

# 基于 WS-BPEL 的 Web 服务组合技术<sup>①</sup>

刘树军<sup>1</sup>, 王炳同<sup>1</sup>, 李莉<sup>2</sup>

<sup>1</sup>(石家庄邮电职业技术学院 中国邮政网络培训学院, 石家庄 050021)

<sup>2</sup>(宁夏大学 数学计算机学院, 银川 750021)

**摘要:** 随着 Web 服务技术的不断发展, 越来越多的 Web 服务已经发布在互联网上。然而独立的 Web 服务功能通常有限, 不能满足用户或企业实际应用的需求。因此, Web 服务的组合显得越来越重要。介绍了 Web 服务组合和 WS-BPEL 的相关知识, 使用最新的业务流程执行语言 WS-BPEL 描述业务流程, 在 J2EE 集成开发环境下设计并实现了基于业务流程思想的贷款审批应用实例, 在 Web 服务执行引擎 Active BPEL 上进行部署, 对 Web 服务进行了有效的组合。

**关键词:** Web 服务; Web 服务组合; WS-BPEL; 业务流程; SOA

## Web Service Composition Technology Based on WS-BPEL

LIU Shu-Jun<sup>1</sup>, WANG Bing-Tong<sup>1</sup>, LI Li<sup>2</sup>

<sup>1</sup>(China Post Online Training College, Shijiazhuang Posts and Telecommunications Technical College, Shijiazhuang 050021, China)

<sup>2</sup>(School of Mathematics and Computer Science, Ningxia University, Yinchuan 750021, China)

**Abstract:** With the gradual development of Web services technology, more and more web services has been published on the Internet. However, a single web service just provides limited functionality, and cannot meet the needs of user or business practice. Therefore, Web service composition is becoming increasingly important. This paper introduces web services composition and knowledge of WS-BPEL, by using the latest business process execution language WS-BPEL to describe business process, by using business process, a loan approval application case is designed and realized in J2EE integrated development environment, by using the web service executive engine Active BPEL to deploy, the web services are combined effectively.

**Key words:** Web service; Web service composition; WS-BPEL; business process; SOA

## 1 引言

Web 服务作为一种实现 SOA 的方式, 以其开放性、可复用性、松散耦合性、平台独立性等特点, 解决了异构应用之间互操作、集成和通信的问题。然而单个的功能简单的 Web 服务难以满足现代企业实际应用的需求, 为了解决互联网应用的协作和集成问题, 需将独立的 Web 服务组合起来以实现强大的业务功能。WS-BPEL (Web Services Business Process Execution Language, Web 服务业务流程执行语言) 是当前进行 Web 服务组合的首选标准流程语言<sup>[1]</sup>, 在 Web 服务集成中备受关注, 并被广泛应用。使用 WS-BPEL 进行

企业业务流程建模, 可以真正实现流程的自动化及各种业务流程的持续优化和改进, 提高了现代企业的运行效率。

## 2 Web 服务组合

随着 Web 服务组合的深入研究, Web 服务组合的概念对于不同的研究机构也有着不同的认识。Stanford 大学认为“Web 服务组合是研究如何通过组合自治的 Web 服务而获得新的复杂功能的问题, 利用 Web 服务组合可以减少新应用的开发时间和费用”。IBM 认为“Web 服务组合通过 Web 服务间复杂的交互和确定不

① 基金项目: 宁夏自然科学基金(NZ0920)

收稿时间: 2011-11-15; 收到修改稿时间: 2011-12-17

同 Web 服务的执行顺序来实现的。Web 服务组合是支持业务流程逻辑的一组 Web 服务，它自身既可以是其它新的 Web 服务，也可以是最终的应用”。HP 实验室认为“Web 服务组合是服务提供者将已存在的 Web 服务按照组合规则集成为新的功能更复杂服务的增值模块，这种增值体现在可以提供更高的可用性以及服务质量，并且保障新的组合服务满足特定需求的能力”<sup>[2]</sup>。综合已有的各种定义，本文对 Web 服务组合提出一个更为完整和通用的定义。

**Web 服务组合：**利用 Internet 上分布的现有 Web 服务，根据用户总的应用需求(包括功能和非功能的需求)，在服务组合支撑平台的支持下，选择一系列符合一定规则的单个 Web 服务，组成满足总需求的服务流程，通过流程中各个服务的协同来最终完成用户的服务请求<sup>[3,4]</sup>。

### 3 Web 服务组合语言

#### 3.1 WS-BPEL

WS-BPEL(Web Services Business Process Execution Language)<sup>[5,6]</sup>是由 OASIS 组织推出的业务流程执行语言，同时是一种抽象的、高级的、可执行的建模语言，早期版本为 BPEL4WS1.1，最新版本为 WS-BPEL2.0，其前身是 IBM 的 WSFL 和 Microsoft 的 XLANG，由于 WSFL 和 XLANG 分别基于 Petri 网和 Pi-calculus，因此 WS-BPEL 借鉴并吸收了 Petri 网和 Pi-calculus 两者的长处，不仅支持面向图形的流程，同时也支持流程的结构化构造。WS-BPEL 不但将流程自身暴露为 Web 服务，而且也实现 Web 服务间流程的交互与编排。

#### 3.2 WS-BPEL 的核心元素

WS-BPEL 描述的业务流程指定了一组 Web 服务间共享的数据、业务流程涉及的合作伙以及这些伙在业务流程中扮演的角色、Web 服务操作的可能执行顺序及其相互间的依赖关系、Web 服务的补偿处理、错误处理及异常处理以及关于多个组织和服务是如何参与等一系列问题。为了实现这些功能，WS-BPEL 引入了活动(Activity)、伙伴链接(Partner Link)、相关集(correlation Sets)、变量(Variables)、事件处理程序(event Handlers)、补偿处理程序(compensation Handlers)、错误处理程序(fault Handler)等关键元素<sup>[7]</sup>。

WS-BPEL 的核心元素<sup>[8]</sup>：

(1) 活动(activity) 活动是指 BPEL 业务流程中被执行的操作，是 BPEL 业务流程的基本构成单位<sup>[9]</sup>。在 BPEL 业务流程中有基本活动和结构化活动两类不同的活动。

#### (2) 伙伴链接(partners link)

在 BPEL 业务流程中，伙伴链接引用 Web 服务描述语言定义中的伙伴链接类型定义，它是 WS-BPEL 与所调用服务之间的桥梁，伙伴链接类型表示 WS-BPEL 业务流程与包括调用 BPEL 流程的客户端以及 BPEL 流程调用的 Web 服务的相关方之间的交互。

#### (3) 相关集(correlation Sets)

在面向对象领域通过对象引用进行有状态的交互。对象引用本身提供了访问具有合适的交互状态和历史的某个对象(实例)的能力。

#### (4) 变量(variables)

每个变量的类型可以是 XML 模式简单类型、WSDL 消息类型或 XML 模式元素。属于流程作用域的变量称为局部变量；属于全局流程作用域的变量称为全局变量。

#### 3.3 WS-BPEL 的特点

由于 WS-BPEL 本身的特点，使得 WS-BPEL 成为了当前使用最广泛的一种流程语言，使得建立跨系统、跨部门甚至跨企业的业务流程成为可能。

#### (1) 松散耦合

WS-BPEL 业务流程和其所调用的 Web 服务间是松散耦合的，它们不会影响另一方面可以独立地被修改或替换。

#### (2) 服务可重用

Web 服务一旦开发完成后，可以为其客户进行长时间的服务。WS-BPEL 流程的调用对象不仅可以向该流程提供服务，而且另外的客户和流程也同时可以调用它们的所提供的相同服务。

#### (3) 递归组合方式

在 WS-BPEL 中，被组合后的服务可以作为一个新的服务被再次发布出去，使新服务可以作为组件参与后续的更进一步的组合过程，这样这个新的服务又可以参与到下一步的服务组合，这是最能体现可重用性优点的一个地方。

#### (4) 有状态的会话模式

WS-BPEL 中使用变量来保存相关的业务流程与 Web 服务交互所运用到的消息数据，在运行中将包

含多个具有状态的会话过程,可将会话状态保存下来。

### 4 基于BPEL的Web服务组合的实现

在实例实现方面,开发环境如下:应用服务器 Tomcat5.0,开发工具 J2EE IDE Eclipse Helios,Web 服务引擎 Axis2.0,Active BPEL 引擎 ActiveBpel5.0.2,BPEL 设计工具是 Apache ODE。

贷款审批实例是银行对客户是否可以进行贷款申请的资格的审查,首先客户进行贷款申请,如果客户进行申请的数额较小,则进行风险的评估;如果客户进行申请的数额较大,则直接由评审部门进行评审,最后返回客户请求的结果。

#### 4.1 实例的分析

在贷款审批的业务流程实例执行时,客户首先发送贷款申请,该申请包括个人姓名信息和贷款金额。负责贷款审批的 Web 服务根据客户输入的信息作为参数去调用风险评估服务(Assessor)和评审批准服务(Approver),如果客户申请的贷款金额小于 10000 元并且该客户信用度较好,则该审批通过;如果该客户审批的贷款金额较大,则需要调用评审批准服务提供的功能去判定,最后返回同意贷款或拒绝贷款的运行结果。“风险评估服务”的功能是查询客户的信用度,“评审批准服务”的功能是获取相关银行对该审批项目的具体评审意见。该业务流程图如图 1 所示:

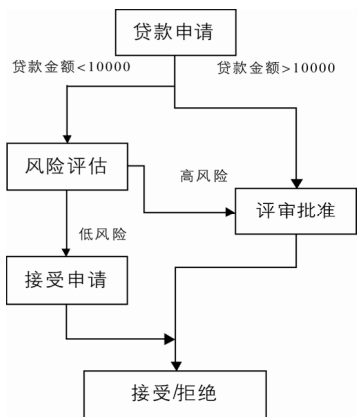


图 1 贷款审批流程图

通过对贷款申请业务流程的分析可以看出此流程中包括有贷款申请的客户、评估部门和评审部门三种角色,执行业务流程时需要两个 Web 服务,即分别对应评估部门的风险评估服务 Assessor Web Service 和评审部门的评审批准服务 Approver Web Service。

### 3.2 业务流程的创建

首先添加所需的命名空间,然后定义合作伙伴链接。代码如下:

```

<partnerLinks>
    <partnerLink myRole="approver"
name="customer"
partnerLinkType="Ins:loanApprovalLinkType"/>
    <partnerLink name="approver"
partnerLinkType="Ins:loanApprovalLinkType"
partnerRole="approver"/>
    <partnerLink name="assessor"
partnerLinkType="Ins:riskAssessmentLinkType"
partnerRole="assessor"/>
</partnerLinks>
  
```

根据 partnerRole 属性,说明服务调用伙伴链接的定义。即这个贷款申请业务流程分别调用评估部门的风险评估服务 Assessor Web Service 和评审部门的评审批准服务 Approver Web Service 来进行贷款申请。

伙伴链接后,进行变量的声明。将它们命名为 request、riskAssessment、approvalInfo、error。代码如下:

```

<variables>
    <variable
messageType="loandef:creditInformationMessage"
name="request"/>
    <variable
messageType="asns:riskAssessmentMessage"
name="riskAssessment"/>
    <variable
messageType="apns:approvalMessage"
name="approvalInfo"/>
    <variable
messageType="loandef:loanRequestErrorMessage"
name="error"/>
</variables>
  
```

定义了流程中所需的输入输出变量及中间变量后,定义流程逻辑。本实例中,使用顺序执行结构,使用 Receive 组件,它是这个流程的入口组件。该组件属性指明,贷款申请消息到来,它将创建一个流程,将消息赋值给变量 request。之后流程将调用 invoke 活动。分别调用 invokeapprover、invokeAssessor 并返回结果。

```

<invoke
    name="invokeapprover"
    outputVariable="approvalInfo"
    portType="apns:loanApprovalPT">
    <target linkName="receive-to-approval"/>
    <target linkName="assess-to-approval"/>
    <source linkName="approval-to-reply"/>
</invoke>
<invoke
    name="invokeAssessor"
    outputVariable="riskAssessment"
    portType="asns:riskAssessmentPT">
    <target linkName="receive-to-assess"/>
    <source
        linkName="assess-to-setMessage"
        transitionCondition="bpws:getVariableData('riskAssessm
ent', 'risk')='low'"/>
    <source
        linkName="assess-to-approval"
        transitionCondition="bpws:getVariableData('riskAssessm
ent', 'risk')!='low'"/>
</invoke>

```

最后，调用者得到返回结果。

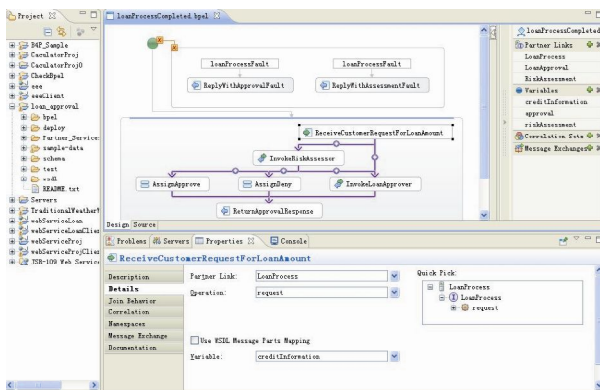


图 2 创建业务流程

### 3.3 流程的部署和执行

部署到 Active BPEL 引擎的每个业务流程，都必须首先创建一个后缀名为 .bpr 的部署流程描述文件，在该文件中指定了描述业务流程的 BPEL 配置文件 build.xml、BPEL 流程文件(后缀名为 .bpel 的文件)、流程部署描述文件(后缀名为 .pdd 的文件)、BPEL 文件，以及为该流程调用的 Web 服务定义 WSDL 文件的位置、所有合作伙伴服务和可选的配置参数<sup>[10]</sup>。

可以使用相应的 JSP 页面查看并调用服务，如图 3 所示，在贷款申请的提交页面显示很多配置文件中的值，在文本框中填写贷款申请 Loan Approval 流程的各参数，包括用户姓名，贷款的金额，评估者的响应信息以及批准者的响应信息等，点击申请贷款“Apply for a Loan”按钮，保存客户输入的值，传送应用需求到 BPEL 流程，完成流程的实例化。之后，可以在流程界面中查看 loan Approval 实例的执行结果及状态。如图所示，申请者 Liu Shujun 申请 50000 元的贷款，由于评估者对申请者的响应信息为风险低，而且批准者的响应信息为可批准，提交申请后申请的响应为“yes”，表明申请贷款成功。

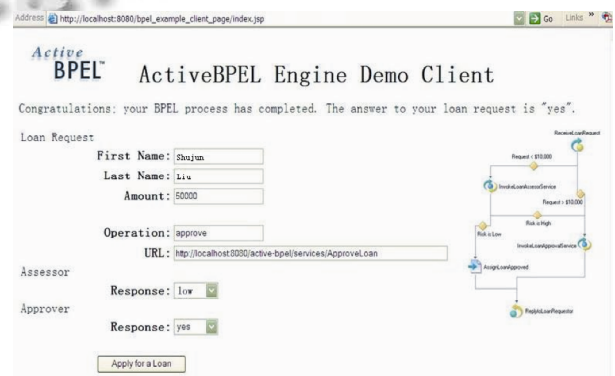


图 3 业务流程执行页面

## 4 小结

本论文介绍了 Web 服务组合技术，介绍了使用 BPEL 设计流程系统的开发模式，使用 WS-BPEL 描述业务流程，在 J2EE 集成开发环境下设计并实现了基于业务流程思想的贷款审批应用实例，在 Web 服务执行引擎 Active BPEL 上进行部署，对 Web 服务进行了有效的组合，从而满足企业级应用集成的需要。

## 参考文献

- 1 BPEL1.1. <http://www.ibm.com/developerworks/library/wsbpel/>, 2003.
- 2 李文娟. 基于 WS-BPEL 的 Web 服务组合执行引擎的设计与实现. 西安: 西北大学, 2009.
- 3 冯名正. Web 服务组合关键技术研究. 南京: 东南大学, 2006.
- 4 郭艳. 基于 BPEL 的 Web 服务组合建模工具的研究与实现. 西安: 西北大学, 2006.

(下转第 177 页)

得相关系数的平均值如表 1 所示。

表 1 加密前后像素相关性分析对比

	水平方向	垂直方向	对角方向
原图像	0.9246	0.9345	0.9152
加密图像	0.0190	0.0031	0.0133

对加密前后图像的所有像素点进行相关性分析,由表 1 可以看出加密前,相邻像素的相关性数分布 0.9 到 0.95 之间,各像素之间的相关性比较强;加密后,相邻像素的相关系数都接近于 0,说明原图像的相关性已经受到了破坏,图像已经变得杂乱无章。因此,通过上表分析可以得知,加密效果良好。

### 3.2.5 实时性分析

加密算法的耗时主要体现两方面:①生成混沌序列;②如何处理混沌序列即生成置乱与置换序列。逐点置乱要对  $m \times n$  个混沌值排序;而分块置乱只对  $(m/p) \times (n/q)$  个混沌值排序。生成置换序列时,本文的加密算法只需生成  $(m \times n) / 16$  个混沌值,文献[8]则要生成  $(m \times n) / 3$  个混沌值。分别取不同分辨率的图像做基于 NCA 的逐点与分块置乱仿真,仿真结果如图 6 所示。

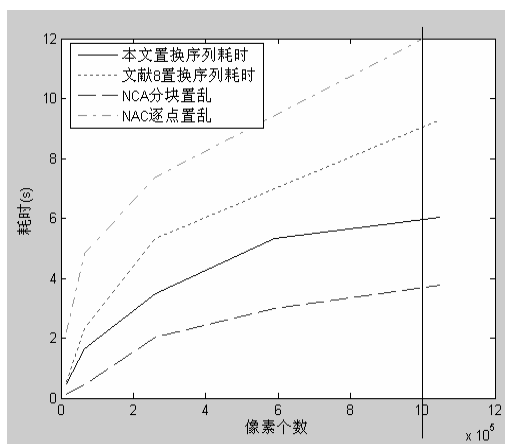


图 6 实时性仿真

由图 6 可以看出,在同时采用混沌序列 NCA 的情况下,随着像素个数的增大,本文的加密算法无论在置乱和置换方面,都表现出了更为良好的实时性,而且当分辨率越大时,效果越明显。而且当点数大于  $1000 \times 1000$  时,克服了传统 NCA 序列不能用于加密的缺点。

## 4 结语

本文中提出的基于 NCA 的图像分块加密算法克服了对高分辨率图像加密时 NCA 对迭代数的限制,仿真实验结果表明本算法在保证图像安全性不受影响下,通过 3.1 的算法减少了生成置乱与置换序列的时间,保证了数字图像加密的实时性,并且当分块大小为  $8 \times 8$  以及取 NCA 混沌值的 48 位有效数字时,能够在实时性与安全性之间得到一个良好的权衡。

## 参考文献

- 1 张燕.图像及视频序列的加密算法研究[硕士学位论文].苏州:苏州大学,2007.
- 2 黄仿元.基于 Arnold 变换的图像置乱算法及实现.贵州大学学报(自然科学版),2008,25(3):276-279.
- 3 郑怀勋,王晓然,郑敏.基于 Baker 映射的混沌图像加密算法.计算机应用与软件,2008,25(7):80-81.
- 4 于志宏,王静波,刘喆.基于 Logistic 和 Baker 映射的视频加密方法.吉林大学学报(自然科学),2008,26(3):253-258.
- 5 朱志良,张伟,于海.基于 Lorenz 混沌系统的 MPEG 视频加密算法.计算机应用,2008,28(12):3003-3006.
- 6 Haojing. A new chaotic algorithm for image encryption. Chaos, Solitons and Fractals,2006,29:393-399.
- 7 穆秀春,晷鸿.一种基于混沌序列的彩色图像加密算法.现代电子技术,2010,325(14):53-55.
- 8 Chen GR, Mao YB, Chui CK. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons and Fractals,2004,21:749-761.

(上接第 209 页)

- 5 OASIS WSBPEL TC. Web Services Business Process Execution Language Version 2.0. <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>,2007.4.
- 6 OASIS. Web Services Business Process Execution Language Version 2.0. <http://www.oasis-open.org>,2007.
- 7 王莉,刘厚泉,吴雪峰.基于 BPFL 的业务流程管理系统架构的研究与应用.计算机工程与设计,2006,27(18):3507-3510.

- 8 BPEL. bpel4ws 规范描述. <http://wenku.baidu.com/view/fde02bd7195f312b3169a5fb.html>.
- 9 朱学文.基于 BPEL 的医疗信息系统集成技术研究.上海:上海交通大学,2009.
- 10 白东伟.基于语义的 Web 服务匹配与发现技术研究.北京:北京邮电大学,2007.