环境和平台上；②XML基于文本，相当容易用低技术水平的编程环境来处理；③XML有特别灵活的格式，它容易用一致的方式来被扩展。

一个SOAP方法可以简单地被看作遵循SOAP编码规则的HTTP请求和响应。SOAP请求是一个HTTP POST请求。SOAP请求的content-type必须用text/xml。而且它必须包含一个请求URI。服务器怎样解释这个请求URI是与实现相关的，但是许多实现中可能用它来映射到一个类或者一个对象。

2. SOAP的结构

SOAP包括三个部分：一个定义框架(描述消息内容和怎么处理它)的Envelope，一套对数据类型进行编码的机制，一个描述远地过程调用及其响应的约定。

一个典型的从证券公司取得最后成交价格的SOAP请求消息如下：

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"
<SOAP-ENV:Envelope
xmlns: SOAP-ENV = "http: // schemas.xmlsoap.org/soap/
envelope/"
SOAP-ENV: encodingStyle = "http: // schemas.xmlsoap.org/
soap/encoding/">
<SOAP-ENV:Body>
<m:GetLastTradePrice xmlns:m="Some-URI">
<symbol>DIS</symbol>
</m:GetLastTradePrice>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP-ENV = http: // schemas.xmlsoap.org/soap/

envelope/为名域 URI，它用 xmlns 来表示。SOAP-ENV:encodingStyle = "http: // schemas. xmlsoap. org/soap/encoding/")是编码。

其响应如下：

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
〈SOAP-ENV:Envelope
xmlns: SOAP-ENV = http: // schemas. xmlsoap. org/soap/envelope/
SOAP-ENV: encodingStyle = "http: // schemas. xmlsoap. org/soap/encoding/"/")
〈SOAP-ENV:Body〉
〈m:GetLastTradePriceResponse xmlns:m="Some-URI")
〈Price〉34. 5〈/Price〉
〈/m:GetLastTradePriceResponse〉
〈/SOAP-ENV:Body〉
〈/SOAP-ENV:Envelope〉
```

2.1 Envelope

如上，一个 Envelope 主要包括三个部分：Envelope、Header 和 Body。Envelope 是 XML Document(代表 SOAP 消息)中的根元素。象 DCOM 和 IIOP 一样，SOAP 支持协议头扩展，SOAP 用可选的〈Header〉元素来传载被协议扩展所使用的信息。注意：如果一个特定的 SOAP 头对正确处理消息是很关键的，这个头元素必须用 SOAP 属性 mustUnderstand = 'true'标记。

Body 一般是报告错误的 Fault 元素，常见的错误如表1所示。

表1 错误元素

值	名字	描述
100	版本不匹配	现有 SOAP 版本不支持此调用
200	Must Understand	一个嵌有 mustUnderstand = "true"属性的 SOAP 请求不能被接收方理解
300	无效请求	请求格式不正确
400	应用程序异常	接收方处理请求时发生异常

2.2 编码

支持简单类型(字符串，枚举)、组合类型(结构)和数组的编码，现有数组的编码加以说明。

一个 int [2]数组的编码如下：

```
〈element name="myFavoriteNumbers"
type="SOAP-ENC:Array"/〉
〈myFavoriteNumbers
SOAP-ENC:arrayType="xsd:int [2]"〉
〈number〉3〈/number〉
〈number〉4〈/number〉
〈/myFavoriteNumbers〉
```

注意 SOAP 数组的类型为 SOAP-ENC:Array。

2.3 远地过程调用及其响应的约定

SOAP 可以用两种方式利用 HTTP，一种是基本的 HTTP 方式(POST)，另一种是 HTTP Extension Framework 方式(M-POST)。一般来讲先用 HTTP 方式，如果返回错误，再利用 M-POST 方式。

3. SOAP 开发

SOAP ToolKit 是微软公司推出的一个 SOAP 开发工具，其核心是 ROPE(远地对象代理引擎)。ROPE 包括了 SOAP 客户端和服务端的基本框架，封装了客户端把 SOAP 调用用 XML 编码，打成 HTTP 请求包发送出去，然后从服务端接收到 HTTP 响应，从中解包分析出结果的整个过程。体系结构如图1所示。

现在用一个简单的取服务器上时间的程序解释一下开发流程。

·客户 VB 代码：利用 ROPE. Proxy 发出调用。

```
Dim oProxy As ROPE. Proxy
Set oProxy = New ROPE. Proxy // 建立一个 Proxy 对象
```

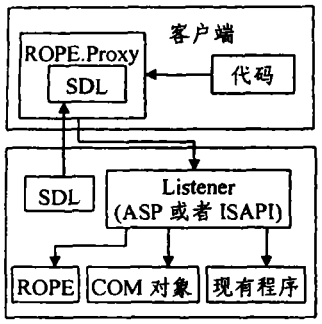

图1 Toolkit 体系结构

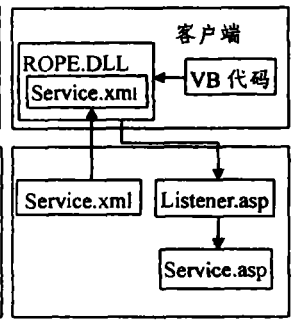

图2 程序体系结构

```
oProxy . LoadServicesDescription icSTRING , Services Description // Proxy 得到 SDL，进而得到服务端方法的 URI
oProxy. GetServerTime() // 可以利用 Proxy 调用服务端的方法 GetServerTime 取得时间
Set oProxy = Nothing // 清除 Proxy 对象
```

·ROPE. Proxy(以 ROPE. DLL 形式出现)：把客户发出的调用 XML 编码，打成 HTTP 请求包发送出去，然后从服务端接收到 HTTP 响应，从中解包分析出结果。

·SDL(服务定义语言)：ROPE. DLL 利用它知道服务接口、怎么样产生调用以及理解服务器的响应。

```
Service. xml
〈requestResponse name="GetServerTime"〉
〈request ref="ss:GetServerTimeReq" /〉
〈response ref="ss:GetServerTimeRes" /〉
〈/requestResponse〉
〈element name="GetServerTimeReq"〉〈/element〉
〈element name="GetServerTimeRes"〉〈type〉
〈element name="ServerTime" type="dt:string" /〉〈/type〉〈/element〉 'ServerTime 就是 GetServerTimeRes 的输出结果。
```

Listener：通过 HTTP 接收客户端传来的 SOAP 请求，解释请求进行处理，打包并传送给客户端，此处是通过 listener. asp 实现，用户一般不要随意修改这个文件。

现在需要真正实现 GetServerTime 这个方法了，它的定义在 service. asp 中。

```
CONST SOAP_SDLURI = http: // localhost/SOAPDemo/service. xml 'SDL 文件的 URI
〈!--#include file="listener. asp"--〉 ' 在 service. asp 中必须包括 listener. asp
〈%
Function GetServerTime(tz)
GetServerTime = Now
End Function
%〉
```

上面介绍了 SOAP 开发中主要涉及到的文件，实际上，可以利用 SOAP ToolKit Wizard 从 SDL 文件中自动生成整体框架，然后在 service. asp 中进行修改扩充，大大方便了开发。另外如果用 ISAPI 实现 Listener，需要一个 SOD 文件，它里面包括了 SDL 文件的位置和 COM 对象的 ProgID。

总结 SOAP 是一个被类型化的序列化格式，它恰巧

(下转第128页)

```
〈service documentation = "AgentGroupManagementAC" name = "AgentGroupManagementACInstance"〉
    〈port binding = "def: AgentGroupManagementInterfaceSOAPHTTPBinding" name = "AgentGroupManagementInterfaceSOAPHTTPPort"〉
        〈soap: address location = "http: // localhost: 9080/soap/servlet/rpcrouter/"/〉
    〈/port〉
    〈port binding = "def: AgentGroupManagementInterfaceSOAPEJBBinding" name = "AgentGroupManagementInterfaceSOAPEJBPort"〉
        〈soap:address
          location = "iiop: // localhost: 900? EJBHomeName = SOAPEJBRouter&amp; ContextFactory = com. ibm. webspher. enaming. WsnInitialContextFactory"/〉
    〈/port〉
〈/service〉
```

AC 客户端可以以两种方式访问 AC，分别是：使用 AC 服务和使用 SOAP 调用。以前者为例：使用 Java 客户机代理，该客户机代理为 WSDL 文档中描述的服务提供自然的 Java 接口，并将精密地镜像原始 bean 的接口。代理封装 SOAP 客户机编程 API 以及有关编写、发送、接收和分析 SOAP XML 文档的详细信息。

```
声明：AgentGroupManagementACServiceProxy AgentGroupManagementACSer-viceid = New AgentGroupManagementACService Proxy;
调用：AgentBaseInfo theAgentBaseInfo = AgentGroupManagementACServiceid. updateAgentProfile (anAgentBaseInfo,aRoleID);
```

以上调用范例使用Java客户机代理为：AgentGroupManagementACServiceProxy，适用于 Java Servlet 和 Applet 这两种编程模式。

4. Web Service 构件构架与传统构件构架的比较

与传统的构件构架（这里主要是指 COM/DCOM、JAVABEAN/EJB、CORBA 等组件构架）相比，Web Service 构件构架具有如下的优势：

1）Web Service 构件构架是一个更具分布式特性的构架，它不仅具有传统构件构架所具有的跨语言、跨平台等能力，而且具有跨传统构件标准的能力，能够有效地整合各种传统的构件模型，实现不同类型构件的相互调用（譬如，可以将 EJB 构件和 COM 构件包装成 Web Service 构件，从而实现两种异种构件的相互调用）。

2）Web Service 将构件看成是一种 Internet 上的服务单元，并把 Internet 当成是一个构件库，更利于构件的广泛重用；而传统的构件构架将构件看成仅是本地的一个可调用到的功能单元，需要专门的构件库。

3）Web Service 构件构架是对传统构件的改进，可以直接在 Internet 上作为服务单元发布，减少了传统软构件需要通过存储介质发送、安装、注册等环节，因而更加廉价、更加方便。

4）Web Service 构件构架更利于实现跨企业、跨区域的程序和数据交互，Web Service 构件通过将服务接口"裸露"的方式，使得外部的应用程序可以轻易地跨越企业、区域间的防火墙等安全屏障，实现商务逻辑的便利互访，而传统的构件架构要实现防火墙跨越，开发难度会非常大，且程序很难维护。

当然，我们可以看出，Web Service 在通过 Web 进行互操作或远程调用的时候是目前最有效的方式，而对于某些情况，譬如单机应用和局域网内的同构应用，Web Service 就没有什么优势了。

结束语 本文介绍了在 WebSphere 上的一个 Web Service 构件实现方案，该方案已在某地银行系统中得到了实施。此外，还需要说明的是：目前，Web Service 虽然浪潮迭起，各大软件开发商也争相提出自己的解决方案（微软提出的方案是.Net，并把 Web Service 称之为是软件开发技术的一次革命，Sun 拿出的方案叫作 Sun One），但是从总体上来讲，目前 Web Service 商业级系统所需的大多数技术和标准才刚开始出现，还没有统一的、最后的定论。不过，Web Service 作为 Internet 时代的一种计算模式，其美好的前景是不容质疑的。

参 考 文 献

1 柴晓路. 架构 Web services 系列. IBM developerWorks 中国网站，2001. 8
2 Li Jin. Web Services Overview. IBM developerWorks 中国网站，2002. 1
3 IBM 公司. IBM WebSphere Business Components Architecture Overview. IBM 公司白皮书，2000. 12
4 包路跃. Web Services 概述. 天极网（www. yesky. com）Visual Studio. net 专栏，2002. 4
5 蒋松，白利强. Web Service——下一代的 WWW. 计算机世界报，2001. 10
6 微软公司. Microsoft . NET 让新一代因特网变成现实. 微软公司白皮书，2000. 6

（上接第112页）

用 HTTP 作为请求/响应消息传输协议。SOAP 被设计为与 XML Schema 规范密切配合，并支持在 Internet 的任何地方运行的 COM、CORBA、Perl、Tcl、和 Java、C、Python 或 PHP 等程序间的互操作性。当然 SOAP 为了体现简单性和扩展性，也牺牲了一些技术，如它不支持对象引用等。

SOAP 的一个主要目标是使存在的应用能被更广泛的用户所使用，为了实现这个目的，没有任何 SOAP API，SOAP 是假设你将使用尽可能多的存在的技术。几个主要的 CORBA 厂商已经在他们的 ORB 产品中支持 SOAP 协议，微软也在 COM 中支持 SOAP，DevelopMentor 已经开发了参考实现。同时. NET 体系结构的客户端也是通过 SOAP 协议访问 asp 提供的 web service，从而执行应用程序的。相信在不久的将来 SOAP 必将成为互联网中一颗璀璨的明星。

参 考 文 献

1 Extensible Markup Language (XML). Available at http: // www. w3. org/XML, April 1997
2 XML Developer Center. Available at http: // msdn. microsoft. com/xml, November, 2000
3 The Apache XML Project. Available at http: // xml. apache. org/soap, 1999
4 Simple Object Access Protocol (SOAP) 1. 1. Available at http: // www. w3. org/TR/SOAP, May 2000
5 Object Management Group. CORBA/SOAP Interworking Request For Proposal. February 5, 2001
6 Microsoft . Net Enterprise Servers. Available at http: // www. microsoft. com/net/, 2000