



独创性声明

本人声明,所提交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知,除了文中特别加以标注和致谢的地方外,论文中不包含其他人已经发表或撰写过的研究成果,也不包含为获得武汉理工大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签名: 张啸军 日期: 2011.5

学位论文使用授权书

本人完全了解武汉理工大学有关保留、使用学位论文的规定,即:学校有权保留并向国家有关部门或机构送交论文的复印件和电子版,允许论文被查阅和借阅。本人授权武汉理工大学可以将本学位论文的全部内容编入有关数据库进行检索,可以采用影印、缩印或其他复制手段保存或汇编本学位论文。同时授权经武汉理工大学认可的国家有关机构或论文数据库使用或收录本学位论文,并向社会公众提供信息服务。

(保密的论文在解密后应遵守此规定)

研究生(签名): 张啸军 导师(签名): 高岩 日期: 2011.5

摘要

随着互联网技术应用的迅速发展及可视化技术在航空、气象、海洋、医学、地质和商业领域广泛应用, 可视化技术需要处理的数据量会越来越大, 而且数据将呈现出分布性, 动态性和异构性的特点。基于当前的计算机软硬件性能的限制, 及日渐复杂的计算模型和不断增长的数据规模, 传统的可视化技术已经不能很好的满足人们各项科学研究的需要了。然而 web 服务技术和网格技术的出现, 为分布式可视化技术带来了新的发展契机。因为网格技术能够很好的实现各种异构系统之间的信息、数据、知识和资源的共享; 而 Web 服务技术是解决各种异构平台之间互操作的重要技术。随着分布式可视化技术的发展, Web 上出现了越来越多的可视化服务资源, 面对数量庞大的服务群, 如何管理并发现满足需要的可视化服务是有效地利用各种可视化服务资源的前提与基础, 而可视化 Web 服务匹配是解决这个问题的第一步也是最重要的一步。可视化 Web 服务匹配是 web 服务匹配在可视化领域的特定应用, 它既具有一般服务匹配的共性, 又具有可视化技术本身的鲜明个性。因此, 研究可视化 Web 服务发布与匹配机制, 实现可视化 Web 服务发布与匹配原型系统将具有重要的理论和现实意义, 它将为可视化服务提供者提供服务发布、查找和管理的平台, 实现可视化服务在更广范围内的共享, 同时也帮助可视化服务用户准确的发现自己需要的服务资源。

目前, 有关可视化服务发布与匹配的研究还鲜见报道。面对这样的一个研究现状, 本文通过在深入研究可视化技术及 Web 服务匹配方法的基础上, 对如何更好的实现可视化 Web 服务发布与匹配作出了初步的探索, 主要的工作如下:

1、可视化 Web 服务的语义描述: 论文通过对可视化技术、语义 Web 服务描述语言 OWL-S 和本体描述语言 OWL 进行深入研究, 设计出了可视化 Web 服务描述模型, 并构建了可视化服务本体库。它们将规范可视化 Web 服务的描述, 并在一定程度上消除了服务描述的异构问题, 为可视化服务发布与匹配奠定坚实的基础。

2、可视化 Web 服务发布与匹配机制研究: 论文通过对可视化技术的研究, 发现可视化技术具有过程鲜明性特点, 因此设计了按可视化过程分类的服务发布方法; 同时通过对现有语义距离相似度计算方法的深入研究, 在综合考虑概

念二元关系及概念在概念树中深度对相似度计算的影响，提出了一种改进的基于语义距离的相似度算法；在该改进算法的基础上，结合可视化技术自身特点，设计出了一种基于功能语义的可视 Web 服务分类别、多阶段的匹配方法，并针对各阶段的差异，设计了相应的匹配算法。

3、可视化 Web 服务发布与匹配模型的设计与实现：为了更好地实现可视化 Web 服务资源的共享及管理，笔者利用对可视化 Web 服务发布与匹配机制研究的成果，结合服务资源管理的需要，设计出了可视化 Web 服务的发布与匹配模型，并采用 J2EE 技术实现了原型系统，它使用户可以方便的实现可视化 Web 服务资源的发布、查询、管理和共享，同时原型系统还具有良好的可扩展性，便于升级。

4、原型系统的性能评估：为了验证可视化 Web 服务发布与匹配原型系统的可行性及有效性。笔者通过构建一些可视化 Web 服务实例，对原型系统进行了服务发布与服务查询的测试，并采用查全率、查准率和响应时间对测试结果进行了定量的分析。

本论文受武汉市国际合作项目：“基于语义的可视化服务发布与匹配平台的研究与实现”（项目编号：200970634269）的资助。

关键词：可视化，Web 服务，Web 服务匹配，本体，OWL-S

Abstract

With the rapid development of Internet technology, the visualization technology is being widely applied into a lot of fields, such as aviation, meteorology, oceanography, medicine, business process and so on, the data, which needs to be visualized, are also becoming larger and more distributed, autonomous, heterogeneous and dynamic. In the current restrictions on computer hardware and software performance, and the increasing complex computing model and the expansion of the data scale, the traditional visualization mode can't well meet the needs of the scientific research. But Web Service technology and grid computing represents one of the most promising advancements for today distributed visualization. The Grid can realize the sharing of information between the heterogeneous systems, the information, such as data, knowledge resources and so on. And the web service technology can allow interoperability between heterogeneous platforms. As the development of distributed visualization, there are more and more visualization services. Facing so many visualization services, how to find the suitable visualization web services is the very important for the effective use of visualization web services resources, and visualization web service matching is the key step to solve this problem. As a specific application in visualization domain, visualization web services matching has the common characters of general service matching technologies, and distinct specific characters of visualization technology itself as well. So researching the way of visualization web services' publishing and matching, and developing the system for visualization web services' publishing and matching will have very necessary and meaningful. This system will allow users to publish and search visualization web services easily, and make user share the visualization web services conveniently.

Nowadays, there are still no reports about visualization web service matching. In response to this situation, we commit an initial exploration on how to publish and match services in visualization domain. Our works and achievements include as following:

1. The semantic descriptions of visualization web service

After researching the visualization technology, semantic web service description language OWL-S and ontology language OWL, the thesis designs a service description model for the visualization web service, and build a library of visualization services. It will regulate the service description for visualization web service, and can eliminate the problem of service description heterogeneous.

2. Visualization web service publishing and matching

By studying the characteristics of visualization technology, the thesis design a way for publishing visualization web service, which is according to visualization process. At the same time, the thesis researchs the existing similarity measure which bases on semantic distance-depth. With considering the concept's binary relations and concept depth, it propos an improved distance-based semantic similarity algorithm. Baseing on this improved algorithm and considering the characteristics of visualization techniques, the thesis designs a visualization web service matching algorithm which is base on functional semantic and has multi-level matching.

3. The design and developing of visualization web service publishing and matching model.

In order to manage and share the visualization web service well, the thesis design a Visualization web service publishing and matching model which bases on the results of researching the visualization web service's publishing and matching. And we develops a model system bases on that visualization web service publishing and matching model.

4. System performance evaluation

The thesis constructs many visualization web service to authentic the feasibility and effectiveness of the visualization web services publishing and matching model system, and analysis the test results by recall, precision and response time.

This work is funded by Wuhan international cooperation project, "Study and Implementation on Semantic-based visualization services Publishing and Matching Platform" (Project number: 200970634269).

Keywords: Visualization, Web Service, Web Service Matching, Ontology, OWL-S,

目 录

第 1 章 绪论	1
1.1 研究背景与意义.....	1
1.2 国内外研究现状与存在问题分析.....	3
1.3 研究的内容及目的.....	5
1.4 论文的组织结构.....	7
1.5 本章小结.....	8
第 2 章 Web 服务及语义 Web 服务	9
2.1 引言.....	9
2.2 Web 服务技术.....	9
2.2.1 Web 服务的定义.....	9
2.2.2 Web 服务的体系结构.....	9
2.3 语义 Web.....	10
2.4 语义 Web 服务.....	13
2.4.1 语义 Web 服务概述.....	13
2.4.2 OWL-S 描述语言.....	13
2.5 本章小结.....	16
第 3 章 可视化技术与本体	17
3.1 引言.....	17
3.2 可视化概述.....	17
3.2.1 可视化技术研究背景.....	17
3.2.2 可视化技术的应用领域.....	18
3.2.3 可视化技术的研究内容.....	19
3.2.4 可视化技术研究的意义.....	20
3.3 本体.....	20
3.3.1 本体论.....	20
3.3.2 本体定义.....	21
3.3.3 本体构成.....	21
3.3.4 本体的描述语言 OWL.....	22
3.4 可视化本体的研究与实现.....	22
3.4.1 可视化本体的研究现状.....	22

3.4.2 可视化本体的构建规则	26
3.4.3 可视化本体的建模	26
3.4.4 可视化本体的实现	29
3.4.5 可视化本体的一致性检测	31
3.5 本章小结	31
第 4 章 基于功能语义的可视化 Web 服务匹配算法的研究	32
4.1 引言	32
4.2 语义 Web 服务匹配算法的研究现状	32
4.3 改进的基于语义距离的相似度算法	35
4.3.1 算法思想	35
4.3.2 算法介绍	36
4.3.3 分配权重	37
4.3.4 构造概念节点路径表	38
4.3.5 计算语义距离	38
4.3.6 概念相似度计算	39
4.4 可视化 Web 服务的描述及发布	40
4.4.1 可视化 Web 服务的描述	40
4.4.2 可视化 Web 服务的发布	41
4.5 基于功能语义的可视化 Web 服务匹配的总体目标	42
4.6 基于功能语义的可视化 Web 服务匹配算法	42
4.6.1 Service Category 匹配	42
4.6.2 textDescription 匹配	43
4.6.3 DataModel 匹配	44
4.6.4 IO 匹配	45
4.6.5 可视化 web 服务综合匹配	45
4.7 本章小结	46
第 5 章 可视化 Web 服务发布与匹配模型的设计与实现	47
5.1 引言	47
5.2 设计思想	47
5.3 模型层次设计	48
5.3.1 模型总体结构	48
5.3.2 数据库层	49
5.3.3 注册/匹配层	49
5.3.4 信息抽取层	50

5.3.5 用户接口层	50
5.4 可视化 Web 服务发布与匹配流程设计	51
5.4.1 可视化 Web 服务发布流程	51
5.4.2 可视化 Web 服务匹配流程	52
5.5 模型的实现	54
5.5.1 实现概述	54
5.5.2 开发环境的搭建	54
5.5.3 数据库设计	55
5.5.4 OWL—S 解析器的实现	56
5.5.5 本体推理机的实现	57
5.5.6 服务发布的实现	58
5.5.7 服务查询的实现	60
5.6 本章小结	61
第 6 章 实验测试与结果分析	62
6.1 引言	62
6.2 测试环境搭建	62
6.3 测试过程	63
6.3.1 可视化 Web 服务实例的构建	63
6.3.2 测试方案	64
6.4 测试结果及分析	64
6.4.1 测试结果	64
6.4.2 结果分析	66
6.5 本章小结	68
第 7 章 总结与展望	69
7.1 工作总结	69
7.2 工作展望	70
致 谢	71
参考文献	72
在校攻读硕士学位期间公开发表论文	75

第 1 章 绪论

1.1 研究背景与意义

可视化 (Visualization) 是利用计算机图形学和图像处理技术, 将数据转换成图形或图像在屏幕上显示出来, 并进行交互处理的理论、方法和技术。它涉足到计算机图形学、计算机辅助设计、图像处理、计算机视觉等诸多领域^[1]。是一项对数据处理、数据表示、及决策分析进行研究的综合技术。可视化技术是在社会的巨大需求和技术水平极大进步的深刻历史背景下出现的。早在 20 世纪初期, 图表和统计等原始的可视化技术已经被人们应用到科学数据的分析当中。计算机的诞生和普及, 将人类社会推向了一个崭新的信息时代, 目前所进行的科学研究的目的已不再是单纯的为了获得数据, 而是为了能够挖掘数据信息背后的内在规律, 更好的探索自然规律。随着科学研究的不断深入, 及处理数据规模的不断膨胀, 传统的可视化技术已经无法满足人们的需求。不过随着人类制造工艺水平、材料科学的发展进步, 计算机软硬件性能的不断提高, 把计算机技术引入可视化领域, 将大大加速可视化技术的发展。可视化主要包括两个方面的技术: 信息可视化技术和科学可视化技术, 而后者是近年来可视化学者和专家研究的重点、热点。“科学可视化”一词是在 1987 年 2 月美国国家科学基金会 (National Science Foundation, 简称 NSF) 主办的一个专题研讨会上被提出来的, 而且在会议上还讨论了“科学可视化”的研究内容, 及未来的发展方向等。此后, 有很多的国家都开始逐渐重视, 并加大对该新兴技术的研究力度, 这也意味着“科学可视化”这门新兴学科在国际范围内已经开始成熟起来。当前它的应用领域已经非常广泛, 例如工程计算, 地理, 生物, 考古和医学等, 而且都取得了了很大的成绩。

但是, 随着电子计算硬件产品的不断高速发展, 各种更高性能计算机不断出现, 各种软件技术不断产生, 人们科学研究中能获得的数据将越来越大。同时因为可视化技术主要就是用于对数据的处理, 所以这样势必大大的增加了可视化过程中需要处理的数据量, 这将是可视化技术发展必然面临的一个重大问题。当前, 在可视化应用中需要处理大规模数据的领域主要有医学影像, 航天通信数据, 超级计算机输出数据及地理勘测数据等。比如, 现在美国宇航局每

天大概要通过卫星获取 3-4TB 的空间数据信息，而且为方便对这些空间数据信息进行后期的处理研究还需要长期的存储工作^[2]。而且随着可视化技术在各领域的进一步深入广泛的应用，可视化技术需要处理的数据量会越来越大，而且数据将呈现出分布性，动态性和异构性的特点。基于当前的计算机电子产品的软硬件性能的限制，及日渐复杂的计算模型和不断增长的数据规模，传统的可视化技术已经不能很好的满足人们各项科学研究的需要了。所以可视化技术必须利用分布式技术的特点，充分挖掘现有的可视化计算资源，向一种新的分布式可视化技术转型。

分布式可视化技术是分布式技术与可视化技术结合产物，是分布式技术在可视化领域的具体应用^[3]。它可以将分散在不同地理位置上的可视化数据资源整合起来进行相关的可视化应用处理。分布式可视化技术中的分布主要是指可视化处理的数据源及可视化执行主体具有地理空间的分散性。任何新技术的发展都是一个艰难的过程，分布式可视化技术也不例外，随着分布式可视化技术的不得不断发展它逐渐暴露出各种问题，比如如何很好的实现异构平台互操作及可视化计算资源在异构系统之间的共享问题等。然而 web 服务技术和网格技术的出现，为分布式可视化技术的带来了新的发展契机。因为网格技术能够很好的实现各种异构系统间各种信息、数据、知识资源的共享；而 Web 服务技术又是解决各种异构平台间互操作的重要有效技术。利用这两种技术可以实现可视化过程更加灵活的构建，基于它们研究可视化技术将具有重要的理论和现实意义，势必将有助于充分整合利用各种已有的分散的可视化资源，更好的实现大规模数据的处理，促进人们科学研究的快速发展。

然而，新的问题总是伴随着情况的复杂化而产生。随着分布式可视化技术的发展，可视化服务的数量急剧增加，面对数量庞大的服务群，如何管理并发现满足需要的可视化 Web 服务是有效地利用各种可视化服务资源的前提与基础，而可视化服务匹配是解决这个问题的第一步也是最重要的一步。可视化服务匹配本质上是通过一定的匹配算法对用户需求描述与可视化服务描述之间相似程度进行计算或评价。可视化服务匹配作为 Web 服务匹配在可视化领域的应用，它既具有一般服务匹配的共性，又具备可视化技术本身鲜明的个性，例如，可视化服务匹配应该特别强调服务行为约束以及服务质量的要求，因为可视化服务的核心是各种可视化算法，同时可视化过程常常具有实时性的要求，所以研究可视化服务发布与匹配机理，具有很大必要性，及重要的理论价值和现实

意义。本文将针对可视化领域自身的特点，以服务发现的高效率、高查全率和高查准率为目标，基于功能语义研究可视化 web 服务的发布与匹配机理。

1.2 国内外研究现状与存在问题分析

目前，有关可视化服务发布与匹配的研究还鲜见报道。而作为 Web 服务各种应用的关键，Web 服务的发布与匹配研究已经成为热点问题。目前研究人员普遍采用服务查全率、查准率和查询效率作为服务匹配算法的评估参数，而且国内外学者对于 Web 服务匹配已经作出很多研究，其中比较有代表的工作如下。

● 国外研究现状

(1) 卡梅隆大学的 Paolucci 等人^[4]提出的基于输入输出语义 Web 服务匹配算法。该算法是建立在领域本体概念继承关系基础上，考虑服务请求和服务广告输入输出参数在本体概念树中的上下位继承关系，定义了四种匹配度：精确匹配 (exact), 插入匹配 (plugin), 包含匹配 (subsume) 和失败 (failed)。这个算法提出时间较早，它突破了单纯基于关键字的服务匹配思想禁锢，是基于语义的 web 匹配的新起点。该算法由于引入了服务语义信息，并基于语义概念本体上下位关系推理，所以提高了服务的“查全率”和“查准率”。然而该匹配算法相对比较粗糙，匹配度的划分粒度比较大。而且考虑的匹配范围较小，只考虑了 Input 信息和 Output 信息。后来也有不少学者，对 Paolucci 等人的算法进行了匹配等级的扩展，并引入了属性匹配，但是目前基于关系推理的概念匹配存在匹配等级粒度加大，匹配等级有限不能进行更细致的区分等等不足。

(2) 美国 Georgia 大学的 METERO-S^[5]项目是关于 Web 服务技术的研究项目。它采用 OWL-S 作为描述语言，对 Web 服务的接口进行计算机可理解的、无歧义的描述。它设计了一种基于服务模板的匹配算法，为用户提供服务查询功能。同时，由于它是基于 P2P 网络进行服务注册信息的管理，采用了基于本体推理的语义服务匹配算法。所以 METERO-S 系统具有伸缩性好、服务查准率高等特点。但是由于系统是基于 P2P 网络拓扑机构的，而且这个网络系统共享一个本体库，随着本体规模的不断扩大，这样使本体越来越难于维护，因此，在某种程度上限制 P2P 网络自身特点的发挥。而且该系统匹配算法是采用基于模板的设计，实现简单，不能满足复杂匹配算法和逻辑推理的需要。

(3) 南洋理工大学的 Van-Quoc Nguyen^[6]等人开发了能同时匹配 WSDL 和 OWL-S Web services 的系统。他们研究的方法，很好的规避了现有服务发现系

统只能发现 WSDL 服务或 OWL-S 服务(既一个 WSDL 请求无法匹配一个 OWL-S 服务)的限制。不足之处是该系统使用的是输入输出匹配算法,不利于匹配算法的移植和扩展。

(4)德国人工智能研究所的 Matthias Klusch,Patrick Kapahnke 和 Ingo Zinnikus 提出了 SAWSDL-MX2^[7]: A Machine-learning Approach for Integrating Semantic Web Service Matchmaking Variants 匹配算法。他们针对 SAWSDL 服务设计了 SAWSDL-MX2 方法,采用逻辑推理和文本相似度相结合,构建匹配模型,同时构建了一个 SVM 平台。该方法的不足之处是复杂度过高,与运用环境有关。

(5)文献[8]提出 Mixed-Integer Programming for QoS-Based Web Service Matchmaking 方法。他们最终开发了两种可选择匹配算法:CP-based 和 QoS-based WS 匹配算法。而且为了增加 OoS-based WS 匹配精确度,设计了一种排序算法。该算法能够很好发掘服务的非功能属性在服务匹配中的形象,但是没有提出一种综合性匹配算法。

(6)文献[9]中提出了 Matching Semantic Web Services Using Learning Accuracy 的算法。算法允许根据服务的相关性来划分服务的等级。learning Accuracy 被定义为一种衡量服务请求和服务发布的相似性度量。算法通过考察服务本体间的概念语义相似性和属性间语义相关性来设计概念间的匹配方法情况。该算法只是基于概念树语义距离相似性算法的扩展,而且没有充分考虑概念树中结点所处层次及结点的密集程度对相似度计算的影响。

● 国内研究现状

(1)清华大学计算系的 SEWSIP 项目组做了相对深入的研究工作^[10]。该项目组主要是开发了一基于语义的 Web 服务管理平台。该平台在服务匹配方面主要设计了服务接口匹配、服务非功能属性匹配和服务逻辑匹配。

(2)史忠植等人提出的多主体服务环境 MAGE^[11]。采用基于描述逻辑的概念层次推理,设计了一种基于描述逻辑的服务匹配算法,进而实现服务主体之间的相似度匹配计算。其设计的匹配算法只是单纯的考虑了概念的层次距离,并没有考虑到本体概念在概念层次树中的深度对概念相似度计算的影响。

(3)同济大学 CIMS 研究中心的刘敏,沈伟民,严俊伟和高琦提出了一个多层次的语义 web 服务匹配框架^[12]。设计了一种扩展语义 Web 服务框架。匹配过程包括五个层次:句法,静态语义,动态语义,定性服务和可靠性服务。该方法在一定程度上提高了服务匹配的查准率,但是其查全率并不高,而且层次划分界限比较模糊。

通过对国内外学者对 Web 服务匹配的研究分析发现,目前对于 Web 服务匹配研究工作中存在下面问题:

(1) 缺乏专门针对可视化领域,基于可视化特性的可视化 Web 服务匹配方法研究。

(2) 目前基于功能语义 Web 服务匹配算法大多都只是研究服务 IO 匹配。而且目前基于语义相似度的语义 Web 服务匹配算法,主要考虑的是概念间的语义距离,而没有考虑到概念在层次结构树中的深度对概念相似度计算的影响。

总之,目前针对可视化特定领域的,基于功能语义的服务发布与匹配方法的研究还鲜见报道。而随着可视化技术应用的不断深入普及,和可视化 Web 服务的不断增加,对于如何高效的查找到用户需要的可视化 Web 服务的研究将具有非常重要的理论与现实意义。

1.3 研究的内容及目的

基于功能语义的可视化 Web 服务发布与匹配机制的研究与实现概括起来主要是指基于服务的功能语义描述,充分考虑可视化技术领域自身的特点,设计可视化 Web 服务的发布和匹配方法,最后在此基础上,设计实现可视化 Web 服务发布与匹配原型系统,完成了可视化 Web 服务资源的有效管理和共享。主要的研究工作如下:

● 可视化 Web 服务的语义描述:

(1) 研究语义 Web 服务描述语言 OWL-S,设计可视化 Web 服务描述模型:随着可视化技术在各领域应用的不断深入,来自不同领域不同机构的可视化 Web 服务的数量不断增多,而且这些服务的描述又存在很大异构性,这将给可视化 Web 服务应用的进一步发展带来极大阻力。因此,基于可视化技术特点设计可视化 Web 服务描述模型,实现对可视化 Web 服务的准确描述是笔者研究的重点之一。

(2) 研究本体描述语言 OWL 及本体构建方法,构建可视化服务本体库:可视化服务本体是可视化领域共同概念的描述,构建可视化服务本体库,将很好的消除服务描述的异构性。

● 可视化 Web 服务发布与匹配:

(1) 研究可视化技术的特点,设计出可视化 Web 服务发布方法:服务发布是服务匹配的前提,服务发布方式的合理性,决定了服务匹配的质量,同时可视化技术领域有其自身的特点,例如,其具有有过程鲜明性,课题将基于可视

化技术的这个特点进行可视化 Web 服务发布方法的深入研究。

(2) 研究现有语义距离相似度计算方法, 提出了一种改进的基于语义距离的相似度算法: 相似度计算是可视化 Web 服务匹配的基础, 笔者将基于可视化技术领域的自身特点, 充分考虑服务本体概念之间的二元关系, 及概念在概念树中深度对相似度计算的影响, 提出一种改进的基于语义距离的相似度算法。

(3) 基于可视化技术的特点, 及笔者提出的改进相似度算法, 设计出可视化 Web 服务总体匹配方法及各阶段的具体匹配算法: 基于可视化特性, 针对笔者设计的可视化 Web 服务发布方法, 及服务描述中各指标描述方法的差异, 设计出可视化 Web 服务的总体匹配方法, 同时设计出总体匹配方法下 DataModel、Input、Output、和 textDescription 各阶段的匹配算法。

● 可视化 Web 服务发布与匹配模型的设计与实现

(1) 设计可视化 Web 服务发布与匹配模型: 为了更好的实现可视化 Web 服务资源的共享及管理, 笔者设计出可视化 Web 服务发布与匹配模型, 该模型将利用课题对可视化 Web 服务发布与匹配机制研究的成果, 结合服务资源管理的需要进行设计, 模型将支持可视化 Web 服务的发布和查找。

(2) 可视化 Web 服务发布与匹配原型系统的实现: 笔者采用 J2EE 技术实现论文设计的可视化 Web 服务发布与匹配模型, 该原型系统将充分利用 J2EE 技术的可扩展性、跨平台性, 通过采用 IE 浏览器做为客户端, 使用户可以方便的实现可视化 Web 服务资源的发布、查询、管理和共享, 同时原型系统还具有良好的可扩展性, 便于升级。

● 原型系统的性能评估

为了验证原型系统的可行性和有效性。笔者将通过构建一些可视化 Web 服务实例, 对原型系统进行服务发布与查询的测试, 并采用查全率、查准率和响应时间对测试结果进行定量的分析。

本论文研究的主要目的是针对可视化技术领域, 基于服务的功能语义描述, 设计可视化 Web 服务发布与匹配方法, 基于该研究成果设计可视化 Web 服务发布与匹配模型, 并实现原型系统。对于可视化 Web 服务发布者, 可以利用相关技术将各种可视化算法封装成可视化 Web 服务, 并利用本原型系统进行服务的发布和管理, 从而实现可视 Web 服务资源的共享。对于可视化 Web 服务的请求者, 原型系统为服务请求者提供了服务查询的用户接口。在用户接口中选择录入服务查询的相关描述信息, 待查询信息提交给系统后, 系统将为用户输出排序的候选服务列表, 服务请求者将根据系统查询结果选择合理的可视化 Web 服

务。基于原型系统最终可实现可视化 Web 服务资源的共享及有效管理。

1.4 论文的组织结构

论文主要通过对可视化技术领域进行深入研究, 针对可视化技术特定领域, 设计可视化 Web 服务描述模型及语义描述方法, 构建可视化服务本体库, 基于该服务描述模型结合可视化领域特点设计了可视化 Web 服务的发布与匹配方法, 在此基础上, 设计了可视化 Web 服务发布与匹配模型, 并利用该成果实现了服务发布与匹配的原型系统。最后通过构建一些可视化 Web 服务实例对原型系统进行了测试, 并采用查全率、查准率和响应时间对测试结果进行分析。

整篇论文的组织机构如图 1-1 所示。

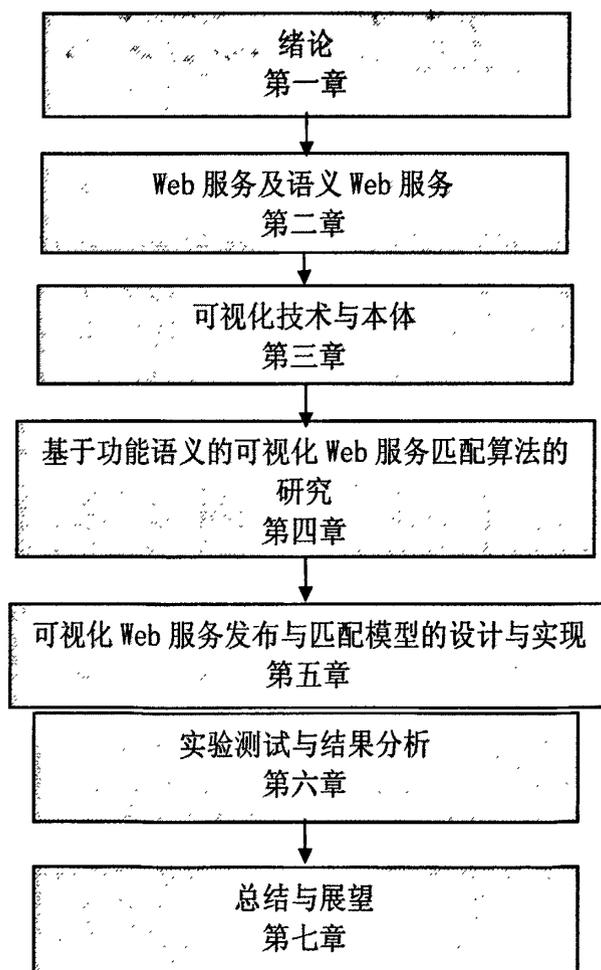


图 1-1 论文组织结构

第一章 绪论: 介绍课题研究的背景, 阐述课题研究的国内外现状, 分析当

前课题研究存的问题，最后介绍了论文研究的主要内容、研究目的及论文的组织结构。

第二章 Web 服务及语义 Web 服务: 本章主要介绍 Web 服务、语义 Web 和语义 Web 服务技术。首先详细说明了 web 服务的基本概念及体系结构，然后阐明了语义 Web 技术，最后着重对语义服务进行了详细的介绍。

第三章 可视化技术与本体: 本章主要介绍可视化技术、本体、本体描述语言 OWL 和可视化本体的研究与实现。首先对可视化技术进行概要性的介绍，然后详细阐述了本体论，包括本体的定义、本体的构成及本体的描述语言 OWL，最后着重介绍了可视化本体的研究与实现。

第四章 基于功能语义的可视化 Web 服务匹配算法的研究: 本章针对现有语义 Web 服务匹配算法的不足及可视化领域自身的特点，提出了一种改进的基于语义距离的本体概念相似度算法；然后针对可视化技术领域自身特点设计了可视化 Web 服务的描述模型及服务的发布方法，最后在上述研究成果的基础上，提出了基于功能语义的可视化 Web 服务匹配的总体目标，并介绍了总体匹配目标中各阶段的匹配算法。

第五章 可视化 Web 服务发布与匹配模型的设计与实现: 本章主要介绍可视化 Web 服务发布与匹配模型的设计与实现，首先阐述了该模型的主要设计思想，然后从宏观上说明了该模型的层次结构及各层的具体设计，最后介绍了原型系统实现的具体过程。

第六章 实验测试与结果分析: 本章主要介绍对可视化 Web 服务发布与匹配原型系统进行实验测试的具体过程及测试结果的分析情况。首先说明了测试环境的搭建，然后对测试过程进行了详细的描述，最后采用查准率、查全率及响应时间对原型系统的测试结果进行了详细分析。

第七章 总结与展望: 对本文在可视化 Web 服务发布与匹配机制的研究及模型设计与实现方面做的工作进行了总结和回顾，并针对工作中的不足，提出了下一步的工作的方向。

1.5 本章小结

本章概要的介绍了课题研究的背景，并在此基础上阐述了课题研究的意义；同时详细介绍课题当前国内外研究现状，并归纳出目前研究中存在的问题；接着详细介绍了论文将要研究的主要内容及课题研究的的目的；最后对论文的组织结构进行了描述，并详细阐述了各章主要内容。

第 2 章 Web 服务及语义 Web 服务

2.1 引言

本章主要介绍 Web 服务技术、语义 Web 技术和语义 Web 服务技术。首先详细说明了 web 服务的基本定义及体系结构，然后阐述了语义 Web 技术，最后着重对语义 Web 服务及其描述语言 OWL-S 进行了详细的介绍。

2.2 Web 服务技术

2.2.1 Web 服务的定义

Web 服务主要是用于实现异构平台间应用和数据的共享和整合，类似的技术有 CORBA、RMI 和 DCOM/COM 等，但是这些技术存在着种种不足，要么是技术过于服务难于实现，要么不能穿透防火墙，不能在互联网上进行部署，而且因为标准的不统一它们之间实现互操作的难度很大。Web 服务技术使用开放的、规范的 XML 文档格式来描述 Web 服务的基本信息、调用接口信息和服务交互的具体细节，包括传输协议、服务部署位置和消息格式等。

W3C 给出的 Web 服务的定义如下：Web 服务是一种软件系统，它支持在网络上计算机和计算机之间的互操作；它采用计算机可以处理格式来描述其服务调用接口；它允许其他机器采用 SOAP 消息^[13]与 Web 服务进行通信。

从 W3C 给出的 Web 服务定义可以看出，Web 服务本质上是一种与平台无关、独立于编程语言及操作系统的，应用与应用程序之间的通信机制。Web 服务将一组在网络上可以通过标准化的 XML 消息传递进行访问的操作定义为服务的软件接口^[14]。而且 Web 服务所使用的协议通常都是开发的标准，例如 HTTP 和 SOAP 等。

2.2.2 Web 服务的体系结构

Web 服务采用了 SOA^[15]面向服务架构，主要包括服务请求者、服务提供者、服务注册中心，Web 服务的体系结构如图 2-1^[16]。

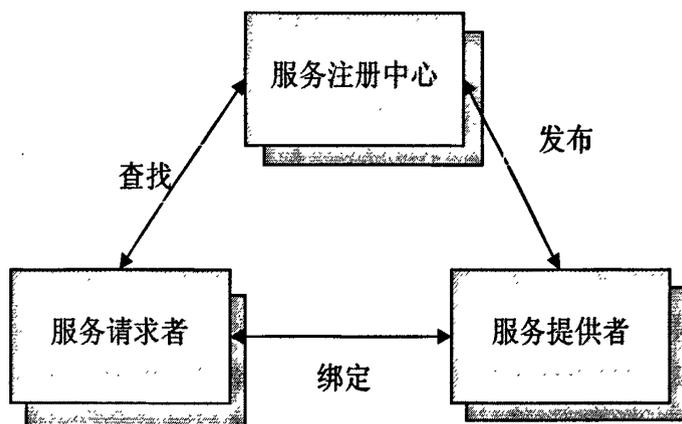


图 2-1 Web 服务体系结构

从 Web 服务的体结构图中可以发现，Web 服务的 SOA 架构实现中主要包括以下三个角色：

服务请求者：定义服务请求的描述信息，然后在服务 UDDI 注册中心^[17]查需所需服务，然后提供服务执行所需要的参数执行服务。

服务提供者：编写自己 Web 服务应用、发布服务，并在 UDDI 注册服务，同时能响应服务调用请求。

服务注册中心：服务注册中心主要提供服务发布注册的平台，对已经注册的服务各种管理，并提供服务查询的接口。

这几个角色之间主要涉及到下面几个互操作：

(1) 发布：服务发布是服务调用的前提，服务发布者需要将服务 WSDL 描述发布到服务注册中心，以方便服务请求者对服务的调用。

(2) 查找：服务请求者通过服务注册中心为用户提供的服务查找接口，录入查找服务的相关信息，便可在服务注册中心查找到相应的服务。服务请求者一般会在两种情况下需要查找服务：在应用设计时为了程序开发而查找服务的接口描述；在应用执行时为了调用服务而查找服务的位置及绑定。

(3) 绑定：在服务调用时需要将请求者与服务进行绑定。在服务绑定操作中，服务请求者使用服务描述中的具体绑定信息来定位和调用服务，从而实现服务的远程调用。

2.3 语义 Web

1998 年 Tim Berners-Lee 首次提出了语义 Web 的概念，并给出了语义 Web

的定义：“语义 Web 是一个可以被机器所理解的 Web”^[18]。Tim Berners-Lee 关于语义 Web 的定义是对未来 Web 体系架构的一个伟大构想。

语义 Web 也被称为下一代 Web 技术，但是它并不是完全抛弃现在的 Web，而是对现有 Web 进行扩展，使 Web 上的信息定义规范化，具有良好的含义，进而实现计算机对 Web 信息自动解析。

当前基于传统 Web 的网页的设计，并不是以提供机器能够理解的信息为出发点，所以它阻碍了计算机对网络信息、资源智能化利用的发展。而语义 Web 研究的目的是通过对当前互联网进行适当语义扩展，使其能表达更多的语义信息，从而使计算机能够自动理解及捕获网页上面承载的信息。语义 Web 技术的目标是通过采用适当的语义扩展将互联网上的巨大资源整合起来，将传统的互联网变成一个海量的“知识数据库”，而不是传统意义上的网页互联的集合。Tim Berners-Lee 在世界 XML 大会上，提出的语义 Web 体系结构图^[19]，如图 2-2。该语义 Web 体系结构从上到下一次分为七层：信任（Trust）、证明（Proof）、逻辑（Logic）、本体（Ontology）、RDF 和 RDF Schema、XML、Unicode 和 URL。下面将各层进行详细的介绍。

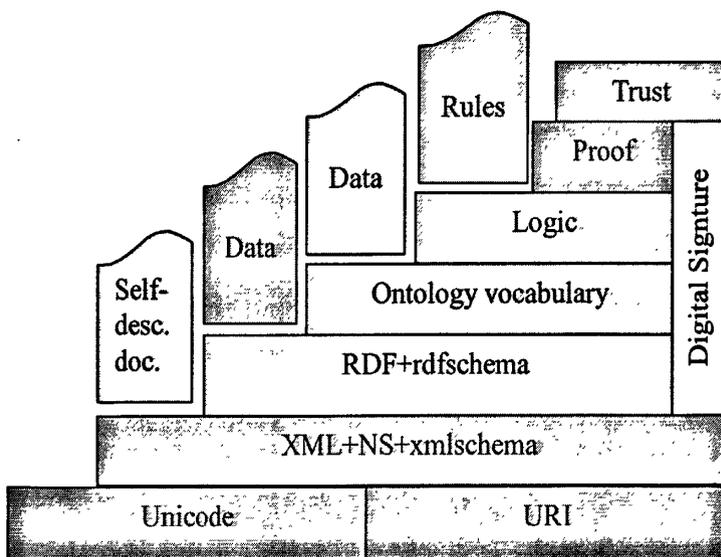


图 2-2 语义 Web 体系结构

●URI 和 Unicode

为了实现在 Web 环境下，应用之间能以计算机可理解的信息格式进行相互通信、传递或发布信息。而这些消息大部分都是对 Web 上资源的描述，因此需要用明确的方式来表示这些资源。语义 Web 是采用 URI 来描述资源及资源的属性。同时为了实现构建全球新网络，语义 Web 采用了统一的 Unicode 编码

作为字符编码方案。所以 URI 和 Unicode 层是语义 Web 体系结构的基础层，它很好的实现了互联网上信息定位及信息跨地区字符的统一描述问题。

●XML+NS+xmlschema

XML+NS+xmlschema 层是位于 URI 和 Unicode 之上的 XML 及相关技术层。XML 是一种元语言，其最大的特点是允许用户根据自己的实际需要自定标签来标记相应内容，同时可以使用 XML Schema 对这些标签进行约束。同时为了解决因为自定义标签带来的潜在冲突，XML 引入了 NameSpace 策略。XML+NS+xmlschema 层通过利用 XML 技术的上述特性，实现了对资源文档的机构描述，并实现了应用之间的语法互操作。但是，因为 XML 只是提供数据描述的格式，它很好实现了资源文档的结构化描述，但是对于文档语内容的语义，及互操作，还需要语义 Web 上层来解决。

●RDF+RDF Schema

RDF+RDF Schema 是位于语义 Web 体系结构 XML 层之上的数据操作层。RDF 是一个开发的元数据框架，用来描述元数据及它们之间的关系，是处理元数据的基础，它为实现应用之间基于机器可理解的信息交互提供了可能性。此元数据框架确定了一种数据语义描述的数据模型，而且是计算机可自动理解的。该数据模型由如下几个对象类型组成。

(1) 特性：特性是对资源属性，特征及它们之间关系的描述。

(2) 资源：资源可以是网页中的一部分、整个网页或者是不基于互联网访问的对象，及对象集合。

(3) 声明：声明是一个 RDF 声明，它由资源、特性及特性值组成。它们在声明中对应的名称分别为：谓词，主题和对象。RDF Schema 进一步定义建模原语，增强了 RDF 对资源的描述能力，可以用来定义一个类之间的各种关系。

●Ontology

本体(Ontology)层是位于 RDF+RDF schema 层之上的，它弥补了 RDF/RDFS 语义描述能力弱得不知。本体概念是领域中共享概念模型的明确的形式化规范描述。本体是一种能在知识和语义层次上领域中的概念模型，能够很好的消除语义描述的异构问题，能够很好的实现对同一语义概念的共享及相互理解。因而 Ontology 层是构建语义 Web 的核心层及关键层。

●Logic、Proof 和 Trust

前面介绍的语义 Web 体系结构中的 RDF+RDF schema 及本体层，能够很好的实现 Web 资源的语义描述。但是对于语义 Web 来说，基于语义描述的 Web

还需要能够根据这些语义描述进行相应的推理或信息抽取。Logic 层就是提供一种方法来描述规则。逻辑层具体实现可以根据语义网具体的应用类型采用不同的描述逻辑语言来实现。

目前关于 Proof 和 Trust, 人们还没有在概念上达成一个共识, 但 Proof 和 Trust 对与语义 Web 的必要性是得到了一致的肯定。用户在知识描述上进行推理的过程就是证明的过程而推理后得到的结论应该是可以信任的。

2.4 语义 Web 服务

2.4.1 语义 Web 服务概述

语义 Web 服务是语义 Web 与 web 服务技术结合的产物, 它是在语义 Web 研究的基础上, 将语义信息引入到 Web 服务中, 使得服务的发现、监视、调用和组合变成自动化。

随着 Web 服务普遍使用, Web 上存在越来越多的 Web 服务了, 作为一种重要的 Web 资源, 也需要得到语义 Web 的语义支持, 进而实现语义 Web 服务。OWL-S 就是一种用来描述 Web 服务躬耕和属性的 OWL 本体规范。它提供了一个可共享的框架, 使用一系列基本的属性和类来描述服务, 使得 Web 服务变成机器可理解的对象, 进而实现服务的自动发现、互操作、调用、组合和执行等。

2.4.2 OWL-S 描述语言

OWL-S^[20]是基于 OWL 本体的语义 Web 服务描述语言, 它建立了一套标记语言结构来描述 Web 服务的功能和属性, 从而实现 Web 服务的自动化。OWL-S 是 WSDL 传统 Web 服务描述语言的基础上发展起来的, 它不仅描述了服务和用户及服务与服务之间的互操作, 而且描述了这些操作的语义内容。

在 OWL-S 中, 主要有三类本体用于描述服务的基本信息: ServiceProfile、ServiceModel 和 ServiceGrounding。其中 ServiceProfile 是描述服务时做什么的; ServiceModel 是描述如何来使用服务; ServiceGrounding 是描述如何来访问服务。用 Service 表示一个发布的 web 服务实例, 则语义 Web 服务 OWL-S 描述的顶层机构如图 2-3。

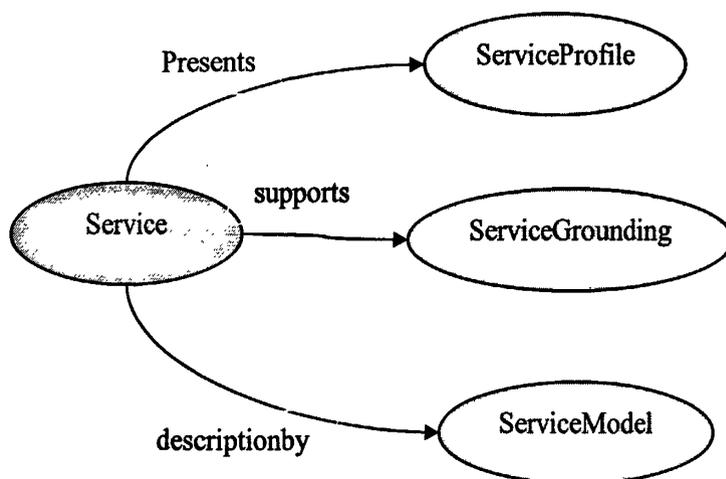


图 2-3 OWL-S 描述模型的顶层结构图

● 服务概要 (ServiceProfile)

类似于服务的黄叶，服务提供者可以用来描述他提供的服务，服务请求者可以用来描述他所需要的服务。ServiceProfile 主要用来描述服务的功能信息、非功能信息及服务的一些附加特征信息。

服务的功能信息主要包括输入、输出、前置条件和执行效果。Profile 本体通过 hasInput、hasOutput、hasPrecondition 和 hasEffect 等属性来描述服务的功能信息。

非功能信息主要包括服务名称，服务提供者信息，这些信息一般不能自动化处理。

附加特征信息主要包括服务的分类信息及服务的 Qos 信息。通过 serviceParameter 及 serviceCategory 可以实现对服务质量及服务类别的描述。

● 服务模型 (ServiceGrounding)

描述了服务的过程模型(Process)。OWL-S 的过程模型主要描述关于 Process 的两方面信息：一是描述从外部看到的 Process 的 IOPE,另一方面是描述 Process 的内部控制结构。其主要包括三种过程：原子过程、简单过程和复合过程。

原子过程是可以直接调用的，而且没有子过程。只要提供服务的输入参数，便可以以对服务请求者透明的方式执行。同时对于每一个原子过程有一个 Grounding 与之对应。

简单过程没有 Grounding 与之对应而且不可以直接被调用。它可以用于简化表示复合过程及原子过程的视图。

复合过程是由其他复合过程或原子过程组合而成的，而且控制构造来定义具体组合方法。控制构造用于定义复合过程中具体执行顺序。

● 服务绑定 (Service Grounding)

Service Grounding 主要用于描述服务如何被访问的细节, 主要包括服务的协议、序列化、消息格式、寻址等相关信息。Service Grounding 可以看作是服务所需元素的抽象秒速到具体描述的映射。

同时因为 WSDL 是传统的 web 服务描述语言, 而且得到业界广泛的支持。所以 OWL-S 选择利用 WSDL 作为 Service Grounding 机制的基础。可以利用 WSDL 和 OWL-S 这两种语言规范各自优势。一方面用 OWL-S 的过程模型来对服务进行抽象描述, 以充分发挥 OWL 语义表达能力强的特点。另一方面, 用 WSDL 来对服务进行具体描述, 以实现 WSDL 相关技术的重用。WSDL 和 OWL-S 之间的映射关系如图 2-4。

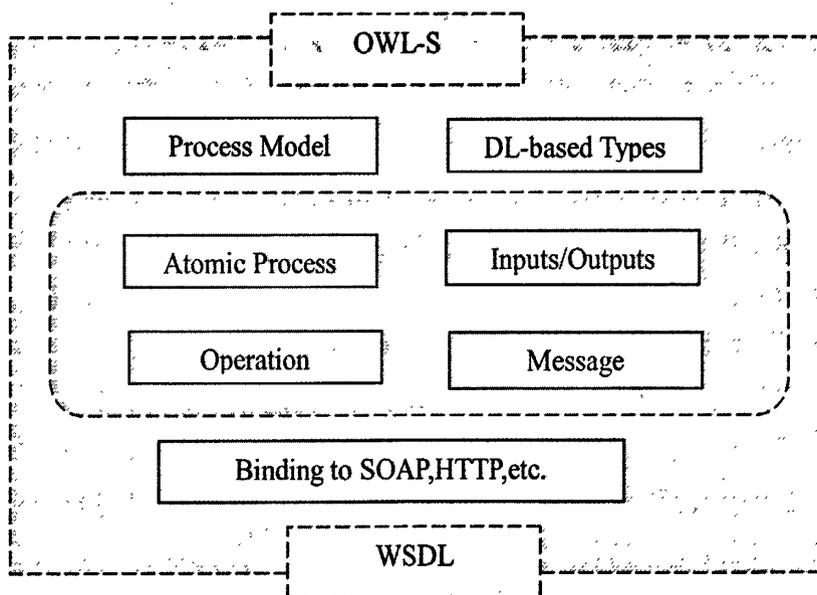


图 2-4 WSDL 和 OWL-S 之间的映射

- (1) OWL-S 的原子过程对应于 WSDL 的操作。它们大多数时候是一一对应的, 也可以是一对多的关系。
- (2) OWL-S 原子过程的输入和输出分别对应于 WSDL 输入和输出消息的概念。
- (3) OWL-S 原子过程中的输入、输出的类型对应于 WSDL 抽象类型的表示。

2.5 本章小结

本章主要介绍了 Web 服务技术、语义 Web 技术和语义 Web 服务技术。首先详细说明了 web 服务的基本概念及体系结构，然后阐明了语义 Web 技术，最后着重对语义 Web 服务及其描述语言 OWL-S 进行了详细的介绍。为后续研究打下理论基础。

第 3 章 可视化技术与本体

3.1 引言

本章主要介绍可视化技术、本体、本体描述语言 OWL 和可视化本体的研究与实现。首先对可视化技术进行概要性的介绍，然后详细阐述了本体论，包括本体的定义、本体的构成及本体的描述语言 OWL，最后着重介绍了可视化本体的研究与实现。

3.2 可视化概述

3.2.1 可视化技术研究背景

可视化 (Visualization) 是利用计算机图形学和图像处理技术，将数据转换成图形或图像在屏幕上显示出来，并进行交互处理的理论、方法和技术。它涉足到计算机图形学、计算机辅助设计、图像处理、计算机视觉等诸多领域。是一项对数据处理、数据表示、及决策分析进行研究的综合技术。

可视化主要包括两个方面的技术：信息可视化技术和科学可视化技术，而后者是近年来可视化学者和专家研究的重点、热点。“科学可视化”一词是在 1987 年 2 月美国国家科学基金会 (National Science Foundation, 简称 NSF) 主办的一个专题研讨会上被提出来的，而且在会议上还讨论了“科学可视化”的研究内容，及未来的发展方向等。此后，有很多的国家都开始逐渐重视，并加大对该新兴技术的研究力度，这也意味着“科学可视化”这门新兴学科在国际范围内已经开始成熟起来。

可视化技术是在社会的巨大需求和技术水平极大进步的深刻历史背景下出现的^[21]。早在 20 世纪初期，图表和统计等原始的可视化技术已经被人们应用到科学数据的分析当中。计算机的诞生和普及，将人类社会推向了一个崭新的信息时代，目前所进行的科学研究的目的已不再是单纯的为了获得数据，而是为了能够挖掘数据信息被后的内在规律，更好的探索自然规律。随着科学研究的不断深入，及处理数据规模的不断膨胀，传统的可视化技术已经无法满足人们

的需求。不过随着人类制造工艺水平、材料科学的发展进步,计算机软硬件性能的不不断提高,把计算机技术引入可视化领域,将大大加速可视化技术的发展。

3.2.2 可视化技术的应用领域

可视化技术凭借自身技术特点能够很好帮助科学研究人员理解分析实验或模拟计算所获得的大规模数据,并且允许研究人员对计算过程进行控制,以实现改进计算结果的目标。而且随着可视化技术的不断发展,它已近从当初只运用于科研领域,发展到生产领域,现在可视化技术的应用领域已经非常广泛,几乎可视应用于工程技术和自然科学所有相关领域。

●分子建模

从 20 世纪 60 年代开始,人们就使用交互式图形生成技术来观察复杂的化学物质,而且已经成为了工业界和学术界对分子及分子之间关系进行研究的主要工具。可视化技术的发展也进一步推动了分子模型构造技术的发生变化。现在,人们已经开始使用一些复杂工具来分析和设计分子机构,例如通过使用超级计算机来构造复杂的 DNA 模型,在遗传工程的药物设计中也通过引入三维彩色立体显示技术来改进已有药物的分子机构或新药物的设计。

●医学

虽然计算机断层扫描和核磁共振图像技术已经在医疗诊断中得到广泛应用,但是这些医疗器械只能提供关于人体内部的二位图像,因此医生只能凭借其从医经验根据二位图像去诊断病灶的形状和大小,进而构思人体该部位病灶与周围组织的三维几何关系,这种对医生经验过度依赖的诊断方法给治疗带来了极大的困难。

可视化技术的主要特点就是能够通过对相关系列的二维图像的处理,构建出它们的三维形体,通过显示设备展现给用户,医生可以很方便的利用可视化技术实现医学诊断、放射治疗、脑结构图分析和矫形手术的手术规划。

●气象预报

天气预报与人们的生活息息相关,特别是对灾害性天气的精确预报能够大大的减少人们生命财产损伤。而精确的气象预报工作的一个难点在于其需要对大规模数据进行计算及计算结果的分析。可视化技术通过将大量计算机结果转化为图像,并将某一时刻的等温面、等压面、云层位置和运动、暴雨区的位置及风力大小、强度和风向显示在屏幕上,这样气象预报工作人员能够对未来

天气状况作出准确的分析和预测。

●地质勘测

地质勘测人员通过采用模拟人工地震的方法，根据模拟实验获得地质岩层的相关信息，并获得大量关于地下矿产资源分布的数据。借助于可视化技术对大规模数据的处理能力及可视化特点，采用可视化方法对所获得的大规模模拟地震数据进行分析、出里，勘测人员以可视化方法处理结果为参考可以大大提高地质勘测的安全性和效率。为人类能源探测发展提供技术支持。

●有限元分析

目前，有限元分析在大坝建设、结构分析、飞机设计和地壳板块运动等方面得到了广泛的应用，是计算机辅助设计的核心技术。通过将可视化技术引入有限元分析中，可以实现有限元分析结果数据和形体的网格划分的图形显示，通过对结果分析，能实现网格剖分优化，使计算结果更加精确和可靠。

除此以外，可视化技术在考古、航空工业等领域也得到了广泛应用。总而言之，随着可视化技术自身的不断发展，它已经在社会各行各业中被广泛的应用。而且随着人类社会科学技术的不断发展，其对可视化技术需求的不断递增，可视化技术将会得到更广泛和更深入的应用。

3.2.3 可视化技术的研究内容

科学计算可视化是可视化的重要分支，因此现代的可视化主要指的是科学计算可视化。科学计算可视化 (ViSC, Visualization in Scientific Computation) 是指运用计算机图像处理技术和计算机图形学，把科学计算过程中获得的数据和计算结构转换为图像或图形，然后再屏幕上显示出来，并进行交互处理的理论、方法和技术。它设计到计算机图像处理、图形学、计算机视觉、计算机辅助处理和人机交互等多个学科。可视化的研究内容主要可分为两大部分：可视化应用的研究和可视化工具的研究。科学计算可视化主要是研究有关可视化参考模型的内在信息，即可视化工程的组成内容，具体内容如下^[22]：

(1) 数据预处理。可视化的数据来自各个领域的计算模拟数据或实验测量数据或数据库数据，因此数据格式是各种各样的，数据预处理部分的主要任务是完成数据格式的转换，为所采用的可视化工具提供其可以接受的数据格式。

(2) 映射。映射将数值数据转换为图元数据，是连接可视化模型与图像模型的桥梁，是可视化技术的核心。它提取数据中的蕴含的各种现象、科学规律

等，然后用可见的点、线和面将这些抽象的现象和规律展示出来。最后由图像引擎根据这些图元数据进行图形绘制。

(3) 绘制。绘制主要是按照映射的点、线、面等相应的绘制方法将其在屏幕上呈现出来。

(4) 显示。显示模块主要是将绘制模块生成的图像数据，按照用户的要求如显示窗口的位置与大小，输出设备的类型，文件的存储格式等进行输出。

3.2.4 可视化技术研究的意义

科学计算可视化是发现数据内在信息，洞察内在规律与关系的有效工具。在科学计算可视化技术的支持下研究人员可以更形象更直观的揭示科学数据中蕴含的内在规律。因此研究科学计算可视化的意义重大。

(1) 在人们的通信、数据处理中实现图像通信，而不是采用基于传统的数字或文字信息的通信，借助于此人们能够方便的观察到科学计算中发生的各种过程现象，为理解和发现科学计算过程中的各种现象及规律提供有力的工具。

(2) 极大加快数据的处理速度，提高数据处理效率。使得人类社会不断膨胀的数据能够得到及时有效的利用。

(3) 提供了计算机辅助下的各种可视化技术方法，从而帮助实现在网络分布环境下的计算机辅助协同设计。

(4) 数据可以图像、二维图形、曲线、三维模型和动画来展示，并可借助可视化技术对数据的模式及内在规律进行可视化分析。

总而言之，科学计算可视化的研究将大大的提高科学计算的速度和质量，将加速科学计算环境和工具现代化的步伐，从而使人类社会的科学研究工作的状况发生根本性的变化，具有非常重大的现实意义。

3.3 本体

3.3.1 本体论

本体论是语义 Web 的基石，它是用来描述语义信息元数据的元数据。在语义 Web 中，人们可以定义自己的语义表述，但是在与其它服务交互时，需要有一种机制申明所定义的概念，否则就不可能解释元数据的含义。而本体论是一种能够在知识层面实现知识重用和共享的工具，它为人们提供了一种明确的语

义定义方式, 计算机根据本体论定义的语义, 能够理解数据的语义及实现机器间的互操作, 从而实现语义 Web 的数据能够自动的被计算机理解、分析和处理的目标, 真正实现网络的智能化。

3.3.2 本体定义

本体论(ontology)是哲学概念, 最早起源于西方哲学, 它是研究存在的本质的哲学问题。但随着信息技术的发展, 这个词被应用到计算机领域, 并在计算机语言、人工智能和数据库理论中发挥着越来越总要的影响。很多文献从不同研究角度和问题域出发, 对本体进行了各自的定义。其中 Studer 等提出的定义已经被广泛的接受: “本体是共享概念模型的明确的和形式化的规范说明”^[23]。该定义描述出本体的四个特征: 概念模型、形式化、明确和共享。概念模型是通过对客观世界中的现象进行抽象提取, 它蕴含的含义不依赖于具体的环境状态。形式化是指本体是计算机可理解、可读取的和可处理。明确是指本体所使用的概念及概念的约束都有明确的定义。共享是指本体描述的是人们共同认可的知识, 是相关领域中公认的概念集地反映。

3.3.3 本体构成

作为不同主体之间进行交流的语义基础。本体由描述某种现实情况的特定术语集和一组关于术语内涵意义的显式假定集合构成。在最简单的情况下本体只用于描述概念的分类层次结构; 在复杂的情况下本体可以通过将一组合适的概念间关系、规则、公理引入到概念分类层次结构中, 以表示概念之间的其它关系, 约束概念的内涵解释。一个完整的本体包括下面五类基本元素, 概念、函数、关系、实例和公理:

●概念是一个广义上的概念。既可以是广义上的概念, 同时也可以是为行为、任务、推理、功能过程和策略等。通常本体通过有各种概念实现一个分类层次结构。

●函数是一种特殊的关系。 n 元函数中的第 n 个元素由前面 $n-1$ 个元素唯一确定。通常, 将函数描述为 $F: C_1 \times C_2 \times C_3 \times \dots \times C_{n-1} \rightarrow C_n$ 。

●关系被定义为概念之间的交互关系。形式上表示为为 n 维笛卡尔积的子集: $R: C_1 \times C_2 \times \dots \times C_n$ 。在语义上关系与对象元组的集合对应。

●实例是指领域概念中的具体例子。领域概念在某领域中的值域就是由该领域的所有实例构成。

●公理是指一些永真式的描述。公理就是领域中在任何条件下都成立的断言。

3.3.4 本体的描述语言OWL

近年来出现多种描述和表示本体的语言。如 RDF Schema、RDF(Resource Description Framework)、DAML+OIL^[24]以及 OWL(Web Ontology Language)语言^[25]。其中前三者都仅仅是基于 XML 规则的描述语言,而 OWL 是一种基于 RDF 和 XML 描述逻辑的语义 Web 本体描述语言,它来源于 DAML+OIL,而且保持了 DAML+OIL 的大多数语法和框架、语义特征,同时针对不同的应用范围做了限制和扩充。

OWL 的基本语法中包括如下几个概念:类、个体、属性等^[26]。

●类:它是组织具有相似特征的资源的一种抽象方式,每个 OWL 类都与一个个体集合关联,这称为类的外延,类的实例就是这个类的外延中的个体,可以通过说明一个匿名类的外延或指定一个类名字来描述一个类。

●个体:个体的相关描述是基于类描述后对一个具体世界的描述。通常可以将个体描述看做是申明具体世界中的事物,而将类描述看做一些高层的规定和约束。

●属性:它表示的是类之间的二元关系,提供类之间关系描述的抽象机制。根据不同的应用需求,OWL 形成了三个子语言^[27]:

OWL Lite 可以用来描述类的简单地约束和层次结构。

OWL DL 是在满足计算可确定性和完全性的条件下,最大限度的支持可表达性。

OWL Full 则是最大限度的与 RDF 兼容,不考虑计算可确定性和完全性的约束,具有最大的可表述能力。

三个子语言的限制由少到多,其表达能力逐次下降,但其可计算行依次增强。

3.4 可视化本体的研究与实现

3.4.1 可视化本体的研究现状

当前在国内外对于可视化本体的研究仍然处于起步阶段,2003 年和 2004

年英国的两届 eScience 会议 “Visualization for eScience” 和 “Visualization Ontology”^{[28][29]} 最早开始对可视化本体进行研究。两次会议主要讨论了建立可视化本体的必要性，及确定了构成可视化本体的一些基本概念，以及使用这些概念构建的可视化本体的顶层模型，而该模型仅仅表达的是一些概念及概念间的关系，因此还需要进一步的丰富和完善。会议提出的基本概念集，如表 3-1，会议提出的顶层模型如图 3-1^[30]。

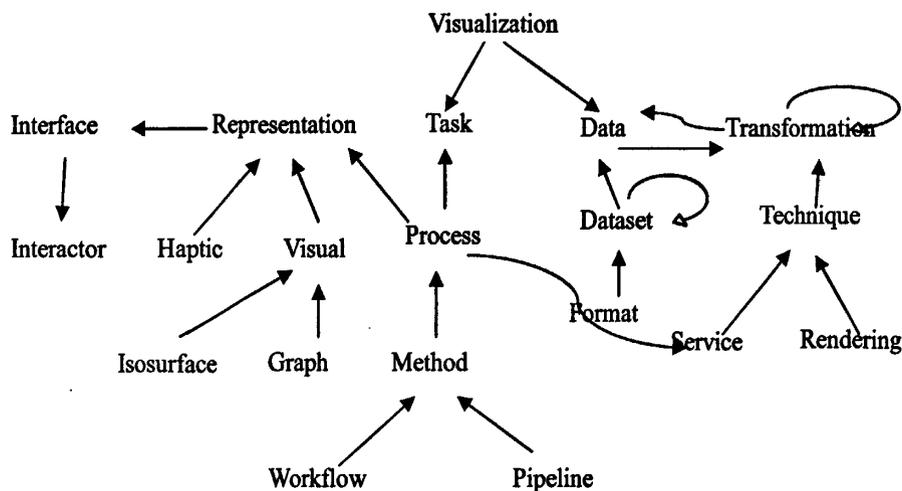


图 3-1 2004 英国 eScience 会议提出的可视化本体的顶层模型

表 3-1 可视化领域的基本概念

Data	Parameter	Filtering/Sampling	Algorithm
Time	Coordinate System Dataset	Abstraction	System
Dimensionality	Rendering	Metaphor	Data Source
Data Type	Representation	Device	Domain Ontology
Format	Transformation	Task	Scenario
Network/pipeline	Process	Interaction	Static/Animation
Data Model			

通过上面分析可知目前可视化本体的研究工作还非常不成熟，基本处于实

现性阶段，同时因为可视化本体的构建需要整个可视化领域的共同认可，而且可视化本体的构建又是一个不断往复和渐进的过程，所以可视化本体的研究是一项任重而道远的工作，需要更多的研究者参与其中。

但是可视化本体的分类学却是已经有了很多研究成果，其中最具有代表性的分类研究成果有以下几种：

●Bergeron 和 Grinstein 于 1989 年提出了基于网格方式的 L_n^k 分类方法，其中 L_n^k 表示了具有 n 维数据的 k 维网络。网格的维数表示了数据的阶，一维网络是一维有序的数据点集，零维网络是一无序点集，二维网络表示的是一个二维有序的数据点集，这种可视化分类方法不是面向数据模型的而是面向数据分布的。

●Brodlic^{[31][32]}将 Bergeron 和 Grinstein 的方法更向前推进了一步，提出了通过矢量、标量、张量等数据类型来构建可视化分类体系。该分类方法的主要出发点是，他认为数据模型比数据分布方式更重要，可视化关注的往往是实体而不仅仅是数据。Bordlie 的分类方法采用 2 次分类法，其关注的重点是数据与表示之间的联系。它主要考虑的两个要素：1) 实体定义域的维度；2) 实体的数据类型。它首先将数据类型作为主分类，它定义了点型、标量型、矢量型和张量型等五中类型，其次以定义域的维度作为此分类的次标准，如一维标量场，三维矢量场。表 3-2 给出了基于数据类型和维度的可视化技术分类表^[33]。

表 3-2 基于数据类型和维度的可视化技术分类表

维数	点型	标量型	2D 矢量型	3D 矢量型
1	1 维散乱点标绘图	线画图 柱形图 直方图		
2	2 维散乱点标绘图	等值线抽取 图像映射 造型法 约束区域图 2 维直方图 2 维柱形图	2 维场箭头 2 维质点轨迹 2 维场拓扑 2 维场线	平面箭头
3	3 维散乱点标绘图	等值面抽取 体绘制 Cuberille 方法		3 维箭头 3 维场线
n	N 维散乱点标绘图			

同时为了准确清晰的描述数据模型，Brodlic 还提出了 E-符号的概念既用符

号 E 来表示数据实体。用 S 表示标量型，用 P 表示点型，用 V 表示向量型，用 T 表示张量型，其中字母 S,P,T,V 用作字母 E 的上标，同时用数据场的维度作为 E 的下标，例如 $E_3^{T_{20}}$ 表示 3 维空间上的 2*3 张量， E_2^V 表示 2 维空间中的矢量。而且，可以在字母 P,V,S,T 前添加整数来表示变量的实体，例如 E_1^m 表示 1 维标量域上的多标量实体。这种分类方法能够从可视化数据的角度比较好的刻画出可视化的特性，因此通过这种描述方法可以非常容易的将数据模型实体与具体的可视化算法对应起来。因为通常特定的可视化算法都是用来处理某种具体的数据模型实体的。如表 3-3 描述了部分数据模型实体与可视化算法的对应关系。

表 3-3 部分数据模型实体与可视化算法的对应表

实体	E-符号	可视化算法
1 维、2 维、3 维多变量数据	E_1^P, E_2^P, E_3^P	<i>Scatterplots</i>
n 维多变量数据	E_n^P	<i>Andrewscurves</i>
一个连续的一维标量域上的一组点	E_1^S	<i>LineGraph</i>
n 个不同的连续一维标量域上的一组点	E_1^{mS}	<i>MultipleLinGraph</i>
与一个枚举集上的实体相关联的一组数值	$E_{(1)}^S$	<i>BarChar; pieChar</i>
一个连续的二维标量域上的一组数值	E_2^S	<i>LineBaseContoring</i>
定义在一个连续的一维标量域上的一组数值	$E_{[1]}^S$	<i>Histogram</i>
同一个连续的二维标量域上的两组数值	E_2^{2S}	<i>ColouredHeight - fieldPl</i>
同一个连续的二维标量域上的多组数值	E_2^{nS}	<i>StickFigure</i>
一个连续的三维标量域上的标量场	E_3^S	<i>Volume Rendering</i>
二维矢量场	E_2^V	<i>2DArrowPlots</i>
二维平面上的三维矢量场	E_2^V	<i>3DArrowsInPlane</i>
体上的三维矢量场	E_3^V	<i>3DStreamLines</i>

3.4.2 可视化本体的构建规则

在可视化领域中，由于不同的研究团队对同一事物的认识和表示也存在差异，使得研究人员关于同一 Web 服务的描述存在冲突，形成语义异构。为了确定可视化领域中的共同认可的概念以及他们之间的相互关系。为描述可视化数据、可视化处理过程提供公共词汇。支持可视化 Web 服务的匹配和描述。使得可视化服务的使用者和开发者、可视化服务之间均能相互理解和准确交互，所以本文提出了建立可视化本。

在业界对于本体的开发暂没有非常诚实的理论方法支持，因为在各种领域都有自身的特点，同时构建本体的过程、设计标准和原则也都各有不同。但是 Gruber 提出了几条在业绩比较认可的本体构建原则^[34]：

- 清晰客观：应该从客观的角度出发，基于实际应用的需求去定义一个概念。
- 可扩展性：本体是大家共同认可的词汇库，它应该能够根据需求的变化而进行相应的扩展。
- 完整一致：本体要使用完整的定义，而且各个概念在逻辑上应该是一致的。
- 最小约束：对待建模对象应该尽可能少列出约束限定条件。

3.4.3 可视化本体的建模

可视化技术是运用计算机图形学的方法和理论，从而挖掘出隐藏在大量数据中的自然规律和复杂物理现象等信息，并将其转化为直观的图像、图形。因此，它涉及到图像处理、计算机图形学、计算机辅助设计和计算机视觉等多种技术和学科。但是概括起来，其基本概念主要包括：可视化的数据模型、可视化的算法/技术、可视化的显示和可视化系统。因此，要建立完善的可视化本体将是一个任重而道远的工作，它需要可视化业界的共同认可和努力，才能不断的改进和完善。

本文根据可视化过程的特点，及从满足可视化服务发布与匹配的需求出发，同时参考可视化 VTK^[35]工具包中的类库定义，首先确定了可视化领域的一些顶层的基本概念及概念之间的相互关系，然后再此基础进行逐步的扩展，添加子类，最后采用 Jena 推理机对本体进行一致性检验。笔者设计的可视化本中的顶层核心类包括：DataSet、Algorithm,其层次机构如图 3-4。

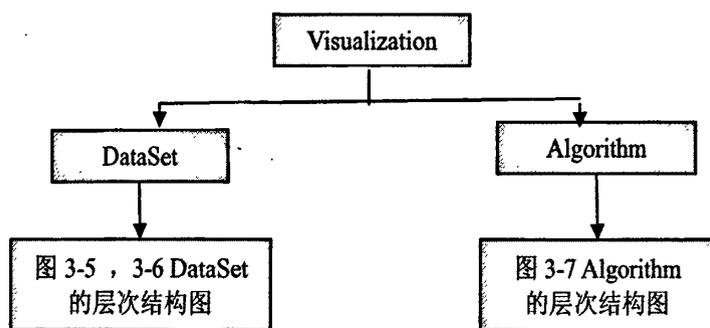


图 3-4 可视化本体顶层概念层次结构

下面是 DataSet 和 Algorithm 的层次结构：

●DataSet 描述的是可视化中的数据集，笔者为其设计了两个直接子类，分别是 DataModel 和 DataType。类 DataType 定义了可视化过程中涉及到的一些数据概念，譬如 JPG、DICOM 等概念与可视化数据中的 JPG 图片和 DICOM 图片格式数据对应，它们都是 ImageData 类的子类；类 DataModel 主要用于描述可视化数据的特性，笔者采用了 K.W.Brodlie 的 E-符号描述 DataModel 中的各子概念，同时因为论文研究的是科学可视化，所以笔者主要针对 DataModel 中的连续性数据模型构建本体，并将该连续性数据模型分为：矢量（EV）、标量型（ES）、点型（EP）、张量型（ET）和多变量（Emulti_var），对这些子类根据具体的维度，还可以进一步细化，DataSet 的部分本体层次结构如图 3-5。

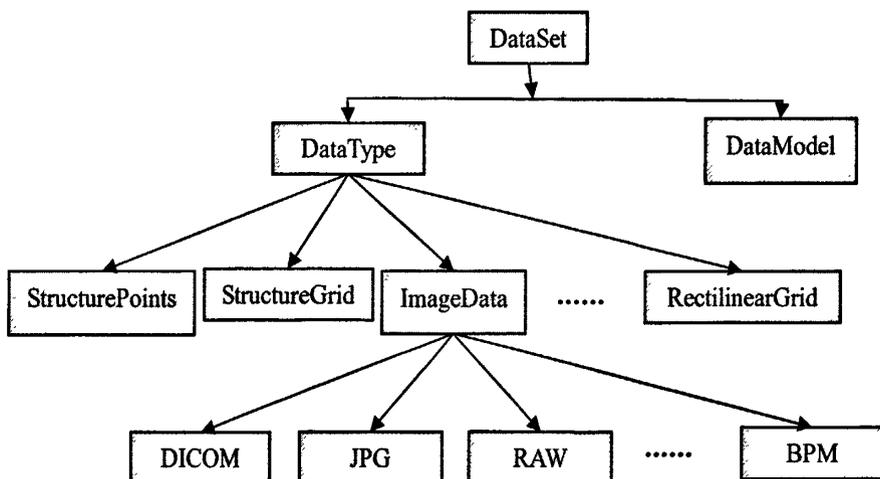


图 3-5 DataType 部分本体概念层次结构图

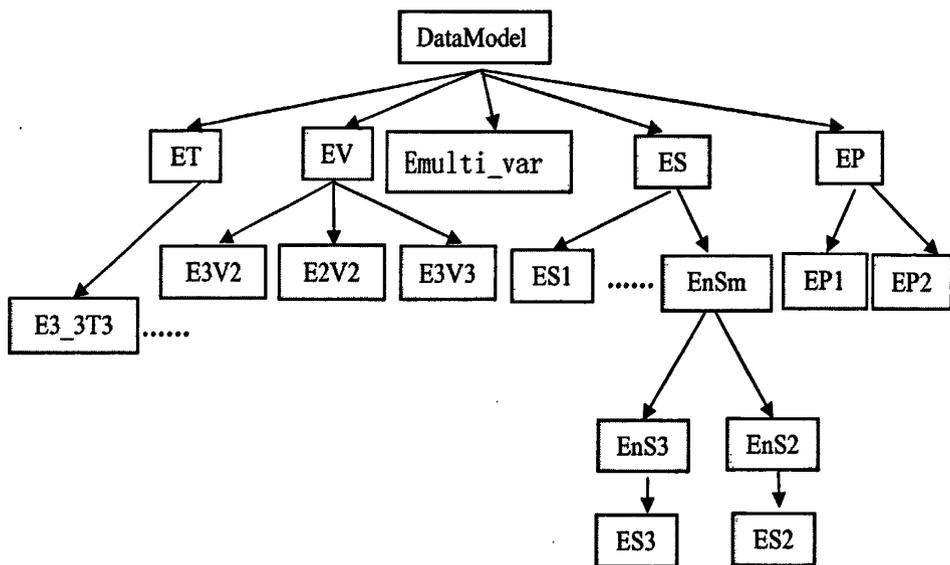


图 3-6 DataModel 部分本体概念层次结构图

其中：EnSm:表示 m 维空间上的一（多）个标量实体。

EnVm:表示 m 维空间上的 n 矢量实体。

En_nTm:表示 m 维空间上的 n*n 的张量实体。

EPm:表示 m 维空间上的点实体。

Emulti_var:表示空间上的点实体。

●Algorithm: 用来定义可视化领域中的各种算法,这些算法是可视化技术的核心,它们主要负责将可视化原始数据转换成用户需要的数据类型,或者直接以图形输出可视化结果。笔者根据可视化 Web 服务发布与匹配的具体需要,及可视化过程鲜明性的特点,将可视化算法按过程分为四类:Reader 类、Filter 类、Mapper 类、Rendering 类和 Integrity 类。其中 Reader 类定义了可视化过程中的数据读取算法,Filter 类对应可视化过程中数据过滤处理,定义可视化领域中的过滤算法,Mapper 类定义了可视化领域中的映射算法,对应可视化过程的映射处理模块。Rendering 类定义了可视化领域中的绘制算法,与可视化过程中的绘制模块相对应。Integrity 类定义的是那些包含整个完整可视化过程数据读取、数据过滤、数据映射和绘制的可视化算法,这是为更好的满足用户对可视化算法发布及使用需求而定义的。Algorithm 的部分本体层次结构如图 3-7。

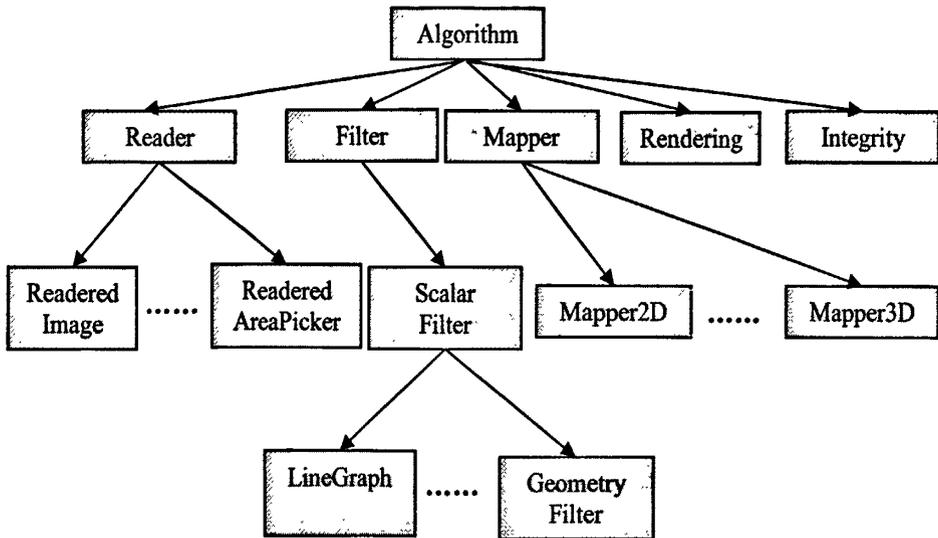


图 3-7 可视化算法部分本体概念层次结构图

3.4.4 可视化本体的实现

3.4.4.1 类及层次结构的实现

本体的描述有很多种，笔者采用 OWL-DL 对可视化本体进行描述。同时当前在本体开发领域有很多本体编辑工具可用，比较流行的有：KAON、Oiled、Ontosaurus、WebOnto 和 Protégé 等。由于 Protégé^[36]是由 java 编写，并且开放源代码，同时支持 OWL、RDF 和关系数据库格式的本体的开发，还能够提供强大的插件支持，其插件支持 Jena 形式的本体推理，所以笔者使用 Protégé 实现可视化本体。通过利用 OWL 类标签，按照前面图 3-4 到图 3-7 中可视化本体结构，很容易在 Protégé 中实现本体中的层次，它是典型的树型结构，根据 Protégé 中默认的 owl:Thing 所建立的类都是其子类，如图 3-8，是可视化本体在 Protégé 中实现的概念层次结构图。

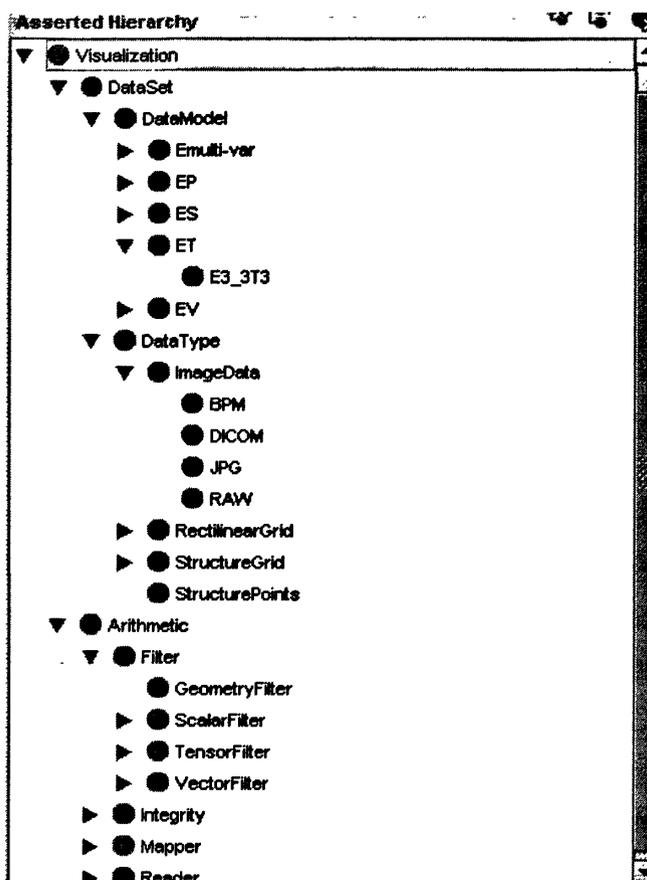


图 3-8 可视化本体类层次结构图

3.4.4.2 属性的实现

OWL 中属性是表达类的实例之间的关系，当定义一个属性时，需要定义以下几个方面：

●属性的类型

Protégé 中属性的类型分为：对象属性（Object property）和数据类型属性（Datatype property），前者表示两个类的实例之间的关系，而后者表达类之间实例与 XML Schema 数据类型值域 rdf 文字之间的关系。

●属性的定义域和值域

因为属性实际上就是表达定义域中的实例与值域中实例之间的关系，无论是哪种类型的属性，其定义都是本体中定义的类，但是对于值域则不同，就对属性而言，其值域是本体中定义的各种类型，但是数据类型属于实例的值域范围则是 rdf 文字或 XML Schema 数据类型中定义的那些简单类型，如：`xsd:int,xsd:float,xsd:double,xsd:decimal,xsd:string,xsd:short,xsd:byte` 等。

据此，可视化本体中定义属性情况如表 3-4。

表 3-4 可视化本体部分属性表

属性名	属性类型	定义域	值域
has-Point	数据类型属性	EP	xsd:int
has-Scalar	数据类型属性	ES	Xsd:int
has-StatepaceSet	对象属性	DataModel	StatespaceSet
transform-form	对象属性	Algorithm	DataModel
transform-into-model	对象属性	Algorithm	DataModel
has-Tensor	数据类型属性	ET	Xsd: int
has-Vector	数据类型属性	EV	Xsd:int
has-input	对象属性	Algorithm	DataType
has-output	对象属性	Algorithm	DataType

3.4.5 可视化本体的一致性检测

在建立好基于 OWL 的可视化本体之后，需要对所建立的可视化本体进行确定和评价，笔者采用一致性检测法，对笔者创建的可视化本体进行检测。一致性检测主要是检测本体间的逻辑关系的一致性。检测过程主要有：首先需要对构建的可视化本体采用 OWL 文件进行存储，然后需要 OWL-API 对可视化本体文件进行读取，并抽取描述文件中各本体概念的信息，最后使用 Jena 推理机^[37]制对所构建的可视化本体进行一致性检测。

至此，整个可视化本体的实现工作就已经完成了。但是本体实现之后，其维护过程是一个循环往复、螺旋式上升的过程，需要在领域专家的协助下不断的完善。

3.5 本章小结

本章主要介绍了可视化技术、本体、本体描述语言 OWL 和可视化本体的研究与实现。首先对可视化技术进行概要性的介绍，然后详细阐述了本体论，包括本体的定义、本体的构成及本体的描述语言 OWL，最后着重介绍了可视化本体的研究与实现。

第 4 章 基于功能语义的可视化 Web 服务匹配算法的研究

4.1 引言

本章首先对现有语义 web 服务匹配算法进行了深入的分析,并归纳出它们存在问题,针对现有语义 Web 服务匹配算法的不足及可视化技术领域自身的特点,提出了一种改进的基于语义距离的个体概念相似度计算方法;然后从基于功能语义匹配角度出发,定义了可视化 Web 服务的描述及发布方法,在此基础上提出了基于功能语义的可视化 Web 服务的总体匹配目标,其主要包括 Service Category 匹配、DataModel 匹配、IO 匹配、textDescription 匹配和综合匹配;最后论文详细介绍上述各匹配阶段的匹配算法。

4.2 语义 Web 服务匹配算法的研究现状

作为 Web 服务应用的基础,Web 服务匹配已经成为了研究的热点问题,并且以查全率、查准率和查询效率作为重要的评价标准。目前国内学者在语义 web 服务匹配方面做了很多有意义的工作。概括起来,目前语义 Web 服务的匹配算法主要分为两种:基于关系推理匹配算法或基于概念相似度计算的匹配算法。

● 基于关系推理的匹配算法

基于关系推理的匹配算法最有代表性的研究是卡梅隆大学的 Paolucci^[4]等人提出的基于输入输出语义 Web 服务匹配算法。该算法是建立在领域本体概念继承关系基础上,考虑服务请求和服务广告输入输出参数在个体概念树中的上下位继承关系,定义了四种匹配度:精确匹配(exact),插入匹配(plugin),包含匹配(subsume)和失败(failed)。其核心匹配算法 Input 为例:

```
inputMatch(inputR,inputA){
    if inputA=inputR then return exact
    if inputA subclassOf inputR then return exact
    if inputR subsumes inputA then return plugin
    if inputA subsumes inputR then return subsumes
    otherwise fail}}
```

算法中的 $inputR$ 是指用户请求的输入, $inputA$ 是指服务广告的输出, $subclassOf$ 表示子类关系, $subsume$ 表示包含关系。以上算法描述为: 当用户请求输入与服务广告输入相同时, 为完全匹配; 当用户请求输入包含服务公告输入是, 为插入匹配; 当服务广告输入包含用户请求输入是为包含匹配; 否则是匹配失败。

这个算法提出时间较早, 它突破了单纯基于关键字的服务匹配思想禁锢, 是基于语义的 web 匹配的新起点。概算法由于引入了服务语义信息, 并基于语义概念本体上下位关系推理, 所以提高了服务的“查全率”和“查准率”。然而该匹配算法任然相对比较粗糙, 匹配度的划分粒度比较大不能进一步细分。而且考虑的匹配范围较小, 只考虑了 Input 信息和 Output 信息。后来也不少学者, 对 Paolucci 等人的进行了匹配等级的扩展, 并引入了属性匹配, 但是目前基于关系推理的概念匹配存在匹配等级粒度加大, 匹配等级有限不能进行更细致的区分的不足。

●基于概念相似度计算的匹配算法

基于概念相似度的匹配算法是通过计算本体概念之间的语义相似度, 其相似度值一般是介于 0-1 之间的数值, 所以相似度匹配算法能够将匹配等级扩展到无限等级, 使服务匹配能得到更精确的匹配结果, 也方便了服务请求者对最优服务的选择。很好的解决了基于关系推理匹配算法划分等级有限的问题。目前, 国内外出现了许多对于基于本体概念相似度计算的匹配算法的研究成果, 但概括起来主要有三个方面: 基于几何距离的相似度计算的匹配算法, 基于信息容量的相似度计算的匹配算法和基于属性相似度计算的匹配算法。

(1) 基于几何距离相似度计算的匹配算法

基于几何距离相似度计算的匹配算法主要是根据概念在层次结构树中的位置来计算概念之间的语义相似性, 一般是直接根据概念在概念层次树中的几何距离来计算语义相似度, 距离越短则相似度越高^[38]。此类算法早期比较有代表性的是 Rada^[39]利用概念层次结构中各概念之间最短相连路径中“边”的数目作为几何距离, 来实现基于几何距离的相似度匹配算法。后来陆续有不少学者对该算法提出了改进方法, 主要是在几何距离的计算中引入了带权路径的计算方法。不过改进算法并没有充分考虑概念在概念结构树中的深度及位置的密度对匹配算法的影响, 而且也没有针对特定领域对基于几何距离相似度计算的研究。

(2) 基于信息容量相似度计算的匹配算法

基于信息容量相似度计算最早是由 Resink 等人^[40]提出来的, 他们认为概念

之间的语义相似性可以通过概念间共享的信息量的大小来衡量，既共享的信息越多，则概念越相似。后来也有不少学者该 Resink 算法理论的基础上提出了很多新的研究成果，但是基于信息论容量的相似度计算方法主要都是通过对文本集的统计分析，来计算出各概念的信息量的大小。他们一般采用下面公式来计算概念的信息量，公式定义如下：

$$IC_r(c) = -\log p(c) \quad (4.1)$$

上式的主要思想是通过计算概念词汇在文本集中出现的概率来计算各概念的信息量。其中 $P(c)$ 表示 C 在文本集中出现的概率，如果层次结构树中只有一个概念节点，则该节点出现的概率为 1。总的来说该定义具有很好的直观性，信息量的增加与概率增长成反方向发展，既概率增加，信息量减少，既概念越抽象其信息量越少。在此信息量计算方法的基础上 Resnik 等人提出了语义概念相似性的定义如下：

$$Sim_r(c_i, c_j) = \max_{c \in S(c_i, c_j)} IC(c) \quad (4.2)$$

从 Resnik 的相似度计算方法中可是看出，随着概念 c_i 和 c_j 最具体共同概念在层次结构树中的层次的深入，它们拥有的信息量也越大，因此概念 c_i 和 c_j 之间的语义相似度也越大。这正好符合了概念在层次结构树中的深度对其概念相似性计算的影响规律。因此可以说基于信息容量的相似度计算方法已经解决了基于几何距离中最短路径距离计算方法的不足。随着更多学者加入到基于信息容量计算的匹配算法的研究中来，也陆续出现了一些改进的基于信息容量计算的匹配算法。

概括起来基于信息容量相似度计算的匹配算法，对于与概念层次结构树的机构信息依赖不强，各概念的信息量是通过基于文本统计的方法来计算的，语义相似度的计算不用考虑概念层次、密度和路径距离的影响。但是因为信息量的计算方法是基于文本统计的，而这种文本统计自身的不足，将影响到相似度的准确性。而且基于信息容量的相似度计算方法所需的前期工作较多，需要在制定本体时需要计算法本体层次结构树中各概念的信息量，并标注在概念属性中，随着本体规模的增大，这将是一项非常耗时的工作，所以基于信息容量的相似度计算的匹配方法实施起来将消耗更多的资源。

(3) 基于属性相似度计算的匹配

基于属性相似度计算的匹配算法是通过依据概念之间的不同于相同属性的多少来计算概念间的相似度。概念间相同的属性越多，则它们越相似，否则则越不相似。基于属性相似度计算的匹配算法最早是由 Tversky^[41]提出来的，他定

义的相似度计算公式如下:

$$Sim(c_1, c_2) = \theta f(c_1 \cap c_2) - \alpha f(c_1 - c_2) - \beta f(c_2 - c_1) \quad (4.3)$$

公式中的 $\alpha, \theta, \beta \geq 0$; $c_1 \cap c_2$ 表示 c_1 和 c_2 的共有属性, $c_1 - c_2$ 表示 c_1 有而 c_2 没有的属性, 同时定义中考虑了 c_1 和 c_2 相同属性和不同属性集对相似度计算影响权重有所不同, 所以定义了权重因子 α, β 和 θ 。通过分配给相同属性或不同属性分配不同的权重策略, 能够使算法模型更具有灵活性, 更好的符合实际规律, 但是现阶段对于如何通过控制三个权重因子的值来达到最佳匹配效果, 仍然处于研究阶段, 还需要研究人员的更多努力。

4.3 改进的基于语义距离的相似度算法

4.3.1 算法思想

通过上面对现有基于语义相似度计算的 Web 服务匹配算法的研究, 可以发现目前的基于语义相似度计算的匹配算法都或多或少存在着各方面的不足, 而且很少看到针对可视化领域的语义相似度匹配算法的研究。随着可视化技术的发展, 及人们对可视化技术需求的增加, 将可视化技术发布为 Web 服务, 并针对可视化领域自身特点设计可视化服务的匹配算法显得越来越重要。通过笔者对现有基于语义相似度计算方法的研究, 同时结合可视化领域本体概念自身的特点, 笔者将全面考虑下面因素的基础上提出改进算法:

(1) 考虑本体概念间的继承关系及二元关系。此处的二元关系是指属性关系。因为可视化领域本体之间存在很普遍的二元关系, 例如 Algorithm 本体概念与 DataModel 本体概念之间就存在 transform from 和 transform into 的二元关系。而且, 同样距离的继承关系和二元关系, 通过继承关系关联的两个概念的相似度要大于二元关系关联的概念相似度。目前考虑概念间二元关系的匹配算法的研究还很少, 针对可视化 Web 服务设计同时考虑继承关系、二元关系的匹配算法的研究尚无报道。

(2) 概念在层次结构树中的所处层次会影响到概念间相似度计算的准确性。一般来讲概念所处的层次越高, 它的抽象程度越高, 相互之间的语义距离越远, 相似程度越低; 而层次越低则其越具体, 概念间的语义距离越近, 因此相似度也越强。

笔者在综合考虑上述因素后, 提出了一种改进的基于语义距离的相似度算

法，该算法兼顾继承关系和二元关系，同时引入了概念层次深度对匹配结果的影响。

4.3.2 算法介绍

改进算法是在基于几何距离的相似度算法的基础上，引入二元关系，及概念层次深度对相似度计算的影响。算法基于带权重的有向图，计算两个概念节点之间的语义距离，进而计算出两者的语义相似度值。考虑到继承关系和二元关系间同样距离的概念之间的相似度不一样，及概念层次深度对相似度计算的影响，笔者引入了不同的权重，因此首先需要对本体图进行满足上述规则的权重分配。基于带权本体有向图，可以方便的计算出两个概念之间的语义距离，但是每次计算时都需要把整个本体图遍历一次，这样会影响算法的执行效率，因此笔者设计了节点路径表来解决重复遍历的问题，每个节点的节点路径表记录了该节点到根节点所有的路径及权重值，这样就可以通过进行数值比较得到概念间语义距离了。为了更好的细分概念匹配的等级，最后需要将语义距离转化为[0-1]区间上的相似度值。算法处理流程如图 4-1。

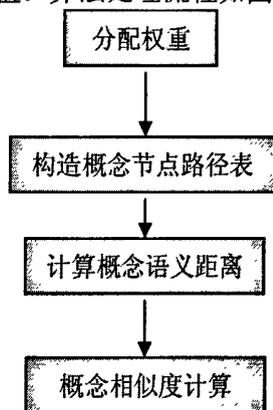


图 4-1 算法处理流程

下面将对算法处理流程中各步骤进行详细的介绍。在开始介绍各处理步骤之前需要做如下约定：

- 用 $distance(c_1, c_2)$ 定义概念 c_1 和概念 c_2 的语义距离。
- 用 $sub(c_1, c_2)$ 表示概念 c_1 和概念 c_2 之间的继承关系，其中 c_2 是 c_1 的父概念。
- 用 $attr(c_1, c_2)$ 表示概念 c_1 和概念 c_2 之间的二元关系，其中 c_2 是 c_1 的属性。
- 关系 r 包含属性关系 $attr$ 和继承关系 sub 。
- 用 $path(c_1, c_2, \dots, c_n)$ 表示节点路径表中的路径记录，其中根节点为 c_n 。
- 概念 c_1 和概念 c_2 存在直接路径关系是指在概念 c_1 或者概念 c_2 的路径表中

显示两个概念相邻。

● 概念 c_1 和概念 c_2 存在间接路径关系是指在概念 c_1 或者概念 c_2 的路径表显示两个概念存在路径相连但不相邻。

4.3.3 分配权重

算法兼顾本体概念间的继承关系和二元关系，继承关系是指概念本体之间的父子关系，二元关系是指概念之间的属性关系。而且考虑到继承关系的语义距离要比二元关系的语义距离小，以及概念之间的语义距离随着深度的增加而减小，所以笔者定义了两个公式分别用来分配概念间二元关系权重和继承关系权重。

$$W[attr(c_1, c_2)] = m * [1 + \frac{1}{f(dep(c_2))}], (m > 1) \quad (4-4)$$

$$W[sub(c_1, c_2)] = 1 + \frac{1}{f(dep(c_2))} \quad (4-5)$$

其中公式 4-4 是二元关系权重分配，公式 4-5 是继承关系权重分配。 $dep(c)$ 表示的是概念在概念层次树中的深度，概念的深度等于它到根节点概念的路径长度，而且规定根节点概念的深度为 0。 $f(dep(c))$ 是 $dep(c)$ 自变量函数，而且 $f(dep(c))$ 是单调递增的。采用不同函数得到的结果差别只在于相似度的数值，而对于整个本体概念相似度的趋势没有影响。在实际应用中具体使用哪个函数与本体图中节点的平均深度有关。公式 4-4 中的 m 是权重调节因子，用于控制二元关系的权重值，论文中暂取为 2。在图 4-2 所示的本体图中，笔者取 $f(dep(c)) = k^{dep(c)}$ 且 $k=2$ ，然后根据上面权重分配公式计算，并将结果标示在图 4-2 中。

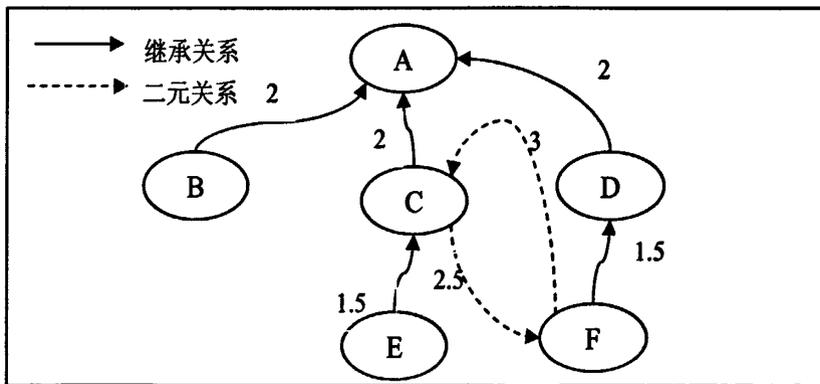


图 4-2 权重分配

4.3.4 构造概念节点路径表

考虑到如果每次计算本体两个概念之间的语义距离时都要去遍历本体概念树的话，这样将降低算法的执行效率，因此笔者提出了构造概念节点路径表来存储概念到根概念节点的所有路径以及每条路径上节点间的关系权重值。通过这个概念节点路径表可以直接计算并比较两个概念节点之间的路径长度来计算出它们的最短语义距离，因而不用每次计算前都对整个本体概念树进行遍历推理，从而大大提高算法的执行效率。

概念节点路径表的构建原则如下：每个节点的路径表由若干条路径记录组成，每条记录包括节点路径和权重两部分，其中节点路径是从该节点到根节点的路径，权重是该条路径上两两节点之间的权重。其构建方法如下：从根节点起进行广度优先遍历，子节点记录的路径部分为其父节点的路径加上本节点构成，权重部分由其父节点的权重列表添上该节点和其父节点之间的权重构成，如图 4-3。

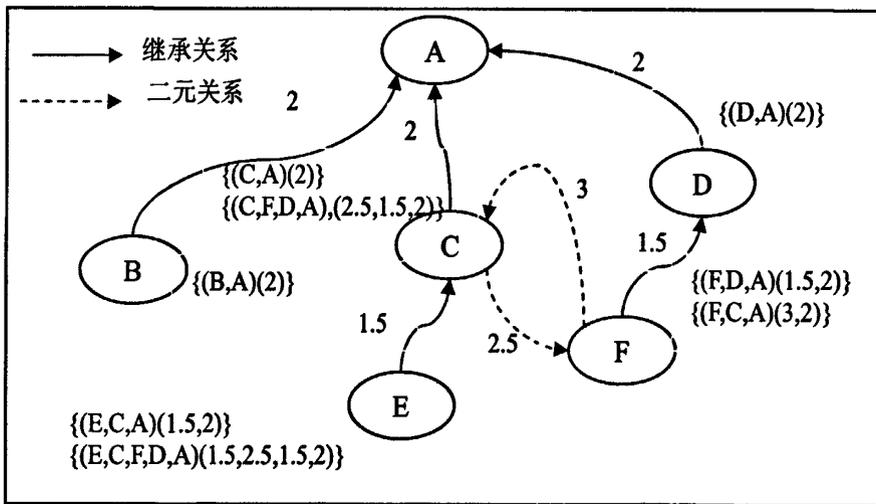


图 4-3 构造概念节点路径表

4.3.5 计算语义距离

对于本体概念树中任意两个概念节点 c_1 和 c_2 之间的语义距离，可以按如下方法进行计算：

- (1) 查看 c_1 和 c_2 是否是同一概念。
 如果是，则 $dis(c_1, c_2) = 0$ ；
 如果否，则转入 2。

(2) 查看 c_1 和 c_2 之间是否存在直接路径关系。

如果是，在 c_1 或 c_2 的概念节点路径表中它们应该是相邻的，则 $dis(c_1, c_2) = \min\{W[r(c_1, c_2)], W[r(c_2, c_1)]\}$;

如果否，转入 3。

(3) 查看 c_1 和 c_2 之间是否存在间接的路径关系。

如果是，那么 c_1 或 c_2 的概念节点路径表中应该存在某些记录如 $path(c_1, c_m, \dots, c_{m+1}, c_2)$

且记 $dis_1(c_1, c_2) = \text{sum}\{W[r(c_1, c_m)] + \dots + W[r(c_{m+1}, c_2)]\}$ ，

如果 c_1, c_2 之间有多条路径，则取加权路径和最小的路径对，

记 $dis_i(c_1, c_2) = \min\{dis_1(c_1, c_2), \dots, dis_n(c_1, c_2)\}$;

同时记 $dis_r(c_1, c_2) = \min\{dis(c_1, c_n)\} + \min\{dis(c_2, c_n)\}$

则 $dis(c_1, c_2) = \min\{dis_i(c_1, c_n), dis_r(c_2, c_n)\}$

如果否，则转入 4。

(4) c_1 和 c_2 之间不存在间接路由关系，语义距离应该等于概念 c_1 和 c_2 分别到根节点的最短语义距离，即 $dis(c_1, c_2) = \min\{dis(c_1, c_n)\} + \min\{dis(c_2, c_n)\}$ 。

例 1: 计算 $dis(E, D)$

满足算法的第三种情况，E 到 D 存在一条间接的路径关系 $\{(E, C, F, D, A) (1.5, 2.5, 1.5, 2)\}$ ，因此 $dis_i(E, D) = 1.5 + 2.5 + 1.5 = 5.5$ ， $dis_r(E, D) = \min\{dis(E, A)\} + \min\{dis(D, A)\} = (1.5 + 2) + 2 = 5.5$ ，所以 $dis(E, D) = \min\{dis_i, dis_r\} = 5.5$ 。

例 2: 计算 $dis(B, E)$

满足第四种情况， $dis(B, E) = \min\{dis(B, A)\} + \min\{dis(E, A)\} = 2 + (2 + 1.5) = 5.5$ 。

4.3.6 概念相似度计算

根据语义距离计算方法计算出两个概念间的语义距离后，需要将概念语义距离转化为概念相似度值，因此相似度计算方法设计的好坏，将直接影响到概念间相似趋势。而且该相似度函数需要满足下面几个主要的特性。

(1) 当语义距离为 0 时，相似度为 1。

(2) 此函数输出的相似度值必须是介于 [0-1] 闭区间内。

(3) 函数随语义距离的增加而减小。既概念间语义距离越大它们之间的相似度就越小。

符合以上三个主要特性的相似度函数有很多，但同时考虑函数下降速率的要求，笔者设计的相似度函数采用公式 4-6:

$$SF = \frac{1}{factor * dis + 1} \quad (factor < 1) \quad (4-6)$$

其中, *factor* 是用来控制语义距离对相似度取值大小的影响, 同样语义距离下 *factor* 取值越大相似度值就越小。直观看来, 合理的 *factor* 取值需要保证在语义距离为 1 时, 相似度在 0.5 以上。

4.4 可视化Web服务的描述及发布

4.4.1 可视化Web服务的描述

Web 服务的描述模型通常是指采用结构化的或半结构化的方式对服务的调用接口、服务功能等相关信息进行描述。规范化的描述模型解决服务发布与服务请求信息异构的关键, 同时也是 web 服务应用的基础。只有将 Web 服务的描述规范化, 才能真正的实现 web 服务资源的共享与交互, 进而实现 web 服务匹配、发现及组合的自动化。

为了更好的实现可视化 Web 服务的发布与匹配, 笔者充分考虑到可视化技术领域的特点, 在 OWL-S 描述框架下, 根据可视化 Web 服务关联的服务基本信息、服务功能信息, 服务的算法和服务的数据模型, 定义了如下的可视化 Web 描述模型: $vWS = \langle WS_info, WS_fun, WS_algorithm, WS_datamodel \rangle$ 。

WS_info 是指服务的基本信息, 包括服务名称、服务的类别, 服务的描述、WSDL 信息和商家信息等服务。

WS_fun 是指服务的功能信息, 其中包括服务的 Input、Output 等。

WS_algorithm 是用于描述服务关联的可视化技术/算法。

WS_datamodel 是用于描述该可视化 Web 服务能处理的数据模型。

虽然 OWL-S 对 Web 服务的语义描述提供了很好的支持, 但是针对论文设计的可视化 Web 服务描述模型中的 *WS_algorithm* 和 *WS_datamodel* 没有直接的描述项, 所以需要对 OWL-S 进行扩展以支持对 *WS_algorithm* 和 *WS_datamodel* 的描述。

OWL-S 规范中 *ServiceProfile* 为用户提供 *ServiceParameter* 类, 用户可以通过扩展 *ServiceParameter* 类来为服务添加其他的描述信息。因此笔者将在 *ServiceProfile* 的 *ServiceParameter* 中添加服务的 *Algorithm* 属性参数 *hasAlgorithm*, 其中包括 *ws_algorithm* 和 *ws_datamodel*, 分别描述服务所使用的算法及能处理的数据模型, 扩展的描述模式如图 4-4。

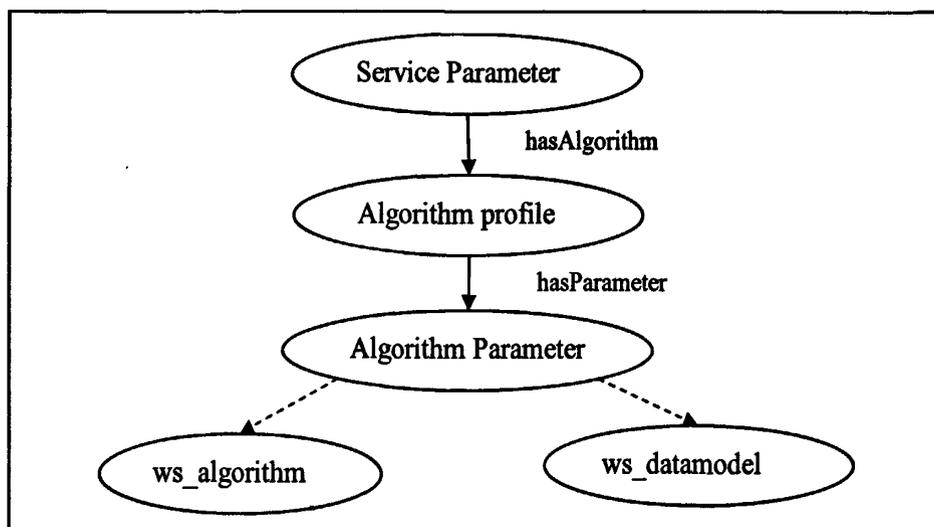


图 4-4 扩展的描述模式

4.4.2 可视化Web服务的发布

可视化 Web 服务的发布就是服务提供者通过系统提供的接口注册他们开发的服务，从而实现可视化服务资源的共享。而可视化过程一般要经过这样几个过程：数据获取、数据预处理、映射、绘制显示，如图 4-5。

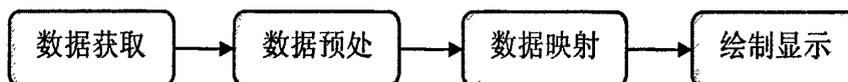


图 4-5 可视化过程

考虑到可视化技术具有过程鲜明的特点，笔者将可视化 Web 服务按其过程分为四类：Reader 服务、Filter 服务、Mapper 服务和 Rendering 服务，它们分别与可视化处理过程的数据获取、数据预处理、数据映射和绘制相对应。同时考虑到可视化应用的具体需要定义一个 Integrity 服务，它定义的是那些包含一个完整可视化过程数据读取、数据过滤、数据映射和绘制的可视化服务。

基于上面对可视化服务的分类方法，及可视化 Web 服务的描述模型，用户在发布时，用户根据自己的可视化资源的具体情况，选择将其发布为 Reader 服务、Filter 服务、Mapper 服务、Rendering 服务或 Integrity 服务，并按描述模型的规范进行服务描述。

4.5 基于功能语义的可视化Web服务匹配的总体目标

笔者是在基于 OWL-S 语义 Web 服务描述框架下,研究基于功能语义的可视化 Web 服务匹配。而基于功能语义的 Web 服务的匹配度是由 Service Profile 中的 Service Category、textDescription、Input 和 Output 一起决定的。其中 Service Category 是对服务类别描述; textDescription 主要是用于对服务功能的概要描述; Input 是用于描述服务的输入参数; Output 是用于描述服务的输出参数。同时考虑到可视化领域自身的特点,比如数据模型与可视化算法之间通常存在密切关系的,因此笔者还增加了可视化 Web 服务的 DataModel 的匹配。而且论文考虑到服务请求与服务发布之间可能存在语义描述异构,所以设计 Input、Output、DataModel 是基于本体概念的匹配的, Service Category 是基于字符串匹配的, textDescription 则及文本向量空间法进行匹配的。以下笔者将对 Service Category 匹配、textDescription 匹配、IO 匹配, DataModel 匹配进行详细的介绍。

4.6 基于功能语义的可视化Web服务匹配算法

4.6.1 Service Category 匹配

Service Category 匹配是服务类型的匹配,在多层次、多阶段匹配系统中,服务类型匹配能够根据请求服务的类别对候选服务库进行初步筛选,过滤掉大部分不相干的服务类,这样将大大的提高服务匹配系统的匹配效率。而笔者根据可视化具有过程鲜明的特点,将可视化算法按其过程发布为服务并分为四类:Reader 服务、Filter 服务、Mapper 服务和 Rendering 服务,同时考虑到用户对可视化服务发布与请求的特殊需求外加一类 Integrity 服务,所以基于笔者构建的可视化本体, Service Category 将有五种取值。

考虑到 Service Category 取值的有限性及固定性,笔者在可视化 Web 服务的 OWL-S 描述文件中直接采用字符串对其 Service Category 进行标示。而且在研究可视化 web 服务的发布机制时,按照服务描述的 Service Category 信息进行分类存储。基于上述事实,此处 Service Category 匹配伪代码描述如图 4-6。

```

输入: req_category // service Category 描述字符串
输出: services //候选服务集
if req_category equals Reader
    return services =Reader_collection;
if req_category equals Filter
    return services =Filter_collection;
if req_category equals Mapper
    return services =Mapper_collection;
if req_category equals Rendering
    return services=Rendering_collection;
if req_category equals Integrity
    return services=Integrity_collection;
else return services=null;
end

```

图 4-6 Service Category 匹配伪代码

4.6.2 textDescription匹配

在 OWL-S 服务描述框架下, textDescription 是 service profile 的一个子节点, 主要用于对于服务进行自然语言注释。考虑到可视化 web 服务的请求用户, 对自己需要获取的服务通常只有模糊的概念, 而且往往可能只知道需请求的服务大概功能, 因此它对服务的匹配影响是不容忽视的。基于上述事实, 笔者将 textDescription 匹配引入到论文设计的多层次匹配系统中, 这样能够为用户匹配到更加理想的服务。笔者约定此处的 textDescription 主要是通过自然语言对服务功能的概要描述, 而且此处的功能描述不包括 IO 的相关信息。

因为 textDescription 是基于自然语言的描述, 因此笔者将采用信息检索领域中的向量空间模型来完成服务 textDescription 文本信息的匹配。下面将详细介绍向量空间模型。

向量空间模型(Vector Space Model)是信息检索领域常用的一种文本检索模型^[42]。它首先将文档分割成一组单词, 并取所有单词组成的集合 (t_1, t_2, \dots, t_n) , 其中 n 表示集合中单词的个数。而且这个集合中消除了填充字符, 合并了派生词。

基于向量空间模型, 任何一个文档都表示成 n 维向量, 例如文档 T 可以表示为 $T = \{t_1, t_2, \dots, t_k, \dots, t_n\}$, 其中 n 是指文档的特征项的个数, 而且项 t_k 通常具有一定的权重 w_k , 它表示对于文档该特征项的重要程度, 因此可以把文档 T 表示为 $\{t_1, w_1, t_2, w_2, \dots, t_n, w_n\}$, 或者也可以表示为 $\{w_1, w_2, \dots, w_n\}$ 。如果把 t_1, t_2, \dots, t_n 看做坐

标系的话, 则 w_1, w_2, \dots, w_n 便是与其对应的坐标值, $T\{w_1, w_2, \dots, w_n\}$ 便可以理解为 n 维空间中的一个向量。用 W_{ik} 表示文档 i 中单词 k 的权重, 计算公式如下:

$$W_{ik} = tf_{ik} \times idf_k = tf_{ik} \times \log_2(N / df_i) \quad (4-7)$$

其中 tf_{ik} 表示单词 k 在文档 i 出现的次数, N 表示文档集中所有文档的总数, df_i 是文档集中包含单词 k 的所有文档的总数。因为我们每次匹配的都是两个文档之间的匹配, 所以 $N=2$ 。

基于上述理论, 两个文档可以分别用向量 T_i 和 T_j 表示, 则可以采用这两向量的余弦距夹角来计算它们的相似度 $Sim(T_i, T_j)$:

$$Sim(T_i, T_j) = \frac{\sum_{k=1}^n w_{ik} \times w_{jk}}{\sqrt{\left\{ \sum_{k=1}^n w_{ik}^2 \right\} \times \left\{ \sum_{k=1}^n w_{jk}^2 \right\}}} \quad (4-8)$$

其中 w_{jk} 表示单词 k 在文档 T_j 中的权重, w_{ik} 表示单词 k 在文档 T_i 中的权重。

我们将服务 service profile 中的 textDescription 抽取出来, 把其看文档, 将可以应用上面的基于向量空间的相似度计算方法来计算服务请求与服务广告之间 textDescription 的匹配相似度了。

4.6.3 DataModel 匹配

可视化技术有其鲜明的过程性, 其执行过程一般包括数据读取、数据过滤、映射和绘制显示。而笔者也正是考虑到可视化技术的这个特征及服务执行的需要, 将可视化化算法按期执行过程分别发布为可视化 Web 服务, 其中主要包括 Reader 服务、Filter 服务、Mapper 服务、Rendering 服务和 Integrity 服务。而且, 可视化技术还有一种重要特征就是可视化算法与可视化数据实体之间有着密切联系, 我们可以根据数据实体所属的数据模型类别, 来找能处理该数据模型的可视化算法。而且在平时的服务请求中, 用户通常对自己需要调用的可视化算法不太了解, 但是对自己数据的数据类型及数据模型类别还是比较了解。

基于上述事实, 笔者将 DataModel 匹配引入了可视化 web 服务的描述模型中, 如 4.3 节所示, 并设计了 DataModel 匹配。

DataModel 匹配的主要思想: 基于笔者构建的可视化本体库, 通过从服务请求和服务广告中抽取出 req_DataModel 和 adv_DataModel 概念语义信息, 然后采用本章 4.2 提出的改进算法进行基于本体语义距离相似度的计算, 从而计算出服务 DataModel 匹配相似度。

4.6.4 IO匹配

服务 IO 是服务输入、输出参数的描述, 是服务描述及服务接口调用的重要信息。因此 IO 匹配一直都是服务功能匹配的重点。而且可视化技术本身和其参数信息有着非常密切的联系, 例如不同的 Reader 服务可以读取不同类型的文件, 在 VTK 中 DICOMReader 只能读取 DICOM 文件, JPGReader 只能读取 JPG 文件。而且 Mapper 服务按照它处理数据实体的不同可以分为多种类型, 譬如 PolyData 则需要用 PolyDataMapper 处理。所以输入、输出参数信息的匹配是可视化 web 服务匹配的重点之一。

同时笔者考虑传统 IO 匹配中, 因为服务请求与服务发布之间的描述异构对匹配准确性有很大的影响, 而且缺乏语义信息的解析, 所以查全率也不高, 为此笔者设计的 IO 匹配时基于本体概念之间的概念相似性的匹配, 因此采用了本体概念来描述 IO 参数信息, 这样一定程度上消除由于服务描述异构带来的问题, 提高服务查询的“查全率”和“查准率”。

IO 匹配算法的过程描述如下:

(1) 基于可视化本体库, 根据从服务请求中解析出来的服务 input 概念信息, 用笔者提出的语义距离相似度计算方法, 计算服务请求 input 概念与服务广告中的 input 概念匹配的相似度值 Sim_{input} , 并按预设阈值 k_{input} , 过滤掉 $sim_{input} < k_{input}$ 的服务, 经过 input 匹配筛选得到 servicesByInputs 候选服务集合。

(2) 根据从服务请求中解析出来的服务 output 概念, 用笔者提出的语义距离相似度计算方法, 分别计算服务请求 output 与服务广告集 servicesByInputs 中各服务的 output 概念匹配相似度 Sim_{output} , 并按预设阈值 k_{output} , 过滤掉 $Sim_{output} < k_{output}$ 的服务, 经过 input 匹配和 output 匹配的赛选后得到 servicesByIOs 的候选服务集合。并按 $Sim_{IO} = \min\{Sim_{input}, Sim_{output}\}$, 计算出 servicesByIOs 中每个服务的 IO 匹配相似度 Sim_{IO} 。

4.6.5 可视化web服务综合匹配

服务综合匹配是服务匹配度排序的重要依据, 也是客户选择理想服务的综合指标。因此服务综合匹配方法设计的好坏将直接影响到服务匹配系统的服务质量。

笔者设计的总体匹配目标中包括 Service Category 匹配、IO 匹配、text Description 配和 DataModel 匹配。同时考虑到可视化技术领域自身的特点, 数据

读取服务和映射服务与数据类型的关系比较突出，而过滤服务与数据模型关系非常密切，所以笔者针对不同服务类型设计了两条匹配路线。对于 Reader 和 Mapper 服务将依次进行：Service Category、IO 和 text Description 匹配；对于 Filter、Rendering 和 Integrity 服务将依次进行：Service Category、DataModel、IO 和 textDescription 匹配。因此针对这两条匹配路线，笔者给它们对应的服务综合匹配度的计算方法：

$$Sim = W_{dm} * Sim_{DataModel}(ws_r, ws_a) + W_{io} * Sim_{IO}(ws_r, ws_a) + W_{tx} * Sim_{text}(ws_r, ws_a) \quad (4-10)$$

公式 4-10 为可视化 web 服务综合匹配度计算公式，其中 W_{dm} 、 W_{io} 和 W_{tx} 分别表示 DataModel 匹配、IO 匹配和 textDescription 匹配在服务层次匹配中权重，具体取值由用户根据具体情况而定，但需要满足下面条件：

(1) 当请求匹配服务是 Reader 或 Mapper 服务时需满足： $W_{dm} + W_{io} + W_{tx} = 1$ ， W_{io} 和 W_{tx} 均处于(0-1]区间，且 $W_{dm} = 0$ 。

(2) 当请求服务类别是 Filte、Rendering 或 Integrity 服务时需满足： $W_{dm} + W_{io} + W_{tx} = 1$ ，且 W_{dm} 、 W_{io} 和 W_{tx} 均处于(0-1]区间。

4.7 本章小结

本章首先对现有语义 web 服务匹配算法进行了深入的分析，并归纳出它们存在问题，针对现有语义 Web 服务匹配算法的不足及可视化领域自身的特点，提出了一种改进的基于语义距离的本体概念相似度计算方法；然后从基于功能语义匹配角度出发，定义了可视化 Web 服务的描述及发布方法，在此基础上提出了基于功能语义的可视化 Web 服务的总体匹配目标，其主要包括 Service Category 匹配、DataModel 匹配、IO 匹配、textDescription 匹配和综合匹配；最后论文详细介绍上述各匹配阶段的具体匹配算法。

第5章 可视化 Web 服务发布与匹配模型的设计与实现

5.1 引言

本章主要介绍可视化 Web 服务发布与匹配模型的具体设计与实现, 首先阐述了该模型的主要设计思想, 然后从宏观上说明了该模型的层次结构, 接着对模型层次结构中的各功能模块, 及基于该模型的可视化 Web 服务发布流程和发现流程进行了详细的介绍, 最后描述了基于 J2EE 实现模型原型系统的详细过程。

5.2 设计思想

可视化 Web 服务发布与匹配原型系统需要能够满足服务发布者对可视化 Web 服务发布的需要, 同时也要能够满足服务请求者对可视化 Web 服务查询的需求。可视化 Web 服务发布与匹配原型系统需要提供两方面的基础服务, 一方面为服务发布者提供服务发布的接口, 根据用户提供的服务相关信息, 完成可视化 Web 服务的发布, 另一方面为服务请求者提供服务查询的接口, 根据服务请求者对自己数据集、及其希望获得的服务相关信息的描述, 完成可视化 Web 服务的后台匹配作业, 并根据匹配相似度对所有匹配结果进行排序输出, 为服务请求者提供更好的服务查询支持。

在基于 OWL-S 服务描述框架下, 通过对可视化技术具有鲜明过程性及可视化算法与可视化数据实体之间具有密切关联等特性的研究, 笔者提出了可视化 Web 服务发布与匹配模型的主要设计思想:

(1) 服务发布

对于可视化 Web 服务发布, 笔者根据可视化技术具有鲜明过程性, 及可视化 Web 服务发布的特殊需求, 将可视化 Web 服务分为五类: Reader 服务、Filter 服务、Mapper 服务、Rendering 服务和 Integrity 服务, 而且针对每类服务都提供了相应的服务发布接口, 为此笔者设计了一个服务注册器来完成可视化 Web 服务的发布。

(2) 服务匹配

在 OWL-S 服务描述框架下, 以可视化特性为基础, 围绕提高服务匹配“查

全率”、“查准率”和“匹配效率”的目标。笔者通过充分挖掘可视化 Web 服务的功能语义信息，并结合可视化领域本体库，采用了分类别、多阶段的层次过滤匹配方法。在该多阶段匹配过程中将涉及：Service Category 匹配、DataModel 匹配、IO 匹配和 textDescription 匹配。为此在匹配原型系统中，笔者相应的设计了 Service Category 匹配器、DataModel 匹配器、IO 匹配器和 textDescription 匹配器。同时考虑到可视化 Web 服务发布与匹配原型系统不应该是仅提供服务的匹配，应该是在匹配结果的基础上，为用户提供理想的服务选择方案，因此笔者设计了服务匹配排序器，根据匹配相似度结果及相关信息对服务进行规则排序。

5.3 模型层次设计

5.3.1 模型总体结构

基于上述的可视化 Web 服务发布与匹配模型设计思想，模型对应的层次结构设计如图 5-1。

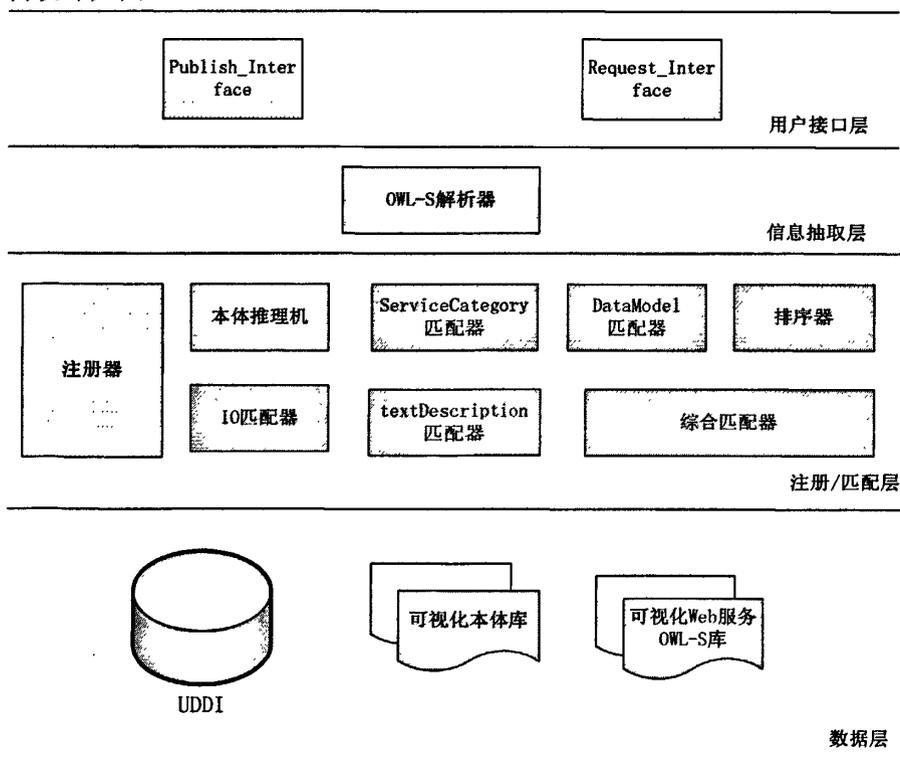


图 5-1 模型层次结构图

由图 5-1 可以发现可视化 Web 服务发布与匹配模型的框架主要分成四层，

从下往上依次是数据库层、注册/匹配层、信息抽取层和用户接口层。下面分别对各层进行详细的介绍。

5.3.2 数据库层

数据库层位于模型层次结构的最低端，为原型系统的运行提供数据服务支撑。它主要包括 UDDI 服务注册中心、可视化本体库和可视化 Web 服务 OWL-S 库。

●UDDI: 主要是用于存储服务的基本信息，不包括服务的语义信息，而且此处使用 UDDI 注册数据库也能实现语义 Web 服务和传统 Web 服务的兼容。

●可视化本体库: 主要是存储可视化技术领域当中的一本体概念，包括可视化算法及数据模型概念等，为以上服务匹配及语义推理提供了统一的概念，可以消除服务描述的异构问题。

●可视化 Web 服务 OWL-S 库:采用 OWL-S 文档集的方式来存储语义可视化 Web 服务。

5.3.3 注册/匹配层

注册匹配层是可视化 Web 服务发布与匹配模型的核心层，在数据库层的支持下，通过接受信息抽取层提供的服务相关信息，完成服务的发布及服务的匹配查询。它主要包括下面几个功能模块：注册器、本体推理机、Service Category 匹配器、DataModel 匹配器、IO 匹配器、textDescription 匹配器和综合匹配器和排序器模块。

●注册器模块: 在服务发布时，根据信息抽取层提供的关于服务的基本信息及语义信息，完成将服务基本信息注册到数据库层的 UDDI 库中，根据 OWL-S 语义 Web 服务的描述规则，构建及规范化可视化 Web 服务的 OWL-S 文档，并将其存储在可视化 Web 服务 OWL-S 集中。

●本体推理机模块: 在可视化本体库及服务语义描述 OWL-S 支持下，本体推理机可以通过 OWL-API 抽取中本体描述中的本体概念信息，进而采用推理机制可以计算出本体概念间的语义距离，为其他服务匹配提供的数据准备，是服务匹配中的一个重要关键点。

●Service Category 匹配器模块: Service Category 匹配是基于简单字符串的匹配，它能够很好的过滤掉大部分与服务请求不相干的服务，可以大大提高原

型系统的执行效率。

●DataModel 匹配器模块: DataModel 匹配充分考虑到可视化技术与可视化数据模型之间的密切关联性, 采用笔者改进的基于语义距离相似度算法, 完成对服务 DateModel 之间的匹配。

●IO 匹配器模块: IO 匹配是服务功能匹配的重要组成部分, 采用了笔者改进的基于语义距离相似度算法, 实现对服务 IO 之间的 IO 匹配。

●textDescription 匹配器模块: textDescription 匹配是指对服务请求与服务广告之间的服务功能描述信息进行匹配, 它主要采用传统信息检索领域的向量空间模型来进行匹配相似度计算的, 它很好的捕获了用户对服务功能的概要描述信息, 在一定程度上提高了服务匹配的“查准率”和“查全率”。

●综合匹配器模块: 根据笔者设计的分类别多阶段的匹配策略, 综合匹配将根据具体匹配的服务类别, 利用 IO 匹配、DataModel 匹配和 textDescription 匹配的结果, 进行加权的综合匹配相似度计算。为排序器模块提供数据支持。

●排序器模块: 排序器负责对经过各阶段服务匹配筛选而保留的服务按照排序策略进行排序输出。首先按照服务综合匹配相似度排序, 如果综合匹配相似度相同, 则按照 DataModel、IO 和 textDescription 的优先顺序对各部分的相似度大小排序, 如果四个相似度都相同则按同一序号输出。

5.3.4 信息抽取层

信息抽取层主要是从服务发布接口或服务查询接口提交的服务相关信息中提取出服务的基本基本信息、服务的分类信息、服务的 IO 信息、服务 textDescription 描述信息, 并将这些信息提交给模型的注册/匹配层。而且信息抽取层功能的实现完全依赖 OWL-S 解析器。

●OWL-S 解析器: 因为笔者在服务用户接口中为用户提供了两种服务信息提交的方法, 一种是基于用户接口录入项录入服务相关信息, 另一种是允许用户以 OWL-S 文件的格式描述服务相关信息, 并直接提交该 OWL-S 文件。所以 OWL-S 解析器是基于 OWL-S API, 能够直接从 OWL-S 文件中抽取出的服务的相关信息, 同时也能根据用户接口中各录入项的信息, 抽取出的服务的相关信息。

5.3.5 用户接口层

用户接口层是可视化 Web 服务发布与匹配模型的最外层, 它提供了模型与

外界交互的接口。它分为服务发布接口和服务查询接口。

●服务发布接口：服务发布者通过使用发布服务接口，可以提交服务的 OWL-S 描述文档，也可以直接在客户端界面上按服务发布要求分别录入服务描述的各项信息，信息提交后，发布服务接口将用户提交的信息提交给信息抽取层，经信息抽取层抽取服务各项描述信息后，再将抽取信息传送给注册器完成服务的注册，及服务 OWL-S 规范描述。

●服务查询接口：服务请求者通过服务查询接口提供的接口，可以提交服务的 OWL-S 描述文档，也可以直接在客户端界面上按服务查询要求分别录入服务描述的各项信息，信息提交后，服务查询接口将用户提交的信息提交给信息抽取层，经信息抽取层抽取服务各项描述信息后，在将抽取信息传送给模型中的 Service Category 匹配器，然后开始服务的匹配作业，最后由排序器将匹配结果通过服务查询接口反馈给服务请求者。

5.4 可视化Web服务发布与匹配流程设计

5.4.1 可视化Web服务发布流程

可视化 Web 服务发布过程重要包括下面几个步骤：

(1) 用户登陆验证。服务发布者使用本模型发布服务之前，需要采用在本系统注册的用户名和密码登陆，如果是新用户需要注册自己的用户信息获取合法使用账号及密码，在用户提交用户名和密码后原型系统后台将验证用户的登陆信息的合法性如果通过验证，系统将为用户并分配相应的使用权限。

(2) 录入服务相关信息。服务发布者经过服务登陆认证后，即可在服务发布接口提供的客户端上发布服务了，服务发布接口需要用户输入关于服务的以下信息：服务名称、服务描述、服务类别、商家信息、服务可处理的数据模型、服务采用可视化算法、服务发布 WSDL、服务输入输出参数信息和服务 OWL-S 上传等。其中服务的输入输出参数、服务采用的算法和服务可处理的数据模型都是基于本体的下拉列表选择录入，这样能够很好的消除服务请求者与服务发布者对服务描述的异构问题；另外 OWL-S 文件上传是为了满足一些专业用户的需要，因为有些专业用户对 OWL-S 描述规则非常熟悉，所以可能希望直接采用 OWL-S 文件上传。

(3) 发布验证。服务发布者录入完服务各项相关信息后，并选择提交信息后，服务后台将对用户输入的各项信息进行合法性验证，如果输入的各项信息

都满足系统相应要求，则 OWL-S 解析器将对服务各项描述信息进行抽取、规范化，然后将数据移交给注册器，由注册器完成服务的发布任务。

5.4.2 可视化Web服务匹配流程

笔者在设计可视化 Web 发布时是充分考虑到可视化技术过程的鲜明性及服务发布特殊需求，因此将服务分为五种：Reader 服务、Filter 服务、Mapper 服务、Rendering 服务和 Integriy 服务，利用服务分类信息可以大大的缩小服务匹配的范围，提高服务的匹配效率，而且如果用户请求和服务广告库中的每个服务都依次进行 DataModel、IO、textDescription 和综合匹配，再根据匹配相似度结果进行排序，这样将会降低原型系统的匹配效率。

因此笔者设计了一种分类别多阶段的匹配策略，其服务匹配流程，如图 5-2。

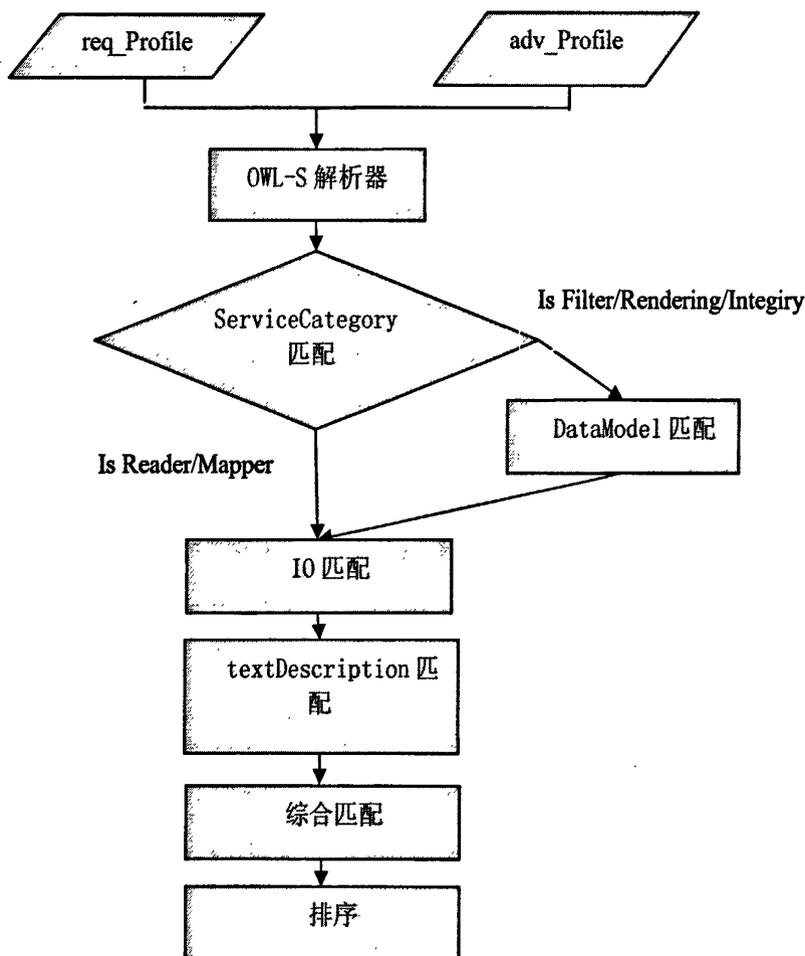


图 5-2 可视化 Web 服务匹配流程图

首先通过服务类别匹配, 淘汰大部份不相关的服务, 缩小服务匹配范围; 然后再依次进行 DataModel、IO、textDescription 和综合匹配, 并且在每个服务匹配阶段都按照预设的阈值 M 对候选服务进行过滤, 如果匹配相似度小于该阶段预设阈值 M 将被过滤掉, 不能进入下一阶段的匹配, 这样将很好的提高原型系统的匹配效率。

同时笔者还考虑到 Reader 服务和 Mapper 服务与输入输入参数有密切的关系, 而 Filter 服务、Rendering 服务和 Integrity 服务与 DataModel 之间有着密切关联, 所以笔者基于可视化特性设计了两条匹配路线: 一条是 Filter、Rendering 和 Integrity 服务匹配路线, 它包括 Service Category、DataModel、IO、textDescription 和综合匹配; 另一条是 Reader 和 Mapper 服务匹配路线, 它主要包括 Service Category、IO、textDescription 和综合匹配。对于过滤阈值, 原型系统预设了 M_d 、 M_{io} 、 M_t 和 M_F 分别表示 DataModel、IO、textDescription 和综合匹配各阶段的阈值。各阈值的取值范围都在 [0-1] 闭区间上, 其具体取值决定了模型在各匹配阶段的过滤粒度及对模型匹配效率的提高程度。当取 0 时表示不过滤, 这样对匹配效率的没有任何改善, 当取 1 时表示指保留完全匹配。当 M 的值越大时, 将有更多不符合要求的服务被过滤掉, 参与最终匹配的不相关服务就越少, 将大大的提高模型的匹配效率; 但是, 过大 M 在一定程度上消减了服务匹配的“查全率”。因此笔者为用户提供设置 M 值的接口, 用户可以自己根据自己的实际需要定义各阶段的预设阈值。同时, 如果用户自己不清楚如何设置各阶段的阈值, 可以将各阶段的阈值预设设在 [0.5-0.8] 闭区间上, 这样既可过滤掉大部份与服务请求不相干的服务, 而且也能提高模型的匹配效率。

用户使用本模型进行服务匹配的具体步骤如下:

(1) 用户登陆验证。服务请求者使用本模型查询服务之前需要, 采用在本模型注册的用户名和密码登陆, 如果是新用户需要注册自己的用户信息获取合法使用账号及密码, 在用户提交用户名和密码后原型系统后台将验证用户的登陆信息的合法性如果通过验证, 模型将为用户并分配相应的使用权限。

(2) 录入服务查询信息。服务请求者经过服务登陆认证后, 即可在服务查询接口中输入关于服务请求的一些信息: 服务名称、服务描述、服务类别、商家信息、服务可处理的数据模型、服务发布 WSDL、服务输入输出参数信息和服务 OWL-S 上传等。其中服务输入输出参数和服务可处理的数据模型都是基于本体的下拉列表选择录入, 这样能够很好的消除服务请求者与服务发布者对服务描述的异构问题; 另外 OWL-S 文件上传是为了满足一些专业用户的需要, 因

为有些专业用户对 OWL-S 描述规则非常熟悉, 所以可能希望直接采用 OWL-S 文件上传。

(3) 服务查询信息验证。服务请求者录入完服务查询各项相关信息后, 并选择提交信息后, 服务后台将对用户输入各项信息进行合法性验证, 如果输入的各项信息都满足系统相应要求, 则 OWL-S 解析器将对服务各项描述信息进行抽取、规范化, 然后将数据移交给 Service Category 匹配器, 开始服务的具体匹配工作。

(4) 查看查询结果。模型的排序器模块将根据服务各阶段及综合排序的结果, 采用设定的排序策略将匹配服务按相似度大小排序输出, 同时用户也可以查看到每个输出服务各阶段匹配的具体相似度值。这样为服务请求者查询理想服务提供了有力帮助。

5.5 模型的实现

5.5.1 实现概述

模型采用 J2EE 的轻量级开发方法 SSH 来实现, 该开发方法是基于 B/S 模式的, 采用 MVC 架构, 其中 M(模型层)用 Hibernate 实现, V(视图层)采用 JSP 和 html 实现, C(控制器)层采用 Struts 实现。通过采用 IE 浏览器做为客户端, 使用户可以方便的实现可视化 Web 服务资源的发布、查询、管理和共享。采用 MySQL 作为数据库管理系统。基于 J2EE 技术的实现原型系统将具有更高可扩展性和跨平台性, 而且便于系统的升级。

5.5.2 开发环境的搭建

因为笔者是基于 J2EE 的轻量级开发方式实现原型系统, 所有主要的开发工具如下:

- (1) 一台 CPU 为 Intel(R) Core (TM) 2 Duo, 主频为 2.1GHz, 内存为 2G, 采用 Windows 7 操作系统的 PC 机。
- (2) J2EE 工具包: 主要包括 jdk-6u20-windows-i586、struts2.0.jar, 因为笔者是采用基于 struts 的 J2EE 的轻量级架构实现可视化 Web 服务发布与匹配的原型系统。
- (3) 数据库工具: MySQL5.0, 用来实现对 UDDI 服务注册库及缓存库创建和

管理。

(4) JUDDI 开发包: Apache uddi4j, 用来搭建系统的 UDDI 私有服务注册中心, 同时也利用其提供的与 UDDI 注册库交互的 API, 服务基本信息的存储及相关操作。

(5) 本体开发工具: Protege3.4.4, 用来构建可视化本体库。

(6) 语义推理工具: Jena2.6.4、OWL-S API, 用于实现对本体概念的解析, 及概念之间关系推理。

(7) Web 服务器: Apache Tomcat6.0, 用来部署原型系统的实现平台。

(8) 服务发布工具: AXIS 2.0^[43], 用于部署开发的可视化 Web 服务。

5.5.3 数据库设计

基于前面设计的可视化 web 服务发布与匹配模型, 本原型系统中的数据库层主要包括: UDDI 注册库、可视化本体库和可视化 Web 服务描述 OWL-S 库。

● UDDI 注册库: 笔者采用 JUDDI 来构建私有的 UDDI 注册中心, 主要用于存储可视化的商家信息和服务发布者的注册信息等与服务相关的基本信息, 并且是采用 MySQL 作为原型系统的数据库管理系统, 用于对 UDDI 中的信息存取进行有效管理, 数据库部分 CDM 设计如图 5-1。

● 可视化本体库: 主要是存储可视化领域当中的一本体概念集, 包括可视化算法及数据模型概念等, 服务匹配及语义推理提供了共同的本体概念, 可以消除服务描述的异构问题。笔者直接采用文档管理的方式存储可视化本体 OWL 文档集。

● 可视化 Web 服务 OWL-S 库: 为了便于可视化服务信息的抽取, 及逻辑推理, 实现系统采用文档管理方式来保存可视化 Web 服务 OWL-S 文档集, 每个 OWL-S 对应着一个可视化 Web 服务。

对于可视化本体库和可视化 Web 服务 OWL-S 库将直接采用 OWL-S API 进行相关的访问操作及数据维护。

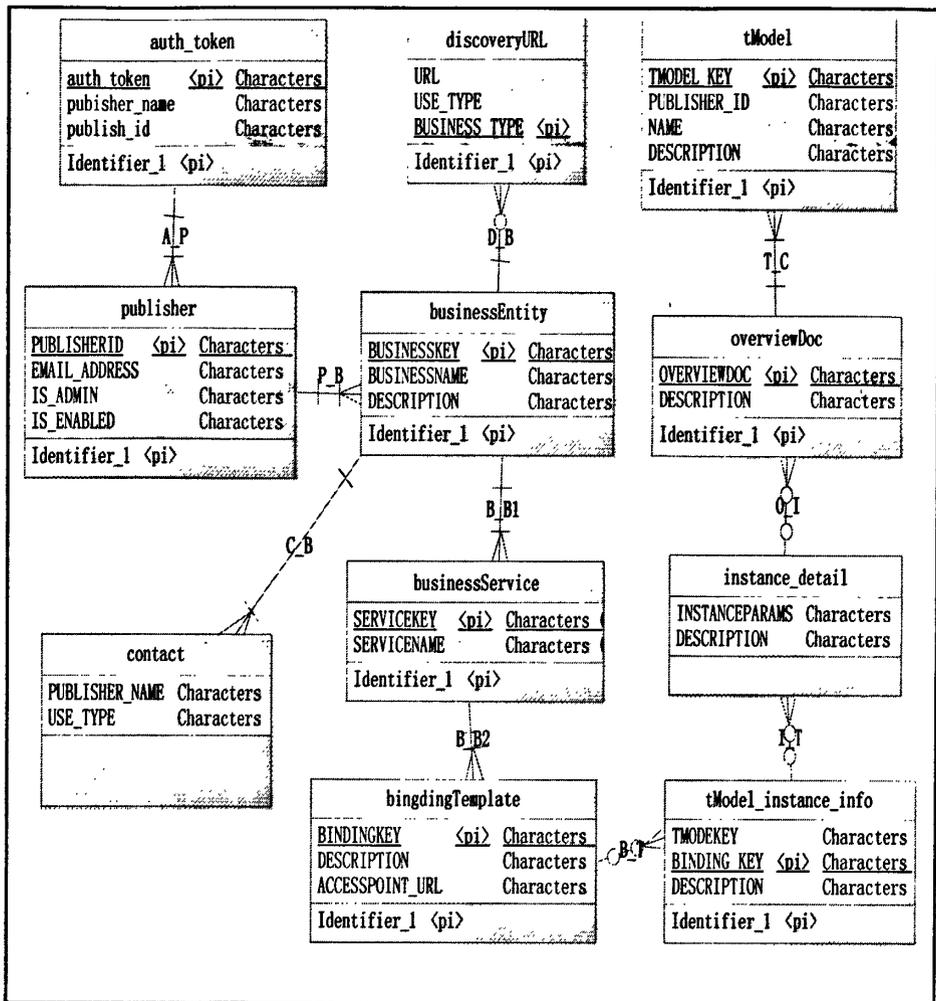


图 5-3 数据库 CDM 部分图

5.5.4 OWL-S解析器的实现

OWL-S 解析器是用于对用户提交的 OWL-S 文件及用户通过用户接口提交服务的各项描述信息进行解析, 抽取出服务的分类信息、DataModel 信息、IO 信息和 textDescription 信息, 并将信息传递给注册/匹配层次进行下一步处理。笔者基于 OWL-S API 实现 OWL-S 解析器。OWL-API 提供了 java 编程接口, 通过其 API 可以方便的读取 OWL-S 文档的服务描述, 而且它的数据结构设计 OWL-S 本体也是一致的。基于 OWL-S 提供的基础接口笔者为 OWL-S 解析器设计了如下方法: getServiceCategory(), getInput(), getOutput(), getDataModel() 和 getTextDescription () 等方法。注册/匹配层将利用这些接口完成可视化 Web 服务的发布与匹配工作, 程序片段如图 5-4。

```

public class OwlEditor{
    private String serviceCategory;
    private OwlModel Input,output;
    private String textDescription;
    //获取 owl-s 中的 serviceCategory 信息
    public String getServiceCategory(){
        serviceCategory=OwlEditorDaoImpl.findServiceCategory();
    }//获取 owl-s 中的 Input 信息
    public OwlModel getInput(){
        Input=OwlEditorDaoImpl.findInput();
    } //获取 owl-s 中的 outPut 信息
    public OwlModel getOutput(){
        outPut=OwlEditorDaoImpl.findOutput();
    }//获取 owl-s 中的 text_description 信息
    public String getTextDescription(){
        textDescription=OwlEditorDaoImpl.findDescription();
    }
}

```

图 5-4 OWL-S 解析器代码片段

5.5.5 本体推理机的实现

本体推理机主要用于服务各本体描述概念之间关系的推理，本文采用 Jena2.6.4 作为原型系统推理机。Jena 描述推理机是基于 Tableau 算法的实现的，可以解析各种描述逻辑本体，支持对 OWL-DL 的解析。而且它是基于 Java 语言实现的，提供很多的基础接口调用。Jena 读取本体和相关操作的程序片段如图 5-5。

```

//创建一个 OntModel 对象
OntModel m=ModelFactory.createOntologyModel(
    OntModelSpec.OWL_MEM,null);
m.getDocumentManager().addAltEntry(
    "http://www.whut.edu.cn/visualization","file:owl/visObject.owl");
//读入本体文件

```

```

m.read("http://www.whut.edu.cn/visualization");//获取概念Data对应的实例
String prefix="http://www.whut.edu.cn/visualization.owl#";
OntClass data = m.getOntClass(prefix + "Data");//得到类名
String name = data.getLocalName();

```

图 5-5 利用 Jena 读取本体和相关操作的程序片段

5.5.6 服务发布的实现

原型系统设计的服务发布流程为：用户通过合法身份登录原型系统后，进入服务发布界面，根据服务发布界面提供的各项服务发布要求，如录入服务对应的相关信息后，原型系统后台将通过信息抽取层抽取用户输入的服务信息，并将信息传递给服务注册模块完成服务的注册，及服务语义描述的保存，服务发布的后台代码片段如图 5-6。

```

//创建一个 UDDIProxy() 代理对象
UDDIProxy proxy = new UDDIProxy();//设置服务发布及发现的有效地址
proxy.setPublishURL("http://localhost:8088/juddi/publish");
proxy.setInquiryURL("http://localhost:8088/juddi/inquiry");
//在注册服务之前,首先获取一个认证
AuthToken token = proxy.get_authToken("userid","password");
Vector entities = new Vector();
//为了发布一个服务,必须要有一个商业实体,因此首先创建一个商业实体
BusinessEntity be = new BusinessEntity("", "businessName");
entities.addElement(be);
// 通过有效的认证信息来保存所创建的商业实体到注册中心
BusinessDetail bd =
    proxy.save_business(token.getAuthInfoString(), entities);
//使用绑定模板信息来创建一个服务
BusinessService businessService = new BusinessService();
businessService.setDefaultNameString(serviceName, null);
//设置用来发布服务的商业实体的见键值
BusinessService.setBusinessKey(businessKey);

```

```

Vector services=new Vector();
Services.addElement(businessService);
//最终通过有效的认证信息来保存所发布的服务到注册中心
ServiceDetail
serviceDetail=proxy.save_service(token.getAuthInfoString(),
                                services);
    
```

图 5-6 服务发布程序片段

原型系统的服务发布接口主要是为用户提供录入服务相关信息的通道，笔者是采用 J2EE 技术实现可视化 Web 服务发布与匹配原型系统，而且系统将采用普通 web 浏览器作为服务发布接口的客户端。用户成功登陆系统后，将获得服务发布的接口，在服务发布接口界面上为用户设计了如下录入项目：服务名称、服务描述、服务类别、商家信息、可处理的数据模型、服务采用的算法、服务 WSDL 地址、输入参数、输出参数和 OWL-S 上传。服务发布接口客户端界面如图 5-7。

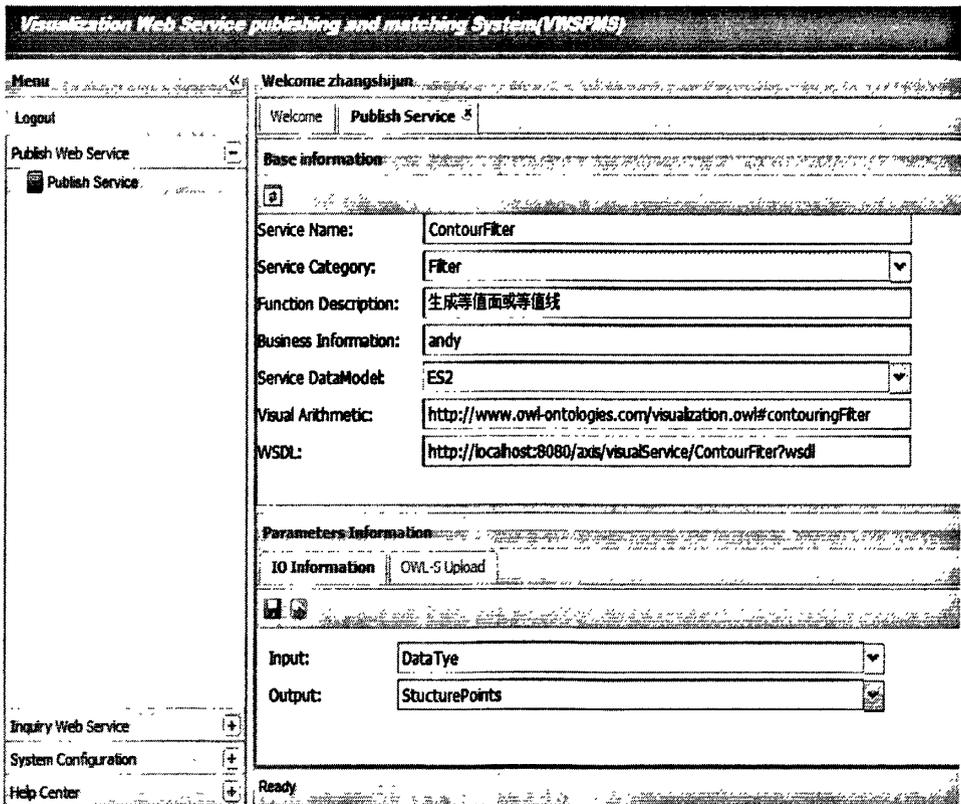


图 5-7 服务发布界面

5.5.7 服务查询的实现

5.5.7.1 服务匹配算法的实现

根据可视化 Web 服务发布与匹配模型的设计, 原型系统的匹配方法实现如下:

(1) 使用 OWL-S 解析器从 req_Profile 中抽取服务的 Service Category、textDescription、IO 和 DataModel 信息。

(2) 通过 Service Category 信息判断服务类别, 假如服务所属类别为 Reader 或 Mapper 服务, 则调至步骤(3), 假如服务所属类别为 Filter、Rendering 或 Integrity, 则调至步骤(4)。

(3) 对于该类型中的每一个服务, 从服务的 OWL-S 文档中抽取服务的 IO 和 textDescription 信息, 调至步骤(6)。

(4) 对于该类型中的每一个服务, 从服务的 OWL-S 文档中抽取服务的 DataModel、IO 和 textDescription 信息, 调至步骤(5)。

(5) 应用 4.6.3 中的 DataModel 匹配算法进行两服务的 DataModel 匹配, 得到 DataModel 匹配相似度 $Sim_{DataModel}$, 并与 DataModel 匹配预设阈值 M_d 比较, 保留 $Sim_{DataModel} > M_d$ 的服务, 作为下一步服务匹配的获选服务集。

(6) 应用 4.6.4 中的 IO 匹配算法进行两服务的 IO 匹配, 得到 IO 匹配相似度 Sim_{IO} , 并与 IO 匹配预设阈值 M_{io} 比较, 保留 $Sim_{IO} > M_{io}$ 的服务, 作为下一步服务匹配的获选服务集。

(7) 对候选服务集合, 应用 4.6.2 的 textDescription 匹配算法进行两服务的 textDescription 匹配, 得到 textDescription 匹配相似度 Sim_{text} , 并与 textDescription 预设阈值 M_t 比较, 保留 $Sim_{text} > M_t$ 的服务, 作为下一步服务匹配的获选服务集。

(8) 对候选服务集合, 应用 4.6.5 的综合匹配算法计算综合匹配相似度 Sim , 并与预设的综合匹配相似度阈值 M_F 比较, 保留 $Sim_F > M_F$ 的服务, 得到服务的输出结果集 ResultSet。

(9) 应用模型设计的排序方法对 ResultSet 服务集排序, 并返回排序后的结果集 ResultSet。

5.5.7.2 服务查询接口的实现

模型的服务查询接口主要是为用户提供录入服务查询相关信息的通道, 及

查看服务查询的匹配结果。原型系统将采用普通 web 浏览器作为服务查询接口的客户端。用户成功登陆系统后，将获得服务查询的接口，在服务查询接口界面上为用户设计了如下录入项目：服务类别、服务描述、可处理的数据模型、输入参数、输出参数和 OWL-S 上传。服务查询接口客户端界面如图 5-8。

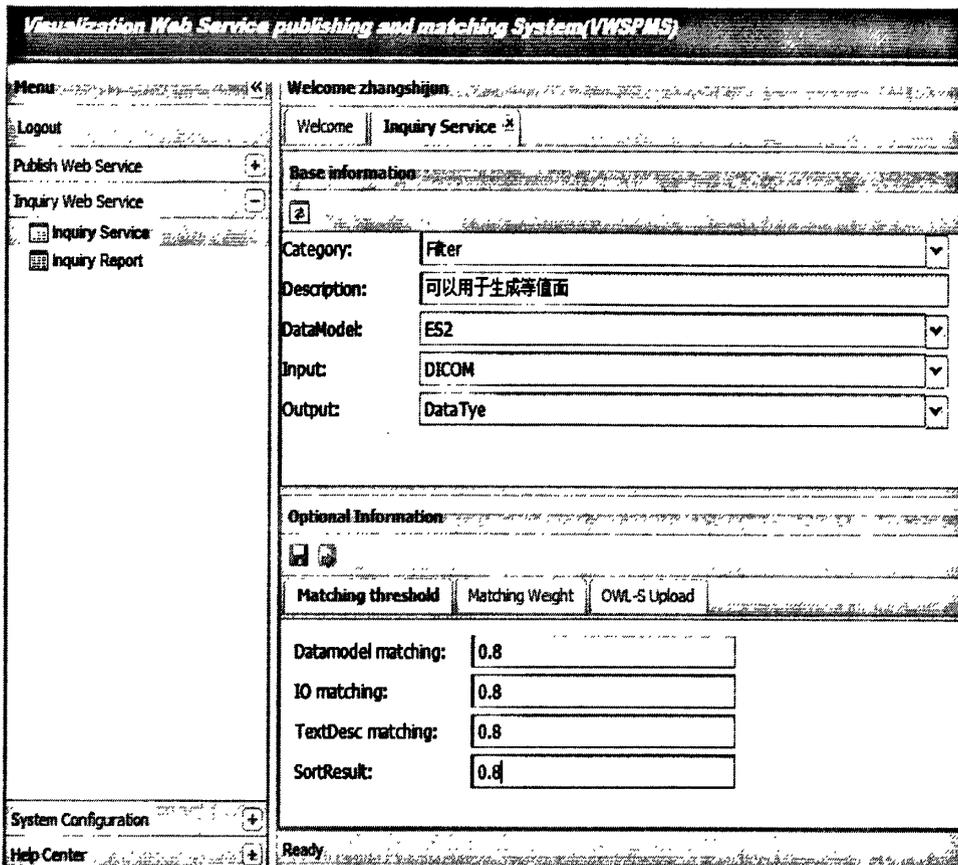


图 5-8 服务查询界面

5.6 本章小结

本章主要介绍可视化 Web 服务发布与匹配模型的具体设计与实现，首先阐述了该模型的主要设计思想，然后从宏观上描述了该模型的层次结构，接着对模型层次结构中的各功能模块，及基于该模型的可视化 Web 服务发布流程和发现流程进行了详细的说明，最后描述了基于 J2EE 实现模型原型系统的详细过程。

第 6 章 实验测试与结果分析

6.1 引言

为了验证可视化 Web 服务发布与匹配原型系统的可行性及有效性。笔者通过构建一些可视化 Web 服务实例，对原型系统进行了服务发布与查询的测试，并采用查全率、查准率和响应时间对测试结果进行了定量的分析。因此本章首先介绍了测试环境的搭建，然后对测试过程进行了详细的介绍，最后采用查准率、查全率及响应时间对系统测试结果进行详细分析。

6.2 测试环境搭建

为了验证可视化 Web 服务发布与匹配原型系统的可行性和有效性，笔者搭建了面向服务的可视化环境，测试环境拓扑结构如图 6-1。

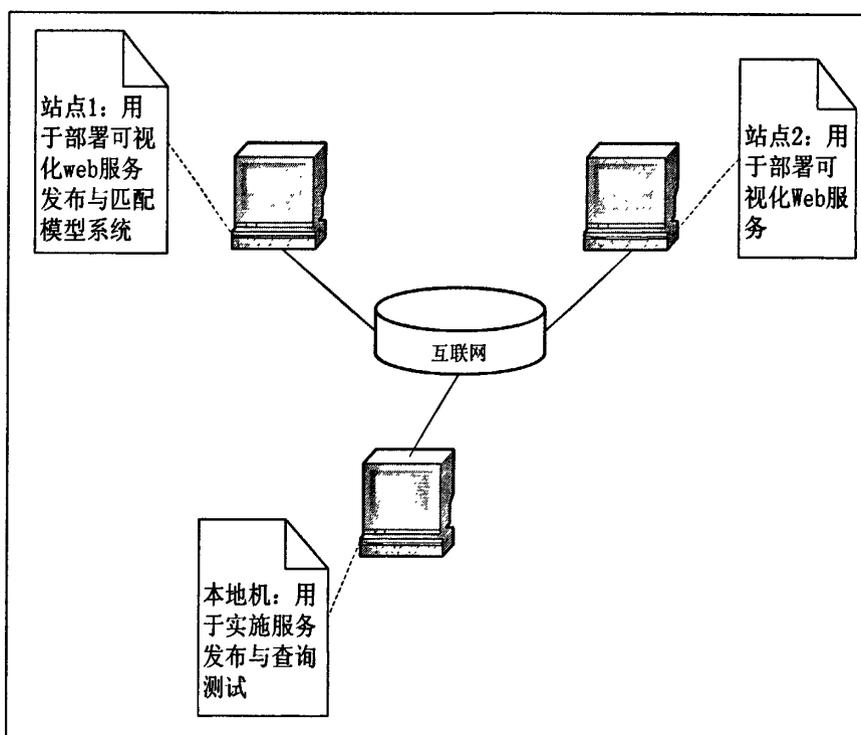


图 6-1 测试环境网络拓扑图

搭建环境所需的软硬件资源如下:

(1) 三台 CPU 为 Intel(R) Core (TM) 2 Duo, 主频为 2.1GHz, 内存为 2G, 采用 Windows 7 操作系统的 PC 机。一台用于部署可视化 Web 服务发布与匹配平台, 一台用于支持服务的远程发布, 一台用作为服务发布与查询的本地机。

(2) 数据库工具: MySQL5.0, 用来实现对 UDDI 服务注册库及缓存库创建和管理。

(3) 本体开发工具: Protege3.4.4, 用来构建可视化本体库。

(4) Web 服务器: Apache Tomcat6.0, 用来部署原型系统的实现平台。

(5) 服务发布工具: AXIS 2.0, 用于部署开发的可视化 Web 服务。

(6) 可视化工具包: VTK5.6, 用于实现可视化 Web 服务实例的开发。

6.3 测试过程

6.3.1 可视化Web服务实例的构建

测试实验中需要开发多组 Reader 服务、Filter 服务、Mapper 服务、Rendering 服务和 Integrity 服务。笔者根据 VTK 类库中提供的算法类, 利用 JAVA 语言将其按功能分别封装成各种类型的服务, 然后利用 AXIS2.0 将它们发布到远程服务部署站点, 并获取服务发布的 WSDL 文档地址。笔者构建可视化 Web 服务实例列表如下:

表 6-1 可视化 Web 服务实例部分表

服务名称	服务类别	描述
DICOMDataReader	Reader 服务	可以读取 DICOM 格式的数据
JPEGDataReader	Reader 服务	可读取 JPEG 格式的数据
ImageDataReader	Reader 服务	可读取图片格式的数据
ContourFilter	Filter 服务	可处理数据模型为 ES2、ES3 的数据
DividingCubes	Filter 服务	可处理数据模型为 EnS3 的数据
LineGraph	Filter 服务	可处理数据模型为 ES1 的数据
PolyDataMapper	Mapper 服务	对 PolyData 类型进行映射
TextMapper	Mapper 服务	对 2D Text 类进行映射
ImageMapper	Mapper 服务	对 Image 类型进行映射
PloyDataSourceWidget	Rendering 服务	用于绘制 PolyDatay 映射数据
GenericRender	Rendering 服务	采用通用方法绘制
OpenGLRenderrer	Rendering 服务	采用 OpenGL 进行绘制

在将所构建的所有可视化 Web 服务部署到远程站点后, 根据所获得的 WSDL 文档地址信息, 分别将它们通过可视化 Web 服务发布与匹配原型系统提供的服务发布接口注册到该可视化 Web 服务发布与匹配原型系统中去, 这样为后面的测试实验做好准备, 同时也是对原型系统服务发布的测试。

6.3.2 测试方案

可视化 Web 服务发布与匹配平台, 为用用提供了服务查询接口, 用户只需要输入 Service Category、DataModel、IO 和 textDescription 就可在平台服务库中查找与用户请求最为匹配的服务或服务列表。笔者设计的测试方案如下:

方案 1: 用户只输入服务的 Service Category、IO、DataModel、textDescription 信息进行服务查询, 设置各项匹配的阈值 M 为 0, 然后对平台输出的服务匹配结果进行查准率、查全率和响应时间的评估。

方案 2: 用户只输入服务的 Service Category、IO、DataModel、textDescription 信息进行服务查询, 设置各项匹配的阈值 M 为 0.6, 然后对平台输出的服务匹配结果进行查准率、查全率和响应时间的评估。

方案 3: 用户只输入服务的 Service Category、IO、DataModel、textDescription 信息进行服务查询, 设置各项匹配的阈值 M 为 0.8, 然后对平台输出的服务匹配结果进行查准率、查全率和响应时间的评估。

6.4 测试结果及分析

6.4.1 测试结果

针对以上四种测试方案, 笔者将采用以下两种用户请求实例对各方案进行测试, 用户请求实例如表 6-2。

表 6-2 用户请求

用户请求	请求分类	输入	输出	DataModel	功能描述
VWS_Req_1	Filter	DICOM	PolyData	ES3	能够对 DICOM 进行预处理
VWS_Req_2	Reader	DICOM	ES3	-	能够读取 DICOM 数据类型
VWS_Req_3	Mapper	PolyData	Cylinder	-	能够对 PolyData 数据进行映射处理

各方案测试结果如表 6-3 所示。

表 6-3 服务匹配结果

查询方法	方案	服务输出列表	相似度值	响应时间
VWS_Req_1	方案 1	ContourFilter	9.3	45s
		DividingCubes	8.2	
		MarchingCube	6.4	
	方案 2	ContourFilter	9.3	36s
		DividingCubes	8.2	
		MarchingCube	6.4	
	方案 3	ContourFilter	9.3	34s
		DividingCubes	8.2	
VWS_Req_2	方案 1	DICOMImageReader	9.5	40s
		ImageDataReader	8.3	
		DataObjectReader	5.8	
	方案 2	DICOMImageReader	9.5	33s
		ImageDataReader	8.3	
	方案 3	DICOMImageReader	9.5	31s
		ImageDataReader	8.3	
VWS_Req_3	方案 1	PolyDataMapper3D	9.3	41s
		PolyDataMapper	8.2	
		GenericMapper	8.0	
		PolyDataMapper2D	7.3	
	方案 2	PolyDataMapper3D	9.3	35s
		PolyDataMapper	8.2	
		GenericMapper	8.0	
		PolyDataMapper2D	7.3	
	方案 3	PolyDataMapper3D	9.3	33s
		PolyDataMapper	8.2	
		GenericMapper	8.0	

通过比较可以看出，三个方案得到的排序基本一致，不同的地方在于。方案 1 由于所有匹配项的过滤阈值 M 都设置为 0，所有在匹配过程中并没根据各阶段匹配情况过滤掉不相关服务，因此其执行时间要比方案 2、3 长，但输出匹配服务的结果要比方案 2、3 多；方案 2 由于其各项匹配的过滤阈值都设置为 0.6，因此在各阶段中，只要服务的相似度值大于等于 0.6，则将被保留作为下一阶段匹配的候选集，所以其两次服务查询结果和方案 1 基本相同，但响应时间小于方案 1 大于方案 3；方案 3 的两次服务查询的结果都要比方案 1、2 少，但是响应时间比方案 1、2 都快，而且方案 3 的查询结果的准确率高于前两个方案。

6.4.2 结果分析

对于普通的语义 Web 服务匹配系统应用通常是从性能和有效性两方面来衡量。性能可以用系统响应时间来度量，有效性可以用服务的查准率及查全率来计算。其中系统响应时间是指从用户提交请求到系统返回匹配服务序列的时间；查准率是指系统查找出匹配结果和领域专家所给结果的重合度；查全率是指系统查找匹配结果与领域专家所给结果交集在系统输出结果的比例。

● 查准率和查全率分析

根据实验结果计算绘制的服务匹配查准率情况如图 6-2，查全率情况如图 6-3。

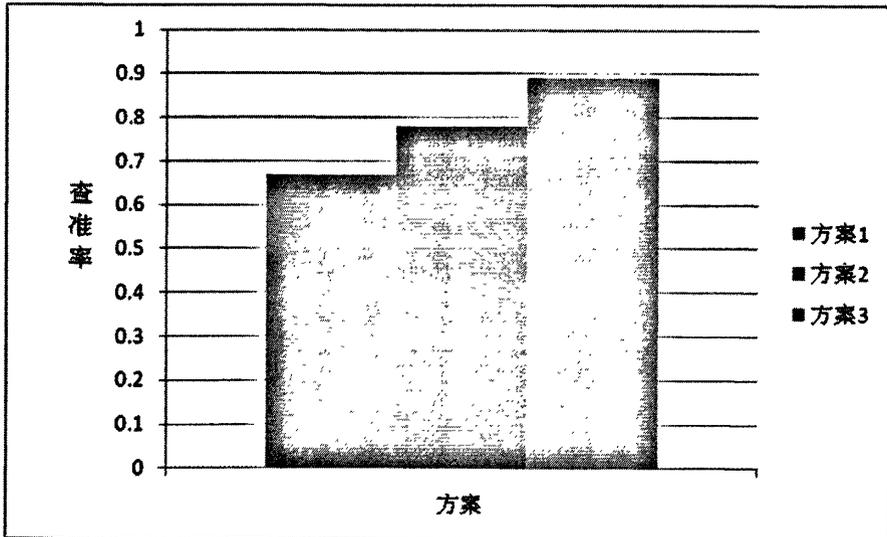


图 6-2 查准率

由图中各方案的查全率情况可以发现，原型系统采用的分类多阶段的匹配

方法是有效的可行的，而且正应为在各匹配阶段引入了过滤阀 M,它能够很好的过滤到大部分不相干的服务，能够很好的提高系统的查全率。而且通过多次试验证明，对于不知道如何设置过滤阈值的用户可以将所有阈值设置为 0.8，这样能保证很高的查准率。

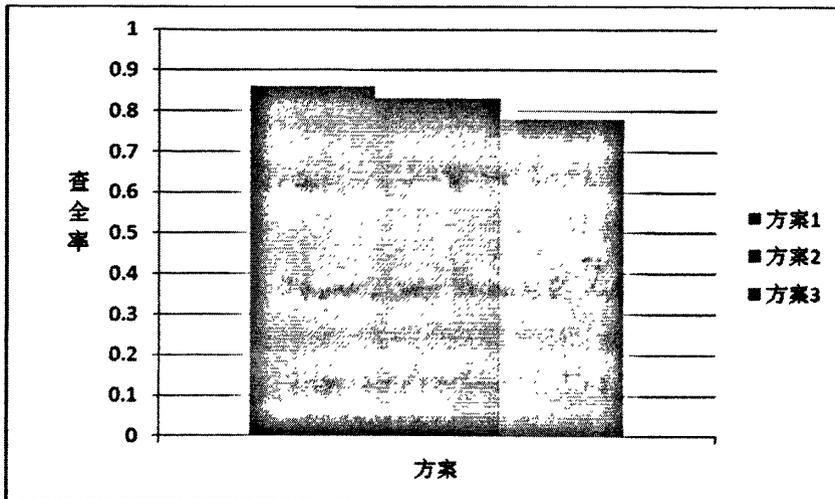


图 6-3 查全率

从图 6-3 中可以发现，方案 1 的查全率是 8.4 大于方案 2 和方案 3，而方案 2 的查全率又大于方案 3。这主要是还因为三个方案各匹配项的过滤阈值设置的情况不同所致。不然发现随着过滤阈值 M 的增大，系统的查全率将越来越小。适当的阈值是保证适当查全率的重要因素。

● 响应时间

三个测试方案对用户请求分别响应时间情况如图 6-4。

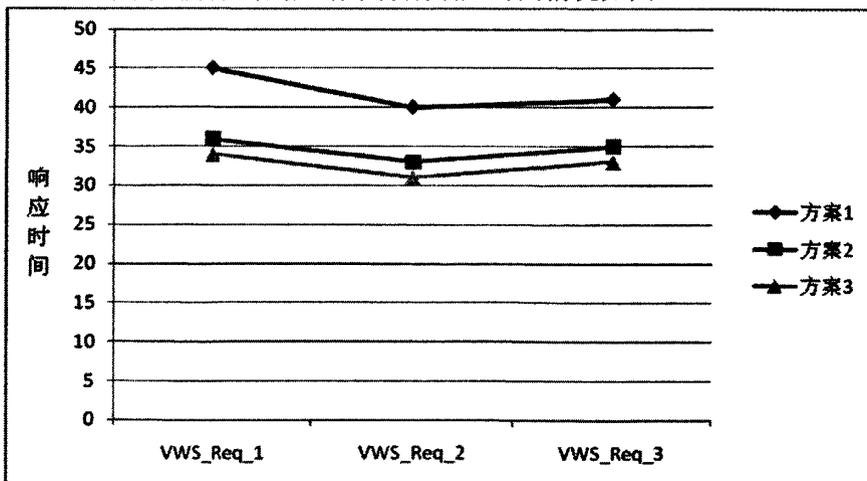


图 6-4 响应时间

响应时间的具体值与原型系统已注册服务的规模有直接的关系,随着服务规模的增大,响应时间将增加。在现有服务规模的情况下,从图 6-4 中可以发现,方案 1 执行 VWS_Req_1、VWS_Req_2 和 VWS_Req_3 的响应时间都要比方案 2、3 长,而方案 2 相应的响应时间又比方案 3 长。这是因为方案 1 的各项匹配阈值等于 0 小于方案 2、3,而方案 2 又小于方案 3,由此可以证明随之原型系统各项匹配阈值的增大,将在相应程度上减少不相干服务的匹配,在一定程度提高了原型系统的匹配效率,缩短相应时间。同时我们也发现对于各方案中 VWS_Req_1 请求相应时间要比 VWS_Req_2 大,这是因为原型系统基于可视化特性,针对不同的服务类型请求提供了两条服务匹配路线,VWS_Req_1 是 Filter 类的服务匹配,需要进行 DataModel 的匹配,而 VWS_Req_2 是 Reader 类服务匹配所以,无需进行 DataModel 的匹配,所以 VWS_Req_1 请求的响应时间要大于 VWS_Req_2 请求。这充分证明了原型系统基于可视化领域特性设计多条匹配路线策略的正确性、有效性。

通过以上对原型系统进行的测试分析,可以看出,笔者设计的可视化 Web 服务发布与匹配原型系统是可行的、高效的,模型基于可视化特性分类别多阶段的匹配策略是有效,而且很大程度的提高了可视化 Web 服务的查全率、查准率和查询效率。它将很好的满足可视化技术领域对于服务发布、管理,及服务共享需求。

6.5 本章小结

本章在搭建好可视化 Web 服务发布与匹配原型系统测试环境后,通过构建一些可视化 Web 服务实例,对原型系统进行了服务发布与查询的测试,并采用查全率、查准率和响应时间对测试结果进行了定量的分析。通过客观的实验结果证明了笔者设计的可视化 Web 服务发布与匹配原型系统的可行性及高效性。

第7章 总结与展望

7.1 工作总结

随着分布式可视化技术的发展，Web 上出现了越来越多的可视化服务资源，面对数量庞大的服务群，如何管理并发现满足需要的可视化 Web 服务是有效地利用各种可视化服务资源的前提与基础，而可视化服务的匹配是解决这个问题的第一步也是最重要的一步。可视化 Web 服务匹配是 web 服务匹配在可视化领域的特定应用，它既具有一般服务匹配的共性，又具有可视化技术本身的鲜明个性。目前，有关可视化服务发布与匹配的研究还鲜见报道。因此研究可视化 Web 服务发布与匹配机制，实现可视化 Web 服务发布与匹配系统将具有很大的必要性，也具有重要的理论意义和现实意义。它将为可视化 Web 服务使用者提供服务发布、查找和管理的平台，实现可视化 Web 服务在更广范围内的共享，同时也帮助可视化 Web 服务用户准确的发现自己需要的服务资源。本文的主要作了以下工作：

- 1、定义了可视化 Web 服务的语义描述模型，并构建了一个可视化服务本体库。论文通过对可视化技术、语义 Web 服务描述语言 OWL-S 和本体描述语言 OWL 进行深入研究，设计出可视化 Web 服务描述模型，并构建了可视化服务本体库。

- 2、基于可视化技术领域自身的特点设计出了可视化 Web 服务发布与匹配方法。论文通过研究可视化技术的特点，考虑到可视化技术具有过程鲜明性特点，设计了按可视化过程分类的可视化 Web 服务发布方法；同时通过对现有语义距离相似度计算方法的深入研究，在综合考虑概念二元关系及概念在概念树中深度对相对计算的影响，提出了一种改进的基于语义距离的相似度算法；在该改进算法的基础上，结合考虑可视化技术的自身特点，设计出了一种基于功能语义可视 Web 服务分类别多阶段的匹配方法并针对各匹配阶段的差异设计了相应的匹配算法。

- 3、设计出了一个可视化 Web 服务发布与匹配模型。为了更好的实现可视化 Web 服务资源的共享和管理，笔者利用课题对可视化 Web 服务发布与匹配机理研究的成果，结合服务资源管理的需要，设计出了可视化 Web 服务发布与匹配

模型。

4、实现了一个可视化 Web 服务发布与匹配的原型系统。基于课题设计的可视化 Web 服务发布与匹配模型，笔者采用 J2EE 技术实现该原型系统，原型系统使用户可以方便的实现可视化 Web 服务资源的发布、查询、管理和共享，同时原型系统还具有有良好的可扩展性，便于原型系统的升级。

5、基于 VTK 可视化类库，开发了一定数量的可视化 Web 服务。

7.2 工作展望

本文针对可视化 Web 服务的发布与匹配作了相关基础性的研究，并实现了可视化 Web 服务发布与匹配的原型系统，但是为了进一步完善该原型系统，提高其服务发布与匹配的效率，查准率和查全率，还需要做如下工作：

(1) 完善可视化服务本体库。构建一个完善的领域本体是一项浩大的工程，它需要整个界的共同努力和认可，需要不断的完善和改进。本课题设计的可视化本体只是为了验证模型的有效性而建立的，其中包括可视化数据、算法及参数等方面的概念还不是很完备，还需要做更多的工作。

(2) 改进和完善可视化 Web 服务的语义描述模型及匹配方法。论文主要是基于服务的功能语义信息研究可视化 Web 服务的发布与匹配，实际上服务的非功能信息，对提高服务的查准率与查全率也有很重要的影响。因此课题下一步工作应该将服务的 QoS 信息引入可视化 Web 服务的描述模型中，并研究怎样实现可视化 Web 服务之间的 QoS 属性匹配。

(3) 完善原型系统的各项功能。由于时间的限制，课题设计的可视化 Web 服务发布与匹配原型系统，主要是基于课题设计的模型，仅仅实现了服务的发布与匹配。但是对于系统一些其他功能模块还需要作一些相关工作，比如可以增加用户信息及服务信息的安全管理模块，同时也可以将服务的远程发布功能集成到本原型系统中来。

致 谢

论文完成之际，感慨万千。它的顺利完成得益于我的导师、同学、朋友和亲人的大力支持，在此完成之际，我想对这些人表达我衷心的感谢！

首先，我要感谢我敬爱的导师高曙教授，高老师在我的论文选题、研究进展、论文修改都给予了细致耐心的指导和帮助。在读研的三年时间里，我从高老师这里学到了很多，尤其是她对科研工作的执著追求以及对学术的严谨精神深深地影响着我，这种对待学术事业的态度将永远激励我不断前进。借此机会，向培育我的恩师高曙教授致以最衷心的感谢和最诚挚的敬意！

我也要感谢武汉理工大学计算机学院的领导和老师们，感谢他们在三年学习期间对我学习和生活的帮助与支持！

我还要感谢在实验室和我一同工作和生活过的兄弟姐妹们，与他们的学习讨论使我受益颇丰，正是他们让我在实验室度过了三年快乐的时光，在此衷心地感谢他们！

另外，对评审论文的各位老师们表示衷心的感谢！他们提出的宝贵建议一定将使我受益匪浅。

同时，我还要感谢我的家人，他们在我研究生整个学业过程中对我永不停止的关心和鼓励是我在学习中不懈努力的源泉！

最后，想对所有关心支持我的家人、老师、同学和朋友们说声：谢谢你们！

参考文献

- [1] 石教英, 蔡文立. 科学计算可视化算法与系统[M]. 北京: 科学出版社, 1999: 2-30.
- [2] Ian H.Witten, Eibe Frank. Data Mining: Practical Machine Learning Tools and Thechniques [M]. 北京: 机械工业出版社, 2005: 100-105.
- [3] 高曙. 网络环境下面向服务的分布式可视化系统研究与实现[D]. 武汉: 武汉理工大学, 2007: 14-16.
- [4] Massimo Paolucci, Takahiro Kawamura, Katia Sycara, et al. Semantic Matching of Web Services Capabilities[C]. In Proceedings of International Semantic Web Conference(ISWC). Palo Alto, California, 2002: 333-345.
- [5] Katia Sycara, Seth Widoff, Matthias Klusch, et al. Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace[J]. Autonomous Agents and Multi-Agent Systems, 2002, 5(2): 173-203.
- [6] Duy-Nagan Le, Van-Quoc Nguyen. Matching WSDL and OWL-S Web services[C]. In IEEE International conference on Semantic Computing. Jan, 2009: 604-620.
- [7] Matthias Klusch, Patrick Kapahnke, Ingo Zininkus. SAWSDL-MX2: A Machine-learning Approach for Integrating Semantic Web Service Matchmaking Variants[C]. In IEEE International Conference on Web Services. Amsterdam, 2006: 320-322.
- [8] Kyriakos Kritikos, Dimitris Plexousakis, Member, IEEE. Mixed-Integer Programming for Qos Based Web Service Matchmaking[C]. IEEE Transactions on services computing. Washington, 2009: 7-8.
- [9] Viorica R. Chifu, Ioan Salomie, Roland Vachter. Matching Semantic Web Services Using Learning Accuracy[C]. In International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. Australia, 2009: 11-12.
- [10] Bin Xu, Tao Li, Zhifeng Gu, Gang Wu. SWSDS: Quick Web Service Discovery and Composition in SEWSIP[C]. In The 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services. USA: SPIE, 2006: 71-71.
- [11] 史忠植, 蒋运承. 基于描述逻辑的主体服务匹配[J]. 计算机学报, 2004, 27(5): 17-21.
- [12] Min Liu, Weiming Shen, Qihao, Junwei Yan, Qi Gao. A Multi-level Matching Framework for Semantic Web Services in Collaborative Design[C]. In IEEE International conference on Semantic Computing. Jan, 2009: 142-145.

- [13] The simple object Access protocol version 1.2 [EB/OL]. [2011-4-25]. <http://www.w3.org/TR/soap/>.
- [14] W3C: OWL Web Services Architecture[EB/OL]. [2011-4-25]. <http://www.w3.org/TR/ws-arch/>.
- [15] SOA: [EB/OL]. [2011-4-25]. http://en.wikipedia.org/wiki/Service-oriented_architecture.
- [16] Michael P. Papazoglou. Web 服务原理和技术[M]. 龚玲, 张云涛, 译. 北京: 机械工业出版社, 2009: 25-33.
- [17] The Universal Description, Discovery and Integration(UDDI) protocol[EB/OL]. [2011-4-25]. <http://www.uddi.org>
- [18] T. Berners-Lee, J Hendler, O Lissala. The Semantic Web[J]. Scientific American, 2001, 284(5): 34-43.
- [19] W3C. Resource description framework[EB/OL]. [2011-4-27]. <http://www.w3.org/RDF/>.
- [20] W3C, OWL-S, Semantic Markup for Web Services[EB/OL]. [2011-4-26]. <http://www.w3c.org/submission/OWL-S>.
- [21] C. Upson, T. Faulhaber, D. Kamins, et al. The Application visualization system: a computational Environment for Scientific Visualization[C]. In IEEE Computer Graphics and Applications. USA, 1989: 30-42
- [22] 李晓梅, 黄朝军, 蔡勋等. 并行与分布式可视化技术及应用[M]. 北京: 国防工业出版社, 2001.
- [23] Studer R, Benjamins R, et al. Knowledge Engineering: Principles and methods[J]. Data and knowledge engineering, 1998, 25(2): 161-197.
- [24] DAML-based Web service ontology[EB/OL]. [2011-4-25]. <http://www.daml.org/services/daml-s/0.9/>.
- [25] Web Ontology Language(OWL)[EB/OL]. [2011-4-26]. <http://www.w3c.org/2009/OWL/>.
- [26] OWL Web Ontology Language Overview[EB/OL]. [2011-4-26]. <http://www.w3c.org/TR/REC-owl-feature-20040210/>.
- [27] OWL Features[EB/OL]. [2011-4-27]. <http://www.w3.org/TR/2004/REC-owl-feature-20040210/>.
- [28] National e-Science Centre, Visualization for e-science[EB/OL]. [2011-4-26]. http://www.nesc.ac.uk/esi/events/130/Workshop_report.pdf.
- [29] National e-Science Centre. Visualization ontologies[EB/OL]. [2011-4-27]. http://www.nesc.ac.uk/vis_ontogy_report.pdf.

- [30] D.J.Duke, K.W.Brodlied and D.A.Duce.Building an Ontology of Visualization[C].In 15th IEEE Visualization .Los Alamitos,Californian, 2004: 24-25.
- [31] K.W.Brodlied,1992.Visualization Techniques,in Scientific Visualization-Techniques and Applications,edited by K.W.Brodlied,L.A.Carpenter, R.A.Earnshaw, J.R.Gallop, R.j.Hubold,A .M.Mumford, C.D.Osland and P.Quarendon,Chapter 3,Spring-Verlag.37-86.
- [32] K.W.Brodlied.A Classification Scheme for Scientific Visualization[C]// R.A.Earnshaw, D.Watson.Animation and Scientific Visualization,Academic Press, 1993: 125-140.
- [33] Selan Rodrigues dos Santos.A Framework for the Visualization of Multidimensional and Multivariate Data[D]. The University of Leeds, 2004.
- [34] T.R.Gruber.Towards Principles for the Design of Ontologies Used for Knowledge Sharing[J]. International Journal of Human Computer Studies, 1995, 43(10): 907-928.
- [35] VTK technology introduction[EB/OL]. [2011-4-30]. <http://www.vtk.org/VTK/help/book.html>.
- [36] Stanford Center for Biomedical Informatics Research. Protégé[EB/OL].[2011-4-28].<http://Protégé.stanford.edu/download/registered.html>.
- [37] Jena-A Semantic Web Framework for java[EB/OL]. [2011-4-30]. <http://jena.sourceforge.net>.
- [38] Graeme H, David S.Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms[C].In Felbaum ,1998: 305-332.
- [39] Roy Rada,Hafedh Mili,Ellen Bicknell, et al.Development and application of a metric on semantic nets[J].IEEE Transaction on Systems,Man and Cybernetics,1989,219(1):17-30.
- [40] Resnik P.Using Information Content to Evaluate Semantic Similarity a Taxonomy[C].In Proceedng of the 14th International Joint Conference on Artificial Intelligence.Montreal,1 995:124-130.
- [41] Tversky A.Features of Similarity[J]. Psychological Review.1977, 84(4): 327-328.
- [42] Salton G,Wong A.On the specification of term value in automatic Indexing[J].Journal of Documentation, 1973, 29(4) :351-372.
- [43] Web Services-Axis[EB/OL]. [2011-4-30]. <http://ws.apache.org/axis/>.

在校攻读硕士学位期间公开发表论文

[1] 张嗜军、高曙. 一种改进的增量式 JVM 垃圾收集算法, 计算工程, 2011 年第 24 期(正刊发表)。

[2] Gao Shu, Shijun Zhang, Wei Xu. Selecting Visualization Web Services Based on Visualization Characteristics. Proceedings of 2011 3rd International Conference on Machine Learning and Computing, 2011,448-452.(To be indexed by EI).