

# 基于 BPEL 的业务流程集成<sup>①</sup>

## Business Process Integration Based on BPEL

饶锡如<sup>1,2</sup> 文俊浩<sup>1</sup> 赵瑞锋<sup>1</sup>

(1.重庆大学 计算机学院 重庆 400044; 2.重庆市汽车运输集团 微机室 重庆 40010)

**摘要:** 本文首先介绍了 BPEL(Business process execution language)的定义及其元模型,针对传统流程构造技术的不足,提出了一种基于 SOA 的业务流程集成的体系结构,并研究了业务流程模型到 BPEL 的转换规则,最后用实例介绍了如何使用 BPEL 进行流程描述。

**关键词:** 业务流程执行语言 业务流程 面向服务体系结构

近年来,Web 服务作为一种新型的分布式应用技术,具有开放性、平台独立性、基于标准协议规范、松耦合性和可复用性等优点,已经在电子商务、电子政务、企业应用集成等领域得到广泛的应用。Web 服务是把 XML、SOAP、WSDL 和 UDDI 等标准综合起来的技术,在实现异构系统之间应用集成方面有着得天独厚的优势。但是随着用户需求的不断增加,业务流程的不断再造,系统的应用环境越来越复杂多变,而 Web 服务之间是孤立和透明的,并且单个的 Web 服务只能表示单一的业务功能,很难满足系统之间复杂的业务协作需求,无法实现业务流程的集成。如何将 Web 服务有机的组合起来构建新的协同服务应用、实现更粗粒度的重用就越发具有现实意义。

### 1 BPEL简介

BPEL 即业务流程执行语言,它融合了早期 IBM 的 WSFL 和微软的 XLANG 规范的诸多优点,并且基于 XML 和 Web 服务技术,是当前进行 Web 服务编排和协作的首选标准流程语言<sup>[1]</sup>,通过它可以创建服务调用、操纵数据、抛出故障或终止一个流程等活动<sup>[2]</sup>。随着 SOA 概念的出现,服务成为最基本的操作单元。BPEL 可将 SOA 系统中的孤立服务单元按照预定的规则进行调度与协调,而 BPEL 本身也可作为 Web 服务的实现向外界提供服务调用接口,其 BPEL 的元模型如图 1 所示。流程由一系列活动组成,流程通过伙伴

链接来定义与流程交互的其他服务,服务中可以定义变量,而流程引擎可以通过关联集合将一条消息关联到特定的流程实例。下面简要介绍一下 BPEL 元模型的主要节点及含义:

(1) partnerLink 伙伴链接: BPEL 把与流程交互的其他服务称为伙伴。每个伙伴都由伙伴链接类型来定义,其中的“myRole”和“partnerRole”属性定义流程和伙伴的角色。

(2) Variable 变量: 此节点定义了 BPEL 业务流程中的中间数据和消息存放。

(3) Activity 活动: BPEL 流程由一系列步骤所组成,它们被称为活动。活动可以分为两大类:基本活动和结构化活动。基本活动包括:invoke、receive、reply、assign 等,结构化活动包括:sequence、switch、while、pick、flow 等。

(4) CorrelationSet 关联集: 此节点定义了 BPEL 业务流中各业务交互的相关集。它会利用某种形式的会话标识将消息关联起来,并将消息自动转发到正确的服务实例上<sup>[3]</sup>。

### 2 基于SOA的业务流程集成体系结构

传统的业务流程一般将各项工作按照流程模式进行静态的定义,采取相对固定的集成策略。而在实际应用中,我们往往需要根据不同需求动态集成工作流程。而把 Web 服务和 BPEL 相结合可为建设具有高度

① 收稿时间:2009-01-11

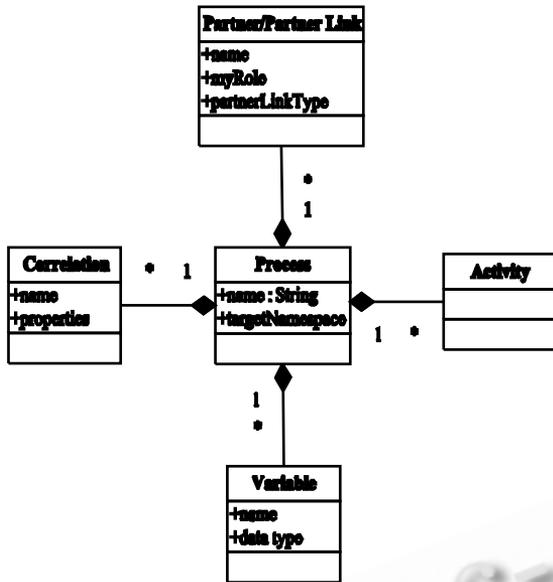


图1 BPEL元模型

供相应的环境。因为服务定义层只定义了彼此独立的

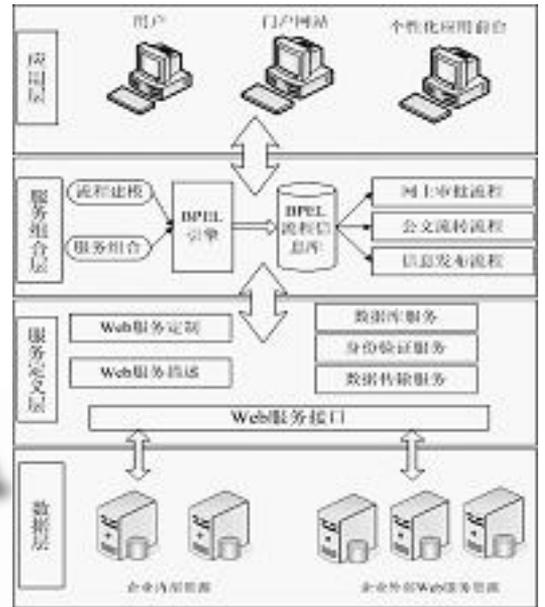


图2 业务流程集成体系结构

灵活性和敏捷性的实际应用提供了一种解决方案，用 Web 服务抽象出基本服务单元，并用 WSDL 描述成统一的接口并发布到 UDDI, 通过 UDDI 查找相应的服务，并由 BPEL 进行服务组合形成新的流程服务。基于这种 SOA 流程集成的体系结构分为四层，分别是数据层、服务定义层、服务组合层和应用层，如图 2 所示。

(1) 数据层：由于实际应用中业务处理流程很可能是跨多个部门或单位，而他们都有各自的应用系统，这些系统往往具有不同的硬件平台、操作系统、网络环境以及数据库系统。在异构数据库中存储了业务相关的基础数据，在业务流程执行过程中，为了保证流程的正确执行，必须进行访问控制。这就首先需要通过 Web 服务将数据库操作封装成统一的调用接口，为上层服务层提供统一、透明的服务调用接口。

(2) 服务定义层：服务层主要功能是提供一个 Web 服务运行环境。首先需要对整个业务流程进行分析，定义出最基本的服务单元，并用标准、规范的 XML 描述出这些 Web 服务接口。基础服务是系统实现的核心，需要抽象出实际应用中的具体服务单元，比如数据库服务、身份验证服务等<sup>[4]</sup>。然后将这些最基本的 Web 服务部署在 Web 服务容器中，并在 UDDI 中注册发布。

(3) 服务组合层：该层主要包括支持 BPEL 的工作流引擎和 BPEL 流程信息库，它的主要功能是实现服务组合的流程定义，为 BPEL 流程实例的运行管理提

服务，本身并不能形成集成应用，这就需要通过服务单元之间的协作和组合。在该层中可使用 BPEL 建模工具来设计服务流程定义，BPEL 建模工具将生成的 BPEL 定义文件作为输入，由 BPEL 引擎执行。BPEL 引擎根据流程定义调用层部署在 Web 服务运行环境中的 Web 服务，通过各种 Web 服务的组合完成相应的业务流程功能。一个 BPEL 定义模板可能生成多个流程实例，当流程实例需要调用 Web 服务时，该调用将由 BPEL 引擎中的服务调用处理器借助 Web 服务处理器调用伙伴服务提供者所关联的服务，事件处理器将对流程执行中所发生的事件进行处理。最后，工作流引擎通过 SOAP 消息将结果返回给调用方<sup>[5]</sup>。

(4) 应用层：BPEL 是一种实现无关的标准，可采用 J2EE、.Net 等作为实现平台调用已部署完毕的 BPEL 服务接口，通过向 BPEL 引擎发送 SOAP 消息来产生流程实例，实现流程的自动执行。例如可采用 B/S 结构构造客户端，实现表示层与业务逻辑的分离，可采用 JSP、ASP、AJAX 等技术向用户提供实际应用的图形用户界面。

### 3 业务流程模型的BPEL转换

我们可以通过建立图形模型来表示现实业务需求和业务的逻辑关系，比如 Petri 网，活动图，流程图

等，利用这些建模工具可以对业务流程进行正确性分析和验证。业务流程的图形模型都是由一些基本流程结构构成，对于复杂的业务模型，我们可以用多个基本模型的组合来实现。BPEL 定义了专门的结构化活动来表达流程活动，与业务流程模型之间有很好的映射关系，我们可很容易把流程结构模型转换成 BPEL，并部署到执行引擎，通过解析、执行达到用户的目标需求<sup>[6]</sup>。流程基本模型与 BPEL 映射关系如表 1 所示：

表 1 流程基本模型与 BPEL 映射关系

流程基本结构图	模型名称	映射为 BPEL 元素
	顺序模型	<pre>&lt;sequence&gt; &lt;invoke name=""&gt; &lt;invoke name=""&gt; &lt;/sequence&gt;</pre>
	并行模型	<pre>&lt;flow&gt; &lt;invoke name=""&gt; &lt;invoke name=""&gt; &lt;/flow&gt;</pre>
	选择模型	<pre>&lt;switch&gt; &lt;case condition=""&gt; &lt;invoke name=""&gt; &lt;/case&gt; &lt;/switch&gt;</pre>
	循环模型	<pre>&lt;while condition=""&gt; &lt;invoke name=""&gt; &lt;/while&gt;</pre>

## 4 案例研究

网上审批系统是电子政府系统的重要组成部分，是实现“一站式”办公的重要途径。下面以一个电子政务系统的网上行政审批流程为例来说明 BPEL 集成相关业务服务的过程。该业务流程图如图 3 所示。

该流程涉及申请部门、审批部门 A、审批部门 B 三个不同部门，可实现跨部门的网上协同审批。当审批流程接受到用户的服务申请，便按照定义好的流程步骤，初始化流程，并调用部门 A 和部门 B 的审批服务，审批结束则调用信息发布服务，公布审批处理结果信息，下面按照符合标准规范的 BPEL 来描述该流程定义过程。

### 4.1 定义 BPEL 流程的接口 WSDL

首先定义了 BPEL 流程的接口 WSDL，BPEL 引擎可以根据请求消息创建相应的 BPEL 实例，其代码片段如下：

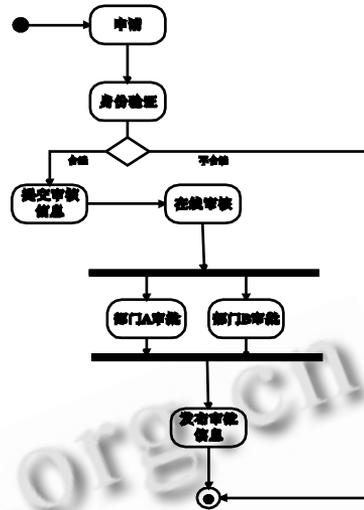


图 3 网上审批业务流程图

```

<!--定义 ApplyInfo 端口类型-->
<portType name="ApplyInfo">
  <operation name="getApplyInfo">
    <input message="tns:geyApplyRequest"/>
    <output message="tns:getApplyResponse">
  </operation>
</portType>
<plnk:partnerLinkType name="ApplyIn - foPL">
  <plnk:role name="ApplyInfoService" portType="tns:ApplyInfo"/>
</plnk:partnerLinkType>
  下面是所要调用的审批部门 A 和审批部门 B 的服务，它们的服务接口一样，可采用同一个 WSDL 文件，其 WSDL 文件代码片段如下：
  <!--定义部门服务端口类型-->
  <portType name="Depart">
    <operation name="getExamInfo">
      <input message="tns:getExamInfo - foRequest"/>
      <output message="tns:getExamInfo - Response"/>
    </operation>
  </portType>
  <plnk:partnerLinkType name="DepartPL" >

```

```
<plnk: rolename="DepartService"
portType="tns:Depart"/>
</plnk:partnerLinkType>
```

另外该流程还定义了一个审批信息发布的外部服务。同理,在其 WSDL 文件中定义名称为“Release”的 portType 和名称为“ReleasePL”的 partner LinkType。

#### 4.2 定义合作伙伴的链接

定义本流程的合作伙伴链接,通过该链接与需要调用的服务关联起来,其代码片段如下:

```
<bpel:partnerLinks>
<bpel:partnerLink myRole="ApplyInfoService"
name="ApplyInfo" partnerLinkType="tns:
ApplyInfoPL"/>
```

```
<!--定义 DepartA,DepartB,Release 的合作
伙伴-->
```

...

```
</bpel:partnerLinks>
```

#### 4.3 声明变量

上面定义的每一个合作伙伴都需要定义一个输入和输出变量,定义变量的代码片段如下:

```
<!--声明合作伙伴的输入输出变量-->
```

```
<bpel:variables>
```

```
<bpel:variable name="request" message
Type="tns:getApplyInfoRequest"/>
```

```
<!--声明合作伙伴的输入输出变量-->
```

...

```
</bpel:variables>
```

#### 4.4 BPEL 主流程定义

BPEL 主流程定义是构建 BPEL 业务流程的核心,其中<receive>是整个 BPEL 的起点,它接收客户端的请求消息,流程最后调用信息发布服务接口完成整个业务流程。代码片段如下:

```
<!--BPEL 主流程执行活动定义-->
```

```
<bpel:sequence>
```

```
<bpel:receive name="request" partner-
Link="ApplyInfo"
```

```
PortType="tns:ApplyInfo"operation="getAppl
yInfo"variable="request"createInstance="yes">
```

```
<bpel:flow>
```

```
<bpel:invoke partnerLink="DepartA"
```

```
portType=" dp:DepartPL" operation="
```

```
getExamInfo"
```

```
inputVariable=" req-depA" outputVa-
riable=" res-depA" />
```

```
<!--调用 DepartA 和 DepartB 的伙伴链接-->
```

...

```
</bpel:flow>
```

```
<bpel:rely partnerLink="Release" portType
="re:ReleasePL"
```

```
operation="getReleaseInfo"
```

```
variable="getReleaseInfoResponse"/>
```

```
</bpel:sequence>
```

至此,一个简单的网上联合审批 BPEL 流程基本定义完成了。在实际开发中往往使用图形化的建模工具进行 BPEL 的创建,我们可以通过 IBM 提供的 WID 和 WPS 更加快捷的开发和部署 BPEL 流程。

## 5 结论

Web 服务和 BPEL 的服务组合技术作为新的基于开放标准是实现 Web 服务组合的重要途径,目前已成为 workflow 技术发展的主流,可以解决应用集成中的异构问题以及为构造动态、开放环境下松耦合的业务流程集成应用提供了可行的解决方案。改善了传统业务流程构建缺乏灵活性、构建复杂等缺点,很好地实现了服务重用和服务组合。

### 参考文献

- 1 Alonso G,Casati F,Kuno H. Web Service-Concepts, Architectures and Applications. Springer-Verlag,2004.
- 2 Andrews T, Curbera F, Golland Y. Business Process Execution Language for Web Services (BPEL4WS), 2003,3.
- 3 Business Process Execution Language for Web Services version1.1, <http://www.ibm.com/developerworks/library/specification/ws-bpel/>
- 4 单既如,马殿富,朱岩.Web 服务和 BPEL 规范在人力资源管理中的应用.计算机工程与设计, 2007, 28(16),3989-3993.
- 5 王紫瑶,南俊杰,段紫辉,等. SOA 核心技术及应用.北京:电子工业出版社, 2008.
- 6 程永上,王志坚.Web 服务组合在水利领域中的应用.计算机工程与应用, 2008,44(7),209-212.