
国家环境信息与统计能力建设项目

环境信息服务注册技术规定

（征求意见稿）

《环境信息服务注册技术规定》编制组

2010年7月

目 录

1 适用范围	1
2 术语和定义	1
3 缩写	1
4 服务注册内容	1
5 服务注册范围	2
6 服务注册模式	3
7 服务注册实现模型	4
8 服务注册中心	6
附 录 A（规范性附录） 基于 W3C UDDI/WSDL 的环境信息服务注册参考实现	9
A.1 UDDI 概述	9
A.2 UDDI 服务注册体系结构	9
A.3 UDDI 分层 Web 服务协议栈	9
A.4 UDDI 服务注册接口实现	10
A.5 UDDI 服务注册数据结构	10
附 录 B（规范性附录） 服务元数据 XML 模式(XMLSchema)	12
附 录 C（资料性附录） 服务元数据示例	13
C.1 服务注册消息 XML 实例	13
C.2 服务注册程序代码实现	14

环境信息服务注册技术规定

1 适用范围

本技术规定规定了环境信息服务注册的内容、范围、实现模型、操作模式，规定了服务注册中心的规格要求、部署与管理，提供了对环境信息服务注册的参考实现。

适用于环境信息系统的设计与开发、环境信息服务的注册，以及环境信息服务注册中心的部署与管理维护。

2 术语和定义

本部分对本技术规定中涉及的术语进行一致性定义。

2.1 服务 Service

实体通过接口提供的明确的功能。

2.2 服务注册 Service Register

对服务的基本功能、接口规格、消息及参数进行定义注册，通过服务资源的发布—发现—访问机制，实现服务资源共享。

2.3 元数据 Metadata

关于数据的数据。

2.4 服务注册中心 Service Registry

对服务统一注册管理的中心，是服务发布者和服务使用者的中介。

2.5 环境信息服务 Environmental Information Service

在分布式网络环境下，环境信息系统为实现互联互通、信息共享、业务交互所提供的具有一定功能的信息服务。

2.6 注册服务 Resisted Service

已在服务注册中心进行信息注册的服务资源。

2.7 环境信息服务分类 Environmental Information Service classification

把具有某种共同属性或特征的环境信息服务归并在一起，通过属性或特征来对环境信息服务进行的归类，据以建立环境信息服务分类体系，具体实现环境信息服务采集、管理、服务、使用等，以便有序管理和开发利用环境信息服务资源。

2.8 标识符 Identifier

用于标识或命名一个数据项目并可能指出该数据确定性质的一个或一组字符。

3 缩写

GUID	全球唯一标识符 (Globally Unique Identifier)
HTTP	超文本传输协议 (Hypertext Transfer Protocol)
SOA	面向服务的体系架构 (Service-Oriented Architecture)
SOAP	简单对象访问协议 (Simple Object Access Protocol)
UDDI	统一描述、发现和集成协议 (Universal Description Discovery and Integration)
UML	统一建模语言 (Unified Modeling Language)
URI	统一资源标识符 (Uniform Resource Identifier)
WSDL	服务描述语言 (Web Services Description Language)
WSFL	Web 服务流程语言 (Web Services Flow Language)
XML	可扩展标记语言 (Extensible Markup Language)

4 服务注册内容

4.1 注册服务类型

环境信息服务包含基于 HTTP 和 SOAP 的 Web Service，以及以 URI 等形式存在的其他服务资源。

4.2 注册服务分类

环境信息服务可按照行业、主题、服务资源形态等方法进行服务分类。依据服务分类对注册服务进行识别、导航和定位。

注册服务分类采用开放式的管理，允许各部门根据本部门的特点细分和增加类目，编制适合本部门的细分规则。

4.3 注册服务元数据

环境信息服务元数据是对服务的描述信息，包含对服务资源的详细说明，涉及服务提供者的基本情况、服务内容、用途，以及服务定位和访问信息。主要由以下各部分组成：

- a) 服务提供者的基本信息。包含服务提供者名称、描述、联系人、提供者分类属性、所在业务领域等信息。
- b) 服务的基本信息。包含服务名称、描述、标识符、用途、发布日期、关键字、在线资源链接地址、服务地址、服务类型、版本等信息。
- c) 服务访问接口的描述。包含接口标识、支持的操作、数据类型、绑定规则以及访问协议等信息。

4.4 注册服务标识

注册服务标识符应采用统一规则进行标识，满足完整性、唯一性，在服务的生命周期内保持不变。标识符编码方法主要有：

- a) 服务标识符由前段码和后段码组成。
前段码统一分配管理，具有唯一性，用以标识各级环境信息服务的管理者或提供者。
后段码由拥有前段码的单位在注册服务元数据时自行管理和分配，用来对所管理或拥有的服务进行唯一标识，必须保证本级范围内的唯一性。
- b) 服务标识符为全球唯一标识符。采取GUID等方式对服务进行全球唯一标识。
- c) 其他。

5 服务注册范围

5.1 业务应用系统适用范围

服务注册适用于分布式、异构环境下各级部门的环境信息化应用，满足跨部门、跨行业以及面向公众的信息共享、业务协同，实现信息系统级别的互联互通。

- a) 环境管理业务应用系统。涉及环境管理过程中各环保部门之间，或环保部门与管理对象之间的业务应用系统。包含了建设项目管理系统、环境统计系统、环境监测管理系统、污染源监控管理系统、监督与执法管理系统、生态保护管理系统、核安全与辐射管理系统、环境应急管理系统、环境决策支持系统等业务应用系统。
- b) 应用支撑平台。包含数据交换平台、信息资源共享平台、GIS空间服务平台等基础应用支撑平台。
- c) 政务信息协同。包含电子政务、政务信息共享等政务信息系统。
- d) 公众信息发布。面向公众政务信息公开的信息系统。
- e) 其他。

5.2 服务范围

按服务形式和实现功能区分，环境信息服务包含了易于在分布式网络环境下使用的数据服务、计算服务、报表服务、GIS空间数据服务（空间Web地图服务、空间Web要素服务、空间Web覆盖服务等）等。

5.3 应用范围

按功能应用，注册服务可以分为：

- a) 面向应用交互。应用于环境信息系统之间、环境信息系统与外部信息系统之间的业务协作、流程协同。

- b) 面向信息资源共享。用以实现各类环境信息资源的交换和共享。
- c) 面向数据处理。用以加工处理各类环境信息资源。

5.4 服务来源

根据来源，服务提供者可以分为：

- a) 环境管理部门。为满足职能部门之间、行业内部的应用需求，来自环境信息系统的服务资源。
- b) 其他业务部门。为满足政务信息交换，来自其他部门业务信息系统的服务资源。
- c) 环境管理对象。为满足环境管理业务，来自企业信息系统的服务资源。

6 服务注册模式

6.1 模式组成

环境信息服务注册模式涉及四个角色和六项活动。四个角色是环境信息服务的服务提供者、服务注册中心、服务注册中心管理者和服务使用者。六项活动包括规划、注册、授权、管理、发现、访问。

6.2 角色

服务提供者负责本部门环境信息服务的规划、调查、收集、整理，根据有关标准、规范，开发部署服务以实现特定功能，在服务注册中心注册、授权，并负责及时更新。

服务注册中心是服务提供者和服务使用者的中介，为服务提供者提供发布服务资源的平台，为服务使用者提供发现服务资源的渠道。

服务注册中心管理者负责环境信息服务注册的总体规划，进行服务标识符的分配、管理和使用，负责服务注册中心的系统维护。

服务使用者为实现特定的应用系统功能，在服务注册中心检索服务，在授权范围内使用服务。

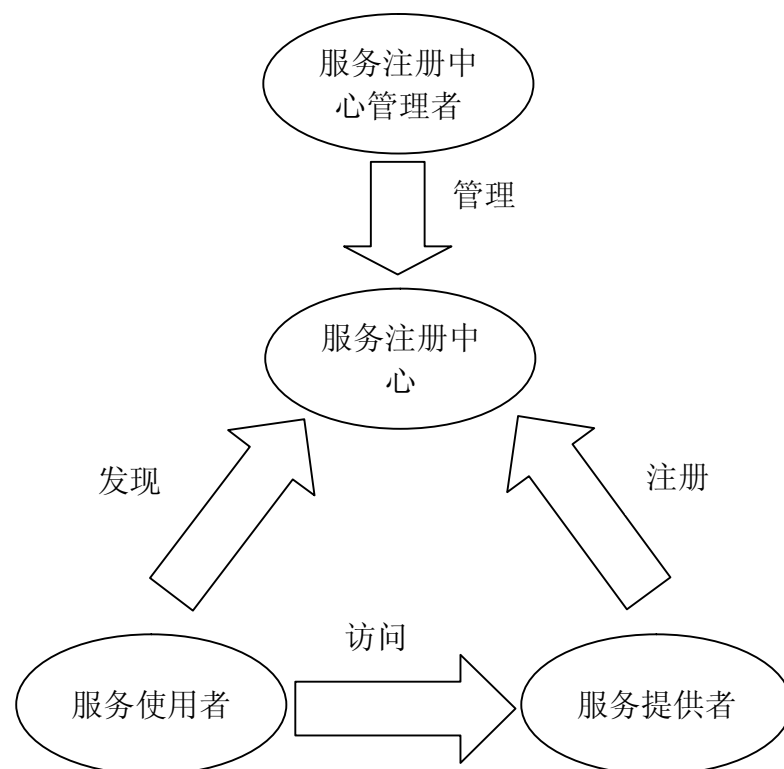


图 1 服务注册模式架构

6.3 活动

环境信息服务注册、管理和使用过程中可分为以下活动环节：

- a) 规划

服务提供者征集服务使用者的需求，制定本部门环境信息服务的总体规划，在运行过程中根据使用者的需求进行调整；服务注册中心管理者对服务注册中心的实现、部署、管理、运维等进行规划。

b) 注册

服务提供者将服务元数据提交至服务注册中心，服务注册中心管理者对服务元数据进行审核校验，接受符合规范的服务进入服务注册中心，未通过审查的返回提供者修改。注册服务信息的更新维护由服务提供者负责，应确保信息的及时性。

服务注册有手工或程序化两种方式。手工即由服务提供者手动在服务注册中心系统界面下操作，填写并提交有关服务元数据信息；程序化即由系统程序调用服务注册中心的标准接口进行注册。

c) 授权

服务注册中心在建立、管理和运行中，应建立相应权限管理机制，以此对提供者、管理者和使用者的操作权限进行范围界定，保证服务注册中心和注册服务信息的安全性。

权限管理应满足多层次多用户多种权限组合方式，灵活配置调整权限。

d) 管理

服务注册中心的更新维护工作由服务注册中心管理者承担，应建立相应的维护管理机制来确保服务注册中心的高可用性。

e) 发现

服务注册中心提供服务信息的分类导航、查询服务，服务使用者在分布的、异构环境中也能通过服务描述信息发现所需的服务资源。

f) 访问

服务使用者从服务描述信息中获得服务的定位及访问信息，如：服务的网络地址，通讯协议，消息格式等。通过手工或程序化方式构造服务访问请求，并连接服务地址，发送服务请求消息，以实现服务的绑定和互操作。

访问服务内容的数据格式应遵循《环境信息交换技术规定》和《环境空间数据交换技术规定》。

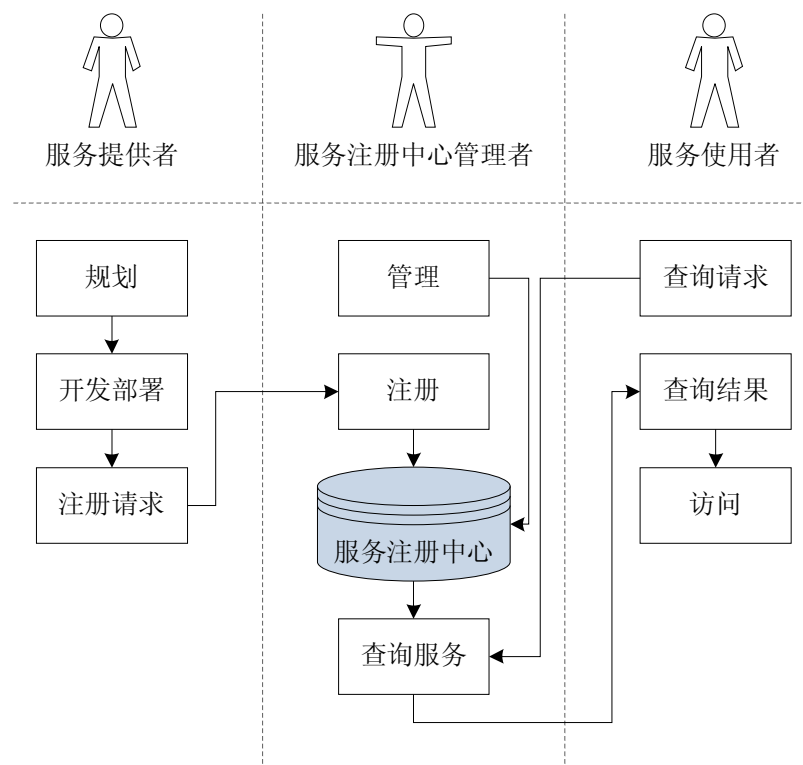


图 2 环境信息服务注册管理活动

7 服务注册实现模型

7.1 功能模型

服务注册功能主要包含服务注册和服务查询。

服务注册：提供服务描述信息的注册功能，帮助服务提供者发布服务实例。应用客户端通过服务实例的描述信息了解服务访问的细节，访问服务实例所提供的具体功能。

服务查询：向服务使用者（应用客户端）提供注册服务的发现功能，帮助应用客户端寻找所需的服务资源。

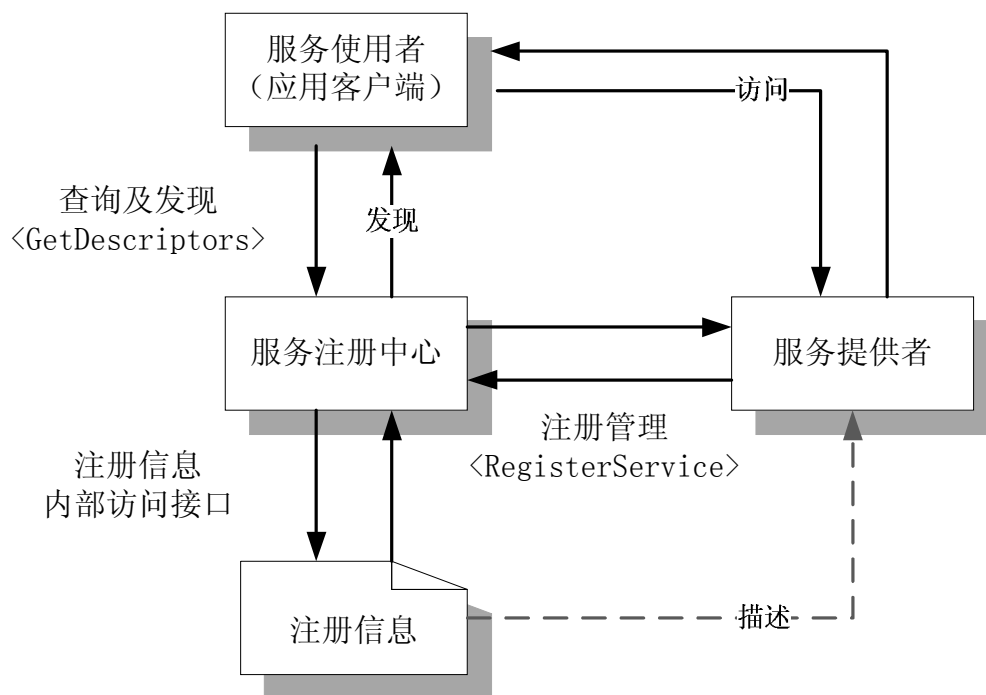


图 3 服务注册实现模型

7.2 接口模型

服务注册中心提供接口，实现服务注册、发现和管理功能。

a) 服务注册接口。位于服务提供者和服务注册中心之间，用于对服务描述信息进行注册、更新、组织和管理。

该接口提供注册、注销、注册信息更新、注册库建立和删除功能。接口参数是注册服务的注册管理请求消息，请求消息中指定服务注册中心的具体操作。服务注册中心处理管理请求消息，根据请求所指定的具体操作进行注册、注销等，并根据操作执行状态返回响应消息。

接口参数：注册管理请求

返回值：注册管理响应

b) 服务查询接口。位于服务使用者与服务注册中心之间，用于人工或程序化方式发现发布服务的注册信息。

该接口提供注册服务客户端对注册信息的检索功能。接口参数是注册服务的检索请求消息。服务注册中心处理检索请求消息，根据检索请求消息指定的操作细节对注册信息库进行检索，产生符合要求的注册信息，并以检索响应消息返回客户端。

接口参数：检索请求

返回值：检索响应

8 服务注册中心

8.1 服务注册中心技术要求

为保证能够在开放、异构的环境中运行，服务注册中心应满足以下技术要求：

a) 遵循Web服务标准协议。环境信息服务注册将遵循Web服务标准协议，应支持标准Web服务接口，可以从任何平台或语言进行访问，确保服务注册中心的开放性和互操作性。

b) 应支持服务注册中心之间的互联互通。可以通过建立关联关系以实现，对服务注册中心之间的注册服务信息进行交换同步。

8.2 服务注册中心性能要求

服务注册中心在运行、维护和管理过程中，应满足以下性能要求：

a) 高可用性。应防止各种系统错误、故障、停机事件发生，保证长期运行。

b) 稳定性。保证系统正常稳定运行。

c) 安全性。保证数据、系统安全。

8.3 服务注册中心的逻辑位置

服务注册中心在环境信息系统体系中处于共享层，位于数据层和交换层之上，为支撑层和应用层提供支持。其逻辑结构位置如下图。

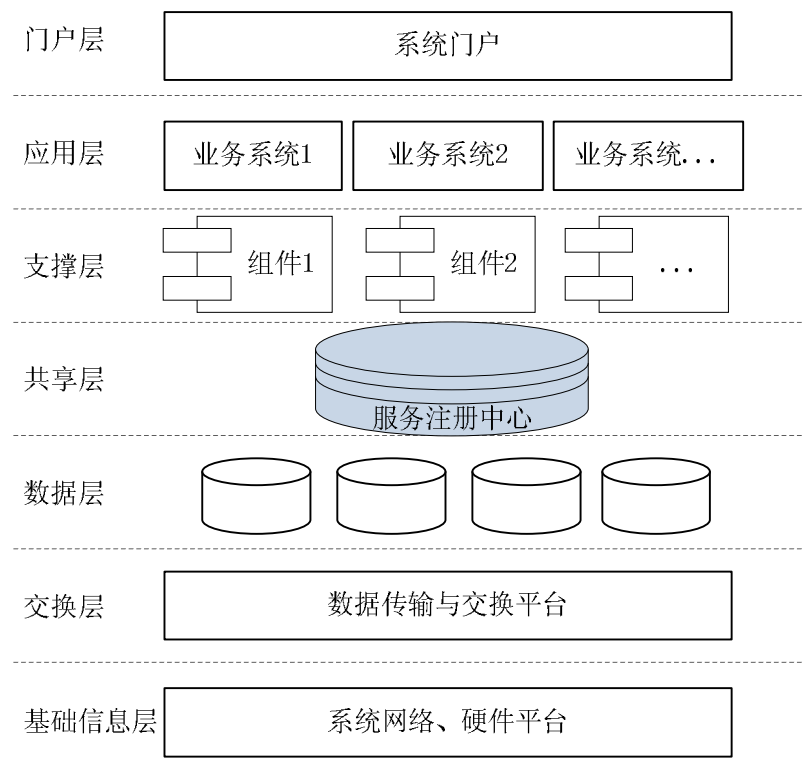


图 4 服务注册中心的逻辑位置

8.4 服务注册中心的部署方式

环境信息服务注册中心根据环境信息系统所在网络环境（环境保护业务专网、城域网、公网等）分别部署。

a) 集中部署方式

统一部署服务注册中心，各级环境业务应用将服务集中注册到一个服务注册中心。

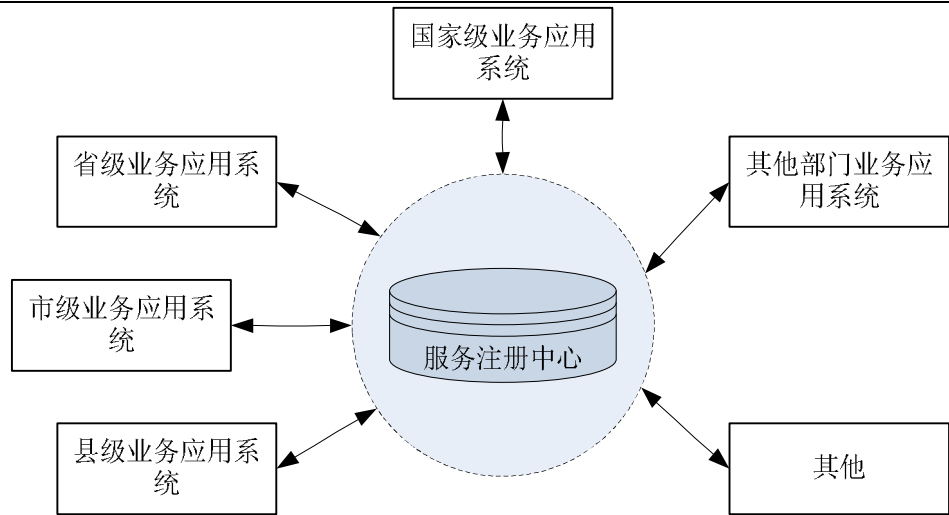


图 5 服务注册中心集中部署方式

b) 分级部署方式

服务注册中心采用分级部署方式，本级服务注册中心与上级服务注册中心应保持互联互通，实现注册信息交换，形成层次化服务注册中心管理体系模型。根服务注册中心位于体系的最顶层。

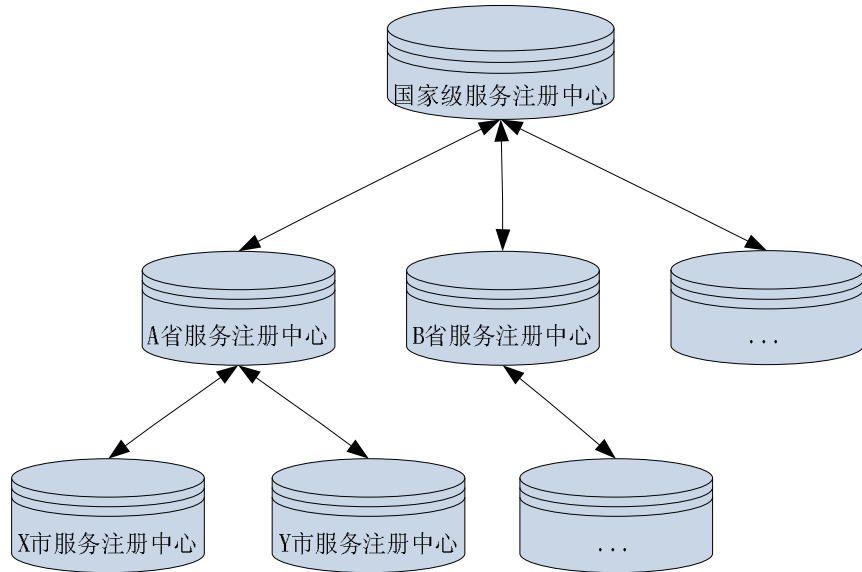


图 6 服务注册中心分级部署方式

c) 分散集中部署方式

面向主题业务应用的服务注册中心统一部署，形成各个分散的主题服务注册中心，各个主题服务注册中心之间由根服务注册中心集中相连。

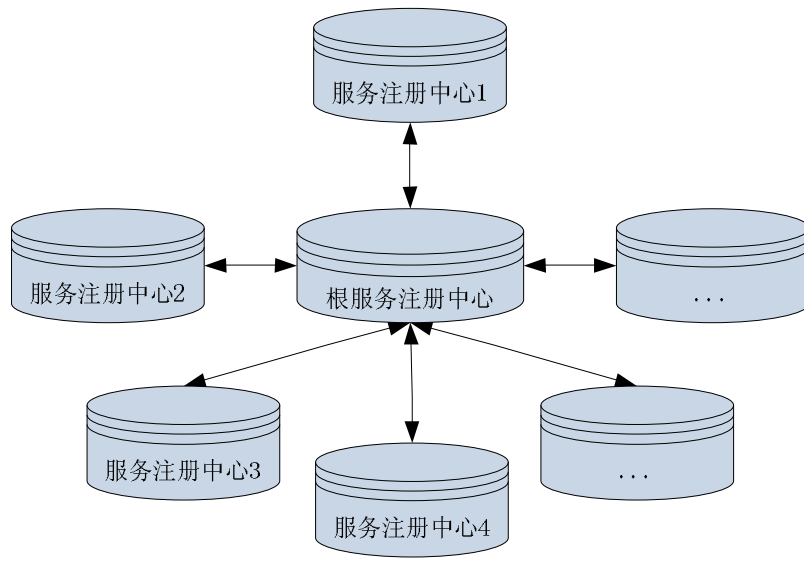


图 7 服务注册中心分散集中式部署方式

附录 A
(规范性附录)

基于 W3C UDDI/WSDL 的环境信息服务注册参考实现

A.1 UDDI概述

UDDI 始于2000年,由 Ariba, IBM, Microsoft 和其他33家公司创立。它是一个分布式的互联网服务注册机制,它集描述(Universal Description)、检索(Discovery)与集成(Integration)为一体,其核心是注册机制。UDDI实现了一组可公开访问的接口,通过这些接口,网络服务可以向服务信息库注册其服务信息、服务需求者可以找到分散在世界各地的网络服务。

A.2 UDDI服务注册体系结构

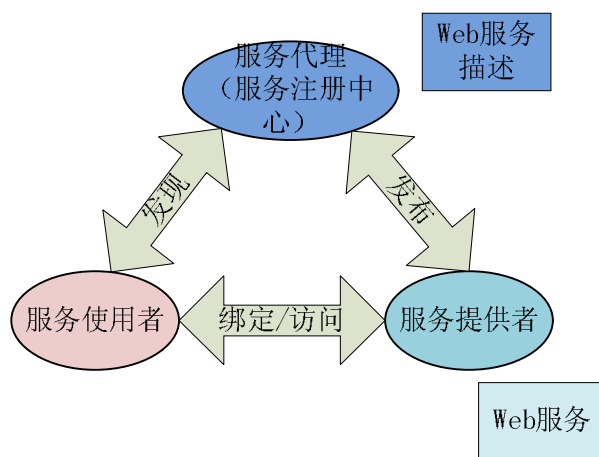


图 8 UDDI Web 服务注册体系结构图

发布、发现、访问是 Web 服务注册体系结构的三个基本组件所执行的三个基本操作:服务提供者通过向服务代理注册来配置和发布服务;服务使用者通过查找服务代理中被发布服务登记记录来找到服务;服务使用者绑定服务提供者并使用可用的服务。

A.3 UDDI分层Web服务协议栈

UDDI 包含于完整的 Web 服务协议栈之内,而且是协议栈基础的主要部件之一,支持创建、说明、发现和调用 Web 服务。

UDDI 构建于网络传输层和基于 SOAP 的 XML 消息传输层之上。发布服务使用了“通用描述、发现和集成(UDDI)API”,查找服务使用了 UDDI 和 Web 服务描述语言的组合(WSDL),绑定服务处理使用了 WSDL 和简单对象访问协议(SOAP)。此外,使用 Web 服务流程语言(Web Services Flow Language, WSFL)提供 Web 服务工作流描述、安全性、管理和服务质量功能,从而解决系统可靠性和可用性问题。

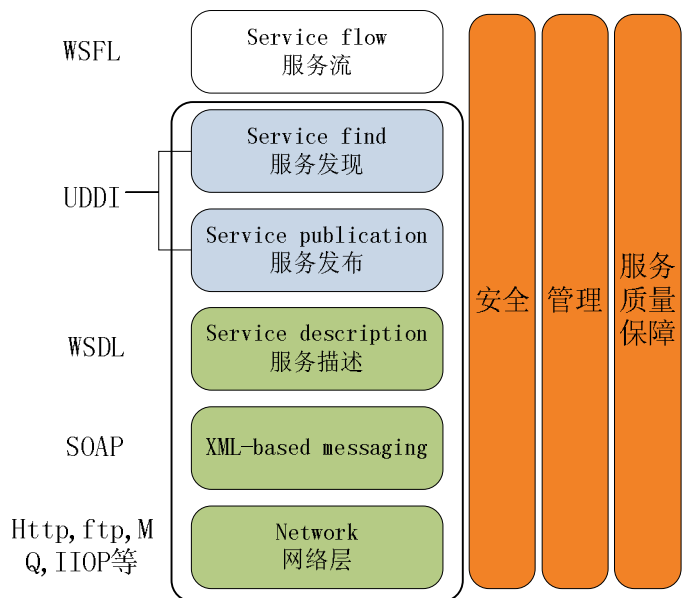


图 9 UDDI 的分层 Web 服务协议栈

A.4 UDDI 服务注册接口实现

基于UDDI/WSDL的环境服务注册参考实现实现了检索接口、注册管理接口，接口实现说明如下表所示：

接口	实现接口	功能
检索接口	UDDI Inquire	利用UDDI的Inquire API实现对注册信息库的检索，查询符合要求的服务资源描述—UDDI注册信息。
注册管理接口	UDDI Publication	利用UDDI的Publication和Replication API实现对注册信息的发布，包括：注册、注销、注册信息更新。

A.5 UDDI 服务注册数据结构

基于UDDI/WSDL的注册服务参考实现采用基于XML表示的注册信息，其数据结构中包括四个核心部分：业务实体 (businessEntity)、业务服务 (businessService)、绑定模板 (bindingTemplate) 和技术模型 (tModel)，它们的结构及相互之间的关系如下图所示：

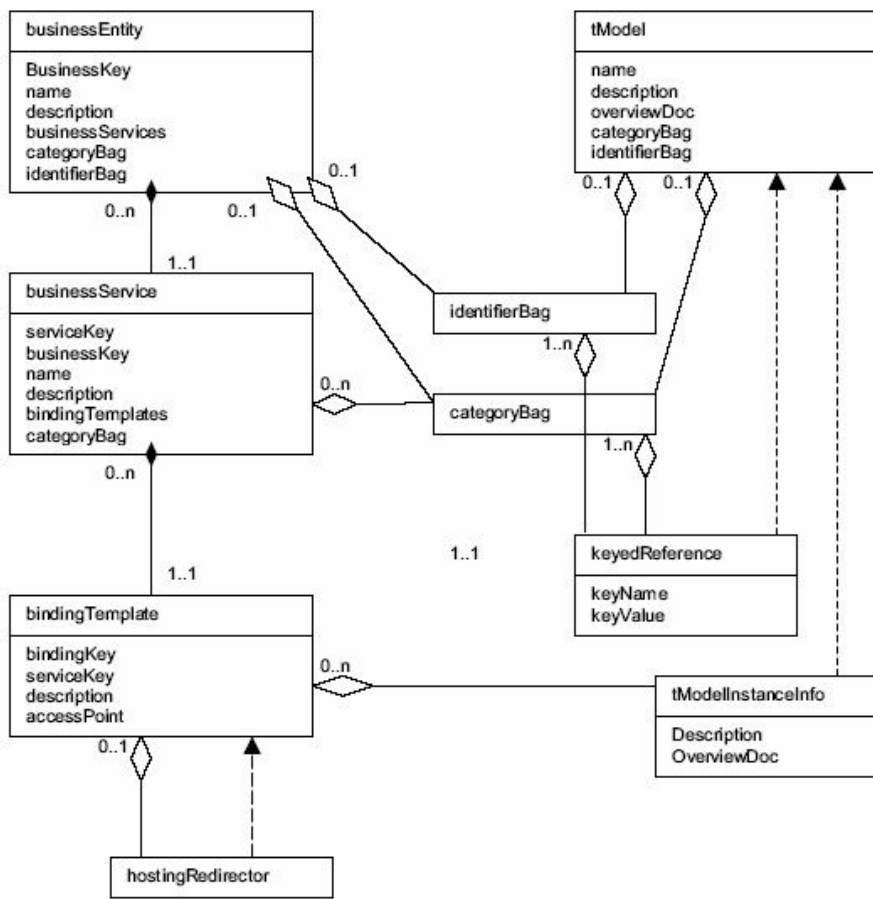


图 10 服务注册数据结构图

其中业务实体 (businessEntity) 用于对提供服务的单位、机构等的基本情况进行描述，同时包含有该单位或机构提供服务资源的描述信息—业务服务。

业务服务 (businessService) 用于对服务资源的内容、分类体系等基本信息进行描述，同时包含有对服务资源技术信息的描述内容—绑定模板。

绑定模板 (bindingTemplate) 用于描述服务资源的访问信息。服务资源访问信息以服务描述语言 (WSDL) 进行描述。技术模型相对独立于其它三个部分，用于描述服务资源相关的技术规范、协议、命名空间等信息。

附录 B
(规范性附录)
服务元数据 XML 模式 (XMLSchema)

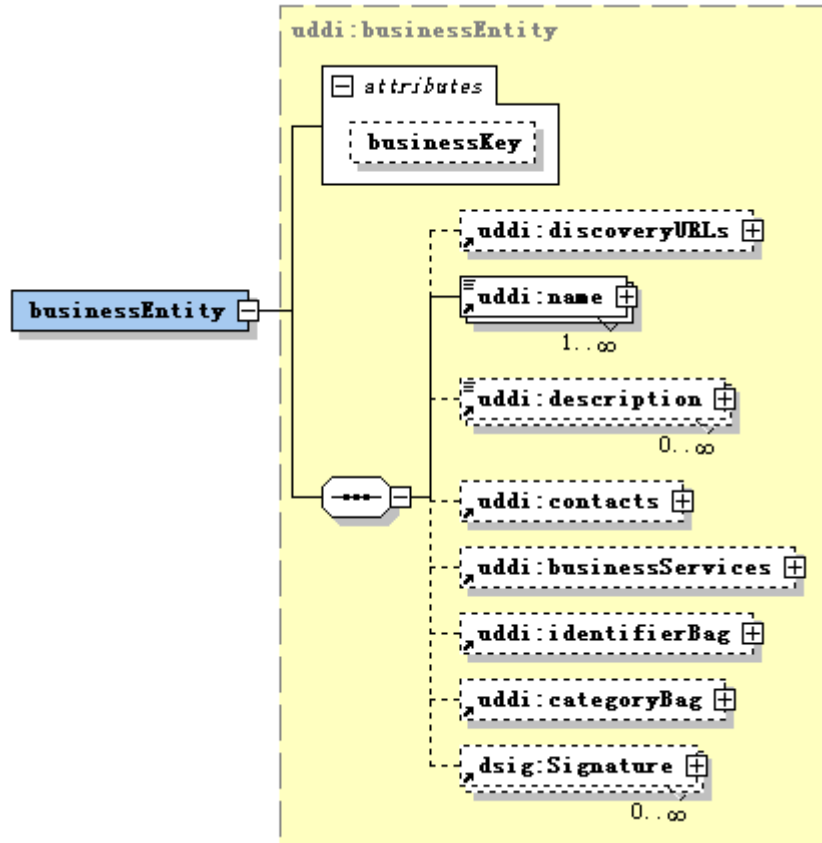


图 B.1 服务元数据 XMLSchema 结构图

附 录 C
(资料性附录)
服务元数据示例

C.1 服务注册消息XML实例

```
<?xml version="1.0" encoding="GB2312"?>
<businessEntity businessKey="1A3DB880-D5F4-11DA-B880-F94D3591C691">
  <name>商业实体名称</name>
  <contacts>
    <contact>
      <personName>联系人</personName>
      <email>person@example.com</email>
    </contact>
  </contacts>
  <businessServices>
    <businessService serviceKey="064B4170-D5F5-11DA-8170-A74C17FA61A7">
      <name>服务名称</name>
      <bindingTemplates>
        <bindingTemplate bindingKey="904BD800-D53A-11DA-B055-850A1DA99D79">
          <accessPoint>http://www.example.com/services/ClassifiedService
        </accessPoint>
        <tModelInstanceDetails>
          <tModelInstanceDetail>
            <tModelInstanceInfo tModelKey="66999A50-D5F4-11DA-9A50-FA44D6AD622A">
              <instanceDetails>
                <description>技术实现的描述信息。
              </description>
              <overviewDoc>
                <overviewURL>
                  http://www.example.com/ClassifiedService-impl.wsdl 实际定义服务的文档的 URL
                </overviewURL>
              </overviewDoc>
            </tModelInstanceInfo>
          </tModelInstanceDetail>
        </tModelInstanceDetails>
      </bindingTemplate>
    </bindingTemplates>
    <categoryBag>
      <keyedReference tModelKey="C1ACF26D-9672-4404-9D70-39B756E62AB4"
        keyName="uddi-org:types" keyValue="wsdlSpec"/>
      <keyedReference tModelKey="COB9FE13-179F-413D-8A5B-5004DB8E5BB2"
        keyName="ntis-gov:naics:1997" keyValue="511110"/>
    </categoryBag>
  </businessService>
</businessServices>
<identifierBag>
  <keyedReference tModelKey="4064C064-6D14-4F35-8953-9652106476A9" keyName="DUNS Number"
    keyValue="55555555"/>
  <keyedReference tModelKey="" keyName="US Tax ID" keyValue="11111111"/>
</identifierBag>
<categoryBag>
  <keyedReference tModelKey="COB9FE13-179F-413D-8A5B-5004DB8E5BB2" keyName="optional"
    keyValue="511110"/>
</categoryBag>
```

```
</businessEntity>
```

C.2 服务注册程序代码实现

基于.Net和Microsoft UDDI SDK编写，实现程序化的服务注册，程序代码如下：

```
using System;

using Microsoft.Uddi;
using Microsoft.Uddi.Businesses;
using Microsoft.Uddi.Services;
using Microsoft.Uddi.TModels;

namespace ConsoleApplication1
{
    public class UpdateMyBiz
    {
        public static void Main(string[] args)
        {
            try
            {
                // 创建到UDDI服务器的连接。Create a connection to the UDDI server that is to be accessed.
                UddiConnection myConn = new UddiConnection("http://uddi.example.com/inquire",
                "https://uddi.example.com/publish");

                // 创建注册服务的用户验证信息。Create authentication credentials for save operations.
                myConn.AuthenticationMode = AuthenticationMode.UddiAuthentication;
                myConn.Username = "输入用户名";
                myConn.Password = "输入密码";

                // 创建商业实体。Create a named business entity.
                BusinessEntity myBiz = new BusinessEntity();
                myBiz.BusinessKey = "输入business key";
                myBiz.Names.Add("输入business name");

                // 按多种语言添加描述信息。Add multi-language descriptions for the business.
                myBiz.Descriptions.Add("英文描述", "en");
                myBiz.Descriptions.Add("中文描述", "zh");

                // 添加分类信息和标志信息。Add classification and identification information for this
                business.
                myBiz.CategoryBag.Add("uuid:c0b9fe13-179f-413d-8a5b-5004db8e5bb2", "51121", "NAICS:
                Software Publisher");
                myBiz.CategoryBag.Add("uuid:4e49a8d6-d5a2-4fc2-93a0-0411d8d19e88", "US-WA",
                "Washington");
                myBiz.IdentifierBag.Add("uuid:8609c81e-eelf-4d5a-b202-3eb13ad01823", "00-000-0000",
                "D-U-N-S");

                // 添加联系人。Add contact details for relevant individuals.
                myBiz.Contacts.Add("John");
                myBiz.Contacts[0].UseType = "Web Developer";
                myBiz.Contacts[0].Emails.Add("John@example.com", "work");

                myBiz.Contacts.Add("Sally");
                myBiz.Contacts[1].UseType = "Technical Support";
                myBiz.Contacts[1].Phones.Add("1-800-555-1212", "mobile");

                // 添加单个服务的详细信息。Add details for a single service.
                BusinessService myBizServ = new BusinessService();
```

```

myBizServ.Names.Add("输入服务名称");
myBizServ.Descriptions.Add("输入服务描述");

// 添加访问服务的绑定信息。Add a binding to specify where and how the service can be
accessed.
BindingTemplate myBindingT = new BindingTemplate();
myBindingT.AccessPoint.Text = "http://uddi.example.com";
myBindingT.AccessPoint.UrlType = UrlType.Http;

// 如果该服务符合已发布的tModel, 则这项服务加注这项信息。If the service conforms to a
published tModel, annotate this service appropriately.
TModelInstanceInfo myTInfo = new
TModelInstanceInfo("uuid:ac104dcc-d623-452f-88a7-f8acd94d9b2b");
myBindingT.TModelInstanceInfos.Add(myTInfo);
myBizServ.BindingTemplates.Add(myBindingT);

// 将服务附加到商业实体。Attach the service to the business.
myBiz.BusinessServices.Add(myBizServ);

// 保存商业实体到UDDI服务器。Create an object to save a business.
SaveBusiness sb = new SaveBusiness();
sb.BusinessEntities.Add(myBiz);

// Display XML sent to UDDI server.
Console.WriteLine(sb.ToString());

// Send the save business request.
BusinessDetail savedBiz = sb.Send(myConn);

Console.WriteLine("Business: " + savedBiz.BusinessEntities[0].Names[0].Text);
Console.WriteLine("Allocated key: " + savedBiz.BusinessEntities[0].BusinessKey);
}
catch (Microsoft.Uddi.UddiException e)
{
    Console.WriteLine("UDDI error: " + e.Message);
}
catch (Exception gen)
{
    Console.WriteLine("General exception: {0}", gen.Message);
}
}
}
}

```

参考文献:

- | | |
|--------------------|-------------------------|
| SDS/T 2212—2004 | 目录服务规范 |
| SDS/T 2213—2004 | 数据与服务注册规范 |
| SDS/T 2221.2—2004 | 数据访问服务接口规范——WEB 要素服务规范 |
| SDS/T 2241—2004 | 数据分发服务指南与规范 |
| HJ/T 417—2007 | 环境信息分类与代码 |
| GB/T 21063.1—2007 | 政务信息资源目录体系 第1部分: 总体框架 |
| GB/T 21063.2—2007 | 政务信息资源目录体系 第2部分: 技术要求 |
| GB/T 21063.6—2007 | 政务信息资源目录体系 第6部分: 技术管理要求 |
| UDDI Version 3.0.2 | UDDI 3.0.2 版本 |