

文章编号: 1002-0640(2012) 05-0010-04

## 分布式 UDDI 的数据一致性优化模型\*

张燕琴

(解放军 61135 部队, 北京 100842)

**摘要:** UDDI 注册中心本质上是一个存储着 Web 服务注册信息的数据库, 针对分布式存储与检索体系架构中出现的**数据不一致性的问题**, 提出了一种分布式 UDDI 的数据一致性优化模型。首先分析了分布式 UDDI 及 UDDI 的核心数据模型, 以及在分布式环境下的**异构数据源问题**。在此基础上, 分析了数据的一致性问题, 构建了 UDDI 节点—数据模型间的可达性矩阵, 最后研究了分布式 UDDI 的一致性数据存储策略。该优化模型能有效减少存储冗余和检索网络带宽的消耗。

**关键词:** 分布式 UDDI, 异构数据, 一致性, 优化模型

中图分类号: TP393

文献标识码: A

## A Data Consistency Optimized Model in Distributed UDDI

ZHANG Yan-qin

(Unit 61135 of PLA, Beijing 100842, China)

**Abstract** Essentially speaking, UDDI registry center is a data of Web service registration information. To solve the problems of data inconsistency in distributed storage and search architecture, the paper brings forward a data consistency optimized model in distributed UDDI. Firstly, the paper analyzes not only core data model but also isomorous data source problem in distributed surroundings. After that, data consistency problem is analyzed, and UDDI node-data model accessible matrixes are constructed. Finally, the consistency storage strategy in distributed UDDI is studied. The optimized model can reduce the storage redundancy and search bandwidth consumption.

**Key words** distributed UDDI, isomorous data, consistency, optimized Model

### 引言

UDDI 注册中心是 Web Service 技术体系中的重要组成部分, UDDI 的数据特征和操作特征要求能够将异构数据按照层次模型组织起来, 并能对以查询为主的数据操作作出灵活、高效的响应。传统的关系数据库在处理上述问题时不仅效率和灵活性较低, 而且实施成本较高。

随着网络技术的发展, 分布式体系结构在越来越多的信息系统中得到应用。但是, 分布式系统的分

散性和异构性给实现更大范围内的资源共享和网络管理带来了许多困难。在分布式 UDDI 的环境下, 随着服务数量的增加和注册服务的更新, 注册数据间的关系变得复杂, 正确性和一致性都极易受到破坏, **很容易产生数据的冗余及一致性问题, 导致存储空间浪费和网络带宽的过度消耗**。数据的不一致性问题更严重的会导致服务调用的失败以及服务响应的异常。本文提出一种分布式 UDDI 的数据一致性优化模型, 可以减少数据存储的冗余及检索响应时间。

### 1 分布式 UDDI 的数据一致性问题

#### 1.1 分布式 UDDI 及 UDDI 数据模型

UDDI 注册中心本质上是一个存储着 Web 服务注册信息的数据库, 为用户提供相应的查询检索功能。目前的数据库多采用传统的关系型数据库。虽

收稿日期: 2011-03-21

修回日期: 2011-05-25

\* 基金项目: 国家自然科学基金 (70601036); 装备预研基金资助项目 (51306040101)

作者简介: 张燕琴 (1966-), 女, 山西太原人, 工程师, 研究方向: 遥测遥控。

然关系型数据库在数据处理上有强大的能力,但从 UDDI 的数据结构特点、操作模式和实施成本等方面来考虑,层次化可扩展的分布式技术更符合 UDDI 中心的需求

UDDI 注册中心是所有提供公共 UDDI 注册服务的站点的通称。它是一个逻辑上的统一体,在物理上则是以分布式系统的架构实施的,而不同站点之间是采用 P2P 架构,因此访问其中任意一个站点就基本等于访问了 UDDI 注册中心。UDDI Operator Site (UDDI 注册中心操作入口站点,简称 UDDI 操作入口)是 UDDI Registry 中每一个对等节点,对 UDDI Operator Site 的查询所获得的结果是覆盖全 UDDI 注册中心中的信息的,信息查询无需身份认证;而在 UDDI Operator Site 上进行信息发布则必须使用该 UDDI Operator Site 自身的用户方能实施,同时以后的更新、删除都必须通过这个 Operator Site,并使用初始发布时使用的用户进行权限认证。

UDDI 协议定义了由多种数据模型构成的注册中心的构成规范,以及接收其他 Web 服务的发布与发现的 Web 服务接口规范等内容。UDDI 在逻辑上分为两部分:商业注册和技术发现。前者是用来描述企业及其提供的 Web 服务的一份 XML 文档;后者则定义了一套基于 SOAP 的注册和发现 Web Service 的编程接口。这两部分的框架全由 XML Schema 定义。凡是实现 UDDI 规范的站点被称为 UDDI 操作入口站点,这些站点通过复制 (Replicate) 机制保持彼此间的内容同步。

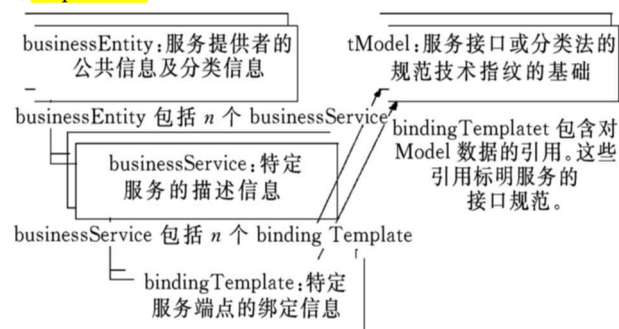


图 1 UDDI 核心数据模型

UDDI 协议规范使用 XML Schema 定义了用于注册中心的数据类型,这几种类型包含了客户访问某服务系统所需的 Web 服务技术信息。

UDDI 核心数据模型包括以下 6 项<sup>[1]</sup>: 商业实体信息 (businessEntity)、服务信息 (businessService)、绑定信息 (bindingTemplate)、规范描述指针和技术标志 (tModel)、关联信息 (publisherAssertion)、订阅信息 (subscription)。前 4 项为 UDDI 协议规范的

核心数据结构,它们包括了服务的主要信息和分类信息,其相互关系如图 1 所示。发布者声明是用来声明多个实体信息之间的相互关系,比如同一实体内的子实体。操作信息包含了系统创建核心数据结构的操作信息,比如类型和时间等,用来记录核心数据结构的操作信息。

## 1.2 异构数据源

在 UDDI 规范 V2 中,所有的 UDDI Registry 都是同级的关系,它们之间形成一个环状。而它们之间的数据是通过一个安全通道进行复制,最终是要使所有 UDDI Registry 的数据完全相同。而达到服务用户在发现服务的时候,无论通过任一个 UDDI Registry 查询的数据都是相同的。目前几大 UDDI 注册商之间的数据复制是每 24 h 复制一次。

V3 规范提出了“一次注册,随处发现”。即在一个节点上发布注册服务后,在其他任意一个节点上都可以发现该服务。这就需要借助于节点间的数据同步,以获得统一的数据资源。为实现这一目标,目前有以下几种方案:一是所有节点使用统一的集中式数据库,节点都在服务器里查询客户请求的资源,但是集中式数据库容易成为瓶颈,特别是在存在大量的服务请求的情况时,会严重影响系统性能;二是查询分发即在本节点找不到请求者所需服务的时候,转而向其他节点转发服务请求,但它会使得网络中的冗余数据的传输量大大增加;三是借助第三方中转更新数据来实现节点的数据同步,它的问题同方案一;四是传统复制方案,这是一种完全的数据复制,缺点是复制代价太高,没有考虑到数据的有效性和网络环境的复杂性。

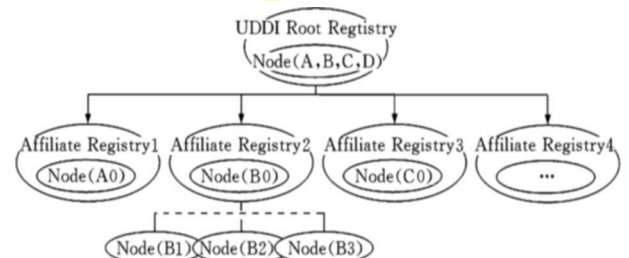


图 2 UDDI V3 规范拓扑结构

V3 规范把原来 UDDI 的平行结构,设计成为一种层次型的结构,这便于全球分布式 UDDI 的管理。在顶层设立了根节点,其作用是给其他的 Affiliate Registry 分配全球唯一的标识符,同时也赋予它们生成标识符的权力,使全球数据唯一和统一管理。但根节点的功能仅限于对标识符的分配和管理,并不象分布式数据库的中心数据库那样对主要的数据内容进行总的调配和管理,如图 2 所示,这是一个 Node

的拓扑局部复制图, A B C D 构成了一个 Root 下的一个 Affiliate Registry; 而 Node 下面又由子 Node 构成, 如 B0 B1 B2 B3 构成了 Node B, B0 是 Node B 的代表, 负责和上层 Node 进行数据复制。

## 2 一种分布式 UDDI 的数据一致性优化模型

本文研究一种新的分布式 UDDI 的数据一致性优化模型, 首先分析了分布式环境下的数据一致性问题, 在此基础上构建了 UDDI 节点——数据模型间可达性矩阵, 最后分析了分布式 UDDI 的一致性数据存储策略。

### 2.1 数据的一致性问题

随着网络的发展, 对现存系统中数据共享的需求不断增加, 因此迫切需要一种途径来集成管理不同数据源的数据, 并提供给用户一个统一的界面来访问所有数据源。然而分布在异构环境中的各种数据源有着不同的数据格式、存储方式、访问控制策略、数据模型、操纵语言、数据语义。同时, 由于这些数据源的自治性, 其中数据的可共享性、共享方式、共享内容等也可能随时发生变化<sup>[2]</sup>。

UDDI 是一个统一体, 但在物理上则以分布式系统架构实现, 不同的站点之间采用对等网络结构实现, 因此访问其中任意一个站点就等同于访问 UDDI 注册中心的数据。针对 UDDI 注册中心服务器的崩溃和不正常运行, 可以为服务器配置备用服务器。备用服务器完全复制了第一服务器中 UDDI 注册中心的数据, 并与第一服务器的注册中心同步更新数据以保证一致性。当主服务器不正常运行时, 备用服务器执行第一服务器 UDDI 注册中心的工作。当主服务器恢复正常运行后, 复制备用服务器 UDDI 注册中心的数据, 重新接管 UDDI 注册中心的工作。但当网络应用不断增大, 就需要越来越多的服务器对注册的服务信息进行管理, 简单地靠完全复制数据将不能够满足越来越庞大的网络需求, 这就需要分布式的 UDDI 配置和良好的数据一致性优化管理策略。

下面是分布式环境下 4 个 UDDI 注册中心节点的示例代码

```
public static String getNeighbourUddiOne() throws
IOException
{
    if(is Updated)
    {
        return neighbouruddione;
```

```
    }
    else
    {
        readProperties();
        return neighbouruddione;
    }
}

public static String getNeighbourUddiTwo() throws
IOException
{
    {.....}
}

public static String getNeighbourUddiThree() throws
IOException
{
    {.....}
}

public static String getNeighbourUddiFour() throws
IOException
{
    {.....}
}
```

UDDI 协议是面向服务体系结构协议栈中的重要成员, 并且已经成为发布标准或规范及遵循这些标准或规范的应用服务的一个统一描述、发现与集成的工具。分布式 UDDI 中的数据模型主要存在以下 3 种不一致性问题。R 表示 UDDI 根节点, N 表示 UDDI 节点。BE 表示 BusinessEntity, BS 表示 BusinessService, BT 表示 Binding Template, TM 表示 TModel 技术指纹数据。

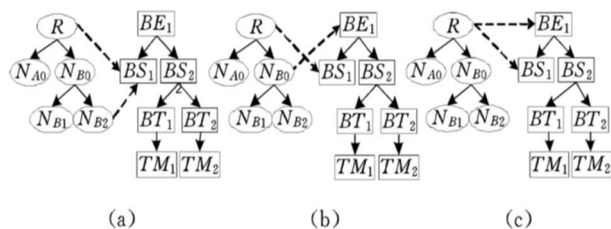


图 3 UDDI 节点数据模型不一致的情况

根据图 1 的 UDDI 核心数据模型和图 2 中 V3 规范拓扑结构, 设计图 3 表示 UDDI 节点和数据模型的层次分解图<sup>[4-5]</sup>, 图 3 中的虚线箭头表示了节点存储数据模型的情况。图 3(a) 中服务数据同时由具有父子关系的 UDDI 节点 R 和 N<sub>B2</sub> 管理; 图 3(b) 表示父业务实体数据 BE<sub>1</sub> 由子节点 N<sub>B0</sub> 管理, 而子服务数据 BS<sub>1</sub> 则由根节点 R 管理; 图 3(c) 表示父业务实体数据 BE<sub>1</sub> 和子服务数据 BS<sub>1</sub> 同时由根节点 R 管理。

### 2.2 构建 UDDI 节点——数据模型间可达性矩阵

该部分的主要任务是将分布式 UDDI 节点的层次分解关系、数据模型的层次分解关系以及数据模型与 UDDI 节点的存储关联关系以矩阵的形式进行表示, 该过程分为两步:

(1) 将节点 (或数据) 之间的父子关系用邻接矩

阵表示,由邻接矩阵建立可达性矩阵,可获得节点(或数据)之间的遗传关系,详细过程如下所示:

根据 UDDI 节点的逻辑部署关系确立节点间的层次分解关系,由此可构建表示 UDDI 节点父子关系的邻接矩阵  $M_{on}$ , 矩阵  $M_{on}$  的行和列都是分布式 UDDI 存储节点,而且节点的排列顺序相同,为表述方便,用  $N_i (i \in N)$  依次表示各节点,矩阵  $M_{on}$  的元素记为  $(M_{on})_{ij}$  其中  $i, j \in N$ . 则有:

$(M_{on})_{ij} =$

- 1, 位于第  $i$  行的节点  $N_i$  是第  $j$  列节点  $N_j$  的父结点
- 0, 位于第  $i$  行的节点  $N_i$  不是第  $j$  列节点  $N_j$  的父结点

邻接矩阵  $M_{on}$  表示节点之间的父子关系,然后在邻接矩阵的基础上通过 Warshall 算法计算出矩阵  $M_{on}$  的可达性矩阵  $MA_{on}$ , 在  $MA_{on}$  中则有:

$(MA_{on})_{ij} =$

- 1, 节点  $N_i$  到节点  $N_j$  可达, 表示节点  $N_i$  是节点  $N_j$  的祖先
- 0, 节点  $N_i$  到节点  $N_j$  不可达, 表示节点  $N_i$  不是节点  $N_j$  的祖先

当  $i = j$  时,  $(MA_{on})_{ij}$  的值恒为 0, 表示任意节点到自身都不可达

同理, 根据 UDDI 核心数据模型的逻辑关系, 建立表示数据父子关系的邻接矩阵  $M_{od}$  和节点之间可达性的矩阵  $MA_{od}$ , 并用  $D_i$  依次表示核心数据结构。

(2) 建立 UDDI 节点与核心数据的布尔 (0 或 1) 关联矩阵

根据 UDDI 核心数据模型的数据在分布式 UDDI 的存储关系, 可建立表示 UDDI 节点与核心数据模型存储关系的矩阵  $M_{on\_od}$ . 其中矩阵的行表示 UDDI 节点, 节点的排列顺序要与矩阵  $M_{on}$  相同, 矩阵的列表示核心数据, 数据的排列顺序要与矩阵  $M_{od}$  相同. 矩阵  $M_{on\_od}$  的元素记为  $(M_{on\_od})_{ij}$ , 则有:

$(M_{on\_od})_{ij} =$

- 1, 节点  $N_i$  管理数据  $D_j$ , 也表示  $N_i$  到  $D_j$  一步可达
- 0, 节点  $N_i$  不管理数据  $D_j$ , 也表示  $N_i$  到  $D_j$  一步不可达

因此, 矩阵  $M_{on\_od}$  也可称为分布式 UDDI 节点到核心数据模型的一步可达矩阵

### 2.3 分布式 UDDI 的一致性数据存储策略

在文献 [3] 中表明, 很少有服务提供者会在注册服务信息后会主动更改服务信息, 按照传统方案不加区分地完全地复制数据, 效率显而易见. 如果仅仅只把有变化的服务信息通知给与它有可达关系的链路节点, 无疑就节省了时间和资源, 此后节点间的数据同步就不需要复制所有数据, 只需要在优化链路上传输这些产生变化了的数据即可. 图 4 为一个分布式 UDDI 节点复制拓扑结构的示例图.

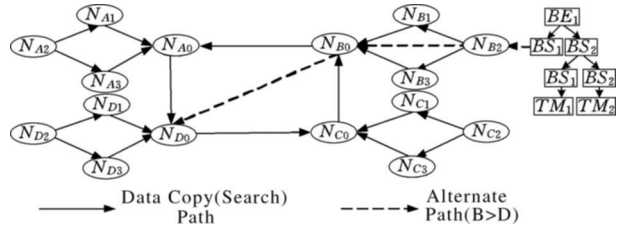


图 4 分布式 UDDI 节点复制拓扑结构图

图 4 拓扑结构中 UDDI 节点与节点的局部可达关系矩阵  $(NB_0, NB_1, NB_2, NB_3)$  以及局部可达关系矩阵  $(NA_0, NB_0, NC_0, ND_0)$  分别为:

$$MA_{on1} = \begin{matrix} & NB_0 & NB_1 & NB_2 & NB_3 \\ NB_0 & 0 & 1 & 0 & 0 \\ NB_1 & 1 & 0 & 0 & 0 \\ NB_2 & 1 & 1 & 0 & 1 \\ NB_3 & 1 & 0 & 0 & 0 \end{matrix}$$

$$MA_{on2} = \begin{matrix} & NA_0 & NB_0 & NC_0 & ND_0 \\ NA_0 & 0 & 0 & 0 & 1 \\ NB_0 & 1 & 0 & 0 & 1 \\ NC_0 & 0 & 1 & 0 & 0 \\ ND_0 & 0 & 0 & 1 & 0 \end{matrix}$$

核心数据模型之间  $(BE_1, BS_1, BS_2, BT_1, BT_2, TM_1, TM_2)$  以及与核心数据与节点之间  $(BE_1, BS_1, BS_2, BT_1, BT_2, TM_1, TM_2, NB_0, NB_1, NB_2, NB_3)$  的局部可达关系矩阵为:

$$MA_{od} = \begin{matrix} & BE_1 & BS_1 & BS_2 & BT_1 & BT_2 & TM_1 & TM_2 \\ BE_1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ BS_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ BS_2 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ BT_1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ BT_2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ TM_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ TM_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$$M_{on\_od} = \begin{matrix} & NB_0 & NB_1 & NB_2 & NB_3 \\ BE_1 & 0 & 0 & 0 & 0 \\ BS_1 & 0 & 0 & 1 & 0 \\ BS_2 & 0 & 0 & 0 & 0 \\ BT_1 & 0 & 0 & 0 & 0 \\ BT_2 & 0 & 0 & 0 & 0 \\ TM_1 & 0 & 0 & 0 & 0 \\ TM_2 & 0 & 0 & 0 & 0 \end{matrix}$$

根据 UDDI 节点和数据模型的存储关系矩阵  $M_{on\_od}$ , 可确立新的数据复制过程正如图 4 所示. 如

(下转第 19 页)

表 3 基于多资源匹配元任务插入的启发式聚类算法结果

聚类任务	包含任务	时间窗口		观测角 (°)	匹配资源
		开始时间	结束时间		
$C_{1.5.1}$	$t_1, t_3$	17: 22 10. 31	17: 23 26. 84	22. 0	$s_1$
$C_{1.5.2}$	$t_5, t_7, t_8, t_{21}$	17: 24 33. 29	17: 27 59. 15	12. 9	$s_1$
$C_{1.5.3}$	$t_{20}, t_{24}$	17: 31 28. 34	17: 34 09. 66	- 6. 5	$s_1$
$C_{1.5.4}$	$t_2, t_{22}$	17: 37 12. 58	17: 38 37. 08	- 9. 4	$s_1$
$C_{1.5.5}$	$t_{15}, t_{27}$	17: 53 44. 68	17: 55 07. 88	- 18. 1	$s_1$
$C_{2.8.1}$	$t_6, t_{29}$	00: 50 36. 69	00: 52 16. 94	- 10. 2	$s_2$
$C_{2.8.2}$	$t_9, t_{11}, t_{12}, t_{30}$	00: 54 02. 71	00: 57 33. 69	- 14. 5	$s_2$
$C_{2.8.3}$	$t_{14}, t_{18}$	01: 13 02. 43	01: 16 59. 53	11. 4	$s_2$
$C_{2.8.4}$	$t_{18}, t_{19}, t_{25}$	01: 43 52. 46	01: 46 22. 55	- 3. 7	$s_2$
$C_{2.8.5}$	$t_{17}, t_{23}$	01: 48 29. 54	01: 50 46. 53	- 6. 9	$s_2$

5 结束语

本文提出了多卫星成像侦察任务聚类问题,分析了多卫星任务聚类的特点,界定了任务聚类问题的输入输出要素,在单星任务聚类方法的基础上建

立了多星成像侦察任务聚类数学规划模型 针对多资源匹配元任务的特点,提出了基于元任务插入的多资源聚类启发式算法,并结合一个实例验证了算法的可行性,任务聚类后,减少了任务规模,提高了卫星资源的利用效率。

参考文献:

[1] Cohen R H. Automated Spacecraft Scheduling-the Aster Example [ J]. Jet propulsion Laboratory: Ground System Architectures Workshop, 2002, 85 (2): 17-18.

[2] 薛波.卫星战术应用分解及资源匹配研究 [ D]. 长沙:国防科技大学, 2008.

[3] 许语拉,徐培德,王慧林,等. 基于团划分的成像侦察任务聚类方法研究 [ J]. 运筹与管理, 2010, 19 (4): 143-149.

[4] 徐雪仁,宫鹏,黄学智,等.资源卫星(可见光)遥感数据获取任务调度优化算法研究 [ J]. 遥感学报, 2007, 11( 1): 109-114.

[5] 王均.成像卫星综合任务调度模型与优化方法研究 [ D]. 长沙:国防科技大学, 2007.

[6] 白保存.考虑任务合成的成像卫星调度模型与优化算法研究 [ D].长沙:国防科技大学, 2008.

(上接第 13 页)

节点  $N_{B2}$  存储有服务数据  $B_{S1}$ ,按照  $V3$  的全局复制策略,从数据源  $B_{S1}$  到检索入口  $N_{D0}$ ,可能的复制路线是 $B_{S1} \rightarrow N_{B2} \rightarrow N_{B1} \rightarrow N_{B0} \rightarrow N_{A0} \rightarrow N_{D0}$ ,根据新的存储关系矩阵  $MON_{OD}$ ,由  $MON_{OD}= 1$  的一步可达关联路径可确定优化的存储与检索路径,减少存储空间和网络带宽的消耗。那么新的复制路线如  $B_{S1} \rightarrow N_{B2} \rightarrow N_{B0} \rightarrow N_{D0}$ ,比全局复制策略减少两跳的长度。

3 结束语

本文针对分布式 UDDI环境下,存储与检索体系架构中出现的数据不一致性的问题,提出了一种数据一致性优化模型<sup>[6]</sup>。首先分析了分布式 UDDI 及 UDDI的核心数据模型,以及在分布式环境下的异构数据源问题 在此基础上,分析了数据的一致性问题,构建了 UDDI节点——数据模型间的可达性矩阵,最后研究了分布式 UDDI的一致性数据存储策略,该优化模型能有效减少存储冗余、数据传输量和检索网络带宽的消耗。

参考文献:

[1] Tom Bellwood, Luc Cé ment. UDDI Spec Technical Committee Specification, UDDI Version 3. 0 [ EB/OL ]. <http://uddi.org/pubs/ud2-di2v3.002published220020719.htm>, 2002 2072-19.

[2] Kim S M, Rosu M C. A Survey of Public Web Services. Fieldman S I, Uretsky M, Najork M, et al. Proc of the 13 th Int’ l Conf on the WorldWideWeb [ C ]//New York: ACM Press, 2004.

[3] Kim S M. Population of PublicWeb Services [ EB/OL ]. [http://nclab.kaist.ca.kr/~smkim/ws\\_survey/index.html](http://nclab.kaist.ca.kr/~smkim/ws_survey/index.html), 2005.

[4] Moreau L, Ananthakrishnan R, Wilde M, et al. Exposing UDDI Service Descriptions and Their Metadata Annotations as WS-Resources [ C ]//7th IEEE/ACM International Conference on Grid Computing, 2006.

[5] Peterkin R, Solomon J, Ionescu D. Role Based Access Control for UDDI Inquiries [ C ]//Second International Conference on Communication Systems Software and Middleware ( COMSW ARE 2007), IEEE, January 2007.

[6] 柴晓路. Web 服务架构与开放互操作技术 [ M ].北京:清华大学出版社, 2002.