

# 基于 Web 服务的 .NET 和 J2EE 互操作模型

## Interoperability Model of .NET and J2EE in Web Service

曾锻成

ZENG Duan-cheng

(广西标准技术研究院组织机构代码管理中心, 广西南宁 530022)

(Organization Code Centre of Guangxi Institute of Standards and Technology, Nanning, Guangxi, 530022, China)

**摘要:** 探讨在 Web 服务中实现 .NET 和 J2EE 互操作的方法, 提出实现两者互操作的模型, 并以数字图书馆的实现为例阐述该模型具体的实施过程。

**关键词:** Web 服务 J2EE .NET 互操作

**中图分类号:** TP393.09 **文献标识码:** A **文章编号:** 1002-7378(2013)01-0014-03

**Abstract:** Methods of .NET and J2EE interoperability were explored in web services. A interoperability model was achieved, and its implementation process was explained by using the realization of digital library as an example.

**Key words:** Web Service, J2EE, .NET, interoperability

随着互连网技术的不断进步, 基于 Web 的应用得到了广泛的发展。Web 服务实现了网上异构信息资源的整合, 为实现不同平台和不同语言的应用系统之间的集成提供了一种新的方法。目前, 开发 Web 服务的框架主要有微软的 .NET 和 Sun 的 J2EE, .NET 基于 Windows 平台和技术, 充分发挥其与操作系统结合的优势, J2EE 可适用于任何操作系统, 具有与平台选择无关的优势。已经有很多企业根据自身的特性和需求建立了基于 .NET 和 J2EE 的 Web 服务, 然而由于两种框架各自都存在一定的局限性, 它们之间完全协同工作还有很大的困难<sup>[1]</sup>。因此实现不同框架下 Web 服务之间的应用集成就成了一个必须解决的问题。

Web 服务用发现机制来定位服务(实现松散耦合), 使用服务描述来定义如何使用服务(实现普遍的通信), 使用标准的传送格式进行通信(实现统一的数据格式)<sup>[2]</sup>。这就为实现互操作提供了可能。Web 服务采用开放的结构, 其调用框架中客户和服

务之间通过标准的 Web 协议(HTTP 或 SMTP)和 XML 进行通信, 这些协议完全独立于平台, 容易实现。Web 服务数据被转换成标准 XML/SOAP 格式数据包在双方之间进行传递, 解决了异构系统带来的数据不兼容的问题<sup>[3]</sup>。而且 Web 服务将共享理解最小化, 实现了异构系统的互操作, 通过将遗留应用程序包装成 NASSL (Network Accessible Service Specification Language) 和 WDS (Well-Defined Service) 文档, 并暴露为服务, Web 服务体系结构能很容易的与这些遗留应用程序进行互操作<sup>[4]</sup>。本文探讨在 Web 服务中实现 .NET 和 J2EE 互操作的方法, 并提出实现两者互操作的模型。

### 1 互操作模型

Web 服务中的互操作是指采用异构开发框架下的服务提请求者和提供者能够协同工作, 无障碍地进行通信和数据共享。无论服务提供方是采用 .NET 开发框架还是 J2EE 开发框架, 请求方都能与其绑定, 调用其服务。我们假定请求方是 J2EE, 提供方是 .NET, 其互操作模型如图 1 所示。

由图 1 可知, J2EE 请求方在通过查询发现代理后, 找到所需服务的描述, 就按照服务描述所定义的

收稿日期: 2012-11-16

修回日期: 2012-12-10

作者简介: 曾锻成 (1980-), 男, 工程师, 主要从事组织机构代码管理工作。

方式与 .NET 请求系统完成绑定。服务的 WSDL 文档回答了关于 Web 服务的下列问题:被交换的数据是什么?如何与该服务进行交互?以及该服务位于何处?有了这些信息之后,服务请求者就能很明确地按照这种方式与请求者绑定,如同调用本地接口一样调用 Web 服务。

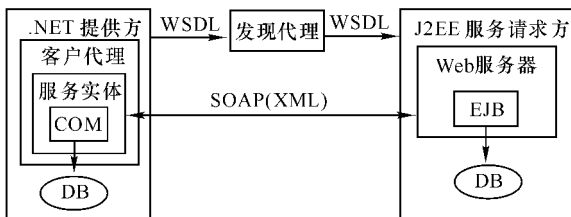


图 1 互操作模型

## 2 互操作模型实施过程

以数字图书馆的实现为案例,阐述 .NET 和 J2EE 在 Web 服务中的互操作的具体实施过程。设图书馆是服务提供者,校园师生是服务请求者,按照图 1 的互操作模型进行设计。提供方使用的开发工具是 Visual Studio .NET,数据库支持为 Oracle8i,服务请求方使用 Tomcat 5.0 作为 Web 服务器,Axis 作为 SOAP 服务器。

(1)创建 Web 服务。在 Visual Studio .NET 中创建 Web 服务,先新建一个 ASP.NET Web 服务,名字为 LibraryService。再在 LibraryService.asmx.cs 中添加两个方法,分别为 SearchByAuthor 和 getBookInformation。

```
[WebMethod]
public String SearchByAuthor(String author)
{ //按作者查询 }
[WebMethod]
public String getBookInfo(String TP_ID)
{ //按图书分类号查询 }。
```

(2)发布和部署这个服务。按 Ctrl+F5 编译这个程序,在下面的界面中列出刚才添加的 2 个方法,可以进入相应的页面进行测试,在浏览器地址栏里输入 `http://localhost/LibraryService/Service1.asmx?WSDL` 就可以得到这个 Web 服务的 WSDL。

(3)创建 J2EE 客户端程序。要实现与服务提供方的绑定,调用 Web 服务,首先需要服务的描述,即 WSDL 文档。在查找工程中,请求者可能向服务的注册处查找服务说明的位置,也可能会直接得到服务说明。查找操作可以出现在请求者生命周期

的两个不同阶段:设计时刻和运行时刻,前者获得服务的接口说明用于程序的开发,后者获得服务的绑定和定位说明用于服务的调用<sup>[3]</sup>。由于此案例是在本地开发的,通过前一步的开发我们已经得到了服务的说明,因此绑定和开发请求的客户端程序已经结合在一起。所以在 .NET 中只要得到了服务的 WSDL 文档,VS .NET 就能很快地识别它并建立客户代理,我们只需要创建服务的实例,然后调用即可。而 J2EE 作为服务请求调用端,不能很快地识别出服务的 WSDL,需要把它转换成 Java 文件。

把服务的 WSDL 转换成 Java 文件的方法如下:

查看服务的 WSDL 文档,可以得到服务的命名空间。由于本案例中命名空间为 `http://tempuri.org/`,那么在命令行中输入命令:

```
java org.apache.axis.wsdl2java Qname=http://tempuri.org/=temp http://localhost/Library-Service/Service1.asmx?WSDL
```

就能把服务的 WSDL 文档中描述的类和方法转换成本地 Java 文件。在生成的 `org.tempuri` 的包中,有 4 个文件,其中有一个文件是 `Service1SoapStub.java`。它包含了服务提供方法的具体内容和和其他的一些属性。另外 3 个文件分别是 `Service1.java`、`Service1Soap.java` 和 `Service1Locator.java`。在调用时要引用这些文件。

开始创建 J2EE 应用程序时,需新建一个 JSP 的程序 `RequestTest`。首先加入引用信息:

```
<%@page language="java" import="java.rmi.RemoteException"
import="javax.xml.rpc.ServiceException" import="temp.Service1"
import="temp.Service1Locator" import="temp.Service1Soap"
import="temp.Service1SoapStub"%>
```

再调用服务并显示:

```
try{
    Service1Locator locator = new Service1Locator();
    Service1Soap service = locator.getService1Soap();
    ResultSet resultset = service.SearchByAuthor(author);
    ResultSet books = service.getBookInformation(tp_id);
```

```

/***** 网页显示部分 *****/
*****/
}catch(Exception e){.....}。

```

(4)启动 Tomcat 服务器。在地址栏里输入 `http://localhost:8080/axis/RequestTest.jsp`, 即可以看到运行结果。

### 3 结束语

本文研究 Web 服务下 .NET 和 J2EE 的应用集成, 给出了两者互操作的模型, 为现有的企业实现不同开发框架下异构资源的整合提供了一定的参考。如今, Web 服务已基本上覆盖了传统分布式计算技术的应用领域, 但是更加复杂的应用领域要求将现有不同的应用程序无缝的组合起来, 形成功能更加

强大, 更完善的应用程序, 这还需进一步的研究。

参考文献:

- [1] 杨德华. 利用 J2EE 实现 Web 服务模型与应用[J]. 计算机工程与应用, 2004, 9: 123-125.
- [2] Champ ionM, Ferris C, et al. Web Services Architecture [R]. <http://www.w3.org/TR/2002/WD2ws2arch220021114>, 2002.
- [3] 郑小平. NET 精髓-Web 服务原理与开发[M]. 北京: 人民邮电出版社, 2002.
- [4] 刘宏.. NET 与 J2EE 在 Web Service 领域之比较[J]. 电脑学习, 2004, 4: 2-3.

(责任编辑: 尹 闯)

(上接第 13 页)

本文方法在计算功率谱时引进 FFT 算法, 减少了传统 DFT 变换的计算次数; 在计算信噪比时, 通过重复计算实验, 考察 Z-curve 映射序列中反映出的三类核苷酸的分布与信噪比的关系。结果显示, 与外显子判别密切相关的信噪比的值可以通过该 DNA 序列相应的数字映射到序列中, 由核苷酸或其组合的分布情况求得。该方法避开了傅里叶变换的繁冗工作, 而且此方法不要求 DNA 序列长度必须为 3 的倍数, 更利于推广和应用。

参考文献:

- [1] Yin Changchuan, Stephen S-T YAU. A Fourier characteristic of coding sequences: origins and a non-Fourier approximation[J]. Journal of Computational Biology, 2005, 12(9): 1153-1165.

- [2] 田元新, 陈超, 邹小勇, 等. 外显子周期三行为特征的研究[J]. 化学学报, 2005, 63: 1215-1219.
- [3] 邵建峰, 严晓华, 邵伟, 等. DNA 序列信号 3-周期特性[J]. 南京工业大学学报, 2012, 34(4): 133-137.
- [4] 马玉韬, 张成, 杨泽林, 等. DNA 映射方法对蛋白质编码区预测准确率的影响[J]. 安徽农业科学, 2012, 40(6): 3234-3238.
- [5] 刘小群, 周云波. 基于 Matlab 的 DFT 及 FFT 频谱分析[J]. 山西电子技术, 2012, 4: 48-49.
- [6] 马玉韬, 车进, 刘大铭. 基于傅里叶分析的蛋白质编码区预测中功率谱密度计算方法研究[J]. 宁夏大学学报, 2011, 32(2): 134-138.
- [7] 张志涌, 杨祖. MATLAB 教程[M]. 北京: 北京航空航天大学出版社, 2012.

(责任编辑: 尹 闯)