

doi:10.3969/j.issn.1002-0802.2014.03.009

服务发布和发现在 SOA 架构中的应用分析

张锋军, 陈宇雷

(中国电子科技集团公司第三十研究所, 四川 成都 610041)

摘要: SOA 目前已成为一种主流的软件架构, 服务发布和发现作为实现 SOA 架构的一项重要基础设施, 其重要性不言而喻。介绍了基于 UDDI (Universal Description, Discovery, Integration, 统一描述、发现和集成协议) 的服务发布和发现技术实现原理, 对服务发布和发现协议、信息模型、与安全服务和管理服务的集成等实现技术进行了重点研究, 提出了设计思路和方法。

关键词: 服务 发布 发现 UDDI 安全服务

中图分类号: TP311 **文献标志码:** A **文章编号:** 1002-0802(2014)03-0275-06

Application Analysis on Service Published and Discovery Technology in SOA

ZHANG Feng-jun, CHEN Yu-lei

(No. 30 Institute of CETC, Chengdu Sichuan 610041, China)

Abstract: SOA now becomes a mainstream software architecture. As a very important infrastructure of SOA, the service release and discovery is significant indeed. This paper describes the technical mechanism of UDDI, with emphasis on UDDI protocol, information model and the integration with security service and management service, and finally provides some design ideas and design methods.

Key words: service; publish; discovery; UDDI; security service

0 引言

Web Services 作为实现 SOA 架构的主要技术标准, 得到了广泛的研究和应用。在基于 Web Services 的 SOA 软件架构中, 服务消费者需要知道如何获取服务、谁提供了服务、服务类型是什么、服务的质量如何、服务的安全策略等信息。服务提供者需要一个代理来实现额外信息的补充说明, 同时服务消费者需要从这个代理那里获取额外的信息。为了满足这样的需求, 主流 IT 厂商和相关组织提出了 ebXML、UDDI、Konark 等多种服务发布发现实现机制。

根据其实现方式的不同, 服务发布与发现的架构分为集中式(有中心, 基于目录, 适用于企业级信息系统)、分布式(无中心, P2P 架构, 适用于移动网络环境)和混合式(基于目录的分布式架构, 兼具前两种方式的特点)三种, 三种方式各有其适用场合和

优缺点^[1]。其中集中式架构以 UDDI 为代表, 在企业级信息系统中应用最为广泛。UDDI 提供了一种发布与发现 Web 服务的标准方法, 用于建立一个开发并且与平台无关的框架以描述、发现和集成服务。

1 服务发布与发现技术原理分析

1.1 服务发布与发现在 SOA 架构中的作用

在基于 Web Services 的 SOA 系统服务交互过程中, 有服务提供者、服务消费者和服务注册中心三种角色。三种角色的交互具体涉及到服务发布、查找和绑定操作。一般情况下, 服务提供者负责提供可通过网络访问的服务, 服务提供者定义好 Web Services 的服务描述, 并将它发布到服务注册中心或者直接发布给服务消费者(在调用关系明确不变和服务数量少、关系简单的情况下适用)。服务消费

者从本地或者服务注册中心进行服务查找操作,搜索到服务描述,然后根据服务描述完成与服务提供者绑定的操作,以调用相应的 Web Services 并与之交互。

UDDI 作为服务注册中心,对服务发布者和消费者分别提供了相应的发布者 API 和发现者 API,供开发者使用。通过 UDDI, SOA 系统可以实现服务重用、互操作、位置透明和服务之间的松耦合。

1.2 服务发布与发现的需求

服务发现的需求主要包含:服务的发布,服务的查询,服务元数据的管理和安全集成^[2]。

(1) 服务发布

服务发布需要将诸如服务提供者、服务、服务实例及其相关元数据放入注册机中。需要支持以下的机制:

1) 手工服务发布——用户或者操作员作为服务发布者,使用服务发布用户界面把服务实体发布到注册机。这是最直接的服务发布方式。

2) 自动服务发布——服务发布者是一个应用程序(可能是服务自己本身),它使用注册机提供的服务发布 Web service 或者 API 把服务实体发布到注册机上。注册机可以通过审查 WSDL 文件定义来获得该服务的大部分信息。

3) 动态更新服务实体——除了自动发布,服务可能需要动态更新它在注册机中的服务定义和元数据,以便于注册机中的服务实体能与实际服务运行状态保持同步。

(2) 服务查询

1) 手工,面向用户的服务查询:用户和开发者需要能够通过基于 web 的用户界面浏览,搜索和审查服务和其他实体。

2) 动态,运行时服务查询:服务消费者可能需要在运行时使用注册机提供的查询 Web Services 接口发现服务。动态查询可以获取位置透明的服务,允许服务消费者在服务位置改变了的情况下,能够连接到服务提供者。

3) 持久化服务查询:在某些情况下,服务消费者可能想要保持对注册机上发布的某个特定发现实体的最新情况跟踪,服务消费者应该能够把该需求传送给发现实体,并能近实时的收到该变化的通知。

4) 除了查询机制,服务查询还应该有足够的表

达能力来支持基于名字和 ID 的查询,和基于任意元数据的复杂查询。

(3) 服务元数据管理

服务发现对服务提供者、服务本身和服务实体三者起到一个协调作用,对发布和查询来说,需要支持以下等级的元数据^[2-3]:

1) “白页”元数据,主要是基本的资源定义,如身份标识,名字,位置和接口定义等。

2) “黄页”元数据,主要指跟服务内容有关的元数据,描述了服务可以提供的功能等。这类元数据属性包括主题,关键字,服务类型,目录,时间和空间约束条件,或其它核心属性。

3) “棕页”元数据,这是指语义层面的元数据,描述了服务的业务/功能能力。这类元数据可以是定义良好的业务辞典、分类辞典、语义本体(ontologies)、XML schemas、领域数据模型或者业务规则等。这种元数据对上层业务应用消费者非常重要。

4) “绿页”元数据,主要指服务的访问能力属性,即访问该服务时所需要的技术和环境相关的元数据,这类元数据包括但不限于安全或 QoP (Quality of Protection) 需求, QoS 属性,传输协议等。

所有上述元数据都可以用 XML 表达。

1.3 发现服务协议

图 1 给出了通用的发现服务协议栈。

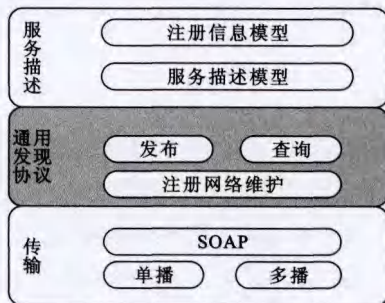


图 1 通用服务发现协议栈

Fig. 1 Protocol stack of universal service discovery

如图 1 所示,在 Web Services 技术体制中, SOAP 作为消息传输协议,同时需要单播和多播绑定,多播可以基于 SOAP-over-UDP 和 http over UDP。通过 SOAP-over-UDP 和 http over UDP 协议,只要在 IP 网络环境下,底层传输不管是有线信道、无线信道都可以支持。SOAP 之上是通用发现协议,可以执行服务发布、查询和发现的通用操作。由于服务注册节点要执行更多的功能,所以需要定义

一些协议轮廓信息,包括统计报告、注册节点列表信息的交换、上传服务分类方法以及服务注册信息模型扩展等。协议栈的最上层是服务描述模型和注册信息模型,其中服务描述模型可能同时存在多种,如基于 UDDI 和 ebXML Registry 的混合模型^[4]。

在 UDDI 服务发现标准体系下,服务消费者实现查询 API,其主要操作包括:通过 UUID 查找服务、通过 UUID 获取服务、通过类型查找服务、通过名称查找服务等。服务提供者实现服务发布的 API,其主要操作为发布、更新、删除服务发布信息及相关内容。

在实际工程应用中,由于仅依赖关键字匹配进行服务查询的查全率和查准率比较低,在具体实现中,可以进行元数据的扩展(上文中提到的棕页和绿页元数据),采用多种实现手段提高 UDDI 服务发现的智能性、高效性和准确性。如在服务元数据信息模型中采用 OWL-S 等语义本体描述语言和推理机制,通过服务关键字的匹配、语义的匹配和 QoS 属性(如服务的响应时间、可用性、可靠性、安全性、信誉度等)的匹配来实现基于服务 QoS 的、高效智能的服务查询和动态服务组合等^[5-7]。

1.4 发布与发现服务的信息模型

UDDI 注册机以 XML 的格式存储四类核心数据,这些数据定义了要使用某个 Web 服务所需的各种信息。这四类数据的层次关系如图 2 所示。

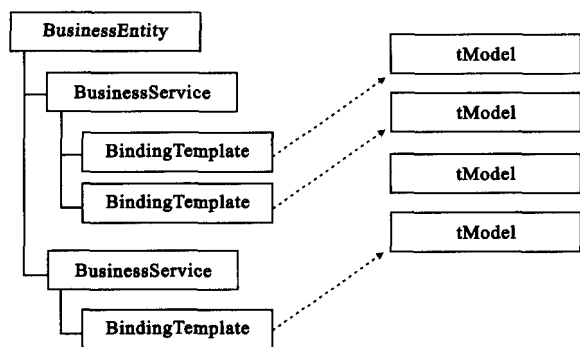


图 2 UDDI 的四种核心数据
Fig. 2 Four core data structures of UDDI

UDDI 有两类顶层数据。一个是业务实体(Business Entity),另一个是技术模型(tModel)。一个业务实体内包含若干个业务服务(Business Service),一个业务服务内又包含若干个绑定模板(Binding Template)。对于一个绑定模板而言,它又包含一个或多个技术模型的引用,从而跟某个技术

模型关联起来。

目前的信息模型研究热点在于采用本体来进行服务行为模型、多属性关系模型和协作上下文模型的定义。在语义化模型中,依据本体的形式化表示方法对服务的静态描述和动态交互模型进行统一的语义模型定义,从而为用户提供按需匹配和基于 QoS 评价选择的服务发现方法,以及基于协作上下文模型的服务组合方法^[8]。

基于 UDDI 的发布与发现服务定义了两类服务接口:一类是查询服务,另一类是发布服务。

(1) 查询服务

查询服务 InquiryService 用于查询 UDDI 注册机中存储的各类数据实体。InquiryService 主要包括对绑定模板、业务实体、服务实体和技术模型进行 find_xxx 和 get_xxxDetail 的操作。

(2) 发布服务

发布服务 PublishService 用于删除或更新 UDDI 注册机中存储的各类数据实体,此外还发布服务相关的安全操作。PublishService 主要包括对绑定模板、业务实体、服务实体和技术模型等数据的删除操作和保存操作,以及 discard_authToken、get_authToken、get_registeredInfo 等安全操作。

2 服务发布与发现在 SOA 中的应用

2.1 基于服务发布与发现的 SOA 系统集成

基于 UDDI 的典型 SOA 集成框架如图 3 所示。

服务提供者通过 UDDI 发布服务发布 Web Service 注册信息到 UDDI 注册机。

应用开发者到 UDDI 服务查询门户浏览或查找需要访问的服务。UDDI 服务查询门户发起服务查询请求,UDDI 查询服务返回匹配的服务实例或组合服务实例(必要时 UDDI 查询服务需要进行语义模型的访问、匹配和解析)。

开发者在开发中使用获得的目标服务实例的端点信息配置它的应用访问地址。配置好的应用被部署到门户服务器。当应用运行起来后,最终用户登录到该门户开始使用该应用。该应用作为 Web Service 消费者,绑定到目标服务实例,并在用户端发起 SOAP 请求,使用该服务提供的功能。

在该典型应用场景中,服务查询是在开发阶段发生的,被发现的服务信息是静态的、硬编码到服务消费者应用中的。

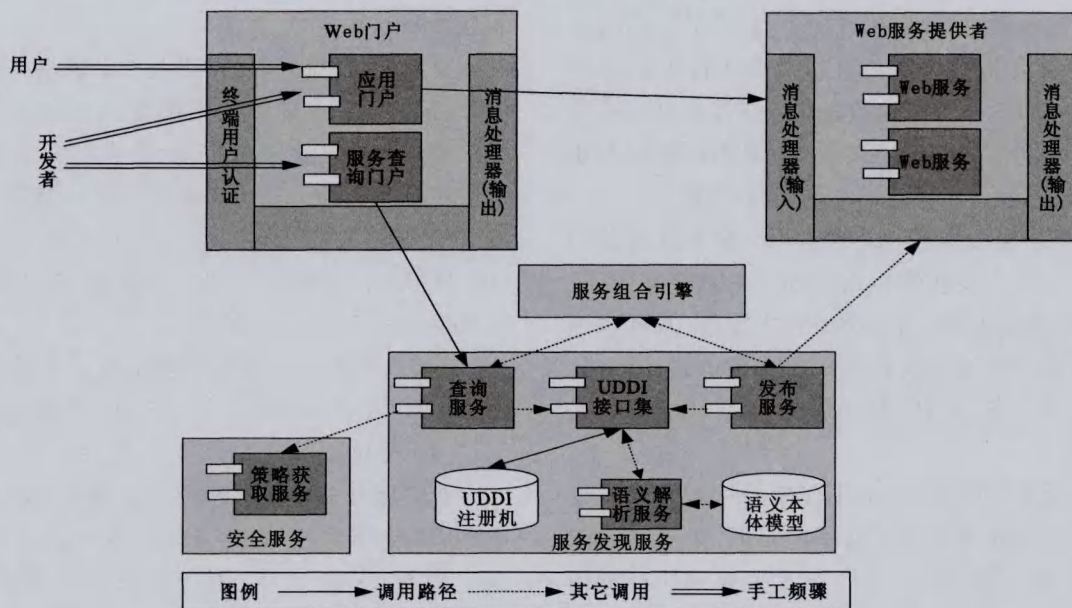


图3 基于UDDI的SOA集成框架

Fig.3 SOA integration framework based on UDDI

2.2 发布与发现服务与安全服务的集成

对于安全敏感的企业级应用来说,服务发现与服务安全紧密相关,这种相关性分为两个方面:一方面,服务发布和查询API和注册机本身需要安全机制和策略来进行保护;另一方面,在服务消费者和服务提供者之间要发现彼此的安全特性,才能建立信任关系。这需要在SOA架构实现中,满足以下需求^[2,3,9]:

(1) 保护发现服务接口

对于发布和查询来说,服务接口需要以下技术手段进行安全防护:

- 1) 建立发布者、查询者和发现服务提供者的身份标识。
- 2) 服务发布和查询的请求和响应是经过鉴别的,其消息完整性是经过检查的。
- 3) 请求和响应需要经过访问控制策略授权。
- 4) 发现实体是可信的。服务拥有者(通常是服务发布者)要能够保证所发布的实体可信。可采用数字签名等手段,以便服务消费者能对这些实体有某种程度的信任。

(2) 对敏感的发现实体的保护

注册机要与安全服务、策略进行集成,以采取措施保证服务注册机中的服务元数据包含的敏感信息不被未授权的团体得到。

2.2.1 保护服务接口

图4分析了使用SOAP消息句柄来保护UDDI服务注册中心查询服务的流程。

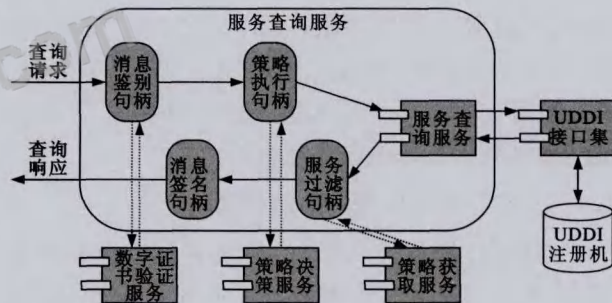


图4 安全的服务查询接口调用流程

Fig.4 Calling process of security service inquiry interface

消息鉴别句柄校验了查询请求的签名,Policy Enforcement Handler基于服务级别的授权策略对该请求授权,Message Signing Handler签名查询请求。另外,Service Filtering Handler被用来提供基于角色的数据级别的访问控制。

2.2.2 保护发现实体

另一方面的安全是要保证发布到UDDI注册机的发现实体的完整性和真实性。UDDI3.0允许在UDDI数据结构(如businessService, bindingTemplates及tModels等UDDI数据结构)中使用数字签名。服务查询者可以通过过滤查询,只请求经过签名的数据。这样服务查询者和提供者都可以信任发现实体数据的完整性和正确性。

2.3 发布与发现服务与管理服务的集成

管理服务是指对基于Web Services的SOA系统的管理,这对于企业级SOA系统的运行维护尤为

重要。管理服务以“管理者-代理模型”为基础,在 WSDL 中定义管理接口,将管理应用作为 Web 服务,用 WSDL 描述可管理 Web 服务,并且使用服务注册中心和 WSDL 来发现 Web 服务。因此管理 Web 服务的时候,必须遵守以下一些特定的原则:

1) 分离管理接口。Web 服务的发现和绑定过程是由接口来驱动的。在 WSDL 文件中接口被描述为端口类型,当在 UDDI 注册机上查询一个 Web 服务的时候,它的目标就是找到在 WSDL 文件中描述的接口,所以管理服务的管理操作应当通过分离的接口描述和发布 Web 服务接口来暴露给用户。

2) 通过运行时框架收集数据。Web 服务通过 SOAP 协议来调用和发送数据。SOAP 消息处理器和相关的 Web 服务基础结构形成了一个控制端,它应允许自动地采集管理信息。

3) 使用事件收集器。对一个管理系统而言,Web 服务需要为发生的一些重要事件发送信息:包

括故障事件、配置数据的改变、操作的调用等。

2.3.1 管理服务接口标准

服务管理基于 WSDM (Web Services Distributed Management) 标准来实现。WSDM 标准分为两部分: MUWS (Management Using Web Services) 和 MOWS (Management Of Web Services)。MOWS 提供了管理 Web 服务的标准接口定义。MOWS 基于 MUWS 标准定义的概念和系统,同时也添加了管理 Web 服务特别需要的资源和功能。MOWS 组件提供了支持远程管理 Web 服务的方法。

MOWS 标准提供了将 Web 服务作为资源来管理的相关定义,并且在 MOWS 标准中还使用了许多由 MUWS 标准定义的概念和结构,添加了管理 Web 服务所特别需要的资源和功能^[10-11]。

2.3.2 服务管理与发现服务集成

在对服务的管理中,服务管理与发现服务集成的实现框架如图 5 所示。

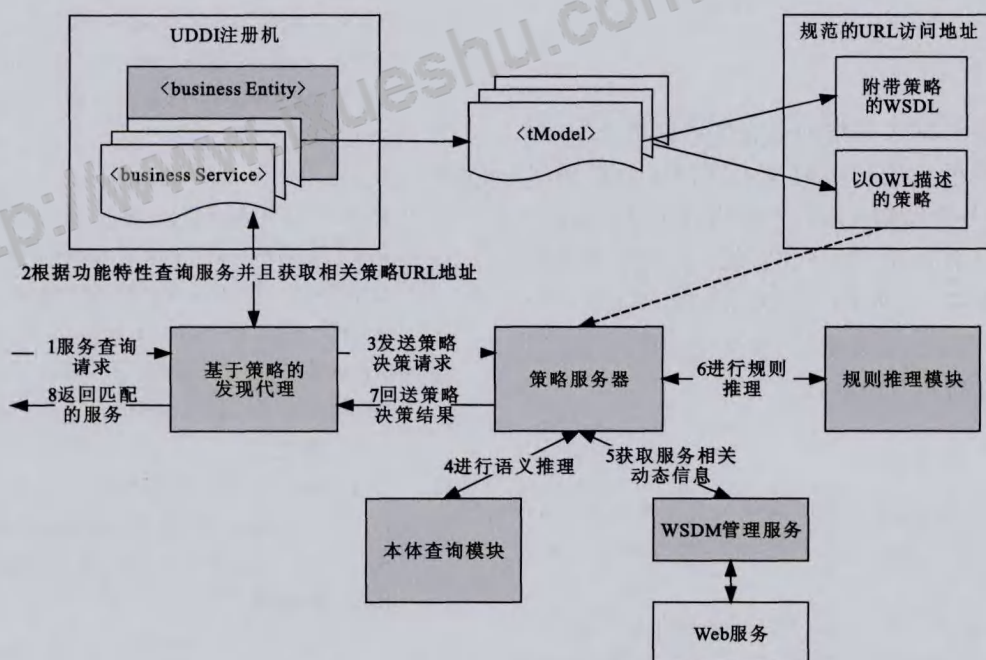


图5 服务管理与发现服务集成的实现框架

Fig. 5 Implementation framework of service management and service discovery integration

图5对标准的基于UDDI的服务查询方式进行了基于策略和语义本体的模型和功能扩展。服务策略主要用于描述服务的非功能的、静态的状态或能力,如QoS级别等。其中,主要组件是基于策略的发现代理、策略服务器、本体查询模块、规则推理模块以及WSDM管理服务,即动态数据采集模块。其

运行过程如下:

基于策略的发现代理负责接收来自服务查询客户端的请求,发现代理根据服务的功能需求查询UDDI注册机,得到服务的WSDL的访问地址。根据地址获取到WSDL后,提取WSDL中以扩展WS-Policy形式提供、以OWL描述的服务策略地址,发

现代代理将来自客户端的请求策略和通过 UDDI 查询到的服务策略地址组合成策略决策请求发往策略服务器^[12]。

策略服务器收到策略决策请求后,从提供策略的地址下载策略,并将得到的策略(包括请求策略和服务策略)发往本体查询模块进行本体推理,最终得到基于策略相关元素组成的策略集合,反馈给策略服务器。然后,策略服务器根据策略集合中对服务 QoS 状态等信息或能力的需求,访问 WSDM 管理服务,获取服务的 QoS 状态等动态管理参数。

策略服务器将从本体查询模块得到最终策略集合,以及从 WSDM 管理服务中得到服务的 QoS 等动态参数转发给规则推理模块进行规则推理,并由规则推理模块返回推理结果。策略服务器根据推理结果得到符合服务请求的服务地址,将其返回给基于策略的发现代理。基于策略的发现代理再向服务的请求者反馈得到的服务 WSDL 地址,服务请求者可以根据返回的 WSDL 地址进行服务的调用了。

3 结 语

文中对基于 SOA 架构的企业级信息系统中,服务发布与发现的应用和实现技术进行了分析和研究,介绍了服务发布和发现技术的实现策略和方法,并在相关项目中进行了验证和实现。后续将继续对基于语义的服务元数据的管理和策略模型实现进行深入研究,实现基于服务质量、安全特性和访问控制策略的服务发布和发现机制。

参考文献:

- [1] 魏强,金芝,李戈,等.物联网服务发现初探:传统 SOA 的可行性和局限性[J]. 计算机科学与探索,2013,7(02):97-110.
WEI Qiang, JIN Zhi, LI Ge. Preliminary Study of Service Discovery in Internet of Things: Feasibility and Limitation of SOA[J]. Journal of Frontiers of Computer Science and Technology, 2013, 7(02): 97-110.
- [2] OASIS. uddi_v3, UDDI Spec Technical Committee Draft, Dated 20041019, UDDI version 3.0.2[S]. America: OASIS Open, 2004.
- [3] Booz A H. Service Discovery Core Enterprise Services (CES) Architecture[R]. DISA: America, 2004.
- [4] Tommy G. Assessing Dynamic Service Discovery in the Network Centric Battlefield[C]//Military Communications Conference, MILCOM 2007, IEEE. Orlando, FL,

USA: [s. n.], 2007.

- [5] 魏娟杰. 基于 Web 服务发现的企业应用集成研究[D]. 东营: 中国石油大学, 2009.
WEI Juan-jie. Research on Enterprise Application Integration Based on Web Service Discovery[D]. China University of Petroleum: China, 2009.
- [6] 刘莉平. 动态 Web 服务组合关键技术研究[D]. 长沙: 中南大学, 2011.
LIU Li-ping. Research on Key Technologies of Dynamic Web Service Composition[D]. China: Central South University, 2011.
- [7] 刘昌鑫, 欧阳春娟, 谭云兰. 基于 Web Service 的 QoS 本体研究[J]. 通信技术, 2012, 43(02): 160-162.
LIU Chang-xin, OUYANG Chun-juan, TAN Yun-lan. Analysis on QoS Ontologies for Web Services[J]. Communications Technology, 2012, 43(02): 160-162.
- [8] 曲明. 基于本体的服务发现与组合方法研究[D]. 吉林: 吉林大学, 2012.
QU Ming. Research on Ontology-based Service Discovery and Composition[D]. China: Jilin University, 2012.
- [9] OASIS. {WSS: SOAP Message Security}-1.0. Web Services Security: SOAP Message Security 1.0 (WS-Security 2004) OASIS Standard 200401[S]. America: OASIS Open, 2004.
- [10] OASIS. cd-wsdm-muws-part1-1.0, Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 1 Committee Draft, January 11th 2005[S]. America: OASIS Open, 2004.
- [11] OASIS. cd-wsdm-mows-1.0, Web Services Distributed Management: Management of Web Services (WSDM-MOWS) 1.0 Committee Draft[S]. America: OASIS Open, 2004.
- [12] NATENAPA S, TWITTIE S, KUNAL V. On Using WS-Policy, Ontology and Rule Reasoning to Discover Web Services[C]//IFIP International Conference. Thailand: Bangkok, 2004.

作者简介:



张锋军(1975—),男,高级工程师,主要研究方向为网络管理,软件工程;

ZHANG Feng-jun (1975—), male, senior engineer, majoring in network management, software engineering.

陈宇雷(1971—),女,高级工程师,主要研究方向为网络管理,软件工程。

CHEN Yu-lei (1971—), female, senior engineer, majoring in network management, software engineering.

论文降重，论文修改，论文代写加微信:18086619247或QQ:516639237

论文免费查重，论文格式一键规范，参考文献规范扫二维码：



[相关推荐：](#)

[服务发布和发现在SOA架构中的应用分析](#)

[基于服务来建模和架构SOA](#)

[SOA面向服务架构在集装箱EDI平台中的应用](#)

[美国农业信息工作](#)

[面向服务架构\(SOA\)在电力行业中的应用](#)

[面向服务架构SOA](#)

[面向服务的架构\(SOA\)在电子商务中的应用](#)

[面向服务架构\(SOA\)研究](#)

[SOA体系架构的应用研究与分析](#)

[Soa架构与安全分析](#)