

Web 服务组合标准规范的研究*

孙洁丽^{1,2,3} 龚立群^{1,2}

¹(中国科学院国家科学图书馆 北京 100080) ²(中国科学院研究生院 北京 100049)

³(河北经贸大学信息技术学院 石家庄 050061)

【摘要】 简要介绍 Web 服务组合,分析几种 Web 服务组合标准规范,包括 BPML、WSCI、WS-CDL 和 WS-BPEL,着重分析 WS-BPEL 服务组合规范。最后指出 Web 服务组合技术在数字图书馆中具有应用价值。

【关键词】 Web 服务 服务组合 WS-CDL BPML WSCI WS-BPEL **【分类号】** TP393 G250

A Study on the Technology Standards of Web Services Composition

Sun Jieli^{1,2,3} Gong Liqun^{1,2}

¹(National Science Library, Chinese Academy of Sciences, Beijing 100080, China)

²(Graduate University of Chinese Academy of Sciences, Beijing 100049, China)

³(College of Information & Technology, Hebei University of Economics and Business, Shijiazhuang 050061, China)

【Abstract】 This paper introduces Web Service composition technologies, and analyzes the major Web Service composition standards including WS-CDL, BPML, WSCI and WS-BPEL. Then it focuses on discussing WS-BPEL. At last the application of Web Service compositions technologies in digital library are pointed out.

【Keywords】 Web Services Service composition WS-CDL BPML WSCI WS-BPEL

1 引 言

Web 服务是一种部署在网络上的应用程序。它的出现代表了分布式计算的最新要求,从而实现信息的共享、数据交换和系统集成。近年来 Web 服务技术得到快速发展和应用,成为实现互操作的一种主要机制,得到产业界和学术界的广泛认可。

单个的 Web 服务功能单一,不能提供完整的解决方案,为了实现复杂的业务逻辑,就必须对 Web 服务进行组合和集成,Web 服务只有通过组合成为更大粒度的服务,才能充分发挥 Web 服务的潜力和作用。Web 服务组合的目标就是解决 Web 服务之间的集成和协作问题,服务组合现已成为国内外工业界和学术界的研发热点。

2 服务组合标准规范及其发展

Web 服务技术的核心标准 SOAP(Simple Object Ac-

cess Protocol)、WSDL(Web Service Definition Language)和 UDDI(Universal Description, Discovery and Integration)都是基于 XML 和 Internet 技术的,并且是构建 Web 服务的基础。Web 服务组合技术的标准是建立在 Web 服务技术的核心标准之上的。由于 Web 服务组合的基础是对服务逻辑(包括服务过程、包含的基本服务、基本服务之间的交互机制等)的描述,所以人们开发了一系列服务组合标准规范,支持对服务逻辑进行有效和规范的描述。

Web 服务组合技术的标准规范包括 WSFL(Web Service Flow Language)、XLANG、BPML(Business Process Modeling language)、WSCI(Web Service Choreography Interface)、WS-CDL(Web Services Choreography Description Language)和 WS-BPEL(Web Services Business Process Execution Language)等。

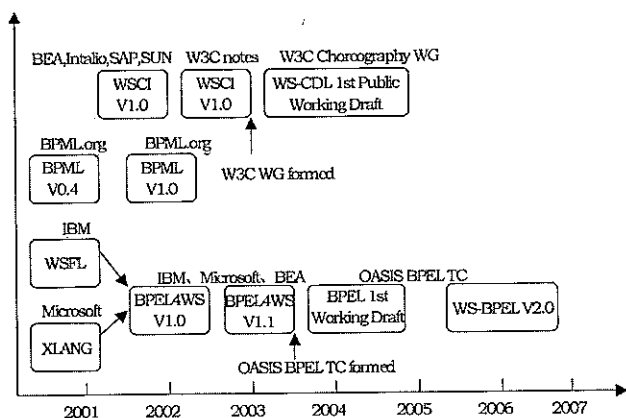
2.1 服务组合技术演变过程

Web 服务组合标准规范在近几年的发展过程可以用 Web 服务组合标准的演化过程图描述,如图 1 所示。

2002 年 6 月, BEA、Intalio、SAP 和 Sun 发布了基于 XML 的 Web 服务协作接口 WSCI。WSCI 描述了在特殊

收稿日期: 2007-03-30

* 本文系国家科技基础条件平台建设项目“我国数字图书馆标准与规范建设”(项目编号:005DKA43503)的研究成果之一。

图1 Web 服务组合标准的演化过程图^[1]

流程中通过 Web 服务实现消息流的交流,并描述了集合性信息在互动的 Web 服务间的交流,提出了一种涉及到多种 Web 服务的复杂流程的观点。随着参与 WSCI 的几个厂商如 BEA、SAP 和 SUN 加入到 WS-BPEL 的制定工作中,WSCI 基本上没有了发展空间。WS-CDL^[2]是通过万维网联盟(Word Wide Web Consortium,W3C)标准过程来实现其功能的,它将多服务交互作用提供更为整体的编排组合。尽管 WS-CDL 现在还不成熟,工具也欠缺,但 WS-CDL 有可能成为下一代 SOA 中服务编排组合的关键技术。WS-CDL 和 WSCI 属于 Web 服务业务协作层次的集成,BPML 利用 WSCI 来描述公用接口和编排方法。目前,由于推动以上标准制定工作的主要参与者基本都已加入了 WS-BPEL 标准规范的制定工作,所以,WS-BPEL 标准规范的发展最为迅速。

WS-BPEL 由 WSFL 和 XLANG 演变而来。WSFL 是由 IBM 提出的,运用流程将多个 Web 服务组合成新的具有工作流性质的 Web 服务,并且支持业务过程描述的 XML 语言。XLANG 是由微软(Microsoft)提出的基于 Web 服务的业务过程自动化规范,定义了参与业务过程的各 Web 服务之间行为的消息交换准则。2002 年 8 月,IBM、Microsoft 和 BEA 联合提交发布了 BPMLAWS(Business Process Execution Language for Web Services)的第一个版本,以取代 WSFL 和 XLANG。2003 年 4 月,为了进行 BPMLAWS 的标准化工作,OASIS 组建了专门的技术委员会。2003 年 5 月,SAP 和 Siebel Systems 加入并共同推出 BPMLAWS 1.1 版。其后,BPMLAWS 改名为 WS-BPEL,SUN 和 ORACLE 也加入了制定 WS-BPEL 标准的行列。随着众多参与者的加入,WS-BPEL 得到了很大的发展。目前,WS-BPEL 2.0 的草案已被提交 OASIS 等待批准。

2.2 BPML

BPML^[3,4]是由 BPMI.org(Business Process Manage-

ment Initiative)所发展的一个 Web 服务组合标准,最初是用于建立基于标准的 e-Business 过程管理,目前主要应用于 EAI 和 Web 服务组合。BPML 包括对控制流、数据流和事件流的全面支持。它利用以下组件来进行服务组合:过程(Processes)、活动(Activities)、上下文(Contexts)、特性(Properties)和信号(Signals)。BPML 从 2002 年 11 月发布以后再没有较大的进展。

一个过程是一个复杂活动。一个过程如果能独立于其它过程而定义,被称为顶层过程(Top-level Process);一个过程如果被定义在其它过程中执行,被称为嵌套的过程(Nested Process)。此外,BPML 中还有两种特殊过程:例外过程(Exception Processes)和补偿过程(Compensation Processes)。

活动是执行特殊功能的组件,若干个活动组成一个过程。BPML 定义了简单活动(AtomicActivity)和复杂活动(ComplexActivity),简单活动是一种元活动,而复杂活动能拆分成更小的简单活动。上下文在 BPML 中非常重要,它定义了相关活动执行的环境,可以用来交换信息和调整活动执行方式。

特性是用来交换信息的,它只存在一个上下文中。一个特性定义有一个名称和一个类型,每个特性实例都有一个对应定义类型范围内的值。

2.3 WSCI

WSCI^[5]是由 BEA Systems 和 BPMI.org 等在 2002 年 6 月提交给 W3C,作为一种基于 XML 的接口描述语言,描述了一个 Web 服务与其它 Web 服务组合时所导致的被交换的消息流。通过复用被定义为一个静态接口的操作,WSCI 描述了参与一个给定消息交换的 Web 服务的动态接口。

WSCI 在 2002 年 8 月作为一个草案递交给 W3C。此后 W3C 成立了一个被称为 Web 服务工作组的新工作组开展 Web 服务活动。WSCI 规范是 WS-CDL 形成的基本输入之一,WS-CDL 是 W3C 工作组当前的运行基础。

WSCI 位于 WSDL 的上层,它定义了包括 WSCI 活动在内的 Web 服务操作。动作经常用于定义一个基本的请求和回复消息。外部服务通过调用来实现。WSCI 也支持约束与例外情况的处理。WSCI 对一个 Web 服务看得见的行为的描述,是根据在被交换消息、特征化先后顺序规则、相互关系、例外处理和事务处理中临时的和逻辑的依赖性来表示的。WSCI 也描述了相互作用的 Web 服务中集体的消息交换。因此,它提供了一个全局的、面向消息的相互作用观。

WSCI 不从事实际驱动消息交换的内部过程的定义

和实现。更确切地说,WSCI的目标是通过面向接口的消息流来描述一个Web服务看得见的行为。这种描述使得开发者、构建者和工具通过理解与Web服务的相互作用,从而能描述和组成动态消息交换的全局观。

WSCI与BPML两者之间存在实质性的重叠。WSCI支持的简单活动有:call、delay、empty、fault和spawn,支持的复杂活动有:all、choice、foreach、sequence、switch、until和while。WSCI是对BPML的一种补充。WSCI定义了多个系统部署的服务之间的相互作用,而BPML定义了每个服务后面的业务过程并把业务活动映像到了消息交换。

2.4 WS-CDL

WS-CDL^[2]是一种基于XML的Web服务组合编排语言,通过定义Web服务参与者的一般行为,描述了Web服务参与者之间P2P的合作。WS-CDL不是一种可执行的业务过程描述语言,因此,它不依赖于任何特殊的业务过程执行语言和支撑平台,能用来定义各种类型Web服务参与者之间的可互操作的P2P合作,使参与者各自自由完全不同的语言来实现。第一个WS-CDL规范说明在2004年4月发布。目前,WS-CDL只是W3C提出的一个工作草案。

在WS-CDL描述的主体——流程编排模型(Choreography Model)中,一个服务组合中有若干个参与者(Participants),它们各自在这个服务组合中担当一定的角色(Roles),而这些角色之间存在一定的关系(Relationships)。一个参与者可能担当两个或更多不同的角色,因而与担当其它角色的其它参与者之间构成了复杂的关系。

WS-CDL描述服务组合的基本工具是组合流程(Choreography),每个流程由一个基本流程(Base Choreography)和可能的若干个子流程(Sub-choreographies)组成,描述业务过程之间的合作过程。一个流程包含一个或若干个工作单元(Work Units),实际完成流程规定的工作任务。每个工作单元中必须包含一个(且只包含一个)活动(Activities),活动定义执行实际的操作逻辑,包括:交互(Interaction),执行参与者间的消息交换,可以是单向方式(One-way),也可以是请求/应答方式(Request/Response);执行(Perform),执行一个已被定义的流程;赋值(Assign),在一个角色中给一个变量赋值;暂停(No Action),在该流程点不采取任何行动。

控制结构(Control Structures)则把各种活动按照服务逻辑排序,说明流程中各个活动的顺序关系,包括线性顺序(Sequence)、并行(Parallel)和选择(Choice)。控制结

构可以嵌套。

WS-CDL定义了一系列工具来描述有关信息,例如类型(Types),描述在一个流程中使用的信息的类型;变量(Variables),用以捕获一个流程中关于对象的信息,有信息交换变量(Information Exchange Variables)、状态变量(State Variables)和通道变量(Channel Variables)之分;记号(Tokens),一个变量或消息中的一块数据的别名。

为了处理例外情况,WS-CDL还设置了流程恢复机制,它由流程例外模块(Choreography Exception Block)和流程终结模块(Choreography Finalizer Block)构成。流程例外模块描述了当一个流程遭遇故障(Faults)、以至无法继续执行时应该采取的恢复行为;流程终结模块描述了当一个流程遭遇故障,必须抵消前面已成功完成的流程时所应采用的逆向终结行为。

2.5 WS-BPEL

由IBM、BEA和Microsoft共同推出Web服务的业务过程执行语言BPEL4WS1.0版,2003年4月6日交给OASIS组织审查并正式公布1.0版本,并且成立BPEL4WS技术委员会,决定未来规范的发展方向。该委员会致力于产生一种用于编写Web服务控制逻辑的,独立于平台的、基于XML的编程语言。2003年5月3日由BEA、IBM、Microsoft、SAP及Siebel Systems五家公司完成BPEL4WS1.1^[6]的版本。

BPEL4WS是为组合Web Services而制定的规范标准。本质上来说,是IBM公司WSFL和Microsoft的XLANG的结合,WSFL支持图形化的流程,微软的XLANG在结构化构造方面有独到的方法,BPEL4WS吸取两者的优点,同时摒弃复杂繁琐的部分,形成较为自然的描述业务流程的抽象程度较高的流程描述语言。BPEL4WS语言规范中没有任何与底层网络协议相关的部分,所有的信息传输都是基于SOAP协议来完成的。

BPEL4WS1.1版本推出后,技术委员会又对BPEL4WS1.1做了大量的修改工作,使之得到全面的提高。随后,推出Web服务业务流程执行语言2.0规范草案,把BPEL4WS语言改称为WS-BPEL^[7](Web Services Business Process Execution Language)。根据网上公示得到的评论以及反馈回来的广泛的意见,技术委员会又对WS-BPEL规范进行了进一步的修改。2007年1月,WS-BPEL技术委员会把经过批准修改后的规范作为委员会草案,提交给OASIS组织,要求批准成为正式的OASIS标准。下面重点讨论WS-BPEL。

3 WS-BPEL技术规范

WS-BPEL是为组合Web服务而制定的一项规范,它

能够描述由 Web 服务参与的复杂业务流程,同时又能将 Web 服务组合而进一步包装成为更高级别的 Web 服务并发布出去。WS-BPEL 定义了用来描述基于流程及其相关方之间互操作的业务流程行为的模型和语法。WS-BPEL 流程定义了这些相关方之间的多重服务互操作是怎样被协调起来达成业务目标的,以及这种协调所需的逻辑和逻辑。

3.1 WS-BPEL 活动介绍

业务流程通过 `<receive>` 活动和相应的 `<reply>` 活动把服务提供给它的伙伴。`<receive>` 活动是让业务流程等候匹配信息到来。`<receive>` 活动指定了它期望从哪个伙伴那里接收,还指定了它期望伙伴调用的端口类型和操作。

```
<receive partnerLink = "ncname" portType = "qname" operation =
  "ncname" variable = "ncname"? createInstance = "yes|no"?
  standard - attributes >
  standard - elements
  <correlations >?
    <correlation set = "ncname" initiate = "yes|no"? > +
  </correlations >
</receive >
```

`<reply>` 活动被用来发送对先前通过 `<receive>` 活动被接受的请求的响应。`<reply>` 活动是让业务流程在内部信息活动,即 `<receive>`、`<onMessage>` 或 `<onEvent>` 收到信息后,发出一条信息作为回答。

```
<reply partnerLink = "ncname" portType = "qname" operation =
  "ncname" variable = "ncname"? faultName = "qname"?
  standard - attributes >
  standard - elements
  <correlations >?
    <correlation set = "ncname" initiate = "yes|no"? > +
  </correlations >
</reply >
```

限于文章篇幅,下面仅介绍 3 项活动的功能。

(1) 基本活动。`<invoke>` 活动是让业务流程在相关方给出的 `portType` 上调用单向操作或请求-应答操作。`<assign>` 活动是用于使用新数据更新变量的值。`<validate>` 活动用于使针对其相关的 XML 和 WSDL 数据定义的变量值生效。`<throw>` 活动用于在业务流程内部生成故障。`<wait>` 活动用于等候一段给定的时间,或是等到达到某一特定时间点。`<empty>` 允许您在业务流程中插入“no-op”指令。`<exit>` 活动是用于当某业务流程实例中包含 `<exit>` 活动时,立即终止它。`<rethrow>` 活动是用来将原来被立即装入的 `faultHandler` 抓到的故障再次弃除。`<extensionActivity>` 元素是用来通过引入新的活动类型来扩展 WS-BPEL。

(2) 结构化活动。`<sequence>` 活动是用来定义一组活

动按词法顺序执行。`<if>` 活动是用来在一组选择中确切地选出一个活动来执行。`<while>` 活动是用来定义当指定的条件为真时重复子活动。`<repeatUntil>` 活动是用来定义重复子活动,直到指定条件为真。`<repeatUntil>` 活动至少用来执行子活动一次。`<forEach>` 活动将子范围活动恰好重复 $N + 1$ 次。`<pick>` 活动是用来阻塞等待某一个合适的消息的到达或超时警报响起。`<flow>` 活动允许指定一个或多个被并行执行的活动。`<scope>` 活动是用来定义嵌套活动。

(3) 补偿活动。`<compensateScope>` 活动是用来对一个已经成功完成的指定的内部范围启动补偿。`<compensate>` 活动是用来按照默认的顺序,对所有已经成功完成的内部范围启动补偿。

3.2 伙伴链接 (Partner Links)、伙伴链接类型 (Partner Link Types) 和终端引用 (Endpoint References)

业务流程与伙伴的关系通常是对等的,这需要在服务级别上有双向的相关性。换句话说,伙伴表示由业务流程提供的服务的消费者,同时,对于业务流程来说,伙伴又表示服务的提供者。伙伴链接这个概念被用来直接模拟对等伙伴关系。为了定义与伙伴的关系形式,伙伴链接定义了交互的两个方向上用到的消息和端口类型。然而,实际的伙伴服务可以在流程中被动态地确定。为了表示描述伙伴服务终端所需的动态数据,WS-BPEL 定义了终端引用这个概念。

(1) 伙伴链接类型。为了描述两个服务间的关系,伙伴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`)。如果伙伴链接类型声明链接了两个不同服务的 `portType`,那么伙伴链接类型声明可以被放在单独的 WSDL 文档中。

下面举例说明伙伴链接类型声明的基本语法:

```
<partnerLinkType name = "BuyerSellerLink"
  xmlns = "http://schemas.xmlsoap.org/ws/2003/05/partner-link/" >
  <role name = "Buyer" >
    <portType name = "buy:BuyerPortType"/ >
  </role >
  <role name = "Seller" >
    <portType name = "sell:SellerPortType"/ >
  </role >
</partnerLinkType >
定义伙伴链接类型的语法如下:
<definitions name = "ncname" targetNamespace = "uri"
  xmlns = "http://schemas.xmlsoap.org/wsdl/"
  xmlns:plnk = "http://schemas.xmlsoap.org/ws/2003/05/partner-link/" >
  ...
  <plnk:partnerLinkType name = "ncname" >
```

```

<plnk:role name = "ncname" >
  <plnk:portType name = "qname"/>
</plnk:role >
<plnk:role name = "ncname" >?
  <plnk:portType name = "qname"/>
</plnk:role >
</plnk:partnerLinkType >
...
</definitions >

```

(2) 伙伴链接。在 WS-BPEL 中,与业务流程交互的服务被模拟成伙伴链接。伙伴链接由 partnerLinkType 来描述。同一个 partnerLinkType 可以描述多个伙伴链接。例如,某个采购流程可以在它的事务中使用多个供应商,但是对于所有的供应商都使用相同的 partnerLinkType。

```

<partnerLinks >
  <partnerLink name = "ncname" partnerLinkType = "qname"
    myRole = "ncname"? partnerRole = "ncname"? > +
</partnerLink >
</partnerLinks >

```

当一个伙伴链接表达两个伙伴流程之间的会话关系时,通常业务伙伴的关系需要建立多个会话关系。为了表示业务伙伴的性能,WS-BPEL 使用 partner 元素。一个 partner 被定义为流程的伙伴链接的一个子集。定义伙伴的语法如下:

```

<partners >
  <partner name = "ncname" > +
    <partnerLink name = "ncname"/> +
  </partner >
</partners >

```

(3) 终端引用。终端引用的基本用途是作为一种机制,用于服务的特定于端口的数据的动态通信。可以使用终端引用在 WS-BPEL 中动态地为某个服务类型选择提供者和调用它们的操作。

3.3 WS-BPEL 特点

(1) 从总体上看,WS-BPEL 2.0 的可扩展性变得更增强了,WS-BPEL 2.0 加入了新的活动,并且在流程中增加了新的属性,使得流程的表达更加灵活,此外,一些元素包括属性的命名也更加规范。

(2) 在修改业务流程和 Web 服务时不会影响应用程序中的其他业务流程或 Web 服务。

(3) 应用程序的开发和测试可以分为两个阶段,业务流程的开发和测试与单个 Web 服务的开发和测试是分开的。在创建和修改应用程序时,这种方法提供了很大的灵活性。

(4) WS-BPEL 可以在不影响应用程序自身的情况下调整已部署的应用程序,能够快速适应应用环境变化的需要。

4 结 语

数字图书馆作为由众多分布、异构和自主的资源与服务组成的开放环境,需要灵活适应具体问题环境或具体业务流程,动态发现、调用和组合相关资源与服务,提供能满足用户需求的服务。Web 服务技术为数字图书馆提供了一种公共机制,在开放环境下发现和调用所需要的资源或服务。但是,用户所需要的服务可能不能直接为某个 Web 服务所满足,这时,需要利用 Web 服务组合技术,将若干 Web 服务按照一定逻辑组合后满足用户需求。因此,Web 服务组合技术对数字图书馆具有重要的应用价值^[8,9]。

参考文献:

- 1 Cheng Yushi, Lee Eng Wah, Dilip Kumar Limbu. Web Services Composition - An Overview of Standards. http://www.itsc.org.sg/synthesis/2004/4_WS.pdf. (Accessed Feb. 3, 2007)
- 2 Nickolas Kavantzias. et al. Web Services Choreography Description Language Version 1.0. <http://www.w3.org/TR/ws-cdl-10/> (Accessed Feb. 3, 2007)
- 3 Business Process Modeling Language, 1.0, June 2002. http://www.ebpm.org/bpml_1_0_june_02.htm (Accessed Feb. 3, 2007)
- 4 W. M. P. van der Aalst. et al. Pattern Based Analysis of BPML (and WSCI). http://tmitwww.tn.tue.nl/research/patterns/download/qt_bpml_rep.pdf (Accessed Feb. 3, 2007)
- 5 Arkin, A. Web Service Choreography Interface (WSCI) 1.0, W3C Note 8 August 2002. <http://www.w3.org/TR/wsci> (Accessed Feb. 3, 2007)
- 6 Business Process Execution Language for Web Services Version 1.1. <http://xml.coverpages.org/BPEL11-May052003Final.pdf> (Accessed Feb. 3, 2007)
- 7 Web Services Business Process Execution Language Version 2.0 Committee Draft, 25 January, 2007. <http://www.oasis-open.org/committees/download.php/22036/wsbpel-specification-draft%20candidate%20CD%20Jan%2025%2007.pdf> (Accessed Feb. 3, 2007)
- 8 张晓青, 张晓林. 数字图书馆服务的动态定制机制. 图书情报工作, 2005, 49(5): 26-29
- 9 张晓青, 相春艳. 基于 Web 服务组合的数字图书馆个性化动态定制服务构建. 情报学报, 2006, 25(3): 337-341

(作者 E-mail: sunjl@mail.las.ac.cn)