

# 一种基于 Web Services 的信息集成方案

李艳霞, 巩九洲, 黎玉琴

(西北师范大学 数学与信息科学学院, 甘肃 兰州 730070)

**摘要:** 随着网络技术的发展, 信息的共享和交换显得越来越重要。针对目前网络中信息的异构性、多样性、分布性等特点, 提出了一种基于 Web Services 的异构信息集成方案。该方案中, 主要采用 XML 技术对各种源信息进行包装, 形成了一个虚拟数据库, 从而实现了异构数据源的共享。然后将信息注册在信息注册中心, 信息请求方获得 XML 格式的信息后, 使用 XML 解析器对其解析。该方案有效解决了“信息孤岛”问题。

**关键词:** 信息集成; Web Services; Wrapper; XML

**中图分类号:** TP393

**文献标识码:** A

**文章编号:** 1673-629X(2008)09-0105-03

## A Scheme Based on Web Services for Information Integration

LI Yan-xia, GONG Jiu-zhou, LI Yu-qin

(College of Mathematics & Information Science, Northwest Normal University, Lanzhou 730070, China)

**Abstract:** With development of Internet technology, the need for sharing and exchanging of information is becoming more and more important currently. In the paper, a heterogeneous information integration schema built on Web Service is proposed, which can accommodate to the heterogeneous, disperse and various information in present Internet. The scheme uses XML to encapsulate the information which will be changed in deferent systems, so the heterogeneous data source can be combined to form a virtual database then registries it to the information registry center. After the information requesters receiving the XML information, analyze it with XML parser. A scheme based on Web Services can effectively to solve the problem of “information island”.

**Key words:** information integration; Web services; wrapper; XML

## 0 引言

随着网络技术的发展, 信息铺天盖地涌来。Internet 技术的迅猛发展推动了计算机广泛和深入的应用, 大量的异构信息源系统被分散在各个网络节点中, 而它们之间往往是相互独立的。为了使这些孤立的数据能够更好地实现资源共享显得格外重要, 这迫切需要提供集成技术给用户提供一个统一、方便的系统环境, 为管理和决策提供及时可靠的服务, 以解决“信息孤岛”问题。集成系统的核心就是要解决异构数据源的分布性、自治性和异构性的问题。随着分布式计算技术的迅猛发展, 如 XML 技术等, 可以通过它们将原来的系统进行封装, 隐藏内部结构, 采用公共数据模型将局部共享数据进行包装。

针对上述问题, 提出了一种基于 Web Services 框

架下的信息集成解决方案。解决问题的思路是利用 XML 技术, 把它作为不同数据源的中间数据格式, 将来自不同数据源的数据进行包装, 以服务的形式发布。

## 1 Web Service 框架概述

Web Service<sup>[1,2]</sup>是一种开放的分布式应用程序的模型, 它能在所有支持 Internet 通讯的操作系统上实现。使用 Web Service 技术可以以独立于平台的方式, 通过标准的 Web 协议, 建立可以由应用程序通过网络访问的应用程序逻辑单元。它的运行机制见图 1。

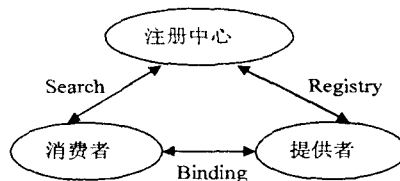


图 1 Web Services 框架

Web Service 的运行机制基于三种角色(服务提供者、服务注册中心和服务请求者)之间的交互。交互涉及注册、查找和绑定操作。这些角色和操作一起作用

收稿日期: 2007-12-13

基金项目: 甘肃省科技攻关计划项目(2GS047-A52-002-04)

作者简介: 李艳霞(1981-), 女, 甘肃天水人, 硕士研究生, 研究方向为网络技术、Web 计算; 导师: 冯百明, 教授, 研究方向为网络技术、Web 计算。

于 Web Service 构件: Web Service 软件模块及其描述。图 1 表示了这些操作、提供这些操作的组件及它们之间的交互。

这些操作具体为:

(1)注册:为了使服务可访问,需要发布服务描述以使服务请求者可以查找它。发布服务描述的位置可以根据应用程序的要求而变化。

(2)查找:在查找操作中,服务请求者直接检索服务描述或在服务注册中心中查询所要的服务类型。对于服务请求者,可能会在两个不同的生命周期阶段中牵涉到查找。

(3)绑定:最后,需要解决的问题是如何实现对服务的调用。在绑定操作中,服务请求方通过分析从注册服务器中得到的服务绑定信息,可以知道调用该服务所需的详细要求,包括服务的访问路径、服务调用的参数、返回结果、传输协议、安全要求等。服务请求方根据这些信息对自己的系统进行相应配置,实现对服务的远程调用。

## 2 基于 Web Services 的异构分布信息的集成方案

### 2.1 整体框架图

目前,在开发信息集成系统时采用的方法虽各不相同,但其基本的方法可分为两类:物化方法(Materialized,也称数据仓库法)和虚拟方法(Virtual,也称中间件法)。在文中,基于 Web Services 框架进行信息集成采用虚拟方法<sup>[3]</sup>。包装器包装数据源,把底层的数据对象转换为统一的数据模型,用服务进行封装并且在注册中心进行注册。在本系统中将异构信息分为三类:数据库信息(如 Oracle、SQL Server、Sybase、Access 等)、文本信息(如现有的 XML 文档)、其它信息(如传感器的动态数据、遗留代码等)。

Web Service 封装了各种信息的功能并隐藏了内

部的复杂度。XML 语言用来包装各种信息将其转换成统一的数据模型。

### 2.2 包装器的设置

信息源层处于最低层,是系统的数据提供者。在此包括各种类型、各种系统,在数据管理上采用“虚拟集中”<sup>[3]</sup>的方式进行。每一个异构信息由一个数据源和一个“外套(wrapper)”构成。集成系统是面向各种信息源的,数据类型往往多种多样。由于 XML<sup>[4,5]</sup>具有可扩展性和结构性等特点,因此,用 XML 模型作为集成系统的公共模型。

#### 2.2.1 信息包装

从数据源中读取数据,生成 XML 统一的文件格式。如下所示:

```
<? xml version="1.0" encoding="gb2312"? >
<root>
  <StudentItem>
    <Id>2006001</Id>
    <Name>Larry</Name>
    <Sex>女</Sex>
  </StudentItem>
</root>
```

StudentItem 的个数是可扩展的,可以是一个或多个且子元素的名称不固定,StudentItem 下面子元素的名称是随着数据库表中栏目名称的不同而变化的。

#### 2.2.2 包装器的实现过程

包装部分采用 Java 语言来实现,以满足系统的可移植性和良好的跨平台性能,开发平台采用 Eclipse, Eclipse 是一个开放源码的、可扩展的应用开发平台,可以用于构建 Web Service、J2EE 等各种类型的应用。本系统使用 JDBC 访问数据库。通过驱动程序管理器, JDBC API 可以利用不同的驱动程序链接不同的数据库系统。JDBC 的 DatabaseMetaData 接口提供了一系列方法可以检验 DBMS 对特定特性的支持,从而能对特定数据库的特性给予支持。本系统采用 JDOM 来

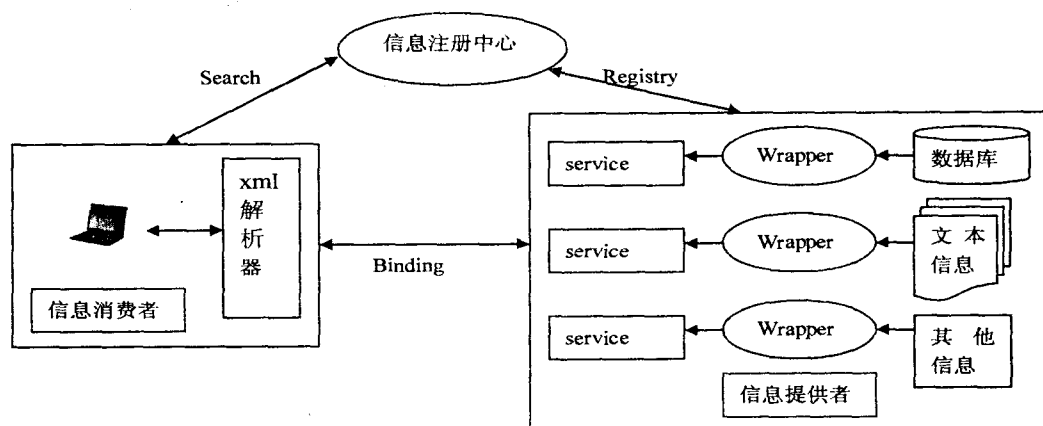


图 2 基于 Web Services 的信息集成框架

自动生成 XML 文件,因为 JDOM 弥补 DOM 和 SAX 在实际应用当中的不足之处。

下面是一个将数据库表 studentinfo 包装成 XML 中间信息的关键代码:

```
import java.io.IOException;
import org.jdom.Content;
import org.jdom.Document;
import org.jdom.Element;
import org.jdom.output.Format;
import org.jdom.output.XMLOutputter;

public class JdomCreateXml{
    public static void main(String[] args)
    {
        try{ String strSQL="select * from studentinfo";
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");//载入驱动
        Connection con = DriverManager.getConnection("jdbc:odbc:Mydb");
        //建立数据表连线,Mydb 是数据源名称
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(strSQL);//建立查询后的 ResultSet 集合
        ResultSetMetaData rsmd = rs.getMetaData();
        Element rootElement = new Element("root");
        Document mydoc = new Document(rootElement);
        while(rs.next()){
            Element element = new Element("element");
            rootElement.addContent(element);
            for (int j = 1;j <= rsmd.getColumnCount();j++)
            {try{
                String date = new String(rs.getString(j));
                Element meta = new Element(rsmd洗ColumnNane(j)).setText
                (date);
                element.addContent(meta);
            }catch (Exception e){}
            }
        }
        XMLOutputter xmlOut = new XMLOutputter("",true,"gb2312");
        xmlOut.output(mydoc,System.out);
        FileWriter writer = new FileWriter("myfile.xml");
        xmlOut.output(mydoc,writer);
        writer.close();
        stmt.close();
        con.close();
    }catch (Exception e) { System.out.println(e.getMessage
    ());}
}
```

如图 3、4 所示为一个包装前的数据库信息和一个包装后的 XML 信息(图中只截取了一部分)。

编号	学号	姓名	性别	籍贯
1	200601	王会歌	女	河南洛阳
2	200602	李婧	女	甘肃天水
3	200603	吴伟	男	山西大同
4	200604	杜平	男	四川绵阳
5	200605	张林	女	甘肃庆阳
6	200606	王刚	男	山东济南

图 3 包装前的 studentinfo 表

```
<? xml version="1.0" encoding="gb2312"? >
<root>
  <element>
    <编号>1</编号>
    <学号>200601</学号>
    <姓名>王会歌</姓名>
    <性别>女</性别>
    <籍贯>河南洛阳</籍贯>
  </element>
```

图 4 studentinfo 表包装后的 XML 信息

### 2.3 XML 解析器

对于信息消费者,当获得信息提供方返回的 SOAP 消息后,实际上返回的是一个 String 字符串,XML 解析器将得到的 XML 格式的信息文档通过 JDOM 解析器 + JSP 界面设计语言,呈现给用户一个友好的信息访问界面。使用 JDOM 解析器需将 jdom.jar 和 xerces.jar 文件复制到 Tomcat 网络应用程序的 WEB-INF\lib 目录中。解析过程中比较重要的几个方法如下:

(1)实例化一个合适的解析器对象:SAXBuilder sb = new SAXBuilder()。

(2)以包含 XML 数据的文件为参数构建一个文档对象:myDocument Document myDocument = sb.build(xmlpath);//xmlpath 是 XML 文档的路径。

(3)获得根元素及其子元素:Element rootElement = myDocument.getRootElement();List list = element.getChildren()。

## 3 集成方案的特点

通过系统的实现,用 Web 服务集成的应用程序与传统的中间件相比具有以下优点:

(1)开放性与先进性:Web 服务基于开放的标准(UDDI、SOAP、HTTP、XML),将这些技术有机地结合起来,减少了投资。

(2)可移植性:在实现的过程中使用 Java 开发,重用率高。

(3)松散耦合:Web 服务的集成非常灵活,它是建立在发布服务的应用程序和使用服务的应用程序之间

(下转第 130 页)

动攻击和被动攻击。下面将从以下几个方面进行分析:

(1)截取:由于认证消息是经过加密和签名的,恶意节点即使截取了消息,也无法解密消息,更不可能获得合法证书。

(2)修改:AHAN 协议用会议密钥加密数据控制包,而它是包含在每一个传送的包中,这样保证了数据的完整性。

(3)伪造:在 AHMAN 协议中,由于恶意节点无法得到节点的私钥和合法的数字证书,所以无法伪造路由消息。每个节点验证上一节点的数字证书及签名的有效性,目的节点通过源节点的数字证书验证路由请求消息是否真正来源于源节点;同样,源节点通过目的节点的证书判断路由回复消息是否真正来源于目的节点。通过这种端对端的认证过程,有效防止了消息的伪造。

(4)应答攻击:AHMAN 协议通过对消息的签名和时间戳来保证其不可否认性和有效性。

(5)窃听:采用会议密钥加密整个路由消息,防止了恶意节点对信号的窃听、监视等,保证了消息的机密性。

## 4 结束语

提出了一种认证私钥分存方法实现了 $(n, t)$ 门限加密分布式认证,重点讲述了簇节点改变和端到端通信的认证过程。本方案在认证过程中采用不对称密

钥加密证书,用会议密钥加密端到端消息,有效防止了恶意节点对消息的截取、修改、伪造、监视和窃听等攻击。文中的安全分级分布式认证不仅能够容忍一定数量的认证节点崩溃,而且还能够有效防范内部被攻破认证节点对分布式认证的干扰,所以具有很好的健壮性。今后的工作是用仿真的方式对协议进行进一步的验证。

## 参考文献:

- [1] Zhou Lidong, Haas Z J. Securing ad hoc networks[J]. IEEE Network, 1999, 13(6): 24-30.
- [2] 杨微微, 陈远山. 移动 Ad hoc 网络安全技术研究[J]. 计算机应用, 2006(s1): 157-159.
- [3] Bechler M, Hof H J, Pahlke D K, et al. A cluster-based security architecture for ad hoc networks[J]. IEEE Infocom, 2004, 4: 2393-2403.
- [4] 金丽丽, 李家滨. Ad Hoc 网络安全密钥管理服务[J]. 计算机工程, 2004(18): 113-115.
- [5] Steiner M, Tsudik G, Waidner M. Diffie-Hellman key distribution extended to group communication[C]// Proceedings of the 3rd ACM Conference on Computer and Communications Security. [s.l.]: ACM Press, 1996: 31-37.
- [6] Papadimitratos P, Haas Z J. Secure Routing for Mobile Ad Hoc Networks[C]// in Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002). San Antonio, TX: [s. n.], 2002: 27-31.

(上接第 107 页)

的松散耦合,从而使得应用易于修改并且对程序及流程的运行没有任何影响;并且可以方便地连接异构的平台和系统。

(4)动态集成:接口改变后,应用程序能够方便地重新获取服务描述文档,重新生成调用口,并与代码进行动态绑定。

(5)简单:Web 服务比起传统的中间件如 CORBA<sup>[6]</sup>、COM 没有那样复杂的 API,对它的开发、部署非常简单。

## 4 结束语

与传统的信息集成相比,Web Services 架构提供了很多优势,即开放性、方便、经济、高效。文中就如何应用 Web Services 技术构建信息集成过程提出了一种集成方案并进行实现,这种基于 Web Services 的方式能够满足各种信息集成要求,体现了松散耦合、位置透明、协议独立的特点,能够支持按需应变的业务需求,

具有广阔的应用前景。

## 参考文献:

- [1] Mogha R, Preetham V V. Java Web Services Programming [M]. 刘 凌译. 北京:清华大学出版社, 2003.
- [2] Pullen M J, Brunton R. Using Web services to integrate heterogeneous simulations in a grid environment[J]. Future Generation Computer Systems, 2004(9): 98-99.
- [3] 杨先娣, 彭智勇. 信息集成研究综述[J]. 计算机科学, 2006, 33(7): 55-56.
- [4] Qu Changtao, Nejdl W. Integrating XQuery-enabled SC-ORM XML Metadata Repositories into an RDF-based E-Learning P2P Network[J]. Educational Technology & Society, 2004, 7(2): 51-60.
- [5] 杨建物, 陈晓鸥. XML 相关标准综述[J]. 计算机科学, 2002, 29(2): 25-27.
- [6] 毛春丽, 贾 焰, 周 斌. 基于 Web Services 应用集成技术的研究及其实现[J]. 计算机科学, 2007, 34(2): 104-106.