

# 基于 BPEL4WS 的分布式应用系统的研究与实现

刘 涛, 高 珍, 张志浩

( 同济大学 计算机科学与工程系, 上海 200092)

摘 要: 研究了基于 Web 服务的业务流程执行语言 BPEL4WS( Business Process Execution Language for Web Services) 的实现机制和关键技术, 在此基础上, 结合具体应用分析了 BPEL4WS 在分布式应用系统中的应用和实现。

关键词: Web 服务; 业务流程; BPEL4WS; WSDL; TravelFlow

中图法分类号: TP311 文献标识码: A 文章编号: 1001-3695(2004)08-0220-03

## Research and Implementation of the Distributed Application System Based on BPEL4WS

LIU Tao, GAO Zhen, ZHANG Zhi-hao

( Dept. of Computer Science & Engineering, Tongji University, Shanghai 200092, China)

**Abstract:** This paper makes research on the performance system and key technology of BPEL4WS( Business Process Execution Language for Web Services) . Besides, with the specific application, it elaborates on the analysis of the development and realization of the distributed application system based on BPEL4WS.

**Key words:** Web Service; Business Process; BPEL4WS; WSDL; TravelFlow

### 1 引言

Web 服务自 20 世纪末诞生以来, 备受业界人士的关注。基于 Web 服务构架企业分布式应用系统的思想在 Microsoft, IBM, SUN 等公司的支持下得到了迅速推广。由于 Web 服务是一种崭新的技术, 还没有形成完整的规范和成熟的体系, 目前的 Web 服务就像是互联网上孤立的逻辑程序单元, 不能很好地融合起来。如何冲破这种孤立, 进行 Web 服务的集成, 继而形成企业分布式应用系统成为 Web 服务发展的一个重要的问题。为了解决这一问题, 出现了一个崭新的发展方向——业务流程, 它充分体现了 Web 服务集成的思想, 将不同的 Web 服务按照指定的业务需求或功能需求联系起来实现一个更复杂的功能体。基于业务流程开发企业分布式应用系统成为当前基于 Web 服务开发应用系统的主流方案。2002 年 8 月, IBM, Microsoft 和 BEA 联合提出了有关 Web 服务集成的新规范——基于 Web 服务的业务流程执行语言 BPEL4WS( Business Process Execution Language for Web Services)。

BPEL4WS 是基于 XML 的流程定义语言, 可以定义如何使用业务流程相互配合。BPEL4WS 的出现取代了原有的 IBM 的 Web 服务流程语言( Web Service Flow Language, WSFL) 和微软的 XLANG, 它集 WSFL 和 XLANG 两家之长( 前者支持面向图形的流程, 后者则支持流程的结构化构造), 形成了一个支持以很自然的方式实现各种类型的业务流程的规范。

### 2 BPEL4WS 的实现

BPEL4WS 支持两种截然不同的使用情形: 实现可执行的

业务流程和描述不可执行的抽象流程。作为可执行流程的实现语言, BPEL4WS 的作用是将一组现有的 Web 服务整合起来, 从而定义一个新的 Web 服务。与其他任何 Web 服务一样, 整合服务的接口也被描述为 WSDL portType 集合。整合( 也称为流程) 指明了服务接口与整合的总体执行的配合情况。从外部看到的 BPEL4WS 流程的执行情况如图 1 所示。

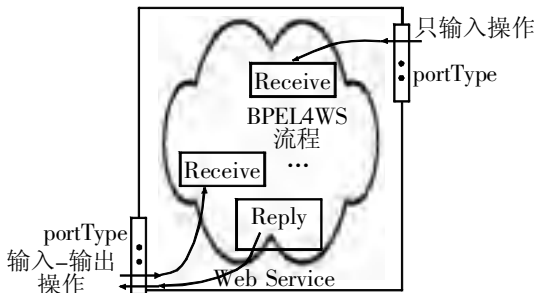


图 1 BPEL4WS 流程实现的 Web 服务视图

不同于用传统的编程语言来实现 WSDL 服务, 图 1 中每个 portType 的每项操作并不映射成 BPEL4WS 中的一个独立的逻辑块。服务的整个类型( 即该服务的 portType 集合) 由单个 BPEL4WS 流程实现。与调用接口操作的外部用户相对应的特定“入口点”在 BPEL4WS 描述中指明。这些入口点接收从只输入( Input-only) 操作或输入 - 输出( Input-output) 操作传入的消息, 对于后一种情况, 流程还必须指明输出消息在哪里生成。BPEL4WS 只使用和支持 WSDL, 只输入操作和输入 - 输出 / 请求 - 响应( Request-response) 操作; 只输出( Output-only) / 通知( Notification) 操作和输出 - 输入( Output-input) / 要求 - 响应( Solicit-response) 操作既不需要也不支持。

### 3 BPEL4WS 的关键技术

BPEL4WS 支持的业务流程能够指定一组 Web 服务操作

的可能执行顺序, 这些 Web 服务间共享的数据、业务流程涉及哪些伙伴以及这些伙伴在业务流程中扮演什么角色, 一组 Web 服务的共同异常处理以及关于多个服务和组织是怎样参与的其他问题。为了实现这些功能, BPEL4WS 引入了活动、服务链接、伙伴和服务引用、容器、相关性、错误处理、作用域和消息属性等关键元素。

(1) 活动(Activity)。BPEL4WS 流程类似于用来表达算法的流程图。流程的每一步称为一个活动。活动包括基本活动(Basic Activity)和结构化活动(Structured Activity), 基本活动负责实现一定的原子功能, 如赋值数据(Assign)、终止业务流程(Terminate)等; 结构化活动则利用循环(While)、分支(Switch)等元素对基本活动进行组装整合以实现系统所需的逻辑功能 BPEL4WS 允许递归地组合结构化活动, 以表达任意复杂的算法, 这些算法代表和体现了服务的具体实现。

(2) 服务链接(ServiceLink)、伙伴(Partner)和服务引用(ServiceReference)。BPEL4WS 把与流程交互的其他服务称伙伴。每个伙伴由服务链接类型来描述。服务链接用来直接模拟对等伙伴关系, 为了描述两个服务间的关系, 服务链接类型定义了关系中每个服务所扮演的“角色”, 并指定每个角色所提供的端口类型(portType)。为了表示描述伙伴服务所需的动态数据, BPEL4WS 定义了服务引用, 服务引用指定了服务的描述信息、服务名称及服务引用相关的信息。其中, 引用属性主要用于传递一些附属信息, 通过与消息属性及赋值<Assign>操作相结合, 服务引用使得在 BPEL4WS 中能动态地为某个服务类型选择提供者和调用它们的操作。

(3) 容器(Container)。容器提供的方式可用于保存组成业务流程状态的消息, 所保存的消息往往是已从伙伴那里接收到的消息或将被发送给伙伴的消息。容器可被指定为 Invoke, Receive 和 Reply 等活动的输入容器或输出容器, 保存在服务间流动的数据消息。在流程开始的时候, 所有的容器都未被初始化, 可以通过赋值<Assign>活动或<Receive>活动接收消息来初始化容器。在业务流程执行过程中, 一般通过赋值活动来实现数据在不同容器间的流转。

(4) 相关性(Correlation)。在业务流程中服务间的交互是有状态的交互, 在面向对象的世界中, 这种有状态的交互需要通过对象引用来实现, 这种实现方式适用于紧密耦合的实现, 而在松散耦合的 Web 服务的世界中, 这种引用的使用将造成与实现之间脆弱的依赖关系, 这种依赖关系在每个业务伙伴采取不同的实现细节的情况下是无法适应的。在松散耦合的情况下, 解决的方法是依靠定义伙伴间线路层的合同的业务数据和通信协议头并尽可能地避免在实例路由中使用特定于实现的标记。BPEL4WS 提出了相关性概念, 以解决实例路由的问题。相关集被用于 Invoke, Receive 和 Reply 等活动以指出哪个相关集出现在被发送和接收的消息中。

(5) 错误处理(faultHandler)。类似其他编程语言, BPEL4WS 使用 Catch 元素来响应错误处理。每个定义的 Catch 活动能拦截某种指定的故障, 由全局唯一的故障名和与该故障关联的数据的容器来定义。如果没有故障名, 那么 Catch 将拦截全部有适合类型的故障数据的故障。CatchAll 子句可被用来捕获所有未被更特定的 Catch 处理程序捕获的故障。业务流程中的错误处理在很大程度上依赖于补偿的概念, 即通过进

行必要的操作, 撤销先前活动造成的影响。

4 BPEL4WS 的工作引擎

IBM AlphaWorks 首先推出了支持 BPEL4WS 的工作引擎——BPWS4J(Business Processes in Web Services for Java)。BPWS4J 主要包括两大部分 BPWS4J- Editor, BPWS4J- Engine。前者主要用于实现对 BPEL4WS 流程的定义(它可以与 IBM 提供的 Eclipse 平台结合使用, 实现可视化的业务流程定义); 后者主要提供对业务流程执行的支持。BPWS4J 软件包于 2002 年 8 月在 AlphaWorks 上发行, 它提供了一个运行时引擎以及一个用于创建流的工作流编辑工具。

完整的、可部署的 BPWS4J 运行时单元由以下三部分组成: 描述流程的 BPEL4WS 文件; 一个 WSDL 文件, 它描述了流程定义所引用的消息、操作、端口类型以及其他信息(服务链接类型、相关性属性等); 流程中包含的每个伙伴的 WSDL 定义, 除非流程不使用伙伴所提供的任何 WSDL 操作。在这三部分当中, BPWS4J 编辑器允许创建描述流程的 BPEL4WS 文件。要编辑部署 BPEL4WS 流程所需的 WSDL 文件, 可以使用其他工具。不过, 这些工具将不支持 BPEL4WS 规范中已经提议的非标准的 WSDL 扩展(如服务链接、相关性属性等)。

5 应用实例

某旅行社试图为客户提供以下便捷的旅游服务: 旅游计划制定; 机票预订和宾馆预订。客户可以根据业务需要、经济条件、个人兴趣爱好等条件制定自己的旅游计划。旅行社获取客户的旅游计划后, 向客户提交符合客户要求的航班和宾馆预订计划。在得到客户确认信息后, 实行预订计划, 并将结果(包括预订的机票、宾馆信息等)告知客户, 完成一次旅游服务操作。

5.1 基于 BPEL4WS 的系统实施方案

基于 Web 服务和 BPEL4WS 的分布式应用系统的实现主要包括以下几个步骤:

- (1) 行业协会定义并发布 Web 服务接口文档;
- (2) Web 服务提供商获取 Web 服务接口文档, 实现并发布 Web 服务;
- (3) Web 服务集成应用提供商基于 BPEL4WS 定义并实现业务流程;
- (4) 执行业务流程, 为用户提供最终服务。

针对旅行社分布式应用系统, 首先应保证基础 Web 服务的提供: 航空公司将其提供的航班查询、机票预订的服务以 Web 服务的形式, 提供给用户等。在此基础上, 旅行社提供旅游集成服务功能, 我们把它称为 TravelFlow, 用 BPEL4WS 来描述其业务流程, 根据分析, 可以得出 TravelFlow 的主要业务流程图如图 2 所示。

图 2 中的方框代表由几个 Web 服务组成的 Web 服务业务流程, 可以看出该业务流程 TravelFlow 的主要功能是从客户处获取旅游计划, 根据旅游计划查询 UDDI 注册中心, 返回符合客户要求的航空公司和宾馆列表。客户从该列表中选择指定的航空公司和宾馆, TravelFlow 返回它们的服务地址, 以便后续的预订工作。航空公司和宾馆服务确定后, TravelFlow 便可以

与指定的 AirlineService, HotelService 进行交互, 返回符合客户要求的航班及房间信息。客户选择指定的航班及房间, TravelFlow 进行相应的预订工作。所有工作完成后, TravelFlow 将客户的旅游路线( 包括电子票据等信息) 返回给客户, 实现整个旅游服务。

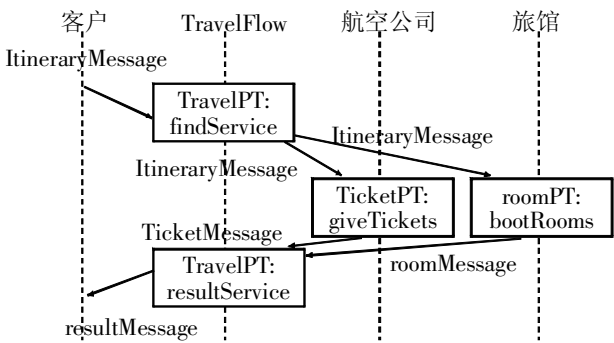


图 2 TravelFlow 业务流程图

与不使用业务流程的传统方案比较, 基于 BPEL4WS 的分布式应用系统 TravelFlow 的特点及优势在于整个业务流程一经实现, 就不再需要业务员的干涉, 客户可以直接与 TravelFlow 交互, 消费其提供的服务功能。业务流程中涉及到的航空公司及宾馆的 Web 服务是从 UDDI 注册中心( Public UDDI Registry 或 Private UDDI Registry) 获取的, 这样, 如果有新的航空公司或宾馆提供了相应的服务, 可以通过将其 Web 服务注册到该注册中心, 由业务流程通过 UDDI 自动发现, 业务流程不需做任何修改, 因而使得应用系统具有良好的扩展性。

### 5.2 实例分析

从 TravelFlow 的业务流程图中可以看出, 整个流程从收到客户的旅游计划开始, 顺序经历了航空公司和宾馆选择, 航班和房间选择, 机票和房间预订几个步骤, 最终将旅游路线返回给客户。相应的 BPEL4WS 定义文档, 由 < receive > 活动接收客户旅游计划开始启动, 生成业务流程实例, 将客户的旅游计划交由 TravelService 服务处理, 后续的几个步骤由 < sequence > 活动定义顺次执行。值得注意的是, 具体的 AirlineService 和 HotelService 一经确定, 查询、预订航班和查询、预订房间的服务操作可以并行执行, 这一操作由 < flow > 活动定义。当上述所有的服务操作完成后, 业务流程将各参与服务返回的信息发送给客户, 由 < reply > 活动来定义。

TravelFlow 的关键流程示例如下:

```
< sequence >
  < receive partner = "customer" portType = "travelplanPT"
operation = "sendTravelPlan"
  container = "travelPlanRequest" createInstance = "yes" >
  < /receive >
  < invoke partner = "travelService" portType = "travelPT"
operation = "chooseTravelPlan"
    inputContainer = "travelPlanRequest" outputContainer = "travelPlan-
Response" >
    < /invoke >
    < assign >
    < copy >
    < from container = "travelPlanResponse" part = "airlineService" / >
    < to partner = "airlineService" / >
    < /copy > ...
    < /assign > ...
  < flow >
  < sequence > ... < /sequence > ...
  < reply partner = "customer" portType = "travelplanPT" operation = "
sendTravelPlan"
```

```
container = "travelPlan" / >
< /flow >
< /sequence >
```

TravelFlow 业务流程的服务链接、伙伴和服务引用部分示例如下:

```
< partners >
  < partner name = "customer" serviceLinkType = "lns: travelPlanLT"
partnerRole = "travelPlanService" / >
  < partner name = "travelService" serviceLinkType = "lns: travelLT"
partnerRole = "travelService" / > ...
< /partners >
```

TravelFlow 业务流程的容器部分示例如下:

```
< containers >
  < container name = "travelPlanRequest" messageType = "ta: travel-
PlanRequest" / >
  < container name = "travelPlanResponse" messageType = "ts: travel-
PlanResponse" / > ...
< /containers >
```

TravelFlow 业务流程的错误处理部分示例如下:

```
< faultHandlers >
  < catch faultName = "cannotCompletePlan" faultContainer = "travel-
PlanFault" >
    < reply partner = "customer" portType = "travelPlanPT" operation = "
sendTravelPlan"
      container = "travelPlanFault" faultName = "cannotCompletePlan" / >
  < /catch > ...
< /faultHandlers >
```

## 6 结束语

基于业务流程执行语言 BPEL4WS 开发分布式应用系统 TravelFlow 为开发人员带来了极大的便利, 同时也增强了系统的可维护性和易扩展性。业务流程执行语言 BPEL4WS 使由不同服务提供商提供, 在不同平台下使用不同技术的 Web 服务结合起来, 以一个功能体的形式为用户提供强大的功能。但我们也应该看到 BPEL4WS 并不完善, 作为一个正在不断发展的新规范, 它仅提供了支持基于 Web 服务的业务流程的一些基本功能, 还缺少一些其他功能, 如业务流程状态查询、业务流程的集成和递归组合等, 这些实现在目前来说尚有一定难度, 但它将作为一个很好的发展方向为人们所关注。

### 参考文献:

[ 1 ] anthosh Kumaran, Prabir Nandi. Dynamic E-business Using BPEL4WS, WS- Coordination, WS- Transaction, and Conversation Support for Web Services[ EB/OL] . http: // www. research. ibm. com/convsupport/pa- pers, 2002.

[ 2 ] EA, IBM, Microsoft. Business Process Execution Language for Web Services ( BPEL4WS) 1. 0[ EB/OL] . http: //xml. coverpages. org/ bpel4ws. html, 2002-07.

[ 3 ] rank Leymann, Dieter Roller. 在 Web 服务世界中的业务流程 [ EB/OL] . http: //www- 900. ibm. com/developerWorks/cn/webser- vices/ws-bpelwp/index..shtml, 2002-08.

[ 4 ] irmal Mukhi. 使用 BPEL4WS 的业务流程: 学习 BPEL4WS[ EB/ OL] . http: //www- 900. ibm. com/developerWorks/cn/webservices/ ws-bpelcol/part8/index..shtml, 2002-11.

### 作者简介:

刘涛, 男, 硕士研究生, 主要研究方向为计算机网络与信息技术; 高珍, 女, 博士研究生, 主要研究方向为计算机网络与信息技术; 张志浩, 男, 教授, 博士生导师, 主要研究方向为计算机网络与数据库。