

SCA 规范在信用卡申请系统中的应用^①

Application of SCA in the Credit Cards Applying System

吕欢欢 (辽宁工程技术大学 电子与信息工程系 辽宁 葫芦岛 125105)

刘万军 (辽宁工程技术大学 软件学院 辽宁 葫芦岛 125105)

摘要: 针对现有信用卡申请系统存在的问题,论述了基于 SCA 框架建立银行信用卡申请系统的优势。首先介绍了服务组件架构 SCA 的基本概念、特点和组成,并结合具体案例详细描述了运用 SCA 框架构建信用卡申请系统架构的过程和策略,解决了现有信用卡申请系统存在的主要问题,使系统达到了高效、稳定、灵活的目的。

关键词: SCA 组件 模块 服务 架构

1 引言

信用卡市场目前处在高速发展时期,银行信用卡用户每年都在以 100% 以上的速度增长,未来几年的发展将会决定各家银行在市场最终的地位,每家银行都在快速的发展自己的用户,抢占更多客户资源,竞争非常激烈,传统的应用集成技术只能在一定程度上实现信用卡申请系统的应用集成需求,它们都有一个共同的缺陷:普遍采用某种专用中间件技术,以专有的、偏向于系统实现方式的 API 进行设计,系统间耦合度高,对业务需求适应性差。如何节约投资成本、追求工作高效、灵活适应需求变化,合理使用资源,降低违约风险,实现应用系统、业务数据和流程的无缝集成,实现各部门、各单位协同办公,为客户提供便捷、高效的服务成为银行信用卡申请系统亟待解决的重要课题。

本文提出了一种基于 SOA 架构应用 SCA 框架的系统集成解决方案,很好地满足了信用卡申请系统的各项应用集成需求。首先介绍了服务组件架构 SCA 的基本概念、特点和组成,以及信用卡申请系统的总体设计和构建,实现了应用系统的集成研究。

2 服务组件架构 SCA

2.1 SCA 概述

SCA(Service Component Architecture)^[1]即

“服务组件架构”,是 IBM 和 BEA 等公司提出的一套面向服务 SOA 编程模型或者说是编程规范,是 SOA 思想的一种实现方式。

SCA 简化了使用 SOA 构建应用系统的开发过程,它将传统中间件编程从业务逻辑分离出来,从而使开发人员集中精力编写业务逻辑,而不必将大量的时间花费在更为底层的技术实现上。SCA 可以自由绑定各种传输协议,SCA 还简化了业务组件的开发、组装和部署流程,提高了系统的可移植性、可重用性和灵活性,使系统的可测试性也有很大的提高。

2.2 SCA 规范

在 SCA 架构里面,一个组件去调用另外一个组件时,不需要知道对方的技术系统,这就是 SCA 规范的实质,也是 SCA 架构所要达到的目标。组件之间处于一种松散耦合的状态,不需要在自己的代码中加入对方组件的接口代码。SCA 规范可以使开发者更侧重于业务逻辑,而不需要关注诸如可靠性、安全性、事务等系统属性,这些属性都配置到配置文件中^[2]。SCA 的组成部分如图 1 所示。

(1) Component: 是 SCA 规范中最基本的元素,是粒度最小的执行单元或者实现单元,其封装了实际的业务逻辑,它是搭建应用系统的积木。

(2) Property: 用来控制 Component 的行为,在部署期间可以被改变。

^① 收稿时间:2009-02-25

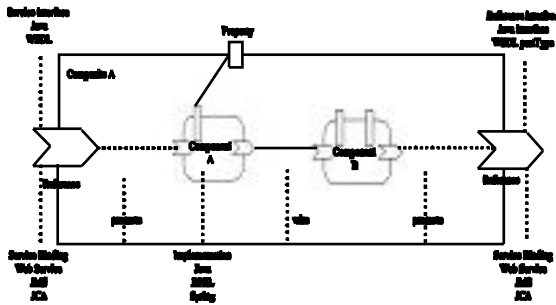


图1 SCA规范图

(3) Service: 用来让其他组件调用, 它是一个接口, 目前只有基于 JAVA 的接口和 WSDL 的 PortType 接口。

(4) Reference: 一个组件可能需要调用其他组件, 需要创建一个其他组件的引用。

(5) Composite: 它描述了一个由互相连接的 Component 所构成的集合, Composite 也声明了 Service 和 Reference, 它们被暴露到 Composite 外部, Composite 的 Service 和 Reference 是 Composite 内部的 Component 的 Service 和 Reference 的升级。一个 Composite 内部的 Component 彼此连接就如同创建一个运行在同一进程中的紧耦合的应用程序。将 Composite 通过 Service 和 Reference 连接在一起, 则形成了一个更加松耦合的系统, 系统中的每一个 Composite 都可能运行在一个单独的进程或处理器中, 在网络中通过各种传输协议绑定连接起来。

(6) Wire: 是连接 Service 和 Reference 的连线。

(7) Promote: 是 Wire 的特殊表现形式, 是把 Component 级别 Service 或者 Reference 升级为 Composite 级别的连线。

3 基于SCA规范的信用卡申请系统总体设计

3.1 系统架构模型设计

本文设计的基于 SCA 的应用集成参考架构, 是以 SCA、SDO(Service Data Object)、DAS(Data Access Service)相结合的应用架构, 采用分层的方法设计成五层, 分别是数据源层、数据服务层、SCA 组件构建和组装层、服务发布与调用代理层、服务调用层, 集成架构模型如图 2 所示。

数据服务层 DAS 屏蔽了底层异构数据源的差异性, 向 SCA 组件层提供 SDO 数据服务, SDO 可以贯

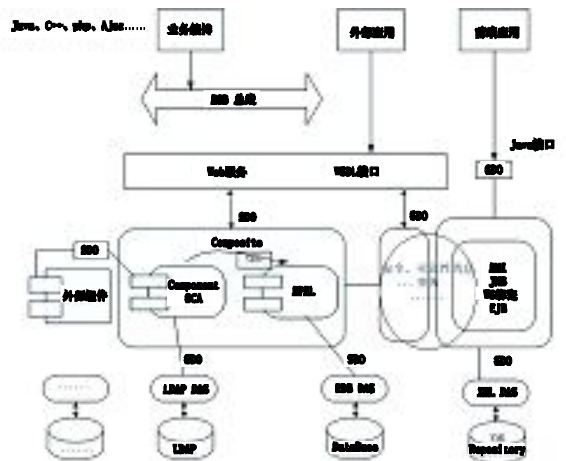


图2 基于SCA的应用集成架构

穿于整个集成架构模型中, 即除了数据服务层屏蔽数据源层的技术细节外, 其他层与层之间或者同一层的组件之间交互的数据单元可以部分或者全部为 SDO。SCA 组件构建与组装层是使用 SCA 编程模型, 把已有的或者新开发的应用程序功能单元封装成SCA组件并进行组装。服务发布与代理层相当于服务代理者, 服务调用层相当于服务消费者。

3.2 系统业务流程设计

当申请者向银行工作人员提供相关的个人信息(身份证、申请卡种等)后, 银行工作人员需要将申请者的信息录入系统, 并送出案件, 系统将会调用行内服务组件进行查询(包括主机上的取疵户、既有客户查询、系统内部的黑名单和伪件查询), 并将问题件自动婉拒。若行内查询通过, 则送至行外查询, 调用职业状况服务组件、公共付费记录组件、社会信用评价组件进行相关查询, 若行外查询通过, 则将案件派发给征信人员, 征信人员通过电话照会后, 填写征信照会记录且给出征信意见, 并将案件送至审核人员, 审核人员可以参考系统给出的评分结果以及征信意见, 给予审核结果[3]。

从银行的业务角度, 该业务流程的外部参与者为最终用户(申请者), 内部参与者包括银行的接待人员、征信人员, 审核人员和制卡人员。从技术的角度, 该业务流程包括自动化的内部功能(行内查询)和外部功能(行外查询), 也包括人工活动。

该系统针对我国正在飞速发展的信用卡申请现状提出了解决方案, 根据用户提供的身份证信息查出用

户的个人基本信息、职业状况信息、公共付费记录、社会信用评价, 据此评估客户是否能够申请信用卡。

3.3 服务模块的构建与组装

SCA 是通过 Composite(模块)将 SCA 的组件集成在一起的, Composite 相当于 J2EE 里面的 Module, 或者说模块, 这些组件可以是不同类型的组件, 比如 POJO、EJB、流程组件等。在构建模块时, 模块还可以包含服务、引用、模块件交互的连线、属性等元素, 模块也可以作为一个模块的一个组件实现, 形成更高级别的模块。模块有选择的将组件的接口对外公开, 提供了标准的接口让外部调用, 模块中包含多个组件, 组件之间通过接口和引用之间的连线(Wire)来达到调用的目的。模块内对外部服务的引用, 表示对其他 SCA 模块提供的远程服务进行调用, 模块也可以通过绑定对非 SCA 模块提供的服务进行调用, 其调用方式与访问模块内其他组件提供的任何服务一样。

根据系统需求, 确定了信用卡申请系统中行内查询和行外查询两个核心模块, 每个模块下又包括各自不同的服务组件, 另外, 也可以在这些服务和引用之间的连线上设定需要的 Qos 要求, 比如事务、安全等模块。根据系统的业务需求, 系统的系统集成结构如图 3 所示:

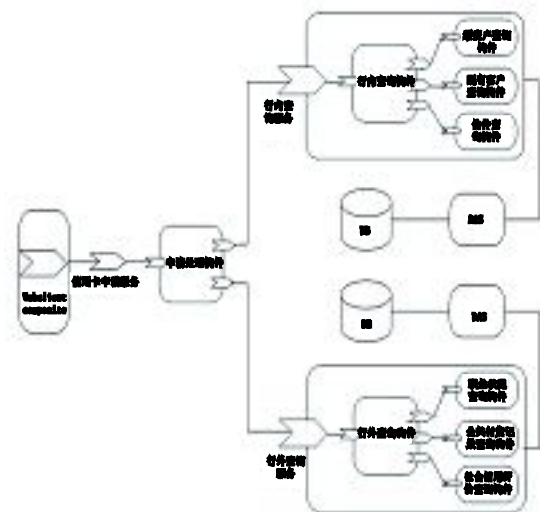


图 3 信用卡申请系统集成结构图

在构建的过程中, 把多少个服务组件放在一个模块中, 或者把哪些服务放在一起主要取决于业务需求和部署上的灵活性要求, 一个模块的内容也可以通过

包含被另一个模块使用。模块可以作为 SCA 系统的一个单元进行发布, 也可以通过被包含的方式或一个组件实现的方式被部署^[4,5]。

4 基于SCA规范的信用卡申请系统总体设计

4.1 组件及其实现

组件(Component)作为构建应用的积木, 为我们提供服务, 可以把传统的实现包装成 SCA 中的服务组件, 但组件的实现对外是透明的, 调用者无需知道该服务是如何实现, 以及采用什么技术实现的^[6,7]。

实现是业务逻辑的具体实现, 提供服务, 而且需要引用。SCA 允许不同方式的实现技术, 如 J2EE、BPEL、C++、JavaScript、SQL 等等。并且 SCA 通过扩展机制来引入新的实现类型, 模块也可以作为模块的一个组件实现, 从而形成更高级别的模块。下面的配置显示了组件实现的类型为模块:

```
<component name="ReplyComponent">
  <implementation.composite
    name="ReplyComposite"
    scdlLocation="./reply.scdl"/>
</component>
```

同一个实现可以通过不同的引用和属性配置成为多个组件。例如, 不同的组件通过对同一实现配置不同的引用和设置具体的属性值, 来调用不同的服务。组件通过连线来相互连接调用从而组成了模块(Composite)。

4.2 服务模块的实现与服务发布

本系统集成过程中, 确定了行内查询和行外查询两个模块, 每个模块又包括不同组件。模块可以作为 SCA 系统的一个单元而进行发布, 也可以通过被包含的方式活一个组件实现的方式被部署。在这里我们通过 Web 服务实现模块, 并将服务模块发布成 Web Service 供外部调用。创建信用卡申请的一个 Web Service 实现模块主要完成的工作如下:

(1) 创建 WSDL 文件, SCA 创建的 Web Service 将会以 WSDL 的 portType 作为 SCA 的服务接口。本系统中一个服务的 WSDL 文件形式如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions      targetNamespace="
http://outcheck"xmlns:tns="http://outcheck"
xmlns:wSDL="http://schemas.xmlsoap.org
```

```

/wsdl/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="outcheck">
.....>
.....
</wsdl:definitions>

```

(2) 创建 SCA 的 Java 实现类, 因为要实现 Web Service 的远程调用, 所以需要在 Java 接口中加入远程调用注释 @Remotable 和服务注释 @Service, 同时也需要在 Java 实现类中加上服务注释 @Service。

(3) 配置 SCA 服务模块, 主要是在配置文件中加上 Web Service 绑定。

4.3 服务模块的引用

Web Service 引用模块, 不实现服务, 只通过 Web Service 来引用前面所创建的服务模块的服务。创建一个信用卡申请的 Web Service 引用模块主要完成下面几个工作。

创建 WSDL 文件, SCA 创建的 Web Service 应用模块会通过 <binding.wsdlElement=...#wsdl.port...> 实现 Web Service 的绑定, 这里不需要另外创建 WSDL, 直接用 4.2 节的 WSDL 文件即可。

创建 SCA 的 Java 实现类, 一个是 Java 接口类, 另一个是 Java 实现类。在 Java 实现类中直接调用远程接口的方法即可。

配置 SCA 服务模块, 主要是在配置文件中通过引用 (Reference) 加上 Web Service 的绑定^[8]。

通过上面的工作, 基本上就完成了服务模块的引用。

4.4 系统运行情况评价

经过测试, 本系统在 Apache Tuscany 的运行环境中能够稳定、高效地运行。各个服务之间的转换时间和效率比以往系统有了较大的改善。传统信用卡申请系统中, 需要运行多个系统才能达到目的。采用面向服务设计, 各个系统之间松散耦合, 便于插拔; 采用先进 SCA 进行设计, 对于复杂多变的信用卡业务可以利用基本单元快速组装、重复利用, 以迅速满足业

务需求, 达到随需而动, 并且能非常轻松的集成银行现有的诸多系统。

5 结语

通过应用基于 SOA 架构的 SCA 规范, 使企业可以向快速响应、灵活应变、集中目标和富有弹性的方向转型, 并始终保持这些特性。也就是说, 使企业达到随需应变的状态。应用 SCA 规范的信用卡申请系统是一个组件模型, 它将应用程序的不同功能单元(称为服务)通过这些服务之间定义良好的接口和契约联系起来。本系统中的接口是采用中立的方式进行定义的, 独立于实现服务的硬件平台、操作系统和编程语言。这样, 构建在各种各样的系统中的服务可以以一种统一和通用的方式进行交互。

SCA 提供了规范的模块化设计方法, 同时它继承了 SOA 的多种优点和特性, 在构建企业应用时体现了明显的优势。运用 SCA 搭建的应用系统构架可以从根本上解决企业信息系统中存在的诸如“信息孤岛”等问题, 从而使企业内部和企业之间能够更好地实现数据共享和信息交互。

参考文献

- 1 Beisiegel M, Blohm H. Service Component Architecture Whitepaper, 2005,11:91-94.
- 2 谭永明, 苏斌. 面向服务架构体系的研究. 计算机技术与展望, 2007,17(3):132-135.
- 3 吴双跃, 康丰罗, 攀峰, 符小宝. SOA 在信用卡系统中的研究与应用. 信息前沿, 2005,20(10):16-18.
- 4 柴晓路. Web 服务架构开发互操作技术[硕士学位论文]. 北京:清华大学, 2002.
- 5 王明文, 朱心清, 等. Web 服务架构. 计算机应用研究, 2005,11(3):93-95.
- 6 李建华, 陈松桥, 马华. 面向服务架构参考模型及应用研究. 计算机工程, 2006,(20):100-102.
- 7 尧飘海. 基于 SCA 的面向服务的设计与实现. 计算机系统应用, 2008,17(8):26-29.
- 8 岑文初. 基于 SCA 规范的服务框架设计与实现. 程序员, 2008,(9):106-109.