第 32 卷　第 18 期　　计　算　机　工　程　　2006 年 9 月
Vol.32　№ 18　　Computer Engineering　　September 2006
·开发研究与设计技术·　文章编号：1000—3428(2006)18—0261—03　文献标识码：A　中图分类号：TP311

基于 Web 服务的跨政府工作流整合与实现

汪恒杰[1]，张　嵩[2]
(1. 宝信软件股份有限公司商务智能软件事业部，上海 201203；2. 复旦大学管理学院，上海 200433)

摘　要：在 Web 服务 BPEL、WS-Transaction、WS-Coordination 基础上，提出了用业务流程、协调和事务管理机制实现跨政府工作流(IGW)整合。利用事务的概念对传统工作流中的活动和任务进行分析，设计相应的 Web 服务，说明了 IGW 整合的实现机制。阐述了 IGW 整合的实现过程，并以某房地局建筑用地单独选址流程中的征询活动作为整合对象，提供了基于 Web 服务的技术路线。

关键词：Web 服务；跨政府工作流；整合

Integration and Realization of Inter-government Workflow Based on Web Service

WANG Hengjie[1], ZHANG Song[2]
(1. Business Intelligence Software Dept., Baosight Software Co., Ltd., Shanghai 201203;
2. School of Management, Fudan Univ., Shanghai 200433)

【Abstract】Process, coordination and transaction management mechanism is proposed in order to realize integration of inter-government workflow (IGW) based on Web service of BPEL, WS-Transaction and WS-Coordination. Principle of IGW integration is explained using conception of transaction and an example of Oracle Web service. Steps of IGW integration are described, and IGW integration based on Web service is illustrated by an instance.

【Key words】Web service; Inter-government workflow; Integration

从电子政务理论研究和实践来看，电子政务越来越强调以公众服务为中心构建一个透明化的系统，这往往需要采用类似企业流程重组的方法对组织结构、流程等整个政府的运作进行规范化，打破政府组织结构和政府事务程序两大迷宫。

新的 Web 服务的业务流程执行语言(BPEL)、Web 服务事务(WS-Transaction)和 Web 服务协调(WS-Coordination)规范提供了一种全面的业务流程自动化框架，使政府部门能够利用 Web 服务体系结构的技术优势来创建跨组织事务并将其自动化。由于 BPEL 允许指定业务流程以及流程与 Web 服务的关系，支持业务伙伴之间的业务协议规范和内部复杂业务流程视图，同时，WS-Coordination 和 WS-Transaction 两个补充规范制定了在将 Web 服务连接起来以创建可靠的应用程序时所需要的机制，阐释了 Web 服务应当如何按照整个应用程序输出结果的协定来参与工作和进行合作。因此，提出了一种业务流程、协调和事务管理机制来实现 IGW。

1 Web 服务简介

Web 服务的构件包括服务和服务描述。Web 服务是由服务描述(标准、规范的 XML 概念)来描述的接口，服务描述的实现就是该服务。服务是一个软件模块，它部署在由服务提供者提供的可以通过网络访问的平台上。服务描述囊括了与服务交互需要的全部细节，包括消息格式、传输协议和位置。接口隐藏了实现服务的细节，独立于服务所基于的硬件或软件平台和服务所用的编程语言，使得基于 Web 服务的应用程序成为松散耦合、面向组件并得以跨技术实现。

对于利用 Web 服务的应用程序，必须发生以下 3 个行为：发布，查找服务描述以及根据服务描述绑定或调用服务，如图 1 所示。发布是为了使服务可以得到访问，发布服务描述的位置可以根据应用程序的要求而变化。查找是服务请求者直接检索服务描述或在服务注册中心查询所要求的服务类型。在绑定操作中，服务请求者使用服务描述中的绑定细节来定位、联系和调用服务，从而在运行时调用或启动与服务的交互。

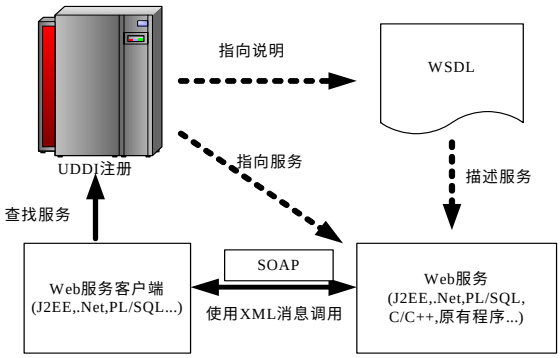

图1 Web服务技术

2 基于 Web 服务的跨政府工作流(IGW)整合

2.1 整合的实现机制

IGW 在实现过程中需要将业务伙伴提供的 Web 服务集成到本地业务流程中，并把业务流程的具体活动(原子事务)作为 Web 服务公开化、具体化。运行时根据来自流程本身的

作者简介：汪恒杰(1975—)，男，博士，主研方向：电子政务，分布式事务处理和数据仓库；张　嵩，博士
收稿日期：2005-10-20　**E-mail：**zhangsong@fudan.edu.cn

数据，动态地链接到不同提供者提供的服务，协调整个业务事务中的 Web 服务活动，从而实现 IGW。

传统意义的工作流中包含的活动由许多任务组成，活动和任务之间是一对多的关系。许多工作流管理系统均建立在数据库基础之上，作为其核心的工作流执行服务由多个工作流引擎组成，其主要功能是构建实例化、执行过程模型以及为过程模型和活动的执行进行导航。

可以从事务的角度来对传统工作流进行优化。事务是一种用来确保应用程序中的所有参与方都能达到彼此协定输出结果的机制，分为原子事务和业务活动。

原子事务(Atomic Transaction，AT)用于协调持续时间短并且在有限的信任域内执行的活动，具有“全做或者全不做”的特性。AT 规范的协议使现有的事务处理系统能够包装它们的专有协议，并能够横跨不同的硬件和软件进行互操作。

业务活动(Business Act, BA)用于协调持续时间长并希望应用业务逻辑来处理业务异常的活动。BA 规范的协议使现有的业务流程和工作流系统能够包装它们的专有机制，并能够跨信任边界和软硬件平台实现互操作。

无论是 AT 还是 BA，其事务均可分为 3 个阶段：激活、注册和完成/协调，利用 WS-Coordination 和 WS-Transaction 可以实现对事务的处理，具体过程见表 1。

表 1 WS-Coordination 和 WS-Transaction 事务处理过程

	原子事务	业务活动
激活	1 应用程序使用 WS-Coordination 框架的激活服务创建一个事务活动 2 应用程序使用 Coordination Contex 中指定的注册服务进行注册 3 应用程序执行 Web 服务上的操作	1 应用程序创建一个业务活动 2 协调服务返回 Coordination Context 3 应用程序创建一个活动 4 协调服务返回 Coordination Context 5 应用程序执行 Web 服务上的操作
注册	4 注册一个资源以进行处理 5 接受注册服务 6Web 服务执行操作并返回结果	6 业务任务向业务活动注册 7 注册服务发送一条响应
完成/协调	7 应用程序完成活动 8 协调器执行协定协议 9 资源发送 Prepared 通知 10 协调器收集投票 11 资源接受 Commit	8 业务活动检查哪些任务已完成 9 业务任务响应并告知已经完成 10 应用程序决定放弃事务(可选) 11 业务任务执行补偿操作(可选)

用事务的概念对传统工作流中的活动和任务进行分析，可以设计相应的 Web 服务。多数情况下，数据库作为服务提供者和服务使用者，图 2 以 Oracle 体系结构为例说明如何在不同的数据库之间进行相互调用。

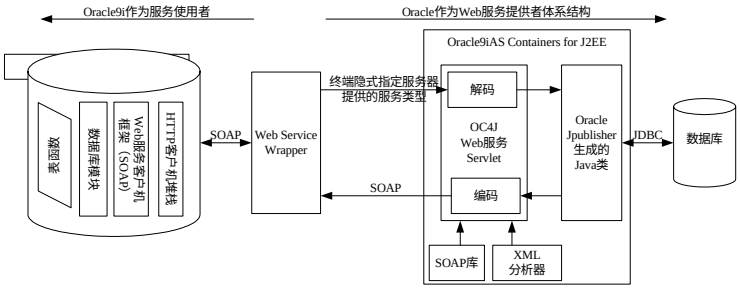

图 2 Oracle 在 Web 服务方面的技术体系结构

2.2 整合的实现过程

首先，要对计划实现的业务流程进行定义并存档。BPEL 提供了一种 XML 语法，可以用它来创建预定流程的简要描述。BPEL 文档是可执行的脚本，可以由业务流程引擎解释，从而实现所描述的流程。流程中的每一步都对应于一个业务活动，每个活动都通过与业务主流程或其业务伙伴提供的 Web 服务之间的交互而实现。该脚本中有一部分需要被具体化，从而使信息能够流入流程中并触发各种行为。业务流程执行引擎能够理解 BPEL 并负责提供实现以上所有任务的资源。系统开发者只需要定义流程、为每个活动提供业务逻辑并告知引擎如何对业务伙伴的 Web 服务以及与流程进行交互的客户进行定位。

其次，要提供一种机制来协调所有与业务流程有关的活动，从而产生一个可靠的结果。WS-Coordination 和 WS-Transaction 规范作为 BPEL 的补充，以可扩展、可互操作的方式来加强自动进行、长期运行的业务事务的可靠性。每个业务活动包括很多 Web 服务，通过协调服务的活动来保证服务的共同运行，从而使流程能够成功进行。在协调过程中，每个 Web 服务都要共享一些通用信息，用于将单独的活动链接到整个流程中。WS-Coordination 规范定义了一个框架，使 Web 服务可以通过这个框架利用共享的“协调上下文”运行，这个“上下文”包含链接各个活动所需的信息。另一方面，WS-Transaction 允许工作流监控每个单独活动的成功或失败。WS-Transaction 基于 Web

服务这一事实意味着，在政府部门特定的事务管理应用程序之间可以将事务支持变为事务可互操作。

3 整合的具体实例

以某房地局建筑用地单独选址流程说明 Web 服务基础上的跨政府工作流整合的具体实现过程。系统设计的关键在于将流程中的某些事务从旧应用转变成可通过 Web 服务访问的新应用，为方便起见，仅将流程中的征询活动作为整合的一个具体事务进行分析和设计。

简化后的征询活动业务流程见图 3。受理人接受来自企业的申请后，向房地局规划处提交征询单，规划处随后向市规划局提交征询单并接收有回复的征询单。由于规划局不是基于 Web 服务的应用，因此采用通过 SOAP 访问旧应用的技术路线，如图 4 所示。

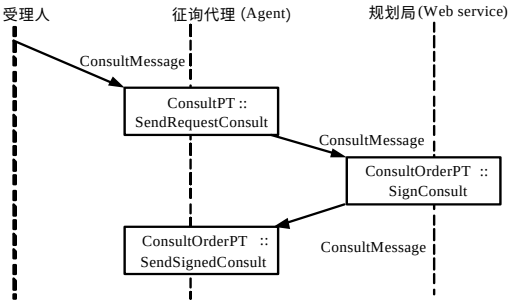

图 3 征询活动的简化业务流

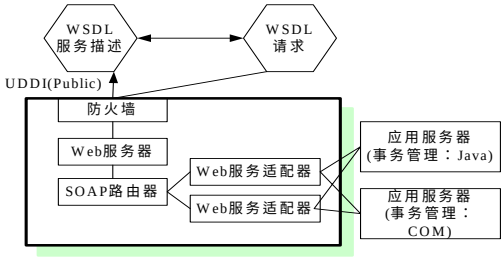

图 4 跨组织工作流的实现：利用 SOAP 访问旧应用

首先需要描述业务服务流程(BPEL，见下面的程序清单)：定义了和代理人的流程进行交互的伙伴集；描述由流程保持其持久性的消息名叫 Containers，它包含了与业务伙伴之间的 WSDL 消息。为了定义必须以怎样的顺序执行活动，ConsultOrder 流程把它的活动组成一个有向图 flow，活动被表示为节点，links 表示为连接活动的边。

ConsultOrder 的 BPEL 清单：

```
<process name="ConsultOrder">
    <partners>
        <partner name="LocalUser" serviceLinkType="agentLink" myRole="agentService"/>
        <partner name="RemoteUser" serviceLinkType="SignLink" myRole="ConsultRequester" partnerRole="ConsultService"/>
    </partners>
    <containers>
      <container name="Consult" messageType="ConsultMessage"/>
    </containers>
    <flow>
        <links>
            <link name="order-to-RemoteUser"/>
            <link name="RemoteUser-to-agent"/>
        </links>
        <receive partner="LocalUser" portType="ConsultPT" operation="sendConsult" container="Consult" >
                <source linkName"order-to-RemoteUser"/>
        </receive>
        <invoke partner="RemoteUser" portType="ConsultPT" operation="SignConsult" inputContainer="Consult">
                <target linkName"order-to-RemoteUser"/>
                <source linkName"RemoteUser-to-agent"/>
        </invoke>
        <receive partner="RemoteUser" portType="ConsultPT" operation="sendSignConsult" container="Consult" >
                <target linkName"RemoteUser-to-agent"/>
        </receive>
    </flow>
  </process>
```

其次描述服务链接类型，它指定了连接的两个伙伴之间相互提供和共同要求的 Web 服务。清单如下：

服务链接类型定义：

```
<serviceLinkType name="ConsultLink">
    <role name="agentService ">
      <portType name=" ConsultPT "/>
    </role>
    <role name="ConsultService ">
      <portType name=" ConsultPT "/>
    </role>
 </serviceLinkType>
```

在业务流程中定义伙伴时将引用服务链接类型，这个服务链接类型位于相应的流程和伙伴间的双边关系之下。伙伴定义进一步指定流程本身接受下层服务链接类型的哪个角色(myRole)以及伙伴必须接受哪个角色(partnerRole)。接受角色意味着承担提供相应 Web 服务的责任。partnerRole 属性能够引用流程期望从伙伴那里得到的 Web 服务，myRole 属性能够引用流程提供的且伙伴可以依赖和使用的 Web 服务。

业务流程中的重要操作是等待从业务伙伴那里接收消息，这是通过<receive>或<pick>活动来指定的。前者指定了消息是从哪个伙伴那里接收的，以及被伙伴用来传递消息的流程所提供的端口和操作；后者指定了所有可以从相同或不同的伙伴那里接收到的消息。无论哪个消息最先到达都将创建一个流程实例。在初始消息收到后，业务流程等待第 2 个消息。例如，如果第 1 个消息是来自某个受理者，那么流程实例被创建，然后业务流程等待来自伙伴(执行 SignConsult 操作)消息的到达。

这种方法不需要显式的生命周期命令(例如创建流程实例的命令)，对于 Web 服务的请求者来说，无须知道流程实例是否已被创建，就可以和代表业务流程的 Web 服务之间进行交互。

最后，通过如下技术步骤实现图 4 所示的具体业务过程：

(1)服务请求者选择了一个服务时，将使用 WSDL 描述找出访问该服务的方法。一旦找到该方法，WSDL 描述便生成 SOAP 请求消息发送给应用服务器，SOAP 请求作为一条 HTTP Post 请求发出。

(2)穿过防火墙后的 SOAP 请求消息由 HTTP 服务器处理。HTTP 服务器分析 HTTP 头信息并找到 SOAP 路由器组件的名称。请求消息被传递到指定的 SOAP 路由器，分析 HTTP 头并找到某个 Web 服务适配器的位置，把该请求传递到适配器。

(3)对于每个 SOAP 服务请求，Web 服务适配器调用一个后端应用。后端请求的组合结果被合并成一个 SOAP 响应回传给服务请求者。

4 结论

本文利用 Web 服务 BPEL、WS-Transaction、WS-Coordination 框架实现跨政府工作流整合，并通过案例说明了整合实现的过程和技术路线。该方法使业务流程的整合具有灵活性、可靠性和可扩展性，有利于政府部门之间的流程重组和协同工作，实现电子政务的“一站式”服务。

参考文献

1 Papantoniou A. A Critical Success Factor for E-government[C]. Proceedings of the 12th International Workshop on Database and Expert Systems Applications, 2002: 402-406.

2 Toyouchi J. Development of Service Integration Platform for One-stop Service Applications[C]. Proceedings of the 3rd International Workshop on Advance Issues of E-commerce and Web-based Information System, 2001: 123-25.

3 Greunz M. Integrating E-government Infrastructures Through Secure XML Document Containers[C]. Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001: 1690