

知识路由中基于描述逻辑的语义 Web 服务复合的研究

摘 要

语义 Web 通过本体技术为 Web 资源提供了丰富的可供机器处理的语义信息,使得机器可以利用这些语义元数据进行自动化的信息访问,从而提供一种高质量的新服务。同时,语义 Web 研究的兴起也为 Web 服务技术带来了新的活力,这两种技术的结合产生了另一个新兴的研究课题——语义 Web 服务。语义 Web 服务是指用语义 Web 标记语言来描述服务的语义,使 Web 服务成为计算机可以理解的实体,从而支持服务的自动发现、执行和复合等。其中,服务的复合体现了 Web 服务的重要思想:未来的应用将由一组利用网络的服务复合而成。

同时,随着语义 Web 和语义 Web 服务的不断发展,面对语义 Web 上分布的海量服务和知识,如何在这些海量服务和知识中按照用户的需求快速准确的定位目标知识,如何将已有的 Web 服务自动地复合起来、协同工作完成用户的任务,使服务可以灵活地满足用户的需求,实现“按需服务”,也成为迫切需要解决的问题。

本文正是针对上述问题,结合国家自然科学基金《知晓内容和环境的本体知识路由研究》,利用语义 Web 相关技术、以 P2P 网络为底层架构、以本体为知识标记方法、以描述逻辑为语义 Web 服务复合逻辑基础,提出了一种用户需求驱动的、结合 P2P 网络和描述逻辑的本体知识路由机制,

并以此为背景，对其中语义 Web 服务复合进行了更为深入的研究和探讨。

主要工作成果包括以下四个方面：

- (1) 将 P2P 技术和本体技术相结合，利用描述逻辑概念包含性的基本原理，采用动态虚拟社区的思想，结合传统 P2P 网络结构所具有的分布性和开放性等特点，提出了一种本体驱动的基于 P2P 网络结构的本体知识路由机制，并详细讨论了知识路由算法的设计与实现。
- (2) 在此知识路由中，通过考虑用户需求偏好信息和扩展原有 OWL-S 模型，提出一种将用户需求偏好信息考虑在内的扩展 XOWL-S 模型；
- (3) 将描述逻辑所具有的良好知识表示能力和强大推理能力与 AI 规划所具有的行为变化建模能力相结合，以扩展 XOWL-S 模型为基础，提出一种面向知识路由的基于描述逻辑的语义 Web 服务动态复合机制 DLAIP (AI Planning based on DL)，并对其进行形式化证明。
- (4) 设计和开发基于本体的 E-learning 教学平台 OBETP (Ontology Based E-learning Teaching Platform)，该平台是知识路由实验系统的基础性工作，为系统的最终完成提供了有价值的开发经验。

关键词：知识路由，P2P，描述逻辑，服务复合，语义 Web

RESEARCH ON SEMANTIC WEB SERVICES COMPOSITION EMPLOYING DESCRIPTION LOGIC IN KNOWEDGE ROUTING

ABSTRACT

The Semantic Web provides affluent machine-readable semantic information for the web resources through Ontology. Based on these semantic metadata, the machine is able to achieve the automatic information access and therefore builds a kind of high-quality new service. At the same time, the birth of Semantic Web also brings new motivation for the Web Services development. The confluence of Web Services and Semantic Web gives birth to another new technology-Semantic Web Services. The Semantic Web markup language can be used to describe the semantics of Web and make them computer-interpretable entities so as to enable programs or Agents to locate, invoke and compose Web services automatically. Particularly, the services composition reflects one of the important initiatives of web: the future applications would consist of a series of Web Services.

At the same time, with the development of the Semantic Web and Web Services, there is more and more abundant knowledge in the Semantic Web.

Therefore, it also becomes much more critical and emergent to effectively and immediately acquire accurate knowledge and composite the existing web services to fulfill the user's requirements, meaning the so-called service on demand.

Considering the above problems, based on the Natural Science Foundation of China: *Research on Content & Context-aware Ontology Knowledge Routing*, this thesis proposes a kind of users' requirements-driven Knowledge Routing mechanism through employing the P2P and DL (Description Logic), and especially focuses on the Semantic Web Services composition. The main points of this thesis are as follows:

- (i) It designs an Ontology-driven Knowledge Routing mechanism based on the combination of P2P with Ontology, the employment of the concepts subsumption principle in DL and the introduction of Dynamic Virtual Community;
- (ii) In this Knowledge Routing, through considering the user preference and extending the OWL-S model, it proposes an extended OWL-S model XOWL-S through adding the user preferences into the original OWL-S model;
- (iii) Combining the strong knowledge representation and reasoning capabilities of the DL with the modeling capability of the action state transformation of the AI planning and based on the extended model XOWL-S, it presents a Knowledge Routing-oriented Semantic Web

services composition mechanism *DLAIP* (AI Planning based on Description Logic) and testifies its availability in DL;

- (iv) An Ontology Based E-learning Teaching Platform, called *OBETP*, is designed and developed. This platform is the initial work of Knowledge Routing system development and also provides the useful experiences for it.

KEY WORDS: knowledge routing, p2p, description logic, services composition, semantic web

声 明

本人郑重声明：所呈交的学位论文，是本人在指导教师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含其他个人或集体已经发表或撰写过的科研成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律责任由本人承担。

论文作者签名： 李英杰 日期： 2007.5.22

关于学位论文使用权的说明

本人完全了解太原理工大学有关保管、使用学位论文的规定，其中包括：①学校有权保管、并向有关部门送交学位论文的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③学校可允许学位论文被查阅或借阅；④学校可以学术交流为目的，复制赠送和交换学位论文；⑤学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

签 名： 李英杰 日期： 2007.5.22

导师签名： 李增明 日期： 2007.5.22

第一章 绪 论

本章首先介绍基于知识路由及语义 Web 的服务复合的研究背景；然后介绍本文的研究内容并概括本文的主要工作；最后说明论文的整体结构。

1.1 研究背景

World Wide Web (WWW)是一种在 TCP/IP 协议之上的特殊结构框架,通过链接机制可以访问遍布全球的 Internet 资源,有关 WWW 的各种技术统称为 Web 技术。当前 Web 技术的体系结构主要是基于 HTML 和 HTTP 协议。WWW 是一种基于 Client/Server 模式的 Browser/Server 结构:在服务器端存放有 URL 标识的各种 Web 资源,客户端通过浏览器访问 WWW 上的文档资源。

World Wide Web (WWW)自上个世纪 90 年代初由 Tim Berners-Lee 发明以来,经过 10 余年的发展,已经在 Internet 中完全占据了主导地位。随着 Web 迅速地普及,基于 Web 的应用也由最初的简单应用延伸到种类日益繁多的复杂应用和计算。然而,传统的 Web 体系结构已越来越不能满足新的发展需求,WWW 的潜力也还远远没有发挥出来。因此,近年来,WWW 的发展出现了两个新兴的重要发展趋势^[1]:

一个是 Web 服务(Web Services)。Web 服务是一些自包含、自描述、模块化的程序,可以发布在 Web 上,并被发现和调用。Web 服务使得 WWW 从静态的、松散的 Web 页面集合逐渐演化成动态的、互联的应用和服务平台。越来越多的服务提供商开始基于 Web 向用户提供服务,例如 B2B 应用、B2C 应用等^[2]。

另一个是语义 Web (Semantic Web)。语义 Web 的概念^[3]由 Tim Berners-Lee 于 1998 年首次提出。语义 Web 为 Web 资源提供了丰富的可供机器处理的语义信息,弥补了传统 Web 缺乏语义的不足。语义 Web 的基本思想就是提供基于机器可处理的语义元数据,并应用这些语义元数据进行自动化的信息访问,使得 Web 提供一种高质量的新服务。语义 Web 通过赋予 Web 中的所有信息以定义良好的语义,让计算机能够理解 and 处理,从而使人和计算机能够更好地协作,充分发挥 WWW 这个巨大的分布式信息系统的潜力。

语义 Web 的研究使 WWW 由信息的海洋变为知识的海洋。一般来说,在语义 Web 上,一个给定的知识库包括两类知识。一类是可以明确声明的服务知识,即可以使用单个类表达的知识;另一类则是无法使用单个类表达的知识,该类知识主要指的是有关领域方面的知识。基于这些丰富的知识语义信息,计算机便可智能地实现 Web 知识的发现、定位以及复合。但是由于语义 Web 上分布着海量知识,如何获取并有效利用这些语义知识,建立一种高效的知识路由发现机制就成为迫切需要解决的问题。

同时,语义 Web 的研究也为 WWW 上的另一种重要资源——Web 服务带来了新的活力。如果用语义 Web 的知识标记手段来描述服务的语义,将使得 Web 服务成为计算机可以理解的实体,从而支持服务的自动发现、执行和复合等^{[4][5]}。而 Internet 环境的动态、异构等特性要求基于 Web 提供的服务可以根据用户的需求匹配、发现和复合服务,从而灵活、可靠地满足客户的需求,实现“按需服务”。

此外,作为语义 Web 知识标记语言 OWL (Web Ontology Language)与 OWL-S (Web Ontology Language for Service)逻辑基础的描述逻辑(Description Logic)近年来也得到了迅速的发展。描述逻辑^{[6][7]}是一种建立在概念(Concept)和关系(Relation, Role)之上的非经典逻辑。其中概念解释为对象的集合,关系解释为对象之间的二元关系。描述逻辑具有合适定义的语义和很强的表达能力。一个描述逻辑系统包含四个基本组成部分:表示概念和关系的构造器;Tbox 包含断言;Abox 实例断言;Tbox 和 Abox 上的推理机制。其表达能力和推理能力取决于对以上几个要素的选择以及不同的假设。

在众多知识表示的形式化方法中,描述逻辑在十多年来受到人们的特别关注,主要原因在于:它们有清晰的模型-理论机制;很适合于通过概念分类学来表示应用领域;并提供了强大的推理服务。

本课题就是在研究语义 Web、语义 Web 服务及复合和描述逻辑等相关技术的基础上,提出了一种本体驱动的基于 P2P 网络结构的本体知识路由机制,并以此为基础,设计了一种面向用户需求的基于描述逻辑的语义 Web 服务动态复合机制——DLAIP (Description Logic AI Planning)。

下面分别就语义 Web、语义 Web 服务及复合和描述逻辑等概念作初步阐述。其具体内容将在后续章节详细介绍。

1.1.1 语义 Web 研究的兴起

语义 Web 是对未来 Web 体系结构的一个伟大构想, 被定义为“由一些可以被计算机直接或间接处理的数据组成的 Web”。当前基于 HTML 的 Web 网页是设计给人看的, 计算机并不能理解其中信息的含义。而语义 Web 的基本思想是通过本体(Ontology)和 Web 内容的语义标记, 使计算机(程序或智能 Agent)能够理解并处理 Web 上的信息, 从而更好地协助人们在 Web 上发现知识、处理事务。Ontology 是共享概念的形式化、显式的定义。在语义 Web 中, 隐含在数据、页面、程序和其它 Web 资源中的语义都通过合适的语义 Web 标记语言来显式表示, 构成一个基于知识的 Web, 使计算机可以智能地处理和集成这些信息, 从而使 Web 提供的服务实现一次质的飞跃, 因此语义 Web 技术被誉为“下一代的 Web 技术”, “Web 技术的革命”等。

语义 Web 研究的重点是如何把信息表示为计算机能够理解和处理的形式, 即带有语义的形式。Berners-Lee 在 XML2000 国际会议上提出了语义 Web 的层次结构^[8]。它主要基于 XML(eXtended Markup Language) 和 RDF (Resource Description Framework)/RDFS(RDF Schema), 并在此之上提供本体和逻辑推理规则, 以完成基于语义的知识表示(KR)和推理, 从而能为计算机所理解和处理。语义 Web 在理论上以知识工程(KE)成熟的理论、技术和方法为核心, 知识表示、Ontology 和智能 Agent 等都是重要的研究内容, 它们是不可分割的。

1.1.2 语义 Web 服务及复合

语义 Web 研究的兴起为 Web 服务带来了新的活力, 结合语义 Web 技术的 Web 服务(即语义 Web 服务)^{[9][10]}将是一种更为智能的服务, 是 Web 服务未来的发展趋势。

通过语义 Web 标记语言来标记 Web 服务、用户的需求以及 Agent 的处理过程, 并采用相应的能识别和处理这些标记的 Agent 技术, 这些目前 Web 服务技术不可能实现的任务都将通过语义 Web 服务来实现。在这种语义 Web 服务的方式下, 服务的发现、执行和复合不再需要由人来完成, 而是可以由 Agent 来自动完成。由于服务的描述是基于 Ontology 的, 是计算机能够理解的形式, 因此服务的查找可以是基于语义的匹配, 单个的服务也可以由 Agent 自动地复合起来, 以完成用户提出的复合任务。这就是服务的复合问题, 它包括服务的选取和执行。传统方式下服务的选取和执行通常是由用户来完成的, 而在语义 Web 环境下, 计算机则需根据必要的信息自动地、动态地选取和复合服

务,即语义 Web 服务自动复合。这些必要的信息就是语义 Web 服务的语义信息,其描述手段就是语义 Web 服务标记语言 OWL-S (Web Ontology Language for Services)。因此,语义 Web 服务复合就是研究如何有效地利用这些语义信息实现 Web 服务的自动复合的问题。但是,由于语义 Web 服务标记语言 OWL-S 的逻辑基础是描述逻辑,因此,反映的多是与领域或服务相关的静态语义信息,而对于描述服务复合等动态过程中的语义信息却存在着不足。而且,采用 OWL-S 描述语义 Web 服务时,并未考虑与用户需求相关的个人偏好和约束信息。

1.1.3 描述逻辑简介

描述逻辑^{[6][7]}是语义 Web 知识表示的逻辑基础,又称为术语逻辑或类 KL-ONE 系统,它的基本组成部分是概念(concept)、角色(role)和个体(individuals),简单的概念和角色可以通过复合方式表示复杂的概念和角色。根据不同的应用环境,描述逻辑有许多变种,OWL 就是基于其中的一类 SHIQ 描述逻辑。描述逻辑的推理功能集中在以下两方面:

- 包含(subsumption): 即判断一个概念是否为另一个概念的子集,主要用于概念的自动分类;
- 可满足性(satisfiability): 即判断一个概念与已有的概念集(本体)是否相容,主要用于进行概念集合的一致性检测。

采用描述逻辑可以实现对特定领域的语义建模。但是,描述逻辑反映的多是与领域或服务相关的静态语义信息,而对于描述服务复合等动态过程中的语义信息却存在着不足。

1.2 论文研究内容及意义

本文的研究工作是在语义 Web 及其相关技术的基础上展开的,主要研究内容包括:

- 建立一种本体驱动的知识路由机制,能够根据用户的知识需求,快速有效地发现、定位和返回目标知识;
- 对现有语义 Web 服务描述模型 OWL-S 进行扩展,建立反映用户偏好的扩展 OWL-S 模型 XOWL-S,并将其应用于语义 Web 服务动态复合;
- 结合扩展的 XOWL-S 模型,建立一种用户需求驱动的语义 Web 服务动态复合机制——DLAIP (AI Planning based on Description Logic),并对其合理性和复杂度

给予相应的形式化分析和证明；

- 设计和开发基于本体的 E-learning 教学平台 OBETP (Ontology Based E-learning Teaching Platform)。

本文的研究意义在于语义 Web 服务动态复合机制 DLAIP 不仅基于知识路由，面向用户需求，考虑了与用户需求相关的个人偏好和约束信息，弥补了 OWL-S 描述服务时未考虑用户偏好的不足，且具有良好的知识表示能力和强大推理能力，可以弥补当前语义 Web 标记语言 OWL、OWL-S 对服务复合等动态过程建模能力的不足，而且还可以通过利用 AI 规划方法解决服务复合时所具有的效率高的优点来弥补直接利用 OWL-S 语义进行服务复合效率低的缺点。因此，采用该机制，将会有助于提高语义 Web 上服务复合的效率和准确率。

1.3 论文结构

文章组织结构如下：

第一章绪论主要介绍了本文的研究背景以及所要研究的内容；

第二章主要介绍了语义 Web 服务复合的相关概念和关键技术，并对其进行初步分析，引出本体知识路由的概念；

第三章首先介绍和分析 P2P 网络的基本结构和特点，然后提出建立一种基于 P2P 网络的知识路由机制，并对其相关概念和工作机制进行详细叙述；

第四章提出了面向知识路由的基于描述逻辑的语义 Web 服务动态复合机制 DLAIP，并以描述逻辑为形式化证明工具，对该机制的合理性和复杂度进行了理论分析和证明；

第五章基于语义 Web 和语义元数据等相关技术，重点介绍了基于本体的 E-learning 教学平台 OBETP (Ontology Based E-learning Teaching Platform)的设计与开发；

第六章是对全文工作的总结和对进一步研究工作的展望。

第二章 语义 Web 和本体知识路由

本章首先分析和介绍语义 Web、语义 Web 服务及语义 Web 服务复合的相关概念，然后以当前 Web 发展所要求的快速、准确、高效地定位目标知识为宗旨，提出了本体知识路由的概念，并对其进行了初步的阐述。

2.1 语义 Web 对 Web 服务发展的影响

2.1.1 Web 服务概述

Web 服务是一种新兴的分布式技术，它的使用将改变目前的开发模式和应用部署的费用规模。Web 服务能够统一地封装信息、行为、数据表现以及商务流程，而无需考虑应用所在的环境是使用何种系统和设备。它成为目前应用环境中最为合理的解决方案^[1]。Web 服务提供了一种崭新模式：人们通过程序来自动启动和处理商业事务，并能够在分布式的计算环境中动态地描述、发布、发现和调用所需的服务，而不用关心服务在什么地方，以及它是如何实现。

Web 服务使用的是 SOA 架构(Service Oriented Architecture, 面向服务的架构)。Web Services 是独立的、模块化的应用，能够通过 Internet 来描述、发布、定位以及调用。Web Services 的体系架构中包括三个角色：服务提供者(Service Provider)，服务请求者(Service Requester)和服务注册器(Service Registry)。角色间有三个操作：发布(Publish)，查找(Find)，绑定(Bind)。如图 2-1 所示。

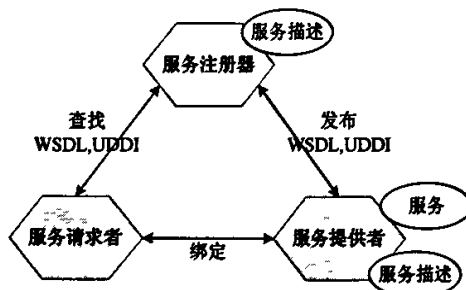


图 2-1 Web Services 的运行模式

Fig. 2-1 Web services running pattern

下面分别从 Web 服务发现机制 UDDI (Universal Description, Discovery and Integration) 和 Web 服务描述语言 WSDL (Web Service Description Language) 两方面对 Web 服务作进一步的阐述。

(1) Web 服务的基石 UDDI

UDDI^[12]是一套基于 Web 的、分布式的、为 Web 服务提供信息注册中心的实现标准规范,同时也是促使企业能将自身提供的 Web 服务进行注册,以使别的企业能够发现服务的访问协议的实现标准。它主要由 Ariba、IBM、Intel 和 Macrosoft 等公司共同推出,其 v1 版本于 2000 年 9 月正式发布,2001 年 6 月发布了 v2 版本,意图建立一个全球化的、平台无关的、开放式的架构,通过创建、使用一个全球性的商务注册中心,以共享信息在定义的 Internet 交互方式下使得企业能对外发布自己的服务并且轻松快捷地找到需要的 Web 服务,从而加速全球 B2B 电子商务的应用。

UDDI 的核心组件是 UDDI 商业注册,它使用一个 XML 文档来描述企业及其提供的 Web 服务。从概念上来说,UDDI 商业注册所提供的信息包含白页、黄页和绿页三个部分。所有的 UDDI 商业注册信息存储在 UDDI 商业注册中心中,通过 UDDI 注册,各企业可以将自身的描述、服务描述以及服务访问方式的描述公开发布。

通过使用 UDDI 的发现服务,企业可以单独注册那些希望被别的企业发现的自身提供的 Web 服务。企业可以通过 UDDI 商业注册中心的 Web 界面,或使用实现了“UDDI Programmer's API 标准”所描述的编程接口的工具,将信息加入到 UDDI 的商业注册中心。UDDI 商业注册中心在逻辑上是集中的,在物理上是分布式的,由多个根节点组成,相互之间按一定规则进行数据同步。当一个企业在 UDDI 商业注册中心的一个实例中实施注册后,其注册信息会被自动复制到其他 UDDI 根节点,于是就能被任何希望发现这些 Web 服务的人或单位所发现。

UDDI 核心技术包括核心部件、UDDI 运用、互操作协议栈、技术发现层、P2P (Peer to Peer)数据同步以及商务发现功能等六部分。

(2) Web 服务描述语言 WSDL

Web 服务的描述是进行服务发现和复合的基础。随着通信协议和消息格式在 Web 中的标准化,以某种格式化的方法来描述通信变得越来越重要,其实现的可能性也越来越大。服务提供者向一个或多个服务注册处发布服务描述。服务提供者使用服务描述通知服务请求者所有必要事项,以便请求者理解如何调用服务。服务描述对于查找操作也

同样是重要的。服务描述是服务请求者查找服务注册的基础，查找操作的最终结果就是获得服务描述。服务描述对于绑定操作的重要性体现在它确切描述了调用 Web 服务的消息格式以及目标网络地址。

目前大多数的 Web 服务描述规范都是语法性的^[13]，典型的的就是 WSDL。WSDL 由 Ariba、Intel、IBM 和 Microsoft 等开发商提出，采用标准的 XML 格式描述 Web 服务。它用一种和具体语言无关的抽象方式定义了给定 Web 服务收发的有关操作和消息。WSDL 定义了一套 XML 语法描述的网络服务方式。它是描述 Web 服务的规范。在 WSDL 文档中描述了服务的地址，所支持的操作以及通信的细节，比如对参数数目和参数类型等都作了详细的描述。将服务描述成 WSDL 的形式，服务提供者能够隐藏服务内部复杂的实现机制，而服务请求者能够绕过平台细节从而根据 WSDL 规范来调用服务。

一个 WSDL 文档包含了 Web 服务的四个重要信息。分别是：接口信息、数据信息、绑定信息以及地址信息。一个 WSDL 文档在定义网络服务时使用如下元素：

- 类型(Types)：一种使用某种类型系统（如 XSD）定义数据类型的容器，定义了所有数据类型集合。
- 消息(Message)：通信数据抽象的、有类型的定义，定义了通信中使用的消息的数据结构。
- 操作(Operation)：Web 服务支持的某种行为的抽象描述。
- 端口类型(PortType)：对于某个访问入口点类型所支持的操作的抽象集合，使用 Messages 部分的消息定义来描述方法（操作名称、输入输出参数等）。
- 绑定(Binding)：将特定端口类型的具体协议和数据格式规范绑定。
- 端口(Port)：定义为协议 / 数据格式绑定与具体 Web 访问地址组合的单个服务访问点，描述的是一个服务访问入口的部署细节。
- 服务(Service)：相关 Web 服务访问点的集合。

此外，WSDL 还定义了通常的绑定机制。这种机制用于为抽象的消息、操作或端点附加一种特定的协议、数据格式或结构。它允许抽象定义的重用。

2.1.2 语义 Web 概述

语义 Web 的目标是使得 Web 上的信息具有计算机可以理解的语义，满足智能软件代理(Agent)对 WWW 上异构和分布信息的有效访问和检索^[14]。语义 Web 的研究和开发

是按层次进行的,每一层都建立在更低层次的基础上^[15]。图 2-2 是 Tim Berners-Lee 为未来 Web 发展提出的基于语义的体系结构—语义 Web 体系结构:

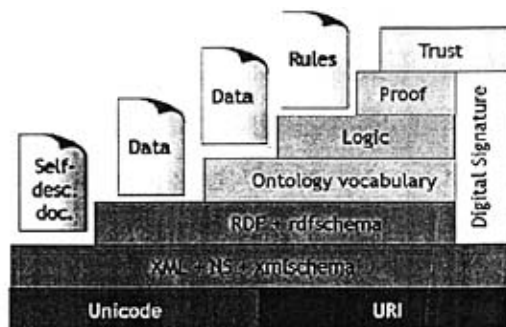


图 2-2 语义 Web 层次结构图^[1]

Fig. 2-2 The hiberarchy of semantic web^[3]

(1) URI 和 Unicode

Web 环境中的应用需要相互通信,资源之间直接或间接地以机器可理解的数据格式传递和发布信息。这些信息是对 Web 上资源的描述。因此,首先应该以明确的方式来标识这些资源。语义 Web 采用统一资源标识符(Uniform Resource Identifiers, URI)来标识资源及其属性,URI 是一个 Internet 标准,记载于 RFC2396^[16]。它和 WWW 常用的统一资源定位符(Uniform Resource Locator, URL)以及统一资源名称(Uniform Resource Name, URN)的区别在于 URI 泛指所有以字符串标识的网络资源,包含了 URL 和 URN。另外由于语义 Web 的最终目的是要构建一个全球信息的网络,在这个网络上应该涵盖各种语言和文字的信息资源,所以它采用统一编码 Unicode 作为字符的编码方案。这一层是这个语义 Web 的基石,它成功地解决了 WWW 上资源的定位和跨地区字符编码的标准格式的问题。

(2) XML、NameSpace 和 XML Schema

在 URI 和 Unicode 之上,是 XML 及相关技术层。XML 允许用户根据需要自定义一些“有意义的”标签对发布的内容进行标记,并使用文档类型定义(Document Type Definition, DTD)或 XML Schema 来约束这些标签的结构。由于 XML 标签可以由用户根据自己的需要来定制,这样不可避免地会造成标签同名的情况,为了避免这样的冲突,W3C 采用了 NameSpace 机制。例如:用户可以制定<student>标签:

```
<k:student xmlns:k=http://219.226.86.57/xml/student.dtd>
```

这样即使其他用户也自定义了<student>标签,只要它们的 NameSpace 不同,也不会造成冲突。因此,这一层通过 XML 的特性,实现了文档对自身结构的描述,实现了跨应用的语法互操作层,这是传统的 HTML 语言所无法完成的。有一点值得注意,<student>这个标签对人来说很容易理解,但对计算机而言,XML 标签<student>和 HTML 标签<H1>没有本质的区别,因为计算机并不能真正理解 XML 标签<student>的含义。也就是说,XML 是底层的数据交换格式,它只是解决了文档内容的次序,结构的问题,并没有解决文档内容的语义、联系的问题。标签的具体含义的定义和互操作要交给上一层去解决。

(3) RDF 和 RDF Schema

XML 层的上一层是数据互操作层——资源描述框架(Resource Description Framework, RDF)和 RDF Schema。RDF 本身并没有规定语义,但是它为每一个资源描述体系提供了一个能够描述其特性需求的语义结构的能力。它定义了一种机器可理解的数据语义的数据模型。该数据模型主要包含下面的三个对象类型:

- 资源(Resources): 资源可能是整个网页,或网页的一部分;或页面的全部集合;或者是不能通过 Web 直接访问的对象。
- 特性(Properties): 特性是描述某个资源特定的方面、特性、属性或关系。
- 声明(Statements): 一个特定的资源和特性名称加上该特性的值一起构成了一个 RDF 声明。一个声明中包含三个部分,分别称为:主体(subject),谓词(predicate)和对对象(object)。从本质上说,RDF 定义 object-property-value 三元组作为基本建模原语并为它们引入了标准的语法。

RDF 和 XML 之间的关系一直是一个容易混淆的问题,它们之间有着明确的功能分工: RDF 解决如何无二义性地描述资源对象的问题,使得描述的资源的元数据信息成为机器可以理解的信息。RDF 通过基于 XML 语法的明确定义的模型来帮助建立语义协定(RDFS)和语法编码(XML)之间的桥梁,并以此来实现元数据的互操作功能^[17]。这部分解决了部分通用语义的问题,但是 RDF/RDFS 描述语义的功能非常有限,需要进一步扩展。

(4) Ontology

RDF Schema 可以定义类、子类、超类,并且可以定义属性和子属性,以及它们的约束如:领域(domain)和范围(range)等,因此在某种意义上说,RDF Schema 本身就是一种简单的本体(Ontology)语言。但是 RDF/RDFS 对特定应用领域的词汇描述能力比较弱,需要进行扩展,我们把这个 RDF/RDFS 之上的扩展层称为 Ontology 层。

Ontology 作为一种能在知识层提供知识共享和重用的工具在语义 Web 中获得应用,能够描述 Web 资源的概念及其相互关系,使计算机对 Web 资源的无二义的自动处理成为可能,符合 Web 高度分散的特点。

(5) Logic

在 Ontology 层之上的 Logic 层主要定义逻辑描述原语,为更高层的逻辑推理提供语义上的支持。

(6) Proof 和 Trust

在 Logic 层之上的 Proof 和 Trust 层主要是在下面各层提供的功能基础上进行逻辑推理、证明等操作。

2.1.3 语义 Web 服务概述

语义 Web 研究的兴起为 Web 服务带来了新的活力,结合语义 Web 技术的 Web 服务(即语义 Web 服务)^{[18][19]}将是一种更为智能的服务,是 Web 服务未来的发展趋势,如图 2-3 所示。

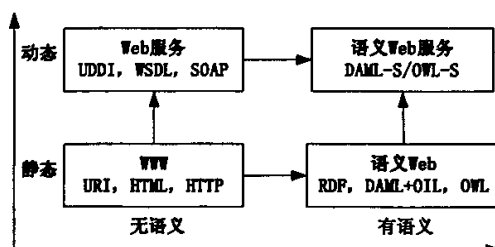


图 2-3 语义 Web 服务是 Web 服务发展的趋势

Fig. 2-3 Semantic web service is the trend of web service

表 2-1 给出了传统 Web 服务与语义 Web 服务几个主要的不同之处。下面通过举例来解释这些不同之处。

表 2-1 传统 Web 服务与语义 Web 服务的比较

Table 2-1 The comparison between semantic web service and web service

比较项	“传统”的Web服务	语义Web服务
服务	简单的	复合的
服务请求者	人	计算机(程序或agent)
服务描述	基于词汇	基于ontology
数据交换	基于语法的	基于语义的

例如,假设一位教授要从北京去上海参加一个国际会议,他想在网上预订机票和酒店房间。

在目前的 Web 上,他必须自己先通过门户网站查询到可以订购机票的网站。然后逐个访问,直至找到符合自己要求(如航空公司、付费方式等)的服务并访问调用它。然后再同样地去找合适的酒店房间预订服务并访问它。

在这种传统的方式下,Web 所提供的服务(如机票预订服务、酒店房间预订服务等)是简单的、单个的。对它们的查找、访问和复合是用户自己来完成的。而且这种查找和访问是基于关键词的,靠用户自己来理解服务的语义。

而在未来的语义 Web 上,用户可以委托自己的智能软件 Agent 帮他查找符合自己要求的服务,然后用户自己再去访问该服务,提交订单。或者,他可以进一步要求 Agent 直接帮他从找到的网站订购机票或订房间。再或者,他可以直接要求 Agent 帮他安排参加这次会议的所有行程(包括查找服务、订机票和订房间等)。

上述第一种应用是服务的自动发现,是指 Agent 自动地为用户寻找符合特定要求的服务。第二种是服务的自动执行,是指 Agent 根据用户的个人信息自动为该用户执行某个服务。第三种是服务的自动复合和交互,是指 Agent 自动选择、复合和交互服务来完成特定的复杂任务。

通过语义 Web 标记语言来标记 Web 服务、用户的要求以及 Agent 的处理过程,并采用相应的能识别和处理这些标记的 Agent 技术,这些目前 Web 服务技术不可能实现的任务都将通过语义 Web 服务来实现。在这种语义 Web 服务的方式下,服务的发现、执行和复合不再需要由人来完成,而是可以由 Agent 来自动完成。由于服务的描述是基于 Ontology 的,是计算机能够理解的形式,因此服务的查找可以是基于语义的匹配,单个的服务也可以由 Agent 自动地复合起来,以完成用户提出的复合任务。

2.2 语义 Web 服务复合的关键技术

支持语义 Web 服务自动复合的主要关键技术包括:

- 语义 Web 标记语言

Web 服务的语义信息需要通过合适的语言来标记。标记语言的语义表达能力和形式化支持直接影响到 Web 服务复合的自动化程度和灵活性。因此,语义 Web 服务自动复合技术与语义 Web 标记语言的发展密切相关。

- Web 服务的语义描述模型和本体

有了语义 Web 标记语言的支持, 还需要考虑描述服务的哪些信息, 也就是为 Web 服务建模。完整的模型既需要能描述服务的属性, 如功能信息、性能信息、物理信息等, 也需要能描述用户的需求、个人偏好和约束等。模型建立后, 需要用合适的语义 Web 标记语言将其描述为可以共享的本体。

- Web 服务的复合方法

如果服务资源的语义描述模型已经建立, 用户的需求描述模型也已经确定了复合服务需要实现的目标, 则需要考虑如何将多个子服务集成在一起完成复合服务的任务, 即研究服务自动复合规划的方法。

下面对上述三方面的相关研究做一个简单综述。

2.2.1 语义 Web 标记语言

在语义 Web 的发展过程中, 除了 W3C 的语义 Web 成员之外, 世界各地的各个研究机构和科研人员都在密切关注着它的发展, 从理论到实践、从标准到规范, 都在做着不懈的努力。其中, 语义 Web 标记语言的研究是其核心内容。

SHOE (Simple HTML Ontology Extension)^{[20] [21]}是美国 Maryland 大学 James Hendler 教授(也是后来的 DAML 项目和 W3C 语义 Web 项目的负责人之一)和他的学生们, 从 1995 年开始研究和开发的第一个实际意义上的基于 Ontology 的语义 Web 标记语言和原型系统, 从网页的语义标记语言到语义网页的爬行机器人再到基于语义的搜索引擎, 都有一个完整的展示。SHOE 的语法基于 HTML, 语义基于 Horn 逻辑。

德国 Karlsruhe 大学 AIFB 研究所的 Ontobroker^[22]是基于 HTML 语法和 Frame-logic 的语义 Web 语言。它在很多方面与 SHOE 有相似之处, 整个系统包括 Ontology 的定义语言、网页的标记语言、网络爬行者、推理机和查询接口。

欧共体的 On-To-Knowledge 项目^[23]资助开发的 OIL (Ontology Interchange Language 或 Ontology Inference Layer)^[24]也是一种在 Web 上描述 Ontology 的语言。主要的合作者有: 英国的 Manchester 大学、荷兰 Amsterdam 的 Vrije 大学、美国 Stanford 大学、德国 Karlsruhe 大学、荷兰 Administrator Nederland、美国贝尔实验室、MIT 大学等。OIL 通过扩展 RDFS 标准来克服 RDFS 的限制, 是在 RDFS 层之上增加定义的一层, 基于描述逻辑(Description Logic)以提供形式化的语义和推理功能。

美国 DARPA 资助的 DAML (DARPA Agent Markup Language)^[25]项目于 2000 年 8 月正式启动, 主要研究活动有: 制定该标记语言的规范; 研究和开发知识标记的工具; 构造能理解 DAML 的多 agent 系统等等。早期的 DAML 语言规范版本称为 DAML-ONT (2000 年 10 月版)。后来为了结合已有的工作, 吸收 SHOE、OIL 等其它一些标记方法最好的特性而定义一个统一的 Web Ontology 语言框架, 2000 年 10 月, DARPA 和欧共体 IST 联合组成了 US/EU Agent Markup Language 特别委员会, 包括了 XML、RDF、OIL、SHOE 等相关项目的核心人员。参与 DAML 研究的有 Stanford、Manchester、Karlsruhe、MIT、CMU、Yale、BBN 等 16 家科研机构和公司, 还包括很多 W3C 的成员, Tim Berners-Lee 自己也参与了该项目。由于 DAML 更多地和 OIL 保持了一致性, 因此后来的版本称为 DAML+OIL (2000 年 12 月版和 2001 年 3 月版)。

为了推出 Web Ontology 语言的标准, W3C OWL (Web Ontology Language)工作组在 DAML+OIL 的基础上进行了一定的改进, 于 2003 年 7 月 W3C 公布了 OWL 语言的最初工作草案。2004 年 2 月 10 日, OWL 正式成为 W3C 推荐的标准。

DAML+OIL 和 OWL 都是建立在 RDF 和 RDFS 之上, 基于描述逻辑以提供形式化的语义和推理功能, 从而为语义 Web 提供了 Ontology 层。Ontology 层的研究已经比较成熟, 但其之上的逻辑规则层还没有出现成熟的语言。因此, 目前的语义 Web 语言能表达的语义仅限于描述逻辑, 而不能表达一般形式的规则。

2.2.2 Web 服务的语义描述模型

DAML 框架下的 DAML-S (DAML-Service, 简称为 DAML-S)^[26]是国际上语义 Web 服务描述模型方面的主要研究成果。它是基于 DAML 语言为描述 Web 服务而定义的一个本体, 主要通过 ServiceProfile, ServiceModel 和 ServiceGrounding 三个类来描述服务做什么、服务如何做、服务如何访问等三方面的语义, 从而允许服务的自动发现、执行、复合和运行的监视。

随着 OWL 成为 W3C 推荐的 Web Ontology 语言标准, DAML-S 也演化为相应的 OWL-S^[27]。OWL-S 是用 OWL 语言写的本体, 因此它具有定义良好的语义。OWL-S 的上层本体如图 2-4 所示。

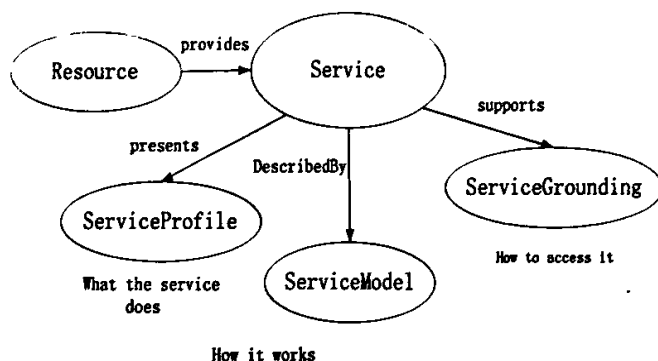


图 2-4 OWL-S 整体结构

Fig. 2-4 Framework of OWL-S

在 OWL-S 中，一个 Service 由三部分来描述 ServiceProfile，ServiceModel，ServiceGrounding。简单来说，ServiceProfile 描述服务是做什么的，ServiceModel 描述服务是怎么做的，ServiceGrounding 必须和一个 Service 相关联。

(1) ServiceProfile

在 OWL-S 框架下，ServiceProfile 主要指服务能给用户或智能主体提供些什么^[13]。它提供了搜索服务本体所必须的信息和服务的能力描述，从而能够决定这个服务本体是不是所需要的。ServiceProfile 告诉了服务做什么，即给出服务代理发现服务时所需要的信息类型，这些信息包括输入输出、前置条件、后置条件和绑定方式。

在 ServiceProfile 中，有三种基本的类型来描述服务，分别是：服务的提供者信息，服务的功能信息和一组标定服务特点的属性值。服务的功能描述通过该服务所产生的变换来表示。ServiceProfile 描绘了服务两个方面的功能：信息转换（通过输入即 Inputs 和输出即 Outputs 描绘）和服务执行产生的状态变换（通过前提即 Preconditions 和影响即 Effects 描绘），简记为 IOPE。例如，进行新生注册，注册服务要求输入学号和密码，而且以学号有效为前提。注册的结果就是输出确认该新生已经成功注册，可以产生进行新生的选课服务的影响。

Profile 本体中为 Profile 类定义了如下指示 IOPE 的属性^[13]：

- hasInput 属性指有一个 Input 参数，关联一个相应的类。
- hasOutput 属性指有一个 Output 参数，以一个在 Process 本体中定义的 ConditionalOutput 实例为值域。UnConditionalOutput 是 ConditionalOutput 的一个子类。

- hasPrecondition 属性指定了服务的前提并以一个根据 Process 本体的模式定义的 Precondition 实例作为值域。
- hasEffect 属性指定了服务的影响(effects)。它的值域是一个 ConditionalEffect 实例。类似 Output, UnConditionalEffect 是 ConditionalEffect 的子类。

(2) ServiceModel

Service Model 主要是服务提供者用来描述服务的内部流程,即描述服务是如何工作的。对于简单的服务,它描述服务的输入输出和执行的前提以及执行后产生的结果:对于复杂的服务,它还要描述服务的过程模型。这些信息可以被用于:

- 在找到此服务后做更深层次的分析;
- 当执行特定任务时对多个服务进行复合;
- 在服务运行时,对不同参与者进行协调工作;
- 监控服务的执行情况。

(3) Service Grounding

Service Profile 和 Service Model 都是关于服务的抽象描述,而 Service Grounding 是涉及到服务的具体的规范。服务是如何被访问的。这就是服务绑定 (Service Grounding)。它说明了如何访问服务的细节,包括通信协议,消息格式及一些其它细节,比如通信时的端口号等。另外,服务绑定必须详细说明在服务模型中所阐述的每一个抽象类型。

OWL-S 和当前的 Web 服务标准 UDDI、WSDL 等结合可以较好地支持未来的语义 Web 服务^[28]。如图 2-5 所示。

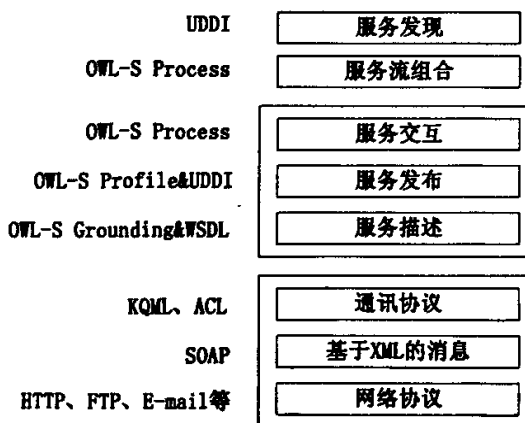


图 2-5 语义 Web 服务体系结构

Fig. 2-5 The semantic web services architecture

然而, 由于 OWL 在规则方面表达能力的局限性, OWL-S 模型没有包含用户在选择和使用服务时所表现出来的偏好和约束的描述, 因为它们通常表现为规则的形式。这使得用户偏好和约束不能为计算机所理解并处理, 意味着 Agent 在执行复合服务时如果遇到不确定的情况 (例如, 有多个服务可供选择) 则需要用户的干预, 还不能实现完全的自动化。

2.2.3 Web 服务的复合方法

在 Web 服务语义描述模型出现之前, 服务的复合一般是以基于 XML 的工作流描述语言和工作流技术为基础, 例如 HP 实验室的 eFlow 系统^[29], UC Berkeley 的 Ninja^[30], Florida 的 DynFlow^[31]和澳大利亚 NewSouthWales 大学的 SELF-SERV^[32]等, 它们基本上是一种静态组合、动态绑定的方式, 复合的自动化和动态适应性程度不高。

而 DAML-S 和 OWL-S 的出现为新的复合方法提供了可能, 它们将 Web 服务看成是人工智能(AI)中的行为(action), 用参数、前提和结果等来描述服务, 可以比较自然地映射为行为的形式化描述, 这使得服务的复合问题可以利用 AI 中的方法来解决。同时, 它们也是从 Agent 的角度出发来建模的, 服务可以当成是 Agent 的行为, 这样可以充分利用 Agent 的自治性、主动性和推理性等特性。

在基于语义服务模型复合服务的方法研究方面, Stanford 大学的知识系统研究室提出了一种基于 DAML-S 和 Agent 的服务自动复合技术^[33]。Maryland 大学给出了一种基于 DAML-S 和 SHOP2 的服务组合方法^[34]。SHOP2 是他们研究的一种适于过程模型的层次任务网络 (Hierarchical Task Network) 的任务规划器。

概括起来, Web 服务复合方法主要分为三类: 工作流模式、AI 规划模式和语义模式。

- (1) 工作流模式指的是利用原子服务根据静态或动态生成的工作流模型进行服务复合。如文献^[35]提出一种基于静态工作流模式的服务复合平台 EFlow。基于该复合平台, 可以实现 Web 服务的静态复合, 但无法动态生成服务复合链; 文献^[36]提出一种静态工作流模式和动态工作流模式相结合的服务复合机制 PPM。该机制可以初步实现 Web 服务的静态复合和动态复合, 但由于缺乏对语义的利用, 大大降低了服务复合的准确率。
- (2) AI 规划模式指的是采用传统 AI Planning 方法解决服务复合问题。如文献^[37]

提出一种基于情景演算的服务复合语言 Golog, 文献^[38]提出一种基于规则的服务复合机制 SWORD, 但无论采用 Golog 还是 SWORD 实现服务复合时都缺少对语义的利用, 而且在复合之前必须描述服务复合要遵守的约束, 缺乏对复合动态变化的描述。

- (3) 语义模式指的是通过利用服务的相关语义实现服务的复合。而语义 Web 服务描述语言 OWL-S (Web Ontology Language for Services) 的出现使得基于语义模式的语义 Web 服务复合成为可能。文献^[34]提出一种半自动化的语义 Web 服务复合的机制。但是, 该服务复合机制在复合过程中需要人的参与, 无法实现真正的 Web 服务自动复合。

2.3 本体知识路由概述

如前所述, 在语义 Web 上分布着海量知识。所谓知识, 是指信息经过加工整理、解释、挑选和改造而形成的、有结构的系统^[39], 它是实现语义 Web 的有效途径。然而, 对于传统 Web, 通过利用可视化的标记语言如 HTML 对资源进行标识, 提供了用户可以理解但机器不可理解的知识信息。因此, 相对于机器, 传统 Web 缺乏丰富的语义知识信息。而语义 Web 则通过本体技术为 Web 资源提供了丰富的可供机器处理的语义信息, 弥补了传统 Web 缺乏语义的不足。其基本思想就是提供基于机器可处理的语义元数据, 并应用这些语义元数据进行自动化的信息访问, 使得 Web 提供一种高质量的新服务。计算机利用这些语义信息便可实现知识的发现、定位以及复合等功能。但是, 由于语义 Web 上分布着大量异构而且来源于不同本体的知识, 因此如何在语义 Web 上实现最优语义知识的发现、定位和返回, 并建立一种高效的知识路由机制就成为迫切需要解决的问题。

从目前国内外的研究现状来看, 对知识路由机制已经有了一些相关研究, 主要分为两大类:

- (1) 基于传统关键字技术, 即从用户需求中提取关键字以寻找满足用户需求的知识及服务。一般说来, 采用关键字技术主要有三大缺点^[40]: 检索结果的低准确率、语义信息的低理解率和用户上下文信息的丢失性。
- (2) 基于语义技术, 即通过捕捉用户需求及与用户需求相关的语义, 包括需求内容和上下文信息, 使计算机能在这些语义信息的启发下定位满足用

用户需求的目标知识及服务。

在这些已有的相关研究中,文献^[41]针对智能信息提取系统对语义处理的要求,提出一种知晓文本内容的机制。该机制通过本体建模、语义标识以及智能推理,能够实现对文本语义的知晓。但是,由于本体建模、语义标识和智能推理所固有的复杂性以及高成本性,大大降低了该机制的效率,无法满足 Web 知识发现的效率要求。在文献^[42]中,针对现行基于用户模型捕捉用户语义的趋势,提出一种通用用户模型。采用该模型,可以有效地捕捉用户需求信息。但这种模型是一种静态模型,一经建立,就无法改变,缺乏模型的动态进化。文献^{[43][44]}则是针对普适计算领域中用户上下文信息比重较大、变化频繁的特定需求,提出一种通过对用户上下文信息进行感知理解,发现和定位目标资源的机制。但是,作为下一代 Web 发展趋势的语义 Web 及其相关应用,不仅要求感知用户上下文语义,而且更需要理解用户需求内容,从而找到满足用户需求的目标资源。

同时,在基于语义技术的知识路由相关研究中,最近也出现了一种基于 P2P 网络、本体 Ontology 和多 Agent 系统的语义搜索路由技术,引起了学术界的广泛关注。如文献^[45]中提出了一种基于半结构化数据模型 RDF 的语义搜索平台 TAP,通过使用 TAP 可以实现初步的语义检索,但是这种语义检索主要依赖于半结构化的 RDF 数据模型,没有考虑对本体建模语言 OWL 的处理。而在文献^[46]中,提出一种面向 P2P 网络的基于集中式索引的全文本搜索系统,在文献^[47]中,提出一种固定模式的信息检索算法,但是在这些信息检索系统中并未考虑基于本体的语义信息处理机制,而且仅仅只能从本地知识库中寻找目标信息,也没有考虑如何实现面向全局网络的知识信息检索。此外,在文献^{[48][49]}中,提出一种通过计算概念之间相似度而定位目标概念的机制。但是,该机制仅仅从概念角度考虑,并没有考虑针对特定用户需求的概念复合、知识社区划分等问题,而且当知识库中的概念较多时,该机制的效率将会变得较低。

本文正是基于上述研究背景,将 P2P 技术和本体技术相结合,以语义 Web 技术为基础,利用描述逻辑中概念包含性的基本原理,采用动态虚拟社区的思想,结合传统 P2P 网络所具有的分布性和开放性等特点,提出了一种本体驱动的基于 P2P 网络和层次网络的知识路由机制。

在本文中,知识路由是基于国家自然科学基金项目《知晓内容和环境的本体知识路由的研究》而提出的,是在语义 Web 背景下进行研究的。

所谓知识路由,指的是一种能够根据用户的相关语义信息发现、定位目标知识并将

目标知识从知识源穿过网络传递到用户的知识发现机制。从功能上讲,知识路由能够根据用户的请求内容、上下文和相关网络环境信息准确快速地定位满足用户需求的目标知识,即知识路由是自醒(self-aware)的。具体来讲,知识路由应该具有如下功能:

- (1) 知晓内容:该知识路由机制应当具有一个知晓内容中心,能够捕捉与用户需求相关的显示和隐式语义信息;
- (2) 知晓环境和上下文:该知识路由机制应当具有一种知晓用户上下文和网络环境信息的机制,能够智能获得用户的上下文信息、知晓当前网络环境;
- (3) 本体映射:该知识路由机制应当具有一种本体映射机制,能够解决异构本体库和知识库之间的交互和互理解问题;
- (4) 知识路由推理中心:该知识路由机制应当具有一个路由推理中心,能够根据相关的用户内容、上下文信息和网络环境信息,实现服务知识和相关领域知识的复合,并动态生成知识路由表,从而能够快速准确地发现、定位和访问知识;
- (5) 评估机制:该知识路由机制应当具有一个评估中心,能够对满足用户需求的目标知识进行评估排序,确定满足用户需求的最优及次优目标知识。

有关知识路由的具体内容将在第三章给予详细阐述。

2.4 本章小节

本章主要论及以下内容:

- (1) 介绍了语义 Web、语义 Web 服务、语义 Web 服务本体描述语言等相关概念;
- (2) 深入分析并详述了语义 Web-服务复合的关键技术:语义 Web 标记语言、Web 服务的语义描述模型和 Web 服务的复合方法;
- (3) 初步介绍了与本体知识路由相关的国内外研究现状及相关研究内容。

第三章 基于 P2P 网络的本体知识路由

本章首先介绍和分析 P2P 网络的相关知识, 然后以 P2P 网络为基础, 利用 P2P 网络的优点, 重点介绍基于 P2P 网络的本体知识路由的概念及其工作原理。

3.1 P2P 技术

3.1.1 P2P 的概念

Oram^[50]给 Peer-to-Peer (P2P) 一个简单的定义: P2P 是利用 Internet 边缘的存储、CPU 计算周期、内容及人力等资源的一组应用程序。因为访问这些分布式的资源意味着运行在一个连接不稳定、IP 地址不可预测的环境下, 所以 P2P 节点必须运行于 DNS 之外并且具有自治的中央服务器。简言之, P2P 是应用层一个特殊的分布式系统, 每对对等体之间可以利用路由协议进行相互通信。图 3-1 (b) 为一个一般的 P2P 网络。每个节点(对等体)维持一个对象数据库, 并且每个对等体可以通过 P2P 层的逻辑连接查询其它对等体上它想要的对象。现在, 对 P2P 概念进行了扩展, 如 IBM 公司认为: P2P 系统由若干互联协作的计算机构成, 且至少具有如下特征之一: 系统依存于边缘化(非中央式服务器)设备的主动协作, 每个成员直接从其它成员而不是从服务器的参与中受益; 系统中成员同时扮演服务器与客户端的角色; 系统应用的用户能够意识到彼此的存在, 构成一个虚拟或实际的群体^[51]。

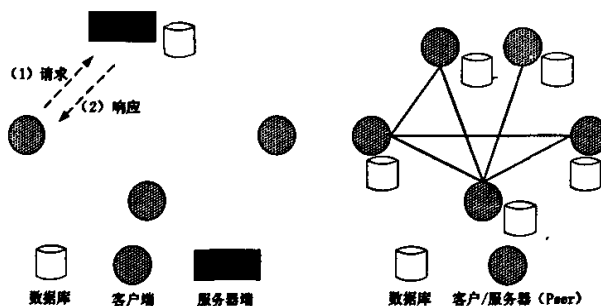


图 3-1 传统 C/S 模型和一般 P2P 模型

Fig. 3-1 The traditional C/S model and the general P2P model

为了更清楚的理解 P2P，我们将 P2P 模型与传统的 Client/Server 模型进行比较（表 3-1）。在 Client/Server 模型中，节点要么是客户端要么是服务器，资源（如存储或计算能力）可以在客户端和服务器之间共享。C/S 模型中的服务器实际上是个中央控制点。在 P2P 模型中，节点既是客户端又是服务器。作为客户端，节点可以从其它节点（对等体）查询或下载它想要的对象，同时它又作为服务器向其它节点提供对象。

表 3-1 C/S 模型与 P2P 模型比较

Table 3-1 The comparison between C/S model and P2P model

Model	Role of node	Sharing Relationship	Central Control
C/S	Client or Server	C $\leftrightarrow$ S	Yes (Server)
P2P	Servent	Peer $\leftrightarrow$ Peer	No

根据生命周期，P2P 节点一般有四个阶段：加入、查询、下载和离开。首先，一个刚刚到来的节点必须活跃的加入 P2P 系统中。在这个阶段，它可以获得一些基本的信息（如其邻居节点）来启动，同时发布它所持有的信息。其次，节点可以查询它想要的对象。P2P 定位协议将帮助该节点确定目的节点，而 P2P 路由协议则将查询消息路由到目的节点。再次，如果查询成功（通常返回目的节点的 IP 地址），节点可以直接从目的节点下载对象。最后，在理想的情况下，节点会宣布它离开系统。因此，P2P 系统有三个重要的组成部分：邻居发现、定位协议和路由协议。

3.1.2 P2P 的关键技术及特性

P2P (Peer to peer)对等网络技术是网络世界中最流行和最有前途的技术之一。P2P 技术让用户可以直接连接到其他用户的计算机，进行文件共享与交换，另外 P2P 技术在深度搜索、分布计算、协同工作等方面也大有用途。P2P 技术特别适合于今天企业日益增长的非集中化特性。

简单地说，P2P 就是一种用于不同 PC 用户之间直接交换数据或服务的技术，它允许 Internet 用户直接使用对方的文件。每个人可以直接连接到其它用户的计算机，并进行文件的交换，而不需要连接到服务器上再进行浏览与下载。因为消除了中间环节，P2P 技术使得网络上的沟通变得更容易、更直接。P2P 改变了 Internet 现在的以大网站为中心的状态、重返“非中心化”，并把权力交还给用户。从某种意义上讲，P2P 体现了 Internet 的本质。P2P 是一种基于互联网环境的新的应用型技术，主要为软件技术：

- (1) 对于互联网上众多计算机, P2P 应用比其它应用要更多考虑那些低端 PC 的互联, 它们不具备服务器那样强的联网能力, 同时对于以往的 P2P 应用技术, 现在的硬件环境已经更为复杂, 这样在通信基础方面, P2P 必须提供在现有硬件逻辑和底层通信协议上的端到端定位(寻址)和握手技术, 建立稳定的连接。涉及的技术有 IP 地址解析、NAT 路由及防火墙。
- (2) 在应用层面上, 如果两个 Peer 分别代表两家不同的公司, 而且它们已经通过互联网建立连接, 那么一方的信息、就必须为另一方所识别, 所以当前互联网上关于数据描述和交换的协议, 如 XML, SOAP, UDDI 等都是一个完善的 P2P 软件所要考虑的。
- (3) 有通信就要有安全保障, 加密技术是必须要考虑的。
- (4) 其它需考虑的有: 如何设置中心服务器, 如何控制网络规模等

P2P 技术特性:

- (1) 既是服务器端又是客户端, 如何表现取决于用户的要求, 网络应用由使用者自由驱动。
- (2) 信息在网络设备间直接流动, 高速及时, 降低中转服务成本。
- (3) 构成网络设备互动的基础与应用。在使网络信息分散化的同时, 相同特性的 P2P 设备可以构成存在于互联网这张大网中的子网, 使信息按新方式又一次集中。

3.1.3 P2P 模式的主要形式及系统结构

(1) 集中式 P2P:

集中式 P2P^[52]有一个中心服务器来负责记录共享信息、以及回答对这些信息的查询。每一个对等实体对它将要共享的信息以及进行的通信负责, 根据需要下载它所需要的其它对等实体上的信息。这种形式具有中心化的特点, 但是它不同于传统意义上的 Client/Server 模式。传统意义上的 Client/Server 模式采用的是一种垄断的手段, 所有资料都存放在服务器上, 客户机只能被动地从服务器上读取信息, 并且客户机之间不具有交互能力。而集中式 P2P 形式则是, 所有网上提供的资料都分别存放在提供该资料的客户机上, 服务器上只保留索引信息, 此外服务器与对等体以及对等实体之间都具有交互能力。其结构示意图见图 3-2, 其中 Bob 和 Carol 进行关于“X 文件”的操作。

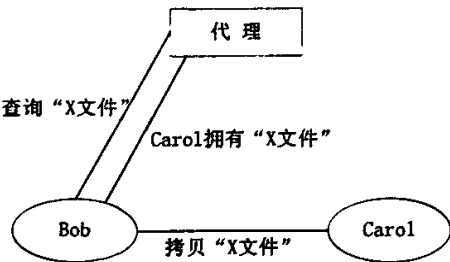


图 3-2 集中式 P2P

Fig. 3-2 The centralized P2P

(2) 分布式 P2P:

分布式 P2P 形式是一种纯 P2P 模式。这种形式不需要有中心服务器和中心路由器，其上的每一个 Peer 都作为对等实体，地位是完全平等的。每一个 Peer 既可以作为客户机又可以作为服务器，并且它们与相邻的 Peer 有相同的能力。具体地说，它有两种路由结构，一种是分布式的目录结构，另一种是直接消息传递。其结构示意图见图 3-3，其中，Bob 和 Carol 进行关于“X 文件”的操作。

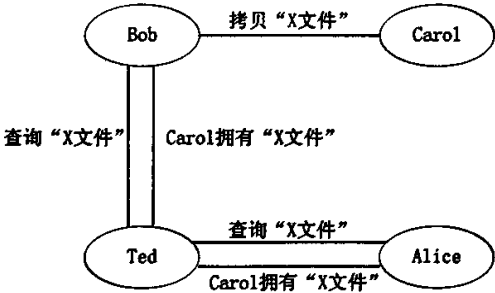


图 3-3 分布式 P2P

Fig. 3-3 The distributed P2P

(3) 混合式 P2P:

集中式 P2P 形式有利于网络资源的快速检索，以及只要服务器能力足够强大就可以无限扩展，但是其中心化的模式容易遭到直接的攻击；分布式 P2P 形式解决了抗攻击问题，但是又缺乏快速搜索和可扩展性。混合 P2P 形式结合了集中式和分布式 P2P 形式的优点，在设计思想和处理能力上都得到进一步优化。它在分布式模式基础上，将用户结点按能力进行分类，使某些结点担任特殊的任务。

3.2 本体知识路由系统

如 3.1 节所述，从“非中心化”意义上讲，P2P 改变了以大网站为中心的状态，增

加了用户和节点的自主性，在一定程度上体现了 Internet 的本质，反映了当前 Web 发展的一种重要趋势。而且混合式 P2P 又兼有集中式 P2P 和分布式 P2P 的优点，在设计思想和处理能力上都得到进一步优化。因此，本文基于混合式 P2P 分布性、自制性、灵活性等特点，结合语义 Web 的优点，针对 Internet 复杂多变的客观要求，建立了一种基于混合式 P2P 网络的本体知识路由机制。该路由机制具有如下特点：

- (1) 分布性：知识路由的关键是针对某一特定知识需求找到能够满足该需求的目标知识节点。采用集中式路由方法意味着存在一个知晓全局知识的知识服务器。然而，在文献^[53]中已经证明该方法的效率比较低，无法满足 Internet 的需求。与集中式路由相比，分布式路由不仅能够弥补集中式路由的缺点，而且更能适应 Internet 复杂多变的特殊要求。在分布式路由系统中，系统的各个构件在知识传输、复合方面都具有同等的地位。而混合式 P2P 网络技术便是实现这种分布式知识路由的最有效选择^[54]；
- (2) 开放性：有效的知识路由系统还必须具有开放性，即能够实现不同种类移动设备的互联；
- (3) 自治性和社会性：基于 P2P 的知识路由节点还必须具有自治性和社会性，即能够和用户以及其它知识路由节点进行知识通信，且每一个知识路由节点能够知晓与其自身知识相近的其它知识节点的知识；
- (4) 可扩展性：在 Internet 环境中，知识路由系统规模将会呈指数级增长^[55]。因此，有效的知识路由系统必须能够实时处理知识路由节点的动态增加、删除等行为；
- (5) 语义原则性：在知识路由系统中，知识路由节点必须能够准确理解和处理知识的本体描述，通过计算本体概念之间的相似度，利用知识节点之间的语义关系找到满足用户知识需求的目标知识节点。

3.2.1 知识路由概念

知识路由是一种能够根据用户的知识需求，快速有效地发现、定位目标知识并将目标知识从知识源穿过网络返回给用户的知识发现机制。其形式化定义如下：

定义 3.1 知识路由是一个六元组 $KR = \langle CnA, CxA, KD, KC, KP, KE \rangle$ ，其中各个分量意义如下：

- (1) **CnA (Content-Aware)**: 表示知晓内容。指机器能够准确理解用户的语义信息（显式及隐式语义），并利用这些语义信息为发现、定位和返回目标知识所服务。
- (2) **CxA (Context-Aware)**: 表示知晓上下文。指机器能够获取用户的当前上下文信息并以这些信息为依据使机器更好地为知晓内容所服务。
- (3) **KD (Knowledge Discovery)**: 表示知识发现。指机器利用 **CnA** 和 **CxA** 所得到的用户知识需求信息，建立用户需求和目标知识结点之间的映射，发现满足用户需求的目标知识。
- (4) **KC (Knowledge Composition)**: 表示知识复合。指当现有的知识无法满足用户需求时，通过将现有知识进行复合得到新知识，从而满足用户的需求（注：此处暂不考虑知识的粒度）。
- (5) **KP (Knowledge Positioning)**: 表示知识定位。指机器通过对当前网络状况和传输路径等信息进行分析，确定能够到达满足用户知识需求的目标知识节点的最佳路径。
- (6) **KE (Knowledge Evaluation)**: 表示知识评估。指机器通过对 **KD**、**KC** 和 **KP** 所确定的目标知识结点进行满意度评估，确定能够满足用户需求的最优及次优目标知识。

定义 3.2 (知晓内容) 知晓内容是一个五元组 $CnA = \langle S, R, K, A, CxA \rangle$ ，其中 $S = \{s_i, i=1,2,\dots,n\}$, ($n=|S|$)，表示知识库或服务本体库集合； $R = \{r_i, i=1,2,\dots,n\}$, ($n=|R|$)，表示用户需求知识的有限集； $K = \{k_i, i=1,2,\dots,n\}$, ($n=|K|$)，表示从用户需求中提取的有效知识有限集， $K \subseteq R$ ； $A \subseteq (S \times K) \cup (K \times R)$ ，表示连接知识库与用户有效知识、用户有效知识与用户需求知识的有限关系集； $CxA = \{CxA_i, i=1,2,\dots,n\}$, ($n=|CxA|$)，表示与知晓内容相关的知晓上下文机制。

定义 3.3 (知晓上下文) 知晓上下文是一个五元组 $CxA = \langle S, C, K, A, CnA \rangle$ ，其中 $S = \{s_i, i=1,2,\dots,n\}$, ($n=|S|$)，表示知识库集合； $C = \{c_i, i=1,2,\dots,n\}$, ($n=|C|$)，表示用户上下文知识的有限集； $K = \{k_i, i=1,2,\dots,n\}$, ($n=|K|$)，表示从用户上下文中提取的有效上下文知识有限集， $K \subseteq C$ ； $A \subseteq (S \times K) \cup (K \times C)$ ，表示连接知识库与用户有效上下文、用户

有效上下文与用户上下文的有限关系集： $CnA=\{CnA_i, i=1,2,...n\}$ ， $(n=|CnA|)$ ，表示与知晓上下文相关的知晓内容机制。

定义 3.4（知识发现） 知识发现是一个五元组 $KD=<CnA, CxA, DM, OK, B>$ ，其中 $CnA=\{CnA_i, i=1,2,...,n\}$ ， $(n=|CnA|)$ ，表示与用户需求有关的知晓内容机制；其中 $CxA=\{CxA_i, i=1,2,...,n\}$ ， $(n=|CxA|)$ ，表示与用户需求有关的知晓上下文机制； DM 表示能够定位满足用户需求的目标知识发现机制； $OK=\{OK_i, i=1, 2, ..., n\}$ ， $(n=|OK|)$ ，表示满足用户需求的目标知识； $B\subseteq(CnA\times CxA\times OK)$ ，表示连接知晓内容、知晓上下文和目标知识的有限关系集。满足关系式： $OK\subseteq DM(x, y), (x\in CnA, y\in CxA)$ 。

定义 3.5(知识复合) 知识复合是一个四元组 $KC=<AK, CK, CM, E>$ ，其中 $AK=\{AK_i, i=1,2,...,n\}$ ， $(n=|AK|)$ ，表示当前存在的用于复合的原子知识； $CK=\{CK_i, i=1,2,...n\}$ ， $(n=|CK|)$ ，表示复合之后所形成的新知识； $CM=\{CM_i, i=1,2,...n\}$ ， $(n=|CM|)$ ，表示原子知识的复合机制，包括顺序复合、并行复合和混合复合等复合形式^[56]； $E\subseteq(CK\times AK)$ ，表示连接新知识和原子知识的有限关系集。满足关系式： $CK\subseteq CM(x), (x\in AK)$ 。

定义 3.6（知识定位） 知识定位是一个五元组 $KP=<OK, SE, NOK, PM, D>$ ，其中 $OK=\{OK_i, i=1,2,...,n\}$ ， $(n=|OK|)$ ，表示经过知识发现阶段但未将网络状况和传输路径等信息考虑在内而返回的目标知识； $SE=\{SE_i, i=1,2,...n\}$ ， $(n=|SE|)$ ，表示网络状况、传输路径等网络信息； $NOK=\{NOK_i, i=1,2,...n\}$ ， $(n=|NOK|)$ ，表示将 SE 信息考虑在内而返回的满足用户需求的目标知识； PM 表示完成定位功能的定位机制； $D\subseteq(OK\times SE\times NOK)$ ，表示连接目标知识、网络信息和目标知识的有限关系集。满足关系式： $NOK\subseteq PM(x, y), (x\in OK, y\in SE)$ 。

定义 3.7（知识评估） 知识评估是一个四元组 $KE=<RK, NOK, EM, E>$ ，其中 $RK=\{RK_i, i=1,2,...,n\}$ ， $(n=|RK|)$ ，表示达到满意度要求返回给用户的目标知识； $NOK=\{NOK_i, i=1,2,...n\}$ ， $(n=|NOK|)$ ，表示经过知识路由 KD, KC, KP 阶段后得到的待

评估的知识； EM 表示知识评估机制； $E \subseteq (NOK \times RK)$ ，表示连接待评估知识和返回目标知识的有限关系集。满足关系式： $RK \subseteq EM(x), (x \subseteq NOK)$ 。

根据定义 3.1 可以看出，有效的知识路由机制能够根据用户的知识需求，快速准确地发现、定位和返回满足用户需求的最优和次优目标知识。在知识路由中，知识发现、知识复合、知识定位是关键内容和核心部分，知识发现及定位已在本项目论文^[57]中进行了深入研究，本文则在第四章将对知识复合进行深入探讨。

3.2.2 知识路由系统框架

本文将 P2P 技术和本体技术相结合，提出了一种本体驱动的基于混合 P2P 网络结构的知识路由机制。如定义 3.1 所示，与一般的知识发现机制相比，本体知识路由机制不仅能够知晓用户的内容和上下文信息，智能地对结果知识进行评估，而且还能够根据用户的知识需求，构建动态虚拟知识社区，实现基于动态虚拟知识社区的知识发现、复合和定位。本节主要集中讨论知识路由的知识发现、知识复合、知识定位以及知识路由算法。下面从知识路由体系结构、动态虚拟知识社区、知识路由主节点、知识路由从节点和知识路由算法五个方面介绍本体知识路由。

(1) 知识路由体系结构

在知识路由系统中，主要包括两类知识路由结点：知识路由主节点和知识路由从节点。

- 知识路由主节点：指代理各个动态虚拟知识社区的知识路由结点。知识路由主节点构成了知识路由系统的主体体系结构。其中每个知识路由主节点代表其相应动态虚拟知识社区的最高层次或最抽象层次的概念；知识路由主节点是知识路由系统中各个动态虚拟知识社区交互的对外接口。
- 知识路由从节点：指处于动态虚拟知识社区内部的知识路由结点。在动态虚拟知识社区中，知识路由从节点具有相同或相近的兴趣、知识或语义。其中每一个知识路由从节点代表该节点本地知识库中的最高层次或最抽象层次的概念。

针对上述两种知识路由结点，知识路由系统的体系结构也应该包括两部分：知识路由主节点体系结构和知识路由从节点体系结构。

- 知识路由主节点体系结构：指的是代表不同动态虚拟知识社区的知识路由主节

点之间的组织结构，反映的是动态虚拟知识社区的组织方式。从当前国内外研究现状来看，知识路由主节点之间的体系结构主要有两种构建方案：集中式体系结构和分布式体系结构。但是，如文献^[53]所述，由于集中式体系结构所具有的不足以及所建立的知识路由机制是面向 Internet，且必须具有分布性、开放性、自治性、社会性、可扩展性和基于语义的路由等特性。因此，在本知识路由系统中，知识路由主结点之间的体系结构采用基于 P2P 网络和本体的分布式体系结构。

- 知识路由从节点体系结构：主要指动态虚拟知识社区内部的具有相同或相近兴趣、知识或语义的知识路由从结点之间的组织结构。如文中 3.2 节所示，动态虚拟知识社区的构建是基于语义相似度的。因此，处于同一动态虚拟知识社区的知识结点之间的知识库都具有一定程度的语义相似性。而从描述逻辑的角度来看，语义相似性反应的是知识概念或知识库之间的包含程度。因此，基于这些知识路由从结点之间的包含程度就可以建立该动态虚拟知识社区的层次体系结构。故在本体知识路由中，知识路由从节点之间的体系结构采用基于包含度的层次体系结构。

知识路由系统的体系结构如图 3-4 所示。

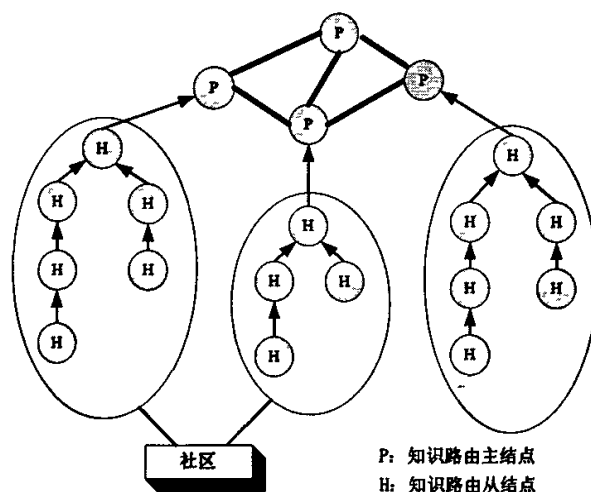


图 3-4 知识路由系统体系结构

Fig. 3-4 The framework of knowledge routing

(2) 动态虚拟知识社区

在本文中，动态虚拟知识社区定义如下：

定义 3.8(动态虚拟知识社区): 动态虚拟知识社区 DVC (Dynamic Virtual Community) 是一个四元组: $DVC = \langle KRN, CKRN, SF, \theta \rangle$ 。其中, KRN (Knowledge Routing Nodes) $= \{n_i, i=1, 2, \dots, m\}$, ($m=|KRN|$), 表示知识路由系统中所有知识路由节点集; $CKRN$ (Community Knowledge Routing Nodes) $= \{n_i, i=1, 2, \dots, m\}$, ($m=|CKRN|$), 表示经过语义相似度计算之后满足语义相似度阈值 θ 的构成动态虚拟知识社区的知识路由节点集; SF (Similarity Function) $= \langle SF: x, y \rightarrow z \rangle$ ($x, y \in KRN, z \in \{0, 1\}$), 表示语义相似度计算函数, 用于计算知识路由节点之间的语义相似度; $\theta \in \{0, 1\}$ 表示检验知识路由节点是否属于同一动态虚拟知识社区的语义相似度阈值。

DVC 中各个分量的关系如下所示:

设有两个知识路由节点 x, y :

if $(x, y) \in KRN$ and $SF(x, y) \geq \theta$, then $(x, y) \in CKRN_i$,

即节点 x, y 可归于同一动态虚拟知识社区 $CKRN_i$ 。

从定义 3.8 可以看出, 动态虚拟知识社区 DVC 是由具有相同或相近的兴趣、知识或语义的知识结点所组成的群体, 该群体是受用户需求驱动的。因此, 针对不同的用户需求所构建的动态虚拟知识社区也会有所区别。故该动态虚拟知识社区是动态产生和动态调整的。在本知识路由中, 主要是通过计算知识路由结点之间的语义相似度构建动态虚拟知识社区, 因此构建 DVC 的关键是语义相似度计算函数 SF 的确定。本文采用的是基于多角度的方法计算概念或知识之间的语义相似度, 见文献^[58]。构建动态虚拟知识社区的具体步骤如下:

- ① 对网络上的每个知识节点, 确定该知识节点本地知识库的概念层次关系, 找到其中最抽象的概念 (顶层概念), 并将该概念指定到一个 Agent H , 表示该知识节点, 即确定知识路由从节点 Agent;
- ② 计算知识路由从节点 Agent 与用户知识需求之间的语义相似度, 将满足语义相似度要求的知识路由从节点按照定义 2 划归到同一个动态虚拟知识社区 DVC , 其中每一个 DVC 实质上代表一个语义重叠区;
- ③ 根据第②步计算的知识路由从节点之间的语义相似度, 得出每一个动态虚拟知识社区 DVC 的包含层次结构, 并以此为依据确定相应 DVC 的层次化体系结构;
- ④ 根据第③步确定的知识路由从节点的层次体系结构, 为动态虚拟知识社区 DVC 中的最高层概念指定一个代理 Agent P , 并将此 Agent 作为代表

该 DVC 的知识路由主节点。

动态虚拟知识社区的构建方法见算法 1(Algorithm 1)。

Algorithm 1 DVC Construction: DVC_Construction (Q, P, C)

Require: Queue of Queries Q, Queue of peers P, Queue of peers in DVC C

```

1: Queue O=  $\Phi$            //O is the queue of all peers' local ontology bases.
2: Queue MGC=  $\Phi$          //MGC is the queue of every peer's most general concept.
2: Queue CommunityPeers= $\Phi$ 
3: for all p $\in$ P do
4:   O=getLocalNodeRepository(p)
5:   for all o $\in$ O do
6:     MGC=retrieveMostGeneralConcept(o)
7:   end for
8: end for
9: for all q $\in$ Q do
10:  for all mgc $\in$ MGC do
11:   if SF(q, mgc) $\geq$   $\theta$  then
12:    C=C.append(mgc)
13:   end if
14: end for
15: end for
16: return C

```

(3) 知识路由主节点

在这一部分，将从内部角度和外部角度讨论知识路由主节点的设计。其中内部角度主要考虑将每一个知识路由主节点看作单个独立的单元，重点讨论知识路由主节点的知识管理模块(Knowledge Management)。外部角度主要考虑基于 P2P 网络体系结构的知识路由主节点之间的交互通信，涉及 Agent 管理模块(Agent Management)、知识发现模块(Knowledge Discovery)、知识复合模块(Knowledge Composition)、语义查询模块(Semantic Query)和通信接口模块(Communication Interface)。知识路由主节点的内部结构如图 3-5 所示。

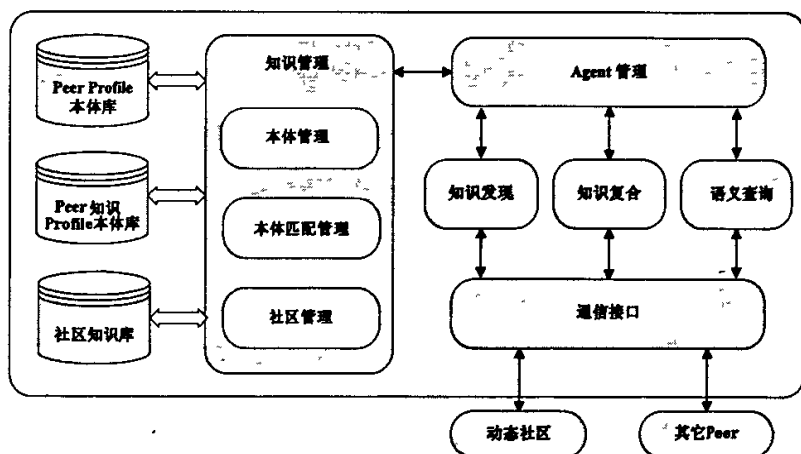


图 3-5 知识路由主节点内部结构图

Fig. 3-5 The internal structure of knowledge routing primary mode

① 节点本体库

节点本体库从语义角度描述每个知识路由节点的知识信息。节点本体库包括两方面的内容:

- **节点 profile 本体 (Peer Profile Ontology):** 对知识路由主节点的特征信息进行本体建模, 包括节点本体、邻接节点本体和协议本体; 其中, 在节点本体中, 必须设置如下属性:

Overall Confidence (PP. OC) : 描述该节点对网络上知识的知晓程度, 取值范围是 $[0, 1]$ 。1 表示该节点知晓网络上的所有知识; 反之, 0 则表示该节点对网络上的知识一无所知。

- **节点知识 profile 本体 (Peer Knowledge Base Profile Ontology):** 对知识路由主节点的本地知识以及相关的资源进行本体建模, 包括节点本地知识本体和资源本体。节点知识反映该知识路由节点所知晓的相关知识。其中, 在资源本体中必须设置如下两个属性:

hasPeer (PKBP. peer): 表示该节点知识本体隶属于哪一个知识节点, 建立与节点本体的关联关系;

Resource-specific Confidential (PKBP. RC): 表示对于某一特定的资源来说, 该知识节点对这一特定知识资源的知晓程度, 取值范围是 $[0, 1]$ 。

采用节点本体库的优点主要体现在如下几个方面:

- 通过节点本体库, 实现了所有知识路由节点知识结构的统一处理;
- 通过节点本体库, 有助于实现领域知识的重用;
- 通过节点本体库, 实现了领域知识与节点操作知识的分离;
- 通过节点本体库, 使领域知识的本体建模更加清晰化。

② 知识管理模块

如图 3-5 所示, 知识管理模块主要包括三个构件: 本体管理构件、本体匹配管理构件和社区知识管理构件。

- **本体管理构件:** 主要负责节点本体库的定义和维护, 建立知识路由节点与其邻接知识路由节点的联系;
- **本体匹配管理构件:** 负责知识路由节点本体知识之间的匹配比较; 在当前国内外研究中, 本体匹配的方法主要有实例模式匹配、元素结构匹配、语言约束匹

配等方法^[59]。在本知识路由系统中,已有的方法无法完全满足匹配需求。因此,本文采取的本体匹配方法将从结构和实例两个层次进行匹配:

- 社区知识管理构件:这是知识路由主节点和知识路由从节点内部结构的最主要区别,该管理构件主要完成知识路由主节点对相应动态虚拟知识社区知识的有效管理。

③ 知识发现模块

根据定义 3.4,知识发现是指找到能够满足需求的目标知识路由节点,即指知识路由节点的发现。在 P2P 网络中,节点发现的评估取决于最终所选择的网络节点是否能够满足用户的需求^[60]。在传统 P2P 网络中,节点的发现依赖于一个中央索引服务器,而在基于语义的 P2P 网络中,知识节点的发现则更多依赖于节点概念之间的语义相似程度。

本文的知识发现则是采用语义路由的机制,通过构建语义重叠区——动态虚拟知识社区,计算知识路由节点之间的语义相似度,实现知识发现。实际上,这里所构建的动态虚拟知识社区反映了知识路由节点之间的语义相似度。因此,在本知识路由系统中,任何知识请求总是首先被传送到语义上最相近的动态虚拟知识社区。知识发现过程如下:

如定义 3.2 和定义 3.3 所述,知晓内容和知晓上下文必须能够从已有的知识库 S 中定位满足用户需求的目标知识 K ,即要求存在一个二元关系 $r \in (S \times K)$ 。此处,该二元关系即为定位目标知识的知识发现机制。目标知识的发现和定位由知识路由节点中知识发现构件 KD (Knowledge Discovery)完成。KD 包括知识路由表 KRT (Knowledge Routing Table)、基于概念格^[61]的知识路由推理引擎 KRREC (Knowledge Routing Reasoning Engine based on Concept-lattice)和知识库 KB (Knowledge Base)。

- 知识路由表 KRT: KRT 的主要功能是根据用户需求模型 URM (User Requirement Model)寻找与用户需求相匹配的虚拟社区。KRT 结构如表 1。

表 3-2 知识路由表

Table 3-3 Knowledge routing table

社区 (Community)	知识类型 (Knowledgetype)
1	类型 1
2	类型 2
....	
n	类型 n

KRT 主要有两个字段：社区(Community)和知识类型(Knowledgetype)。社区字段的一个值表示一个对应的虚拟知识社区。知识类型字段代表知识类型，该字段与用户需求模型 URM 中的 T 分量相对应。通过将 URM 中的 T 分量和 KRT 中的服务类型字段相匹配可以得到满足用户需求的虚拟社区。关于 URM 将在本章 3.3 节给予详细介绍。

- 知识路由推理引擎 KRREC: KRREC 的主要功能是参照对用户上下文模型 UCM (User Context Model) 的满意度确定满足用户需求的服务列表。KRREC 根据接收到的精确匹配和近似匹配知识列表，执行推理，确定满足用户需求的最优和次优服务列表。关于 UCM 也将在本章 3.3 节给予详细介绍。
- 知识库 KB: 存储推理过程所需的知识。

④ 知识复合模块

根据定义 3.5，知识复合是指通过搜集已存在的语义相关的原子知识，并基于这些原子知识构造出新知识的过程。这些原子知识包括 RDF 或 OWL 描述的概念三元组以及采用 OWL-S 建模的语义 Web 服务。由于在知识路由系统中，分散的知识路由节点所具有的知识彼此之间具有或多或少的语义关联，因此，利用这些语义关联，将相应的知识节点进行复合，便可构造出原先不存在的新知识，进而满足用户的知识需求，从而解决知识复合之前所无法解决的问题。在知识复合方面，本文将重点介绍语义 Web 服务知识的复合，其具体内容将在论文第四章给予进一步阐述。

⑤ 语义查询模块

语义查询模块的主要功能是帮助知识路由节点在知识路由系统中查寻发现目标知识，这是知识路由节点与外部进行知识通信的接口部件，其实现语义查询的基础是动态虚拟知识社区。通过构建动态虚拟知识社区，可以有效减少语义搜索空间，提高查询的效率和准确率。例如，存在三个知识路由节点 A 、 B 和 C ，一个查询请求 q 。经过语义相似度计算，按照本章 3.2.2 (2) 节所列方法，将节点 A 和 B 构成一个动态虚拟知识社区 C_1 ，节点 C 构成另外一个动态虚拟知识社区 C_2 。当用户在虚拟知识社区 C_1 中提出知识请求 q 时，知识路由系统就会优先查找该社区中的所有知识节点 A 和 B ，只有当该社区无法满足用户的知识需求时，才要继续查找节点 C 所在的知识社区 C_2 。

一般而言，语义查询模块面向的语义查询对象包括知识路由节点和用户知识请求两类。当面向知识路由节点时，可采用计算语义相似度的方法查找定位知识；但是，当语

义查询模块直接面向知识请求用户时,则可以通过构建动态多维进化用户模型来查找满足用户需求的知识节点。

(4) 知识路由从节点

知识路由从节点主要指处于动态虚拟知识社区内部的知识路由结点,与知识路由主节点的最大不同是知识路由从节点面向的是动态虚拟知识社区内部的交互,而不处理动态虚拟知识社区之间的交互。因此,知识路由从节点的内部结构和知识路由主节点内部结构的最大区别是在知识管理模块中不用设立社区知识管理构件,其余的功能则和知识路由主节点完全相同。其内部结构图如图 3-6 所示。

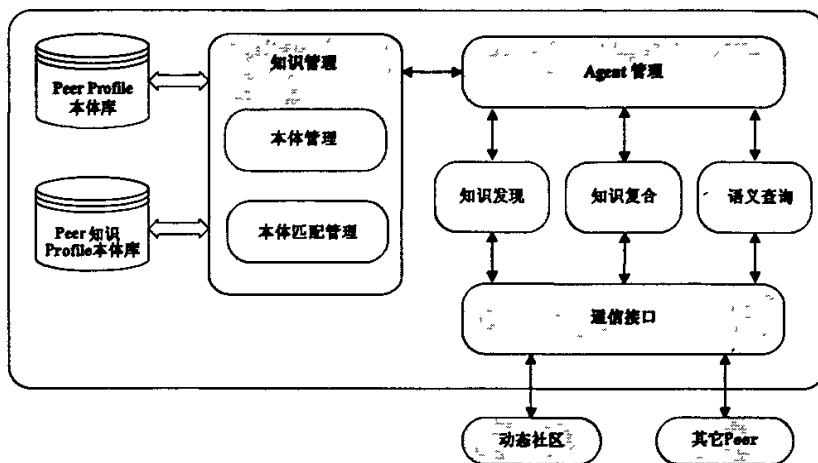


图 3-6 知识路由从节点内部结构图

Fig. 3-6 The internal structure of knowledge routing secondary node

(5) 知识路由算法

在本知识路由系统中,知识路由过程主要包括三步:提交请求、响应请求和接收响应。具体过程如下:

假设处于某一知识社区的节点用户通过某一知识路由从节点向知识路由系统发出一个知识请求 Q 。针对此请求,知识路由系统首先查询该从节点的本地知识库,如果存在满足用户需求的目标知识,则直接将该目标知识返回给用户;如果不存在这样的目标知识,则将该知识请求提交给该路由节点所在的动态虚拟知识社区,进行进一步的查找。

- 提交请求:当动态知识虚拟社区接收到用户的知识请求时,首先调用节点选择算法 2 (Algorithm 2),通过计算用户请求与知识路由从节点之间的概念包含关系,找到包含用户知识请求的最具体概念所在的知识路由从节点 (A_I),并根据

动态虚拟知识社区的层次体系结构返回最有可能满足用户知识请求的知识路由从节点集 $(A_1, A_2, \dots, A_n)$ 。 Q 与知识路由从节点 $A_1, A_2, A_3, \dots, A_n$ 的满足如下包含关系: $Q \subseteq A_1 \subseteq A_2 \subseteq A_3 \subseteq \dots \subseteq A_n$ 。 如果无法找到满足上述要求的知识路由从节点集 $(A_1, A_2, A_3, \dots, A_n)$, 则调用算法 3 (*Algorithm 3*) 对用户请求 Q 进行请求放大 (*Query Relaxation*), 然后查找满足用户要求的知识路由从节点集; 如果仍然没有找到满足要求的知识路由从节点集, 则继续调用算法 3, 直到找到满足要求的知识路由从节点集 $(A_1, A_2, A_3, \dots, A_n)$ 。 然后再将用户的知识请求分别发送到这些知识路由从节点进行匹配。 如果在动态知识虚拟社区中仍未找到满足用户请求的目标知识路由节点, 则继续将用户知识请求向上提交到代理本动态知识虚拟社区的知识路由主节点, 然后通过调用算法 1 和算法 3, 直至找到满足用户请求的知识路由节点, 并将用户的知识请求分别发送到这些知识路由节点进行匹配。 此时, 由于重复调用算法 1 和算法 3, 不断对用户请求进行放大, 因此, 总能找到满足用户请求的知识路由节点。

- **响应请求:** 当通过提交请求阶段找到的知识路由节点 (包括知识路由主节点和知识路由从节点) 接收到用户知识请求时, 该节点首先根据用户的知识请求信息, 创建一个关联该用户知识请求的个体, 并将此个体与该节点自身的节点个体库进行匹配。 在创建该个体过程中, 同时也创建了与该用户知识请求相关联的资源个体。 根据匹配结果, 将能够满足用户知识需求的目标知识返回给用户。
- **接收请求:** 经过提交请求和响应请求阶段, 当满足用户知识需求的目标知识返回给请求知识节点时, 请求知识节点通过调用算法 4 (*Algorithm 4*) 对所接收到的目标知识进行排序, 最终将能够满足用户知识需求的最佳知识返回给知识请求者。

下面将对知识路由过程中用到的知识路由算法——节点选择算法(*Algorithm 2*)、节点排序算法(*Algorithm 3*)、请求放大算法(*Algorithm 4*)和结果排序算法(*Algorithm 5*)进行详细讨论。

- **节点选择算法 (*Algorithm 2*):** 节点选择算法的主要功能是通过计算用户请求与知识路由从节点之间的概念包含关系, 返回最有可能满足用户知识请求的知识路由从节点集, 并通过整合节点本体的 *PP. OC* 和节点知识库本体的 *PKBP. RC*, 对知识路由从节点进行排序 (*Algorithm 3*)。 但是, 如果在一次查找中没有找到

满足要求的节点集，则需要重复调用请求放大算法 4 (Algorithm 4) 对用户请求进行放大，直至找到所需的知识路由从节点有序集。节点选择算法 (Algorithm 2) 和节点排序算法 (Algorithm 3) 如下所示。

Algorithm 2 Peer Selection: peerSelection (O, Q, P)

Require: LocalOntologyBase O, Queue of Queries Q,
Queue of peers P
1: $Q_{relaxed} = \Phi$
2: Queue selectedPeers = Φ
3: for all $q \in Q$ do
4: $S_q = \text{performQuery}(O, q)$
5: for all $S \in S_q$ do
6: $PKBP = \text{retrieveAllResources}(O, S)$
7: for all $pkbp \in PKBP$ do
8: selectedPeers.push (pkbp.peer, pkbp.RC)
9: end for
10: end for
11: $Q_{relaxed} = Q_{relaxed} + \text{relaxQuery}(O, q)$
12: end for
13: $P = P.append(\text{rankPeers}(\text{selectedPeers}))$
14: if $|P| = \Phi$ then
15: $P = \text{peerSelection}(O, Q_{relaxed}, P)$
16: end if
17: return A

Algorithm 3 Peer Rating: rankPeers (selectedPeers)

Require: Queue of pair's selectedPeers
1: Set $P = \{p | \exists pkbp.RC: (p, pkbp.RC) \in \text{selectedPeers}\} = \{p_1, p_2, p_3, \dots, p_n\}$
2: for all $p \in P$ do
3: $RC = \{RC(p, pkbp.RC) | \text{selectedPeers}\}$
4: for all $rc \in RC$ do
5: $strength(p) = tc * p.OC + (1 - tc) * \frac{1}{RC} \sum_{rc \in RC} rc$
6: end for
7: end for
8: Queue rankedPeers = $\{p_1, p_2, p_3, \dots, p_n\}$
 where $strength(p_1) \geq strength(p_2) \geq \dots \geq strength(p_n)$
9: Return rankedPeers

- 请求放大算法 (Algorithm 4): 如前所述，针对某一特定的用户知识请求，相应的知识路由从节点总是存在一次找不到满足要求的知识路由节点集的可能性。因此，必须对该知识请求按照某些规则进行基于语义的放大，以便最终能够找到满足要求的知识路由节点集。具体放大规则如表 3-3 所示。在表 3-3 中， s 表示 *subject*, o 表示 *object*, $super(s)$ 表示 s 的父类, $super(p)$ 表示 p 的父属性, $ROOT$ 表示根概念或根属性。请求放大算法如 Algorithm 4 所示。

Algorithm 4 Query Relaxation: relaxQuery(O, Q)

Require: O, Q
1: state = determineState(Q)
2: $O = \text{newQuery}(O, O, \text{state})$
3: return O

表 3-3 请求放大规则

Table 3-3 Query relaxation rules

状态	查询请求	放大后的查询请求
1	(s, p, o)	(s, *, o)
2	(s, p, *)	(s, *, *)
3	(*, p, o)	(*, *, o)
4	(s, *, o)	(s, *, *)
5	(*, *, o)	(o, *, *)
6	(*, p, *)	(*, super (p), *)
7a	(s, *, *)	(super (s), *, *)
7b	(s, *, *)	(class (s), *, *)
8	(*, *, *) $\vee$ (ROOT, *, *) $\vee$ (*, ROOT, *)	

- **结果排序算法 (Algorithm 5):** 该算法的主要功能是对目标知识节点的结果进行排序。对于该算法, 本文采用基于概念格和本体技术的排序算法。过程如下: 设经知识发现阶段找到三个满足用户需求的目标知识节点: Obj1, Obj2 和 Obj3。三者具有的属性集分别为 $\text{Property}(\text{Obj1}) = \{\text{Attr1}\}$, $\text{Property}(\text{Obj2}) = \{\text{Attr2}, \text{Attr3}\}$, $\text{Property}(\text{Obj3}) = \{\text{Attr1}, \text{Attr2}\}$ 。采用概念格^[61], 分别构造三个服务对象的十字表(图 3-7)和线状图(图 3-8)。

A	B	C	D
	Attr 1	Attr 2	Attr 3
Obj 1	X		
Obj 2		X	X
Obj 3	X	X	

图 3-7 十字表

Fig. 3-7 The 'X' table

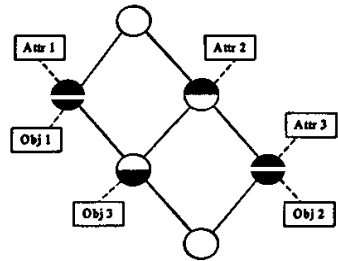


图 3-8 线状图

Fig. 3-8 The line diagram

对图 1 和图 2 的进行分析, 可得如下结论:

$$\left. \begin{array}{l} \text{Property (top)} < \text{Property (Obj1)} < \text{Property (Obj3)} < \text{Property (bottom)} \\ \text{Property (top)} < \text{Property (Obj2)} < \text{Property (bottom)} \end{array} \right\} \Rightarrow$$

$$\text{Property (top)} < \text{Property (Obj1)} < \text{Property (Obj2)} = \text{Property (Obj3)} < \text{Property (bottom)}$$

显然, 具有更多的属性意味着在某种程度上具有更大的满意度。因此, 对象 Obj1、Obj2、Obj3 的满意度符合如下排序:

$$\text{Satisfiability (Obj2)} = \text{Satisfiability (Obj3)} > \text{Satisfiability (Obj1)}$$

3.2.3 知识路由总结

如前所述, 本文通过将 P2P 技术和本体技术相结合, 利用描述逻辑概念包含性的基本原理, 采用动态虚拟社区的思想, 结合传统 P2P 网络结构所具有的分布性和开放性等特点, 提出了一种本体驱动的基于 P2P 网络结构和层次网络结构的知识路由机制。下面分别从合理性和先进性两方面, 通过与其他研究对比, 对本知识路由机制进行总结。

(1) 本体知识路由的合理性

- 知识表示的合理性: 知识路由是面向知识的路由机制, 本体作为语义 Web 的知识表示基础^[3], 以其在表达资源语义属性方面所具有的优势, 已经得到了 W3C 组织和广大研究人员的认可。因此, 采用本体作为知识路由的知识表示机制是合理可行的;
- 网络结构选择的合理性: 如 3.1 节所言, 从“非中心化”意义上讲, P2P 网络结构增加了用户和节点的自主性, 在一定程度上体现了 Internet 的本质, 反映了当前 Web 发展的一种重要趋势。因此, 采用基于混合式 P2P 网络结构的知识路由机制, 能够较为容易地实现与 Internet 的衔接, 适应当前 Web 的发展方向; 而且混合式 P2P 又兼有集中式 P2P 和分布式 P2P 的优点, 在设计思想和处理能力上都能得到进一步优化, 克服了集中式 P2P^{[46][52]}所具有的不足, 提高了该框架的适应性;
- 逻辑基础的合理性: 本文通过利用描述逻辑概念包含性和层次性原理, 将其与混合型 P2P 网络结构相结合, 构建了以描述逻辑和 P2P 网络结构为基础的知识路由机制, 实现了逻辑基础和网络结构的合理结合。因此, 从描述逻辑角度来看, 本知识路由框架具有一定的合理性。
- 知识路由算法的合理性: 如 3.2.2 节所述, 在本知识路由中, 共设计了节点选择算法(Algorithm 2)、节点排序算法(Algorithm 3)、请求放大算法(Algorithm 4)和结果排序算法(Algorithm 5)四个子算法。通过这四个算法的协调工作, 保证了在正常情况下始终能够发现满足用户需求的最佳目标知识, 弥补了与用户需求零匹配的不足^{[42][56]}。从这一角度讲, 本知识路由机制具有一定的实用性。
(注: 此最佳目标知识是相对最佳, 而非绝对最佳。)

(2) 本体知识路由的先进性

- 由于本体知识路由基于 P2P 网络, 因此该知识路由继承了 P2P 网络所具有的

分布性、开放性、自治性、社会性和可扩展性等优点^[50];

- 采用基于语义匹配的知识发现定位机制, 弥补了传统 Web 资源发现方法缺乏语义的缺点^{[43][44][46][47]}, 有助于提高定位发现目标知识的准确率和查全率;
- 采用多维用户模型, 有利于提高用户模型的重用性, 增加用户模型生成的智能性, 节省存储空间, 加快模型生成速率 (详见 3.3 节);
- 采用动态虚拟社区思想, 根据用户具体需求构建动态知识社区, 从而能够动态定位满足用户需求的目标知识, 克服了文献^{[48][49]}中缺乏动态满足用户需求的缺点;
- 通过节点选择算法(Algorithm 2)、节点排序算法(Algorithm 3)、请求放大算法(Algorithm 4)和结果排序算法(Algorithm 5)能够定位发现满足用户需求的目标知识, 与文献^{[42][56]}相比, 提高了知识发现的查全率和准确率。

3.3 用户模型构建

在本地知识路由系统中, 用户需求必须经过预处理, 建立与用户需求相对应的需求模型。用户模型的建立, 在现有文献^[62]中都是建立单个用户模型的模型库或用户偏好集。而本文则提出一种多维用户模型, 描述同一类用户的公共属性。采用这种用户模型不仅可以提高用户模型的重用性, 增加用户模型生成的智能性, 节省存储空间, 而且还可以加快模型生成速率, 更好反映用户的真实需求。文中构造两种用户模型, 用户需求模型 URM (User Requirement Model) 和用户上下文模型 UCM (User Context Model)。

- (1) 用户需求模型 URM: 根据定义 3.2, 知晓内容必须从用户需求知识的有限集 R 中提取有效知识集 K , 即存在二元关系 $r \in (R \times K)$ 。此处, $r = RM \in (K \times R)$, RM 为多维用户模型 $RM = \langle D, V, W, DI, DO, T \rangle$ 。其中, $D = \langle D_0, D_1, \dots, D_n \rangle$ 是描述用户的一个多维集合, D_i 表示从第 i 维描述用户, 它表示用户行为的一个方面, 例如在描述汽车时其中的一个 D_i 可以表示发动机, D_j 描述车型。 $V = \langle V_1, V_2, \dots, V_n \rangle$, V_i 表示 D_i 的值域, $V_i = \{v \mid v = E_i(D_i)\}$, 其中, $E_i(D_i)$ 表示在 D_i 维允许的取值。 $W = \langle W_1, W_2, \dots, W_n \rangle$, 表示 G_i 所对应的权值, 该权值反映从该维角度描述用户的重要性, $W_i = F(x_i, y_i)$ $i=1, \dots, n$ 。其中, $F(x, y)$ 为一函数, 用于确定 D_i 所对应的权值, $x_i \in \{D_i \mid i=1, 2, \dots, n\}$, 通过函数 F 返回 D_i 的权值 Y_i 。 DI 和 DO 分别代表用

户需求的输入和输出, T 表示用户需求或用户请求服务的类型。

- (2) 用户上下文模型 UCM: 根据定义 3.3, 知晓上下文必须从用户上下文知识有限集 C 中提取有效上下文知识集 K , 即存在二元关系 $r \in (C \times K)$ 。此处, $r = CM \in (K \times C)$, CM 为多维用户上下文模型 $CM = \langle C, V, W \rangle$ 。其中, $C = \langle C_0, C_1, \dots, C_n \rangle$ 是描述用户上下文的一个多维集合, C_i 表示从第 i 维描述用户上下文, 它表示用户上下文的一个方面, 例如描述在超市购物时, 其中一个 D_i 表示用户的位置, D_j 描述天气。关于 V 和 W 的定义见用户需求模型 URM。

此外, 对于每一个用户, 还需建立一个基于用户需求模型 M 的需求概貌 (Profile)。如下:

$$P = \{E_1(D_1), E_2(D_2), \dots, E_n(D_n)\} \text{ 其中 } E_i(D_i) \subseteq E(D_i)$$

在与用户交互的过程中, 对用户的每一步会话过程进行监测, 检测用户 profile, 实现基于 profile 的进化。

profile 的进化功能如下: 给定一个用户 U_1 , 把用户与系统的每一次交互当作一次会话, 定义在会话 k 中, 用户 U_1 的行为 $H_{1,k}$:

$$H_{1,k} = (h_1, h_2, \dots, h_n)$$

$$h_1 \in E_1(D_1), h_2 \in E_2(D_2), \dots, h_n \in E_n(D_n)$$

经过 q 步的会话, 用户 U_1 的行为是:

$$H_{1,k+q} = \{h_{1+q}, h_{2+q}, \dots, h_{n+q}\}$$

在这里引入变量 $\Delta_{1,k \rightarrow k+q}$ 表明用户 U_1 的行为集在 q 个会话后发生的变化。

$$\Delta_{1,k \rightarrow k+q} = H_{1,k+q} - H_{1,k}$$

如果 $\Delta_{1,k \rightarrow k+q} = \phi$, 那么用户的 profile 没有改变, 可以继续采用此 profile; 如果 $\Delta_{1,k \rightarrow k+q} \neq \phi$ 且 $\Delta + P \subseteq P$, 表明用户 profile 发生变化, 但 P 仍然能反映用户的初始行为, profile 保持不变; 否则用户的 profile 改变, 生成新的用户需求 profile: P' ; 同时, 把这个模型加入到用户 profile 库中, 如果同一类用户再次请求服务时, 就可以直接采用模型 P' , 无需重新建立用户的 profile。使得用户的 profile 有了重用性, 同时减少了存储空间。

3.4 本章小结

在本章中，我们首先深入分析了 P2P 网络的相关概念、关键技术和主要形式，并且论证了采用以 P2P 网络为基础的本体知识路由系统的优越性，然后重点介绍了与本体知识路由相关的概念和体系结构及工作原理。这些工作为第四章阐述本体知识路由中基于描述逻辑的语义 Web 服务动态复合奠定了基础。

第四章 知识路由中基于描述逻辑的语义 Web 服务动态复合

本章以第三章本体知识路由系统为基础,针对本体知识路由中服务知识复合的功能,提出了一种基于描述逻辑的语义 Web 服务动态复合机制,并从其逻辑基础——描述逻辑角度对其合理性和复杂度进行了理论证明。

4.1 描述逻辑

4.1.1 描述逻辑概述

描述逻辑(Description Logic)是基于对象的知识表示的形式化,它吸取了 KL-ONE 的主要思想,是一阶谓词逻辑的一个可判定子集。它与一阶谓词逻辑不同的是,描述逻辑系统能提供可判定的推理服务。除了知识表示以外,描述逻辑还用在其它许多领域。描述逻辑的重要特征是很强的表达能力和可判定性,它能保证推理算法总能停止,并返回正确的结果。在众多知识表示的形式化方法中,描述逻辑在十多年来受到人们的特别关注,主要原因在于:它们有清晰的模型——理论机制;很适合于通过概念分类学来表示应用领域;并提供了很有用的推理服务。

描述逻辑最开始只是用来表示静态知识的。为了考虑在时间上的变化,或者在一定动作下的变化,Wolter 等对具有模态算子的描述逻辑进行了深入系统的调查分析,并证明在恒定的领域假设下多种认知和时序描述逻辑是可判定的。他将描述逻辑和命题动态逻辑 PDL 相结合,提出了动态描述逻辑。

同样,为了对动作和规划能在统一的框架下进行表示和推理,A. Artale 和 E. Franconi (1998)提出了一个知识表示系统,用时间约束的方法将状态、动作和规划的表示统一起来。为了能使该表示方法进行有效的推理和具有明确的语义,它又和描述逻辑结合起来,从而形成了一个很好的知识表示方法。它具有以下优点:(i)能用统一的方法表示状态、动作和规划,这一点与情景演算不同;(ii)能进行高效的推理,该框架下的可满足性问题和包含检测问题等都是多项式时间;(iii)有明确的语义;(iv)能自动进行规划识别。

可满足性问题是描述逻辑推理中的核心问题,因为其它许多问题(如包含检测、一

致性问题等)都可化为可满足性问题。为了能用计算机自动判断描述逻辑中可满足性问题, Schmidt - Schaub 和 Smolka 首先建立了基于描述逻辑 ALC 的 Tableaux 算法, 该算法能在多项式时间内判断描述逻辑 ALC 概念的可满足性问题。目前, Tableaux 算法已用于各种描述逻辑中(如 ALCN, ALCQ 等), 并且 Tableaux 算法也可用于判断实例检测等问题。现在主要研究各种描述逻辑中 Tableaux 算法的扩展、复杂性及优化策略等。

本文正是在研究描述逻辑子语言 ALCQIO 的基础上, 将描述逻辑与 AI 规划相结合, 以知识路由为背景, 通过利用描述逻辑的强大推理功能, 提出一种基于描述逻辑的语义 Web 服务动态服务机制。下文将首先介绍描述逻辑子语言 ALCQIO。

4.1.2 描述逻辑子语言——ALCQIO

如前所示, 在知识表示中, 描述逻辑^[6]是基于对象的知识表示的形式化工具, 它吸取了 KL-ONE^[63]的主要思想, 是一阶谓词逻辑的可判定子集, 能够提供可判定的推理服务, 并且具有语义特征。描述逻辑的最基本语言是 *AL* (*Attributive Language*), 一般用大写字母如 “*A*” 或 “*B*” 表示概念描述。描述逻辑基本句法由原子概念(一元谓词)、原子角色(二元谓词)和个体(常量)组成。*ALC* 是一种基于 *AL* 的比较简单描述逻辑, 只包含简单的交、存在、任意等构造算子。在 *ALC* 的基础上通过添加相应的构造算子可以构成其它类型的描述逻辑子语言。这些构造算子包括 *Q*、*I*、*O* 等。其中, *Q* 代表数字约束 (*number restriction*) 算子, *I* (*inverse roles*) 代表逆关系构造算子, *O* (*nominal*) 代表名字算子。例如, 在 *ALC* 基础上分别添加 *O*、*Q*、*I* 算子可以构成 *ALCO*、*ALCQ* 和 *ALCI* 子语言。其它类似的子语言还有 *ALCQO*、*ALCIO*、*ALCQI*、*ALCQIO* 等。其中, *ALCQIO* 是 OWL-DL 的核心逻辑基础, 它是在 *ALC* 的基础上同时添加 *Q*、*I* 和 *O* 三类算子而成。因此, 本文在证明服务复合时, 主要采用 *ALCQIO* 逻辑。

在描述逻辑中, 概念通过一系列构造算子加以定义。描述逻辑中的构造算子决定着相应的描述逻辑的表达能力。表 1 给出了 *ALCQIO* 的基本构造子, 通过这些基本构造子可以定义更加复杂的构造子。设 N_c 表示 *ALCQIO* 所定义的概念集; N_r 表示 *ALCQIO* 所定义的二元关系集; N_i 表示 *ALCQIO* 声明的个体集。

表 4-1 *ALCQIO* 的语法和语义

 Table 4-1 *ALCQIO* syntax and semantic

名字	语法	语义
Inverse role	s^-	$\{(y, x) (x, y) \in s^I\}$
conjunction	$C \cap D$	$C^I \cap D^I$
negation	$\neg C$	$\Delta^I \setminus C^I$
At-least number restriction	$\geq n \text{ r } C$	$\{x \text{card}(\{y (x, y) \in r^I \wedge y \in C^I\}) \geq n\}$
nominal	$\{a\}$	$\{a^I\}$

在表 1 中, 第一行列出的是 *ALCQIO* 的二元关系构造子。*ALCQIO* 的二元关系可以是某一关系 $s \in N_r$, 也可以是其逆关系 s^- 。*ALCQIO* 的概念则由表 1 中的其它构造子构造而成, 其中 r 表示关系, n 为一正整数, a 为某一个体。利用这些构造子, 可定义如下构造子:

$$C \cup D = \neg(\neg C \cap \neg D) \text{ (概念并)}$$

$$\top = A \cup \neg A \text{ (全集)}$$

$$\exists r.C \Rightarrow \geq 1 \text{ r } C \text{ (存在量词约束)}$$

$$\forall r.C = \neg \exists r. \neg C \text{ (值约束)}$$

$$\leq n \text{ r } C = \neg(\geq (n+1) \text{ r } C) \text{ (at-most 约束)}$$

ALCQIO 的概念和关系是通过特定领域的解释 I 赋予其语义, $I = (\Delta^I, \cdot^I)$ 。解释 I 的领域是由个体和解释函数 $\cdot^I$ 构成的非空集。解释函数的映射功能如下:

- 对于每一个概念 $A \in N_c$, 映射成为 Δ^I 的子集 A^I ;
- 对于每一个关系 $s \in N_r$, 映射成为基于 Δ^I 的二元关系 $s^I \in (\Delta^I \times \Delta^I)$;
- 对于每一个个体 $a \in N_i$, 映射成为一个元素 $a^I \in \Delta^I$ 。

此外, 解释函数 $\cdot^I$ 可以扩展到任意元概念和关系, 如表 1 第三行所示。

TBox 是一个描述领域结构的概念定义的有限公理集。概念的定义公式为: $A \equiv C$ 。此处, A 是被定义概念名, 具有唯一性; C 是已定义的 *ALCQIO* 概念。公式左边的概念称为被定义概念, 右边的概念称为原始概念 (*primitive*)。如果 TBox 定义的概念中不存在循环依赖定义, 则称为该 TBox 是非循环的。如果将循环定义引入到 TBox 中, 则称

为该 TBox 是循环的。出于证明方便, 本文只关注非循环的 TBox。

TBox 的语义定义为: 当且仅当解释 I 满足 TBox T 中的所有概念时, 则称解释 I 是 TBox T 的一个模型 (model)。此时, 对于 T 中所有的 $A \equiv C$, 则有 $A^I = C^I$ 成立。在非循环 TBox 中, 对于概念和关系的解释可以扩展成为 TBox 的一个模型^[64]。

ABox 是一个描述关于具体个体事实的公理集, 包含的是外延性的知识 (推断知识)。是关于概念和关系的断言集。断言的声明公式为: $C(a)$, $s(a, b)$ 或者 $\neg s(a, b)$ 。其中, C 是某一概念, s 是某一关系名。此外, $C(a)$ 还可以记为 $a: C$ 。

ABox 的语义定义为: 当且仅当解释 I 满足 ABox A 中的所有概念时, 则称解释 I 是 ABox A 的一个模型 (model)。此时, 对于 A 中所有断言 $C(a)$, $s(a, b)$ 或者 $\neg s(a, b)$, 则有 $a' \in C^I$, $(a', b') \in s^I$ 或者 $(a', b') \notin s^I$ 。如果 μ 是 A 的一个断言, 解释 I 满足 μ , 则有 $I \models \mu$ 。

描述逻辑的推理功能主要包括可满足性推理、包含性推理、一致性推理和实例检验性推理。本文所用到的推理功能主要是可满足性和一致性推理。如下:

- TBox T 中的概念 C 是可满足的, 当且仅当存在一个 T 的解释模型 I , 满足 $C^I \neq \emptyset$;
- TBox T 和 ABox A 是相一致的, 当且仅当存在一个既是 TBox T 又是 ABox A 的解释模型 I 。

4.2 基于描述逻辑的语义 Web 服务复合机制——DLAIP

在本地知识路由中, 我们仍采用 OWL (Web Ontology Language)^[65] 和 OWL-S (Web Ontology Language for Service)^[66] 作为知识标记和建模语言, 仍以描述逻辑 DL 作为本体知识路由的逻辑基础, 其推理工作仍集中在包含性和可满足性检验。

如前所述, 采用描述逻辑可以实现对特定领域的语义建模。但是, 在采用 OWL 和 OWL-S 对领域以及领域相关的服务等信息进行语义建模时, 由于其逻辑基础是描述逻辑, 因此, 反映的多是与领域或服务相关的静态语义信息, 而对于描述服务复合等动态过程中的语义信息却存在着不足。而且, 由于 OWL 和 OWL-S 在表达一般规则方面有局限性^[67], 它不能很好地表示用户的偏好和约束。而这部分语义信息对于服务的自动复合是十分重要的。因为有了用户偏好和约束的标记, Agent 才能在动态的 Internet 环境中, 无需用户干预, 自动地选择合适的服务来执行。

在人工智能领域, AI 规划^{[68][69]}方法通过对行为 (action) 的描述, 采用基于行为的推理, 从而实现对某一行为执行前后所引起的相关世界变化的建模。因此, 采用 AI 规划方法, 将 Web 服务映射为相应的行为, 然后采用基于行为的推理能够实现 Web 服务的动态复合。如文献^[70]提出一种基于情景演算的服务复合语言 *Golog*, 文献^[71]则提出一种基于规则的服务复合机制 *SWORD*。但是, 在文献^{[7][8]}中, 采用 *Golog* 和 *SWORD*, 尽管能够动态实现服务的复合, 由于缺乏对语义的利用, 从而降低了服务复合的准确率。此外, 文献^[72]还将 OWL-S 与 HTN (*Hierarchical Task Network*) 相结合, 提出一种将 OWL-S 的 *Process Models* 映射为 HTN 规划系统 *SHOP2* (*Simple Hierarchical Ordered Planner*)^[73]的映射机制, 然后通过 *SHOP2* 的规划实现语义 Web 服务的复合。该机制侧重于建立一种映射关系, 没有考虑从相应的逻辑角度给予证明。

本课题正是针对上述不足, 在扩展 OWL-S 的基础上, 充分考虑描述逻辑和 AI 规划的优缺点, 通过将描述逻辑所具有的良好知识表示能力和强大推理能力与 AI 规划所具有的行为变化建模能力相结合, 提出了一种面向本体知识路由的基于扩展 OWL-S 模型和描述逻辑的语义 Web 服务复合机制——DLAIP (*AI Planning based on DL*)。并且从描述逻辑角度对其合理性进行了证明。

4.2.1 扩展的 OWL-S 模型-XOWL-S

由于用户的偏好与约束对于服务复合十分重要, 在某种程度上是判定服务复合是否成功的关键, 但原有的 OWL-S 模型并未考虑用户的需求偏好信息。因此, 本节在 W3C 组织推荐的 OWL-S 模型基础上, 引入用户的偏好约束, 形成扩展后的 XOWL-S (*eXtended OWL-S*) 模型。该模型的顶层结构图如图 4-1 所示。

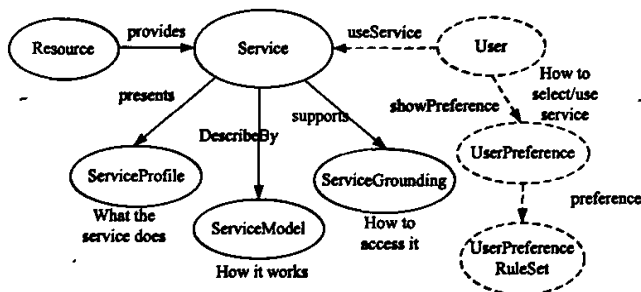


图 4-1 扩展后的 XOWL-S 模型

Fig. 4-1 The extended OWL-S model: XOWL-S

其中, 本体是模型的具体存在形式。扩展部分解释如下:

- 在 XOWL-S 的顶层本体中增加了 User、UserPreference、UserPreferenceRuleSet 三个类，分别表示用户、用户偏好和偏好规则集这三个概念。其中，UserPreference 类用于描述用户如何选择和使用服务，UserPreferenceRuleSet 类则用于描述用户的具体偏好规则。
- 定义了新增类相互之间的关系属性。User 类有两个属性：useService 和 showPreference，分别表示 User 类和 OWL-S Service 类的“使用”关系、User 类和 UserPreference 类的“表现出”关系。UserPreference 类有一个属性 has-preference，其值是 UserPreferenceRuleSet 类的实例。
- 定义了属性的约束。其中 useService、showPreference 和 preference 都是一对多的关系，因此它们的势值约束都大于 0。

4.2.2 DLAIP 相关概念

如上所述,通过对扩展 XOWL-S 模型描述语义 Web 服务的特点和 AI 规划建模行为的特征进行分析。可以看出:在 AI 规划中,通过考虑行为执行的先决条件(*precondition*)和执行结果(*effects*),能够根据需求动态实现行为之间的组合,从而确定符合需求的规划方案,但 AI 规划在对行为相关的领域建模方面却存在着不足;而对于 OWL-S,由于基于描述逻辑,因此具有良好的领域建模能力。但是,在 OWL-S 中,尽管也引入了先决条件和执行结果,但反映的是与服务相关的静态信息,却无法描述服务复合等动态过程中的语义,而且未考虑用户的个人偏好与约束信息。因此,将扩展 XOWL-S 模型和 AI 规划相结合,利用二者的优点,可以实现对服务复合动态过程的语义建模以及推理。下面,首先给出服务复合中原子服务与复合服务的形式化定义。

定义 4.1 (原子服务) 设 T 表示 $ALCQIO$ 知识库 $KB = \{TBox, ABox\}$ 的非循环 TBox 构件。 N_c 表示知识库定义的概念集; N_r 表示知识库定义的二元关系集; N_i 表示知识库声明的实例集。基于 T 的原子服务定义为

$$S = (pre, sname, inputs, outputs, post, pref).$$

其中各分量解释如下:

- **pre**: 基于 ABox 的有限断言集,描述服务的先决条件(pre-conditions),包括动态和静态信息;
- **sname**: 服务的名称标识;

- **inputs**: 与服务输入相关联的领域类, 包括 $A(a)$ 或者 $r(a, b)$, 其中 A 是 T 中定义的原子概念, $A \in N_c$, r 是二元关系名, 且 $a, b \in N_i$;
- **outputs**: 与服务输出相关联的领域类, 其定义同 **inputs** 分量;
- **post**: 表示服务执行所产生的结果, 包括动态和静态信息。采用公式 P/E 表示, 其中 P 是 ABox 的声明, E 是 TBox 中定义的原始概念 (primitive), 如 $A(a)$ 、 $\neg A(a)$ 、 $r(a, b)$ 或 $\neg r(a, b)$ 。公式 P/E 表示当 P 为真时, 执行服务结果 E 为真。
- **pref**: 基于 TBox 与 ABox 的有限断言集, 描述用户的需求偏好。采用公式 $A(a)$ 或者 $r(a, b)$ 表示, 其中 A 是 T 中定义的原子概念, $A \in N_c$, r 是二元关系名, 且 $a, b \in N_i$ 。

定义 4.2 (复合服务) 设 T 表示 $ALCQIO$ 知识库 $KB = \{TBox, ABox\}$ 的非循环 TBox 构件。 N_c 表示知识库定义的概念集; N_r 表示知识库定义的二元关系集; N_i 表示知识库声明的实例集。基于知识库 KB 的复合服务 S 指的是由一系列原子服务 $S_1, S_2, \dots, S_k$ 组成的有限有序服务流, 记作

$$S = \{SL, CF\}.$$

其中, SL (Services List) 表示构造复合服务的服务列表, $SL = \{S_1, S_2, \dots, S_k\}$, CF (Control Flow) 表示服务列表 SL 中的服务的执行顺序, 即复合服务的控制流, 可以是顺序、并行等服务复合模式^[56]。 SL 和 CF 的关系为:

$$SL = CF \{S_1, S_2, \dots, S_k\}.$$

根据定义 4.1, 服务的先决条件说明服务在什么样的条件或环境下是可用的, 服务的执行结果则说明在服务执行后会使世界状态发生什么样的变化。服务的输入输出参数说明服务所要处理的数据对象。

根据定义 4.2, 一个复合服务是由若干个原子服务所组成的有限有序服务流。在这些原子服务复合的过程中, 由于不同原子服务的执行会引起相关世界状态的变化, 因此, 针对不同的复合服务或者同一复合服务的不同复合阶段, 所建立的解释模型 I 也会随之不同。为了表示这些解释模型之间的关系, 需要定义解释之间的二元关系 “ $<$ ” (定义 4.3)。

定义 4.3 (二元解释关系) 设存在非循环 TBox T , 基于 T 上的服务 $S = (pre, sname,$

$inputs, outputs, post, pref$), 以及建立在 T 上的解释模型 I 。解释模型 I', I'' 满足 $I' < I''$, 当且仅当:

$$(1) ((A^I \oplus A^{I'}) \setminus \{a^I | A(a) \in \{sname, inputs, outputs\}\}) \subseteq A^I \oplus A^{I''}$$

$$(2) ((r^I \oplus r^{I'}) \setminus \{(a^I, b^I) | r(a, b) \in \{sname, inputs, outputs\}\}) \subseteq r^I \oplus r^{I''}$$

其中 “ $\oplus$ ” 表示集合的对称差, I'' 称为 I' 的后继模型。

由定义 4.3 可得, 执行某一服务 $S = (pre, sname, inputs, outputs, post, pref)$ 时, 如果对于两个解释 I, I' 满足: $I' \vdash post$, $I < I'$ 且不存在其它的解释 I'' 符合条件 $\{I < I'' < I' \cap I'' \vdash post\}$, 则可得结论: 通过执行该服务, 使得该服务所处的世界状态从 I 转换到 I' 。即如果 $post = P/E$, 则有 $\{I, I' \vdash post\} \Leftrightarrow \{if I \vdash P \text{ then } I' \vdash E\}$ 成立。由此可得服务可执行性定义 (定义 4.5)。此外, 在服务执行前, 必须保证服务与知识库相一致, 即服务相对于某一知识库是可执行的。因此, 在给出服务执行性定义之前, 首先给出服务一致性定义 (定义 4.4)。

定义 4.4 (服务一致性定义) 设存在非循环 TBox T , 基于 T 上的服务 $S = (pre, sname, inputs, outputs, post, pref)$, 以及 T 的解释模型 I , 但不存在解释模型 I' 满足定义 4.3, 即服务执行后, I 不存在后继模型, 则认为该服务 S 与 T 的解释模型 I 不一致。否则, 认为该服务 S 与 T 的解释模型 I 一致。

定义 4.5 (服务执行) 设存在非循环 TBox T , 基于 T 上的一致性服务 $S = (pre, sname, inputs, outputs, post, pref)$, 以及 T 的解释模型 I, I' 且 I, I' 基于相同的领域 (domain)。则经过服务的执行解释模型 I 将转变为 I' ($I \Rightarrow I'$) 当且仅当:

$$(1) I, I' \vdash post, \text{ 并且}$$

$$(2) \text{ 不存在解释模型 } J \text{ 满足 } I, J \vdash post, j \neq I' \text{ 且 } J < I'.$$

基于定义 4.5, 当一致性服务复合链 $S = \{SL, CF\}$, $SL = CF\{S_1, S_2, \dots, S_k\}$ 执行时, 则必然存在一系列的解釋模型 $I_0, I_1, \dots, I_k$, 这些解释模型随着服务复合链的执行而有序的改变。因此, 复合服务执行的形式化定义如定义 4.6。

定义 4.6 (复合服务执行) 设存在非循环 TBox T , 基于 T 上的服务复合链 $S = \{SL, CF\}$,

$SL=CF\{S_1, S_2, \dots, S_k\}$, 则经过复合服务的执行, 解释模型 I 将转变为 I' , 当且仅当存在基于 T 的模型 $I_0, I_1, \dots, I_k$, 满足 $I=I_0, I'=I_k$, 同时 $I_{i-1} \Rightarrow I_i (1 \leq i \leq k)$ 。

4.2.3 DLAIIP 逻辑证明

设已知知识库 $KB=\{TBox, ABox\}$, 其中, $TBox$ 满足非循环性质。基于知识库 KB , 存在一个待执行的复合服务 $S=\{SL, CF\}$, $SL=CF\{S_1, S_2, \dots, S_k\}$ 。直观地说, 如果复合服务 S 可执行, 那么对于构成复合服务 S 的每一个原子服务 $S_i (1 \leq i \leq k)$ 来说, 其执行前后所引起的世界状态的变化相对于当前知识库而言必然是一致的。因此, 应符合如下条件: 设 T' 、 A' 分别为原子服务 S_i 执行前知识库 KB 的 $TBox$ 和 $ABox$, T'' 和 A'' 分别为原子服务 S_i 执行之后的知识库 KB 的 $TBox$ 和 $ABox$, 那么对于 $S_i (1 \leq i \leq k)$ 则应满足: 执行前 A' 与 T' 相一致, 执行之后 A'' 与 T'' 仍然相一致。下面给出具体的证明过程。

根据定义 4.5, 执行每一个原子服务, 其相关的解释模型 I 将会转变为 I' 。同时, 随着解释模型的变化, $TBox$ 中的相关概念 (A) 和关系 (r) 的解释也会发生变化, 分别由 A^I 转变为 A'^I , r^I 转变为 r'^I 。因此, 先给出求解 A'^I 、 r'^I 的引理 (引理 4.1)。

引理 4.1 (A'^I 、 r'^I 求解) 设存在非循环 $TBox T$, 基于 T 上的服务 $S=(pre, sname, inputs, outputs, post, pref)$, 以及 T 的解释模型 $I, I', I \Rightarrow I'$ 且 I, I' 与服务 S 相一致。如果 A 是 T 中定义的原始概念, r 是关系名, 则

$$\begin{aligned} A'^I &= (A^I \cup \{a^I \mid P/A(b) \in post \cap I \Rightarrow P\}) \cup \{a^I \mid P/\neg A(b) \in post \cap I \Rightarrow P\}, \\ r'^I &= (r^I \cup \{(a^I, b^I) \mid P/r(a, b) \in post \cap I \Rightarrow P\}) \cup \{(a^I, b^I) \mid P/\neg r(a, b) \in post \cap I \Rightarrow P\}. \end{aligned}$$

在执行复合服务 S 的每一个原子服务 S_i 时, 首先必须确定原子服务 S_i 的 $pref_i$ 满足用户的个人需求和偏好, 这一性质称为需求偏好可满足性; 第二步必须判断原子服务 S_i 所处状态是否满足其先决条件 pre_i , 只有当先决条件 pre_i 得到满足时, 该原子服务才能被执行, 这一性质称为可执行性; 如果原子服务 S_i 满足可执行性, 且已知 S_i 的当前解释模型为 I , 后继模型为 I' , 原子服务 S_i 的执行结果为 $post_i$, 则 S_i 必须同时满足条件 $I' \vdash post_i$, 即原子服务的执行结果必须有效, 该原子服务才可被执行, 这一性质称为执行结果有效性; 如果原子服务同时满足需求偏好可满足性、可执行性和执行结果有效性, 则执行复合服务 S 时, 其前驱原子服务 S_i 的输出 $outputs$ 必须与后继原子服务 S_{i+1} 的输入 $inputs$ 相匹配 ($1 \leq i \leq k-1$), 即只有当匹配结果满足一定范围时, S_i 与 S_{i+1} 才可复合, 这

一性质称为输入输出相容性。

综上所述, 复合服务 $S=\{SL, CF\}$, $SL=CF\{S_1, S_2, \dots, S_k\}$ 可被执行, 当且仅当复合服务 S 必须同时满足需求偏好可满足性、可执行性、结果有效性和输入输出相容性。见定理 4.1。

定理 4.1 (服务复合有效性判定) 设存在非循环 TBox T , ABox A 以及基于 T 上的服务复合链 $S=\{SL, CF\}$, $SL=CF\{S_1, S_2, \dots, S_k\}$, 其中 $S_i=(pre_i, sname_i, inputs_i, outputs_i, post_i, pref_i)$, 如果满足下述条件则认为复合服务有效。

- (1) 需求偏好可满足性: $S_1, S_2, \dots, S_k$ 满足需求偏好, 当且仅当对于 A 和 T 的所有解释模型满足如下条件:
 - $I \vdash pref_i$
 - 对于任意属于 $1 \leq i \leq k$ 的 i , 以及所有解释模型 I_i , 且 $I \Rightarrow I_i$, 则有 $I_i \vdash pref_{i+1}$ 。
- (2) 可执行性: $S_1, S_2, \dots, S_k$ 相对于 A w.r.t T 是可执行的, 当且仅当对于 A 和 T 的所有解释模型满足如下条件:
 - $I \vdash pre_i$
 - 对于任意属于 $1 \leq i \leq k$ 的 i , 以及所有解释模型 I_i , 且 $I \Rightarrow I_i$, 则有 $I_i \vdash pre_{i+1}$ 。
- (3) 结果有效性: 设声明 $post$ 是复合服务 $S_1, S_2, \dots, S_k$ 的执行结果, 当且仅当对于任意属于 $1 \leq i \leq k$ 的 i , 以及所有解释模型 I_i , 且 $I \Rightarrow I_i$, 则有 $I_i \vdash post$ 。
- (4) 输入输出相容性: 服务复合链 $S_1, S_2, \dots, S_k$ 中, S_{i-1} 的输出 $outputs$ 与 S_i 的输入 $inputs$ 必须相匹配 ($1 \leq i \leq k$), 即匹配度必须达到一定范围。Matching_Degree ($outputs_{S_{i-1}}, inputs_{S_i} \in (\theta, 1)$, (θ 为某一固定值))。

根据定理 4.1, 一个复合服务可被执行, 当且仅当该复合服务必须满足需求偏好可满足性、可执行性、结果有效性和输入输出相容性四个条件。对于需求偏好可满足性可以将其归约为描述逻辑知识库的一致性推理; 对于可执行性和结果有效性的判定, 根据引理 4.2 可以证明这两个条件是可以相互转化的。因此, 只需考虑将这两个条件之一转化为基于描述逻辑的一致性证明便可证明其逻辑有效性 (定理 4.3)。而对于输入输出相容性, 则只需给出相应的基于描述逻辑的匹配公式, 并将匹配公式归结为描述逻辑的可满足性即可证明 (定理 4.4)。

定理 4.2 需求偏好可满足性可归结为知识库 KB 中 TBox 和 ABox 的一致性问题。

证明：设 P_r 表示用户请求的需求偏好， P_p 表示已发布服务的 $pref$ 分量， Q_r 表示用户需求偏好实例， Q_p 表示已发布服务实例。需求偏好可满足性成立当且仅当满足如下两条规则：

$$(1) KB \cup P_r \cup P_p \vdash \forall x: Q_r(x) \rightarrow Q_p(x) \Leftrightarrow KB \cup P_r \cup P_p \cup \{c: Q_r \hat{\cap} \neg Q_p\};$$

$$(2) KB \cup P_r \cup P_p \vdash \forall x: Q_p(x) \rightarrow Q_r(x) \Leftrightarrow KB \cup P_r \cup P_p \cup \{c: Q_p \hat{\cap} \neg Q_r\};$$

规则 (1) 和 (2) 可归结为判定 $KB \cup P_r \cup P_p \cup \{c: Q_r \hat{\cap} \neg Q_p\}$ 与 $KB \cup P_r \cup P_p \cup \{c: Q_p \hat{\cap} \neg Q_r\}$ 的可满足性。 $\square$

引理 4.2 复合服务 $S=\{SL, CF\}$ ， $SL=CF\{S_1, S_2, \dots, S_k\}$ 执行过程中，可执行性和结果有效性是可以相互转换的。

证明：(1) 可执行性转化为结果有效性：设 “ pre_1 ” 表示复合服务刚开始执行时所需满足的初始条件，即构成复合服务的第一个原子服务 S_1 的先决条件 “ pre_1 ” 在服务执行时必须得到满足，即 $I \vdash pre_1$ 。而条件 $I \vdash pre_1$ 可等价于是一个空服务 $S=(\Phi, \Phi, \Phi, \Phi, \Phi, \Phi)$ 的执行。即 $I \vdash pre_1 \Leftrightarrow S=(\Phi, \Phi, \Phi, \Phi, \Phi, \Phi)$ 的执行。

(2) 结果有效性转化为可执行性：设 P 是复合服务 S 的执行结果。同时构造新的复合服务 $S=\{SL, CF\}$ ， $SL=CF\{S_1, S_2, \dots, S_k, S'\}$ ，其中 $S'=(\Phi, \Phi, \Phi, \Phi, \Phi, \{P\})$ 。则 P 可看作复合服务 $S=\{SL, CF\}$ ， $SL=CF\{S_1, S_2, \dots, S_k, S'\}$ 的执行结果，当且仅当复合服务 $S=\{SL, CF\}$ ， $SL=CF\{S_1, S_2, \dots, S_k, S'\}$ 是可执行的。 $\square$

定理 4.3 结果有效性和可执行性可归结为知识库 KB 中 TBox 和 ABox 的一致性问题。

证明：根据服务 $S = (pre, sname, inputs, outputs, post, pref)$ 的各个分量，构造与该服务相对应的 TBox T' 、ABox A' ，则 ABox A' 必然 $\subseteq$ 基于 TBox 的 ABox A'' 。因此，当服务结果有效时，则 A' 必然与 T' 相一致，否则必然不一致。反之亦然。因此，只需构造出服务执行前后的 TBox T' 和 ABox A' ，然后再利用推理机证明 T' 与 A' 的一致性即可。

准备工作：

- ①将输入、输出中出现的所有概念及其父子概念组成的集合记为 C ；对于每一个 $c \in C$ ，引入概念 T_c^i 表示在相应的模型 I_i 中 c 对应的概念定义；
- ②将输入、输出中出现的每一个原始概念记为 A^i ；
- ③将输入、输出中出现的每一个关系记为 r_i ；
- ④将输入、输出中出现的实例组成的集合记为 Ins ，同时对于任意 $a \in Ins$ ，引入辅助关系 r_a ；
- ⑤将在输入、输出中出现且已命名的名字集合记为 N ；
- ⑥对于附加的 $a \notin Ins$ ，则用 a_n 表示（引入 r_a 时引入）。

(1) 构建 T' ：

$$① T_1 = \{N = \bigcup_{a \in Ins} \{a\}\}$$

② $T_2 = \{\text{一系列由 } T_c^i \text{ 组成的并、交、非公式集，记为 } F\}$

$$F = \{T_A^i \equiv (N \cap A^i), T_{\neg c}^i \equiv \neg T_c^i, T_{C \cap D}^i \equiv T_C^i \cap T_D^i, T_{C \cup D}^i \equiv T_C^i \cup T_D^i$$

$$T_{\geq m, r, c}^i \equiv (N \cap \bigcup_{0 \leq j \leq m} ((\geq j \ r^i (N \cap T_c^i)) \cap (\geq (m-j) \ r^i (\neg N \cap T_c^i))))),$$

$$T_{\leq m, r, c}^i \equiv (N \cap \bigcup_{0 \leq j \leq m} ((\leq j \ r^i (N \cap T_c^i)) \cap (\leq (m-j) \ r^i (\neg N \cap T_c^i))))\}$$

$$③ T' = T_1 \cup T_2$$

(2) 构建 A' ：

①引入附加关系，同时引入相应的实例，这部分实例所组成的集合如下：

$$A_1 = \{a: (\exists r_b \{b\} \cap \forall r_b \{b\}) \mid a \in Ins \cup a_n, b \in Ins\}$$

②服务执行后所生成的实例集：

$$A_2 = \{\bigcup_{1 \leq i \leq k} post_i\}$$

③不受服务执行影响而保持不变的概念实例集, A 为原始概念;

$$A_3 = a: ((A^{i-1} \cap \neg \forall r_a.T_c^i \cap \neg \forall r_a.\exists r^i.\{b\} \cap \neg \forall r_a.\exists r^i.\neg\{b\}) \cap (A(a) \notin post_i) \rightarrow A^i)$$

④不受服务执行影响而保持不变的关系实例集, r 为关系;

$$A_4 = a: ((\exists r^{i-1}.\{b\} \cap \neg \forall r_a.T_c^i \cap \neg \forall r_a.\exists r^i.\{b\} \cap \neg \forall r_a.\exists r^i.\neg\{b\}) \cap (r(a,b) \notin post_i) \rightarrow \exists r^i.\{b\})$$

⑤服务复合链执行之前的最初实例集: A_5

$$⑥ A' = A_1 \cup A_2 \cup A_3 \cup A_4 \cup A_5$$

(3) 检验 A' 与 T' 的一致性 采用集于描述逻辑的推理机 *Racer*^[74] 可判之。 $\square$

定理 4.4 服务复合输入输出相容性判定可归结为知识库 KB 中 TBox 和 ABox 的可满足性问题。

证明: 设 D_r 表示请求服务, D_p 表示发布服务, S_r 表示请求服务实例, S_p 表示发布服务实例, 满足如下三条规则任意一条则定义为输入输出相容:

① 概念连接:

$$KB \cup D_r \cup D_p \cup \{\exists x: S_r(x) \wedge S_p(x)\} \Leftrightarrow KB \cup D_r \cup D_p \cup \{c: S_r \sqcap S_p\},$$

可归结为判定 $KB \cup D_r \cup D_p \cup \{c: S_r \sqcap S_p\}$ 的可满足性;

② 概念包含: $KB \cup D_r \cup D_p \vdash \forall x: S_r(x) \rightarrow S_p(x)$

$$\Leftrightarrow KB \cup D_r \cup D_p \cup \{c: S_r \sqcap \neg S_p\};$$

$$KB \cup D_r \cup D_p \vdash \forall x: S_p(x) \rightarrow S_r(x)$$

$$\Leftrightarrow KB \cup D_r \cup D_p \cup \{c: S_p \sqcap \neg S_r\};$$

可归结为判定 $KB \cup D_r \cup D_p \cup \{c: S_r \sqcap \neg S_p\}$ 与 $KB \cup D_r \cup D_p \cup \{c: S_p \sqcap \neg S_r\}$ 的不可满足性。

③ 概念交: $KB \cup D_r \cup D_p \vdash \exists x: S_r(x) \wedge S_p(x)$

$$\Leftrightarrow KB \cup D_r \cup D_p \cup \{S_r \sqcap S_p \sqsubseteq \perp\},$$

可归结为判定 $KB \cup D_r \cup D_p \cup \{S_r \sqcap S_p \sqsubseteq \perp\}$ 的可满足性。 $\square$

根据引理 4.2、定理 4.2、定理 4.3 和定理 4.4 的证明,可以得出定理 4.1 给出的检验服务复合有效性的四个条件是基于描述逻辑且可判定的。

4.2.4 DLAIP 复杂度分析

根据定理 4.1,服务复合的有效性判定问题可归结为描述逻辑 $ALCQIO$ 的可满足性和一致性推理问题。因此,本文可以利用 $ALCQIO$ 的复杂度对 DLAIP 中服务复合的复杂度进行分析。如定理 4.5 所示。

定理 4.5 (服务复合复杂度分析) 基于非循环 TBox 的服务复合机制的复杂度分析如下:

- 对于描述逻辑 ALC 、 $ALCO$ 、 $ALCQ$ 和 $ALCQO$,如果其数字约束 (*number restrictions*) 只能采用一元谓词表示,则服务复合复杂度是一个 *PSPACE-complete* 问题;
- 对于描述逻辑 $ALCI$ 和 $ALCIO$,服务复合复杂度是一个 *EXPTIME-complete* 问题;
- 对于描述逻辑 $ALCQI$ 和 $ALCQIO$,无论其数字约束采用一元谓词表示还是二元谓词表示,服务复合复杂度是一个 *co-NEXPTIME-complete* 问题。

4.2.5 关于循环 TBox 的讨论

在上述讨论中,我们主要讨论的是非循环 TBox 前提条件下 DLAIP 的推理、证明和复杂度分析。但是,在现实应用中,循环 TBox 的情况也普遍存在。如果在讨论中,对 TBox 不加任何限制,既可以是循环 TBox,也可以是非循环 TBox,则会产生如下语义问题:对于非循环 TBox 而言,任何概念都是由原子概念唯一确定;但是,对于循环 TBox 而言,则存在对概念的循环定义。此时,增加了概念的不确定性。例如,设存在如下知识库:

$$A \equiv \{Bird(a)\} \quad T \equiv \{Bird \equiv \exists parent.Bird\} \quad post \equiv \{Cat(b)\}$$

显然,按照上述知识库, $Bird(a)$ 并不是服务 $S=(\Phi, \Phi, \Phi, \Phi, post, \Phi)$ 的执行结果。而事实上,在某种情况下, $Bird(a)$ 完全可以是 $S=(\Phi, \Phi, \Phi, \Phi, post, \Phi)$ 的执行结果。

针对上述问题,我们完全可以采用描述逻辑中最小或最大语义固定点理论^[6],将循环定义概念转变为非循环定义概念以消除循环定义,从而初步解决此类问题。

4.3 应用举例

本节以一个新生到图书馆借书问题为例说明 DLAIP 的工作流程。该问题描述如下：设研究生一年级新生 Tom 持录取通知书去图书馆办借书证并借至少六本书。已知学生必须持有通知书才可进入图书馆办证。如果该学生是本科生，则最多能借五本书；如果该学生是研究生，则最多可借十本书。对上述实例进行分析，可以抽象出四个服务：

$S_1=\{pre_1, sname_1, inputs_S_1, outputs_S_1, post_1, pref_1\}$, $S_2=\{pre_2, sname_2, inputs_S_2, outputs_S_2, post_2, pref_2\}$, $S_3=\{pre_3, sname_3, inputs_S_3, outputs_S_3, post_3, pref_3\}$, $S_4=\{pre_4, sname_4, inputs_S_4, outputs_S_4, post_4, pref_4\}$ 。其中服务 S_1 表示该学生持有通知书； S_2 表示对持有通知书的学生办理借书证服务； S_3 表示研究生借书服务； S_4 表示本科生借书服务。各个服务的相关分量如下所示：

Service S_1 :

$pre_1=\Phi$;
 $sname_1=hold_notice$;
 $inputs_S_1=\{notice(b), new_student(a)\}$;
 $outputs_S_1=holds(a, b)$;

$post_1=\{holds(a, b), notice(b)\}$;
 $pref_1=\{\geq 6 \text{ has_books_borrowed}\}$;

Service S_2 :

$pre_2=\{new_student(a), register.record(a)\}$;

$sname_2=apply_for_library_card$;
 $inputs_S_2=\{holds(a, b), notice(b)\}$;
 $outputs_S_2=\{holds(a, c), library_card(c)\}$;

$post_2=\{holds(a, c), holds.notice(b)/library_card(c), holds.notice(b)/library_no_card(c)\}$;
 $pref_2=\{\geq 6 \text{ has_books_borrowed}, \leq 5 \text{ has_books_borrowed}\}$;

Service S_3 :

$pre_3=\{graduate(a), holds(a, b), library_card(b)\}$;
 $sname_3=borrow_book$;
 $inputs_S_3=\{holds(a, b), library_card(b)\}$;
 $outputs_S_3=\{ten_books_borrowed(a, c), ten_books(c)\}$;

$post_3=\{holds(a, c), ten_books(c)\}$;
 $pref_3=\{\geq 6 \text{ has_books_borrowed}\}$;

Service S_4 :

$pre_4=\{undergraduate(a), holds(a, b), library_card(b)\}$;

$sname_4=borrow_book$;
 $inputs_S_4=\{holds(a, b), library_card(b)\}$;
 $outputs_S_4=\{five_books_borrowed(a, c), five_books(c)\}$;

$post_4=\{holds(a, c), five_books(c)\}$;
 $pref_4=\{\leq 5 \text{ has_books_borrowed}\}$;

根据服务 S_1 、 S_2 、 S_3 、 S_4 中出现的概念，可定义知识库中如下非循环 TBox T 和 ABox A ：

A :

$T=\{student \equiv graduate \cup undergraduate \cup new_student, \\ book \equiv five_book \cup ten_book, \\ \{holds(domain, range) \mid domain \in \{student\}, range \in \{librarycard, books\}\} \\ \{has_books_borrowed(domain, range) \mid domain \in \{student\}, range \in \{xsd:integer\}\}\}$
 $A=\{new_student(Tom), notice(graduate_notice), \exists register.record(Tom), \geq 6 has_books_borrowed\}$

根据初始条件 ABox A 的信息，利用 4.2.3 节定理 4.1 可得复合服务 $S=CF\{S_1, S_2, S_3, S_4\}$ ，其中 CF 表示顺序控制流，满足新生 Tom 的服务请求。

4.4 本章小结

本章以描述逻辑为基础，通过扩展 XOWL-S 模型，提出了一种基于描述逻辑的面向本体知识路由的语义 Web 服务动态复合机制——DLAIP。具体论述以下内容：

- (1) 介绍了包含用户需求偏好的扩展 XOWL-S 模型；
- (2) 论证了服务复合机制 DLAIP 的逻辑合理性，并对其复杂度给予初步分析；
- (3) 通过一个应用实例详细介绍 DLAIP 的工作流程。

第五章 基于本体的 E-learning 教学平台 OBETP 的设计和开发

本章利用前述章节介绍的语义 Web、本体知识路由的相关知识, 设计和开发了一个基于本体语义技术的 E-learning 教学平台 OBETP (Ontology Based E-learning Teaching Platform), 该平台主要包括考试子平台(EP)、学习子平台(LP)、授课子平台(TP)和文献检索子平台(SLRP)。本文主要实现其中的文献检索子平台以及学习子平台和授课子平台中作业相关的子功能。该工作是国家自然科学基金《知晓内容和环境的本体知识路由研究》实验平台的基础性工作, 为实验平台的最终完成提供了有价值的开发经验。

5.1 OBETP 的设计

在知识经济时代, 教育信息化主要体现在如何利用信息技术提高教学质量、增强教学效果和扩大教学规模以及科学、高效的教学管理等方面。教育活动包括教育者、被教育者、教育内容和教育媒体。教育信息化的基础和核心是建设丰富的优质的教学信息资源, 并开发教学支撑平台, 对教学活动进行有效的支持。但目前的教学信息资源建设和教学工具支持尚不尽人意, 仍然存在不足:

- 缺乏实用的规范, 以至于教师没有将有用的精力投入到该做的事情上。
- 教学信息资源制作缺乏专业工具。
- 教学支撑平台缺乏组装、定制和扩充能力。每个单位需要的教学支撑平台都要自己从头开发, 资源重用性差, 造成大量财力和人力浪费。

总之, 从当前教学信息化的情况看, 无论是教学信息资源建设还是教学支撑工具上, 都缺乏对教育信息化本身的深入理解, 对信息技术如何对教学进行有效支持缺乏系统的规范的方法。

本实验原型正是考虑上述不足, 采用本体论的思想, 从技术方面提出了一种更高层次的可共享、可重构、可复用、可扩充、可定制的基于本体的教学支撑平台OBETP, 并对其进行了初步实现。

5.1.1 OBETP 总体结构的设计

OBETP的系统结构如图5-1所示。从图5-1可知，OBETP的结构分为四层：本体概念与发布层、辅助概念描述层、代码模块层、资源描述层。本体概念与发布层包括两部分，一是教学本体，二是组装发布功能：接收用户提出的请求，利用本体查询语言，根据教学本体得出用户请求功能的概念描述，然后按照所得的概念描述组装成满足用户需求的的教学行为系统。辅助概念描述层主要保存本体中涉及的各种概念描述，为系统组装提供支持。代码模块层主要包括本体中包含概念的具体模块实现，这些实现不同于一般的程序模块，该层的模块要求具有URI的性质，该模块的功能可从网络上通过URI直接调用实现^[75]。资源描述层主要描述相关的教学资源如课件制作、教材等。

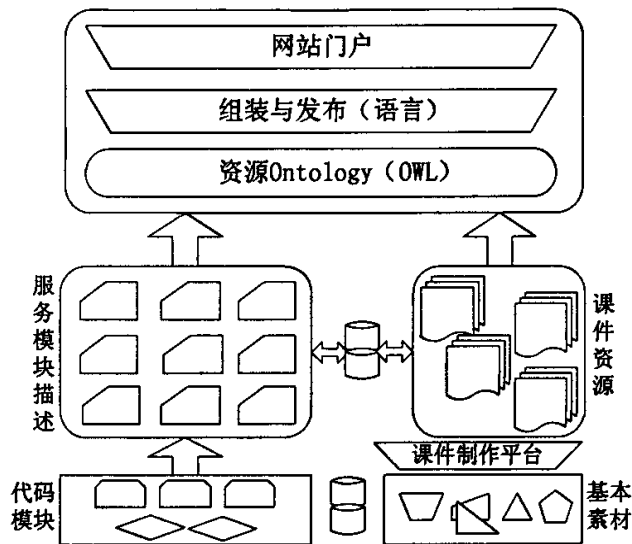


图5-1 OBETP系统结构

Fig. 5-1 The OBETP system framework

● 本体概念与发布层

该层主要包括三部分：网站门户、组装与发布、教学本体。其中教学门户网站是用户使用的界面，该门户网站界面应当具有用户可定制的特点。组装与发布的主要功能是对用户请求的功能进行描述，同时组成一个服务系统。其中目标系统的描述主要采用描述语言的方式，如DAML-S、OWL-S等，该语言描述可通过对教学本体的查询重新组合自动产生。教学平台本体主要描述教学平台系统的有关概念、模块及其相互关系，该本体实质上是教学支撑平台的显式描述。

● 辅助概念描述层

该层的主要功能是对教学本体中的辅助概念进行较为精确、全面的描述,即对教学有关的基本概念进行定义。同时对概念之间的关系进行具体的描述,如调用关系、参数传递等。一般来说,该部分只需要对本体中的基本概念进行定义,而一些组合概念可以利用基本概念进行定义。

● 代码模块层

该层主要是对概念描述层中的基本概念进行实现,并以URI 的方式供组装发布层使用。该部分中的模块代码的结构要进行特别设计,这些模块必须能以URI 的方式与其他模块一起形成一个更大的有机系统,而且该系统也必须可作为URI 方式使用。也就是说,这些模块的设计要支持教学本体中的模块对应的概念组合。例如,答疑和讨论中都要用到文字提交服务,在设计时若文字提交服务作为一个URI,那么该服务就必须能同时与其他业务服务组成答疑和讨论系统。如何设计提交服务模块才能完成以上功能是关键技术。

● 资源描述

在教学系统中,资源占据重要的位置。本系统中的资源主要是指课件资源、教材、参考教材等,这些资源都以URI 的形式存放在网络上。对于课件本身来说,其基本组成部分称为基本素材。由于基本素材数量巨大,而且每门课程常用的基本素材较为稳定,因而根据基本素材的特点可以进行不同程度的优化。例如,可以将常用的基本素材存于同一数据库中,这样同一课件所访问的基本素材在同一个机器上的概率就很大,这样一次请求可访问多个素材,提高了请求效率,加快了访问速度。

5.1.2 教学本体建模

在本原型系统中,采用本体建模语言 OWL (Web Ontology Language)作为教学本体建模的工具。本体编辑工具为 protégé 3.1^[76]。

(1) 主要本体类及属性结构:

在本教学本体中,包括 28 个类,29 个属性。其中包括 4 个主体类: examPlatform 类, learningPlatform 类, teachingPlatform 类和 literatureRetrievalPlatform 类。

● examPlatform 类属性结构如下:

Class (elearning: examPlatform)

DatatypeProperty (hasName domain (elearning: examPlatform) range (xsd: string))

ObjectProperty (hasexamResult domain (elearning: examPlatform) range (elearning: ExamResult))

ObjectProperty (hasexamReview domain (elearning: examPlatform) range (elearning: ExamReview))

ObjectProperty (hasexamSubmit domain (elearning: examPlatform) range (elearning: ExamSubmit))

ObjectProperty (hasonlineAnswer domain (elearning: examPlatform) range (elearning: QuestionAnswer))

ObjectProperty (hasquestionBase domain (elearning: examPlatform) range (elearning: QuestionBase))

● learningPlatform 类属性结构如下:

DatatypeProperty (hasName domain (elearning: learningPlatform) range (xsd: string))

ObjectProperty (hashomeworkSubmit domain (elearning: learningPlatform) range (elearning: HomeworkSubmit))

ObjectProperty (haslquestionAnswer domain (elearning: learningPlatform) range (elearning: QuestinAnswer))

ObjectProperty (hasnetworkCourse domain (elearning: learningPlatform) range (elearning: NetworkCourse))

ObjectProperty (hasonlineLearning domain (elearning: learningPlatform) range (elearning: OnlineLearning))

ObjectProperty (hasforum domain (elearning: learningPlatform) range (elearning: Forum))

● teachingPlatform 类属性结构如下:

DatatypeProperty (hasName domain (elearning: teachingPlatform) range (xsd: string))

ObjectProperty (hascourseDiscussion domain (elearning: teachingPlatform) range (elearning: CourseDiscussion))

ObjectProperty (hashomeworkAssign domain (elearning: teachingPlatform) range (elearning: HomeworkAssign))

ObjectProperty (hashomeworkReview domain (elearning: teachingPlatform) range (elearning: HomeworkReview))

ObjectProperty (hasquestionAnswer domain (elearning: teachingPlatform) range (elearning: QuestionAnswer))

● literatureRetrievalPlatform 类属性结构如下:

DatatypeProperty (hasName domain (elearning: literatureRetrievalPlatform) range (xsd: string))

ObjectProperty (hasliteratureRetrival domain (elearning: teachingPlatform) range (elearning: literatureRetrieval))

(2) 本体类层次示意图:

教学本体类的层次结构示意图如图 5-2 所示:

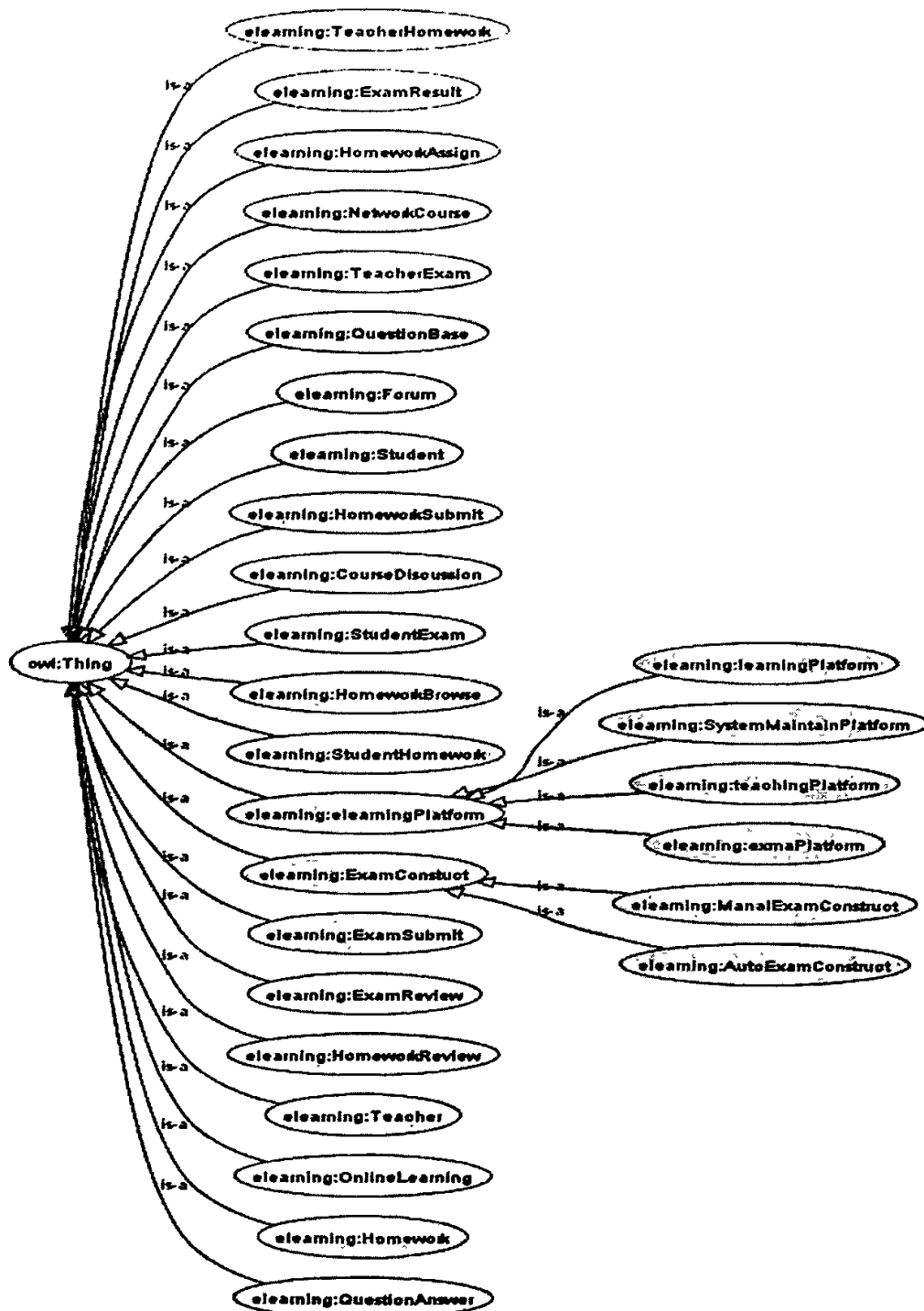


图 5-2 教学本体类层次结构示意图

Fig. 5-2 The hierachical framework of e-learning ontology

5.2 文献检索子平台 SLRP 的设计

在E-learning平台中,文献检索功能是用户常用功能之一。然而传统文献搜索引擎一般只针对网页文件,采用关键字技术,对一些其它文献的搜索力度还远远不够。仍然存在如下局限性:

- 信息检索本质上是语义检索,而传统信息检索系统都是基于独立词索引,词汇间的同义性和单个词汇的歧义性,是造成检索结果不理想的客观原因。
- 词汇间同义性引起的弊端体现在:不同的检索者,其检索目的、知识背景、语言习惯之间的差异,会导致他们对同一概念用不同词汇来表达。
- 单个词汇的歧义性表现为一个词汇在不同的语言环境下,或被不同的人使用,则可能代表不同的含义。

本文正是针对上述缺点,以E-learning平台为依托,利用本体技术,通过抽取文献语义元数据,然后利用这些语义元数据实现对文献的语义检索。该子平台的设计主要包括文献元数据的抽取和文献本体的构建两部分。

5.2.1 文献元数据的抽取

在文献检索子系统中,主要检索对象是科技论文。然而,论文不像一般形式的网页,它们一般是自由格式的文本,从中抽取元数据有一定的难度。但是,根据观察和经验分析,科技论文还是有一定的固定结构。比如说,大多数论文可以归纳为以下结构:

- 论文标题
- 作者名字,作者的相关描述信息
- 论文的摘要
- 论文的关键词(可以没有,英文论文不太注重写关键词)
- 论文的主体
- 论文的引文

当然,一篇文章还有其它一些方面的属性,比如大小、发布地址等在网络爬虫进行文献抓取时就能够自动确定。在本系统中,抽取工作所涉及到的元数据如表5-1所示。

表5-1 论文元数据

Table 5-1 The metadata of one paper

元素	说明
Title	文章标题
AuthorName	文章作者名字
AuthorAff	作者描述信息
Abstract	文章摘要
Keywords	文章关键字
References	文章引文

对于论文元数据的抽取，可以使用很多方法，但是最常用的是正则表达式。可以考虑先利用如下的正则表达式的规则（用‘RT’表示换行，单引号中的文字为字符串）：

- (Title) ‘RT’
- (((CreatorName ‘,’)*(CreatorName)) ‘RT’
- (CreatorAffiliation))* ‘RT’
- ‘Abstract’ Description ‘RT’
- (‘Keywords’ (SubjectKeyword ‘,’)*(SubjectKeyword))* ‘RT’ (Body)
- ‘References’ ‘RT’
- (RefBody)

类似地，还可以定义其它的一些RT规则。对于一个文档，系统可以判断它是不是符合这些正则表达式规则。如果符合，那么就对它进行相应的元数据抽取。如果不符合，那么暂时不对它进行处理，等待细化规则后进行处理。

5.2.2 文献本体建模

文献本体的构建仍然采用 protégé 3.1。

(1) 主要本体类及属性描述

在文献本体中，主要包括四个主体类：文章类（pub: Article）、文章作者类（pub: Person）、文章出版机构类（pub: Publication）和文章所属主题类（pub: Subject）。其中每个类的属性描述如下：

- 文章类属性结构：

Class (pub: Article)

DatatypeProperty (Abstract domain (pub: Article) range (xsd: string))

DatatypeProperty (hasAffiliation domain (pub: Article) range (xsd: string))

DatatypeProperty (hasEmail domain (pub: Article) range (xsd: string))

DatatypeProperty (keywords domain (pub: Article) range (xsd: string))

DatatypeProperty (hasTitle domain (pub: Article) range (xsd: string))

ObjectProperty (bothCited domain (pub: Article) range (pub: Article))

ObjectProperty (bothCiting domain (pub: Article) range (pub: Article))

ObjectProperty (Citing domain (pub: Article) range (pub: Article))

ObjectProperty (hasAuthor domain (pub: Article) range (pub: Person))

ObjectProperty (hasSubject domain (pub: Article) range (pub: Subject))

ObjectProperty (isCited domain (pub: Article) range (pub: Article))

ObjectProperty (isPublished domain (pub: Article) range (pub: Publication))

- 文章作者类属性结构:

Class (pub: Person)

DatatypeProperty (has_Male domain (pub: Person) range (xsd: string))

DatatypeProperty (has_Title domain (pub: Person) range (xsd: string))

DatatypeProperty (has_Affiliation domain (pub: Person) range (xsd: string))

- 文章出版机构类属性结构:

Class (pub: Publication)

DatatypeProperty (has_Name domain (pub: Publication) range (xsd: string))

DatatypeProperty (has_Rank domain (pub: Publication) range (xsd:
nonNegativeInteger))

- 文章所属主题类属性结构:

Class (pub: Subject)

Class (pub: ComputerScience partial Annotation (rdfs: comment "Computer
Science forms a subclass of Subject.") pub: Subject)

Class (pub: Artificial_Intelligence partial Annotation (rdfs: comment "AI forms
a subclass of ComputerScience.") pub: ComputerScience)

Class (pub: Network_Security partial Annotation (rdfs: comment "Network

Security forms a subclass of ComputerScience.") pub:
ComputerScience)

Class (pub: Parallel_Computing partial Annotation (rdfs: comment "Network
Security forms a subclass of ComputerScience.") pub:
ComputerScience)

(2) 本体类层次示意图

在文献本体中，目前总共定义了八个类：文章类 (pub: Article)、文章作者类 (pub: Person)、文章出版机构类 (pub: Publication)、文章所属主题类 (pub: Subject)、计算机学科类 (pub: ComputerScience)、人工智能类 (pub: Artificial_Intelligence)、网络安全类 (pub: Network_Security) 和并行计算类 (pub: Parallel_Computing)。其类层次结构示意图如图5-3所示：

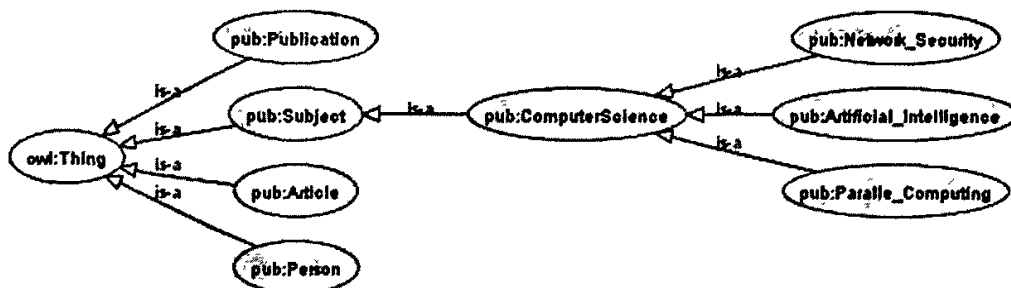


图5-3 文献本体类层次结构示意图

Fig. 5-3 The hierarchical framework of literature ontology

5.3 作业子功能的设计与实现

在 E-learning 系统中，作业(Homework)流程主要包括作业查看(HomeworkBrowse)、作业提交 (HomeworkSubmit)、作业批阅 (HomeworkReview) 和作业布置 (HomeworkAssign) 四个过程。其中作业提交、作业查看的主要参与者为学生，作业批阅和作业布置的主要参与者为教师，且四个过程分别隶属于学习子平台和授课子平台。

5.3.1 问题描述

当某一用户登录 E-learning 系统，输入需求为 Homework 时，系统可以根据用户需求通过访问教学本体返回满足用户需求的四个作业子过程。同时，如果确定该用户为学

生用户，则只返回作业查看（HomeworkBrowse）和作业提交（HomeworkSubmit）两个子过程；如果该用户为教师，则返回作业批阅（HomeworkReview）和作业布置（HomeworkAssign）两个子过程。

5.3.2 功能设计与实现

在作业子功能设计实现阶段，主要包括作业本体建模和作业子过程语义 Web 服务构建两部分。

● 作业本体建模

在作业本体中，通过构建作业本体类（Homework）与作业子过程类（HomeworkBrowse、HomeworkSubmit、HomeworkReview、HomeworkAssign）之间的关系属性而建模其中的语义。描述如下：

Class (elearning: Homework)

ObjectProperty (hashomeworkAssigned domain (elearning: Homework) range (elearning: HomeworkAssign))

ObjectProperty (hashomeworkBrowsed domain (elearning: Homework) range (elearning: HomeworkBrowse))

ObjectProperty (hashomeworkReviewed domain (elearning: Homework) range (elearning: HomeworkReview))

ObjectProperty (hashomeworkSubmitted domain (elearning: Homework) range (elearning: HomeworkSubmit))

● 作业子过程语义 Web 服务构建

在本功能中，包括四个语义 Web 服务。下面以 HomeworkBrowse 为例加以介绍。

（1）确定服务的输入与输出

Sname = HomeworkDisplay

Input = {Student_ID: xsd: String, Student_name: xsd: String, Major(a), Advisor(b), Research_field(c), Student(d), $\exists$ have(d, a), $\exists$ instructedby(d, b), $\exists$ major_in(d, c)}

Output = {Homeworklist: xsd: String}

（2）对服务进行 OWL-S 语言建模

利用语义 Web 服务建模工具 protégé+OWL-S editor 对语义服务 HomeworkDisplay

建模，如图 5-6 所示。



图 5-4 语义 Web 服务 HomeworkDisplay

Fig. 5-4 The semantic web service HomeworkDisplay

5. 4 开发工具的选择

5. 4. 1 本体编辑工具-Protégé 简介

在本系统中，本体编辑工具选用的是美国 stanford 大学开发的本体编辑开发工具 Protégé 3.1。Protégé 最早源于 Mark Musen 在 1987 年建立的一个小程序，目的是为了了一些特定医学的系统建立一套知识获取系统^[76]。Protégé 在上世纪 90 年代后期改用 Java 语言开发，并且开放源代码，目前可运行在 Windows、Linux、Solaris 等多种平台上。归纳起来，Protégé 有以下几项突出的特点：

- (1) 它是一个交互式地进行本体设计和知识获取的图形化开发环境。它可以用于对描述某一特殊领域的本体的建模，并可以创建对应于特定领域数据的本体的实例。本体的开发者随时可以根据需要来获取相关的信息，并且可以使用树型的控制机制直观的控制浏览和管理一个知识库。
- (2) 支持各种标准的存储格式，例如关系型数据库和 XML, RDF, OWL 等等语言。
- (3) 提供了扩展接口，使用户可以依据自身的需求开发特殊用途的 plug-in，其官

方网站提供的 plug-in 主要分成 Tabs、Slot widgets、Storage models 和 Other 这几类。

- (4) 很多本体也采用 Protégé 开发，可以通过 <http://protege.stanford.edu/ontologies.html> 下载我们所需的本体。

Protégé 本身是一个三层的架构：最外层是用户接口层(user interface layer)，我们可以通过不同的 Tab 或 Slot 类型的 plug-ins 来改变用户接口；中间层则是知识模型层(knowledge model layer)，负责把知识储存在 Protégé 的运行时内存(runtime memory)中；最底层则是存储层(persistent storage layer)，通过映射的方式，负责把储存在知识模型中的知识以不同的格式储存起来。

5.4.2 本体应用 API-Jena 简介

Jena^[77]是一个基于 Java 语言的先进的语义网开发工具。它的第一个版本是 Jenal，发布于 2000 年，已经被下载了 10,000 多次。Jena2 修改了前一个版本的内部架构，又提供了一些新功能，于 2003 年 8 月发布。现在已经被下载 13,000 多次。

语义网推荐规范中的本体描述语言的核心是 RDF 图(Graph)，这是全球通用的数据结构。一个 RDF 图是由一组三元组(Predicate (P), Subject (S), Object (O))组成，P 是(S, O)的一个二元谓词关系。Jena2 同样是围绕 Graph 作为核心接口，然后来构建其它的组件。

Jenal 的主要贡献是为处理 RDF 图提供了丰富的用于 Model 类的 API。围绕着这些 API，Jenal 提供了大量的工具，包括对模型(Model)的多种方式的输入输出 I/O (RDF/XML, N3, N-triple)，RDF 查询语言 RDQL。使用这些 API，用户可以选择将 RDF 图存储在内存中或者是持久性存储（文件或数据库方式）。Jenal 还提供了附加的用于处理 DAML+OIL 数据的 API，但是 Jenal 并不支持 OWL。

后来，在很多用户的反馈中建议 Jena 更好的整合对 DAML+OIL 和对 RDF 的支持，以求达到更强的功能，例如存储 DAML 数据模型到数据库中。对于 Jenal 来说实现起来很困难，因为 Model 类的 API 已经很多了，无法再继续实现更多的 API。

为了解决这些问题，Jena2 采用了更好的系统架构。以求达到两个关键目标：

- 对于应用开发者可以提供多种灵活的表现 RDF 图的方案。这需要允许用户可以使用更高层接口或使用底层接口的不同方式访问处理 RDF 图数据。
- 提供一种 RDF 图的最简单的视图方式—三元组方式，主要便于系统级程序

开发人员处理数据。这对基于 RDFS 和 OWL 推理是非常有用的。

第一点实际上是建立在第二点的基础上的。两个关键目标都是为系统程序员提供扩展的功能点。RDF 图的表现方式不仅是现有的 Model API 而且是新的 Ontology API (用于 OWL, DAML+OIL, RDFS 等) 的基础。而在 RDF 图的表现层应该允许新三元组的开发, 包括存储在数据库或内存的三元组以及运算处理过程中动态产生的一些三元组 (比如推理过程中产生的数据源)。

Jena2 支持基于 RDFS 和 OWL 等语义推理。Jena 支持一种语义 Web 查询语言: RDQL。Jena2 还拥有一种表现层接口是 RDF Web API。它能提供 Web 客户端查询 RDF 图。这种基于 Web 查询的数据获取方式当然也可以为系统和应用程序员提供接口。这种方式有可能是 Jena 以后的发展方向。

目前, Jena2 支持本体描述语言 RDFS、DAML+OIL 和 OWL, 其推理机可以选择其内嵌的推理机, 也可导入其它本体推理机, 如 Racer、Pellet 等。

5.3.3 语义 Web 服务访问接口-OWL-S API 简介

OWL-S API 提供了一个 Java API 编程接口, 可以读取、执行和编写 OWL-S 服务描述。它支持读取不同版本的 OWL-S (包括 OWL-S 1.0、OWL-S 0.9、DAML-S 0.7) 的描述。这个 API 还提供了一个执行引擎 (ExecutionEngine), 既可以调用原子进程 (AtomicProcess), 还可以调用复合进程 (CompositeProcess)。原子进程带有 WSDL 或者 UPnP 基础 (UPnP groundings), 而复合进程可以使用控制结构, 如顺序、无序的和分开的 (Sequence, Unordered, 和 Split)。

OWL-S API 里的数据结构被设计成紧密符合 OWL-S 本体论的定义 (OWL-S 1.0)。当然, 随着 OWL-S 本体论的变化和一些关键问题 (如进程的定义和进程的出现, process definition and process occurrence 复合进程的数据流的规范) 的解决, OWL-S API 的最新版本也会发生相应的变化更新。

API 里面的所有对象都是继承 OWLResource 类, 因而具有 accessor 函数, 来获取 OWL 属性的值。也可能来获取底层的数据模型 (基于 Jena) 用于更高级的查询。

OWL-S API 的数据结构的设计非常吻合 OWL-S 本体, 尤其是 OWL-S 1.0。

Java 中的包、接口和方法的名称与 OWL-S 本体中的本体、类和属性的名称一致。

利用 OWL-S API 可以实现对语义 Web 服务的访问和操作。

5.5 系统运行结果及分析

5.5.1 文献检索子平台运行结果及分析

输入用户登录信息,进入文献检索子平台登录界面(图 5-5)。在该子平台中,包括基于语义查询子功能和基于关键字查询子功能。如 2.3 节所述,基于传统关键字技术,是从用户需求中提取关键字以寻找满足用户需求的知识及服务;基于语义技术,即通过捕捉用户需求及与用户需求相关的语义,寻找满足用户需求的知识及服务。

在文献检索子平台中,我们以主题为 Semantic Web 的用户查询请求为例对比关键字查询和语义查询的区别。如果选择关键字查询,则查询结果只返回具有 Semantic Web 关键字的文献,如图 5-6 所示;如果选择语义查询,则查询结果不仅返回具有 Semantic Web 关键字的文献,同时也要返回和 Semantic Web 同属计算机领域的相关文献,如图 5-7 所示。

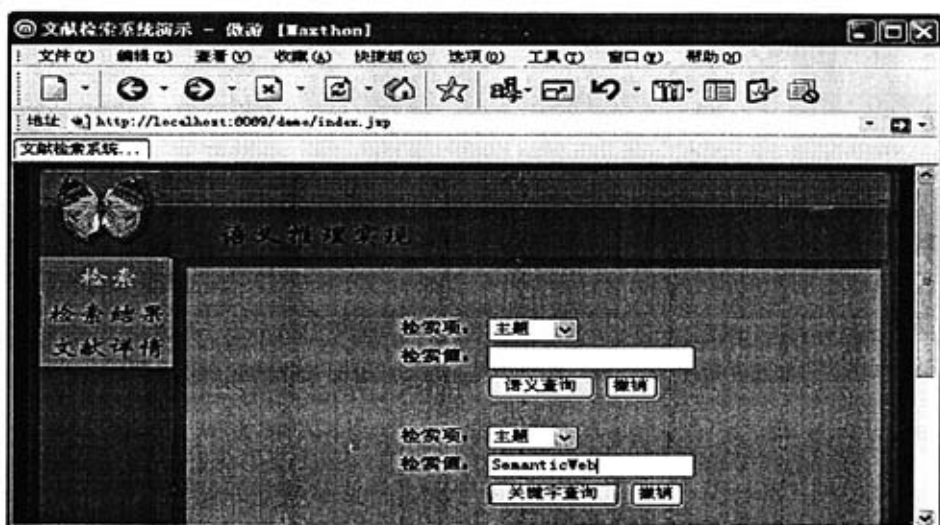


图 5-5 文献检索子平台登录界面

Fig. 5-5 The login interface of literature retrieval sub-Platform



图 5-6 基于关键字的文献检索结果

Fig. 5-6 The searching results of keyword-based literature retrieval



图 5-7 基于语义的文献检索结果

Fig. 5-7 The searching results of semantic-based literature retrieval

结果分析

通过对比图 5-6 与图 5-7 可以看出，基于语义的文献检索返回满足用户需求的目标文献要全于基于关键字的文献检索。因此，从语义角度出发，采用语义文献检索策略将会提高传统的基于关键字检索结果的查全率和准确率。

此外，本文献检索子平台在语义推理机实现部分，还自定义了用于发现文献之间互引用关系的推理规则，如下所示：

Rule1: [bothCiting: (?a http://www.domain2.com#Citing ?c), (?b http://www.domain2.com#Citing ?c), notEqual(?a, ?b) → (?a http://www.domain2.com#bothCiting ?b)]

Rule2: [bothCited: (?a http://www.domain2.com#Citing ?b), (?a http://www.domain2.com#Citing ?c), notEqual(?b, ?c) → (?b


```

http://www.domain2.com#bothCited ?c)]
Rule3:  [hasSameSubject:  (?a  http://www.domain2.com#hasSubject  ?c), (?b
http://www.domain2.com#hasSubject  ?c), notEqual (?a,  ?b)-> (?a
http://www.domain2.com#hasSameSubject ?b)]
    
```

通过采用上述推理规则，文献检索子平台在检索过程中可以自动发现文献之间的互引用关系，并将其返回给用户。

5.5.2 作业子模块运行结果及分析

输入用户登录信息，进入教学平台OBETP。如5.3.1节所述，进入用户服务请求登录界面（图5-8）。当用户输入Homework请求时，系统返回满足用户请求的四个子服务：HomeworkAssign, HomeworkReview, HomeworkSubmit和HomeworkBrowse。运行结果见图5-9。如果用户输入StudentHomework请求，系统则返回与学生相关的作业子服务：HomeworkSubmit和HomeworkBrowse。运行结果见图5-10。同理，如果用户输入TeacherHomework请求，系统则返回与教师相关的作业子服务：HomeworkAssign和HomeworkReview。运行结果见图5-11。



图5-8 用户服务请求登录界面

Fig. 5-8 The login interface of user service query

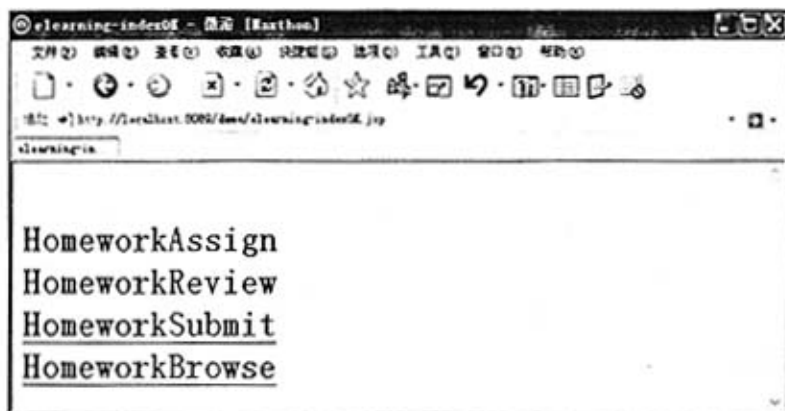


图 5-9 作业子服务列表

Fig. 5-9 The Homework services list

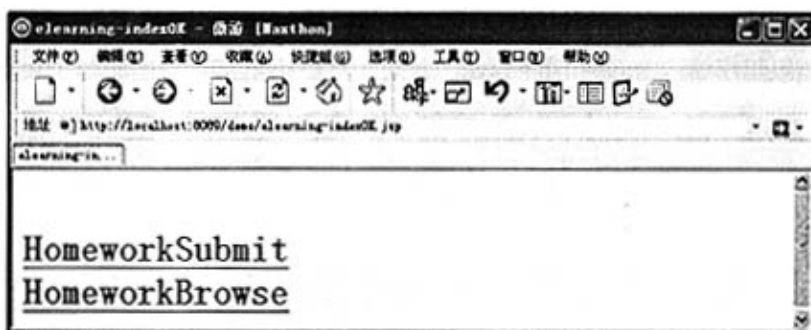


图 5-10 学生作业子服务列表

Fig. 5-10 The student Homework services list



图 5-11 教师作业子服务列表

Fig. 5-11 The teacher Homework services list

如图 5-10，访问子服务 HomeworkBrowse，运行结果见图 5-12，图 5-13。

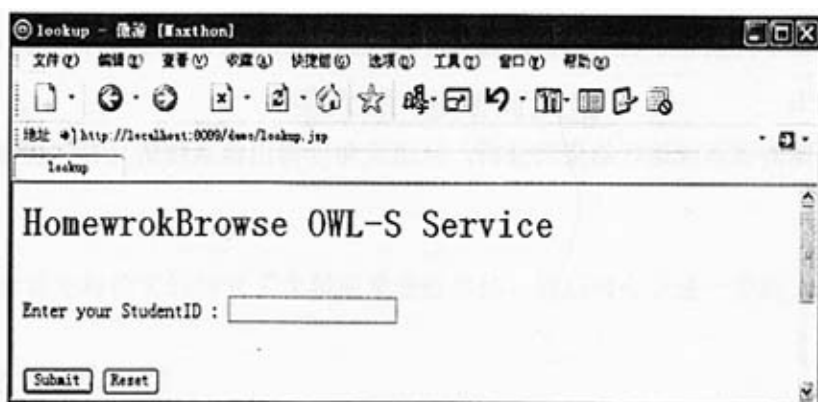


图 5-12 语义 Web 服务 Homework 访问界面

Fig. 5-12 The accessing interface of semantic web service Homework

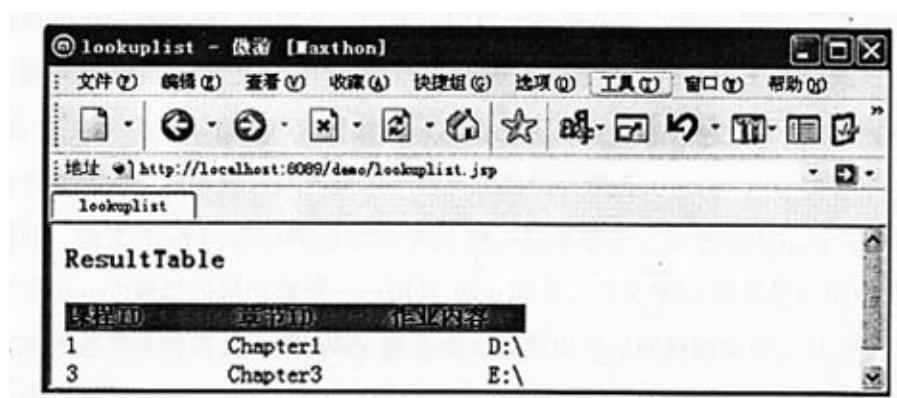


图 5-13 语义 Web 服务 Homework 运行结果

Fig. 5-13 The running results of semantic web service Homework

● OBETP 平台后台运行流程如下：

接受用户请求→访问作业本体库→根据语义关系，确定满足用户请求的子服务列表→返回子服务列表→在子服务列表中选择目标语义 Web 子服务→访问目标语义 Web 子服务→返回语义 Web 服务运行结果；

5.6 本章小结

本章以本体知识路由为背景，利用语义 Web 和元数据相关技术，设计和开发了一个基于本体的 E-learning 教学平台 OBETP。主要包括如下内容：

- (1) 详细介绍了该平台的设计思想和本体构建过程；
- (2) 重点介绍了文献检索子平台(SLRP)和作业功能模块的设计与实现过程；

- (3) 介绍了开发该系统所用到的相关开发工具: protégé 3.1, Jena API 和 OWL-S API;
- (4) 根据对该系统运行结果的分析, 从语义角度得出该系统所具有的优点。

第六章 总结与展望

本章首先对论文的研究工作和成果进行总结，然后对今后进一步的工作进行展望。

6.1 论文工作总结

传统 Web 通过利用可视化的标记语言如 HTML 对资源进行标识，提供了用户可以理解但机器不可理解的知识信息。因此，相对于机器而言，传统 Web 缺乏丰富的语义信息。而语义 Web 的诞生，弥补了传统 Web 缺乏语义的不足。语义 Web 通过本体技术为 Web 资源提供了丰富的可供机器处理的语义信息，使得机器可以利用这些语义元数据进行自动化的信息访问，从而提供一种高质量的新服务。

同时，语义 Web 研究的兴起也为 Web 服务技术带来了新的活力，这两种技术的结合产生了另一个新兴的研究课题——语义 Web 服务。语义 Web 服务是指用语义 Web 标记语言来描述服务的语义，使 Web 服务成为计算机可以理解的实体，从而支持服务的自动发现、执行和复合等。其中，服务的复合体现了 Web 服务的重要思想：未来的应用将由一组利用网络的服务复合而成。

本文正是在上述背景下，结合国家自然科学基金《知晓内容和环境的本体知识路由研究》，利用语义 Web、P2P 网络、本体、描述逻辑等相关技术，提出了一种用户需求驱动的、结合 P2P 网络和描述逻辑的本体知识路由机制，并以此为背景，对其中语义 Web 服务复合进行了更为深入的研究和探讨。完成的主要工作成果包括以下几方面：

- 将 P2P 技术和本体技术相结合，利用描述逻辑概念包含性的基本原理，采用动态虚拟社区的思想，结合传统 P2P 网络结构所具有的分布性和开放性等特点，提出了一种本体驱动的基于 P2P 网络结构和层次网络结构的本体知识路由机制，并详细讨论了知识路由算法的设计与实现。初步分析表明，该知识路由机制能够根据用户的知识需求，快速有效地发现、定位、评估和返回目标知识；
- 针对原有的 OWL-S 模型并未考虑用户的需求偏好信息的局限性，提出了一种将用户需求偏好信息考虑在内的扩展 XOWL-S 模型；
- 将描述逻辑所具有的良好知识表示能力和强大推理能力与 AI 规划所具有的行

为变化建模能力相结合,以扩展 XOWL-S 模型为基础,提出一种基于描述逻辑的语义 Web 服务动态复合机制 DLAIP,并对其进行了形式化证明。结果表明,该机制不仅具有良好的知识表示能力和强大推理能力,可以弥补当前本体语言 OWL、OWL-S 对服务复合动态过程建模能力的不足,而且还可以通过利用 AI 规划方法解决服务复合时所具有的效率高的优点来弥补直接利用 OWL-S 语义进行服务复合效率低的缺点。因此,采用该机制,将会有助于提高语义 Web 上服务复合的效率和准确率;

- 设计和开发了基于本体的 E-learning 教学平台 OBETP (Ontology Based E-learning Teaching Platform),实现了语义技术在 E-learning 领域的应用。

6.2 进一步的展望

本文围绕语义 Web 及语义 Web 服务复合等问题,对涉及到的本体知识路由、扩展的 OWL-S 模型、基于描述逻辑的语义 Web 服务动态复合等内容作了一些探索性的工作,并通过逻辑证明和实验原型系统的开发与设计初步证明了本文研究成果的有效性。但是,由于时间等原因,还有以下相关问题尚未深入研究,留待将来进一步研究:

- 本体知识路由算法的效率:在本文中,提出了较为系统的本体知识路由算法,但并未在大规模网络环境下测试知识路由算法的效率和有效性。在下一步的研究中,可以通过分析知识路由中的并行特性,对提出的串行路由算法进行并行化处理,并在大规模网络环境下测试其效率和有效性;
- 服务的自动发现和执行:服务的自动发现和执行是语义 Web 服务的另一个研究课题,其研究成果也可以集成到服务复合的研究中来。在这方面,需要研究如何根据用户的输入信息和服务的 Grounding 以及 WSDL 描述,来构造具体协议(例如 SOAP)-下的服务调用消息,与服务交互。
- 循环 TBox 知识库中 DLAIP 的有效性证明:在本文中,提出采用固定点语义的思想证明 DLAIP 的有效性,但并未对具体证明过程和技术细节作进一步的探讨,如何处理循环定义的 TBox 知识库中的服务复合及其推理证明也将是下一步研究工作的重点;
- 实验原型系统 OBETP 的完善:在本文 5.1 节 OBETP 的设计中,该平台包括考试子平台(EP)、学习子平台(LP)、授课子平台(TP)和文献检索子平台(SLRP)四

个平台，本文只实现了其中的文献检索子平台(SLRP)及作业子功能模块，其余子平台的实现也是下一步研究工作的重点。

参考文献

- [1] S McIlraith, T Son, Adapting golog for composition of semantic web services, In Proc of the Eighth International Conference on Knowledge Representation and Reasoning. Toulouse, France, 2002
- [2] 王丰锦, 基于语义 Web 服务的动态服务组合技术的研究: (博士学位论文), 北京: 中国科学院计算技术研究所, 2003
- [3] T B Lee, J Hendler, O Lassila, The semantic web, Scientific American 2001(5): 34-43
- [4] S A McIlraith et al., Semantic web services, IEEE Intelligent Systems, 2001(2): 46-53
- [5] S A McIlraith et al., Mobilizing the Semantic Web with DAML-Enabled Web Services, In: Proc. of SemWeb'2001, Hong Kong, China, 2001:82-87
- [6] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors, The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2003
- [7] 王淞昕, 王飞, 周水庚, 周傲英, N-SHOQ(D): 描述逻辑 SHOQ(D)的一个非单调扩展, 计算机研究与发展, 42 (4): 570~575, 2005
- [8] T B Lee, XML and the web, Presentation at XML World 2000, 2000
<http://www.w3.org/2000/Talks/0906-xmlweb-tbl/slide9-6.html>
- [9] S A McIlraith et al., Semantic web services, IEEE Intelligent Systems, 2001(2): 46-53
- [10] J Peer. Bringing together semantic web and web services. In Proc. of the First International Semantic Web Conference, Sardinia, Italy, 2002: 279-291
- [11] 邓志红, 唐世渭, 张铭等, Ontology 研究综述, 北京大学学报(自然科学版), 2002, 38(5): 730~738
- [12] 柴晓路, 梁宇奇, Web Services 技术、架构和应用, 北京:电子工业出版社, 2003
- [13] 李慧, 语义 web 服务发现机制的研究 [学位论文], 江苏, 江苏大学, 2005
- [14] 张超, 基于语义 Web 的 Web 服务发现机制研究[学位论文], 重庆, 重庆大学, 2004
- [15] P F Patel-Schneider, D Fensel, Layering the semantic webs problems and directions, In:

- Proc. of the First International Semantic Web Conference, Berlin Heidelberg: Springer-Verlag, 2002: 16-29
- [16] T.Berners-Lee, RFC2396, <http://WWW.faqs.org/rfcs/rfc2396.html>
- [17] Michel Klein XML, RDF and relatives, New York:IEEE Intelligent Systems, 2001, 16(2): 30~38
- [18] S A McIlraith et al., Semantic web 2001(2): 46-53. Seances. IEEE Intelligent Systems, 2001(2): 46-53
- [19] J Peer. Bringing together semantic web and web services. In Proc. of the First International Semantic Web Conference, Sardinia, Italy, 2002: 279-291
- [20] J Heflin, Towards the semantic web: knowledge representation in a dynamic, distributed environment: [Ph D Dissertation], College Park, IV1D: Maryland University, 2001
- [21] J Heflin, J Hendler, A portrait of the web in action. IEEE Intelligent Systems, 2001(2): 54-59
- [22] Ontobroker Project, <http://ontobroker.aifb.uni-karlsruhe.de/index ob.html>
- [23] On-To-Knowledge Project, <http://www.ontoknowledge.org>
- [24] D Fensel et al., OIL: An Ontology infrastructure for the semantic web, IEEE Intelligent Systems, 2001(2): 38-45
- [25] J Hendler and D L McGuinness, DARPA Agent Markup Language, IEEE Intelligent Systems, 2001(6): 72-73
- [26] DAML Services, <http://www.daml.org/serviceslowl-s>
- [27] D Martin, et al. The OWL Services Coalition, OWL-S 1.0 Release, 2403: <http://www.daml.org/services/owl-s/1.0/>.
- [28] T Sollazzo et al., Semantic web towards the service architecture-evolving web service standards towards the semantic web, 2002:<http://www.aifb.un-karlsruhe.de/WHS/sst/Research/Publications/sub-flairs2002.pdf>
- [29] F Casati, et al., Adaptive and dynamic service composition in eFlow, In Proc. of the International Conference on Advanced Information Systems Engineering, Stockholm, Sweden, 2000
- [30] S E Czerwinski et al., An architecture for a secure service discovery service, In Proc. of the 5th Annual International Conference on Mobile Computing and Networking. Seattle,

MA ACM Press. 1999

- [31] J Meng, et al., Achieving Dynamic Inter-Organizational Workflow Management by Integrating Business Processes, Events, and Rules, In: Proc. of the 35th Hawaii International Conference on System Sciences, Hawaii, and USA, 2002
- [32] Q Z sheng, et al., SELF-SERV: a platform for repid composition of web services in a peer-to-peer environment, In Demo Session of the 28th International Conference on Very Large Databases, 2002
- [33] S Narayanan, S McIlraith, Simulation, verification and automated composition of web services, In: Proc of the 11th Internatiflnal World Wide Web Conference, Honolulu, Hawaii, USA, 2002
- [34] E Sirin, J Hendler, B Parsia, Semi-automatic composition of web services using semantic descriptions, In: Proc of Web Services: Modeling; Architecture and Infrastructure workshop in conjunction with ICEIS2005, 2005
- [35] F. Casati, S. Ilnicki, and L. Jin, Adaptive and dynamic service composition in EFlow, In Proceedings of 12th International Conference on Advanced Information Systems Engineering (CAiSE), Stockholm, Sweden, June 2000. Springer Verlag
- [36] H. Schuster, D. Georgakopoulos, A. Cichocki, and D. Baker, Modeling and composing service-based and reference process-based multi-enterprise processes, In Proceeding of 12th International Conference on Advanced Information Systems En-gineering (CAiSE), Stockholm, Sweden, June 2000. Springer Verlag
- [37] S. McIlraith and T. C. Son, Adapting Golog for composition of Semantic Web services, In Proceedings of the 8th International Conference on Knowledge Repr-esentation and Reasoning (KR2002), Toulouse, France, April 2002
- [38] S. R. Ponnekanti and A. Fox, SWORD: A developer toolkit for Web service composition, In Proceedings of the 11th World Wide Web Conference, Honolulu, HI, USA, 2002
- [39] 陆入铃, 研究知识科学, 发展知识工程, 推进知识产业, 信息技术快报 2003 年, 第 6 期
- [40] Preeda Rajasekaran, John Miller, Kunal Verma, Amit Sheth, Enhancing Web Services Description and Discovery to Facilitate Composition, 2004
- [41] Jose Saias and Paulo Quaresma, Semantic enrichment of a web legal information

- retrieval system, In T. Bench-Capon, A. Daskalopulu, and R. Winkels, editors, JURIX'2002 - Fifteenth Annual International Conference on Legal Knowledge and Information Systems, volume 89 of Frontiers in AI and Applications, pp 11-20, London, UK, Dezember 2002. IOS Press
- [42] Dominik Heckmann, Tim Schwartz, GUMO-the General User Model Ontology, Germany, 2005
- [43] Rudi Belotti, Corsin Decurtins, A Context-Aware Component Registry for Ubiquitous and Mobile Applilcations, 2004
- [44] Mithun Sheshagiri, Norman M. Sadeh, Using Semantic Web Services for Context-Aware Mobile Applications, 2004
- [45] R. Guha and R. McCool, TAP: a semantic web platform, Computer Networks, 42(5): 557-577, August 2003
- [46] A. Crespo and H. Garcia-Molina, Routing indices for peer-to-peer systems, In Proceedings of the 22nd nternational Conference on Distributed Computing Systems, pages 23.32. IEEE Press, 2002
- [47] M. Schlosser et al, A scalable and ontology-based P2P infrastructure for Semantic Web Services, In P2P-2002-Proceedings of the 2nd Int. Conf. on Peer-to-Peer Computing, pages 104.111. IEEE Press, 2002
- [48] Keeney, J., Lynch, D., Lewis, D., O'Sullivan, D., "On the Role of Ontological Semantics in Routing Contextual Knowledge in Highly Distributed Autonomic Systems", Technical Report (TCD-CS-2006-15), Department of Computer Science, Trinity College Dublin. 2006
- [49] Antonio Carzaniga, Alexander Wolf, Forwarding in a content-based network[C], In: Proc of the ACM SIGCOMM 2003, New York: ACM Press, 2003.163-174
- [50] Oram, editor, "Peer-to-peer: Harnessing the power of disruptive technologies," O'Reilly&Associates, March 2001.avaitable at: <http://www.oreilly.de/catalog/peertopeer/>
- [51] <http://www.peer-to-peer.org>
- [52] 王丹, 魏红 P2P 模式的系统结构研究沈阳航空工业学院学报, 2003. 6
- [53] S. Castano et al, Ontology-addressable contents in p2p networks, In Proc. of WWW'03 1st SemPGRIDWorkshop, 2003

- [54] W. Nejdl et al, EDUTELLA: A p2p networking infrastructure based on rdf. In Proc. of the International World Wide Web Conference 2002 (WWW2002), pp. 604–615, Hawaii, USA, (2002)
- [55] T.Berners-Lee, J. Hendler, and O. Lassila, 'The semantic web', Scientific American, (284), 35–43, (May 2001)
- [56] Yu Xing, Yingjie Li, Xueli Yu, Using Semantic Matching, Research on Semantic Web Services Composition, P.S. Szczepaniak et al. (Eds.): AWIC 2005, LNAI 3528, pp. 445–450, 2005, Springer-Verlag. 2005
- [57] 行宇, 基于本体的语义 Web 服务发现机制的研究 (硕士学位论文), 太原理工大学, 2006
- [58] Alexander Maedche, Steffen Staab, Measuring Similarity between Ontologies, In Proc. of the European Conference on Knowledge Engineering and Knowledge Management (EKAW), Springer Verlag, 2002, 251-263
- [59] Erhard, R., Bernstein, P.A. (2001), A survey of approaches to automatic schema matching, VLDB Journal, 10, 334-350
- [60] Syed Sibte, Raza Abidi, Xiaolin Pang, Knowledge Sharing Over P2P Knowledge Networks: A Peer Ontology and Semantic Overlay Driven Approach, 2004
- [61] Ganter, B., Stumme, G. Formal concept analysis: Methods and Applications in Computer Science, TU Dresden
- [62] Konrad Pfadenhauer, Schahram Dustdar, Burkhard Kittl, Challenges and Solutions for Model Driven Web Service Composition, 2005
- [63] R.J. Brachman and J. Schmolze, An Overview of the KL-ONE Knowledge Representation System, Cognitive Sci 9(2), 1985
- [64] B. Nebel, Terminological reasoning is inherently intractable. AIJ, 43:235-249, 1990
- [65] F. Baader, I. Horrocks, and U. Sattler, Description logics as ontology languages for the semantic web, In Festschrift in honor of J org Siekmann, LNAI. Springer-Verlag. 2003
- [66] The OWL-S Coalition, OWL-S 1.1 draft release, 2004.
<http://www.daml.org/services/owl-s/1.1/>
- [67] M K Smith, Web Ontology issue status, 2003: <http://www.w3.org/2001/swiWebOnt/webont-issues.html#H4.4-Extra-logical-feature-set>

- [68] R. Reiter, *Knowledge in Action*, MIT Press, 2001
- [69] F. Baader, M. Milicic, C. Lutz, U. Sattler, and F. Wolter, Integrating description logics and action formalisms for reasoning about web services, LTCS-Report LTCS-05-02, Chair for Automata Theory, Institute for Theoretical Computer Science, TU Dresden, Germany, 2005
- [70] S. McIlraith and T. C. Son, Adapting Golog for composition of Semantic Web services, In Proceedings of the 8th International Conference on Knowledge Representation and Reasoning (KR2002), Toulouse, France, April 2002
- [71] S. R. Ponnekanti and A. Fox, SWORD: A developer toolkit for Web service composition, In Proceedings of the 11th World Wide Web Conference, Honolulu, HI, USA, 2002
- [72] E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau, HTN planning for web service composition using SHOP2, *Journal of Web Semantics*, 1(4): 377-396, 2004
- [73] D. S. Nau, T. C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman, SHOP2: An HTN planning system, *JAIR*, 20:379-404, 2003
- [74] Volker Haarslev and Ralf Moller, Description of the RACER system and its applications, In Proc. of the 2001 Description Logic Workshop (DL 2001), pages 132-141. CEUR Electronic Workshop Proceedings, <http://ceur-ws.org/Vol-49/>, 2001
- [75] Dan Wu, Bijan Parsia, Evren Sirin, James Hendler, and Dana Nau, Automating DAML-S Web Services Composition Using SHOP2, In Proc. 2nd International Semantic Web Conference (ISWC2003), Sanibel Island, Florida, USA, October 20-23, 2003
- [76] Protégé 2000, <http://protege.Stanford.edu>
- [77] Jena 2.0, <http://jena.sourceforge.net/ontology/index.html>

致 谢

在我论文完成之际，衷心感谢所有关心，帮助过我的老师，同学，朋友和亲人们。

首先感谢导师余雪丽教授在三年学习历程中对我的指导和帮助，余老师学术上的谆谆教诲和热情鼓励使我得以顺利完成硕士学业和研究课题。学高为师，身正为范。余老师具有渊博的学识和长者风度，对工作兢兢业业，做学问一丝不苟，在学习和生活中都给予我极大的帮助，余老师对待工作，对待研究的态度是我一生学习的榜样。

感谢王莉副教授在我的论文选题，构思，研究过程中给予了我许多有益的指导和建议，从而使我的论文能够顺利完成。感谢谢红薇老师、冯秀芳老师、孙静宇老师、吴兴老师，他们平易近人，对我的学习和研究工作都给予了无私的帮助。她们积极向上的精神，认真负责的工作态度为我以后的工作树立了榜样。

衷心感谢同窗学友耿丽丽，师弟王睿、张勇键、梁弈德，师妹李建林、王亚彬。共同作为一个研究小组的成员，在学习上他们自始至终都给了我极大的支持和帮助。感谢同窗学友李三燕、牛瑞萍、王新奇和何飞，他们在平时学习中独特的见解同样给了我极大的鼓舞；同时我还要深深感谢实验室的同学以及同班同学对我的帮助，他们总是在我思路停顿时带给我欢乐，由衷地感谢你们！

身在异乡为异客，父母多年来默默地付出，给予了我无私的爱，感谢他们在我学习期间给我的鼓励，使我顺利完成学业。

最后，对本文所有被引用文献的作者表示感谢，也感谢各位专家和教授对本文的评阅和赐教。

攻读学位期间发表的学术论文目录

- [1] Yingjie Li, Xueli Yu, Lili Geng, Li Wang. Research on Reasoning of the Dynamic Semantic Web Services Composition. 已被国际期刊 Web Intelligence and Agent System (EI 源刊) 录用. 2007
- [2] Yingjie Li, Xueli Yu, Lili Geng, Li Wang. Research on Reasoning of the Dynamic Semantic Web Services Composition. The 2006 IEEE / WIC / ACM International Conference on Web Intelligence, IEEE Computer Society. 2006.
- [3] Yingjie Li, Li Wang, Xueli Yu, Wen Li, Yu Xing. DLAIP: A Description Logic Based Approach for Dynamic Semantic Web Services Composition. Advances in Web Intelligence: Forth International Atlantic Web Intelligence Conference, AWIC 2006 proceedings. Studies in Computational Intelligence Series, Springer-Verlag. 2006.
- [4] 李英杰, 余雪丽, 王睿. 基于描述逻辑的语义 Web 服务动态复合推理和验证. 第一届中国 web 智能学术: CRSSC-CWI-CGrC'2007, 广西师范大学学报. 2007
- [5] 李英杰, 王莉, 余雪丽. 本体驱动的语义 Web 服务动态复合的研究. 计算机工程与应用. 2007
- [6] 李英杰, 王莉, 余雪丽. 本体驱动的知晓内容和上下文的知识路由研究. 计算机工程与应用. 2006
- [7] Yingjie Li, Xueli Yu, Yu Xing. Research on Content & Context-awareness of Knowledge Routing in Knowledge Communication. 6th International Symposium On Test and Measurement, Vols 1-9, Conference proceedings: 1750-1753. 2005.
- [8] Yu Xing, Yingjie Li, Xueli Yu. Using Semantic Matching, Research on Semantic Web Services Composition. Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), v 3528, Advances in Web Intelligence: Third International Atlantic Web Intelligence Conference, AWIC 2005 proceedings. P 445-450. Springer-Verlag. 2005.
- [9] Li Wang, Yingjie Li, Wen Li, Yu Xing, Xinqi Wang, Xueli Yu. The Semantic Matching of the Semantic Web Services. The 2004 IEEE / WIC / ACM Workshop: KGGI 2004.

2004.

- [10] Li Wang, Wenli Wu, Yingjie Li, Xueli Yu. Content-aware Trust Statement for Semantic Grid. The 2nd International Conference on Semantics, Knowledge and Grid (SKG2006). 2006.
- [11] Rui Wang, Xueli Yu, Yingjie Li. Behavior Analysis Based Automatic Composition of Semantic Web Services. Advances in Web Intelligence: Fifth International Atlantic Web Intelligence Conference, AWIC 2007 proceedings. Studies in Computational Intelligence, Springer-Verlag. 2007.