

文章编号: 1005-0523(2007)02-0074-04

# 基于 Web Service 的面向服务架构(SOA)的研究

李培松, 刘觉夫

(华东交通大学 信息工程学院, 江西 南昌 330013)

**摘要:**介绍了 Web Service 的关键技术, 分析了面向服务的体系结构和特点. 提出了一种基于 Web Service 的面向服务架构的实现方案, 并结合实例对企业应用集成的过程进行了探讨.

**关键词:** Web Service; 面向服务架构; 企业应用集成

**中图分类号:** TP393

**文献标识码:** A

## 0 引言

Web Service 是一个独立的, 基于标准的应用, 是基于互联网标准和 XML 技术建立起来的, 用于描述、发布到网络供其他应用程序调用的一种技术. Web Service 是分布式系统的一个新的发展趋势. Web Service 技术的日臻成熟使得面向服务的架构(SOA)思想得到了很好的应用.

SOA<sup>[1]</sup>并不是新鲜的事物, 但它却是更传统的面向对象模型的替代模型. 虽然 SOA 的系统并不排除使用面向对象的设计来构建单个服务, 但其整体设计确实是面向服务的. SOA 系统原型的一个典型例子是通过对象请求代理结构(Common Object Request Broker Architecture, CORBA). 随着网络技术的发展, 今天的 SOA 是基于已经广泛接受的 Web 服务标准, 从而提供了在每个不同厂商解决方案的相互性. SOA 被誉为下一代 Web 服务的基础框架, 目前已经成为计算机信息领域的一个新的发展方向<sup>[2]</sup>.

## 1 Web Service 技术关键

### 1.1 XML

Web Service 的所有协议都是建立在可扩展标记语言 XML 的基础上. 所以, XML 是 Web Service 的基石.

XML 是 W3C 组织于 1998 年 2 月发布的标准, W3C 组织制定 XML 标准的初衷是: 定义一种互联网上交换数据的标准. XML 有三个关键要素: Schema (模式), XSL (eXtensible Stylesheet Language 可扩展式样语言) 和 XLL (eXtensible Link Language 可扩展连接语言).

Schema 规定了 XML 文件的逻辑结构, 定义了 XML 文件中的元素, 元素的属性以及元素和元素的属性之间的关系, 它可以帮助 XML 的分析程序校验 XML 文件标记的合法性.

XSL 用于将 XML 数据翻译为 HTML 风格或其他风格. XSL 提供了一种叠式页面 CSS 的功能, 使开发者构造出具有表达层次结构的 Web 页面来, 以有别于 XML 的数据结构. XSL 也能和 HTML 一起构造叠式页面, XSL 可以解释数量不限的标记, 使得 Web 版面更丰富多彩.

XLL 是 XML 的连接语言, 它与 HTML 的连接相似, 但功能更强大. XLL 支持可扩展的连接和多方向的连接. 它打破了 HTML 只支持超级文本概念下最简单的连接限制, 能支持独立于地址的域名, 双向联络, 环路, 多个源的集合连接等. XLL 连接可不受文档制约, 完全按用户要求来咬定和管理.

### 1.2 SOAP

SOAP (Simple Object Access Protocol, 简单对象访

问协议)是一个基于 HTTP 和 XML 的请求/响应 RPC 协议,把 XML 的使用代码化为请求和响应参数码模式,并用 HTTP 作传输,采用几行代码和一个 XML 解析器,HTTP 服务器(如 MS 的 IIS 或 Apache)立刻成为了 SOAP 的 ORB(SOAP 对象请求代理)<sup>[3]</sup>.

SOAP 消息的具体格式:每一个 SOAP 消息都有一个根元素 Envelope(信封),它包含 header 和 body 两部分. Header 包含了上下文数据,可以不存在,而 body 包含了具体的消息,必须存在. 下面是一个通过 HTTP 协议请求 Borland 公司股票价格的 SOAP 消息. POST 网络地址 HTTP/1.1

```
Host :www . stock . com
Content -Type :text /xml ; charset = "utf -8"
Content -Length :SOAP 消息长度
SOAPAction : "urn : stock - quote - services " w
<SOAP - ENV : Envelope xmlns : SOAP - ENV = "命名空间" SOAP - ENV : encodingStyle = "命名空间" >
<SOAP - ENV : Body >
<m : GetLastTradePrice xmlns : m = "Some - URI" >
<symbol > Borland 公司股票代码 </ symbol >
</ m : GetLastTradePrice >
</ SOAP - ENV : Body >
</ SOAP - ENV : Envelope >
```

### 1.3 WSDL

WSDL(Web Service Description Language, web 服务描述语言)是为分布式系统提供自动执行应用程序通信中所涉及细节的一种描述性文档,它的基本构想是将网络服务描述为能够进行消息转换的通信端点的集合,进而对这些服务在通信中所涉及细节进行结构化描述,调用者可以通过接口界面了解通信所需知道的数据类型,消息结构,传输协议等信息,从而调用相关服务, WSDL 的框架结构有 XML Schema 定义,并由于采用 XML,能使 WSDL 对各种语言实现的接口进行描述,对已有系统的集成有独特的好处.

### 1.4 UDDI

UDDI(Universal Description, Discovery and Integration, 统一描述、发现和集成)是一套面向 Web 服务的信息注册中心的实现标准和规范. 创建 UDDI 注册中心的目的是实现 Web 服务的发布和实现. 利用 UDDI 规范在 Web 上建立发布服务,这些发现服务为所有请求者提供了一致的接口,使得已经发布的 Web 服务能通过编程被需要的请求发现. UDDI 规范文本定义了 UDDI 操作入口站点(UDDI Operator Site)

能够支持的 API 接口和 API 中用 XML 描述的数据结构的具体规定<sup>[4]</sup>.

## 2 面向服务的体系架构及其特点

面向服务的体系架构(Service - Oriented Architecture, SOA)是一种基于互联网的信息集成体系框架,SOA 采用面向服务的软件封装技术,它以服务接口(service interface)和服务实现(service implementation)的方式呈现,它的三个基本要素是服务描述、服务发现和服务调用<sup>[5]</sup>. 从技术角度讲,它是一个组件模型,以 XML 技术为基础,通过使用 WSDL 协议基于 XML 格式的 Web Service 描述语言来描述接口. SOA 不仅仅是一个软件开发框架而且还是一个业务开发框架,它能够将不同类别、不同平台的服务结合在一起,动态地、实时地更新维护一个跨区域的多功能的应用实体.

面向服务的体系结构中主要有三种角色,它们分别是服务提供者,服务调用者和服务注册者. 其中服务提供者负责服务功能的具体实现,并通过服务操作将所提供的服务发布到服务注册器,接收到服务调用者的服务请求时,执行所请求的服务. 服务调用者则是服务执行的发起者,首先需要到服务注册器重查找符合条件的服务,然后根据服务信息进行服务绑定/调用,以获得需要的功能. 面向服务注册器则用来提供服务提供者注册服务,提供对服务的分类和查找功能,以便服务调用者发现服务.(见图 2)

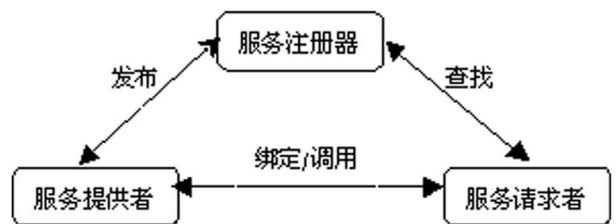


图 2 SOA 服务模型

## 3 基于 Web Service 的 SOA 企业应用集成

某市食品有限公司在进行企业信息系统集成之前,已存在 3 个 CBM(Customer Relationship Management)应用子系统,分别是业务管理子系统,生产经营管理子系统和财务管理子系统. 3 个系统是在不同时期有不同的软件开发商分别开发的,3 个系统不仅在开发环境上存在差异且都具有自己独立的数

据库。呼叫中心是另一个正在开发的 CRM 子系统, 公司根据业务需求要以呼叫中心系统为中枢能够将这 4 个 CRM 应用系统集成起来, 实现 CRM 整体信息的共享。为了使用一致数据表示实现企业信息的交流与共享, 依照 SOA 模型, 以 Web Service 为实现手段进行企业信息系统集成, 其模型如图 3 所示。

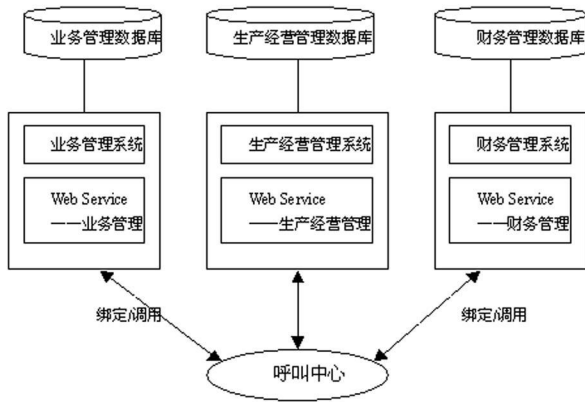


图 3 CRM 系统集成模型

业务管理子系统, 生产经营管理子系统和财务管理子系统分别提供自己的 Web Service 接口, 供呼叫中心程序调用, 达到共享服务的目的, 服务调用者只需通用的 Web Service 接口, 无需考虑 Web Service 的内部实现机制, 操作平台以及开发语言等实现细节。由于 Web Service 的松耦合性, 当某个 CRM 子系统需要业务更改时, 只要保持其对外提供的服务接口不变, 将不会影响其他子系统。

Web Service 的开发工具采用了 .NET .NET 的底层采用通用语言运行环境, 类似于 Java 的虚拟机, 使用程序具备跨语言、跨平台的能力。

下面以财务查询 Web Service 为例, 来说明采用 WSDL 定义 Web Service 的方法。WSDL 文件主要定义了服务的名称、消息数据类型、消息结构、服务操作和服务的端口绑定以及服务所采用的网络传输协议等信息。

```
<? xml version="1.0" encoding="utf-8"? >
```

Types 节是一个数据类型的容器, 包含了所有在消息定义中需要的 XML 元素的类型定义。这里定义了名 FinancialAffairsQuery 的数据类型, 供后面的 FinancialAffairsQuerySoapIn 消息使用。

```
<types>
<s:schema elementFormDefault="qualified"
targetNamespace="http://tempur.org"/>
<s:import namespace="
http://www.w3c.org/2006/XMLSchema"/>
<s:element name="FinancialAffairsQuery">
```

```
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name
="strCardID" type="s:string"/>
<s:element minOccurs="0" maxOccurs="1" name
="strPW" type="s:string"/>
</s:sequence>
</s:complexType>
</s:element>
</types>
```

message 标记定义消息的结构, 这里定义了一个名为 FinancialAffairsQuerySoapIn 的消息。

```
<message name="FinancialAffairsQuerySoapIn">
<part name="parameters" element="s0:
FinancialAffairsQuerySoapIn"/>
</message>
```

PortType 具体定义了财务查询服务访问入口的类型, 定义了一种传入消息模式和一种传出消息模式。

```
<portType name="FinancialAffairs">
<operation name="FinancialAffairsQuery">
<documentation>财务查询</documentation>
<input message="s0:
FinancialAffairsQuerySoapIn"/>
<output message="s0:
FinancialAffairsQuerySoapOut"/>
</operation>
</portType>
```

Binding 结构定义了上述 PortType 与某种具体的网络传输协议或消息传输协议相绑定。这里定义了与 SOAP 消息传输协议的绑定。

```
<binding name="FinancialSoap" type="s0:
FinancialAffairs">
传送的 SOAP 消息
<soap:binding transport=
http://schemas.xmlsoap.org/soap/http style=
"document"/>
<operation name="FinancialAffairsQuery">
<soap:operation soapAction=
"http://tempur.org/FinancialAffairsQuery"
style="document"/>
<operation name="FinancialAffairsQuery(i)">
<soap:operation soapAction=
http://tempur.org/FinancialAffairsQuery
```

```

style="document"/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>

```

上述3部分属于 Web Service 的抽象接口描述部分,与 Web Service 的具体部署信息无关,可以重复使用.这样就可以把业务管理子系统,生产经营管理子系统和财务管理子系统以面向服务的结构集成起来.

#### 4 结语

本文介绍了 Web Service 和 SOA 的关系,就如何

应用 Web Service 技术构建面向服务的信息集成过程进行了阐述.这种基于 Web Service 的方式可以满足信息集成的要求,体现了松散耦合的特点.面向服务的架构可以解决原来企业应用集成的问题,为企业的业务流程的变更提供了很好的灵活性,为企业未来业务的发展提供了良好的环境.

#### 参考文献:

- [1] Mark Endrei, Jenny Ang, Arsanjani, et al. Patterns: Service Oriented Architecture and Web Service. IBM International Technical Support Organization, April, 2004.
- [2] 王颖, 吴荣泉. 一个面向服务的 EAI 框架[J]. 计算机工程. 2006, 32(1): 3-5.
- [3] 朱亚光, 刘峰, 杨芳南. Web Service/CORBA 网关系统的研究与实现[J]. 微计算机信息. 2006. (5): 85-86.
- [4] 应宏, 鄢沛. 基于 Web Service 的跨企业应用模型[J]. 计算机工程. 2005, (1): 140-143.
- [5] 徐罡, 黄涛, 刘绍华. 分布应用集成核心技术研究综述[J]. 计算机学报. 2005, 28(4): 430-432.

## Study on Service Oriented Architecture Based on Web Service

LI Pei-song, LIU Jue-fu

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

**Abstract:** Introduce the critical technology of Web Service, analyse the architecture and specialty of Service Oriented Architecture. It puts forward a scheme of Service Oriented Architecture based on Web Service and discusses the process of enterprise application integration with an example.

**Key words:** web service; service oriented architecture; enterprise application integration