

# 基于 Web Service 的工作流系统实现研究

谭伟 范玉顺

(清华大学自动化系, 北京 100084)

**摘要：**工作流是企业经营过程的计算机实现，是支持经营过程重组和自动化的一项关键技术，而分布式工作流已经成为当前工作流技术发展的主流方向之一。Web Service 技术的出现使异构环境下分布式工作流系统的实现成为可能，并弥补了传统的分布式处理技术的不足。本文提出了一种基于 Web Service 的工作流管理系统的体系结构，描述了各个模块的功能和实现方法。在此基础上提出了基于 Web Service 实现工作流管理的方法，给出了工作流系统 Web Service 接口的实现的示例。

**关键词：**工作流，分布式技术，Web Service，SOAP

## 1. 问题的提出

### 1.1. 工作流技术的发展现状

工作流是一种反映业务流程的模型[1]，是为了在先进计算机环境支持下实现经营过程集成和经营过程自动化而建立的可以由软件系统执行的业务模型。工作流管理系统则是实现这种技术的软件系统。

由于工作流技术要解决企业内或者企业间不同信息和组织环境下的业务的集成和自动化的问题，那么它必然要以分布式的形式出现。这里有两层含义，一是说工作流引擎所面对的客户端是分布的，而且客户端的形式可能是 web，桌面应用程序，甚至是 PDA 和手机用户；另一层含义是说多个工作流引擎之间在地域上是分布的，实现的机制和所在的系统平台可能都有很大的差别。这就需要引入分布式技术来实现工作流管理系统和企业各个业务应用之间的集成。

### 1.2. 分布式对象技术的发展现状

随着软件系统的日益复杂和应用领域的扩大，企业的 IT 部门面临的问题也越来越多。如何跨越异构环境实现软件系统之间的互操作是一个急需解决的问题。在这种背景下，基于面向对象的思想，分布式对象技术应运而生。

分布式对象技术的目的是在分布式异构环境下建立应用系统框架和对象构件，基于系统提供的一些服务，这些对象可以跨越不同的软硬件平台进行互操作。通过将可重用的对象进行组合，可以灵活地构建各种软件应用。目前主要的分布式技术标准主要有 Microsoft 的 DCOM/COM+ 标准，OMG 的 CORBA 标准和 SUN 的 Java RMI 标准。

CORBA 是由 OMG 组织提出的一种公共对象请求代理程序体系结构。由对象请求代理 ORB、对象服务、域接口和应用接口几个部分组成。CORBA 的思想就是作为软件系统的总线，为客户提供对象请求代理 (ORB) 服务。ORB 提供了一种机制，使得用语

可以透明的调用对象提供的方法。CORBA 中采用接口定义语言 IDL 实现了接口定义和接口实现的分离。

DCOM/COM+是微软基于组件-对象模型 (COM) 技术提出的对象分布式标准，它是基于二进制代码的互操作标准，但是这种互操作仅限于 windows 平台。Java RMI (Remote Method Invocation) 是 Java 环境下的分布式对象标准，它内嵌在 Java SDK 中，使得一个 Java 对象能够通过网络访问到另一个 Java 对象。

以上的各种技术都使得应用程序之间的通信成为可能，但是它们的缺点也是显而易见的，这几种分布式组件对象都要求调用方和被调用方的开发使用相同的体系结构。DCOM/COM+只能用于 windows 平台，Java RMI 只能用于 Java 开发的程序之间的通信，而 CORBA 要求每个节点都使用 ORB。

## 2. Web Service 概述

前已述及，已有的几种分布式计算模型并不能很好地解决企业业务集成的问题。而 Web Service 的出现则为人们解决这个问题提供了一个新的途径。Web Service 是基于 HTTP 协议提供的一组操作，这些操作使用标准的 XML 消息传递机制。[2] Web 服务使用一种成为 WSDL 的 XML 语言进行描述，这种描述包括了与该服务交互所需的所有细节。Web 服务通过一种成为 SOAP 的协议进行访问，这种接口使用 xml 作为数据传输格式，使用 HTTP 作为网络通讯协议。这种开放式的接口使得基于 Web Service 的软件系统具有松散耦合，跨平台和语言实现的特性。

Web Service 体系结构基于以下三者的交互：服务的提供者，服务注册中心和服 务请求者。它们之间的交互包括服务的发布，描述和消息传递等。Web Service 的服 务提供者 and 使用者之间进行远程通信的协议栈如图 1 所示。

发现	UDDI	Web Service
描述	WSDL	
路由	WS-Routing, WS-Referral	
消息	SOAP	
传输	HTTP,FTP,SMTP	
网络	IPv4, IPv6	

图 1 Web Service 协议栈

从图中可以看到，Web Service 的整个协议栈由服务的发现、描述、消息格式、编 码、传输和网络这几层组成，下面对每个协议层进行解释。

发现层：客户端如果要访问网络上公开的 Web Service，首要的任务是通过一种方 式来解析远程服务的位置。这个过程就称为服务的“发现” (discovery)。在 Web Service 的标准中使用统一描述、发现和集成 (UDDI, Universal Description,

Discovery and Integration) 规范定义 Web Service 的发布与发现的方法。[3]

描述层：一旦找到了某个 Web Service 的端点，客户端就需要了解它的接口以便与之交互。Web Service 使用 WSDL[4] (Web Services Description Language) 来描述服务界面。WSDL 定义了一套基于 XML 的语法，将 Web Service 定义为服务访问点或者端口的集合。同时 WSDL 分离了接口的定义层和实现层，使得定义层可以被不同的实现所复用。

路由层：该层定义了消息路由的协议。其中 WS-Routing 用来说明消息的传输路径，WS-Referral 用来为转发消息的节点配置关于消息路径的协议。

消息层：为了交换数据，客户端和服务器必须在消息编码和格式设置上达成协议。在 Web Service 中这通过 SOAP[5] (Simple Object Access Protocol) 来实现。SOAP 的功能是使用特定的编码规则来表示特定的调用语义。SOAP 由 envelop, encoding rules, RPC presentation 和 binding 四部分组成。其中 envelop 定义消息的整体表示框架，encoding rules 定义数据的交换机制，RPC presentation 定义表示远程过程调用和相应的约定，binding 则定义 SOAP 和底层传输协议 (HTTP, SMTP 等) 的绑定信息。

传输和网络层：Web Service 消息载荷可以使用 HTTP (这是最常用的情况)，SMTP, FTP 等。网络层则基于 Ipv4 或者 Ipv6。

### 3. 基于 Web Service 的工作流系统

#### 3.1. 系统的优点

Web Service 技术和工作流技术是两个相对独立发展的 IT 技术，但是随着 IT 技术的发展，企业 IT 系统的范围开始向更广的领域延伸，企业各部门之间，各个企业之间的业务系统的交互将变得越来越频繁。而传统的工作流系统往往基于特定的操作系统平台和网络协议，各工作流厂商之间往往使用不同的 API，数据结构也不一致。这就使得跨企业的、异构的、分布式的工作流系统实现变得十分困难。过去的一些有限的集成也都采用紧耦合的方式，使得系统集成的柔性欠佳。

Web Service 的出现可以解决这个问题。首先，Web Service 使用已成为工业标准的 HTTP 协议和 XML 作为通信手段，减少了异构环境之间对私有的适配器和连接器的需要。其次，系统各部分之间是一种松散的耦合关系而又具有高度的集成能力。另外，Web Service 协议栈中的安全协议保障了交互的安全性和交互信息的完整性。[5]

这些特性使得 Web 服务可以完全屏蔽各种软、硬件平台的差异，很好地支持异构工作流系统之间的交互。

### 3.2. 系统结构

参照 WfMC 的工作流系统的参考模型[6]，我们给出了一种基于 Web Service 的工作流系统的体系结构（图 2）。下面结合各个模块来描述基于 Web Service 的工作流管理系统的运作过程。

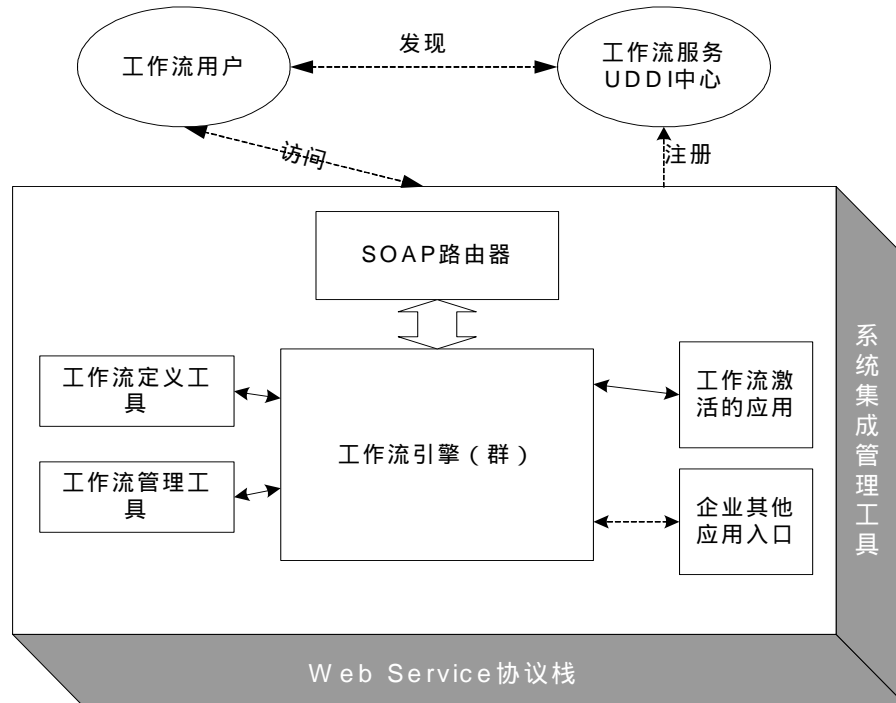


图 2 基于 Web Service 的工作流系统的体系结构

工作流用户首先通过工作流服务 UDDI 定位工作流 web 服务的位置，并获得有关工作流服务的 WSDL 描述，然后通过 SOAP 协议向工作流服务系统发出服务请求。

工作流服务系统中的 SOAP 路由器接收用户的调用请求。并根据一定的设置（路由算法、负载均衡算法等）将该请求发送到工作流引擎群的某一个工作流引擎中。

工作流引擎接受该请求，引用工作流模型数据，调用一定的外部应用程序或者服务，完成任务后将结果送交 SOAP 路由器返回给工作流用户。其中工作流模型数据应该使用 WfMC 提供的基于 XML 的模型描述语言 XPDL[7]表示，外部应用也可以以 Web Service 的形式出现。

系统集成管理工具负责在系统构建时将各个模块的接口封装成 Web Service 的形式构成系统整体；在工作流系统运行时负责维护工作流引擎，包括其他附属模块的正常运行。另外企业其他应用系统（ERP，CRM，CAD 等）也可以被包装成 Web Service 来与工作流系统进行交互。

Web Service 相关协议栈，是系统底层通讯的基础。

### 3.3. 例子

工作流系统的核心是工作流机，而工作流机对外则表现为一组接口。下面我们结合一个简单的例子来说明 Web Service 如何实现工作流机的接口函数调用。接口

调用的形式由 WfMC 给出[8]。

接口函数：创建工作流过程实例

```

WmCreateProcessInstance (in WmTPSessionHandle psession_handle,
                        in WmTPProcDefID pproc_def_id,
                        in WmTPText pproc_inst_name,
                        out WmTPProcInstID pproc_inst_id)
    
```

图 3 给出了用 Web Service 实现这个调用的 WSDL 文件的部分内容。因为该 WSDL 文件较大，我们在此只给出一些重要的部分。

```

<types>
  <xs:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
    <xs:element name="WmCreateProcessInstance">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="psession_handle" type="s:string" />
          <xs:element minOccurs="1" maxOccurs="1" name="pproc_def_id" type="s:long" />
          <xs:element minOccurs="0" maxOccurs="1" name="pproc_inst_name" type="s:string" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="WmCreateProcessInstanceResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="1" maxOccurs="1" name="WmCreateProcessInstanceResult" type="s:long" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="long" type="s:long" />
  </xs:schema>
</types>
<message name="WmCreateProcessInstanceSoapIn">
  <part name="parameters" element="s0:WmCreateProcessInstance" />
</message>
<message name="WmCreateProcessInstanceSoapOut">
  <part name="parameters" element="s0:WmCreateProcessInstanceResponse" />
</message>
<message name="WmCreateProcessInstanceHttpGetIn">
  <part name="psession_handle" type="s:string" />
  <part name="pproc_def_id" type="s:string" />
  <part name="pproc_inst_name" type="s:string" />
</message>
<message name="WmCreateProcessInstanceHttpGetOut">
  <part name="Body" element="s0:long" />
</message>
    
```

图 3 一个工作流 Web Service 的 WSDL (部分)

图 4 给出了 SOAP 的请求和响应的 HTTP 载荷。

```

POST /WebService/Service1.asmx HTTP/1.1
Host: 192.168.1.133
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/WmCreateProcessInstance"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <WmCreateProcessInstance xmlns="http://tempuri.org/">
      <psession_handle>string</psession_handle>
      <pproc_def_id>long</pproc_def_id>
      <pproc_inst_name>string</pproc_inst_name>
    </WmCreateProcessInstance>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <WmCreateProcessInstanceResponse xmlns="http://tempuri.org/">
      <WmCreateProcessInstanceResult>long</WmCreateProcessInstanceResult>
    </WmCreateProcessInstanceResponse>
  </soap:Body>
</soap:Envelope>
    
```

图 4 一个工作流 Web Service 的 SOAP 请求和相应

#### 4. 结论

本文针对企业 workflow 系统应用发展的需要，提出了基于 Web Service 的 workflow 系统的实现机制，这种机制能够保证 workflow 系统在异构的环境下实现业务的集成和业务自动化。同时 workflow 技术和 Web Service 技术都在不断的发展，关于将两者相结合的有意义的研究课题包括研究通过 Web Service 集成 workflow 系统[9]，以及通过 workflow 系统创建新的 Web Service[10]等。可以断定，Web Service 给 workflow 系统带来了更强的集成性和适应性，而 workflow 为 Web Service 提供了一个广阔的应用场合。

### 参 考 文 献

- [1] 范玉顺，工作流管理技术基础[M]，北京：清华大学出版社，2002：31~33
- [2] Heather Kreger，Web Services Conceptual Architecture(WSCA 1.0) Part I,  
[http://www-900.ibm.com/developerWorks/cn/webservices/wswsca/part1/index\\_eng.shtml](http://www-900.ibm.com/developerWorks/cn/webservices/wswsca/part1/index_eng.shtml)，2001
- [3] <http://www.uddi.org/>
- [4] <http://www.w3.org/TR/2003/WD-wsdl12-20030303/>
- [5] <http://www.w3.org/TR/SOAP/>
- [6] WfMC，The Workflow Reference Model，WfMC-TC-1003,1995
- [7] WfMC，Workflow Process Definition Interface—XML Process Definition Language(XPDL)，WfMC-TC-1025，2002
- [8] WfMC，Workflow Client Application (Interface 2) Application Programming Interface (WAPI) Specification，WfMC-TC-1009，1997
- [9] Günter Preuner，Michael Schrefl，Integration of Web Services into Workflows through a Multi-Level Schema Architecture，Proceedings of the 4th IEEE Int'l Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS 2002)，2002
- [10] Dinesh Ganesarajah1，Emil Lupu，Workflow-based composition of web-services: a business model or a programming paradigm? Proceedings of the Sixth International ENTERPRISE DISTRIBUTED OBJECT COMPUTING Conference (EDOC '02)，2002