

文章编号: 1672-2892(2011)02-0248-06

基于 OWL-S 的地理空间信息服务描述与发现

魏 斌¹, 刘 迎²

(1.绵阳师范学院, 四川 绵阳 621900; 2.三门峡职业技术学院, 河南 三门峡 472000)

摘 要: 在分析空间信息服务特点的基础上, 对 Web 服务本体语言(OWL-S)进行扩展, 使其可以准确地描述空间信息服务。提出了基于语义的空间信息 Web 服务“三阶段”匹配方法, 实现空间信息服务发现, 分别针对基本信息、功能描述、服务质量对服务进行匹配, 并根据用户的偏好对服务进行优化排序。实验表明, 基于此匹配方法的空间信息服务查准率和查全率明显高于常规的基于关键字的服务匹配方法。

关键词: 地理空间信息服务; Web 服务本体语言; 服务匹配; 服务描述

中图分类号: TN91.02

文献标识码: A

Geospatial information web service description and discovery based on OWL-S

WEI Bin¹, LIU Ying²

(1.Mianyang Normal University, Mianyang Sichuan 621900, China, 2.Sanmenxia Polytechnic, Sanmenxia Henan 472000, China)

Abstract: Basing on the analysis of geospatial information services, this paper expands Ontology Web Language for Services(OWL-S) so that it can accurately describe geospatial information services. A "three stages" matching method of geospatial information web services based on semantic matching is proposed using these expansions. This method matches geospatial information Web services in basic information, function description and quality of service, it also sorts and optimizes services by user preferences. Experiment results show that the precision and recall of discovery for geospatial information Web services based on semantic matching using the proposed method are higher than that adopting conventional service matching methods based on keywords.

Key words: geospatial information Web service; Ontology Web Language for Services; service matching; service description

Web 服务是一种构建应用程序的普遍模型,可以在任何支持网络通信的操作系统中实施运行。它是一种新的应用程序分支,是自包含、自描述、模块化的应用,可以发布、定位、通过 Web 调用^[1]。随着 Web 服务技术的兴起,人们对地理空间信息服务的搜索和集成提出了进一步的要求,基于语义的服务描述和发现能够有效地提高服务查找的查全率和查准率,因此成为了当前 Web 服务领域的研究热点^[2]。但现有的语义 Web 服务发现方法用在地理空间信息服务中存在以下问题: a) 采用的服务描述语言中没有充分考虑到空间信息服务的特点,不能对空间服务进行准确描述; b) 由于空间信息处理对服务质量要求较高,而当前服务描述语言缺乏对服务质量的描述,不能根据服务的质量和用户偏好进行排序,返回最优化的结果。本文正是针对这些问题,对 OWL-S 语言进行适当扩展,提出了“三阶段”匹配算法,以提高空间信息服务的查全率和查准率。

1 空间信息 Web 服务

空间信息 Web 服务是在 Web 服务技术和标准基础之上实现的地理空间信息网在线服务。它利用 Web 服务技术提供的公共接口、交换协议和服务规范,提供应用客户管理、注册服务、编码、处理服务、描绘服务和数据服务等。主要的应用客户包括发现客户、地图浏览客户、影像利用客户、增值客户和传感器客户等^[3]。空间信息 Web 服务除了采用基本的 Web 服务技术协议外,还需要有关地理空间信息及处理的技术协议,目前主要有开发地理信息联盟(Open Geospatial Consortium, OGC)、ISO/TC211 和万维网联盟(World Wide Web Consortium, W3C)

制定。其中, OGC Web 服务(OWS)是典型的空间信息 Web 服务。

OWS 提供了 1 个以服务为中心的互操作框架, 支持多种在线地理数据源、传感器产生的信息和地理信息处理功能的基于 Web 的发现、访问、集成、分析、利用和可视化。其中的服务大多由更基本的服务来组合构建, OGC 把它们分为 3 种类型: 数据服务、处理服务和表现服务。数据服务提供对数据存储和数据库中地理空间数据集的访问机制, 包括 Web 要素服务、Web 覆盖服务、传感器采集服务和影像档案服务等; 处理服务提供操作地理空间数据的服务构建块和面向应用的增值服务, 包括服务连接服务、坐标转换服务、地理编码服务、地名服务和地理分析服务; 表现服务提供支持地理空间信息可视化的专业功能, 给出 1 个或多个输入, 能生成描绘后的输出, 包括 Web 地图服务, 栅格图表现服务和 Web 地形服务等。

在空间信息系统中, 空间数据占有非常重要的位置, 空间信息服务的处理能力也取决于空间数据, 因此对于空间信息服务的描述应加入能够处理空间数据的描述, 即空间元数据信息。空间元数据一般包括: 元数据实体集信息、标识信息、限制信息、数据质量信息、参照系信息、内容信息和分发信息。本文主要加入与服务输入输出相关的元数据信息, 即服务输入输出这些参数的值域信息。以表现服务 WMS(Web 地图服务)为例, WMS 能够根据用户的请求返回相应的地图。WMS1.1.1 版本主要定义了 3 个操作, 分别是 GetCapabilities、GetMap 和 GetFeatureInfo。GetCapabilities 返回服务级元数据, 是对服务信息内容和可接受要求参数的 1 种描述, 请求参数有请求版本(VERSION)、服务类型(SERVICE)、请求名称(REQUEST)等; GetMap 返回 1 个地图影像, 其地理空间参数和大小参数是明确定义了的, 请求参数有 VERSION、REQUEST、空间参考系(SRS)、以 SRS 单位表示的边界框(BBOX)、地图输出格式(FORMAT)、以逗号隔开的 1 个或多个图层列表(LAYERS)、请求图层的再现风格列表(STYLES)等; GetFeatureInfo(可选)返回显示在地图上的某些特殊要素的信息, 请求参数除了 GetMap 操作的所有参数外还包括 x(用像素表达的要素 x 坐标)、y(用像素表达的要素 y 坐标)、QUERY_LAYERS(需要查询的图层列表)。另外, 服务的元数据还包括所处理的数据范围、时间范围、比例尺、分辨率、空间参照系统类型、数据表示类型、数据质量等。因此, 服务匹配时除了服务类型和支持的版本需要和用户的请求进行匹配, 用户请求的数据也必须在服务所能处理的数据范围之内, 否则将会导致服务失败。

2 OWL-S 服务描述模型

OWL-S 是基于 OWL 语言的 Web 服务本体, 是 W3C 推荐的标准。OWL-S 定义了 1 组核心的语言构件, 用于对 Web 服务进行逻辑化描述, 所生成的描述文件支持机器理解, 从而支持代理程序基于逻辑语义实现 Web 服务的自动发现、调用、组合与监控^[4]。OWL-S 由 3 部分组成: Service Profile, 描述服务的功能、属性与接口, 如服务提供什么功能、服务的输入输出、服务的前置条件等; Service Model, 指明服务的工作模式, 它描述了服务流程模型以实现多个服务的组合, 同时在服务流程执行时协调各参与者的活动并监测服务的运行; Service Grounding, 指定调用服务的具体细节, 提供服务的一些底层信息, 描述访问服务的方法, 如网络协议、消息格式、调用端口等^[5]。其中, Service Profile 主要用于服务的发现, 而 Service Model 和 Service Grounding 主要用于服务流程执行和服务调用中, 本文主要介绍 Service Profile, 并结合空间信息服务对其进行扩展。

2.1 服务的基本信息

这些描述主要是供人理解的, 包括服务的名称, 提供者联系信息, 服务的类型或其他以文本形式给出的描述, 以便于用户定位所需服务时进行参考。服务的类型与 OGC 规范一致, 这里增加了空间服务支持的版本信息。以 WMS 为例:

```
<profile:supportversion>
<profile:version>1.0</profile:version>
<profile:version>2.0</profile:version>
<profile:version>3.0</profile:version>
</profile: supportversion >
```

2.2 服务的功能描述

这一部分主要是根据空间信息服务互操作规范, 对各类服务的输入、输出参数以及服务执行所需的先决条件和服务执行所产生的效果进行描述, 另外, 根据前面所述, 还要根据空间元数据信息描述各参数的值域。同样以 WMS 的输入参数 SRS 为例:

```
<profile:input>
  <profile:ParameterDescription rdf:ID="SRS">
```

```

<profile:parameterName> SRS</profile:parameterName>
<profile:restrictedTo rdf:resource="&WMSProcess;#SRS"/>
<profile:refersTo rdf:resource="&WMSProcess;#SRS"/>
<profile:parameterValue>经纬度</profile:parameterValue>
</profile:ParameterDescription>
</profile:input>

```

2.3 服务的质量描述

ServiceProfile 中虽然具有服务的质量保证、质量等级等,但没有提供 QoS 度量属性的详细定义。本文加入服务的响应时间(time)、服务成本(cost)、服务可靠度(reliability)。此外,由于空间服务对数据质量要求很高,因此还要加入数据质量的信息。

3 服务发现

根据空间信息服务的特点,本文主要采用三阶段匹配:首先进行服务的基本信息匹配,进行第1级过滤;然后在此基础上进行服务的功能匹配;最后是服务的质量匹配,根据用户偏好对服务进行排序,返回最优化结果。

3.1 基本信息匹配

空间信息服务在注册时,其分类一般都比较明确,因此可首先按照服务类型进行查找,得到这一类型的所有服务。在进行服务版本匹配时,若服务实例支持多个版本,则客户请求的版本只要与其中1个匹配即可。

3.1.1 字符串匹配

对服务的名称、版本采用字符串匹配,匹配方法是基于“编辑距”来计算的。编辑距是由俄国科学家 Vladimir Levenshtein 提出的1种确定2个字符串差距的方法,它定义了将1个字符串转变成另1个字符串所需插入、删除和替换字符的最小数目。根据编辑距的概念,给出字符相似度的计算公式。假定进行匹配的2个元素为 S_1, S_2 , e 表示编辑距, $|S|$ 表示元素 S 的长度,则 S_1 和 S_2 的相似度为:

$$\text{SynSim}(S_1, S_2) = \max \left[0, \frac{\min(|S_1|, |S_2|) - e(S_1, S_2)}{\min(|S_1|, |S_2|)} \right] \quad (1)$$

3.1.2 文本相似度匹配

服务文本描述匹配采用文本相似度匹配算法^[6],本文采用的是在字符串匹配的基础上实现文本的匹配。文本相似度计算主要有2个步骤:一是抽取关键字;二是计算关键词组的相似度。关键字抽取先按照 WordNet 本体将文本抽取成相应的单词,并且过滤掉一些诸如语气词、助词等虚词,将抽取出的单词按照词性分成名称、动词、形容词和副词4类,在不同的词性集合中计算单词的相似度,最后综合所有单词的相似度即得到文本相似度。具体算法如下:

```

TextSim(Req_Text, Adv_Text)
for each Req_keyword in Req_Text
    for each Adv_keyword in Adv_Text
        Sim1=max(Sim1, SynSim(Req_keyword, Adv_keyword))
    end for
Sum1+=Sim1
end for
Sim1 = Sum1 / Req_Text.count
for each Adv_keyword in Adv_Text
    for each Req_keyword in Req_Text
        Sim2=max(Sim2, SynSim(Req_keyword, Adv_keyword))
    end for
Sum2+=Sim2
end for
Sim2= Sum2/Adv_Text.count
Sim=(Sim1+Sim2)/2
return Sim

```

3.2 服务功能匹配

服务的功能匹配主要是请求服务的输入/输出与发布服务的输入/输出之间的匹配。根据空间信息领域本体计

算输入/输出在本体中的继承关系。本文采用 Paoluci^[6]提出的服务匹配算法并加以改进。

此算法中的类别有 4 种: 完全匹配(Exact)、插拔匹配(Plugin)、包含匹配(Subsume)、匹配失败(Fail)。即在完全匹配之外又增加了 3 种匹配类别, 其中匹配失败类别可以被看作默认类别, 只要不属于前 3 种类别都归于匹配失败。其中插拔匹配和包含匹配 2 种类别的划分用到领域本体中类和类之间继承关系, 因此可以被看成是基于本体的语义匹配。以输出为例: 当用户请求输出与服务输出相同时, 为完全匹配; 当用户请求输出是服务输出的直接子类时, 也为完全匹配; 当服务输出包含用户请求输出时, 为插拔匹配; 反之, 当用户请求输出包含服务输出时, 为包含匹配; 否则, 为匹配失败。由于在服务的功能描述中还存在对输入/输出值域的描述, 因此还要进行请求参数值和提供参数范围之间的匹配, 即检查服务的输入/输出参数的值域是否包含用户请求的参数值。例如当请求 WMS 服务的输入参数比例尺为 1: 4 000 000, 而发布的 WMS 服务提供的比例尺为 1:1 000 000 和 1: 10 000, 这样虽然服务参数类型完全匹配, 但因用户的请求值不在服务提供的范围之内, 所以匹配失败。

核心算法可以如下表达:

degreeOfMatch(outR,outA):

1) if outA=outR and outA.value contain outR.value then return exact

2) if outR subclassOf outA and outA.value contain outR.value then return exact

3) if outA subsumes outR and outA.value contain outR.value then return plugin

4) if outR subsumes outA and outA.value contain outR.value then return subsumes

5) otherwise fail

其中, outR 表示用户请求的输出, outA 表示服务提供的输出, subclassOf 表示子类关系, subsume 表示包含关系。具体的推理算法在文献[3]中已详细描述, 这里就不再赘述。

3.3 服务质量匹配与排序

服务的质量匹配与排序是基于 QoS 约束和用户偏好(如有的用户关心服务执行的价格, 有的用户关心服务执行的时间等)的最优服务选择。QoS 约束主要包括服务的响应时间(time)、服务成本(cost)、服务可靠度(reliability)以及数据质量(dataquality)等。用户在提交服务发现请求时除了指定服务的 QoS 约束值外, 还允许根据自身偏好指定服务各个 QoS 参数的权值。根据用户的请求从多个服务中选择满足 QoS 约束和用户偏好的最优服务返回给用户。本文参考文献[7]的研究成果, 采用多属性决策的方法来进行最优服务选择。整个求解过程分为 3 个步骤:

1) QoS 过滤

首先把满足用户功能需求的服务基于用户的 QoS 约束对这组服务进行过滤, 过滤算法主要根据 QoS 的相似度来完成, 实验中用 reS 表示服务请求的 QoS, adS 表示发布服务的 QoS, 则:

$$QoSSim(reS, adS) = \sqrt[3]{match(time) \times match(cost) \times match(reliability) \times match(dataquality)} \quad (2)$$

式中 $match(dim) = 1 - \frac{|adS \cdot dim - reS \cdot dim|}{reS \cdot dim}$, dim 代表各参数名称。

假设过滤后得到满足用户 QoS 约束的 n 个服务, 以 WMS 服务为例来说明选择操作的过程, 则每 1 个满足用户功能需求和 QoS 约束的服务 QoS 参数组成 1 个 4 维的元组, 由 n 个服务的 QoS 参数可以得到 1 个服务选择的多属性(多 QoS 维)决策矩阵 Q , Q 的行对应于每 1 个服务的 QoS 参数, 列表示服务 QoS 模型的 1 个维度。

$$Q = \begin{bmatrix} q_{1,1} & q_{1,2} & q_{1,3} & q_{1,4} \\ \vdots & \vdots & \vdots & \vdots \\ q_{n,1} & q_{n,2} & q_{n,3} & q_{n,4} \end{bmatrix} \quad (3)$$

2) 归一化

基于用户的偏好信息, 本文利用简单附加权重对步骤 1) 得到的满足用户功能需求和 QoS 约束的服务进行评估和比较, 来确定满足用户偏好的最优服务。服务的不同质量要素往往具有不同的度量单位、不同类型的取值范围。各个质量要素可分为消极维度和积极维度 2 种。消极维度指取值越高服务质量越差的 QoS 参数, 如执行时间和服务成本; 积极维度指取值越高服务质量越好的 QoS 参数, 如可靠性和数据质量。对不同的服务进行评价, 必须首先对服务的不同质量要素进行规范化处理, 即首先需要多属性决策矩阵 Q 进行归一化操作。

消极维度的值可通过式(4)进行归一化操作, 积极维度的值可通过式(5)进行归一化操作:

$$Q_{i,j} = \begin{cases} \frac{q_j^{\max} - q_{i,j}}{q_j^{\max} - q_j^{\min}}, & \text{if } q_j^{\max} - q_j^{\min} \neq 0 \\ 1, & \text{if } q_j^{\max} - q_j^{\min} = 0 \end{cases} \quad (4)$$

$$Q_{i,j} = \begin{cases} \frac{q_{i,j} - q_j^{\min}}{q_j^{\max} - q_j^{\min}}, & \text{if } q_j^{\max} - q_j^{\min} \neq 0 \\ 1, & \text{if } q_j^{\max} - q_j^{\min} = 0 \end{cases} \quad (5)$$

式(4)~式(5)中: $q_j^{\max}$ 为矩阵 Q 中第 j 维的最大取值, 即 $q_j^{\max} = \text{Max}(q_{i,j}) (1 \leq i \leq n)$; $q_j^{\min}$ 为矩阵 Q 中第 j 维的最小取值, 即 $q_j^{\min} = \text{Min}(q_{i,j}) (1 \leq i \leq n)$; 矩阵 Q 经过归一化操作后得到矩阵 Q^* :

$$Q^* = \begin{bmatrix} Q_{1,1} & Q_{1,2} & Q_{1,3} & Q_{1,4} \\ \vdots & \vdots & \vdots & \vdots \\ Q_{n,1} & Q_{n,2} & Q_{n,3} & Q_{n,4} \end{bmatrix} \quad (6)$$

3) 加权计算

在归一化基础上, 基于用户偏好信息可以对各个服务的质量进行度量; 用户偏好描述了用户对服务的各个 QoS 维度的感兴趣程度, 假设用户提供的服务偏好信息为 $W = \{w_1, \dots, w_4\}$, 其中 $\sum_{i=1}^4 w_i = 1$, 各个服务的质量度量值可利用下式进行计算:

$$\text{Score}(gws_i) = \sum_{j=1}^4 (Q_{i,j} * w_j) \quad (7)$$

通过上式对满足用户功能需求和 QoS 约束的 1 组服务进行度量, 并按照度量值对服务进行排序, 把结果返回给用户来满足具体应用需求。

4 实验与性能评价

4.1 实验内容与设置

实验环境包括: windows 2000 server, Java JDK1.5.0, UDDI, Tomcat5.5, Jena, Protégé3000。Jena 是 HP 公司开发的推理机, 能够对 RDF 模型进行读、写、浏览和查询。Protégé3000 是斯坦福大学开发的本体和知识编辑器。

首先进行服务发布, 对空间 Web 服务使用扩展后的 OWL-S 进行描述, 其中引用了空间信息领域本体, 根据 OWL-S 对统一描述、发现和集成(Universal Description, Discovery and Integration, UDDI)进行扩展, 将 OWL-S 发布到 UDDI 中。然后根据用户请求的服务描述和空间信息服务描述进行匹配。第 1 步, 用户输入特定功能的服务需求描述, 包括服务基本信息、功能描述、质量要求以及用户偏好。第 2 步, 系统得到服务查询请求时, 用 OWL-S 解析器对服务需求进行解析, 提取出服务名称、版本、描述、输入输出参数、输入输出数据的元数据、质量参数及用户偏好, 然后将这些信息送给服务发现引擎。

第 3 步, 将 UDDI 中的服务描述与服务请求的各项信息先后通过“3 个阶段”的匹配, 过滤掉匹配失败的服务。第 4 步, 将匹配成功的服务按照质量信息和用户偏好进行排序并返回结果。为了对比实验结果, 本文也同时进行了基于关键字的查询, 系统模型如图 1 所示, 图中①为基于语义的匹配, ②为基于关键字的匹配。

4.2 实验结果与性能分析

与基于关键字的服务匹配相比, 在基本信息匹配时, 使用字符串匹配可以设置不同的匹配阈值来缩放服务查询的范围; 在功能匹配时, 引入了领域本体, 使得输入输出匹配时相关的实体可以匹配, 因此提高了服务的查全率。通过相似度的设置, 可以精确表示服务描述和服务需求之间的匹配程度, 因此可以提高服务的查准率。

在实验中, 分别对 5 000 个模拟空间信息服务进行基于关键字和基于语义的服务, 统计发现, 基于关键字的方法和本文提出的发现方法的平均查全率为 46% 和 86%, 平均查准率为 36% 和 92%。这里设置各个阶段的相似度阈值均为 60%, 返回的是与查询条件相似度在 50% 以上的空间信息服务。实验结果表明, 基于语义的服务匹配的查准率和查全率明显高于基于关键字的匹配。

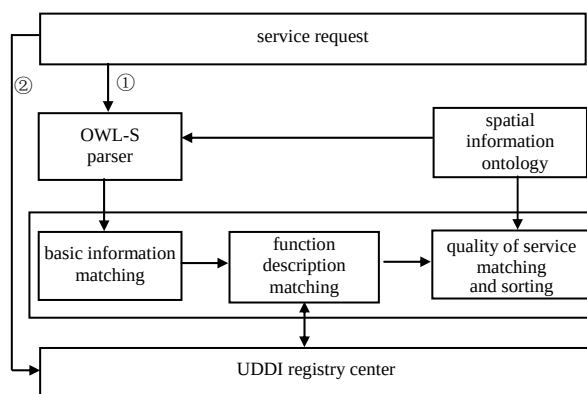


Fig.1 System model of service matching
图 1 服务匹配系统模型

5 结论

OWL-S 是用本体来描述 Web 服务的标记语言, 本文针对空间信息服务的特点对 OWL-S 进行了适当的扩展, 使其可以更加精确地描述空间信息 Web 服务, 同时在这些扩展的基础上, 基于语义匹配技术提出了“三阶段”空间信息服务发现算法, 分别针对基本信息、功能描述、服务质量对服务进行匹配, 并根据用户的偏好对服务进行了优化排序, 提高了查找的精确度。最后通过实验证明基于语义的服务匹配的查准率和查全率明显高于基于关键字的匹配。由于 OGC 的 Web 服务规范还在不断的完善之中, 因此, 文中提到的 OWL-S 服务描述也应随之改进, 另外, 在服务的匹配算法中没有考虑前提和效果的匹配, 这些都是下一步工作研究的方向。

参考文献:

- [1] 王俊, 于爱荣, 曹雷, 等. 分布式 Web Service 在 Net 下的应用研究[J]. 信息与电子工程, 2005, 3(2): 144-147. (WANG Jun, YU Airong, CAO Lei, et al. Research on applications of distributed Web service based on NET[J]. Information and Electronic Engineering, 2005, 3(2): 144-147.)
- [2] 张书亮, 陶陶, 闫国年. 地理信息共享与互操作框架研究[J]. 测绘科学, 2004, 29(6): 58-61.
- [3] Buehler K, McKee L. The OpenGis guide, OpenGIS Consortium, Inc[EB/OL]. (2000-05-11). <http://www.opengis.org/techno/guide.htm>.
- [4] MARTIN D. OWL-S: semantic markup for Web service[EB/OL]. (2005-12-10). <http://www.w3.org/Submission/OWL-S>.
- [5] Jeff Heflin. OWL Web Ontology Language Overview[EB/OL]. (2004-02-10). <http://www.w3.org/TR/2004/REC-owl-features>.
- [6] Corley C, Mihalcea R. Measuring the Semantic Similarity of Texts[C]// Proceedings of the ACL workshop on EMSEE'05. Michigan: [s.n.], 2005.
- [7] 刘书雷. 基于工作流的空间信息服务聚合技术研究[D]. 长沙: 国防科技大学, 2007.

作者简介:



魏 斌(1977-), 男, 四川省绵阳市人, 讲师, 主要研究方向为计算机网络. email: 282732607@qq.com.

刘 迎(1974-), 女, 河南省三门峡市人, 讲师, 主要研究方向为计算机信息技术.