

基于 BPEL 和 UDDI 的动态 Web 服务组合系统架构

A BPEL and UDDI Based Dynamic Web Service Composition System Architecture

张花萍 陈晓苏 刘辉宇 刘剑 (华中科技大学计算机科学与技术学院 430074)

摘要: BPEL 是专为整合 Web 服务而制定的一项规范标准,利用 BPEL 组合 Web 服务,能够创造出复杂的业务流程。然而 BPEL 不支持在运行阶段动态绑定和修改 Web 服务,针对这一问题,本文探索性地提出了动态 Web 服务组合系统(DWSCS)架构。借助 UDDI 动态发现 Web 服务的能力,弥补了传统 BPEL 业务流程只能对 Web 服务进行静态绑定的缺陷,提供了动态绑定 Web 服务和异常情况下动态修改 Web 服务的功能,增强了 Web 服务的业务组合能力。

关键词: 业务流程执行语言 统一描述 发现与集成 Web 服务组合

1 引言

Web 服务(Web Services)通过定义标准协议和格式,为异构应用程序的交互、整合提供了一种具有极大灵活性的手段^[1],但其较为薄弱的业务组合能力限制了在企业中的广泛应用。

为增强 Web 服务的业务组合能力,BEA、IBM、微软和 Web 服务社区联合提出了新一代 Web 服务流技术——业务流程执行语言 BPEL4WS(Business Process Execution Language For Web Service,也可称为 BPEL)^[2]。

BPEL 为业务流程提供了与平台无关的定义和描述手段,并试图提供简单、快速的 Web 服务组合能力,但业务流程中使用的 Web 服务必须在设计或配置阶段进行静态绑定^[3],而不能在运行阶段根据具体情况进行动态绑定或动态修改。

本文探索性地提出了一种动态 Web 服务组合系统 DWSCS(Dynamic Web Service Compose System)架构。该架构基于 BPEL 和统一描述、发现与集成 UDDI(Universal Description、Discovery and Integration)^[4]协议,可支持业务流程在运行阶段动态绑定 Web 服务和异常情况下动态修改 Web 服务。

2 相关技术

2.1 业务流程执行语言(BPEL)

BPEL 是专为整合 Web 服务而制定的规范标准。它结合了 IBM WSFL 和 Microsoft XLANG 两门语言的优点,同时摒弃了一些复杂繁琐的部分,成为一种较为自然的描述业务流程的抽象高级语言。目前,BPEL 已成为业界事实上的标准。

BPEL 位于 WSDL、XML Schema 和 XPath 等 XML 规范之上。WSDL 消息和 XML Schema 类型定义为 BPEL 流程提供各种数据类型,XPath 为数据处理提供支持。

BPEL 的文法是完全基于 XML 规范的,具有赋值、循环、选择、远程调用、错误捕捉、错误抛出等传统程序语言特性。同时还提供对远程调用的同步和异步处理、并行和补偿操作。

2.2 统一描述、发现与集成(UDDI)协议

UDDI 协议是 Internet 上发布和发现 Web 服务和业务信息的一种标准。该标准不仅使得企业能够注册自身提供的 Web 服务,还能使别的企业能够发现已注册的 Web 服务。

UDDI 的核心组件是 UDDI 注册,它使用一个 XML 文档来描述企业及其提供的 Web 服务,所有的 UDDI 注册信息都存储在 UDDI 注册中心。企业可以通过 UDDI 注册中心的 Web 界面,将有关 Web 服务加入到 UDDI 注册中心。

3 DWSCS 架构

DWSCS 利用 UDDI 实现 Web 服务的动态发现,即利用 UDDI 统一发现服务的功能,为运行中的业务流程提供合乎需要的 Web 服务列表,供用户选择。在此基础上,可实现在业务流程运行阶段动态绑定 Web 服务和异常情况下动态修改 Web 服务的功能。

DWSCS 包括客户端、代理、BPEL 业务流程模板库、UDDI 注册中心和 BPEL 引擎五个模块,其结构如图 1 所示。

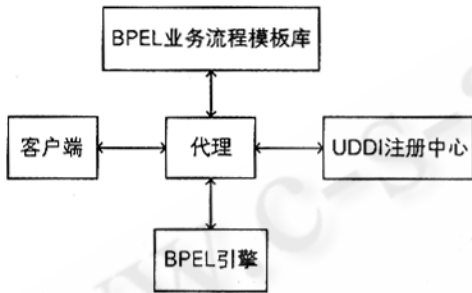


图 1 DWSCS 结构图

(1) 客户端:请求执行 BPEL 业务流程的实体。

(2) 代理:DWSCS 的核心模块,其功能主要包括。

① 接受客户端请求,调用相应的 BPEL 业务流程模板;

② 解析 BPEL 业务流程模板,并向 UDDI 注册中心查询适用的 Web 服务;

③ 将查询结果以列表的形式送往客户端供用户选择;

④ 根据用户的选择动态绑定 Web 服务;

⑤ 将 BPEL 业务流程送往 BPEL 引擎执行。

(3) BPEL 业务流程模板库:本模块的主要作用是存储 BPEL 业务流程模板。其中 BPEL 业务流程模板已绑定默认的 Web 服务,在运行中可用适当的 Web 服务替换默认的 Web 服务。

(4) UDDI 注册中心:Web 服务的发布中心,其功能主要包括:

① 用户发布 Web 服务,供其它用户调用;

② 用户可在此查询需要的 Web 服务,获取相应的 Web 服务信息,以便进行后续绑定。

(5) BPEL 引擎:该模块的主要功能是执行 BPEL 流

程。

4 DWSCS 的操作流程

4.1 动态绑定服务

DWSCS 在运行阶段动态绑定 Web 服务的操作流程如图 2 所示。

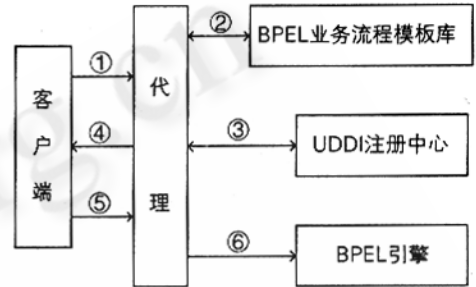


图 2 动态绑定 Web 服务操作流程图

① 客户端向代理发送请求;

② 代理根据客户端发送的请求选择并解析相应的 BPEL 业务流程模板;

③ 代理根据解析结果,向 UDDI 注册中心查询符合该 BPEL 业务流程模板要求的 Web 服务;

④ 代理将可用的 Web 服务以列表的形式发送给客户端;

⑤ 客户端将相应的 Web 服务选择结果发送给代理;

⑥ 代理根据客户端的选择动态绑定业务流程中的 Web 服务,并将业务流程送往 BPEL 引擎执行。

4.2 动态修改服务

在业务流程运行过程中,如果某个 Web 服务出现异常导致其不可用,为避免业务流程中断,DWSCS 中的 BPEL 引擎将启动代理向 UDDI 注册中心查询可替代的 Web 服务对出现异常的 Web 服务进行动态替换,确保业务流程能够继续正常执行。其操作流程如图 3 所示。

① 业务流程运行中如果某个 Web 服务出现异常,暂停业务流程的执行,同时 BPEL 引擎将出现异常的 Web 服务信息通知代理;

② 代理向 UDDI 注册中心查询可替代的 Web 服务;

③ 代理根据查询结果选择一个可用的 Web 服务

发送给 BPEL 引擎,引擎调用新的 Web 服务,业务流程继续执行。

4.3 对传统 BPEL 业务流程的支持

DWSCS 同样可支持传统的静态绑定 Web 服务的 BPEL 流程,其操作流程如图 4 所示。

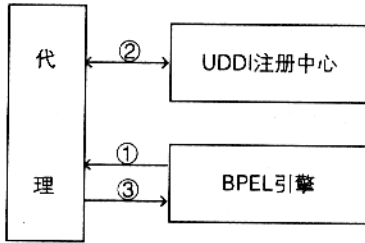


图 3 动态修改 Web 服务操作流程流程图

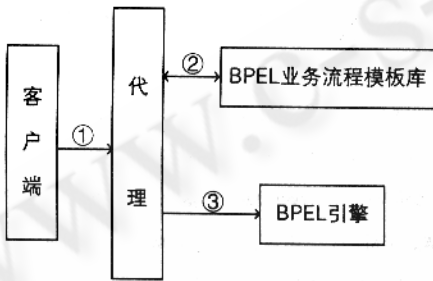


图 4 执行传统 BPEL 操作流程流程图

① 客户端向代理发送请求。此时需选择“接受默认 Web 服务”选项；

② 代理根据客户端发送的请求选择相应的 BPEL 业务流程模板；

③ 代理将 BPEL 业务流程模板送往 BPEL 引擎执行。

5 结论

本文结合 BPEL 和 UDDI 规范,探索性地提出了一种具有业务流程运行阶段动态绑定和修改 Web 服务能力的 DWSCS 架构,该架构具有较好的自适应性和可恢复性,增强了 Web 服务的业务组合能力。

为增强该架构的实用性,在今后的工作中,将加强研究业务流程的安全、Web 服务的自动绑定策略、BPEL 业务流程模板中的 Web 服务粒度划分等问题。

参考文献

- 1 Weerawarana, S., Curbera, F., Leymann, F., Storey, T., Ferguson, D.: Web Services Platform Architecture. Prentice Hall 2005.
- 2 Curbera, F., Golan, Y., Klein, J., Leymann, F., Roller, D., Thatte, S., Weerawarana, S. Business Process Execution Language for Web Services (BPEL4WS) 1.1. May 2003. <http://www.ibm.com/developerworks/library/ws-bpel>.
- 3 Fung, C. K.; Hung, P. C. K.; Linger, R. C.; Walton, G. H.; Extending Business Process Execution Language for Web Services with Service Level Agreements Expressed in Computational Quality Attributes System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on 03 - 06 Jan. 2005 Page(s):166a - 166a.
- 4 OASIS UDDI Specifications TC: UDDI v. 3 Specification. October 2003. <http://www.oasisopen.org/committees/uddi-spec/doc/tcspecs.htm>.