文章编号:1001 - 9081(2005)01 - 0233 - 03

基于语义的 Web 服务自动发现、匹配及执行平台

周中成[1,2],孙荣胜[1]
(1. 江南大学 信息工程学院,江苏 无锡 214036; 2. 苏州科技大学 应用数学系,江苏 苏州 215011)
(bluesky7245@sohu.com)

摘 要:基于 UDDI 技术的普遍的 Web 服务发现、匹配、执行平台有可能产生严重的效率及服务响应延迟问题,文中介绍了一个限定于特定领域的 Web 服务自动发现、匹配及执行系统以试图解决此问题,同时给出了一个基于 OWL-S 本体语言的系统实现。在服务选择过程中实现了地域匹配功能,并为服务请求代理提供了一个初步的行为记忆功能。

关键词:语义 Web 服务;MAS;本体
中图分类号: TP393;TP311.13 **文献标识码**:A

Platform of automatic discovery, match and execution of Web services based on semantics

ZHOU Zhong-cheng[1,2], SUN Rong-sheng[1]
(1. *College of Information Engineering, Southern Yangtze University, Wuxi Jiangsu* 214036, *China*;
2. *Department of Applied Mathematics, Suzhou University of Science and Technology, Suzhou Jiangsu* 215011, *China*)

Abstract: A platform of automatic discovery, match and execution of Web services which uses UDDI technology in service discovery can result in severe problem of response latency of services. A platform which was restricted in a specific domain was introduced. At the same time, a system based on OWL-S ontology language was realized. A function of area match in the service selection was given. A function of behavior remembrance was also provided to the service requests.

Key words: semantic Web services; Multi-agent system; Ontology

0 引言

由于语义的引入,作为集成语义 Web 技术和 Web 服务技术而产生的语义 Web 服务使我们有望构建自动的 Web 服务发现、匹配及执行系统,其基本目标可以表述为:在面对大量的同类型 Web 服务可供选择的情况下,用户可以免除手工寻找、选择 Web 服务这种效率低下的工作,而由程序(典型的程序是智能软件代理)根据用户自己的状态、要求及 Web 服务的参数实现自动寻找、选择 Web 服务并自动执行优选出来的 Web 服务。其中一种典型想法是试图建立普遍的 Web 服务自动发现、匹配及执行平台,其中的服务发现机制一般是采用目前的逻辑集中式 UDDI 技术[1]。随着 Web 服务的广泛普及,UDDI 服务器中的注册服务数量可能会异常庞大,服务请求者将很快发现服务响应时间的不可忍受的延迟,而服务发现平台提供者(UDDI 服务器)也将很快发现服务器面临巨量的访问请求并可能因此而出现问题。这样的系统很可能因为其低下的效率而难以得到有效的应用。

本文提出一种限定于特定领域的 Web 服务自动发现、匹配、执行平台,例如书籍相关领域的 Web 服务平台,其中可能包括各书籍服务提供商的书籍在线浏览、书籍在线销售 Web 服务等。因为将服务平台限定应用于特定的领域而不是无所不包的范围,通过相对高效的服务发现机制(例如本文提出的 JADE 平台的 DF 机制),将可以实现比较高的系统效率,将服务延迟限定在用户可接受的范围内。另外,因为系统平台限定于特定的领域,服务各参与方相对比较容易达成共识并在此基础上构造公共本体,从而可以方便实现在机器层次上的自动处理。

基于上述思想,我们给出了一个基于 OWL-S 本体语言的系统实现。作为一种具有语义表达功能的 Web 服务描述语言,OWL-S 本体语言在本系统中扮演着重要角色。它使服务提供者(在本系统中是提供服务的软件代理)与服务使用者(在本系统中是请求服务的软件代理)可以共享公共语义从而实现机器间的互操作,为自动化处理提供了基础。另外,正是 OWL-S 所提供的语义支持,使本系统在自动选择、匹配过程中具备了初步的智能性,能够初步体现用户的需求。

1 语义 Web 服务本体语言 OWL-S 及其扩展

实现语义 Web 服务的关键步骤是对 Web 服务进行语义描述。OWL-S 就是一种描述 Web 服务的本体语言。OWL-S(以前叫做 DAML-S)基于语义 Web 标准 OWL[2]。OWL-S 的高层本体结构见图 1[3]。Resource 代表提供 Web 服务的 Web 资源,类 Service 作为 Web 服务本体分类的根节点,代表 Web 服务自身。OWL-S 本体结构是根据 Service 的三个属性:presents、describedBy 和 supports 来建造的,这三个属性分别涉及 Web 服务的三个最基本的问题:此 Web 服务提供什么样的服务;Web 服务具体是怎样工作的以及服务调用方具体怎样

收稿日期:2004 - 05 - 28;**修订日期**:2004 - 08 - 28

作者简介:周中成(1973 -),男,硕士研究生,主要研究方向:Web 服务、语义 Web、基于组件的软件工程; 孙荣胜(1944 -),男,副教授,主要研究方向:Web 服务、语义 Web、软件工程.

使用 Web 服务。这三个问题分别由上述三个属性的 range——ServiceProfile、ServiceModel 及 ServiceGrounding 来回答。

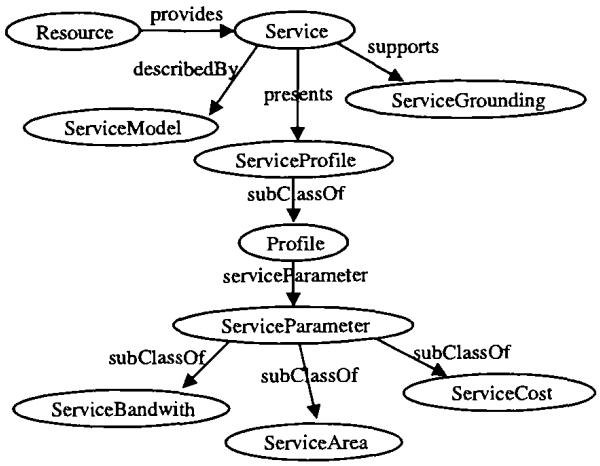

图 1 OWL-S 的高层本体结构及对 ServiceParameter 的扩展

ServiceProfile 的主要功能是提供 Web 服务的相关信息供用户(人)或服务查找代理(程序)确定服务是否符合查找要求(一般而言,这些信息是关于 Web 服务的总体简略信息,某些情况下,为使搜索、匹配过程更为精确,可能还需要 ServiceModel 中的详细信息)。ServiceModel 描述服务执行的具体过程。ServiceModel 中最重要的实体是过程(Process)。过程共有三种类型:原子过程(AtomicProcess)、简单过程(SimpleProcess)及组合过程(CompositeProcess)。原子过程不包含子过程,而组合过程由子过程(原子过程或组合过程)组成,一般而言,含有组合过程的 Web 服务即是组合 Web 服务。Servicegrounding 提供细节信息让用户或智能软件代理知道如何存取服务。一般来说,grounding 要给出通信协议、消息格式以及其他和特定服务相关的细节如联系服务时的端口号。在当前的版本中,grounding 还需要依赖 WSDL 来完成 Web 服务的调用执行。

另外,系统设计需要 Web 服务提供者在服务描述文件(系统设计采用 OWL-S 语言)中提供服务地域、服务费用及最大服务响应属性,这个任务由对类 ServiceParameter(见图 1,ServiceParameter 是 Profile 的属性 serviceParameter 的 range,而 Profile 是 ServiceProfile 的子类)进行扩展来完成,系统定义了 ServiceParameter 的三个子类 ServiceArea、ServiceCost 以及 ServiceMaxResponse。

2 系统结构及运行原理

因为软件代理(Agent)及 MAS(Multi-Agent System)系统的特点,MAS 平台非常适合作为系统的实现平台[3,4]。本系统的 MAS 平台采用了开放源码项目 JADE。除了提供系统服务的代理(如目录设施 DF)外,系统中共有两种类型的代理:服务提供代理与服务请求代理,服务请求代理与服务提供代理都有服务类型标识,服务请求代理只与同类型的服务提供代理进行协商交互,具体见图 2 所示(实际系统会更为复杂,图 2 只是简单示例)。用户首先通过代理选择器选择一个服务请求代理,服务请求代理在 JADE 主容器的目录设施 DF(在一定程度上,可以认为 DF 替代了 UDDI 在当前常见的 Web 服务应用架构中的角色,即提供服务发现机制)中查找同类型的服务提供代理,然后获得各服务提供代理相应的 Web 服务(一个服务提供代理对应一个 Web 服务)的参数,结合自身的状态和参数,根据一定的算法选择最优服务提供代理。被选中的服务提供代理通过封装了访问 Web 服务功能的类——Web 服务转接类来执行 Web 服务。Web 服务转接类调用语义 Web 服务 API——OWL-S API 执行远程或本地 Web 服务(OWL-S API 还要调用 Axis API)。另外,为管理方便起见,系统利用 servlet 技术构建了基于 Web 的服务提供代理管理平台,可以通过 Web 页面远程启动、关闭服务提供代理或将其从主容器的目录设施 DF 中注销。

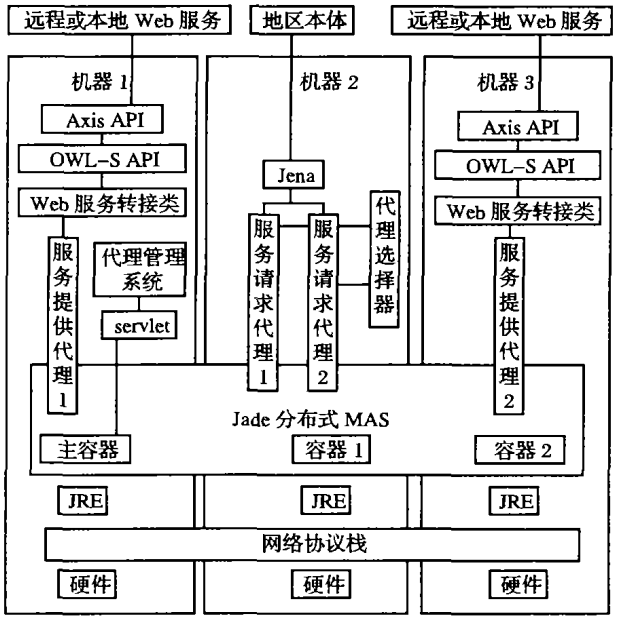

图 2 系统总体结构

3 系统的具体实现

3.1 语义 Web 服务 API 与语义 Web 服务的执行

系统利用马里兰大学信息及网络动态实验室(MIND)开发的 OWL-S API 来具体实现语义 Web 服务的调用执行,此 API 目前还处于 Beta 版阶段,在使用过程中,修正了其中的几个错误并编码实现了几个当前 API 还未实现的功能(例如关于服务参数的获取功能)。系统采用 OWL-S 来描述 Web 服务并使用相应的 OWL-S API 来调用执行 Web 服务,这使本系统具有下述重要特点:

(1)可以实现组合 Web 服务(composite Web services)的执行。语义 Web 服务技术的重要目的之一就是支持 Web 服务的组合(composition)[5]。利用组合过程(CompositeProcess)及数据流(DataFlow),OWL-S 具有表达 Web 服务间交互的能力,故而,OWL-S 可以作为组合 Web 服务的描述语言,同时,OWL-S API 也给出了组合 Web 服务的执行功能。目前,系统要求服务提供者手工编写 OWL-S 文档并在其中定义组合过程来具体实现组合 Web 服务的功能(这是组合 Web 服务的静态执行方法),当然动态的组合 Web 服务执行需要比较复杂的运行机制,本系统下一步将提供基于 OWL-S 语言的动态组合 Web 服务执行功能;

(2)更完善的服务描述。用于描述 Web 服务的 WSDL 并不能很好地表达 Web 服务的语义信息,这对服务信息的共享、服务执行的自动化都是很大的技术障碍[8],这也是系统

采用 OWL-S 来描述服务的原因之一。语义 Web 服务描述语言 OWL-S 在 ServiceProfile 中提供了标识服务特征(如服务提供者的联系方法、服务分类等)的一系列方法。本系统中要求服务提供者提供服务地域、最大服务响应及费用参数以供服务请求者作为选择依据,这些参数都是在 Web 服务提供方所提供的服务描述文件中给出的。

3.2 地域匹配功能的实现

这里假定 Web 服务具有服务地域的限制,例如亚马逊在中国就不提供网上书籍销售 Web 服务,其服务地域为美国。当然,也许国内的书籍销售商会提供中国地域的网上书籍销售 Web 服务(当然目前还没有),一个中国消费者要求其服务请求代理完成书籍购买任务,此服务请求代理就应该将亚马逊 Web 服务及其相应的服务提供代理排除在候选者之外。每个服务请求代理都标识其所在地域,每个服务提供代理都从相应的 Web 服务描述文档(采用 DAML-S 或 OWL-S 本体语言)获得服务地域参数(如果没有标识此参数,默认认为服务地域没有限制,即可以提供全球服务)。服务请求代理要匹配服务提供代理的服务地域属性,只选择包含了其所在地域的服务提供代理。

地域匹配功能基于 Jena(Jena 是开发基于 W3C 语义 Web 标准 RDF 及 OWL 的语义 Web 应用的 Java 开发工具箱)开发实现。系统首先采用 OWL 本体语言建立了地域本体,在其中定义了类地域(Area)及地域的子类国家(Country),并创建传递属性(owl:TransitiveProperty)子地域属性(subAreaOf)来表达地域之间的隶属关系,子地域属性的 domain 与 range 都来自地域类(Area),本体构造具体参见图 3(c)。例如地域类实例"苏州市"是地域类实例"江苏省"的子地域,"江苏省"又是国家类实例"中国"的子地域,而"中国"又是"全球"的子地域,如图 3(a)所示。假如服务请求者的地域为苏州市,服务提供者的服务地域为中国,本体没有直接声明二者的子地域关系,而在本体中穷尽这种隐含的子地域关系是不现实且没有必要的。隐含的子地域关系(子地域属性属于 owl:TransitiveProperty,具有传递性[6])由具有推理功能的推理机(Jena 采用基于规则的推理机)来推理得出,系统中的推理机实际采用了 OWL DL 推理规则[6,7]。

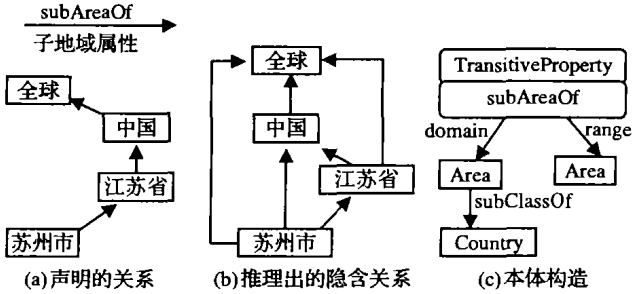

图 3　本体构造及演示实例

3.3 服务自动选择的实现及服务请求代理的行为记忆功能

系统的服务请求代理对用户的行为具有简单的记忆功能,它可以记住用户的选择偏好。有的用户对服务的质量要求较高,如要求服务响应很快,延迟很小,而费用可以不予考虑;有的用户对服务费用比较敏感,对服务质量(如服务响应时间)要求不是很苛刻;当然,更多的用户可能要在这二者之间实现某种平衡。对每个服务请求代理,系统利用一个文件记录用户输入的偏好选择次数。每次用户启动服务请求代理时可以有两个选择:一是不选择用户偏好,让代理根据用户以前的行为自动选择服务提供者,各个偏好的次数不变,这时,代理实际是参考用户的历史偏好来进行服务选择;二是用户给出各个偏好的相对比例,如服务响应:费用 = 5:5(系统要求输入的各个偏好取值范围在 0 ~ 10 之间,并要求各偏好之和为 10),即表示用户本次执行要同等重视服务响应与费用,这时,要相应增加各偏好的次数(本例中各增加 5),同时,代理不参考用户的历史偏好选取,直接根据用户输入的偏好相对值来进行服务选择。

对每个在 JADE 主容器中注册并符合地域匹配的同类型服务提供代理,服务请求代理要计算出其相应的用户偏好值并选择取值最大的服务提供代理,计算公式如下:

$$\text{服务的用户偏好值} = \frac{\text{服务响应选择次数}}{\text{次数总和}} \times \left(100 - \frac{\text{最大服务响应}}{6}\right) + \frac{\text{费用选择次数}}{\text{次数总和}} \times \left(100 - \frac{\text{费用}}{10}\right)$$

费用(假设取值范围在 0 ~ 1 000 单位货币)与服务响应延迟(假设取值范围在 0 ~ 600 秒)的具体数值由服务提供代理从服务描述文件(OWL - S 语言)获得并通过 ACL 消息发送给服务请求代理。由公式显然可知,服务的用户偏好值的取值范围必然在 0 到 100 之间。

4 结语

系统通过调用自己开发的一个实验性语义 Web 服务以及 MIND 提供的语义 Web 服务 DictionaryService(本体语言版本为 DAML-S 0.7,其服务描述文件为 http://www.mindswap.org/2002/services/Dictionary.daml)获得成功运行。

本系统在 MAS 平台之上构建了特定领域的 Web 服务自动发现、匹配、执行平台,在服务选择匹配过程中实现了地域匹配功能,并给服务请求代理提供了一个初步的行为记忆功能。本系统的某些功能还不是很完善,将来希望在以下几个方面进行改进:一是增加动态组合 Web 服务的执行能力,这样可以比较方便地在已有 Web 服务基础之上提供高效的增值服务;二是增加更多的判断逻辑;三是设计一个通用的服务提供代理注册机制(最好是将其标准化),以降低服务提供方的编程复杂性。

参考文献:

[1] 飞思科技产品研发中心. Java Web 服务应用开发详解[M]. 北京:电子工业出版社, 2002.483 - 484.

[2] W3C. World Wide Web Consortium Issues RDF and OWL Recommendations[EB/OL]. http://www.w3.org/2004/01/sws-pressrelease.html.en.

[3] 李海刚, 吴启迪. 多 Agent 系统研究综述[J]. 同济大学学报(自然科学版), 2003, 31(6):728 - 729.

[4] 史忠植. 智能主体及其应用[M]. 北京:科学出版社, 2000.6 - 7.

[5] The OWL Services Coalition. OWL-S: Semantic Markup for Web Services[EB/OL]. http://www.daml-s.org/owl-s/1.0/owl-s.pdf, 2003 - 12 - 27.

[6] SMITH MK, WELTY C, MCGUINNESS DL. OWL Web Ontology Guide[EB/OL]. http:// www.w3.org/TR/2003/PR-owl-guide-20031215/, 2003 - 12 - 15

[7] BECKETT D. RDF/XML Syntax Specification (Revised)[EB/OL]. http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/, 2004 - 2 - 10.

[8] 岳昆, 王晓玲, 周傲英. Web 服务核心支撑技术:研究综述[J]. 软件学报, 2004, 15 (3):434 - 435.