

《智能信息处理》课程考试

基于本体的网络服务查询机制

曹福笑

考核	到课[10]	作业[20]	考试[70]	课程成绩[100]
得分				

2021 年 12 月 11 日

基于本体的网络服务查询机制

曹福笑

(大连海事大学 信息科学技术学院, 大连 116026)

摘 要 为解决当前网络服务缺乏有效查询机制的问题, 最大化利用网络服务, 提出一种基于语义网络的网络服务查询机制。在该机制中, 采用本体描述网络服务的服务信息、领域相关和领域无关的基本概念信息, 并结合已经实现的基于本体推理的面向 Web 分布式本体系统, 引入领域规则, 结合使用本体推理引擎 F—Logic, 实现了网络服务智能发现系统。同时, 提出了网络服务约束模型(主体, 属性, 值, 动作)和语义补充模型。最后, 通过一个具体的实例, 演示了该系统的推理过程和运行结果, 说明了网络服务查询的自动化, 并对相关工作进行了对比。

关键词 本体; 网络服务; 语义网;

ontology—based Web services query mechanism

Cao Fuxiao¹⁾

¹⁾(School of Information Science and Technology, Dalian Maritime University, Dalian 116026)

Abstract To solve the problem of lacking effective services query mechanism in existing Web services and maximize the using of Web services, a semantic Web technology based Web services query mechanism was proposed. With this mechanism, the Web services related ontology information, domain dependent and independent basic conceptual ontology information were adopted. Combined with a former system Web Oriented Distributed ontology System (WODOS) which supported ontology — based reasoning, a new system Web Services Intelligent Discovery System (WSIDS) based on domain rules and F—logic reasoning engine was implemented. At the same time, Web services constraint model: Subject, Property, Value, Action (SPVA), and semantic supplement model were proposed. Finally, an application case was used to illustrate reasoning process and running result of this system, and was also compared with other related tasks.

Key words Ontology; Web services ; semantic web

1 引言

Web Service 技术, 能使得运行在不同机器上的不同应用无须借助附加的、专门的第三方软件或硬件, 就可相互交换数据或集成。依据 Web Service 规范实施的应用之间, 无论它们所使用的语言、平台或内部协议是什么, 都可以相互交换数据。简单的说, Web Service 就是一种跨编程语言和跨操作系统平台的远程调用技术。所谓跨编程语言和跨操作平台, 就是说服务端程序采用 java 编写, 客户端程序则可以采用其他编程语言编写, 反之亦然。跨操

作系统平台则是指服务端程序和客户端程序可以在不同的操作系统上运行。远程调用, 就是一台计算机的应用可以调用其他计算机上的应用。例如: 支付宝, 支付宝并没有银行卡等数据, 它只是去调用银行提供的接口来获得数据。还有天气预报等, 也是气象局把自己的系统服务以 web service 服务的形式暴露出来, 让第三方网站和程序可以调用这些服务功能。

简言之, 语义 Web 提供的是一种网络语义智能研究方向上的扩展, 而 Web Services 提供的是一种网络服务协作机制研究方向上的扩展。通过合理的设计, 可以充分发挥它们各自的优势, 最终提供一

种基于语义的自动 Web Services 协作机制。这种协作机制主要有 4 个研究环节：Web Services 的自动发现、被发现的 Web Services 的自动执行、Web Services 的自动组装和交互，以及 Web Services 的自动执行监控。

本文所要解决的正是这种自动 Web Services 协作机制中的重要一环，即基于语义的自动 Web Services 发现机制。随着 Web Services 不断走向实际应用，所提供的 Web Services 也越来越丰富，简单的基于统一描述、发现和集成协议(Universal Description, Discovery and Integration, UDDI)的 Web Services 发现机制已经不能满足这种不断发展的需要。例如，一个正在使用 Web Services 的用户可能会有这种要求：为我查找出能提供如下条件的从北京飞往上海的机票的 Web Services。条件是：①希望能够订到国内航空公司的机票；②最好有打折，并且打折越低越好；③此航班最好能使用我持有的某种卡。那么，这个人需要通过某种基于 UDDI 的搜索引擎查到所有订机票的 Web Services，然后仔细阅读每个 Web Services 的描述信息，或者通过执行每个 web Services 来最终确定使用某个 Web Service 订机票。

当前，实现基于语义 Web 的 Web Services 自动发现机制的一个核心思想就是使用某种语义标注语言，来标注和描述 Web Services 信息，之后通过语义 Web 自身已经实现的机器间自动交互，以及对本体信息的自动推理等功能来加强 web services 发现机制的自动性和智能性。这些语义标注语言包括 DAML+OIL 以及被 w3C 列为推荐标准的资源描述框架(Resource Description Framework, RDF)和 Web 本体描述语言(Web Ontology Language, OWL)等。本文将语义 Web 应用于 Web services 中的服务自动发现中，通过分析 Web 服务发现中使用的知识，将其划分为 5 个语义描述层次，并根据其所描述知识的特点分别进行建模。在此基础上描述了一个语义 Web 的 Web 服务自动发现的框架。

2 本体的概念及相关理论

2.1 本体的概念

本体 (Ontology) 的概念源自于哲学领域，在哲学中的定义为“对世界上客观事物的系统描述，即存在论”。哲学中的本体关心的是客观现实的抽象本质。而在计算机领域，本体可以在语义层次上

描述知识，可以看成描述某个学科领域知识的一个通用概念模型。德国学者 Studer 在 1998 年给出了本体的相关定义“本体是共享概念模型的形式化规范说明”。这个定义包含了四层含义：即共享(share)、概念化 (Conceptualization)、明确性 (Explicit) 和形式化 (Formal)。共享：指本体中体现的知识是共同认可的，反映在领域中公认的术语集合；概念化：指本体对于事物的描述表示成一组概念；明确性：指本体中全部的术语、属性及公理都有明确的定义；形式化：指本体能够被计算机所处理，是计算机可读的；

2.2 本体的层次分类

根据形式化的程度，可以将分为以下几类：非形式化、半形式化和形式化。其中，非形式化主要包括自然语言，形式化则是形式语言，而半形式化则包括结构化的自然语言和程度低的形式语言。

根据应用领域的不同，主体研究的重点也不同。具有普遍意义的能接触到客观世界常识的本体被称为顶级身材。特定学科领域的主体被称为领域主体。解决某个问题的解决方案主体被称为任务主体。与问题解决相关的主体被称为应用主体。

其中，顶层本体：研究通用的概念以及概念之间的关系，如空间、时间、事件、行为等，与具体的应用无关，完全独立于限定的领域，因此可以在较大范围内进行共享；领域本体：研究的是特定领域内概念及概念之间的关系；任务本体：定义一些通用任务或者相关的推理活动，用来表达具体任务内的概念及概念之间关系；应用本体：用来描述一些特定的应用，既可以引用领域本体中特定的概念，又可以引用任务本体中出现的概念。各本体间关系如图 2.1 所示。

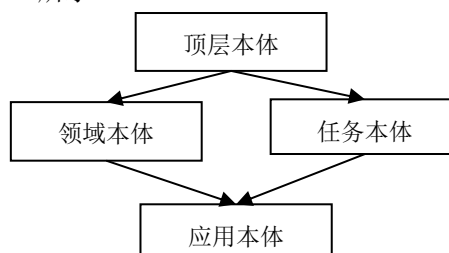


图 2.1 本体分类及关系图

2.3 本体的构建方法

TOVE 法: TOVE 是指多伦多虚拟企业(Toronto Virtual Enterprise)专门用于构建 TOVE 本体(关于企业建模过程的本体); METHONTOLOGY 法: 专用于构建化学本体(有关化学元素周期表的本

体)；骨架法：专门用于构建企业本体，建立在企业本体基础之上，是相关商业企业间术语和定义的集合；KACTUS 工程法：是基于 KACTUS 项目而产生的，KACTUS 是指关于多用途复杂技术系统的知识建模工程。七步法：斯坦福大学医学院开发的七步法，用于领域本体构建。七步骤为：确定领域本体的范畴、复用现有的本体、列出领域内的术语、定义类和类的等级关系、定义类的属性、定义属性的分面；基于叙词表的领域本体构建：叙词表又称为主题词表，是一种语义词典，由术语及术语之间的关系组成，能够反映某学科领域的语义相关概念。

2.4 语义网体系

Berners-Lee 提出了最初的语义网体系结构[5]，随着人们对语义网的深入研究，语义网的体系结构也在不断地发展演变。图 2.2 给出了语义 Web 的体系结构。

第 1 层：基础层，主要包含 Unicode 和 URI(Uniform resource identifier)。其中 Unicode 是一种流行的字符集，采用两字节的全编码，可以表示 65536 个字符，这使得任何语言的字符都可以被机器容易地接受。URI 即通用资源标识符是用于唯一标识抽象或物理资源的简单字符串。网络上的任何资源包括 HTML 文档、程序、图片、音视频等都有一个能被 URI 编码的地址，从而实现对 Web 资源的定位。

第 2 层：句法层，核心是 XML 及相关规范。XML 是 SGML(标准通用标记语言)的一个子集，它以一种自我描述的方式定义数据结构。在描述数据内容的同时能突出对结构的描述，从而体现出数据之间的联系。

第 3 层：资源描述框架，主要包括 RDF 及相关规范。RDF 是一种用于描述 WWW 上资源信息的通用框架，比如网页的内容、作者以及被创建和修改的日期等。

第 4 层：本体层，即定义本体(Ontology)。该层在 RDF 的基础上定义了 RDFS(RDF Schema)和 OWL(Web Ontology Language)帮助用户构建应用领域相关的轻量级的本体。RDFS 和 OWL 定义了语义，可以支持机器在用 RDFS 和 OWL 描述的知识库和本体中进行推理。

第 5 至 7 层分别是逻辑层(Logic)、验证层(Proof)、信任层(Trust)。逻辑层在前面各层的基础上进行逻辑推理操作。

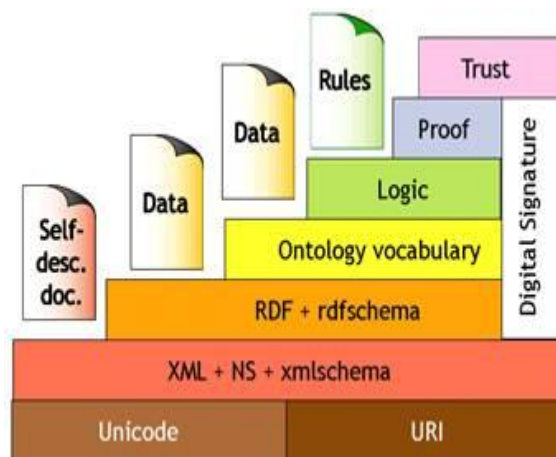


图 2.2 语义网体系结构

3 Web Service 智能发现系统模型

3.1 数据模型设计

此模型的设计目标是建立一个基于本体对 Web Services 进行智能发现的机制与实例，其中本体建立在目前的语义 Web 标准基础之上，也就是说，此模型将是一个在语义 Web 环境下进行 Web Services 智能发现的系统模型。本文提出了一个抽象的层次数据模型，该层次数据模型包含两个不同领域的结合，即 Web Services 层和语义 Web 层。Web Services 层在语义 Web 层之上，基于语义 Web 层的语义处理过程实现发现机制。这样设计的层次结构，从功能和结构的角度自顶向下应分成 5 个层次，分别为应用层、服务层、逻辑层、语义层和数据层。其中，应用层和服务层归于 Web Services 层，而逻辑层、语义层和数据层归于语义 Web 层。如图 3.1 所示。这 5 个层次功能紧密相连，相辅相成，上一层调用下一层的功能，并为下一层传递请求数据和控制指令；下一层作为上一层实现的基础，提供相应的服务，并返回结果数据。

(1)数据层(data layer) 包含 4 种异构数据库，存储不同的数据。Fact DB 用于存储各种用 OWL/RDF 描述的本体信息；Rule DB 存放各种推理规则信息(这些信息的格式可能会与采用不同的推理引擎相关)；Web Services Registry 存储被用语义标注过的 Web 服务描述语言；Consumer Registry 存储消费者信息，此信息由消费者提供。

(2)语义层(semantic layer) 包括 4 种功能各异的模块，主要目的是把用 OWL—s 描述的服务本体信息、WSDL 描述的服务信息和 OWL 描述的领

域相关的服务信息，存入数据层的相应库中，同时用语义标注 WSDL 文件。

(3) 逻辑层(logic layer) 主要封装推理引擎和查询逻辑。此层次中可以采用任何推理引擎，例如在 Web 服务智能发现系统(Web Services Intelligent Discovery System, WSIDS)中采用 F—Logic 推理引擎，此 F—Logic 实现基于一个前期成果面向 Web 的分布式本体系统(Web Oriented Distributed Ontology System, WODOS)，即可以实现基于 RDF 查询语言(RDF Query Language, RQL)查询的 F—Logic 推理。

(4) 服务层(Web Services layer) 该层主要对服务提供者所提供的服务数据进行处理，使之最终用于 Web Services 发现中。

(5) 应用层(application layer) 该层是整个系统中的最高抽象层，主要指在系统框架之上开发的领域相关的应用系统。因此，应用层渗透在其他 4 层之间，并总括其他 4 层，为具体的应用确定相应的领域规则和服务描述，并提供相应的 GUI 人机交互接口。

3.2 系统结构模型设计

此结构模型提供 3 种角色定义作用于数据模型的各个层次，实现不同的功能。

(1) 服务提供者:即 Web Services 的提供商，为服务消费者提供相应的服务数据。此模型不仅需要一个描述 Web Service 的 WSDL 文件，还要求服务提供者提供一个有某种固定格式的服务本体描述文件，这个服务本体描述文件可以采用 OWL—S 标准进行定义。

(2) 服务消费者：即服务所面向的用户，。用户在初次登录时，需要输入用户自身的兴趣规则以及本身的基础信息，此信息可以采用 F—Logic 规则以及 OWL 来共同描述，并可以在再登录时被修改或删除。

(3) 系统管理员：其作用是管理整个系统，并通过一个统一的接口界面提供推理规则信息(如 F—Logic 规则信息)和基本本体信息。其中，推理规则信息包括领域相关的规则信息和领域无关的规则信息(即描述用户喜好的规则信息)。

3.3 系统实现

WSIDS 的实现基于前期工作的成果，即本体支撑软件 WODOS.WODOS 本身可以对用 RDF 描述的 本体进行存储、导出、查询和推理。经过扩展的 WODOS 最新版本可以把用来描述 Web Services 的

WSDL 文件转换成 RDF 本体文件，并进行存储。随着 W3C 将 OWL 标准列为描述本体的推荐标准，

在基本保持 WODOS 原有功能和结构的前提下，又实现了支持 OWL 本体文件的功能模块，作为 WSIDS 中描述本体信息的基础构件。作为对 WSIDS 一个概念层次上的建模，提出了网络服务约束模型。该模型对整个 WSIDS 系统中的各种元素、操作和关系进行了抽象化的定义，并给出了这些概念间存在的必要约束关系。整个系统包含约束模型和语义补充建模模型。

约束模型：该约束模型约束了操作集和关系集作用于元素集时对定义域(domain)和值域(range)的要求，即表中空白格表示无关联的定义域和值域；反之，表格中的操作或者关系的定义域和值域即为相对应的定义域和值域。例如，inherits 关系只能作用于定义域为 S(即主体)，值域也为 S 的元素上；同时，enables 关系是一个动作(A)对一个主体(S)的作用关系，因此其定义域为 A，值域为 S；其他关系或操作以此类推。因此，在 SPVA 模型基础上定义了 SPVA 模型的约束模型(如表 3.1)。

值域	定义域				
	主体(S)	属性(P)	动作(A)	属性值(V)	真/假(True/False)
主体(S)	继承(inherits)		使能(enables)		演绎(deductionOf)
属性(P)	有属性(hasProperty)				
动作(A)	执行(executes)		使能(enables)		演绎(deductionOf)
属性值(V)		取值(valueOf)		运算(operationOf)	
真/假(True/False)				比较(comparisonOf)	

表 3.1 网络服务约束模型

语义补充建模模型：一个服务描述由一个 WSDL 文件与 OWL 文件组成，SSDL 沿用传统的方式；OWL 文件(以后称为服务本体描述文件)对该 WSDL 进行补充说明。服务本体描述文件用于完成类似 OWL—s 中 Services Profile 和 Services Process 的功能。目前的服务本体描述文件按智能服务发现的要求，主要对服务的外在功能和接口标准进行描述，暂不涉及服务的调用过程。该服务本体描述文件的模型定义如图 3.1 所示。

具体来说，一个服务本体描述文件包括以下 3 方面：(1) 基本服务描述信息：这个基本信息中描述了此服务的合同信息、服务类别、服务名称、服务地域、服务具体描述以及服务参数。(2) 动态更新信息：目前，此信息中只包含一个子项，即服务质量等级，它会根据用户对服务的满意程度动态改变，进一步影响下一次服务发现。(3) 网络服务的输入、输出、前提、效果：这里定义了一个服务的 4 个语义属性，即输入、输出、前提、效果，

这4个语义属性在语义补充建模中起非常重要的作用,后面的实例中将会看到 IOPE 在 Web Services 发现中的应用。其中, Service Category 可以采用北美工业分类系统或通过标准产品和服务代码的分类方式进行类别的划分。

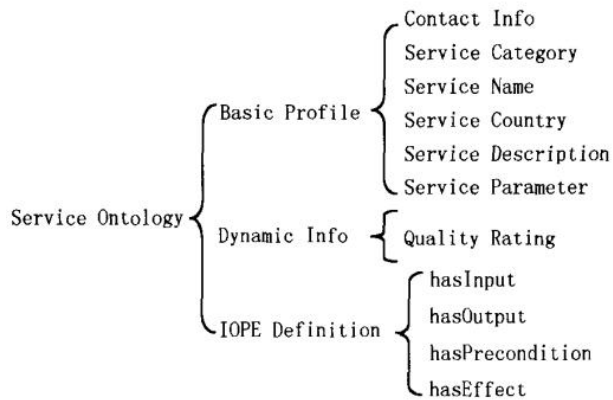


图 3.1 服务本体概念模型

4 总结

Web Services 是描述一些利用标准化的 XML 消息传递机制可以通过网络访问的操作的接口。Web Services 是用标准的、规范的 XML 概念描述的,称为 Web Services 的服务描述(WSDL)。这一描述囊括了与服务交互需要的全部细节,包括消息格式、详细操作、传输协议和位置。该接口隐藏了实现服务的细节,允许独立于实现服务基于的硬件或软件平台和编写服务所用的语言使用服务。这允许并支持基于 Web Services 的应用程序称为松散耦合、面向组件或跨技术实现。Web Services 履行一项特定的任务或一组任务。

本文提出了一种基于语义 Web 的 Web Services 发现机制,并且在语义 Web 支撑软件的工作基础之上实现了 WSIDS 系统。在开发研究过程中提出了比较有特色的本体库和规则库的概念,包括个人基本信息本体库、服务本体库、领域相关及领域无关的 F-Logic 规则库(其中 F-Logic 规则还扩展出了静态规则和动态规则的概念)等。还通过对 WSDL 补充建模的研究扩展了 WSDL 的语义信息,并用于 Web Services 发现的过程中。最后,对比了 SWWS 相关工作的进展情况,提出了一些亟待解决的问题。

除此之外,还有一些研究发展方向需要得到应有的重视。例如,在对 WSDL 做语义补充建模的时

候,如何将不同类别的 WSDL 语义信息集中地描述在此补充建模中;另外,引入数据挖掘的相关技术,通过对个人访问偏好的记录和建模,建立动态的个人兴趣模型,使得 Web Services 发现结果更接近用户的要求,也是一个很重要的发展研究方向。

参考文献

- [1] BERNERS-LEE T, HENDLER J, Lassila O. The semantic web[J]. Scientific American, 2001, 284(5): 34—43.
- [2] SOLLAZZO T, HANDSHUCH S, STAAB S, et al. Semantic Web service architecture evolving Web service standards toward the semantic web[A]. Proceedings of the 15th International FLAIRS conference[c]. Menlo Park, cal., usa: AAAI Press, 2002. 425—429.
- [3] FENSEL D, Bussler C. The web service modeling framework wsMF[J]. Electronic commerce Research and Applications, 2002, 1(2): 113—137.
- [4] SWARTZ A. MusicBrainz: a semantic web service[J]. IEEE Intelligent Systems, 2002, 17(1): 76—77.
- [5] SWARTZ A. MusicBrainz: a semantic web service[J]. IEEE Intelligent Systems, 2002, 17(1): 76—77.
- [6] FENSEL D, MOTTA E, VAN HARMELEN F, et al. The unified problem-solving method development language UPML[J]. Knowledge and Information systems(KAIS): An International Journal, 2003, 5(1): 83—131.
- [7] SRIHAREE N, SENIVONGSE T. Discovering web services using behavioral constraints and ontology[A]. Distributed Applications and Interoperable Systems(DAIS)[c]. Paris, France: Springer-LNCS, 2003. 248—259.
- [8] Knowledge systems Lab in stanford university. wine Agent 1.0[EB/OL]. <http://onto.stanford.edu:8080/wino/index.jsp>, 2003—02—07.