

# Web 服务在 CRM 中的应用研究

## Research on the Application of Web Services in CRM

张新香 (武汉中南财经政法大学信息学院 430064)

**摘要:**本文分析了传统基于中间件技术的多层系统平台架构的缺陷,结合 CRM 系统的特征,提出了基于 Web Services 的 CRM 系统平台方案,并分析了该方案的优越性,最后给出了 CRM 系统中的产品信息服务设计过程实例。

**关键词:**客户关系管理 Web 服务 SOAP WSDL UDDI

### 1 CRM 系统的特征

CRM 是指利用信息技术对客户资源进行集中式管理,把经过分析和处理的客户信息与有关客户的各种业务领域进行无缝接合,让市场营销、产品销售、客户服务和技术等经营流程信息化,实现客户资源有效利用的管理软件系统。其核心思想是以客户为中心,提高客户满意度和忠诚度,从而吸引和保留更多的客户,最终提升企业利润,提高企业的竞争力。其目的是为了保留客户、提高效率、拓展市场。

#### 1.1 系统结构特征

CRM 系统的核心功能是处理有关客户的数据,因此客户数据直接决定了系统的结构。而客户数据具有分散性、动态性、复杂性等特点。客户数据来源渠道广泛,且随时间的变化而变化,客户数据除包括个人基本信息外,还包括客户曾经购买的产品列表、客户对产品喜好程度等潜在数据。

#### 1.2 系统功能特征

CRM 系统为了适应企业日常业务的变化,必须频繁的扩充或更改。一般而言,在 CRM 系统的实施运行过程中通常会每几个月甚至每几个星期就要求增加新的功能模块,如果企业员工或用户的需求发生变化,也会要求更改系统,尤其是当系统是为企业的某个客户、合作伙伴或员工具体定制时,更改更频繁,企业业务的流程会每周或几天要求更改。

#### 1.3 系统终端用户特征

CRM 系统涉及到各种层次的用户,包括企业的决策者、日常管理人员、营销管理人员、业务员、分销商、客户等。这些用户特征各异,需求不同,对系统的要求也不同。

### 2 传统基于中间件技术的多层系统平台

基于中间件技术的多层系统平台不能满足 CRM 系统的要求,原因如下:

#### 2.1 要求系统同构

DCOM 技术的部署主要依赖于微软平台,而 CORBA 技术虽然提出是为了解决异构系统的集成,但是严格的说,

不同的系统也要求部署同一 CORBA 厂商的 ORB 产品,EJB 也一样要求在对等体系结构间才能进行通信,对基于不同体系结构的应用系统将无法实现协作,但要求所有参与者都采用同一种体系结构将是不现实的。

#### 2.2 防火墙的问题

现在的应用系统出于安全性考虑,都在系统外面部署了防火墙,一般只开放 80 端口,允许 HTTP 包通过,传统的基于中间件技术的多层系统平台虽然采用种种技术解决防火墙问题,但是都显得过于复杂。

#### 2.3 很难满足 CRM 系统的频繁扩充

CRM 系统的更新扩充频繁,如果采用传统系统平台,将会大量修改中间业务逻辑层,中间件环境中组件对应的程序代码维护代价太高,并且各类用户的不同需求难以得到满足。

为了克服传统平台中存在的缺陷,我们不妨采用 Web Services 技术。

### 3 基于 Web Services 的 CRM 平台的优势

#### 3.1 Web Services 的概念

Web 服务就是利用 Web 接口提供的某些功能程序段,通过 Internet 标准协议或规范,可以很容易访问这些功能。Web 服务就像一个封装了一定功能的黑盒子,用户可以重用而不必关心它是如何实现的,Web 服务定义了良好的接口,这些接口描述了它所提供的服务,用户可以利用这些接口来调用 Web 服务提供的功能,这样,开发者可通过远程服务、本地服务和用户代码合在一起创建应用程序。

#### 3.2 Web Services 的体系结构

Web Services 的体系结构如图 1 所示:

Web 服务体系结构基于三种角色(服务提供者、服务注册中心、服务请求者)之间交互。Web 服务提供者向 UDDI 注册表注册其 Web 服务;Web 服务请求者查询 UDDI 注册表,寻找并发现所需的 Web 服务,根据 UDDI 注册表返回的定位

信息,向 Web 服务提供者请求并获取描述 Web 服务接口的 WSDL 文件。根据 WSDL 文件生成相应的 SOAP 调用消息,发送给 Web 服务提供者,以实现 Web 服务的调用;Web 服务提供者执行相应的 Web 服务,将服务结果以 SOAP 响应消息的形式返回给 Web 服务请求者。

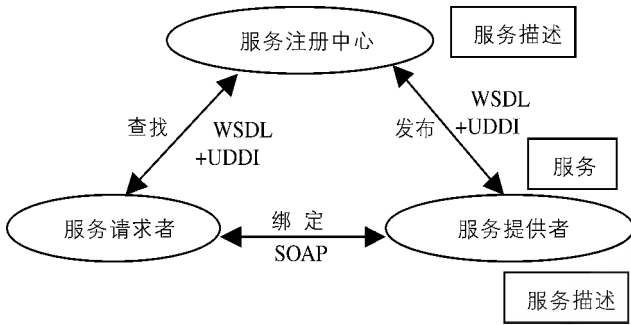


图 1 Web 服务的体系结构

### 3.3 优势分析

构建基于 Web Services 的 CRM 系统平台如图 2 所示,它能有效解决传统系统平台中存在的缺陷,具有以下优势:

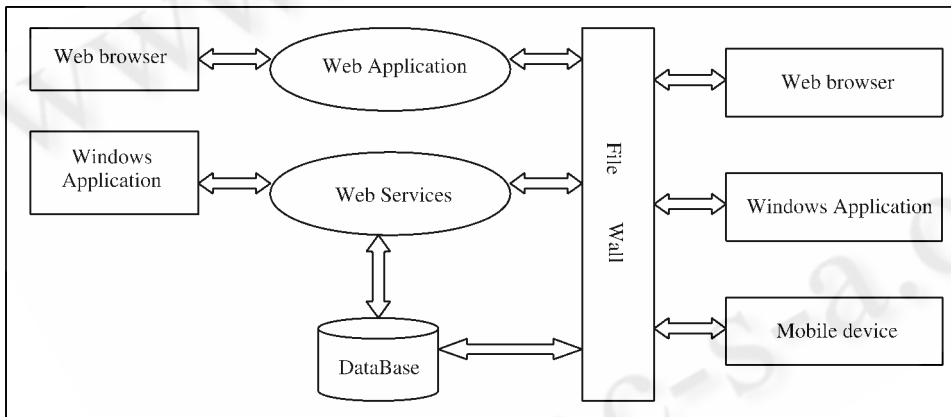


图 2 基于 Web Services 的 CRM 系统平台架构

(1) 不要求系统同构,系统集成简单。采用 Web 服务可构建独立于平台、组建模型和编程语言的 CRM 系统,其可跨平台与任何系统实现无缝连接。具体方案是以 Web 服务接口的形式将各种应用程序和信息系统进行封装、组合和集成,将应用程序改变成可重用的和柔性的组件。这种方案无需考虑应用实施的环境是何种平台和设备,无论是企业的 ERP 系统还是办公自动化,无论是基于 Windows 平台的应用还是基于 UNIX 上的应用系统,都可远程调用和集成 CRM 系统中的 Web 服务。

(2) 实现跨防火墙通信。Web 服务使用的协议是 SOAP (Simple Object Access Protocol), SOAP 为在一个松散的、分

布的环境中使用 XML 对等地交换结构化的和类型化的信息提供了一个简单的轻量级机制。SOAP 本身并不定义任何应用语义,它只是定义了一种简单的机制,通过一个模块化的包装模型和对模块中特定格式编码的数据重码机制来表示语义。SOAP 的这项能力被多种类型的系统用于从消息系统到 RPC 的延伸,并且 SOAP 采用的是超文本传输协议传输,这样就不会受防火墙或代理服务器的限制,因而实现跨防火墙通信。

(3) 扩展性强,维护成本低。由于 Web 服务是通过 XML 进行数据的传输,采用开放通用的 Internet 标准,所以对客户端的平台没有限制,企业各个部门就可以根据实际情况和用户习惯,借助 Web 服务来开发自己的用户程序,因为所有的事务处理集中在 Web 服务上,因而客户端程序开发的工作是大大减少,成本相对降低。

(4) 实现了软件和数据的重用,增强共享性。Web 服务在允许重用代码的同时,可以重用代码背后的数据,实现数据的一次录入,多次使用。另一方面企业所有的核心业务和系统功能被抽象出来,以 Web Services 方式发布出去,实现了软件的重用,也很方便让核心业务和系统功能在企业各部门之间或不同企业间共享。

## 4 应用实例分析

基于以上的分析,本文以 CRM 系统中产品订单管理子系统涉及到的产品信息服务为例,具体介绍 Web 服务的开发流程。

### 4.1 Web 服务的设计

某公司的 CRM 系统中涉及到产品订单管理子系统,该子系统要求可以注册新客户,修改老客户的用户资料,允许客户浏览产品信息,根据给定

条件查找产品信息,预定产品,填写订单并对自己的订单情况进行报表打印。根据分析,设计的功能模块是:客户信息处理、产品信息浏览、购物车操作、客户订单处理和报表打印。相应的 Web 服务是, ProductService (产品信息服务)、Cartservice (购物车服务)、CustomerInfoService (客户信息服务)、OrderService (客户订单服务)、OrderReportService (订单报表服务) 等。

### 4.2 Web 服务的实现

Web 服务的实现可以使用任何一种编程语言来开发 Web 服务方法,也有常见的情况,利用 Web 服务调用传统的组件,然后与其他 Web 服务交互,组件可以是 COM、DCOM、

EJB、CORBA 等。限于篇幅有限,本文只写出 ProductService 服务类和方法的实现,采用 Visual Studio. Net 开发。

产品信息服务主要是实现产品信息的浏览,要用到数据库表 Product,主要字段有 ProductID (int, 产品号), Product-Name (nvarchar, 产品名), Unitprice (money, 单价), Product-supplier (nvarchar, 供应商)。服务类名为 wsproduct,核心代码如下:

```
namespace wsproduct
{ [ Webservice ( namespace = "product/Webservice" ) ]
public class wsproduct: system. Web. services. Webservice
{ //定义数据库连接串,依具体情况定,在此内容略}
private const string strconnection = "....."
//定义 Web 方法
[ WebMethod[ Description = "查询产品信息" ]
public bool Query product ( int product id, string productname)
{ bool bretu = false;
sqlconnection conn = new sqlconnection ( str connection );
try
{ conn open ( );
sqlcommand cmd = new sqlcommand ( " select * from product;
where product ID = productid and productName = productname", conn );
cmd. Execute sql( );
conn. close( );
bretu = true;
}
catch ( Exception e)
{ throw e; }
finally
{ if ( conn. state = = connection state . open ) conn.
close( ); }
return bretu; }
}
```

#### 4.3 Web 服务的描述、注册及发布

Web 服务定义好后,就将它部署到应用服务器上,生成 WSDL 描述。

```
<? xml version = "1.0" encoding = "UTF - 8" ? >
< wsdl: definitions target namespace = "urn : wsproduct" >
< wsdl: types/ >
< wsdl: message name = "query product Response" >
< wsdl: message name = "query product Request" / >
< / >
```

```
< wsdl: prottype name = "wsproduct" >
< wsdl: binding name = "wsproduct soap binding"
type = "impl: wsproduct" >
< wsdl soap: binding style = "rpc" transport = "http://schemas.xmlsoap.org/soap/http" / >
< wsdl: operation name = "query product" >
< wsdl soap: operation soapaction = " " / >
< wsdl: input name = "query product Resquest" >
< wsdl: output name = "query product Response" >
< /wsdl: operation >
< /wsdl: binding >
< wsdl: service name = "productservice" >
< /wsdl: definitions >
```

通过使用 WSDL 将 productservice 这个 Web 服务进行了结构化描述后,为了加强 Web 服务的互操作能力和灾难恢复时的链接保持能力,需要将这个 Web 服务注册到 UDDI 注册中心中去。管理员只要在注册中心中先注册一个商业实体,它是在线服务提供商,然后依次将需要发布的 Web 服务注册到该商业实体下。同时还预先注册一个 Service Type (tmodel),这个 tmodel 描述了需要发布的 Web 服务的调用规范,具体内容是上面定义的 WSDL 文档。在 UDDI 中,注册的是描述的连接。

#### 4.4 Web 服务的调用

Web 服务在 UDDI 注册中心注册以后,客户程序中可编写调用服务的编码,去访问一个端口或者同样的服务接口,实现灵活调用,也即实现了客户程序与 Web 服务的无缝集成。

## 5 结束语

Web 服务是一个创建可互操作的分布式应用程序的新平台,它使用 XML 标准遮蔽数据格式差异和平台差异,它松散耦合应用的特性和完好的封装性能克服传统基于中间件技术的 B/S 和 C/S 结合的平台架构的缺点,便利地实现了 CRM 系统中客户程序与商家 Web 服务的无缝集成,提高商家服务与客户系统的互操作性、系统功能复用性及拓展业务,有利于提高商家的客户关系管理服务水平。

#### 参考文献

- 1 柴晓路, Web 服务架构与开放互操作技术 [ M ], 清华大学出版社, 2002。
- 2 秋林、力士奇, 客户关系管理 [ M ], 清华大学出版社, 2002。
- 3 美 Price, J. 等著, 窦芳等译, Visual Studio. NET 从入门到精通, 电子工业出版社, 2003。