

# 数据库遗留系统向 Web Services 的半自动迁移

胡文江,姜郁峰,高永兵

(内蒙古科技大学 信息工程学院,内蒙古 包头 014010)

**摘要:**目前大中型企业仍然存在棘手的遗留数据,这些遗留数据库无法及时和外部健全的系统通信,新旧数据无法进行交互。为了解决这种困难,文中以 SOA 的思想出发,提出了一种基于集成 Web Services 的轻量级系统集成应用架构。运用 Web Services 强大的应用接口功能,通过 SOAP 进行数据交互,在人工的干涉下,实现数据库接口自动转变成 Web Services 的方法。不仅解决了遗留数据访问的不可操作性,而且对异构平台数据的互操作提供了服务方法和访问接口,实验验证在一定程度上节约企业原有资金成本。

**关键词:**遗留系统;数据库 Legacy System;SOA;Web Services

中图分类号:TP393;TP302.1

文献标识码:A

文章编号:1673-629X(2014)01-0109-04

doi:10.3969/j.issn.1673-629X.2014.01.028

## Semi Automatic Migration of Database Legacy System to Web Services

HU Wen-jiang, JIANG Yu-feng, GAO Yong-bing

(School of Information and Engineering, Inner Mongolia University of Science and Technology,  
Baotou 014010, China)

**Abstract:** Recently there are also thorny legacy data existing in large and medium-sized enterprise. These legacy databases cannot timely communicate with the sound external system, and new data and old data cannot interact with each other. In order to solve this difficulty, from beginning of SOA theory give a lightweight system integrated application architecture based on integrated Web Services. Use the powerful application interface function of Web Services and SOAP to carry out data interaction, realizing database interface automatically transformed into Web Services in the artificial interference. Not only can solve the operational legacy data access, but also provide services access interface to interoperate for the data on heterogeneous platform. To some extent, it saves the original cost of capital for company in the experimental validation.

**Key words:** legacy system; database Legacy System; SOA; Web Services

## 0 引言

随着计算机技术和互联网行业的飞速发展,各企业不仅仅局限于自身内部的系统来进行商业运作。企业在长期的运作过程中,企业信息数据的维护越来越成为一个困难的问题<sup>[1]</sup>。长期运维状态中,随着公司规模增长各个部门出现大量的子系统<sup>[2]</sup>。这些小的系统可能是不同人在不同的环境下用不同的语言编译的,长时间的数据交互过程中出现了大量的遗留数据。一部分存在着遗留系统,但还是保证不了数据很好的交互;一部分只有遗留的数据库,更不能谈到数据之间的处理。传统遗留数据处理方法不能很好地解决遗留

问题。

基于以上的原因,文中提出了一种半自动 Web Services 接口的数据库 Legacy System,在用户查询某个字段数据时自动生成 Web Services 的接口方法,在一定层面上实现了企业内部遗留数据的处理。

## 1 传统遗留数据的处理

BEN NETT<sup>[3]</sup>早年提出了 Legacy System 的概念:遗留系统是大家不知道如何处理但是又是企业中重要数据来源的系统,在其基础上很难做到改进并与现有的新系统进行互操作。在早期的探寻中,这些问题对

于企业来说相当困难和棘手。其中,修改遗留系统花费大量人力,结果却不理想,没有得到好的效果;另外,一些遗留数据库本身系统老化,代码混乱,只剩下数据孤岛,无法对外提供访问。

遗留系统的处理方式如下。

(1) 建立新的系统,新的数据,摒弃原有的遗留系统。这样做能够从根本上改观遗留数据的困扰,但此方式可行性低。一方面降低了企业原有的资源利用率,另一方面大量的数据消耗了大量的人力物力。

(2) 利用软件再工程维护遗留系统。再工程是用更先进的语言来改进和维护已有的遗留系统,在本质上不改变遗留系统的整体架构,能够保持系统的体系结构不变。它是在替换系统和维护系统之间运作,在一定时间和程度上解决了系统的可持续性。陈养平等提出利用中间件技术通过构件化实现对遗留系统的改造<sup>[4]</sup>;Ettore Merlo<sup>[5]</sup>等人提出让遗留系统接口进化成新的接口技术,延长了遗留系统的使用寿命。从技术角度来说,软件再工程是系统进化问题的一个次级解决方案。

(3) 包装遗留系统。其中,最为典型的有 RMI (Remote Method Invocation, 远程方法调用)、CORBA (Common Object Request Broker Architecture, 公共对象请求代理体系结构) 和 DCOM (Distributed Component Object Model, 分布式组件对象模式) 等等<sup>[6]</sup>。这几种技术在一定程度上对解决遗留系统问题有很大改善,但相继存在局限性:RMI 要求通信的两个端口配有 Java 环境,对于不同语言接口访问没有明确解决;CORBA 解决无语言通信方式,但易与防火墙发生对象请求冲突;DCOM 局限在只对 Windows 系统有较好的交互。

其次是数据库遗留的处理。在很多大型企业内部出现遗留数据,现有的系统老化已经没有系统优化可言,无法提供正常的数据库交互,也无法成为系统改进的研究素材,数据库遗留成为了数据孤岛,其问题没有得到大的改善。

通过半自动的 Web Services 方法来建立一个数据库 Legacy System 的解决方案进而解决了遗留数据的访问问题。

## 2 新一代网络服务 Web Services

Web Services 是基于 SOA 架构的强大的网络服务理念,它不是一种软件、一种产品,而是一种方法,维护管理企业数据交互的思想。当今企业都很注重企业信息数据服务和资源共享。像 IBM、Microsoft、Oracle、SUN 等公司在早期都宣布自己的产品里都支持 Web Services,目前已有很多公司拥有自己的基于 Web 服务

的成品在企业内部、企业之间运作<sup>[7]</sup>。Web Services<sup>[8-10]</sup>有自己的 UDDI 注册库,服务请求者发布 Web Services 时可以到注册中心注册;通过 WSDL 语言描述 Web Services 方法;通过 SOAP 进行数据交互。当请求服务者再次发出同样的服务请求后,请求者不必再发布服务描述,提取之前在 UDDI 注册的服务描述 WSDL 进行数据交互<sup>[11]</sup>。Web Services 具有良好的封装性、松耦合、标准协议规范、高度集成等特性<sup>[12]</sup>。图 1 显示了 Web Services 各部分的工作原理。

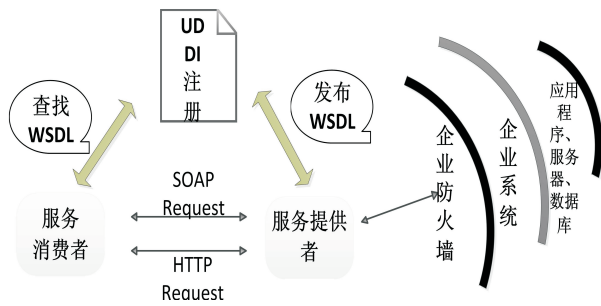


图 1 Web Services 原理

## 3 数据库 Legacy System

### 3.1 数据库 Legacy System 整体框架思想

根据遗留数据的特点,通过 Web Services 封装数据交互,构建稳定的数据库 Legacy System 体系。首先构造整体和集成 Web Services 方法,主要是移植数据库数据的迁移方法,然后构造访问遗留数据某列时半自动生成 Web 服务方法,建立测试客户端检测系统性能,最终解决遗留数据孤立无法访问的问题。图 2 为数据库 Legacy System 框架图。

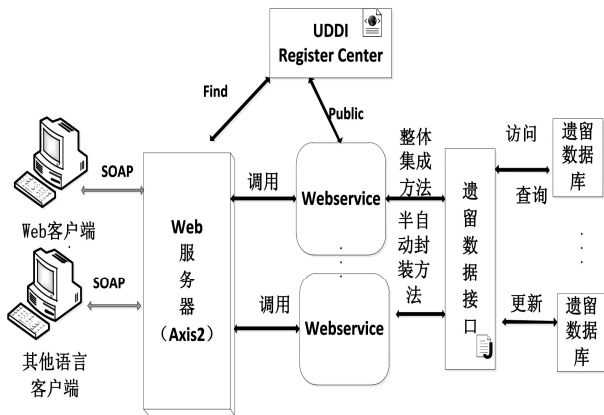


图 2 系统结构

#### 1) 整体流程。

- 当客户端提出 Web 服务请求后,发送 SOAP 请求到 Web 服务器;
- Web 服务器查找 UDDI Register Center, 找到符合的 Web 服务;
- UDDI Register Center 返回符合的 Web Services 的 WSDL 文档描述;

- Web 服务器得到相应的 Web 服务描述,产生 Web Services 代理,绑定到相应的 Web 服务上;
- Web 服务器调用对应的 Web 服务封装方法,提取数据;
- Web 服务器把得到的数据返回给客户端所在的用户。

2) 遗留数据接口模块。

根据遗留数据库基本样式,采用 Java 代码编写传统方法,实现对数据库的访问方法;包括对数据库更新、查询方法的实现。

3) 封装数据接口方法模块。

在已经实现的数据接口方法的基础上,针对其方法实现封装,先实现整体和集成的 Web Services 块,然后构造半自动 Web Services 接口的数据库 Legacy System,在用户查询某个字段数据时自动生成 Web Services 的接口方法。

4) Web 服务器和客户端模块。

通过 Axis2 服务器实现客户端和遗留数据之间的通信。Axis2 很好地利用 SOAP 进行交互,在客户端建立 RPC(远程过程调用,Remote Procedure Call),对不同语言建立的客户端也可以实现访问;在服务端调用封装好的数据接口 Web 服务方法。

3.2 系统体系结构示意图

根据整体架构分析出四层体系:遗留数据层、应用数据接口层、Web Services 封装层和访问层,如图 3 所示。

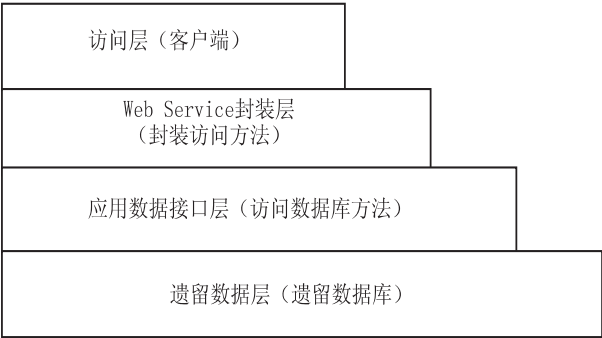


图 3 系统层次

4 实例分析

根据遗留数据的特点,采用 SQLSERVER2000 数据库,提取了 2010 年全国 IP 地址数据作为遗留数据库来进行实验。由于 IP 地址对应的地点不是每年都一成不变,有些商家或企业有变动时,企业的 IP 和地址就不同程度地做出大的改变。所采用的遗留数据字段,字段 ip1 和 ip2 为 IP 范围段,记录所在 IP 地址的范围;ip3 字段是省市的地点;ip4 为具体企业工作地点。例如:16777216.0 至 20185087.0,内蒙古通辽市,

联通;3740490513.0 至 3740490644.0,湖南省邵阳县邵东县,飞翔网吧等。

4.1 自动生成 Web Services 关键模块的实现

1) 采用整体封装遗留数据的方法。在整体查询接口模块中,在用户调用整体遗留数据时,自动发布一个整体的 Web Services 方法,来实现整体访问数据库数据,便于移植数据;

2) 形成自动的 Web Services 访问接口。在单独查询数据库数据时,根据查询的存储字段自动发布并生成 Web Services 接口模块,最终实现在查询的同时拥有自动的 Web Services 访问的方法,在以后的访问中可以调用之前已经有的自动的方法。

采用 WSDL 语法规则编写和发布相关遗留数据接口方法的 WSDL 文件,下面是部分文件,从文件中可以看到域名空间,SOAP 说明,方法名称为 Allwebservice,方法说明等资源描述。

```
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:ns="http://ws.apache.org/axis2" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <wsdl:message name="AllwebservedbRequest">
  <wsdl:message name="AllwebservedbResponse">
  <wsdl:portType name="AllwebservicePortType">
  <wsdl:binding name="AllwebserviceSoap11Binding" type="ns:AllwebservicePortType">
  <wsdl:binding name="AllwebserviceSoap12Binding" type="ns:AllwebservicePortType">
  <wsdl:binding name="AllwebserviceHttpBinding" type="ns:AllwebservicePortType">
  <wsdl:service name="Allwebservice">
</wsdl:definitions>
```

4.2 客户端交互模块的实现

客户端采用 RPC 通信,集成 axis2. AxisFault、axis2. addressing. EndpointReference、axis2. client. Options、axis2. rpc. client. RPCServiceClient 等接口,调用已发布的服务方法。

```
RPCServiceClient client=new RPCServiceClient();
Options options=client.getOptions();
String address="http://localhost:8080/axis2/services/Allwebservice";
EndpointReference epf=new EndpointReference(address);
options.setTo(epf);
QName qname=new QName("http://ws.apache.org/axis2","Allwebservice");
```

4.3 系统界面

根据所有模块编写,系统分为系统功能栏和功能显示界面两大部分。其中系统功能栏包括:LegacySystem 实例图、遗留数据 Web Services 整体查询、整体

UDDI 中心、整体 WSDL 发布、集成 Web Services 模块、应用查询模块,选择查询的检索词,自动发布 Web Services 方法,查询数据。

集成 WDDL 发布、应用查询等模块的实现。如图 4 为

应用 查询

按省市查询:

选择方法

WSDL发布

查询

返回

网吧或使用网点查询:

选择方法

WSDL发布

查询

返回

图 4 系统界面

查询结果为相关的使用 ASDL 的地点:  
[ 1. 035419202E9, 1. 035419226E9, 湖北省咸阳市, ASDL, 1. 035419136E9, 1. 0354192E9, 湖北省咸阳市, ASDL, 1. 035419228E9, 1. 035419556E9, 湖北省咸宁市, ASDL ]

4.4 系统优点

- 1)由于遗留数据是“数据孤岛”,不存在系统,也不存在对外接口,该方法实现了抽象数据访问接口和 Web Services 方法,解决了数据库对外交互;
- 2)运用整体 Web Services 数据封装方法和集成 Web Services 模块,便于数据库的整体移植;
- 3)自动生成的 Web Services 查询接口,有利于数据访问者按需查询优化;
- 4)采用轻量级框架,节省企业系统对改造遗留数据的消耗。

4.5 系统对比

表 1 为系统对比。

表 1 系统对比				
系统名	所用方法	所封装层	服务共享	平台访问
该课题系统	整体、集成自动	数据访问	支持	支持多系统平台
	Web Services	接口层		
Web 服务包装器	聚类分析/手动	应用层	不支持	Windows
Web 服务遗留系统	Dll、sqlWebserver/手动	数据层、事物层、应用程序层	支持	Windows

通过与其他系统的对比,在方法选择上,该课题系统更适合遗留系统的移植和进化;在服务 and 访问平台方面,该系统也是较其他系统更加先进。

5 结束语

经过验证 Web Services 方法在改造数据库遗留方面有很大程度的优点和强势。通过自动生成的 Web

Services 封装方法在很大程度上帮助了遗留数据的充分交互,一定程度上解决了企业数据库遗留的问题。利用发布的 Web Services 方法,基于各个平台的不同语言的企业系统都可以根据协议访问到发布的方法,从而最大程度地优化了企业 Legacy System,优化了遗留数据的访问。

参考文献:

[ 1 ] Lee B Y, Lee J K, Kim W. A knowledge-based maintenance of legacy systems; METASOFT[ J ]. Expert systems with applications, 1997, 12( 4 ): 483-496.

[ 2 ] 周家晶, 邹翔, 沈备军, 等. Web 遗留系统的服务包装器环境设计[ J ]. 计算机工程, 2011, 37( 19 ): 73-75.

[ 3 ] Bennett K H. Legacy system; Copy with success[ J ]. IEEE Software, 1995, 16( 1 ): 19-23.

[ 4 ] 陈养平, 黄士坦. 中间件技术在遗留系统改造中的应用[ J ]. 微电子学与计算机, 2007, 24( 6 ): 25-27.

[ 5 ] Ettore M, Pierre-Yves G, Jean-Francois G, et al. Reengineering user interfaces[ J ]. IEEE design and test of computers, 1999, 16( 2 ): 38-47.

[ 6 ] 丁兆青, 董传良. 基于 SOA 的分布式应用集成研究[ J ]. 计算机工程, 2007, 33( 10 ): 246-248.

[ 7 ] Red Oak software. Legacytransaction integration in a service-oriented architecture( SOA ) [ R/OL ]. 2003-11. [http://www.redoaksw.com/pdf/TransactionInt\\_inaSOA.pdf](http://www.redoaksw.com/pdf/TransactionInt_inaSOA.pdf).

[ 8 ] Mogha R, Preetham V V. Java Web Services programming [ M ]. [ s. l. ]: John Wiley & Sons, Inc. , 2002.

[ 9 ] Monson-Haefel R. J2EE# 8482Web Services [ M ]. [ s. l. ]: Addison Wesley Publishing Company, 2004.

[ 10 ] 顾宁, 刘家茂, 柴晓路. Web Services 原理与开发实践 [ M ]. 北京: 机械工业出版社, 2006.

[ 11 ] 马云峰, 王源. 基于 XML 和 Web Services 的异构数据集成研究与实现[ J ]. 计算机技术与发展, 2010, 20( 11 ): 42-46.

[ 12 ] 崔华, 应时, 袁文杰, 等. 语义 Web 服务组合综述[ J ]. 计算机科学, 2010, 37( 5 ): 21-25.

数据库遗留系统向Web Services的半自动迁移

作者：[胡文江](#)，[姜郁峰](#)，[高永兵](#)，[HU Wen-jiang](#)，[JIANG Yu-feng](#)，[GAO Yong-bing](#)

作者单位：[内蒙古科技大学 信息工程学院, 内蒙古 包头, 014010](#)

刊名：[计算机技术与发展](#)

ISTIC

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2014(1)

本文链接：[http://d.g.wanfangdata.com.cn/Periodical\\_wjfz201401028.aspx](http://d.g.wanfangdata.com.cn/Periodical_wjfz201401028.aspx)