

数字图书馆服务动态定制中的组合工具

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: 本文在简要介绍基于 WEB 服务组合的服务动态定制机制后, 着重分析介绍了包括 BPML、BPSS、WSCI、BPEL4WS 等在内的 WEB 服务组合工具。

关键词: 数字图书馆 WEB 服务组合 动态定制 BPML BPSS WSCI BPEL4WS

Composition Tools for Dynamic Customization of Digital Library Services

Zhang Xiaoqing (Library of Yantai Normal University, Yantai, 264025)

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

Abstract: Based on a brief introduction to dynamic customization of digital library services with web service composition technologies, the paper analyzes and compares the major web service composition tools including BPML, BPSS, WSCI, BPEL4WS.

Keywords: Digital library, Web Service Composition, Dynamic Customization, BPML, BPSS, WSCI, BPEL4WS

1 动态服务定制与 WEB 服务组合

在一个由众多分布、异构和自主的资源系统组成的开放环境中, 数字图书馆应该根据具体的用户需要, 发现、解析和调用所需要的资源和服务, 按照个性化的服务流程和业务逻辑将这些资源或服务灵活组织起来构成新的服务, 从而实现数字图书馆服务的动态定制。WEB 服务组合 (Web Service Composition^[1-2]) 技术为此提供了基本的解决方案:

(1) 按照 Web Services 技术^[3], 可以把各种数字资源与服务看作 WEB 服务, 利用标准语言 WSDL^[4]对它们进行规范描述, 所生成的描述信息被注册到 UDDI 这类 WEB 服务登记系统^[5], 提供公共查询和调用。

(2) 用户系统利用同样的标准语言描述所需服务, 查询 UDDI, 发现与需求匹配的 WEB 服务, 利用 SOAP 协议^[6]调用这个 WEB 服务, 与自己系统绑定后作为自己系统的一个有机部分提供所需要的服务。

(3), 如果用户在 UDDI 上没有发现直接匹配的 WEB 服务, 可以按照特定的服务逻辑, 解析自己的服务需要并将其分解为若干基本服务, 利用流程组合语言描述服务逻辑过程、基本服务类型及角色、基本服务关系及其交互机制等, 形成关于动态定制的组合服务元数据。

(4) 用户通过 UDDI 发现相关的基本服务, 按照组合服务元数据调用这些基本服务, 并根据服务组合生命周期^[7]进行动态集成, 从而生成根据用户需求动态定制的组合服务。

(5) 用户可以将这个新生成的服务在 UDDI 登记, 以备查询、调用或被组合。当然, 负责生成这个组合服务的用户也有责任检验和维护这个组合的有效性。

我们在文献[8]中介绍了基于 WEB 服务组合的动态定制机制。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组合语言来支持对服务逻辑进行有效和规范的描述, 这些语言构成 WEB 服务组合的基本工具, 它们与其它 WEB 服务技术的关系可用图 1^[9]表示。

根据 WEB 服务组合语言的描述对象和描述结果的执行程度, 似乎可以将它们分为两类: 一类是对 WEB 服务要组合的服务逻辑进行抽象描述, 建

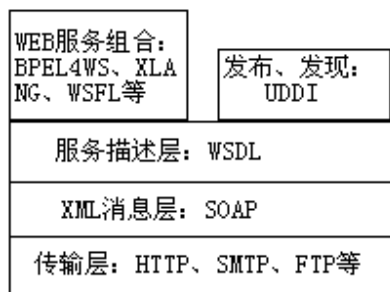


图 1 WEB 服务技术体系与 WEB 服务组合语言

立逻辑模型 (Service Modeling), 描述服务过程、服务活动、服务参与者及其角色、服务流程控制机制, 支持在此基础上的具体业务流程定制, 例如 BPSS^[10]、WS-CDL^[11]等; 一类是在对服务逻辑本身进行抽象描述的基础上, 针对业务流程, 定制执行的流程及其控制机制, 例如 BPML^[12]、WSCI^[13]、BPEL4WS^[14]等。另外还有规划组合语言, 支持对 WEB 服务组合的定义、规划和组合, 例如 XSRL^[15]。本文将对主要的 WEB 组合语言进行分析和比较。

2 BPSS

BPSS (Business Process Specification Schema)^[10]由UN/CEFACT和OASIS提出, 属于ebXML框架^[16]中的组成部分, 按照实际商务过程将所涉及的商务对象、商务环节、商务文件以及相关的限制条件组合起来, 并借助ebXML CPP^[17] (Collaboration Protocol Profiles) 描述具体商业对象及其服务要求, 借助ebXML CPA^[17] (Collaboration Protocol Agreements) 描述商业对象之间的合作协议, 从而实现商务过程组合描述。

BPSS 提供了必要的语义和方法, 可以描述一个完整的多向商业合作过程, 例如: 定义商业事务 (Business Transaction); 定义商业事务中的商业文件流 (Business Document flow); 复用商业事务定义来定义双向商业合作 (Binary Collaboration); 为双向商业合作编排合作流程 (Choreography); 复用下层双向商业合作定义来定义上层双向商业合作; 复用双向商业合作定义来定义多向商业合作 (MultiParty Collaboration)。

这其中的核心概念包括:

商业合作 (Business collaborations): 商业合作是商业伙伴之间的一组商业事务, 每个商业伙伴在其中扮演着一个或多个角色。BPSS支持双边合作和多方合作。双边合作在两个角色之间进行, 多方合作在两个以上角色之间进行, 可以看成是多个双边合作的综合。双边合作表现为两个角色之间的一组商业活动, 每个商业活动反映了商业合作的一种状态。描述商业合作的BPSS/UML元素有: MultiPartyCollaboration、BusinessPartnerRole、Performs、AuthorizedRole、BinaryCollaboration、BusinessActivity、BusinessTransactionActivity、CollaborationActivity。

商业事务: 商业事务是在事务处理中扮演相反角色 (例如请求方和应答方) 的两方间进行, 两个角色间的一个可复用的协议。商业事务是最基本的单元, 不能被拆分成更次一级的商业事务。为达到非常精确和有约束力的事务语义, 一个商业事务是一个非常专门、非常限定的协议, 其语义一般由管理事务的软件来执行, 如一个 ebXML 商业服务接口 (BSI)。描述商业事务的BPSS/UML元素有: BusinessTransaction、Business Action、RequestBusinessActivity、RespondingBusinessActivity。

商业文档流: 一个商业事务通过请求方和应答方之间的商业文档流来实现, 但文档流的具体形态取决于实际的商业事务, 例如单向通知 (one-way notification) 与双向会话 (two-way conversation), 因此并不总是有一个请求文档, 或者与之对应的应答文档。描述商业文档流的BPSS/UML元素有: Document Security、Document Envelope、BusinessDocument、Attachment。

合作流程: 描述商业事务之间或者在一个双边合作中子合作之间的排序 (ordering) 和转换 (transitions)。BPSS使用活动图表概念, 如开始状态 (start state)、完成状态 (completion state)、活动 (activities)、同步 (synchronization)、活动转换 (transitions between activities) 和转换保证 (guards on the transitions) 等来编排合作流程。描述合作流程的BPSS/UML元素有: BusinessState、Transition、Start、CompletionState、Success、Failure、Fork、Join。

模式 (patterns): BPSS 提供了一组基础的商业事务和商业合作模式, 借助这些基本模式, 用户团体 (user community) 能灵活定义特殊的商业事务和商业合作, 而且能够在促进商业流程的快速设计、快速执行的同时保持这些描述的一致性 (consistency)。所有表达模式特性所必要的安全和定时 (timing) 参数在 BPSS 中均以元素属性方式提供。

BPSS有两种版本, 一种是UML版, 一种是XML版。BPSS的UML版与XML版建立了明确的映射关系。BPSS的UML元素在XML版中均有其对应的XML元素。

3 WS-CDL

WS-CDL (web services choreography description language)^[11]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不是一种可执行的商业过程描述语言,因此不依赖于任何特殊的商业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,即使参与者各自自由完全不同的语言来实现。目前WS-CDL只是W3C提出的一个工作草案,因此,我们仅对WS-CDL描述的主体—流程编排模型(Choreography model)的基本要素作简略介绍。

Participants, Roles和Relationships:按照WS-CDL模型,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,因而与担当其他角色的其他参与者之间构成了复杂的关系。

Choreographies: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,描述商业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。可以通过流程组合(Choreography composition)将多个流程组合成更复杂的流程,也可以通过流程输入(Choreography import)将外部的流程定义嵌入当前流程之中。为了处理例外情况,WS-CDL还设置了流程恢复机制,包括流程例外模块(Choreography exception block)和流程终结模块(Choreography finalizer block)构成。前者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;后者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。

Activities和控制结构(Control Structures):活动(Activities)是流程中最低层次的组件,每个工作单元中必须包含一个(且只包含一个)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(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。

Types, Variables和Tokens。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,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。

4 BPML

BPML(Business Process Modeling language)^[12]是由BPML.org(Business Process Management Initiative)发展的一种标准,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)。

过程: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中还有二种特殊过程:例外过程(exception processes)和补偿过程(compensation processes)。

活动: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。BPML的简单活动包括:Action:执行一个输入输出消息交换操作。Assign:把一个新值分配给一个特性。Call:调用一个过程并且等待这个过程完成。Compensate:为一个给定的过程调用补偿。Delay:暂停当前动作,等待一个信号。Empty:不做任何事情的动作。Fault:在当前的上下文中给出一个出错信号。Raise:升起一个标志信号。Spawn:调用一个过程而不用等待该过程完成。Synch:同步一个信号,等待一个信号被给出后再执行。

BPML提供的复杂活动的类型有:All:以并行方式执行所有活动。Choice:在多个可供选择的事物中进行选择。Foreach:对表中的每个元素各进行一次执行活动。Sequence:按序列执

行活动。Switch：有条件地执行一批活动中的一个活动。Until：基于一个出口条件一次或多次执行活动。While：基于一个出口条件零次或多次执行活动。

上下文：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。一个上下文包含有只应用于这个上下文范围内的局部定义（local definitions），局部定义主要包括特性、过程和信号。上下文能被嵌套，一个子上下文（递归地）继承了它的父上下文的定义并可以超脱于其父上下文的定义。作为过程一部分来执行的活动在一个过程中有共享的上下文。

特性：在BPML中，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。人们可以认为特性是一个过程的属性（或者实例变量）。

信号：用来调整在一个公共上下文中被执行活动的执行方式的。可以认为信号是消息。在BPML中有构件去发送这样一个消息（升起信号）和等待一个消息（同步信号）。

5 WSCI

WSCI (Web Service Choreography Interface) [13]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。

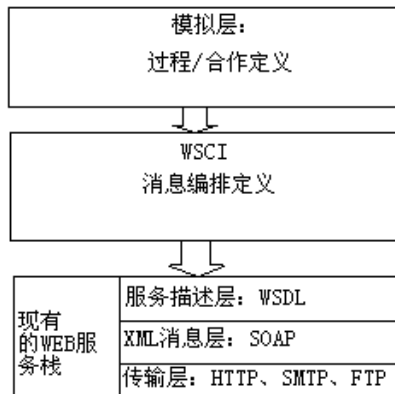


图2 WSCI与WEB服务栈

WSCI是从WSDL结束的地方开始的，它与WSDL协同工作，描述了一个WEB服务看得见的行为。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在现有的WEB服务栈和模拟层之间建立了连接，如图2。WSCI主要参与自下而上的栈，但是预期它会出现和综合在商业合作领域较高层次的层面上。

WSCI与BPML是“姐妹”语言，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。尽管在BPML和WSCI中这些构件存在细微的差别，但他们的功能可类比。注意，assign、raise和synch没有出现在WSCI中，不过在WSCI中有raise和synch的替代活动join。Join将等待被产生的活动完成。

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商业过程并把商业活动映射到了消息交换。BPML和WSCI一起提供了一个end-end的观点。一个end-end的观点描述了整个服务组合过程中每一个商业过程的角色和由每个角色所执行的商业活动。

6 BPEL4WS

WEB服务的商业过程执行语言 BPEL4WS (Business process execution language for web services) [14]是一种XML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，用以替代当时已有的WEB服务组合语言XLANG (web services for business process design) [18]和WSFL (web service flow language) [19]。XLANG由Microsoft发布，是一种块结构化 (block-structured) 语言，具有基本的控制流结构，如sequence、switch、while、all、

pick 等。WSFL 由 IBM 推出, 作为一种面向图表的语言, 主要依赖于控制链 (control link) 的概念, 使用控制链定义的图表能被嵌套而且需要循环。由于 BPEL4WS 是用来取代 XLANG 和 WSFL 的, 在此我们主要讨论 BPEL4WS。

BPEL4WS 在两个层次上对服务组合进行描述: 一方面对商业过程在抽象逻辑层次上进行说明, 另一方面也提供了可执行的过程定义语言。它的抽象商业逻辑描述和可执行商业过程描述共同遵循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 服务接口发生的, 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 BPEL4WS1.0 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 (Partner link types) 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和结构化活动语句, 例如接收 (receive)、应答 (reply)、引发 (invoke)、赋值 (assign)、错误信息生成 (throw)、终止 (terminate)、等待 (wait)、空待 (empty)、顺序 (sequence)、选择 (switch)、条件重复 (while)、条件选择 (pick)、并行 (flow)、嵌套活动 (scope)、补偿 (compensate) 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。

7 现有 WEB 服务组合语言的简单比较

由于 WS-CDL 不依赖于一种特殊的商业过程执行语言, 主要用于定义 WEB 服务参与者之间的可互操作的 P2P 合作, 所以它可能是一种更高层次的组合语言。BPSS 是 ebXML 商业过程规范工具, 它只有与 ebXML CPP/CPA 一起工作才能实现商业过程模拟、组合和配置。因此, WS-CDL、BPSS 与其他语言 (BPML、WSCI、XLANG、WSFL、BREL4WS) 很难相比较。我们此处的比较分析主要针对 BPML、WSCI、XLANG、WSFL、BREL4WS, 主要从组合语言的工作流模式和通信模式入手^{[20][21]}。

文献[20]列举了 20 种工作流模式, 上述五种组合语言均支持与基本路由构件对应的 5 种工作流模式, 即 Sequence、Parallel Split、Synchronization、Exclusive Choice 和 Simple Merge, 并且也都支持 MI (multiple instances) without Synchronization、MI (multiple instances) with a Priori Design Time Knowledge、Cancel Activity 和 Cancel Case。

只有 BPEL4WS 和 WSFL 支持 Multi Choice 与 Synchronizing Merge。

BPEL4WS、XLANG、WSFL 不支持 Multi-Merge, BPML 和 WSCI 有条件地支持。

除 XLANG 外, 其它语言均支持 Implicit Termination。

除 WSFL 外, 其它语言均支持 Deferred Choice。

BPEL4WS 是唯一有条件地支持 Interleaved Parallel Routing 的组合语言。

在文献[21]列出的 6 种通信模式中, 上述五种语言均支持其中的四种通信模式。即 Request/Reply、One-Way、Synchronous Polling 和 Message Passing。

从工作流模式和通信模式的比较结果来看, BPEL4WS 具有更强的功能。考虑到更高层次的组合语言 WS-CDL 和 XSRL 均有赖于 BPEL4WS, 前者“将与 WS-Reliability、WS-CAF (WS-Composite Application Framework)、BPEL4WS 等规范并肩工作并补充它们的不足”^[11],

后者直接以BPEL4WS为基础^[15], 因此数字图书馆的动态定制服务工具应以BPEL4WS为基础。

关于数字图书馆动态定制服务的范例分析, 由于篇幅所限, 我们将在后续文章中以对BPEL4WS中的实例进行编译来专门说明。

参考文献:

- [1] 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: 2004.07.08)
- [2] Shaparau, D. Approaches to Web Service composition. <http://science.unitn.it/~tomasi/think/pdf/pdmi.ppt> (检索日期: 2004.07.08)
- [3] 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: 2004.07.08)
- [4] 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: 2004.07.08)
- [5] Bellwood, T. et al. UDDI Version 3.0.1, UDDI Spec Technical Committee Specification, 14 October, 2003. http://uddi.org/pubs/uddi_v3.htm (检索日期: 2004.07.08)
- [6] 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: 2004.07.08)
- [7] Jian Yang, Mike. P. Papazoglou. Service components for managing the life-cycle of service compositions. Information Systems, 2004,29: 97-125
- [8] 张晓青、张晓林. 数字图书馆服务的动态定制机制. 情报学报 (待发).
- [9] W.M.P. van der Aalst. Don't go with the flow: Web services composition standards exposed. <http://tmitwww.tm.tue.nl/research/patterns/download/ieeewebflow.pdf> (检索日期: 2004.07.08)
- [10] ebXML Business Process Specification Schema 1.1. April 2001. <http://www.ebxml.org/specs/ebBPSS.pdf> (检索日期: 2004.07.08)
- [11] Nickolaos Kavantzias. et al. Web Services Choreography Description Language Version 1.0. <http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427/> (检索日期: 2004.07.08)
- [12] Business Process Modeling Language, 1.0, June 2002. http://www.ebxml.org/bpml_1_0_june_02.htm (检索日期: 2004.7.08)
- [13] Arkin, A. Web Service Choreography Interface (WSCI) 1.0, W3C Note 8 August 2002, <http://www.w3.org/TR/wsci> (检索日期: 2004.07.08)
- [14] 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: 2004.07.08)
- [15] Thatte, S. XLANG: Web Services for Business Process Design. 2001. http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm (检索日期: 2004.07.08)
- [16] ebXML <http://www.ebxml.org/> (检索日期: 2004.07.08)
- [17] OASIS ebXML Collaboration Protocol Profile and Agreement Technical Committee. Collaboration-Protocol Profile and Agreement Specification Version 2.0. <http://www.ebxml.org/specs/ebcpp-2.0.pdf> (检索日期: 2004.07.08)
- [18] 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: 2004.07.08)
- [19] Thatte, S. XLANG: Web Services for Business Process Design. 2001. http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm (检索日期: 2004.07.08)
- [20] W.M.P. van der Aalst. Web Services Composition Languages: Old Wine in New Bottles?. <http://tmitwww.tm.tue.nl/research/patterns/download/wscl-euromicro.pdf> (检索日期: 2004.07.08)
- [21] W.M.P. van der Aalst. et al. Pattern Based Analysis of BPML (and WSCI). http://tmitwww.tm.tue.nl/research/patterns/download/quot_bpml_rep.pdf (检索日期: 2004.07.08)