

## • 计算机应用技术 •

## 基于 SOA 的服务构件封装技术研究

韩 丁<sup>1</sup>, 沈建京<sup>1</sup>, 万 芳<sup>1</sup>, 张 慧<sup>2</sup>

(1. 解放军信息工程大学 理学院, 河南 郑州 450001; 2. 61512 部队, 北京 100088)

**摘 要** SOA 源于早期的基于构件的分布式计算方式, 目前已经被多数企业广泛认可。在 SOA 的实施过程中, 对于系统设计者来讲, 服务的设计是一个非常重要的问题。遗留系统是一笔宝贵的财富。在研究了 SOA 理论的基础上, 根据多粒服务的 SOA 设计原则, 提出了将遗留系统封装成为服务构件。给出了**服务构件的定义和基本概念模型**。结合服务设计的理论和**服务构件封装的实践提出了服务构件设计的一般参考步骤**。并将接口技术作为服务构件封装技术的重点。最后将这种设计方法应用于具体的实践中, 取得了较好的效果。

**关键词** 面向服务架构; 遗留系统; 服务构件; 封装; 模型; 接口描述

**中图法分类号** TP39 **文献标识码** A **文章编号** 1000-7024 (2009) 07-1756-04

## Research on encapsulation of service component based on service-oriented architecture

HAN Ding<sup>1</sup>, SHEN Jian-jing<sup>1</sup>, WAN Fang<sup>1</sup>, ZHANG Hui<sup>2</sup>

(1. Institute of Science, PLA Information Engineering University, Zhengzhou 450001, China;

2. 61512 Troops, Beijing 100088, China)

**Abstract** : SOA is developed from original distributed computation based on components. Nowadays it has been wildly accepted. For designers of SOA, it is very important to design the service. Legacy system is valuable riches. How to wrap legacy system is introduced into service component. The definition and the basic concept model of service component are introduced after analyzed the principle of multi-grain service and the studying of SOA. The designing step of service component on studying the theory of service 's design and the practice of the encapsulation of service component is presented. The method has good result in practice.

**Key words** : SOA; legacy system; service component; encapsulation; model; interface description

## 0 引 言

业界提出面向服务的架构 (service oriented architecture, SOA) 作为软件体系结构的下一个发展阶段, 来帮助软件开发商解决新时期软件开发中存在的问题。**SOA 从根本上解决了“信息孤岛”问题, 它把应用和资源转换成标准的服务, 从而形成资源的共享**。据 Yankee 集团随机调查了 437 家美国企业, 有 75% 的企业已计划进行大规模的 SOA 投资<sup>[1-2]</sup>。从上面这份调查可以看出, SOA 在国外的企业中已经被广泛地接受并且逐步地使用。但是国内企业实施 SOA 架构却还方兴未艾, 国内企业对于 SOA 持观望态度, 很多企业怀疑 SOA 是在炒作概念, 放慢了进行 SOA 部署的计划。这一点与目前国内对于 SOA 缺乏深入的研究有关。目前国内的研究主要有以下几个特点:

(1) 较多的讨论 SOA 架构本身, 相对忽视 SOA 架构建模和具体实现部署的细节。诸如: “SOA 可以灵活地执行任何提交的任务, 而且做得更快、更有效率, 成本也比以往用过的其

它方法来得低廉。”这些相关讨论通常只是空中楼阁, 并未涉及企业真正需要的开发与集成步骤以及部署规划等。

(2) 较多的讨论 SOA 的各种实现技术, 相对缺少案例辅助说明, 解释比较空泛。



(3) 较多讨论 SOA 与 Web 服务的关系, 混淆了两者的概念。因此, 虽然 SOA 现在已经成为国内研究的热点, 但多在理论层面和架构层次上, 真正应用到实践的并不多。如何将 SOA 的优势与 IT 实践相结合, 仍然是有待解决的问题。遗留系统是一笔宝贵的财富。本文在深入研究 SOA 理论的基础上, 重点研究了**基于 SOA 的遗留系统服务构件封装技术**。

## 1 面向服务的架构 (SOA)

## 1.1 SOA 概述

SOA 是一个构件模型, 它将应用程序的不同功能单元 (称为服务) 通过服务之间定义良好的接口和契约联系起来。接口采用中立的方式进行定义, 它应该独立于实现服务的硬件平台、操作系统和编程语言<sup>[3]</sup>。

收稿日期: 2008-04-10; 修订日期: 2009-01-12。

基金项目: 国家自然科学基金项目 (60774041)。

作者简介: 韩丁 (1977 - ), 男, 吉林长春人, 硕士, 助理工程师, 研究方向为分布式计算、人工智能; 沈建京 (1961 - ), 男, 河北人, 博士生导师, 研究方向为分布式计算、人工智能; 万芳 (1983 - ), 女, 河南封丘人, 硕士研究生, 研究方向为分布式计算、人工智能; 张慧 (1980 - ), 女, 硕士, 工程师, 研究方向为分布式计算、地图制图。E-mail: falcon1977@126.com

SOA 采用服务请求的软件架构,从根本上改变了传统软件的开发方式。与传统的软件系统不同,SOA 只限定服务所需的信息并提出服务请求,但是不限定提供服务的模块,这样就完全可以在服务请求模块不知不觉的情况下,由不同的数据源来满足这个服务请求。另一方面,新的数据源也可以去响应其他服务请求者提出的类似请求。图 1 为 SOA 的软件架构的结构<sup>[4]</sup>。

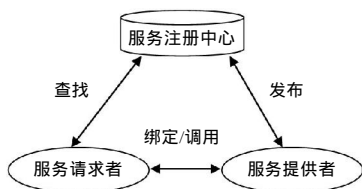


图 1 SOA 的软件架构的结构

## 1.2 多粒度的服务设计

服务的复用性、松耦合性和共享正式契约是 SOA 中服务设计的核心原则。而 SOA 区别于其它体系结构风格的特点是松耦合性。耦合代表的是服务与服务之间的关系。SOA 的初衷就是为了降低系统各个部分之间的耦合性,增强服务设计的复用性。但在 SOA 服务的设计中,服务粒度的大小在服务的复用性和松耦合性上产生了矛盾。服务粒度大,服务的松耦合性越好,但复用性较差;服务的粒度小,服务的复用性越好,但服务之间的耦合度较高。因此,为了解决这一问题 SOA 中往往采用多粒度的服务设计方法。

虽然在复杂的 SOA 环境中服务的不同类型具有不同的粒度,但归结起来可以用两种粒度实现 SOA 中的服务:细粒度服务和粗粒度服务。使用粗粒度的接口作为外部应用集成的最佳实践。服务组合和编排可以用来创建由细粒度服务组成的业务流程的粗粒度接口<sup>[5]</sup>。

多粒度是 SOA 设计中最为重要的方法,那么怎么区分粗粒度与细粒度的服务?下面给出具体的服务划分方法:

(1)粗粒度服务:一个粗粒度服务对服务使用者提供的服务在业务逻辑中是有意义和可测量价值的;它在一个完整的业务目标中与其它粗粒度服务配合和协作以实现业务过程的明确部分。

(2)细粒度服务:一个细粒度服务提供了较细粒度的操作,这些操作没有提供真正的业务价值;它与其它细粒度服务的配合和协作,通过组合封装成为粗粒度的服务以实现业务价值。

本文关注的是相对细粒度服务的封装技术,为构建粗粒度的服务提供基础。这种细粒度的服务本文称为服务构件。

## 2 基于 SOA 的服务构件封装技术

### 2.1 服务构件封装的关键问题

与面向对象技术相似,面向服务已成为独特的设计方法。但当项目经过 SOA 分析阶段,一组候选服务和服务构件摆在我们面前,我们不禁思考起来:什么是服务构件、如何进行设计、如何来实现,如图 2 所示。

以下就服务构件设计应该解决的问题,给出了服务构件的定义,提出服务构件的基本概念模型以及服务构件设计的

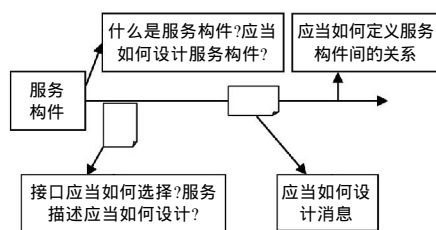


图 2 面向服务原则提出服务构件设计的问题

一般步骤。将接口作为连接面向服务架构与基于服务构件设计的关键。

### 2.2 服务构件的定义

本质上,SOA 是通过服务编排来提供业务附加值和网络价值。从服务消费者的观点,服务是通过类似于构件的实体,隐藏了其服务实现的契约接口来提供的。可以这样说,SOA 中每个服务都是高内聚,松耦合的构件结构,并且对外提供服务。下面我们给出服务构件的定义:

服务构件是一个封装的、自治的软件实体,它基于契约的方式通过其接口提供和实现了服务,并且隐藏了其实现细节,是服务组合和编排的基本单元。

### 2.3 服务构件的基本概念模型

服务是网络环境下具有自治、自描述等特征的构件,因此合理的构件描述模型同样适用于服务构件。我们借鉴 Will Tracz 提出的 3C 模型(Will 1990)来描述服务构件模型。该模型从概念(concept)、内容(content)和上下文(context)3 个方面来刻画服务构件,如图 3 所示。

(1)概念:是对服务构件做什么(What)的抽象描述,可以通过服务构件的概念了解服务构件的功能。服务构件的概念包括服务构件的接口规范和语义两方面。

(2)内容:是对概念具体实现的描述,描述服务构件如何(how)完成概念所刻画的功能。

(3)上下文:是服务构件和服务构件之间执行环境之间的关系。上下文刻画服务构件的运行环境,为服务构件的选择和修改提供指导。

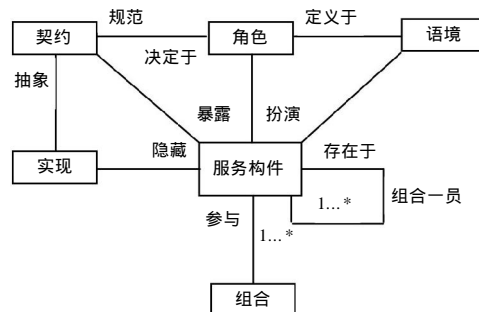


图 3 服务构件设计的基本概念模型

从上面的服务构件的基本概念模型我们可以看出,一个服务构件是一个软件实体。它是独立的、自包含的、具有清晰的目的,区别于其它服务构件而存在。一个服务构件的基本元素是它的语境、契约和实现。服务构件的本质是通过其契约实现内部和外部的分离。一个服务构件不是孤立存在的;

在一个给定的语境下扮演一个特定的角色并实现通信。一个服务构件的角色是通过契约精确定义的。契约通过服务构件的实现而实现,并且通过语境中的中立接口隐藏。在上下文语境中服务构件之间通过服务组合提供更粗粒度的服务。

(1)语境:各个服务构件的单独存在并无太大价值,除非这些服务构件能与其它服务构件一起被组合或其它应用所使用。因此,基本的服务构件虽然是独立的、自包含的,但它并不是孤立存在的。无论在服务组合或业务流程编排中,服务构件都扮演者特定的角色。并且实现与其它服务构件或服务之间的通信。

(2)实现:任何提供 Web 服务支持的执行环境都可以作为服务实现,这里的执行环境通常是某个软件系统或编程语言。所以,服务构件的实现可以有多种方式。比如新开发服务的实现,包装的传统应用的实现。

(3)契约:服务契约是服务的共同元数据的表现。它标准化规则的表达,以及需要由任何想要与服务交互的请求者所要履行的条件。

## 2.4 服务构件的接口选择

从技术角度看,服务构件相对于组件和对象是粗粒度的、可重用的 IT 资产,良好的接口定义令服务的外部访问接口与内部技术实现相分离。对于服务构件的封装,最重要的就是具有符合 SOA 要求的标准接口。SOA 只对服务之间交互机制的标准进行了描述,并没有规定具体的实现方式,因此对于标准接口的实现先后出现了多种方式,如表 1 所示。

表 1 标准接口实现方式的比较

方式	数据格式	端点命名	支撑协议	接口
CORBA	CDR	IOR	互联 ORB 协议	IDL
DCOM	NDR	对象参考	DCOM	COM 继承
RMI	Java 连续对象格式	超文本链接	JRMP	Java 接口
Web Service	XML	超文本链接	SOAP	WSDL

CORBA(common object request broker architecture)即公用对象请求代理程序体系结构,使用它可以实现程序之间的跨平台通讯<sup>[6-7]</sup>。CORBA 定义了一系列标准和协议来实现异构环境下的互操作性,包括:接口定义语言(IDL)、对象请求代理(ORB)以及 ORB 之间的互操作协议 IIOP。这些标准都是 CORBA 专用的。

DCOM 即分布式组件对象模型,是从 COM 发展而来,可以实现异构系统的远程调用,但是使用 DCOM 一般不能跨越防火墙<sup>[8]</sup>。

RMI 即远程方法调用,是一个分布式对象模式,是基于 Java 中的类和对象实现的远程调用机制。RMI 的思路基本上与 CORBA 相同,是 CORBA 技术在 Java 语言中实现的一个特例。

Web Service 它使用 XML 方式来描述所有数据,真正与平台无关,因此可以轻松的穿越防火墙并实现跨系统、跨语言交换数据。同时 Web Service 中所有的公共协议完全使用开放的标准协议进行描述、传输和交换,通用性好<sup>[9-10]</sup>。

基于以上分析和比较,本文决定采用 Web Service 的方式,服务构件的封装采用标准的 Web Service 接口。

## 2.5 服务构件的接口描述

每种服务(粗粒度服务和细粒度服务)都有一个良好定义

的、正式的接口,称作服务契约。服务契约主要作用如下:

(1)明确定义了服务的功能。

(2)实现了服务的外部访问接口与服务的技术实现的明确分离<sup>[11]</sup>。

服务契约几乎定义了所有的 SOA 主要部分,良好定义的服务契约比服务实现更有价值<sup>[12]</sup>。服务契约包括服务的功能和非功能描述。功能描述主要描述了服务的接口和消息规约。非功能描述包括服务的安全问题和层次协议等问题。服务契约可以作为专门的方向进行研究。本文关注的主要是服务的功能描述。对于非功能描述我们不作讨论。下面我们采用扩展巴科斯范式(extended Backus-Naur form,EBNF)对服务构件的功能描述定义如下:

服务构件描述::=<接口规约,消息规约>

接口规约::=<接口名,操作>

操作::=<操作名,输入参数,输出参数>

输入参数::=<参数名,类型,描述>

输出参数::=<参考名,类型,描述>

消息规约::=<消息名,消息描述,消息属性>

消息属性::=<属性名,类型,属性描述>

通过上面定义的 Web 服务构件契约的扩展巴科斯范式,我们定义了服务构件接口描述的主要契约元素。这些主要元素是服务构件描述所必需的,是服务构件基本功能的外部展现。

## 2.6 服务构件的封装方法

Web 服务是一种基于现有的被广泛接受和成熟的 Internet 技术的分布式应用程序技术框架,由 W3C 和 Internet 工程任务组(IETF)等组织设计<sup>[13]</sup>。目前 SOA 的相关的开发技术以 Web 服务为主,但我们一定要认清 SOA 并不等同于 Web 服务。简单将遗留系统的应用封装成为 Web 服务不一定符合 SOA 的一般原则,因此也不能称为 SOA 中的服务。

目前,面向服务的分析与设计(SOAD)作为 SOA 创建服务的主要方法。它主要包括服务发现、服务规约和服务实现。在服务发现上主要包括自上而下(领域分解)方式、自下而上(已有资产分析)方式和中间对齐(业务目标建模)方式<sup>[14]</sup>。服务构件作为 SOA 中服务组合的基本单元,在服务发现阶段应采用自上而下(领域分解)方式和中间对齐(业务目标建模)方式,从业务着手分解成为服务候选列表。下面我们结合理论的研究和服务构件的封装实践,介绍一下在服务规约阶段较为合理的服务构件设计一般参考步骤:

### 步骤 1 审视现存的服务

封装服务的第一步是确认它是否实际上正好是必须的。如果已存在的服务构件已经提供了该服务所能提供的部分或者全部功能,那么该服务所实现的操作功能可以用已存在服务构件来实现。

### 步骤 2 定义消息结构

服务之间的通信是通过消息的传递完成的。因此,通过正式定义服务所需要处理的消息而开始服务接口设计是十分有用的。为此,我们需要在 WSDL types 域中形式化地定义消息结构。

### 步骤 3 确认语境

当进行面向服务分析时,很自然地就集中到直接的业务

需求。其结果是,在这个阶段产生的服务构件候选经常不考虑现存服务构件所建立的语境。因此,重新评估由服务候选所提议的操作分组,这与现存服务构件设计相比将十分重要。在重新评估服务语境的基础上,你会发现哪些操作是遗留系统具有的;哪些操作需要重新创建。通过确认语境你可以从新调整操作分组。

#### 步骤4 派生抽象服务接口

分析服务操作候选,并且遵循下述步骤定义初始的服务构件接口:

(1)使用服务构件候选作为主要输入,确保表示操作候选的逻辑划分的粒度能适当地通用和可复用。

(2)记录处理每个操作候选的文档输入和输出值。

(3)通过增加 portType(或 interface)域(连同它的子结构 operation 结构)和必要的 message 结构来完成抽象服务定义。

#### 步骤5 应用面向服务原则

面向服务原则是服务设计的一般原则。因此这一步骤规范和指导服务构件的设计,重新审视服务构件设计中自治性,松耦合和契约等原则。

#### 步骤6 标准化和简化服务接口

即使服务构件的角色和用途与其它类型的服务(组合服务)不同,但重要的是以相同的基础方式来设计它们。要实现这些,需要确保服务构件的 WSDL 定义基于别人使用的相同的标准和管理。以下给出几点服务构件设计的指导:

(1)使用命名标准。命名标准定义和应用用于:服务端点的名称、服务操作的名称和消息值。

(2)谨慎使用命名空间。尽量使用基于规范元素的通用命名空间。

(3)尽量将服务操作设计为天生可扩展的。

### 3 基于 SOA 服务构件封装的应用

我校电子系承担了包括计算机基础知识、电路实验和系统模拟仿真等繁重的教学实践任务。因此如何高效和合理的管理学生的上机实习和电路实验是困扰已久的问题。经过了多方面的综合研究,新系统决定基于 SOA 模式对学生的实践进行管理和访问控制。根据 SOA 的架构模式本系统分为四层,客户端采用多种模式,包括 B/S、C/S、嵌入式客户端。对各个实验室来说根据不同的场景,选择不同的客户端,调用服务完成不同的操作即可,如图4所示。

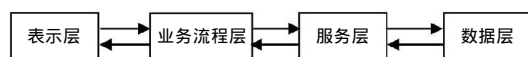


图4 开放式实验管理系统的四层架构

应用本文的服务构件封装技术主要完成了服务层中服务构件的封装工作。将遗留系统中的指纹采集和认证模块封装成为指纹采集服务构件和指纹认证服务构件。此外服务层还包括通过与其它服务构件组合而成的粗粒度的认证服务。具体的业务逻辑都是通过服务的方式发布出去,用户通过不同的客户端调用服务,通过业务逻辑的编排即可实现所需的访问控制功能。

本系统采用SOA模式和多粒度服务的设计具有以下优点:

(1)本系统中采用多粒度的服务设计。通过服务构件组合成为粗粒的服务实现了SOA的复用性,通过粗粒度服务的发布降低了服务之间的耦合度。

(2)服务可在不同平台、不同网络之间进行操作。客户端可以采用J2EE、.NET等多种方式调用服务并且根据需求创建应用。

(3)四层结构在逻辑上各自独立、单独实现,便于对软硬件资源进行同一调配管理。

### 4 结束语

本文研究的将遗留系统封装成为服务构件,符合SOA的面向服务原则,也符合SOA中多粒度服务的服设计思想。文中提出的服务构件设计的一般参考步骤对具体实践也有一定的指导意义。将服务构件的封装技术运用到开放式实验管理系统中服务的创建,取得了较好的效果,体现了SOA的价值。因此,本论文的设计思想有一定的参考和利用价值。但对于服务构件的封装依然有如下待研究的问题:首先,目前服务粒度的划分往往根据实际应用而言,缺乏统一的划分方法。其次,服务契约中非功能描述的研究,主要是安全性和可靠性的问题。

### 参考文献:

- [1] 王兵.基于面向服务架构的应用系统开发与集成研究[D].成都:四川大学,2005.
- [2] 张志雄.基于SOA的移动电子商务系统的研究[D].西安:西北工业大学,2005.
- [3] Steve Wilkes.SOA blueprints specification draft v0.4[EB/OL]. The Middleware Company Research Team,2004.
- [4] IBM DW. 面向服务的体系结构详述 [EB/OL]. <http://tech.ccidnet.com/pub/article/c322-a206969-p2.html>,2005.
- [5] 朱振杰.SOA的关键技术的研究与应用实现[M].成都:电子科技大学,2006.
- [6] Heather Williamson. XML: The complete reference [M]. New York: McGraw Hill, 2002.
- [7] 朱其亮,郑斌.CORBA 原理及应用[M].北京:北京邮电大学出版社,2001.
- [8] Thuan L Thai. DCOM 入门[M].北京:中国电力出版社,2001.
- [9] 齐勇,马莉,齐向明,等.基于CORBA业务组件的领域框架研究[J].小型微型计算机系统,2001,22(10):1213-1215.
- [10] Scott Seely. SOAP:XML 跨平台 Web Service 开发技术[M].北京:机械工业出版社,2002.
- [11] Eric Newcomer Greg Lomow.Understanding SOA with web service[M].北京:电子工业出版社,2006.
- [12] Thomas Erl.SOA 概念、技术与设计[M].北京:机械工业出版社,2007.
- [13] Ferguson D F.Web services architecture: direction and position paper [OL]. <http://www.w3.org/2001/03/WSWS-popa/paper44>, 2001.
- [14] 毛新生.SOA 原理、方法、实践[M].北京:电子工业出版社,2007.