XML和CORBA的集成在金融信息服务系统中的应用

The Application of the integration of CORBA and XML in A Financial Information Service System

摘　要：本文介绍了公共对象请求代理体系结构CORBA和XML,讨论了CORBA和XML的集成,并举例说明了CORBA和XML在金融信息服务系统中的应用。

关键词：XML　CORBA　电子商务

万定生　杨浩文　陈元斌（南京河海大学计算机及信息工程学院　210024）

网络的兴起及蓬勃发展极大地改变了人们的生活、学习和工作方式，它提供给人们的不仅是大量的信息和娱乐活动，而且也带来了无限商机。随着电子商务的深入发展，企业信息系统比以往任何时候更加依赖分布式计算架构。可扩展标记语言(XML)具有良好的自描述性、灵活性、可扩展性，非常适合Web上的数据交换和发布，而CORBA定义了分布式对象之间通信所需要的完整体系结构。在分布式领域中，XML作为一种中间的数据接口，已经显示出其不可替代的重要性。如何将企业已有的CORBA应用与面向消息的、松散耦合的Web技术结合起来，使企业更好地适应新经济的发展，是很多企业分布式应用面临的一个重要课题，XML与CORBA集成为解决该课题提供了一个有益的思路。

1 CORBA技术

通用对象代理体系结构CORBA(common object request broker architecture)是对象管理组织所定义的用来实现分布式软件之间互操作的解决方案, CORBA也是迈向面向对象标准化和互操作的重要一步。CORBA允许应用之间相互通信,而不管它们存在于哪里以及是谁设计的。CORBA定义了接口定义语言(IDL)以及在对象请求代理(ORB)中实现客户对象与服务器对象之间交互的应用编程接口(API)。CORBA规定了各个供应商之间的ORB的通信规则。CORBA标准主要分为3个部分：IDL，ORB以及ORB之间的互操作协议IIOP[2]。ORB是对象之间建立客户 服务器关系的中间件。使用ORB，客户可以透明地调用一个服务对象上的方法,这个服务对象可以在本地,也可以在通过网络连接的其他机器上。ORB截获这一调用同时负责查找实现服务的对象并向其传递参数、调用方法、返回最终结果。客户并不知道服务对象位于什么地方，它的编程语言和操作系统是什么，也不知道不属于对象接口的其他系统部分。这样，ORB在异构分布环境下为不同机器上的应用提供了互操作性,并无缝地集成了多种对象系统。

2 XML语言

XML是一种置标语言,其英文全称为extensible markup language它依赖于描述一定规则的标记和能够读懂这些标记的应用处理工具来发挥它的强大功能。XML的精髓在于“置标”(Markup)，置标的定义是：就数据本身的信息对数据进行编码的方法。在XML中,置标的语法是通过文件类型定义DTD(Document Type Definition)来描述的。也就是说,通过DTD来描述什么是有效的标记，从而进一步定义置标语言的结构，在用XML定义的置标语言中,DTD是与数据文件分离的部分。除了定义置标的语法外，XML还使用与之相连的样式单(Style Sheet)，由它来向应用程序提供如何处理显示的指示说明（比如浏览器）。样式单是一种专门描述结构文档表现方式的文档，既可以描述这些文档如何在屏幕上显示，也可以描述打印效果，甚至声音效果、样式单一般不包含在XML文档内部，而以独立的文档方式存在。W3C正式推荐的样式单标准有两种：一种是层叠样式单CSS；另一种是可扩展样式单语言XSL。处理XML文档需要XML解析器，解析器读入XML文档检查其合法性并进行处理。许多解析器既提供SAX接口又提供DOM接口。SAX是一个为基于事件的XML解析器定义的API，它允许程序和脚本动态地访问和更新XML的内容、结构和文档风格。DOM是一个为基于树型的XML解析器定义的API，它允许程序将XML数据构建成对象并且允许对象间相互结合、访问、操纵。

3 CORBA和XML的集成

诸多理由证明，集成CORBA和XML是非常有用的。在成功的分布式对象计算中，这两种技术之间有很多共同作用。系统结构体系中有很多种方法来集成CORBA和XML。

3.1 从信息的角度来看

系统信息，不论是否是分布式的，都可以方便地用XML来表示。XML是一种很自然的方式来表示组件，服务，服务器等信息。以XML表示的配置信息一般是轻便和可用的。在CORBA系统中使用XML来传递数据会使系统更加灵活。不用再狭隘地在IDL接口中规定操作给定的数据，而是规定接收和发送XML，这样做的好处是可以在不改变接口的情况下改变数据。

3.2 从层的角度来看

大多数CORBA体系结构是由多层构成的。XML可以用在各层的接口中，来提供更大的灵活性，充当结构中的粘合剂。

把XML本身用来存储数据是否有意义，这取决于系统的内容是否是以文档为中心的。对大多商业数据来说，这存在着问题。障碍在于大量的非XML数据和典型使用的一些数据存储的性质。

中间层可以发送XML到客户端，把XML数据送入提供表示控制的XSL处理器中。不同的XSL风格单可以以不同的方式来表现相同的XML。XML和XSL的结合给了表现形式巨大的灵活性。

应用程序可以产生XML，不仅给客户端使用，也可以用来和其他数据系统进行数据交换。这在B2B的处理中尤其有用。

3.3 从通信的角度来看

XML提供了一种在不同系统中传递信息的方法，它可以穿过很多传统上的技术壁垒。目前正在研究通过HTTP使用XML来取代在CORBA系统中采用IDL/IIOP来通信。这可以支持那些不支持CORBA的简单客户，或支持使用不兼容IIOP的协议的客户端。而且使用HTTP来传输XML可以消除很多防火墙的问题。请求/响应消息的XML格式可以被XML—CORBA桥解释：SERVLET、CGI、或者WEB服务器模型可以解析XML请求，把它翻译成CORBA请求，反之亦然。使用XML来连接WEB世界和以前的分布式对象技术是为了发展采用URL、HTTP、XML等核心WEB标准的分布式系统。

4 金融信息服务系统中实现CORBA和XML集成的实例

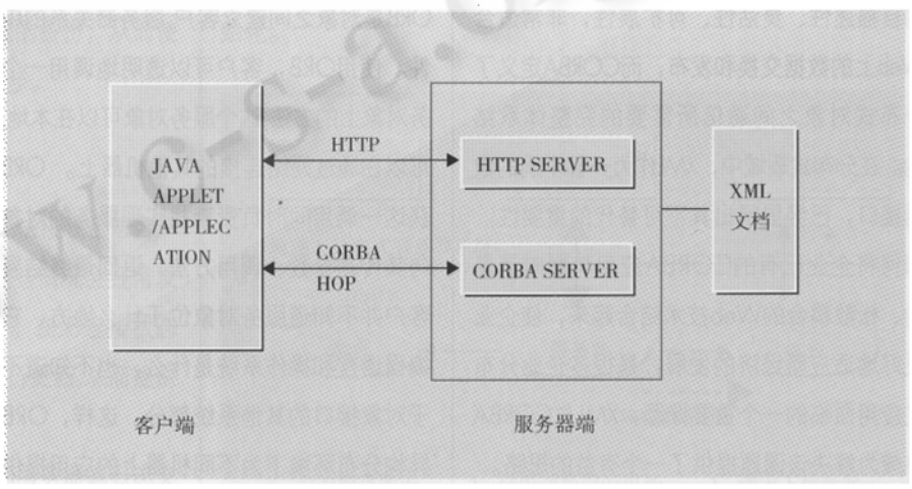
金融信息服务在它的商务服务结构中可以集成使用CORBA和XML。在金融信息服务的结构中面临的最大挑战性问题是如何获得用户需求。

从技术角度来看，金融信息服务需要很好的适应现存的分布式CORBA体系结构。同时它采用高度灵活的结构，以适应用户可能的频繁的界面变化和事先不知道的信用卡处理包。对于应用端和商务数据来说，XML显然是很好的选择。对于应用数据来说，它需要表示购物篮和发货单，而XML可以满足所需要的灵活性。这可以通过采用DOM来解析XML文档,以及采用一个格式服务的方法来为WEB产生HTML文件等来实现。

金融信息服务系统以一系列基于CORBA的服务为中心。这些服务包括：购物篮服务、发货单服务、XML格式服务和信用卡服务。购物篮服务用来管理购物篮，发货单服务为发货单处理提供服务，XML格式服务用来提供XML格式服务，接收XML数据流和象HTML和样式单等格式识别文件。

在金融信息服务系统中,采用JAVA来实现CORBA和XML的集成。

JAVA客户，包括APPLET和APPLICATION，通过桩（STUB）代码向本地的JAVA ORB发出请求，本地ORB再与服务器方的JAVA ORB进行IIOP通信，服务方ORB根据请求的内容调用相关的骨架（SKELETON）代码由指定的对象实现来完成请求，并将请求结果按原路返回给客户。JAVA ORB负责远程对象请求的生成，编码，传输等动作如下图所示：



本文通过一个用户信息查询操作来说明CORBA和XML在金融信息服务系统中的集成运用。

4.1 用接口定义语言IDL设计接口程序

接口程序指明应用程序中用到的对象及它们的接口,例子中的接口程序CUSTOMER.IDL内容如下：

```
module Customerapp{
  interface Customer {
  string search();
```

```
    }
    }
```

接口Customer通过方法search来根据指定用户名搜索该用户的详细信息。

4.2 服务器端程序

```
    try {
    //初始化ORB
      org.omg.CORBA.ORB orb=org.omg.CORBA.
                    ORB.init(args,null);
    //取根POA的一个引用
     POA rootPOA=POAHelper.narrow(orb.resolve_initial
                _references("RootPOA"));
    //创建持久POA的策略
    org.omg.CORBA.Policy[ ] policies=
        rootPOA.create_lifespan_policy(Lifespan_
            PolicyValue.PERSISTENT)
    };
    //用正确的策略创建myPOA
    POA myPOA=rootPOA.create_POA("customer_poa",
      RootPOA.the_POAManager(),policies);
    //创建servant
    CustomerImpl customerServant=
    new CustomerImpl(args[0]);
      //决定servant的ID
    byte[ ] managerId="CustomerManager".getBytes();
    //用myPOA上的ID激活servant
    myPOA.activate_object_with_id(managerId,
    CustomerServant);
      //激活POA管理器
    RootPOA.the_POAManager().activate();
    System.out.println(myPOA.servant_to_reference(
        CustomerServant)+"has been ready");
    //等待接入请求
    Orb.run();
    }
    catch (Exception e) {
      e.printStackTrace();
    }
    }
    //用DOM解析XML文档
    public class CustomerImpl extends
    Customerapp.CustomerPOA{
      public void search(String uname)
    {
    //为解析XML作准备，创建DocumentBuilderFactory实例,指定DocumentBuilder
     DocumentBuilderFactory dbf=
       DocumentBuilderFactory.newInstance();
    DocumentBuilder db = null;
    try {
    db = dbf.newDocumentBuilder();
    } catch (ParserConfigurationException pce)
{
    System.err.println(pce); //出异常时输出异常信息，然后退出，下同
    System.exit(1);
    }
     Document doc=null;
    try { //读入名字为uname的xml文件
    doc = db.parse(uname);
    } catch (DOMException dom) {
    System.err.println(dom.getMessage());
    System.exit(1);
    } catch (IOException ioe) {
    System.err.println(ioe);
    System.exit(1);
    }
    if (doc!=null)
    {
     printDOMTree(doc);
    }
    ..... 解析XML文件
    public void printDOMTree(Node node) {
     int type=node.getNodeType();
     if (type==Node.ELEMENT_NODE) {
    //如果找到了该用户,把他的信息读出来
      if (node.getNodeName==name) and
       (node.getNodeValue()==text1.text)) {
       Node kid=node.getFirstChild();
       Int I=1;
       for (kid=node.getNextSibling();
        kid!=null;kid=kid.getNextSibling())
          text[I+1].setText(kid.getNodeValue
());
      }
     }
    }
    }
```

4.3 客户端程序（略）

5 结束语

本文介绍了XML语言和CORBA分布式技术，并且进而从三个角度对XML语言和CORBA技术结合进行了讨论，最后以实例说明了怎样在金融信息服务中集成XML和CORBA。

参考文献

1 OMG CORBA系统结构、原理于规范、电子工业出版社，2000。

2 RBERT ORFALI，DAN HARKEY，JERI ADWARDS，智能CORBA，电子工业出版社，1999。

3 BORLAND/INPRISE，VISIBROKER FOR JAVA 开发人员指南，机械工业出版社，2000。