

# The Development and Practice of the Geographic Semantic Web Services Based on OWL-S

Shuangyun PENG<sup>1</sup>, Weifeng MA<sup>2</sup>, Kun YANG<sup>1</sup>, Quanli XU<sup>1</sup>, Bangmei HUANG<sup>3</sup>

1. Department of GIS, Yunnan Normal University, Kunming, China

2. School of Geography, Beijing Normal University, Beijing, China

3. Kunming NO.10 Middle School, Kunming, China

Email: frankmei@126.com

**Abstract:** With the development of Semantic Web technology, the Geographic service based on OWL-S is an effective way for sharing and interoperation of heterogeneous information resources in the distributed network environment. This paper focuses on how to dynamic find, bind and invoke Geographic web services by Ontology Web Language for Services (OWL-S). The development of an OWL-S based web service is time consuming and error prone. The main reason is the lack of a integrated development environment. So, we build an integrated developing environment of Geographic semantic web services by using existing tools. This IDE can assist developers to develop Geographic semantic web services. In order to realize the applications of Geographic semantic web services in OWL-S IDE, an experimental system has been developed. In the system we generate Geographic Ontology and develop, publish, bind, find and invoke web services in distributed network.

**Keywords:** semantic web services; Geographic Ontology; OWL-S IDE; Experimental system

## 基于 OWL-S 的地理语义 Web 服务的开发与实践

彭双云<sup>1</sup>, 马卫峰<sup>2</sup>, 杨昆<sup>1</sup>, 许泉立<sup>1</sup>, 黄帮梅<sup>3</sup>

1. 云南师范大学, 昆明, 中国, 650092

2. 北京师范大学, 北京, 中国, 100875

3. 昆明市第十中学, 昆明, 中国, 650011

Email: frankmei@126.com

**摘要:** 随着语义 Web 技术的发展, 基于 OWL-S 的空间信息服务已经成为异构分布式空间信息共享与互操作的有效途径。本文主要讨论如何基于 OWL-S 进行空间信息服务的智能动态发现、绑定及调用。基于 OWL-S 的 Web 服务开发是一个耗时且易出错的工作, 一个主要的原因是缺少统一的开发环境。因此, 本文在现有成熟工具的基础上构建了一个集成开发环境, 通过该集成开发环境可以帮助程序员在一致统一的环境中开发地理网络智能服务。为了检验该集成开发环境, 本文进行了实践开发, 并得出了正确的结果。

**关键词:** 地理语义 Web 服务; 地理本体; OWL-S; 实验系统

### 1 引言

随着 Internet 的飞速发展和 Web 的广泛应用, 特别是“数字地球”研究的兴起和“国家空间数据基础设施”方案的提出, 促进了地理信息系统 (Geographic Information System, 简称 GIS) 向着网络化、标准化、全球化和大众化的方向发展, 从而也产生了面向服务的分

布式地理信息系统新模式。如今, 在 ISO/TC211, OGC 等国际地理信息标准化组织的大力推动下, 地理信息服务的各项标准逐渐成熟和完备, 互联网上地理信息服务的种类和数量也日益增加。如何在诸多的服务中找到符合用户需求的服务, 即服务发现, 已成为利用 Web 服务实现地理信息资源共享必须解决的问题之一。传统的 Web 服务发现是采用基于关键字的匹配方式实现的, 由于缺乏足够的语义信息, 这种匹配方式的查全率和查准

通讯作者: 杨昆, 云南师范大学教授, 武汉大学博士生导师

率都不高<sup>[1]</sup>。地理信息具有时域、空间、专题、多尺度等多种语义特征,将这些地理语义特征信息添加到地理信息服务中,可实现服务的智能发现、绑定、调用和组装。本文通过对系统开发范式的研究,提出了基于 OWL-S 的模型驱动方法。在该方法的指导下建立基于 OWL-S 的地理 Web 服务实现系统,实现了地理对整个系统的功能进行模拟和实验。该实验是对 OWL-S 自身探索过程中的一次实证研究,使用本体作为一种手段实现 Web 服务的智能动态发现、组装和调用。

## 2 代码驱动和 OWL-S 模型驱动的服务模式

### 2.1 代码驱动方法

当前,Web 服务的实现通常是给现有的应用程序功能添加一个接口,如图 1 所示。典型的是,一个功能系统成为一个服务用于分布式应用。应用程序开发者为了使用服务的功能,用 WSDL 文档描述服务<sup>[2]</sup>。因此,使用一种 inside-out 的方法开发服务:功能被开发用于特殊应用,此后该功能通过 Web Services 成为一个有效的服务。

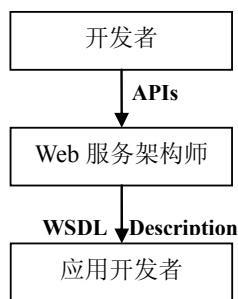


Figure 1. The code-driven service model  
图 2. 代码驱动的方法

如果某机构希望使用 OWL-S 表达它的 Web 服务,它将会使一个 process model 符合 Web 服务接口,最终创建一个本体来表达服务提供的信息。这种方法称为代码驱动的方法<sup>[3]</sup>。然而,代码驱动的方法很繁琐而且充满错误。由于 Web 服务接口最初创建的时候没有考虑到本体,也没有考虑服务使用的方法,因此映射服务 I/O 到服务本体的对应部分是一项相当困难的任务。这种映射很脆弱,而且依赖于对期望数据的假设。除此以外,经常还有参数,如用户名和密码,这些参数与本体的其它部分不符。这就留给应用程序和 Web 服务开发者一些问题:通过修改来纳入新信息而牺牲本体质量,或者将这些值直接写入 Service Grounding 元素中而放弃灵活性。

### 2.2 模型驱动方法

服务的语义表达则需要新的思维和工作方式,那就是采用模型驱动的方法。如图 2 所示,开发者首先必须和领域专家一起工作,共同创建服务所在知识领域的本体。然后,OWL-S 设计师可以创建一个 Process model 用于描述客户如何和服务交互。于是,OWL-S Provider 就可以定义和注册 Web 服务的接口。OWL-S Provider 负责维护一个 OWL-S 匹配引擎,服务提供者可以发布他们服务的 OWL-S 描述到 OWL-S Matchmaker 匹配引擎。OWL-S Provider 将会为那些 Web 服务的消费者提供对 OWL-S 匹配引擎的使用。应用程序开发者将会使用相同的本体建立请求来匹配 Web 服务。

通过这种工作方法,可以这样设计 Web 服务接口,让它直接映射到它的 OWL-S 描述的本体上。创建一个完整的、可扩展的本体来正确的表达服务所在的知识空间,而不是一个简单的服务接口。模型驱动的方法不仅允许更好的匹配服务查询,而且使以后添加新的服务变得简单。

在代码驱动的方法中,通过使用 Java 或其它编程语言实现 Web 服务,通过使用 WSDL2OWL-S 转换器从代码中获取 OWL-S 描述。产生于这种方法中的 OWL-S 描述正在丢失语义信息和控制流信息。模型驱动的方法不是从 Web 服务代码开始、以 OWL-S 描述结束,而是沿着相反的方向,它从功能(开发者期望从 Web 服务中得到的功能)的 OWL-S 规范和 Web 服务交互过程的规范开始,以 Web 服务 Java 代码的产生结束<sup>[4]</sup>。

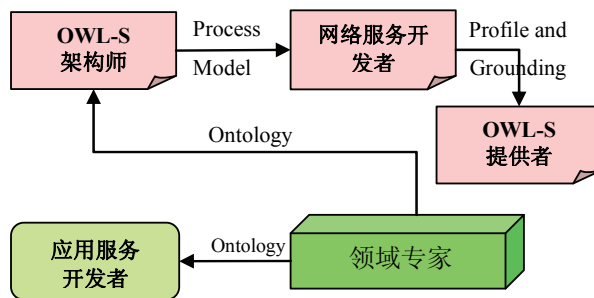


Figure 2. OWL-S driven service model  
图 5.3 模型驱动的方法

## 3 语义 Web Services 开发环境的搭建

实现模型方法的第一步是选择开发环境。模型问题开发过程中选择有效的工具是本质的问题。

### 3.1 集成开发环境的设计

增强语义的 Web 服务的开发需要很多不同的信息类型和行为如 Web 服务的真正实现;还需要 WSDL 描述的编辑; OWL-S Profile、Process Model 和 Grounding 的编辑; 所有输入与输出的语义规范, 以及他们与 XML 框架之间的映射。更重要的是, 所有行为的结果是紧密相关的: Profile 必须表达 Web 服务的功能, Process Model 必须保证 Web 服务的实现, Grounding 必须提供 OWL-S 与 WSDL 之间的一致映射, 最后 Web 服务的实现必须是开源的。因而, OWL-S Web 服务的开发既耗时又容易出错, 而且支持开发者

的少量工具没有形成一个连续的工具集, 因此, 开发者很难在一致的基础上使用<sup>[5]</sup>。

OWL-S 集成开发环境 (IDE) 设计的指导原则是在一个一致的、可扩展的环境中, 整合开发者需要的工具, 即语义 Web 服务实现、编辑和部署过程中涉及到的工具。开发工具的一致性允许开发者无缝地工作于语义 Web 服务的不同方面, 而环境的可扩展性允许其它组织在它的工作中使用语义 Web 服务框架并提供额外的贡献。在这个指导原则下, 我们在现有工具的基础上搭建了 OWL-S IDE, 如图 3 所示。

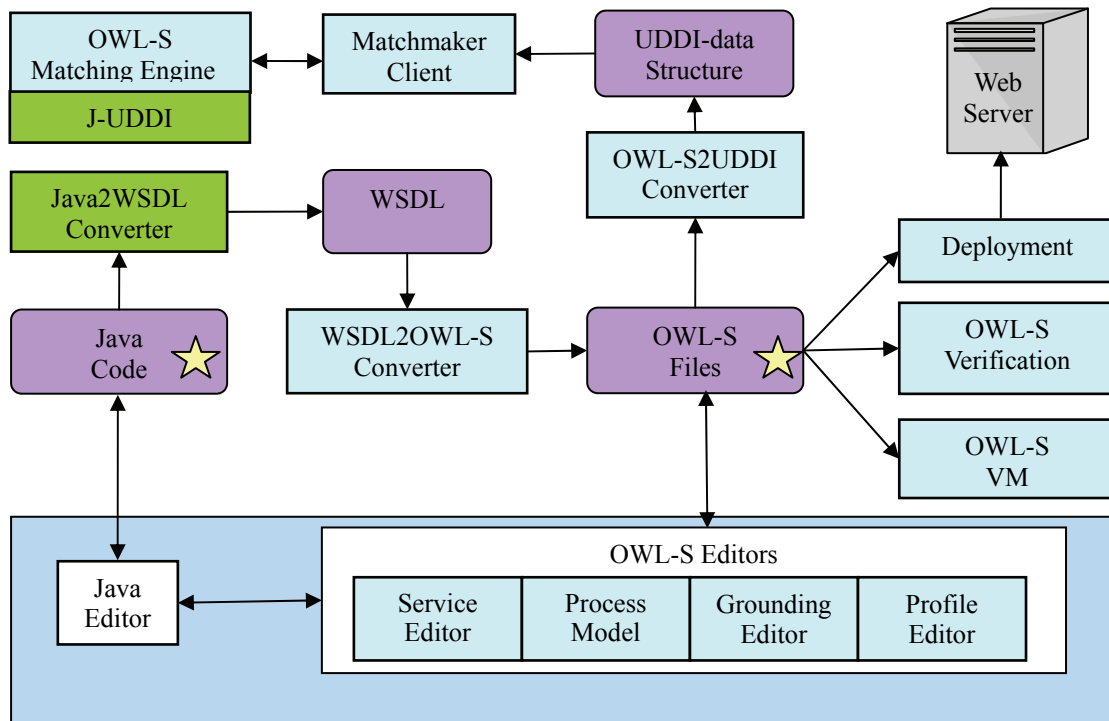


Figure 3. IDE of OWL-S  
图 4. OWL-S 集成开发环境

### 3.2 集成开发环境概述

我们通过提供一个统一集成开发环境 (IDE) 来解决开发者的问题, OWL-S IDE 将 Java IDE 和在 Eclipse 中实现的其它 Web 服务框架无缝地集成在一起。OWL-S IDE 对开发者的支持贯穿于语义 Web 服务开发的整个生命周期, 如图 3 所示, 图中在使用 OWL-S IDE 进行语义 Web 服务开发过程中的活动。虽然图中没有明确的起点, 但有两个明显的起点 (由星号表示出来), 一个是 Java 代码, 一个是 OWL-S 规范。这两个起点定义了两种方法, 一个是代码驱动的方法, 一个是模型驱动的方法, 开发者用这两个方法来对 Web 服务进行语

义描述。在代码驱动的方法中, 通过使用 Java 或其它编程语言实现 Web 服务, 通过使用 WSDL2OWL-S 转换器从代码中获取 OWL-S 描述。产生于这种方法中的 OWL-S 描述正在丢失语义信息和控制流信息。因此这个描述不得不使用 OWL-S Editors 添加以上的信息来完成编辑。

由开发者产生的 OWL-S 描述易用出现语法和逻辑错误。OWL-S 编辑器能够找到语法错误并帮助开发者进行修正。OWL-S 描述中的逻辑错误可以被发觉, 通过使用 IDE 中基于模型验证的检验器。该检验器能够检测 OWL-S Process Model 控制流和数据流的正确性。被

检验过的 OWL-S 描述必须被部署到一个 Web 服务上。该开发框架与其它基于 Eclipse 的 Web 服务框架的结合，提供了必要的工具用于打包 Web 服务代码和 OWL-S 描述以及把它们部署到远程 Web 服务器上。部署这些语义 Web 服务后，必须把它们发布到 UDDI 注册中心来参与到发现过程中。接下来的部分具体描述了这个过程。IDE 也提供了 OWL-S 虚拟机（一个 OWL-S 执行环境）用于自动产生客户端代码并执行该代码。这个过程帮助开发者在开发和编译时间发现问题，降低执行时间出错的可能性。

### 3.3 领域本体的构建

由于本论文最初是面向艾滋病领域提供地理空间专题数据功能，因此构建艾滋病地理本体成为实现系统的基础和关键。但由于没有艾滋病领域的地理空间数据，加之我们在实验过程中用到了微软的 TerraService，所以我采用 TerraServer 的地理空间数据作为实验数据。

本文选择的本体编辑和创建工具是斯坦福大学开发研制的 Protégé3.2.1<sup>[6]</sup>，并最终生成为 W3C 推荐的本体形式化表达语言 OWL。在 protégé 下建立一个 OWL/RDF 文件，创建类和子类，并定义对象以及数据类型之间的关系。对于形成的地理本体树状类结构图，用户可以根据经验和系统的需求在 protégé 的 OWL VIZ 标签下进行更改，直到该地理本体满足特定的用户需求为止。我们基于 TerraServer 构建了一个地图服务本体，如图 4 所示。



Figure 4. The geographic ontology of San Francisco  
图 4. San Francisco 的地理本体

## 4 实验过程

在构建了领域地理本体后，我们需要针对具体的 Web 服务进行实验，可以选择已有的领域地理 Web 服务或重新编写，然后按照如下步骤展开实验。

### 4.1 确定领域内的地理 Web 服务

第一步就是确定选择的 Web 服务，这些服务能够使用 OWL-S 被描述和访问，它们可以是已有的服务也可以是重新编写的服务，但它们都必须遵循以下原则<sup>[7]</sup>：

- ☑ 服务应该通过不同的接口而提供相似的功能。对不同接口而本体含义相似的操作将为含义更深的查询和结果做好准备。
- ☑ 服务应该提供大量能够用于测试 OWL-S 基本功能的简单操作。这些简单的服务将会组成更为复杂的过程。
- ☑ 服务的 WSDL 定义是可用的。
- ☑ 访问服务不要成本开销。

根据以上原则，我们最好选择艾滋病领域已有的 Web 服务来进行实验，以节省时间和开销、缩短实验周期。由于没有可用的艾滋病领域的 Web 服务，而且考虑到重新开发周期较长等原因，我们选用微软 TerraServer 提供的地理数据服务来进行实验。

我们使用 WSDL2OWL-S 组件，生成 Web 服务的 OWL-S Profile。生成 OWL-S Profile 涉及一些执行操作的简单过程，如位置格式之间的转换，以及构成多服务调用的复杂过程。使用 OWL-S Editor 修改 OWL-S Profile，我们映射操作的 I/O 参数到它们在 OWL 本体中的对应项。因为没有直接的映射，使用 XSLT 样式表修改 Service Grounding 元素，使 I/O 在它们的 OWL 表达和服务对应项之间转换。

### 4.2 Web 服务的发布与广告

为了用 OWL-S Matchmaker 来广告 Web 服务，OWL-S Profile 在因特网上必须是可访问的。结果，为了能够让 OWL-S Matchmaker 访问，我们把它发布到一个 Web 服务器上。

使用 IDE 中 OWL-S2UDDI 组件，将 OWL-S Profile 转换成一个 UDDI advertisement 并发布到 CMU 维护的 OWL-S Matchmaker<sup>[8]</sup>。于是 OWL-S Matchmaker 将 OWL-S Profile 注册到它的数据库中，并使这个 OWL-S Profile 可以被发现。

### 4.3 Web 服务的组装

首先要编写 TerraService 的 WSDL 用于描述服务的接口，TerraService 的 WSDL 描述位于 <http://terra-server-usa.com/TerraService2.asmx?WSDL> 中。

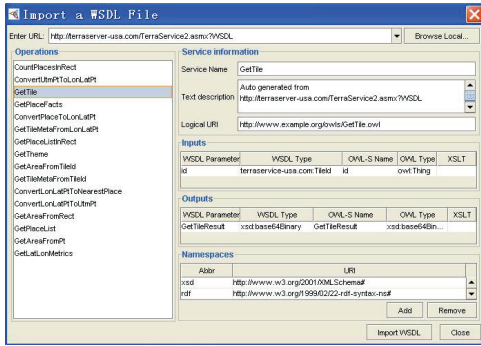


Figure 5. Convert WSDL to OWL-S  
图 5. WSDL 转换为 OWL-S

通过 OWL-S Editor 载入 Web 服务的 WSDL，可以在左边的 Operations 选项中看到服务所包含的所有方法<sup>[9]</sup>（如图 5 所示），我们这里选择 GetPlaceList 和 GetPlaceFacts 作为实验的原子服务，然后保存生成 OWL-S 文件。

新建一个 Service 实例：包括一个 Profile 实例、一个 CompositeProcess 实例和一个 WSDLGrounding 实例，分别命名为 myservice、myprofile、myprocess 和 mygrounding。然后把它们连接起来，首先选中 myservice，把它的 describedBy 属性置为 myprocess，presents 属性置为 myprofile，supports 属性置为 mygrounding。最后通过 Visual Editor 选项生成对 Process 进行编辑，生成该服务过程的结构图，如图 6 所示。从图中可以看出通过输入参数 placeName 和 MaxItem 将由第一个 Process 产生的值赋予 Place 结构体，然后参与到第二个 Process 中，如果地图存在则返回 True，不存在则返回 False。

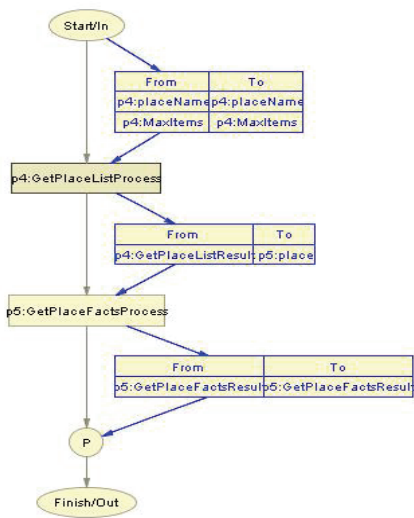


Figure 6. Service process  
图 6. 服务过程结构图

我们通过给 placeName 赋值为 San Francisco，结果返回 True。

#### 4.4 Web 服务的查询及调用

使用 OWL-S Editor，创建一个 OWL-S 请求，包括想要得到的服务的 I/O 的本体描述。如同 OWL-S Profile，OWL-S 请求必须被发布到一个 Web 服务器上（在这个 Web 服务器上，OWL-S 请求可以访问 OWL-S Matchmaker）。使用 IDE 中的客户程序，查询 OWL-S Matchmaker 中与 OWL-S 请求最匹配的服务，OWL-S Matchmaker 成功返回一组服务。OWL-S Matchmaker 既能返回简单服务，也能返回复合服务。我们这里以查询 ConvertLatLonToUTM 服务为例，如图 7 所示。

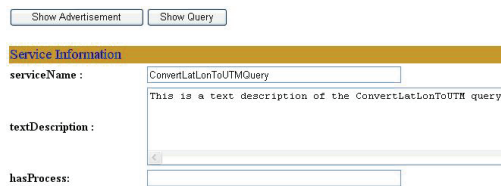


Figure 7. The query interface  
图 7. 服务的查询界面

如果本体已被正确定义以及 OWL-S Matchmaker 对 Profile 和创建的请求起作用，那么下一步就是从一个客户程序发出请求。OWL-S Matchmaker 和 OWL-S VM 都有 API 函数，Java 应用程序可以使用这些 API 函数来发现和调用服务。在运行时间，OWL-S VM 使用 OWL-S Service Grounding 元素和服务 WSDL 描述将 OWL 请求转换为 SOAP-XML 消息，该消息能够被 Web 服务所理解。最后当输入“San Francisco 地图”就可以得到如下图所示的结果。

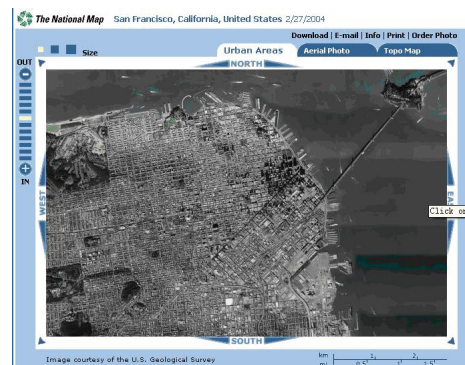


Figure 8. The result of query  
图 8. 查询结果

## 5 结论

本文首先从开发范式上入手,突破现有的代码驱动的方法,基于 OWL-S 提出了模型驱动的方法实施系统的开发。在搭建语义 Web Services 开发环境上,提出了基于现有开源工具来搭建一个集成开发环境进行系统的验。最后按照实验过程对 Web 服务进行一步步的测试,完成了 Web 服务的发布和组装,成功完成了一个简单实验过程。

## 致 谢

本论文得到了国家 863 计划 NO.2007A12Z231 和云南省自然科学基金项目 No.2009FXW001 的资助

## References (参考文献)

- [1] Giang lin, Gong Jian-ya, GeoSpatial Web Service Description and Discovery Based on OWL-S[J], GEOMATICS & SPATIAL INFORMATION TECHNOLOGY, 2007, 30(5),P20-27  
蒋玲,龚健雅,基于 OWL-S 的地理信息服务描述和发现[J],测绘与空间地理信息,2007,30(5),P20-27
- [2] Martin D, Cheyer A, Moran D, et al. OWL-S: Semantic Markupfor Web Services, <http://www.daml.org/-services/owl-s/>.
- [3] Srinivasan, Naveen;Paolucci,Massimo;& Sycara,Katia. CODE:A Development Environment for OWL-S Web Services (CMU-RI-TR-05-48). Pittsburgh,PA:Robotics Institute, Carnegie Mellon University,2005.[http://www.ri.cmu.edu/pub\\_files/pub4/srinivasan\\_naveen\\_2005\\_1/srinivasan\\_naveen\\_2005\\_1.pdf](http://www.ri.cmu.edu/pub_files/pub4/srinivasan_naveen_2005_1/srinivasan_naveen_2005_1.pdf).
- [4] Chris Metcalf, Grace A.Lewis. Model Problems in Technologies for Interoperability:OWL Web Ontology Language for Services (Technical Note CMU/SEI-2006-TN-018) April 2006. <http://www.sei.-cmu.edu/reports/06tn018.pdf> Martin,David,et al. Bringing Semantics to Web Services:The OWL-S Approach.[http://www-2.cs.cmu.edu/~softagents/papers/OWL-S-SWSWPC\\_2004-final.pdf](http://www-2.cs.cmu.edu/~softagents/papers/OWL-S-SWSWPC_2004-final.pdf)(2004).
- [5] A Practical Guide To Building OWL Ontologies Using The Prot'eg'e-OWL Plugin and CO-ODE Tools [EB/OL]. <http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf>
- [6] World Wide Web Consortium.Semantic Web Services Framework (SWSF).[http://www.w3.org/Submission/2005/07/\(2005\)](http://www.w3.org/Submission/2005/07/(2005)).
- [7] World Wide Web Consortium.OWL Web Ontology Language for Services (OWL-S).[http://www.w3.org/Sub-mission/2004/07/\(2004\)](http://www.w3.org/Sub-mission/2004/07/(2004)).
- [8] TerraService.NET An Introduction to Web Services [EB/OL]. [http://research.microsoft.com/research/pubs/view.aspx?msr\\_tr\\_id=MSR-TR-2002-53](http://research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-2002-53).