

## 数字图书馆服务的动态定制机制

张晓青(烟台师范学院图书馆, 山东烟台, 264025)

张晓林(中国科学院文献情报中心, 北京, 100080)

文摘: 本文描述 WEB 服务组合的基本逻辑过程, 并对基于 WEB 服务组合的数字图书馆动态定制服务的构建机制进行了讨论, 包括基本系统架构、服务组合生命周期、基本组合模型以及组合工具。

关键词: 数字图书馆 Web 服务 WEB 服务组合 动态定制

### Web Service Composition for Dynamic Customization of Digital Library Services

Zhang Xiaoqing ()

Zhang Xiaolin (Library of Chinese Academy of Sciences, Beijing, 100080)

Abstract: The paper introduces the basic process of web service composition, and describes the implementation mechanisms for dynamical customization of digital library service bases on web service composition, including system structure, service composition life-cycle, basic composition models, and composition languages.

Keywords: Digital library, Web Services, Web Service Composition, Dynamic Customization

#### 1. 动态定制服务

##### 1.1 基于 Web Services 的动态定制服务

Web Services 技术<sup>[1]</sup>为数字图书馆系统提供了一种公共机制, 在开放环境中发现和调用所需要的资源或服务, 其过程可以简单描述为:

(1) 各种数字资源与服务系统作为开放环境下的 WEB 服务, 利用标准的开放语言(例如 WSDL<sup>[2]</sup>)对自己的服务内容、操作类型、请求与应答消息流、系统绑定方式等进行规范描述, 所生成的描述信息(即关于服务系统的元数据)被注册到网络服务公共登记系统(例如 UDDI<sup>[3]</sup>), 提供公共查询和调用。

(2) 用户(通过用户代理或者数字图书馆系统)利用同样的标准开放语言来描述自己所需要的服务, 查询网络服务公共登记系统, 发现与自己需求匹配的网络服务。

(3) 用户(同样通过用户代理或者数字图书馆系统), 利用标准的网络服务调用协议(例如 SOAP<sup>[4]</sup>)调用相应的网络服务, 与自己系统绑定, 作为自己系统的一个(可能是临时的)有机部分来提供所需要的服务。

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, 需要将若干 WEB 服务按照一定逻辑组合后才能实现, 这要求利用现有 WEB 服务定制新的复杂 WEB 服务。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, 需要为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定制服务<sup>[5]</sup>。因此, 上述问题对数字图书馆具有重要意义。

##### 1.2 基于 WEB 服务组合的动态定制服务

Web 服务组合(Web Service Composition<sup>[6-7]</sup>)技术提出了可能的解决方案, 它根据用户需求, 通过一定的组合逻辑, 将现有的基本 WEB 服务或服务组件动态集成为新的 WEB 服务(图 1):

(1) 服务请求者通过规范形式描述自己的服务需要;

(2) 服务请求者通过 WEB 服务公共登记系统查询是否有自己需要的基本服务存在, 如果有, 直接调用这个基本服务;

(3) 如果没有, 按照特定服务逻辑, 解析服务需要, 分解为对应的若干基本服务;

(4) 按照规范形式描述这若干个基本服务的相互关系(涉及内容、过程、交互机制等);

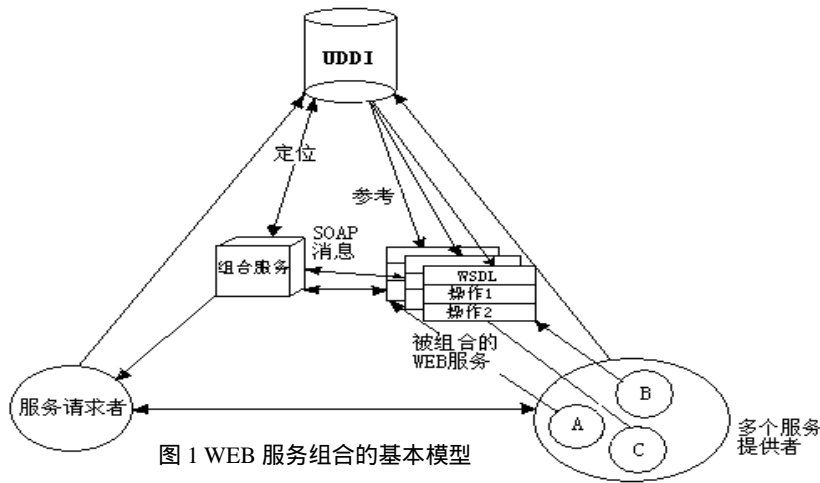


图1 WEB 服务组合的基本模型

(5) 通过 UDDI 发现相关的基本服务, 例如由 A、B、C 提供的基本服务;

(6) 按照服务逻辑描述规范, 调用这些基本服务, 并根据服务组合的生命周期进行动态集成, 从而生成符合用户需求的组合服务, 即用户的定制服务;

(7) 这个新生成的服务也会在 UDDI 登记, 以备查询、调用或被组合。

## 2. WEB 服务组合的基本组合机制

### 2.1 WEB 服务组合的基本体系

根据 1.2 所述过程中不同阶段的实现机制, 可以将 WEB 服务组合分为静态组合 (Static composition) 和动态组合 (Dynamic composition), 前者按照事先确定的业务流程组合多个基本 WEB 服务, 后者则根据用户服务目标来确定业务流程以及要组合的基本 WEB 服务<sup>[7]</sup>。对于静态 WEB 服务组合, 大体有两种做法, 一是直接将多个 WEB 服务通过专门程序或系统组织成一种新的集成服务, 例如我们常见的跨库检索系统; 二是通过业务流程描述语言定义相应的业务流程、这个流程所涉及的多个 WEB 服务、这些 WEB 服务之间的关系以及关系调整控制机制, 例如采用 BPEL4WS<sup>[8]</sup>。对于动态 WEB 服务组合, 也有多种方法, 一是建立服务需求的逻辑模型 (可以利用模型化语言如 UML<sup>[9]</sup>来进行描述), 并根据这个模型去定义服务目标、过程、活动、条件、限制以及对应的 WEB 服务和它们的组合控制; 二是利用 WEB 服务请求与计划机制 (例如 XSRL<sup>[10]</sup>) 来定义、规划和组合多个 WEB 服务来动态完成复杂任务; 还可以采用 Semantic Web Services 技术在 WEB 服务组合机制中嵌入更丰富的语义内容, 以支持根据语义的分析、规划和组合<sup>[6]</sup>。

### 2.2 WEB 服务组合的逻辑集成机制

从业务逻辑上讲, WEB 服务组合有两类组合类型, 一是系统集成 (System Integration), 一是商业逻辑构建<sup>[11]</sup>。系统集成方式不生成任何新的业务逻辑, 组合生成的新服务实际上是各 WEB 服务的简单综合。图 2 代表了系统集成, 两个虚线框分别代表系统 1 及其业务过程和系统 2 及其业务过程, 而中间的实线框, 则代表系统集成过程, 它只是简单地把两个不同系统的业务过程连接在一起, 并没有产生新的商业逻辑。

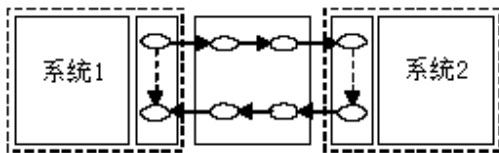


图2 系统集成的组合机制

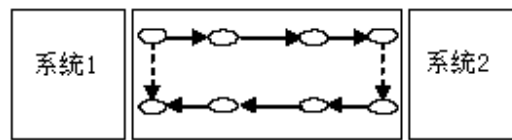


图3 商业逻辑构建的组合机制

图 3 表示商业逻辑构建方式, 必须根据用户的需求构建新的商业逻辑, 服务组合生成的新服务是不同系统有用组件按照新的业务流程重新整合后产生的新服务, 形成了新的独立的业务过程。在商业逻辑构建中, 往往在对服务需求、商业逻辑、组合机制、组合管理、组合服务执行管理等方面采用开放的标准语言进行定义, 支持开放环境中的 WEB 服务动态组合。当然, 不排除许多 WEB 服务组合会将在不同程度上结合运用系统集成和商业逻辑构建方法。

## 2.3 WEB 服务组合的构件组合

WEB服务组合中组合对象(构件)的粒度是可变的。WEB服务组合中,很多时候不需要或不能够把两个或若干个WEB服务系统整体地组合在一起,因为每个WEB服务系统都是由若干个相对独立的子系统构成,这些子系统又可能建立在更次一级的资源组件、应用组件、功能组件和管理组件上。

WEB服务组合的构件块可以是基本服务、可继承性组件,也可以是由基本服务或可继承性组件构成的服务组件<sup>[12][13]</sup>。基本服务是外部用户可以引用的最小服务单位。基于XML的WSDL是描述基本服务接口最普遍使用的语言。基本服务的WSDL接口,使得现有WEB服务可以通过BPEL4WS等组合语言来进行WEB服务组合。可继承组件指那些只支持内部商务过程且不以WEB服务方式发布的固有软件组件(own software component)。可继承组件不具有WSDL接口,不能利用PEL4WS进行集成组合,但是一种面向过程的可视化服务组合语言UML-WSC提供了解决方案<sup>[9]</sup>。服务组件是一种模块化的应用,它将服务接口和相关的商业逻辑捆绑到了一个密不可分概念模型中,可以看成是一个高层次独立的组合服务,通过模块化合成相关服务的功能使Web服务复用、扩展、专业化与服务继承<sup>[12]</sup>。一个服务组件主要由两部分构成:接口规范与

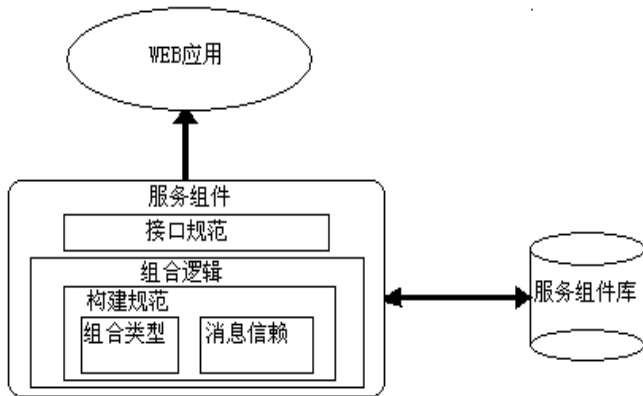


图4 服务组件的构成

组合逻辑,如图4。服务组件给外部世界提供了一个公共接口,这个公共接口是由服务组件元服务(constituent service)的功能签名(operation signatures)、消息类型(message types)和端口类型(portTypes)的统一表现所构成。一个服务组件含有一个组合逻辑(composition logic)部件,组合逻辑部件在内部详细规定了服务组件是如何根据组合类型(composition type)和消息依赖(message dependency)由元服务构成的,从而规定元服务如何才能同步协调。

服务组件中的组合逻辑部件本身包括以下两个构件:

(1) 组合类型: 组合类型表示组合性质,可以采取下列两种形式之一: 次序(order),表示元服务采取顺序方式执行还是采用并行方式执行。可替代服务执行(Alternative service execution),表示能否在一个给定的服务组合中调用可替代服务,而可替代服务可能以顺序方式或并行方式试验。

(2) 消息依赖(information dependency): 表示元服务间的消息依赖关系类型以及元服务与其周围服务组件构件间的消息依赖关系类型,包括三种必要的消息依赖处理机制: 消息合成(message synthesis)、消息分解(message decomposition)和消息映射(message mapping)。

## 3. 基于 WEB 服务组合的动态定制生命周期

### 3.1 WEB 服务组合生命周期

按照某种组合逻辑进行WEB服务组合时,经历了几个渐进阶段,构成服务组合的生命周期:

(1) 计划编制阶段(planning): 根据用户提出的服务请求确定需要检索和聚合的服务操作(或者活动)系列,这个服务请求和计划活动是根据领域模型提供的信息来执行;

(2) 定义阶段(definition): 利用WSDL与BPEL4WS对组合服务进行抽象定义;

(3) 调度阶段(scheduling): 确定各种元服务将会怎样执行、什么时候执行,并为执行这些元服务进行准备,同时提出可替代的组合调度并将其提供给应用开发者进行选择;

(4) 构建阶段(construction): 构建一个具体而明确的服务组合,这个服务组合由一系列想得到的或潜在有效的相匹配的元服务构成;

(5) 执行阶段 (execution) : 在确定服务组合规范的基础上, 进行组合服务绑定, 并执行这些服务。

### 3.2 动态组合模型

对于不同的动态定制服务来说, 服务组合过程的细节可能不一样, 但基本生命周期是相同的。因此, 有研究者<sup>[13]</sup>根据WEB服务组合生命周期提出了动态组合模型, 如图5。

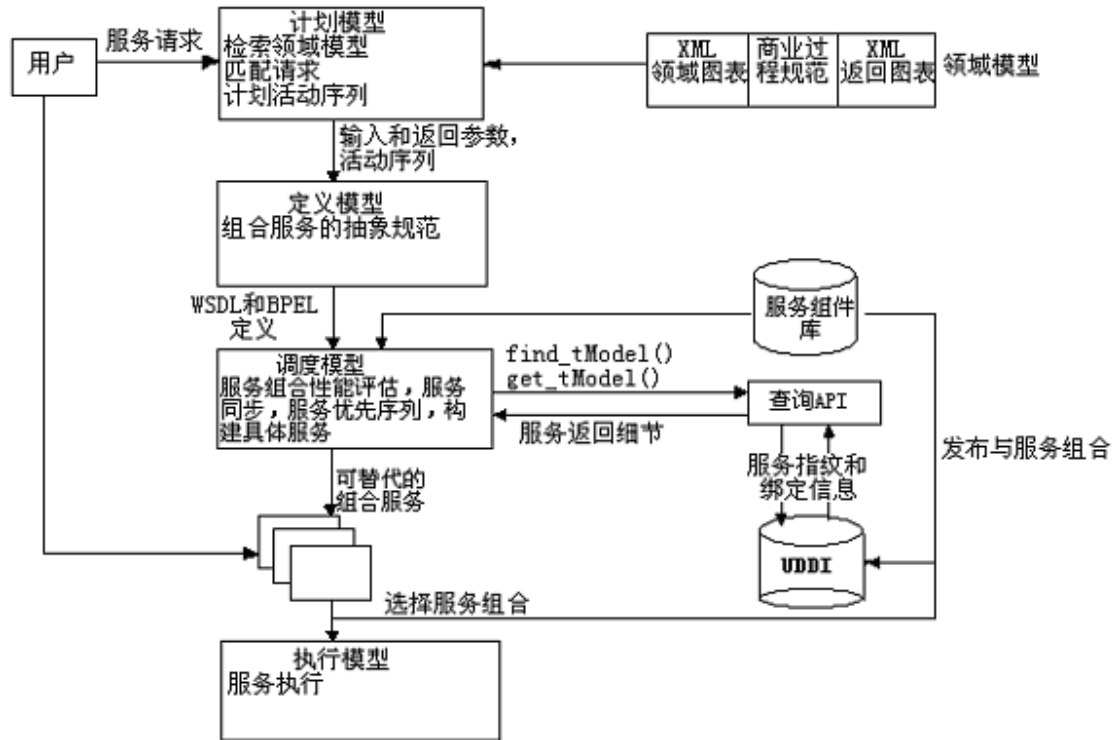


图5 基于服务组合生命周期的动态定制构建模型

在图5描述的过程中, 首先由客户提出一个服务请求, 然后计划模型对领域模型进行检查。计划模型通过发现可潜在地满足客户请求的活动路径 (服务操作) 来检查客户请求与业务过程规范的一致性。如果这个请求与服务领域模型规范一致, 将返回活动计划。随后, 服务定义模型为已计划好的活动序列构建抽象的WSDL和BPEL服务定义, 并把这些抽象服务定义传递给调度模型。在调度模型中, 调度首先需要与服务提供者交互以调用抽象定义详细说明的服务操作。在这个过程中, 调度须恰当地调用UDDI查询API来使抽象定义具体化。为此, 调度必须通过查询UDDI操作去发现和获得UDDI的API细节, 以检索关于服务端口类型、服务元素和服务绑定的详细信息。其次, 调度为获得所要求的服务, 要检索UDDI并使用在UDDI登记系统中发现的信息去建立特殊的调用模式, 并从服务组件库中调用服务组件或通过UDDI提供的信息去调用某些服务。再次, 调度要使各个元服务相关联并检查其兼容性。当这些操作完成后, 一些可替代的服务组合就会提交给客户去选择和批准。最后, 一旦选择的服务被确定, 它们就会存储在一个服务登记系统以备将来使用, 同时会传递给执行模型执行。

### 4. WEB服务组合中的动态定制基本组合模型

WEB服务组合根据被组合服务的相互作用类型, 提供了二种基本的动态组合模型<sup>[14]</sup>: 协作模型 (cooperative model) 和掮客模型 (brokered model), 分别如图6a、图6b。

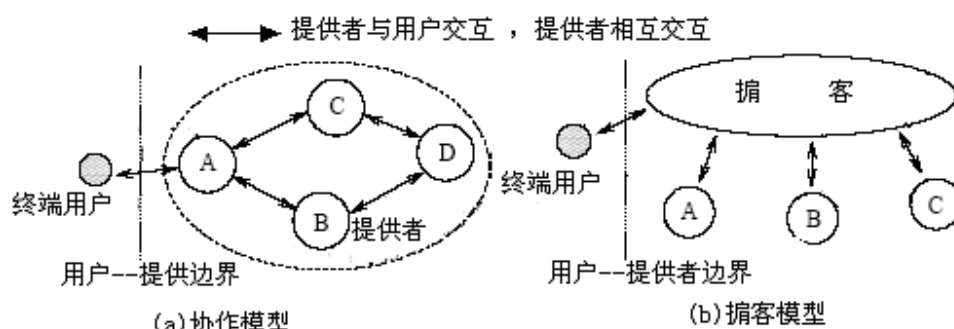


图6 基于服务提供者作用类型的动态定制服务模型

#### 4.1 协作模型

在协作模型中，不同的服务提供者相互依赖和支持。当用户向协作模型中的任一提供者提出服务请求时，接受请求者将根据用户的个性化需求向其它服务提供者发出协作服务请求，然后将协作服务者提供的服务按照用户的要求组合在一起提供给用户。在这个过程中，用户只与一个服务提供者发生关系，其它服务提供者呈分布状态，并且一起以分布负责方式向终端用户提供一个端对端组合服务。在协作模型中，组合服务的特性（如功能性、性能、有效性等）是通过分布式相互作用的设计和交互实体间服务层次的协议来保证的，每个服务提供者仅对它所在领域的组合服务部分提供服务质量保证。

在协作模型中，组合服务的特性（如功能、性能、有效性等）通过分布式相互作用的设计和交互实体间服务层次的协议来保证的，每个服务提供者仅对它所在领域的组合服务部分提供服务质量保证。例如，我们假定各数字图书馆系统的各项功能、资源子系统及其它应用已经经过WEB服务封装，并在WEB服务登记系统登记。当某个用户进入某一数字图书馆系统时，可以利用该系统的服务请求窗口提出服务请求。然后，该系统启动用户需求分析机制，对用户的服务要求进行分析，并根据这个分析查询相匹配的数字图书馆系统或子系统，然后向相关的系统发出协作请求，并将所请求的服务组合成最终符合用户需求的定制服务提供给用户。当然，在这个过程中，不同系统彼此存在信任问题，即每个系统必须验证其它系统的服务是否符合其服务规范，是否保持了提供者所声称的理想特性（可以由提供者和请求者之间的双边服务层次协议—Service Level Agreement—详细规定）。有关研究中<sup>[14]</sup>使用参数验证和使用监测，保证SLA中规定的服务特性得到尊重。

#### 4.2 掮客模型

图6b描述了一个掮客模型。在掮客模型中，所有服务提供者均不与服务请求者发生关系并且彼此也不发生关系。服务请求者只向一个服务提供者提出服务请求，即掮客。掮客是利用底层服务提供者提供的功能性，并把这些功能性封装起来去组合端对端服务。这种方式简化了服务部署，因为组合成员之间无须彼此达成协议，它们只须与掮客达成协议即可。当然，这也使得服务组合能超越相互竞争的服务提供者。同时，掮客对整个服务组合的特性负责。

掮客机制可以利用不同的服务提供者的服务来组合新服务。这种工具对于开放环境下数字图书馆的动态定制服务具有可移植性，即人们可以建立一个专门的动态定制引擎系统，具备强有力的需求描述、业务流程分析、WEB组合构建、WEB服务发现、元数据转换、呈现机制构建等功能，可根据用户请求生成复杂但可逻辑解析的服务模型，可根据服务模型和WEB组合服务描述语言构造新的服务流程，并根据这个流程去发现、确认和组合服务以及这个服务的呈现机制，从而建立符合用户需求的定制服务。当然，因为掮客对底层提供者的服务没有透彻的了解，因此被组合的资源或服务的利用也许达不到最优化。

当然在现实世界，被组合服务者之间的相互作用关系比这更复杂，因此两种模型的组合也是可能的。比如，当用户向某个数字图书馆系统提出服务请求时，该系统现有服务不能满足用户要求，该系统会向其他系统发出协作服务请求；被请求的系统由于自己不能完成相应的服务请求，可以向数字图书馆动态定制服务掮客提出相应的协作服务请求，从而在最后动态定制服务过程中将两个模型结合起来。

## 5. 动态定制服务组合工具

WEB 服务组合的实现需要借助专门的组合工具, 目前趋势是基于 XML 和 WEB 服务的商业过程模拟语言来表达组合 WEB 服务的逻辑, 近年已有多种这类<sup>[15]</sup>, 它们包括 BPML<sup>[16]</sup>、BPSS<sup>[17]</sup>、WSCI<sup>[18]</sup>、XLANG<sup>[19]</sup>、WSFL<sup>[20]</sup>、BPEL4WS<sup>[8]</sup>等。BPEL4WS 综合了以前若干语言的优点, 表现了更强的功能, 我们将通过它简略介绍 WEB 服务组合工具的特点。

WEB 服务的商业过程执行语言 BPEL4WS 是基于 XML、用以描述只依赖 WEB 服务输入、输出的可执行商业过程的语言, 2002 年 7 月由 BEA、IBM 和 Microsoft 共同推出 1.0 版, 2003 年 5 月推出了 1.1 版, 用来替代当时已有的 WEB 服务组合语言 XLANG 和 WSFL。BPEL4WS 一方面对商业过程在抽象层次上进行说明, 另一方面也提供了可执行的过程定义语言。

BPEL4WS 中是抽象商业协议描述和可执行商业过程描述共同遵循一个公共的过程描述核心概念, 并明确地把核心概念与扩展概念 (特别是可执行商业过程模式要求的扩展概念) 分开, 主要关注公共的核心概念。通过这个核心概念, 它为基于过程与其伙伴之间相互作用的商业过程行为描述提供了一个模型和一种语法。商业过程与每个参与者之间的相互作用是通过 WEB 服务接口出现的, 接口层次的关系结构封装在伙伴链接 (partner link) 中。BPEL4WS 过程定义了多个服务怎样与这些伙伴相互作用协调地去达到一个商业目标以及这种协调的状态和逻辑必要性, 也介绍了处理商业例外和过错的系统机制, 并且介绍了当意外出现或者一个商业伙伴要求撤销服务时单个或组合活动怎样才能被补偿的机制。

从 BPEL4WS 角度, 一个商业过程的规范主要由三个部分构成: 参与商业过程的伙伴 (Partners), 与控制流对应的活动 (Activities), 装载消息和数据的变量 (Variables, 在 BPEL4WS 1.0 中采用的术语是 containers)。每个商业过程伙伴都会被赋予一定的角色, 通过伙伴链接类型来定义, 并被赋予相应的端口类型。在执行一个商业过程时, 商业过程会调用这个伙伴的端口类型所提供的操作。商业过程的伙伴们通过商业活动来实现交互和合作, 因此 BPEL4WS 提供了基本的商业活动语句, 也提供了结构化活动语句 (例如序列、选择、并发和同步等)。BPEL4WS 中的变量为装载和传送商业过程状态消息提供了手段, 这些消息包括从商业伙伴接收到的消息或是要发送给商业伙伴的消息, 当然也可以装载那些为保持与过程相关的状态所需要的且永远不会与伙伴交换的数据。每个变量的类型可以是 WSDL 消息类型、XMLS 简单类型或 XMLS 元素。

BPEL4WS 直接利用了已有的 XML 规范: WSDL 1.1, XML Schema 1.0, 和 Xpath 1.0。WSDL 消息和 XML Schema 类型定义提供了 BPEL4WS 过程使用的数据模型, Xpath 对数据处理提供了支持, 所有的外部资源和伙伴都被表示为 WSDL 服务。

WEB 服务组合工具实际上是在开放环境下进行动态定制服务的基本工具, 它能否应用于实际是能否方便实现动态定制服务的关键所在。目前包括 BPEL4WS 在内的许多组合语言, 均在商务领域的应用中取得了进展, 对于我们研究数字图书馆的动态定制服务具有重要的借鉴意义, 限于篇幅, 我们将在后续的文章中展开专门的分析。

## 参考文献:

- [1] Booth, David, et al. Web Services Architecture, W3C Working Group Note, 11 February 2004 <http://www.w3.org/TR/ws-arch/> (检索日期: 2004.06.19)
- [2] Chinnici, R. et al. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Working Draft, 26 March 2004, <http://www.w3.org/TR/wsdl20> (检索日期: 2004.06.19)
- [3] Bellwood, T. et al. UDDI Version 3.0.1, UDDI Spec Technical Committee Specification, 14 October, 2003. [http://uddi.org/pubs/uddi\\_v3.htm](http://uddi.org/pubs/uddi_v3.htm) (检索日期: 2004.06.19)
- [4] Gudgin, M. et al. SOAP Version 1.2 Part 1: Messaging Framework, W3C Recommendation, 24 June 2003. <http://www.w3.org/TR/soap12-part1/> (检索日期: 2004.06.19)
- [5] 张晓林. 开放数字服务体系: 概念、结构与技术. 中国图书馆学报, 2002 (3): 8-14
- [6] Srivastava, B. and Koehler, J. Web Service Composition: Current Problems and Open Solutions. <http://www.zurich.ibm.com/pdf/ebizz/icaps-ws.pdf> (检索日期: 2004.06.19)
- [7] Shapara, D. Approaches to Web Service composition. <http://science.unitn.it/~tomasi/think/pdf/pdmi.ppt> (检索日期: 2004.06.19)



- [8] Andrews, T. et al. Business Process Execution Language for Web Services Version 1.1, 05 May 2003, <http://www.ibm.com/developerworks/library/ws-bpel/> (检索日期: 2004.06.19)
- [9] Sebastian Thone et al. Process-Oriented, Flexible Composition of Web Services with UML. LNCS, 2003, 2784:390-401
- [10] XSRL: A Requester Language for Web Services  
<http://www.webservices.org/index.php/article/articleview/990/1/24/index.htm#section2> (检索日期: 2004.06.19)
- [11] Christoph Bussler. Semantic web services: reflections on web service mediation and composition. Proceedings of the fourth International conference on web Information systems engineering (WISE'03).
- [12] W. van der Aalst. "Don't Go with the Flow: Web Services Composition Standards Exposed". IEEE Intelligent System, IEEE Press, 2003, pp72-76.
- [13] Jian Yang, Mike. P. Papazoglou. Service components for managing the life-cycle of service compositions. Information Systems, 2004,29: 97-125
- [14] Bhaskaran.Raman et.al. The SAHARA Model for Service Composition across Multiple Providers. LNCS, 2002, 2414:1-14.
- [15] Wil M.P.van der Aalst. Web service composition language: old wine in new bottles?. Proceedings of the 29<sup>th</sup> EUROMICRO conference "new wave in system architecture"(EUROMICRO'03).
- [16] Business Process Modeling Language, 1.0, June 2002 . [http://www.ebpm.org/bpml\\_1\\_0\\_june\\_02.htm](http://www.ebpm.org/bpml_1_0_june_02.htm) (检索日期: 2004.06.19)
- [17] ebXML Business Process Specification Schema 1.1. April 2001. <http://www.ebxml.org/specs/ebBPSS.pdf> (检索日期: 2004.06.19)
- [18] Arkin, A. Web Service Choreography Interface (WSCI) 1.0, W3C Note 8 August 2002, <http://www.w3.org/TR/wsci> (检索日期: 2004.06.19)
- [19] Thatte, S. XLANG : Web Services for Business Process Design. 2001. [http://www.gotdotnet.com/team/xml\\_wsspecs/xlang-c/default.htm](http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm)
- [20] Leymann, F. Web Service Flow Language. 1.0. May 2001. <http://www-306.ibm.com/software/solutions/webservices/pdf/WSFL.pdf> (检索日期: 2004.06.19)