

# 浅谈面向服务架构(SOA)的核心理念

张春霞<sup>1</sup> 李旭东<sup>2</sup> 徐涛<sup>3</sup>

(1.天津科技大学 自动化学院 天津 300222; 2.南开大学 软件学院 天津 300071;

3.中国民航大学 计算机科学技术学院 天津 300300)

**摘要:** 首先回顾了计算机技术发展的路线及面向服务的由来,详细探讨了面向服务架构(SOA)的核心理念和判定标准,及其核心元素和软件开发过程,并就 Web 服务和 SOA 进行了比较来进一步认识 SOA 的内涵,并给出可支持构建基于 SOA 应用系统的具体技术,最后探讨了 SOA 进一步研究方向。

**关键词:** 复用;服务;面向服务架构;企业服务总线(ESB);Web 服务;服务组件架构(SCA);服务数据对象(SDO)

## Discussion about the Core Ideas of Service-Oriented Architecture

ZHANG Chun-Xia<sup>1</sup>, LI Xu-Dong<sup>2</sup>, XU Tao<sup>3</sup> (1. College of Automation, Tianjin University of Science and Technology, Tianjin 300222, China; 2. College of Software, Nankai University, Tianjin 300071, China; 3. College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China)

**Abstract:** This paper firstly reviews the evolvement roadmap of computer technology and the reason why service-oriented is introduced. It discusses the core ideas of SOA in detail as well as the judgment standard, the core elements of SOA, and the software development process based on SOA. It digs the essence of SOA by comparing web service to SOA, and presents some technologies which support applications based on SOA. At last, this paper discusses the future study of SOA.

**Keywords:** reuse; service; service-oriented architecture; enterprise service bus; Web service; service component architecture (SCA); service data object (SDO)

## 1 引言

软件复用一直是软件技术发展的重要动力。从面向过程到面向对象编程,追求着软件源代码层次上的复用;从面向组件到面向消息编程,追求着运行时代码层次上的复用;从面向设计模式的软件工程到面向软件体系架构的软件工程,则追求着软件设计思想上的复用。这些都是由技术主导下的计算机软件不断进步。众多 IT 提供商创造出各种 IT 技术为现代企业建立了一个个信息孤岛,这些孤岛若不能有效互联与再造,那么所有孤岛都将成为企业的包袱,而它所为企业获得的利益远远小于企业为它而付出的高额成本。

事实上企业热切盼望着由企业需求主导的计算机软件的不断进步。传统的进程间通信技术、分布式组件技术、跨平台的组件技术都未能切中企业最核心的期盼即:计算机技术能完美地支持企业随需而变的业务需求。1994 年 Gartner 公司的 Alexander Pasik 首次提出了面向服务架构 (Service-Oriented Architecture, SOA)的思想<sup>[1]</sup>,其将满足众多 IT 提供商和诸多企业的多层次复用和随需而变的能力,这是划时代的里程碑。尽管在当时它是超越现实的。随着时间推移、技术的进步,SOA 思想逐渐走向清晰和系统。

基金项目:国家高技术研究发展计划(863)(2006AA12A106);南开大学科学技术创新基金(z1A2006014)

收稿时间:2009-09-22;收到修改稿时间:2009-11-15

总的来讲, SOA 侧重于从业务角度考虑问题, 强调服务概念和服务整合思想, SOA 以独立于具体技术平台的标准化协议、面向消息的、高度分布式的、具有智能路由等规范来整合系统(包括既有企业应用系统和未来新应用系统), 从而真正实现了松散耦合的异构系统透明互联。一方面, SOA 独立于具体技术平台, 其倡导开放的、可集成、自动化、虚拟化技术标准, 各种具体的 IT 技术可以各显其能, 故 SOA 一经提出便受到 IT 技术提供商的广泛推崇; 另一方面, SOA 能有效地适应企业的多变需求, 又能使企业既有 IT 投资成为重要资产而不是包袱, 另外其也使得二次开发的成本大幅降低, 因此 SOA 也同样受到企业用户的青睐。

## 2 SOA的核心理念

### 2.1 可异构互联的服务

SOA 强调的重中之重就是服务, 这是 SOA 的核心理念, 也是区别与以往企业应用集成(EAI)的所有解决方案的最重要不同之处。通常的服务代表着一定的业务功能且可以被其它方请求调用, 它是企业业务系统对内、对外的基石。从这个宽泛的服务概念上讲, 以往的中间件或(分布式)组件都可称为服务。但 SOA 中的服务不同于以往的中间件或组件。

SOA 中服务除了具备一般服务的意义外还需满足如下要求: 即服务接口(或声明)和服务的具体实现是相分离的, 且服务的请求方采用基于文档方式(即消息方式)来请求调用服务提供方的服务。在 SOA 中服务请求方和服务提供方可以是完全异构的实现。而两者之间的消息(即服务协议)则是标准化的和开放的, 并具有良好的服务契约(Well-Defined Service Contracts), 当然必须可以实现互联。

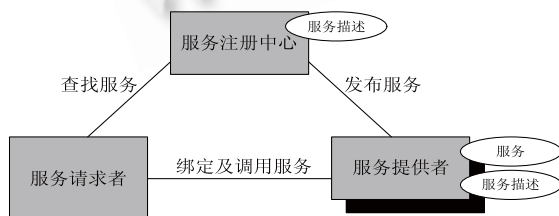


图 1 SOA 概念模型<sup>[2]</sup>

服务是自描述的, 通过标准描述格式定义服务提供的操作、消息格式和消息交换模式(MEPs), 无论是

调用者还是被调用者无需关心其它诸如具体地址、具体实现技术等信息。服务是标准的和开放的。SOA 理念下的具体服务标准可能是多样的, 如 Web 服务、Corba 服务等标准。不同标准服务也需能够互联, 这也是 SOA 中具有吸引力的特点。服务是动态的, 已发布服务可被动态发现和绑定。如图 1, 服务请求方通过服务注册中心动态查找所需的服务, 然后动态绑定并请求调用该服务。这与以往的静态绑定服务的方式完全不同。同时服务也是有质量保障(QoS)的, 为了实现 QoS 服务往往是粗粒度(Coarse-Grained)的, 从而减少多次对消息的封装、传输与拆解的消耗。

服务还可包装已有应用或组件。这一特性使得服务的应用领域更加广泛, 且可使企业现有资产被重用, 从而保护企业已有 IT 投资。同时, SOA 服务也使得分布在异构平台上的组件得以互联, 这既得到 IT 提供商的认同和欢迎, 也使企业长久的心声即业务需求终于摆脱 IT 技术的限制了。

### 2.2 松散耦合

SOA 的松散耦合体现在客户端和服务端最大限度的独立, 两者之间只建立在文档形式的服务契约上。通过松散耦合实现了 SOA 的敏捷性<sup>[3]</sup>。具体表现如下:

#### 1) 服务之间的松散耦合

服务是可独立操作的, 其执行不依赖于其它组件或服务上下文和状态, 而操作是通过标准化方式来封装和发布的。

#### 2) 服务接口和服务实现之间的松散耦合

接口和实现无关。SOA 在强调服务的标准化的同时, 包容了服务具体实现的技术多元化。这是现实的, 也是受到 IT 技术厂商的一致认同。

a) 服务的请求者与服务的提供者可以分别由不同的开发语言来实现; b) 服务的请求者与服务的提供者可以分别由不同的运行平台上; c) 服务的请求者与服务的提供者可以分别采用不同的服务标准来互操作。其中, a), b) 两种情况的松散耦合是由服务采用基于消息机制的协议标准来实现的; 而 c) 种情况的松散耦合是由需要对具体服务的协议标准的转换来实现, 而这种转换只有是透明的才能实现松散耦合, 其需要采用企业服务总线(ESB)来实现的。

#### 3) 服务的接口协议和传输协议之间的松散耦合

服务的接口协议(或业务协议)不依赖于具体网络传输协议(HTTP、JMS、CORBA IIOP、IBM WebSphere

MQ、Tibco Rendezvous，甚至 FTP 等)。由于采用消息机制，完全可不依赖于具体传输协议。而更具体的消息机制可依具体情况指定，这些消息交换机制包括同步机制和各种异构机制(发送/遗忘机制、发送/回调机制、发布/订阅机制等)

4) 服务请求者与服务提供者位置(location)的松散耦合

以往跨平台组件技术中，客户端必须至迟在部署时知道服务的位置，一旦服务器端服务位置变化则必须修改客户端相关配置。在 SOA 中服务请求者可从服务注册中心动态查找特定服务然后再绑定并调用该服务，服务的位置信息是运行时绑定的即对客户而言是透明的。强调这种透明性，是因为服务是动态的，不断变化的，事前是不能预料的，故其实现了服务请求者与服务提供者调用关系的松散耦合。

另外也可采用基于服务总线 ESB 的 SOA 模型(如图 2)实现服务位置的透明化<sup>[4]</sup>，即通过 ESB 来代理服务提供者(也即隐藏了服务的真实提供者)，服务请求者只要知道 ESB 的地址即可，服务请求者通过 ESB 的服务代理，透明地实现了对服务真实提供者服务的调用。对于企业信息平台而言 ESB 一般是不会频繁变动地址的。

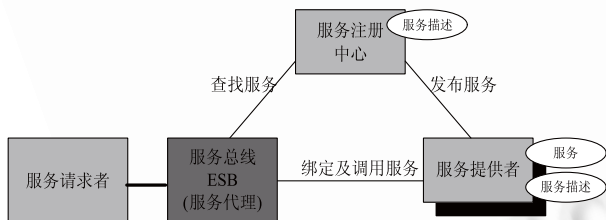


图 2 基于 ESB 的 SOA 模型

5) 业务和 IT 技术的松散耦合

通过上述种种松散耦合、服务接口协议(即消息)与具体 IT 平台无关性以及服务的可组合性，企业最终可建立基于服务(接口)的业务平台，而这一业务平台与具体的 IT 技术平台是无关的。这正是 SOA 所追求的最理想目标。

6) 服务和业务流程的松散耦合

SOA 理念更高标准还体现在单纯的业务层面上追求服务和业务流程的去藕，即服务不应与具体的业务流程相关，只有这样该服务才可被重用到其它业务流程和应用中。

2.3 服务可编排、可编舞

服务是可编排的(Orchestration)，通过组装多个服务形成一个新的组合服务(Composite Service)，组合服务可提供比单个服务更复杂功能和更高灵活度，而被组合的服务也意味着被重用，这正是企业 ROI 所需要的。服务也可相互编舞的(Choreography，即协作)，若干个服务按照业务需求有序协作共同完成一个任务。服务协作不同于服务编排，服务编排事实上是新建立一个大的总控服务，而服务协作中没有集中的总控服务<sup>[5]</sup>。这两种服务组合都有重要应用价值，它们最终成为企业的业务流程(Business Process)。又由于 SOA 服务的动态绑定特性，可以实现运行时的动态发现服务、组合服务从而完成给定任务，因此企业的业务流程很容易随需而变。这也为人工智能打开了崭新的道路。

对于上述两种服务组合，SOA 都没有界定它们的实现技术。然而对于服务编排，基于 XML 的 BPEL (Business Process Execution Language)无疑是很重要的一种实现方式，它的优点在于它是基于文本的 XML 文件的，任何支持 BPEL 规范的 BPEL 引擎都可以执行基于 BPEL 的组合服务。对于服务编舞，基于 XML 的 CDL(Choreography Description Language)是被标准化的重要的实现方式之一。

上述 SOA 三个核心理念可作为判定企业应用系统是否是基于 SOA 标准。另外基于 SOA 的应用系统开发过程中与传统开发过程不同，前者很好地支持交互和迭代式开发，这一点也可以作为判定应用系统是否基于 SOA 的参考标准。

3 服务的层次与规范标准

服务是可组合的，因此一个企业会出现很多服务，这些服务还可以细化为如图 3 层次关系。

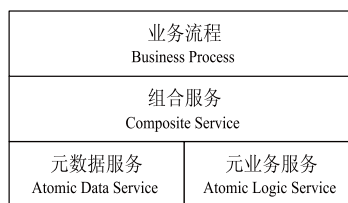


图 3 服务层次模型

元业务服务是对传统组件或应用服务封装。元数

据服务则是对各种途径的数据的服务封装。它们是最底层的、原子型的服务,通称为元服务,其可封装 EJB、DCOM 等组件。

组合服务则是在元服务基础上的进一步组装的服务,它主要是侧重短期运行(short-term)的服务,它已经是一个足够粗粒度的服务,同时也是企业为内部和外部环境提供的最可频繁复用的服务,例如电子商务中银行方的转帐服务。

业务流程同样也是组合服务,又可以称为工作流,它与具体的业务密切相关,而且更侧重长期运行(long-term)的服务,例如电子商务中的持卡购物业务流程,它涉及网上商店、认证中心、支付网关、银行转账等多个服务的调用。正如前述服务应该与业务流程是松散耦合的,这一点是基于 SOA 的分析与设计中的重要目标和特色。

前述的元服务还可进一步细化为公用服务、基础设施服务、外部伙伴服务等<sup>[6]</sup>。

一个理想的 SOA 服务体系,需要针对服务做出如下规范标准:消息规范、传输规范、描述规范、发布和发现规范、组合和协作规范以及监管规范等。其中的消息规范,不仅包括基本的消息交互模式(MEP)和消息处理框架,还应考虑动态寻址、可靠、安全、事务、服务质量等内容。而这些规范应该有能力和包容已有的相应具体技术规范,也具有可扩展性来包容未来新技术规范的能力。同时,其中的描述规范、发布和发现规范以及监管规范等还应独立于 IT 技术平台。

## 4 服务注册中心

正如图 1 所示,SOA 的概念模型中需要“服务注册中心”这样的机构,通过引入服务注册中心,将服务提供者和服务请求者之间建立一种间接联系关系,而不是直接联系,这样向服务位置、消息协议等信息就可以推迟到运行时动态发现和绑定,这无疑体现了 SOA 的松散耦合的理念。服务提供者需要及时将服务的最新信息更新到服务注册中心。

服务注册中心核心的功能就是提供服务的查询,也即服务目录的功能。多个服务注册中心可以按 SOA 服务的标准来组成一个更大的分布式服务目录系统。毫无疑问,服务注册中心是 SOA 中重要的一部分,无论服务注册中心是以何种方式出现在 SOA 的基础设施中,也就是说它可能不作为独立的一个应用而存在。

目前的 UDDI(Universal Description, Discovery and Integration)正扮演着服务注册中心的角色,但是 UDDI 的最初想法并不是仅为服务设计的,同时它是否能很好地诠释服务和服务于服务,还有待进一步发展。而与此同时服务目录规范的标准化仍然需要进一步的研究。

## 5 企业服务总线(ESB)

在 SOA 基本架构中,企业服务总线(Enterprise Service Bus, ESB)不是必须的,但无疑有了它将会更好地支持 SOA 的核心理念,也简化了开发、部署和动态维护的过程,尤其是构筑企业级应用。

理想的 ESB 应提供多种服务标准的互联、数据转换、服务查询、智能路由、授权、安全、可靠、负载均衡、服务管理、服务监控与日志等功能,ESB 核心能力是提供基于 SOA 的异构互联<sup>[1]</sup>。多个 ESB 还可组成一个异构 ESB 分布式系统平台,通过 ESB 网关实现多个 ESB 互联,这对构建基于服务企业(Service-Oriented Enterprise, SOE)尤为重要。通过 ESB 分布式系统,企业内外诸多应用可实现基于服务的彻底互联,从而敏捷地支持随需而变的业务动态变更。

ESB 事实上集成了 SOA 的多个理念于一体,但是 ESB 的诸多功能也可以各自委托第三方系统来协助完成,例如服务查询可以转交给服务注册中心来完成,不同服务标准的数据转换可以交给消息协议转换器来完成等。

目前已有的 ESB 产品,包括 IBM Message Broker, IBM WebSphere ESB, BEA AquaLogic, JBI ServiceMix, Apache Synapse, Celtix/Xfire, Mule, JBOSS ESB<sup>[3]</sup>等等,这些企业级产品或开源平台的侧重角度不同,功能的差异也不小,但基本都支持具有广泛影响力的 Web 服务。

## 6 基于SOA的软件开发过程

服务的生命周期如图 4 所示,一个服务从业务需求中识别服务开始,或者不断迭代,或者最终被撤销,其完整地融入到 SOA 系统的生命周期中。SOA 系统的生命周期依然遵循传统软件工程的几个阶段,只不过需要明确指明面向服务的特点,即面向服务的分析、面向服务的设计、面向服务的开发、面向服务的测试、面向服务的部署、面向服务的管理等几个阶段,而重



点是分析、设计和管理阶段。

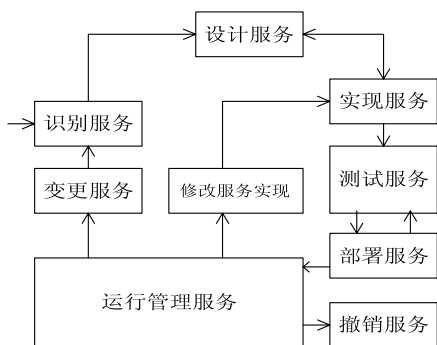


图 4 单个服务的生命周期<sup>[7]</sup>

在面向服务的分析中,首先定义业务需求,然后识别出可自动化的系统,然后服务建模筛选出服务候选。服务建模这一过程又往往先从分解业务流程开始,识别业务服务候选及服务操作,并根据处理需求识别应用服务操作从而确定应用服务候选,再修订服务组合,形成较完整的服务候选。

通过面向服务的设计,明确 SOA 核心标准和扩展规范标准,尽管这些标准都是符合 SOA 的理念,但是具体 SOA 应用必需根据企业实际环境、行业标准以及企业战略考虑选择出具体规范集。在此基础上给出符合 SOA 规范的企业业务数据表示,然后设计以实体为核心的业务服务(一般指元服务),再设计应用服务,并进而设计以任务为核心的业务服务(一般指组合服务),最终设计面向服务业务流程。

面向服务的开发即服务的具体实现,这些基本借助传统软件开发的完成。而面向服务的测试、部署以及管理,都是明显带有 SOA 特点的,因此尚需要进一步研究与实践。

SOA 的生命周期涵盖了目前正火热探讨的服务科学、服务工程、服务管理等内容,其各阶段的具体方法正是软件工程领域尚待进一步研究的内容,毕竟基于 SOA 的应用系统刚刚走进实际企业 IT 环境中。

## 7 SOA与WEB服务

SOA 与 Web 服务不是等同的。尽管因 2001 年 Web 服务的被提出,SOA 理念才真正被人们重新认识和挖掘。SOA 是一种理念,Web 服务则是 SOA 理念下的一个具体实践,准确地说 Web 服务是 SOA 的一

个具体规范标准集实例,尽管似乎先有了很多 Web 服务的规范。

目前而言,Web 服务技术体系也包括传输规范、消息规范、消息扩展规范、描述规范、发布和发现规范、Web 服务组合与协作规范、管理规范等。

其中,Web 服务的传输规范主要以 HTTP(Hyper Text Transfer Protocol)为首选,利用 HTTP 的基本操作(GET/POST/PUT/DELETE 等)发送基于 XML 的消息。

Web 服务的消息规范主要以 SOAP(Simple Object Access Protocol)来为 Web 服务之间交换基于 XML 的结构化信息提供可扩展的消息交互协议和消息处理框架;消息扩展规范(即 WS-\*)则主要来增强基本消息规范中的消息处理框架,例如 WS-Addressing 来实现 Web 服务实例的定位、WS-ReliableMessage 来实现 Web 服务松耦合的消息的可靠传输、WS-Security 来实现消息的安全、WS-Transaction 来实现 Web 服务的事务处理、WS-Policy 来实现服务质量(QoS)等等<sup>[8]</sup>。Web 服务的消息扩展规范不只上述提到的,WS-\*中有的已经成为标准,有的尚待讨论。这些 Web 服务的扩展规范成熟与否也是 Web 服务能否走向实用道路的关键,这些也正是 Web 服务目前研究的热点。

Web 服务的描述规范主要以 WSDL(Web Services Description Language)来定义服务的描述,包括 Web 服务的功能、接口和接口的绑定方式等。

Web 服务发布和发现规范则主要采用 UDDI 来为 Web 服务定义各种元信息,如面向商业实体信息的白页信息、面向标准分类的黄页信息,及描述技术细节信息的绿页信息。UDDI 采用 SOAP 来实现发布、发现、更新服务元信息等。

Web 服务的组合规范是 Web 服务业务过程执行语言 BPEL4WS,这已经是业界事实上的 Web 服务组合语言标准。Web 服务的协作规范则以 W3C 提出的 WS-CDL(Web Services Choreography Description Language)<sup>[9]</sup>为主,其为 Web 服务提供对等协作方式的消息通信规范。

Web 服务的管理规范则以 OASIS 提出的 WS-Manageability 为主,其为 Web 服务的监控和配置提供保障,也是企业成功实现基于 Web 服务的系统平台的关键。

如上 Web 服务技术体系的规范也是 SOA 应该具备的内容。但 SOA 更是一种体系架构风格,一种理念。SOA 的松散耦合理念在 Web 服务技术体系中还未能很好的贯彻,例如 Web 服务中传输规范主要是与 HTTP 协议绑定,也即 Web 服务还是一个具体技术规范集,而不是一种思想。然而也要看到,Web 服务也在不断发展,且正在进一步解耦中。

总之,Web 服务技术体系为丰富 SOA 内涵提供的更多的借鉴,基于 Web 服务可以构建出符合 SOA 理念的系统,但是 Web 服务不等同于 SOA。

## 8 支持构建SOA应用系统的具体服务技术

当前支持构建 SOA 应用系统的具体服务技术有:Web 服务、CORBA 服务、IBM MQ、以及各种 B2B 平台的服务技术(如 ebXML、RosettaNet)等,其中 Web 服务作为 SOA 理念的事实标准,已经有大量的近乎成熟的规范可以直接指导基于 SOA 的应用系统构建,尤其是具有各种服务质量的企业级应用。这些具体服务技术都可以支持基于 SOA 的应用系统构建,但不能说只要用到这些具体技术就是基于 SOA 的应用系统,还需用 SOA 核心理念和判定标准来衡量。

## 9 总结与展望

本文全面探讨了面向服务架构(SOA)的核心理念及其关键元素。事实上 SOA 的内涵与具体实践模式等也一直在不断丰富和发展。

人们对符合 SOA 理念的服务已经有了更深层次的认识。2004 年 IBM 和 BEA 等提出了服务组件架构(Service Component Architecture, SCA)<sup>[10]</sup>, SCA 给出了抽象地独立于技术的服务,它是 SOA 服务的一种实现,但它又不同于以往的具体服务技术。从 IT 技术角度,SCA 强大到可以封装各种平台组件,也可以支持各种符合 SOA 服务标准的访问;从业务角度来看,SCA 就是独立于技术的业务逻辑,因此企业业务人员可以直接对其进行变更和组合等操作从而不断满足企业业务需求。另外与 SCA 相伴的是服务数据对象(Service Data Objects, SDO),SDO 目标是建立企业统一的业务数据模型,从而支持基于 SOA 的上层应用系统可以采用统一的透明的数据访问和处理模式而不需要关心这些数据到底来源于具体的哪些异构数据

源。这两项重要的技术已于 2007 年进入 OASIS 的标准化过程,它们将深深影响到 SOA 的未来具体实践,因此值得进一步研究。

另外,基于 SOA 的各种服务规范、ESB 功能规范、SOA 监管、基于 SOA 软件开发过程等都值得进一步深入探索。

SOA 作为一种超技术的理念,需通过对具体技术的包容来体现它无所不包无所不通,唯一不变的就是其核心理念。基于 SOA 的企业业务系统不是一次建成的,而是需长期建设的,因此需要 SOA 成熟度模型来评价。SOA 也已不仅仅是软件系统的一种体系架构思想了,它正成为一种文化。

## 参考文献

- 1 Josuttis N M. SOA in Practice. Cambridge: O'Reilly Media. 2007. 7 - 8,47 - 56.
- 2 Huhns MN, Singh MP. Service-oriented computing: key concepts and principles. IEEE Internet Computing, Jan-Feb. 2005. 75 - 81.
- 3 梁爱虎编著. SOA 思想、技术与系统集成应用详解. 北京:电子工业出版社, 2007. 7 - 12.
- 4 Thomas E.王满红, 陈荣华译. SOA 概念、技术与设计. 北京:机械工业出版社. 2006. 37 - 39.
- 5 喻坚,韩燕波.面向服务的计算-原理和应用.北京:清华大学出版社, 2006. 171 - 177.
- 6 Kulkarni N, Dwivedi V. The Role of Service Granularity in A Successful SOA Realization - A Case Study. 2008 IEEE Congress on Services, 2008. 423 - 430.
- 7 Jeng JJ, Lian Jun. A System Dynamics Modeling for SOA Project Management. IEEE International Conference on Service-Oriented Computing and Applications (SOCA'07), 2007. 286 - 294.
- 8 Eric NC, Grep L. Understanding SOA with Web Services. Pearson Education, 2005. 273 - 346.
- 9 Yang HL, eds. A Formal Model for Web Service Choreography Description Language (WS-CDL). Proc. of the IEEE International Conference on Web Services. Washington: IEEE Computer Society, 2006. 893 - 894.
- 10 王紫瑶,南俊杰,段紫辉,钱海春. SOA 核心技术及应用.北京:电子工业出版社, 2008. 18 - 53.