

基于 BPEL 的 Web 服务快速组合框架

崔福东, 乔彦友, 常原飞

(中国科学院遥感应用研究所, 北京 100101)

摘要: 针对当前 Web 服务组合语言以及 Web 服务组合过程的复杂性, 提出基于业务流程执行语言(BPEL)的 Web 服务快速组合框架。该框架用有向无环图(DAG)表示服务组合逻辑, 通过对 DAG 解析自动生成发布工作流所需要的各种文档。对原型系统的验证结果表明, 该框架能将 BPEL 的复杂性隐藏起来, 达到 Web 服务快速组合的目的。

关键词: 业务流程执行语言; Web 服务; 服务组合; 有向无环图; 工作流

Web Services Rapid Composition Framework Based on BPEL

CUI Fu-dong, QIAO Yan-you, CHANG Yuan-fei

(Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing 100101)

【Abstract】 For the complexity of Web service composition language and the composition procedure, this paper proposes a framework which can composite Web services rapidly based on Business Process Execution Language(BPEL). The framework represents the service composition logic by Directed Acyclic Graph(DAG), and resolves the DAG to generate all the documents needed automatically. Experimental result on the prototype system demonstrates that the framework archives the target of compositing Web services rapidly by concealing the complexity of BPEL.

【Key words】 Business Process Execution Language(BPEL); Web services; service composition; Directed Acyclic Graph(DAG); workflow

1 概述

Web 服务技术提供了一种分布式环境, 应用程序或组件以一种跨平台、跨语言的方式进行无缝集成^[1]。Web 服务较其他分布应用集成方法最大的特点是完全低耦合、可组合可重用和跨平台特性^[2]。由于 Web 服务平台无关, 其他基于任何语言和平台的应用程序都可对其进行调用, 最大程度实现了应用程序的复用。单一 Web 服务往往难以满足企业级应用, 因此, 需要把已有的 Web 服务组合起来, 创建新的功能^[3]。针对 Web 服务之间的集成问题, 国内外开展了大量的研究工作, 并发布了一些用来描述 Web 服务组合的语言, 如 Web 服务流语言(Web Service Flow Language, WSFL)、XLANG、业务流程建模语言(Business Process Modeling Language, BPMML)、WSCI(Web Service Choreography Interface)和业务流程执行语言(Business Process Execution Language, BPEL)等。文献[4]对以上各种语言进行了比较, 指出 BPEL 更具有表达力。当前, 许多企业开发了针对 BPEL 工作流的产品, 如 Active Endpoints 的 ActiveBPEL、IBM 的 WBI Server Foundation、Oracle 的 BPEL Process Manager、Microsoft 的 BizTalk Server、BEA 的 WebLogic Integration 以及 Eclipse BPEL Project 等。这些工具都提供图形化建模, 自动生成标准流程描述文件, 但都要求用户具有 Web 服务和 BPEL 的专业知识, 工作流创建过程费时费力。针对此问题, 本文提出一种快速组建 Web 服务的框架。

2 BPEL 文档结构

BPEL 是一种基于 XML 的流程描述语言, 是专门为整合 Web 服务而定制的一项工作流管理规范, 目前已经成为 Web 服务组合工业上的事实标准。一个基本的 BPEL 文档包括以下内容:

<import>: 用来导入所需的 WSDL 文档;
<partnerLink>: 表示工作流与 Web 服务之间的伙伴关系;
<variables>: 包含多个<variable>元素, 表示 Web 服务的输入和输出;
<sequence>: 表示一个活动序列;
<receive>: 用来接收工作流的初始输入;
<assign>: 用来完成值的传递;
<invoke>: 用来调用一个 Web 服务;
<reply>: 将结果返还给用户。

3 Web 服务工作流建模

有向无环图(Directed Acyclic Graph, DAG)是指一个无环的有向图, 被广泛应用于描述业务逻辑^[3], 是描述一项工程或系统进程过程的有效工具之一。鉴于 DAG 的简单直观, 用 DAG 对 Web 服务工作流进行建模。这种特殊的 DAG 称为 WF DAG(WorkFlow DAG)。一个完整的 WF DAG 由 3 个部分构成: 开始和结束节点, 活动节点和连接。WF DAG 的设计要符合一定的规则, 称为有效性。

3.1 开始节点和结束节点

开始节点和结束节点分别表示整个工作流的入口点和出口点。开始节点用来记录整个工作流的输入参数列表, 而结束节点用来记录整个工作流的返回值。在 WF DAG 中, 用 2 个特殊的节点表示开始和结束节点, 如图 1 所示, S 表示开始节点, E 表示结束节点。

基金项目: 国家科技支撑计划基金资助项目“国家级林业有害生物灾害监测与预警运行系统”(2006BAD23B04)

作者简介: 崔福东(1983—), 男, 硕士研究生, 主研方向: Web 服务; 乔彦友, 研究员、博士生导师; 常原飞, 助理研究员

收稿日期: 2009-11-24 **E-mail:** fromirsa@hotmail.com

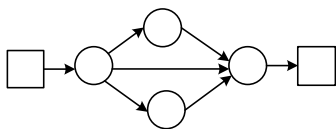


图1 用 WF DAG 表示的工作流

3.2 活动和连接

WF DAG 中开始和结束节点之外的节点称为一个活动，代表某一个 Web 服务的某一操作；一条有向弧段称为一个连接，用来将 2 个相邻的活动进行串连，表示活动间的依赖关系(或运行次序)，并存储 2 个相邻 Web 服务输入与输出的匹配关系。活动如图 1 中小写字母所示，连接如图 1 中的有向弧段所示。

3.3 有效性

WF DAG 设计必须遵循一定的规则，以便能够正确地解析。所谓有效性，是指 WF DAG 必须至少满足如下要求：(1)WF DAG 中不能存在环；(2)WF DAG 之外不能有孤立的的活动存在；(3)每一个连接所代表的参数匹配，两端的参数类型必须一致。

3.4 WF DAG 的遍历及 BPEL 文档的生成

该框架的核心算法主要有 2 个部分：一是有效性检验，二是 WF DAG 的遍历。对于有效性检验，一是判断 WF DAG 是否存在环，判断无环的算法已经比较成熟，其余 2 条有效性规则的判断算法也比较简单，在此不再赘述。本文重点在于介绍如何用 DAG 对 Web 服务工作流进行建模，所以对如何生成 WSDL 文档和发布描述符文档不作具体介绍。下面主要介绍 WF DAG 的遍历算法。

图论中对于图的遍历分为 2 种：深度优先和广度优先，这 2 种遍历方式可以从任意节点开始，可以有多种路径。WF DAG 用来表示 Web 服务工作流的业务流程，对它的各个节点的访问顺序体现为工作流的业务逻辑，即遍历路径不是任意的，而且必须从开始节点开始遍历，所以，不能采用深度或广度优先的遍历方式。工作流中的服务之间有一种依赖关系，如任意相邻的 2 个 Web 服务 a 和 b，连接方向是 a 指向 b，那么 b 依赖于 a，因为 b 能够有效执行的前提是 a 已经执行完毕，并且把 a 的返回结果指配给 b。基于这种依赖关系，采取一种称之为基于依赖的遍历方式。所谓基于依赖，是指在完成对某一节点的访问之前，先要完成对这一节点所有前驱节点的访问。

本框架采用 Visitor 设计模式对 WF DAG 进行设计，将对节点进行的具体访问操作(生成 BPEL 的元素)封装到 Visitor 接口的实现类中，如图 2 所示。Vertex 表示活动，Edge 表示连接，Graph 表示工作流，Graph 完成对所有节点的遍历。

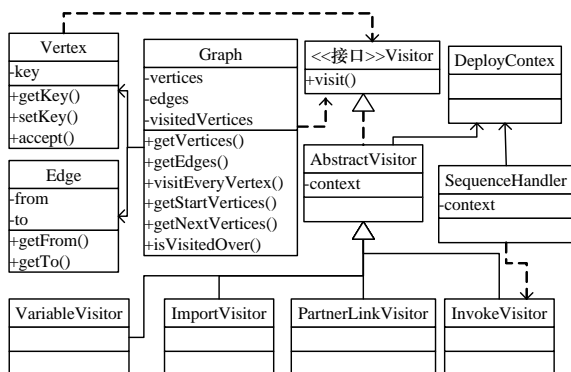


图2 WF DAG 类图

在基于依赖的遍历算法基础上，通过实现 Visitor 的具体类来生成 BPEL 文档的各个部分。

遍历过程如下所示：

```
protected void visitVertex(Vertex v, Visitor visitor) {
    if (!isVisitedOver(v)) //完成基于依赖的控制
        return; //如果 v 依赖的节点未全部访问，返回
    else {
        v.accept(visitor);
        //如果 v 依赖的节点已全部访问 E 访问节点 v
        visitedVertices.add(v.getKey()); //标记节点 v 已被访问
        List<Vertex> nexts = getNextVertices(v);
        for (Vertex vertex : nexts) {
            visitVertex(vertex, visitor); //完成对 v 的后继节点的访问
        }
    }
}

public void visitEveryVertex(Visitor visitor) {
    //获得开始节点列表，对每个开始节点进行访问
    List<Vertex> startVertices = getStartVertices();
    for (Vertex v : startVertices)
        visitVertex(v, visitor);
}
}
```

4 系统框架

系统框架如图 3 所示，工作流设计界面与用户交互，提供服务查询和对工作流进行编辑及发布的功能，协助用户完成工作流设计；客户端以 XML 格式将设计结果提交，服务器端通过解析器从 WF DAG 提取信息，并根据这些信息生成发布工作流所需的发布描述文档、WSDL 文档和 BPEL 文档；最后由发布器将各种文档发布到工作流引擎。UDDI 提供了 Web 服务信息，工作流引擎提供了 BPEL 的运行环境。

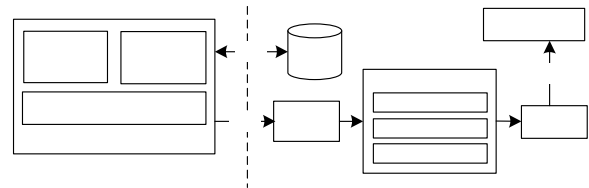


图3 系统框架

原型系统基于 JavaEE 实现。UDDI 和 BPEL 引擎分别采用 Apache 的 jUDDI 和 ODE(Orchestration Director Engine)。jUDDI 是由 Apache 基金组织维护的基于 Java 的开源 UDDI 实现；ODE 是由 Apache 基金组织维护的开源 BPEL 引擎。用户界面基于 Java Applet 实现。

下面给出原型系统的一个应用实例。该实例涉及 5 个 Web 服务：(1)StudentStore：存储学生的基本信息，通过学号获取信息；(2)ScoreStore：存储学生的成绩，通过学号获取信息；(3)PracticeStore：存储学生的实习信息，通过学号获取信息；(4)WeightedScoreCal：根据成绩列表计算加权成绩；(5)ResumeMaker，根据以上 4 个服务返回的数据生成制作学生简历所需要的信息。

用上述 5 个 Web 服务组装成一条工作流 MakeResume，功能是输入学生的学号，输出制作该学生简历所需要的信息。具体流程为：(1)StudentStore 根据用户输入的学生学号，查询该学生的基本信息；(2)ScoreStore 根据学号获得成绩列表；(3)将查询的成绩列表作为 WeightedScoreCal 服务的输入，获得加权成绩；(4)PracticeStore 根据学号获得实习信息；(5)ResumeMaker 将上述 4 个步骤的结果作为输入生成简历信息。构建工作流的过程如图 4 所示。

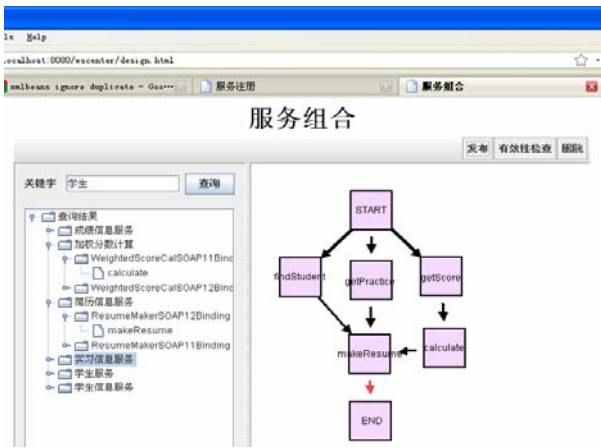


图4 原型系统建模界面

工作流发布成功后,可视为一个新的 Web 服务。通过 Eclipse 的 Web Services Explorer 对实例工作流进行验证,返回的 SOAP 消息的数据部分为:

```
<resume type="demo.Resume">
  <basicInfo xmlns="http://demo/xsd" type="demo.Student">
    <age>23</age>
    <gender>男</gender>
    <name>张三</name>
    <nativePlace>北京</nativePlace>
    <studentNumber>2009</studentNumber>
  </basicInfo>
  <practiceInfo xmlns="http://demo/xsd" type="demo.Practice">
    <content>学生管理系统开发</content>
    <fromDate>2008-09-15</fromDate>
    <toDate>2009-04-20</toDate>
  </practiceInfo>
</resume>
```

(上接第 261 页)

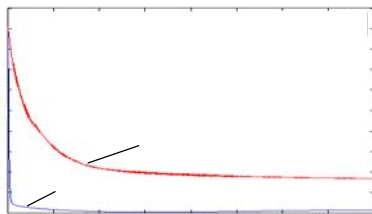


图2 能量下降曲线的对比

表2 改进模型实验相关数据的对比

方法	参数	迭代次数	时间/s	能量
差分方法	$\lambda=0.8, \tau=0.005$	800	62.375	1.330×10^4
对偶方法	$\theta=0.02, \lambda=5, \tau=0.125$	400	30.719	5.218×10^3

6 结束语

本文基于统一表达的稳健的数据项和总变差形式的规则项,系统建立针对小位移光流场计算的对偶方法,并且对基于不同的前提假设的 6 种数据项的选择进行了比较。引进光流场的辅助变量,将原来的能量最小化问题转化为交替迭代的计算,对光流场变量采用对偶方法,避免了人为误差的引入,对辅助变量采用收缩算子,可有效提高计算效率。本文

```
<weightedScore xmlns="http://demo/xsd">3.8</weightedScore>
</resume>
```

5 结束语

本文提出的 Web 服务快速组合框架使用户在浏览器端即可完成工作流的设计。该框架在客户端用 WFDAG 表示工作流,简单直观。后台根据 WFDAG 所表示的业务逻辑自动生成各种文档,将复杂的 BPEL 知识细节对用户隐藏起来,用户只需关心 Web 服务工作流的业务逻辑,大大提高了服务组合的效率,同时可以使更多的人根据自身需要组合 Web 服务。

对原型系统的实例应用表明,该框架设计可行,但是这只是一个原型系统,还存在一些问题需要解决。首先,BPEL 像其他高级编程语言一样,提供了流程控制和异常处理等功能,该框架暂时还未将这些内容融入进来;其次,在生成的 BPEL 文档中,各种活动(Activity)的执行是以串行的方式完成的,为了提高执行效率,需考虑在需要的地方进行并行处理。

参考文献

- [1] Chappell D, Jewell T. Java Web Services[M]. [S. l.]: O'Reilly, 2002.
- [2] 丁兆青,董传良.基于 SOA 的分布式应用集成研究[J].计算机工程,2007,33(10):246-248.
- [3] Chandrasekaran S, Miller J A, Silver G S, et al. Composition, Performance Analysis and Simulation of Web Services[J]. EM-Electronic Markets, 2003, 13(2): 56-64.
- [4] Wohed P, Aalst W M P, Dumas M. Analysis of Web Services Composition Languages: The Case of BPEL4WS[EB/OL]. (2004-10-19). http://www.workflowpatterns.com/documentation/documents/bpel_er.pdf.

编辑 顾逸斐

方法不但计算效率高,而且实施简单,便于在大位移光流计算、运动分割与跟踪、图像配准等研究中推广应用。

参考文献

- [1] Horn B, Schunck B. Determining Optical Flow[J]. Artificial Intelligence, 1981, 17(1): 185-203.
- [2] Weickert J, Bruhn A, Brox T, et al. A Survey on Variational Optical Flow Methods for Small Displacements[M]. Berlin, Germany: Springer, 2006.
- [3] Bruhn A. Variational Optical Flow Computation: Accurate Model and Efficient Numerics[D]. Saarbrücken, Germany: Department of Mathematics and Computer Science, Saarland University, 2006.
- [4] Chambolle A. An Algorithm for Total Variation Minimization and Applications[J]. Mathematical Imaging and Vision, 2004, 20(1/2): 89-97.
- [5] Zach C, Pock T, Bischof H. A Duality Based Approach for Realtime TV-L1 Optical Flow[C]//Proc. of DAGM'07. Berlin, Germany: Springer, 2007.
- [6] Pock T, Urschler M, Zach C, et al. A Duality Based Algorithm for TV-L1-optical-flow Image Registration[C]//Proc. of the 10th International Conference on Medical Image Computing and Computer Assisted Intervention. NY, USA: [s. n.], 2007.
- [7] Li Yingying, Osher S. A New Median Formula with Applications to PDE Based Denoising[J]. Commun. Math. Sci., 2009, 7(3): 741.

编辑 索书志

