基于 SOA 的呼叫中心系统①

谢琼琳 黎福海 王绍源 (湖南大学 电气与信息工程学院 湖南 长沙 410082)

摘 要: 提出了一种基于 SOA 的呼叫中心系统。该系统在分析了面向服务架构 SOA 模型的基础上，采用企业服务总线 ESB 原理及 Web Service 技术搭建，实现了松耦合、模块化以满足更加灵活的业务需求，并应用于实际电信系统建设

关键词: 呼叫中心; 面向服务架构 SOA; 企业服务总线 ESB; Web Service

SOA-Based Call Center System

XIE Qiong-Lin, LI Fu-Hai, WANG Shao-Yuan

(College of Electrical and Information Engineering, Hunan University, Changsha 410082, China)

Abstract: An SOA-based Call Center System is proposed in this paper. On the basis of analyzing the model of service-oriented architecture SOA, we use the principle of Enterprise Service Bus ESB and Web Service technology to build the system, which achieves loosely coupled and modular to meet the more flexible business needs. And it has been applied to the actual construction of the telecommunications system.

Keywords: call center; service-oriented architecture SOA; enterprise service bus ESB; web service

呼叫中心从 2000 开始，就一直致力于支撑中国电信客户服务信息化工作，并在技术、业务等方面不断改进，根据中国电信集团公司 ITSP 关于呼叫中心的整体规划以及相关的规范要求，对 SOA 的需求来源于需要使业务 IT 系统变得更加灵活，以适应业务中的改变。通过允许强定义的关系和依然灵活的特定实现，IT 系统既可以利用现有系统的功能，又可以准备在以后做一些改变来满足它们之间交互的需要。

1 引言

呼叫中心系统采用灵活的、基于模块化/SOA 架构的解决方案框架，为企业信息集成提供了一种面向功能的，松耦合的系统架构方法。通过粗细不同的服务粒度进行整合，在业务流程和服务层对服务进行了封装，使得不必要公开的细节被隐藏，更有利于业务的的划分和流程的调用。服务之间的松耦合，保持了各服务的高度可用性，使得业务变更能够更灵活的进行组合，统一的接口定义，使得新系统可以很方便的集成现有系统并连接外部系统，实现数据共享和交互。本文首先分析了面向服务的体系结构 SOA 及企业服务总线 ESB，在此基础上对基于 SOA 的呼叫中心系统架构进行了研究。

2 SOA概述

SOA(Service-Oriented Architecture)，即面向服务架构。它是指为了解决在 Internet 环境下业务集成的需要，通过连接能完成特定任务的独立功能实体实现的一种软件系统架构。它拥有独立的功能实体，采用大数据量的方式进行信息交换以实现低频率访问，基于文本的消息传递方式不包含任何处理逻辑和数据类型，可以跟不同语言不同平台间的交互很好的兼容。

2.1 SOA 分层模型

SOA 系统中的不同的功能模块可以被分为 7 个层次[1]，如图 1 所示。

SOA 各层具体描述如下:

第一层是系统已经存在程序资源，也叫一流系同。

① 收稿时间:2009-12-02;收到修改稿时间:2010-01-14

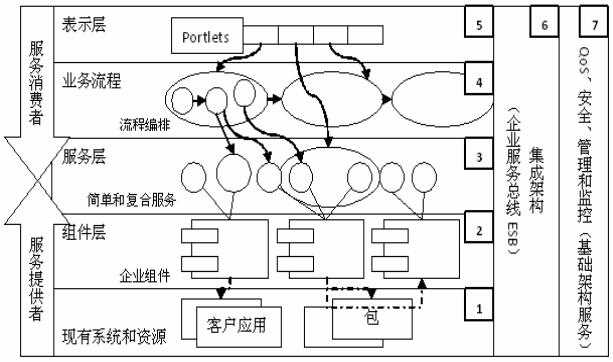

图 1　SOA 分层模

第二层是组建层，用于封装地系统的功能。

第三层是服务层，在本层中用底层功能组件来构建应用所需要的不同功能的服务。总的来说，SOA 中的服务可以被映射成具体系统中的任何功能模块，但是从功能性方面可大致划分为三种类型：(1)业务服务或者是业务过程：可以暴露给外部用户或者合作伙伴使用的服务。(2)业务功能服务：完成一些具体的业务操作，也会被更上层的业务服务调用，一般不会暴露给外部用户直接调用。(3)技术功能服务：完成一些底层的技术功能。

第四层是业务流程层，在这一层中利用已经封装好的各种服务来构建商业系统中的业务流程。

第五层是表示层，用于向用户提供用户接口服务。

第六层是企业服务总线层 ESB(Enterprise Service Bus)，作为一个集成环境用于支持一至五层的运行。

第七层主要为整个 SOA 系统提供一些辅助的功能。

2.2 ESB 原理

ESB 是由中间件技术实现并支持 SOA 的一组基础架构，支持异构环境中的服务、消息以及基于事件的交互。作为 SOA 的一个重要组成部分，ESB 的主要作用是连接服务。它对服务进行集中管理，并为参与通信的双方提供中介服务，包括元数据及服务注册管理，消息路由，以及转化传输协议和消息格式等。

基于 ESB 的上述功能，服务请求者和服务提供者可以不需要关心各自的位置和具体的实现技术，双方可以在保持接口不变的情况下各自独立演变[2]。但服务双方必须通过 ESB 发布查找才能进行交互。

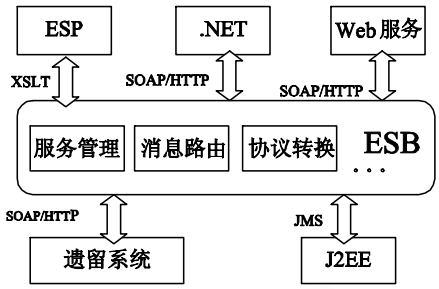

图 2　ESB 体系结构

2.3 OA 主要组件

SOA 计算环境的主要组件包括：服务运行时环境、服务总线、服务注册库、服务网关和服务组装引擎等[3]。

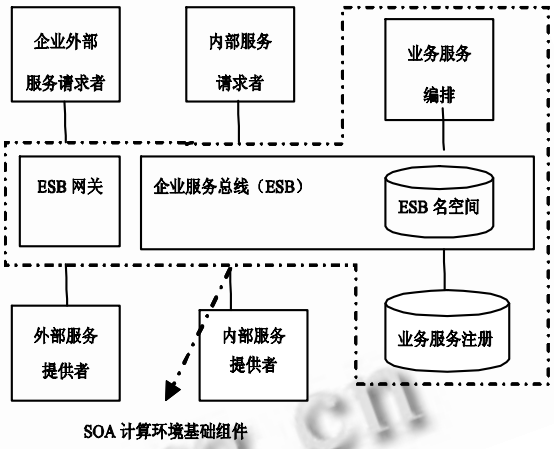

图 3　SOA 计算环境主要组件

服务运行时环境提供服务。（服务组件）的部署、运行和管理能力，支持服务编程模型，保证系统的安全和性能等质量要素；服务总线提供服务中介的能力，使得服务使用者能够以技术透明和位置透明的方式来访问服务；服务注册库支持存储和访问服务的描述信息，是实现服务中介、管理服务的重要基础；而服务组装引擎，则将服务组装为服务流程，完成一个业务过程；服务网关用于在不同服务计算环境的边界进行服务翻译。

2.4 OA 主要技术

SOA 具体的实现技术有很多，包括 Web Service，Session Bean，JINI 等，目前而言最适合实现 SOA 架构的就是 Web Service 技术。

基于 Web Service 的 SOA 架构组成及原理如下[4]：

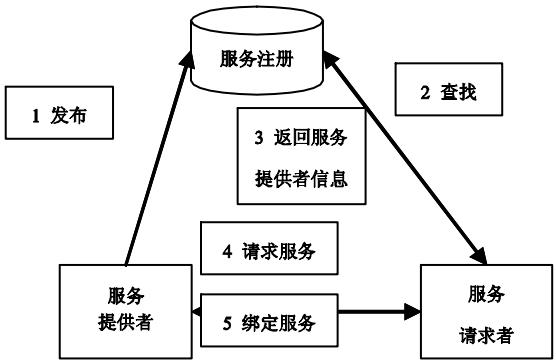

图 4 基于 Web 服务协议

(1)服务提供者：服务提供者是一个可通过网络寻址的实体，它通过 UDDI 服务注册进行发布，并接受和执行来自使用者的请求；

(2)服务使用者：服务使用者是一组使用服务提供者所提供的一项或多项服务的组件，它通过发送请求到 UDDI 查找所需服务，得到来自服务提供者的服务；

(3)服务储备库：服务储备库包含服务的描述，服务提供者在该储备库中注册其服务，而服务使用者访问该储备库去查找及发现所提供的服务。

3 基于SOA的呼叫中心系统

3.1 叫中心系统的体系结构

呼叫中心体系结构如图 5：

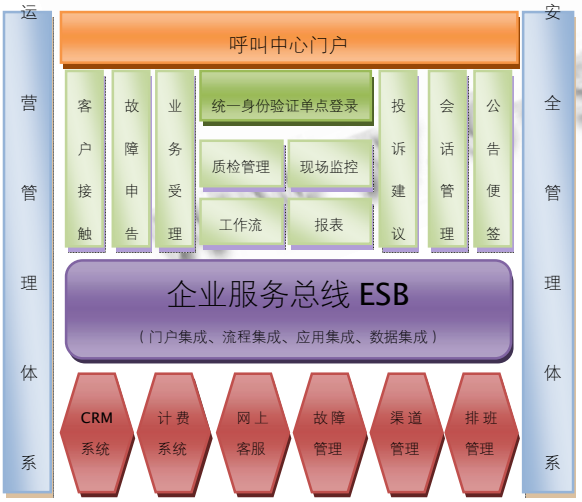

图 5 呼叫中心系统体系结构

呼叫中心系统在基于 SOA 的 7 层模型的基础上，采用了 3 层体系结构，各个层次实现不同的功能并相对独立。手机、短信、Email 等访问方式接入到呼叫中心门户，门户的界面展现为接入层，负责发送业务、服务请求，不同渠道请求的同类服务在底层实现是一致的，对应在业务基础架构中的前端应用程序；企业服务总线属于业务逻辑层，负责接收服务请求并执行相应操作，各个业务逻辑功能模块可以重用，在开发和维护上较两层体系架构更方便。数据库为数据资源层，是数据保存的物理层，主要为数据库以及文件系统。该层存储各类数据，包含业务数据、客户信息等等，该层封装了业务逻辑层对数据库的访问，使得业务逻辑层更专注于应用功能方面的设计和实现。同时，该层也包含了系统本身、自身与其他系统集成的配置数据，各项配置信息存储在数据库中通过其它的二层调用实现，它们和业务实现模块成为架构中的业务服务部分。

3.2 叫中心系统的技术实现

3.2.1 服务模块的实现。

本文选自自顶向下的开发模式，即先完成 WSDL 文件的编写，然后生成代码框架，接着填充代码框架，最后发布服务。

下面以员工信息为例介绍服务模块的实现步骤。

首先，根据需求分析完成基本服务和操作的定义。

表 1 员工信息定义

```
<xsd:complexType name="StaffInfo">
        <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="1"
name="StaffId" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1"
name="StaffName" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1"
name="Staffpass" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1"
name="StaffLastLoginTime" type="xsd:date/>
            <xsd:element maxOccurs="1" minOccurs="1"
name="StaffLoginIp" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1"
name="StaffLoginTimes" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1"
name="StafflsPass" type="xsd:string"/>
            ...
        </xsd:sequence>
</xsd:complexTyPe>
```

上表为简化后的客户信息描述。由于信息保密需要，工作人员要先登录系统才能进行操作，此为还应区分不同

类型的人员具有不同的权限。相应就该定义 Login 登录、Logout 注销、GetCustList 查询客户信息等操作。

表 2 员工登陆请求及响应

```
<xsd:element name="login">
    <xsd:complexType>
        <xsd:seqence>
            <xsd:element maxOccurs="1"
minOccurs="1" name="StaffId" type="xsd:string"/>
            <xsd:element maxOccurs="1"
minOccurs="1" name="Staffpass" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexTyPe>
</xsd:element>

<xsd:element name="loginResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element maxOccurs="1"
minOccurs="1" name="valid" type="xsd:boolean"/>
            <xsd:element maxOccurs="1"
minOccurs="1" name="token" type="xsd:string"/>
            <xsd:element maxOccurs="1"
minOccurs="1" name="flag" type="xsd:int"/>
        </xsd:sequence>
    </xsd:complexTyPe>
</xsd:element>
```

登陆通过 Web 服务进行验证并获取一个令牌(token)，证明验证成功，以便进一步进行 Web 服务请求。传递两段数据(用户名和密码)，返回两段数据(字符串和整型标志位)。其中，valid 表示是否有效，token 表示一个令牌，flag 表示员工权限级别。

然后定义端口类型。

表 3 端口定义及操作

```
<wsdl:portType name="QueryCustTnfoMgrServicePortType">
        <wsdl:operation name="login">
            <wsdl:inPut message="tns:login"/>
            <wsdl:output message="tns:loginResponse"/)
        </wsdl:operation> .
        <wsdl:operation name="logout">
            <wsdl:input message="tns:logout"/>
            <wsdl:output message="tns:logoutResponse"/)
        </wsdl:operation>
        <wsdl:operation name="getCustInfoList">
            <wsdl:input message="tns:getCustInfoList"/>
            <wsdl:output
message="tns:getCustInfoListResponse"/>
</wsdl:PortTvpe>
```

接着完成 WSDL 绑定，指定消息格式、Web 服务的协议类型、在该端口公开的操作等。

表 4 WSDL 绑定

```
<wsdl:binding name="QueryCustInfoMgrServicePortBinding"
type="tns:QueryCustInfoMgrServivePortType">
        <soap:binding style="document"
transport="http://schema.xmlsoap.org/soap/http"/>
        <wsdl:operation name="login">
            <soap:operation
soapAction="http://www.ahu.edu.cn/QueryCustInfoMgrService/login"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/)
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
        </wsdl:operation>
        </soap:binding>
</wsdl:binding>
```

最后进行服务标记。

表 5 服务标记

```
<wsdl:service name="QueryStaffInfoMgrService">
  <wsdl:port binding="tns:QueryuserInfoMgrServicePortBinding"
name="QueryStaffInfoservicePort">
        <soap:address
location="http://localhost:8001/js_callcenter/services/QuerySta
ffInfoMgrService"/>
</wsdl:service>
```

此时就基本完成了用 WSDL 对服务模块的编写，接下来由此 wsdl 文件生成 Java 代码框架，利用工具进行转换，再进一步具体实现其中的服务。

3.2.2 企业服务总线 ESB 的实现。

当前有四种典型的 ESB 产品：开源的 Mule ESB，BEA 公司的 ESB，ServiceMix ESB 和中和威的 ESB[5]。

呼叫中心采用 BEA 公司的 AquaLogic Service Bus (ALSB)来构建 SOA 架构的服务总线。[6]ALSB 能运行在 Windows，Linux，Solaris 等系统上，并且提供了智能的消息代理、动态传送和转换，所有这些都支持与服务生命周期管理功能集成的异构服务端点[7]。

使用 ALSB 构建 SOA 服务总线的步骤是：首先创建服务总线项目，再创建项目中的代理服务、业务服务以及资源服务。并按照需求对服务总线进行配置(即导入 WSDL 资源文件、配置代理服务、配置业务服务)。然后进行代理服务和业务服务的连接，最终便可实现服务请求者通过 ESB 对服务提供者的调用。

(下转第 91 页)

(上接第 136 页)

4 结语

呼叫中心系统发展至今已具备了一定的应用基础，但从发展的眼光看，电信业务流程一直是在不断调整中持续优化的，如何有效地融合各种异构平台，解决结构和技术不兼容的问题，以灵活的架构满足灵活的业务，成为目前最为迫切的需要。SOA 作为一种新型的软件体系结构，只需将应用系统看作用来满足客户需求的一组服务功能集合进行服务编排融合，把应用程序改变成可重用的和柔性的组件。以 Web 服务形式提供，有效利用并发挥现有系统的价值，使不同企业的应用程序之间，企业与企业之间的部署变得更加一致，实现了企业间数据的共享，同时降低 IT 运维技术人员的维护复杂性，促进企业的高速发展。

参考文献

1 Arsanjani A. Service-Oriented modeling and architecture: How to identify, specify, and realize services for your SOA. Whitepaper from IBM, 2004. http://www-128.ibm.com/developerworks/webservices/library/we-soa-design

2 曾文英,赵跃龙.齐德昱.ESB 原理、构架、实现及应用.计算机工程与应用, 2008,44(25):225－228.

3 毛新生.SOA 原理方法实践.北京:电子工业出版社, 2007.7:6－13.

4 Newcomer E, Lomow G. Understanding SOA with Web services(中文版).徐涵译.北京:电子工业出版社, 2006.20－25.

5 丁昭华.基于 ESB 的企业应用集成技术研究与应用.[硕士学位论文].长沙:中南大学, 2007.

6 BEA AquaLogic Service Bus 简介. http://tech.51cto.com/art/200601/19830.htm.

7 用 SLA 保证 Web 服务. http://www.ibm.com/developr-works/cn/webservices/we-sla.

8 Davies J, Krishna A, Schorow D. SOA 权威指南:通过 BEA AquaLogic Service Bus 实现.倪志刚,王铭孜,黄兆勤译.北京:电子工业出版社, 2008.

9 Bieberstein N, Bose S, Fiammante M, Jones K, Shah R. 面向服务架构(SOA)指南.张宁译.北京:人民邮电出版社, 2008.