一种基于本体的网格服务匹配方法*)

张 燕[1] 贾 焰[1] 黄晓斌[2] 周 斌[1] 顾 剑[1]
(国防科技大学计算机学院网络所 长沙 410073)[1]
(空军雷达学院信息与指挥自动化系信息处理教研室 武汉 430019)[2]

摘 要 传统的网格服务匹配一般是基于关键字匹配,这种匹配方法缺乏语义信息,灵活性差,查全率低。本文提出了一种新的基于本体的网格服务匹配方法,该方法使用本体语言OWL-S来描述网格服务,充分考虑了服务的语义信息,同时利用OWL推理机对网格服务进行服务分层,以提高服务匹配的效率。实验结果表明,与其他网格服务匹配方法相比,本文所提出的匹配方法具有较高的查全率和查准率及较短的服务匹配时间等优点。

关键词 网格服务匹配,本体,概念分层,服务分层

An Ontology-based Grid Service Matching Method

ZHANG Yan[1] JIA Yan[1] HUANG Xiao-bin[2] ZHOU Bin[1] GU Jian[1]
(School of Computer Science, National University of Defense Technology, Changsha 410073, China)[1]
(Department of Information Engineering, Air Force Radar Academy, Wuhan 430019, China)[2]

Abstract Traditional grid service matching is based on keyword, and has shortages such as inflexibility, low precision and recall. To improve the efficiency of grid service matching, this paper proposes a novel grid service matching method based on ontology, this method uses OWL-S to describe grid service, which takes into account the service semantic information and DL-reasoner to form service hierarchy. The experimental results show that compared with other methods, this method has higher recall and precision and lower service matching time.

Keywords Grid service matching, Ontology, Concept hierarchy, Service hierarchy

1 引言

随着时间的发展,网格已从最初的计算网格,通过与Web Service技术的融合而发展成为服务网格,其体系结构也从最初的五层沙漏结构发展为开放网格服务结构OGSA(Open Grid Service Architecture)[1]。在OGSA体系结构下,网格环境中的资源以服务的形式呈现给用户,资源的发现过程体现为服务发现的过程。因此,如何从大量的网格服务中定位用户所需要的服务就变得尤其重要。

早期的服务匹配大多是基于关键字匹配,其代表有Web Service中的UDDI[2]和Globus Toolkit 4[3]中基于服务数据匹配的Index Service。基于关键字匹配的服务匹配方法存在灵活性差、查全率低等缺点,这些缺点都是由于目前网格服务缺少有序的语义描述而引起的。因此,如何描述网格服务的语义信息并从语义上对网格服务进行匹配成为了众多研究者的研究目标。

2001年,Web技术发明人Berners-Lee等人提出了语义网概念[4],其理论基础之一是本体论[5]。本体是共享概念模型的明确形式化规范说明[6],它能够以一种明确的、形式化的方式来表示领域知识,同时提供共享的、精确定义的术语源,并具有良好的概念层次结构和对逻辑推理的支持。网格和语义网技术的融合使得深度共享日益增长的网格信息资源成为可能。

近几年,有许多研究者提出了基于语义的服务匹配方法。Song Zilin等人提出了一种网格环境下基于图论的服务搜索策略[7],该方法利用了函数参数的语义相似性来构造网格服务匹配图,从而将用户的请求与网格服务进行匹配。文献[8]提出了一种语义网格节点框架来描述网格服务,同时还提出了一种基于语义的搜索和推理引擎来支持网格服务注册及发现。文献[9]也提出了一种基于语义的网格服务发现框架,其目的是为了更加灵活地找出用户所需要的服务。上述这些基于语义的网格服务发现方法都具有灵活性好、查全率高等优点,但是也存在一定的不足,即服务匹配耗时较高,不能很好地满足用户对服务请求的实时性要求。因此,如何在保持语义服务匹配原有优点的基础上减少服务匹配所花费的时间是一个具有研究价值的问题。

现有的基于本体的网格服务发现的时间耗费主要由两部分组成,即服务发布所花费的时间和服务匹配所花费的时间。一般情况下,由于服务发布阶段耗时较少,而即时的服务请求对实时性要求很高,因此减少服务请求的响应时间是提高网格服务发现效率的关键。基于这个目标,本文提出了一种新的基于本体的网格服务匹配方法,该方法使用本体语言OWL-S来描述网格服务,在网格服务提供者注册网格服务后,利用描述逻辑推理机自动对网格服务本体中的概念进行概念分层;对已经发布的网格服务也进行分层,并采用有向无环图来表示服务间的这种层次关系,并且这些工作在服务匹配之前完成,以减少查询阶段所需的时间。实验结果表明,本文所提出的网格服务匹配方法在保持灵活性强和查全率高等

*)国家973项目(2005CB231804)。张 燕 博士生,研究方向为网格技术和资源管理;贾 焰 教授,博士生导师,研究方向为数据库技术与分布式计算;黄晓斌 博士,讲师,研究方向为模式识别。

优点的同时，减少了匹配所花费的时间，从而提高了网格服务发现的性能。

2 基于本体的语义网格服务匹配

为了能有效匹配网格服务，本文提出了一种新的基于本体的语义网格服务匹配方法，该方法利用 OWL-S 语言来描述网格服务。OWL 沿袭 RDF 的基本事实陈述方式以及 RDFS 的类和属性分层结构，并以描述逻辑为基础而构建语言基础，而描述逻辑具有有效的推理功能，特别是对概念包含关系的自动推理。本文正是将描述逻辑的这一优点与网格服务推理紧密结合起来，充分利用描述逻辑在概念分层中的优势为服务请求者提供高效的语义服务匹配。另外，描述逻辑具有清晰的模型—理论语义，可以保证服务匹配具有可靠性和有效性。

通常情况下，因为服务发布阶段对实时性要求不高，而即时的服务请求对实时性要求很高，所以可以在服务注册阶段完成概念分层和服务分层等工作，以达到缩短服务请求响应时间的目的。下面详细介绍概念分层和服务分层。

2.1 概念分层

本文在对概念进行分层时，借鉴文献[10]中所提出的分层方法，利用 FaCT ++推理机[11]对本体库中的概念进行自动分层，形成概念分层图。与手工建立概念分层结构相比，自动概念分层能降低工作量，减少由于主观性和不确定性所带来的影响。

假设本体库中所有概念组成的集合为 $C=\{C_1, C_2, \cdots, C_n\}$，同时假设该集合中不包含相同(或等价)的概念。概念分层算法详细描述如下：

概念分层算法：ConceptHierarchy(C)

输入：概念集合 C
输出：一个概念分层图 G
1) 初始化 G=null;
2) 如果 C 为空，算法结束，返回 G;
3) 从 C 中取 C_i，$1 \leqslant i \leqslant n$，修改 C←C−{$C_i$};
4) 若 G 为空，生成根节点 root=new node("root")，生成节点 node(C_i)，并令 parent(node(C_i))=root;// 函数 parent(node(C_i))表示求出 node(C_i)的直接父节点
5) 调用 insert(root, C_i)函数;
6) Goto 2)

函数 insert(root,C_i)的描述如下所示：
1) 令 Children=children(root);// 函数 children(node(N))表示求出节点 N 的所有直接子节点的集合
2) 若 Children 为空，生成新节点 node(C_i)，令 parent(node(C_i))=root，函数返回;
3) 从 Children 中取 Child，令 Children←Children-{Child};
4) 根据 FaCT ++推理机，如果 Child. Concept 与 C_i 无关，则 Goto 2);// 两个概念 C_i 和 C_j 不相关是指 Subsumption(C_i,C_j)的值为 0
5) 根据 FaCT++推理机，如果 C_i 包含 Child. Concept，则生成新节点 node(C_i)，令 Parent(node(C_i))= root，并令 Parent(Child)= node(C_i)，同时删除关系 Parent(Child)=root，函数返回;
6) 递归调用 insert(Child,C_i)

由于上文假设所给定的概念集合 C 中不包含相同(或等价)的概念，因此，由概念分层算法所得到的概念分层图是一个有向无环图。同时，根据算法第 4)步和函数 insert(root, C_i)可知返回结果仅有一个有向无环图，并且，该返回结果是以 root 节点为最上层节点的一个有向无环图。

如果概念集合 C 中包含相同(或等价)的概念，则可以在分层过程中采用一些控制策略，例如，一旦发现环的出现，就说明有等价概念，这时可以把所有等价概念看作是一个统一的等价概念集，并将该等价概念集用一个新的概念来替代，通过采取该措施，最终也只能得到有向无环图。

概念分层是一个动态的过程，当本体库中增加了新的概念时，针对该概念，可以调用概念分层算法，将新概念插入到合适的位置。当要注销某概念时，应该从概念分层中删除该概念。另外，概念分层应该在服务匹配以前完成，因为概念分层是一个比较费时的过程，我们采用动态的方法对概念进行分层，得到概念之间的分层有向图，在有向图上进行服务匹配肯定比直接利用逻辑推理机进行服务匹配快得多。

2.2 网格服务分层

为了能缩短服务请求阶段所需的时间，本文对网格服务进行分层，并使用有向无环图(Directed Acyclic Graph)来表示这种分层结构。我们使用谓词 *Match* 来表示服务能力之间的匹配关系。对 *Match* 定义如下：

定义 1 假设用户所请求的服务用 *Req* 表示，*Req* 具有给定的输入集 in_{Req} 和输出集 out_{Req}。同样，任意一个已经发布的网格服务可以用 *Adv* 表示，*Adv* 具有一个输入集 in_{Adv} 和一个输出集 out_{Adv}。对于 *Match* 有如下定义：

$Match(Adv, Req) = (\forall in \in in_{Adv}, \exists in' \in in_{Req}: Subsumption(in', in)>0)$ and $(\forall out' \in out_{Req}, \exists out \in out_{Adv}: Subsumption(out, out')>0)$

在服务本体中，服务的输入和输出是通过概念来描述的，因此，将表示服务输入和输出的概念用 2.1 节中的分层方法来分层，可以达到对服务进行分层的效果。

对服务进行分层所带来的主要好处是减少服务发现阶段所处理的匹配的次数。在服务匹配算法中，两个输入或者输出之间的匹配程度依赖于描述该输入或输出的概念间的匹配程度。网格服务分层具有下面两个性质：

① 如果处于分层图中的顶点所表示的服务与用户所请求的服务不匹配，即如果没有前驱节点的节点匹配失败的话，那么位于该节点的子层的所有服务都不能与用户请求相匹配。

② 如果位于分层图最底层的服务能与用户请求匹配，则该节点的所有前驱节点所表示的服务都能与用户请求匹配。

以上两个性质可以表示如下：

性质 1 $\neg Match(C, Req) \Rightarrow \forall C'$ 使得 $Match(C, C')$：$\neg Match(C', Req)$

性质 2 $Match(C, Req) \Rightarrow \forall C'$ 使得 $Match(C', C)$：$Match(C', Req)$

由于文章篇幅的限制，本文省略了性质 1 和性质 2 的证明。使用性质 1 和性质 2 可以减少网格服务匹配的时间。

2.3 网格服务匹配过程描述

本文所提出的服务匹配主要是指服务能力的匹配。在对服务能力进行匹配时，主要综合考虑服务输出信息匹配、输入信息匹配和服务数据信息匹配。其中输出信息是否匹配，关系到服务能否被接受，而输入信息匹配程度关系到服务能否正常执行。

对输出信息，将用户所请求的服务的输出描述逐项取出，在服务发布方的输出概念分层图中查找，并判断其所属概念是否在服务发布方的输出描述中有对应的匹配项。若所请求的服务的输出信息的每一项均能找到匹配项，则认为输出信息的整体匹配是成功的。在输出信息整体匹配成功的情况下，再来计算整个输出匹配项的匹配程度。用 $W(c)$ 表示概念 c 在整个输出描述项的权值，假定服务的输出项描述集合 O 中拥有概念数为 N，则 $\sum_{i=1}^{N} W(c_i)=1$。整个输出匹配项的匹配度 $MO=\sum_{i=1}^{N} W(c_i) \times \underset{1 \leqslant j \leqslant M}{\mathrm{MAX}}(Subsumption(c_i, c_j))$，其中 M 为需求方输出信息描述集合拥有的概念数；$Subsumption(C_i, C_j)$

的定义见参考文献[12]，它在这里表示发布方输出描述中第 i 个概念和需求方输出描述中第 j 个概念的基本概念匹配程度。

输入信息匹配的基本过程与输出信息匹配大致相同，只是其判断方式为将发布方的输入描述逐项取出在需求方的输入描述中寻找匹配；服务数据可看作一种特殊的输出信息。类似地，我们可以定义 MI 和 MD 分别表示输入匹配项的匹配度和服务数据的匹配度。

网格服务能力的整体匹配度可以通过公式 $S_M(Req, Adv)=W_O\times MO+W_I\times MI+W_D\times MD$ 来获得。其中，W_O，W_I 和 W_D 分别表示输出匹配度的权值、输入匹配度的权值和服务数据匹配度的权值，且 $W_O+W_I+W_D=1$。在本文中，这些权值由人为给出，但是必须遵循一个基本原则，就是 $W_O>W_I>W_D$，这是因为输出信息匹配、输入信息匹配和服务数据匹配在整个服务匹配中的重要性是递减的。

3 实验及实验结果分析

本文设计了三组实验，实验一用于分析网格服务匹配时间开销构成；实验二将本文所提出的服务匹配方法服务同 Globus Toolkit 4 中使用的基于关键字匹配的方法以及文献[13]中所提出的语义网格服务匹配方法在性能上进行了比较；实验三将本文所提出的基于服务分层的服务匹配方法同直接基于 OWL 推理机的匹配方法就时间开销进行了比较。下面将详细介绍这三组实验，并分别对实验结果进行了分析。

实验一：网格服务匹配时间开销构成分析

在本文中，网格服务匹配过程可以分为网格服务发布阶段和服务请求阶段，其中，发布阶段所耗费的时间主要包括本体加载时间、概念分层时间、服务能力分层时间；服务请求阶段所耗费的时间主要指在构造好的服务能力分层图中搜索并返回符合用户需求的服务所花费的时间。为了分析网格服务匹配过程中的时间开销构成，本文设计了如下实验：实验环境为 Sony V505 笔记本，1.5GHz Intel Pentium 处理器，内存为 512M，采用本体 Economy[14]，它包含 293 个概念(类)，38 个抽象角色(对象属性)，8 个具体角色(数据类型属性)，基于 Economy 概念，有 206 个服务样本，11 个服务请求。因为概念分层需要借助 OWL 推理机判定概念的包含关系以形成概念分层图，所以，本文在对概念进行分层时使用的推理机为 Fact ++。实验结果如图 1 所示。

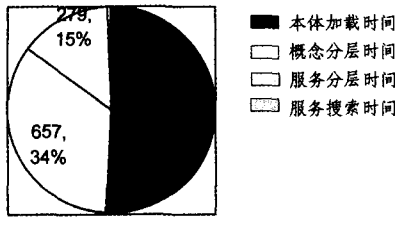

图 1 网格服务匹配时间开销构成

从图 1 中可以看出，服务匹配的大部分时间都花在本体加载以及概念分层上：本体加载耗时最多，占用了总时间开销的 51%；其次是概念分层，占用了总时间开销的 34%，因此发布阶段是耗时较多的阶段。服务分层时间占总时间开销的 15%；在服务分层上搜索服务平均所花时间约占总时间开销的 0%左右。

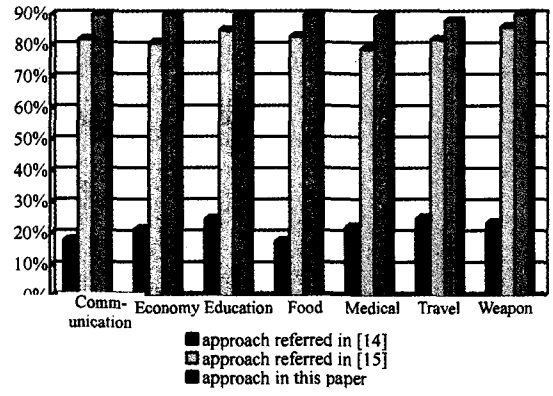

图 2 三种服务匹配方法的查全率

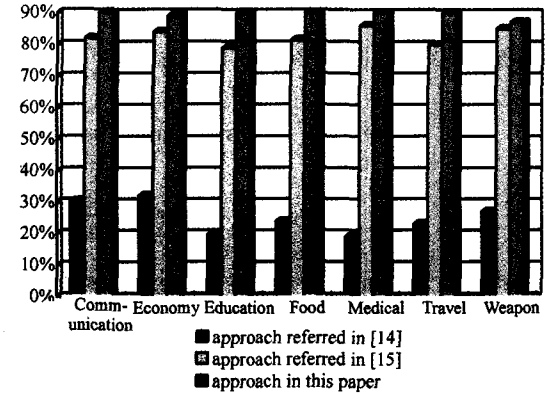

图 3 三种服务匹配方法的查准率

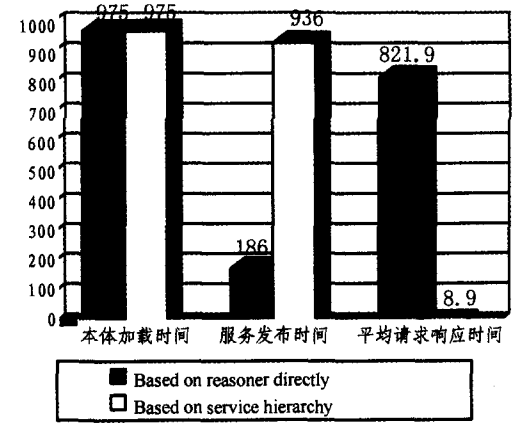

图 4 基于服务分层的匹配方法和直接基于 OWL 推理机方法的时间

实验二：服务匹配性能比较

本文采用查全率和查准率作为衡量网格服务匹配方法性能的指标。查全率衡量匹配的完备程度，查准率衡量匹配的准确程度，二者定义如下：

查全率＝匹配正确的服务数/测试集合中符合要求的服务数

查准率＝匹配正确的服务数/匹配返回的服务数

实验二选择了两种最具代表性的已有的网格服务匹配方法同本文所提出的匹配方法进行比较，这两中方法分别为 Globus Toolkit 4 中所使用的基于关键字的匹配方法，另一种是文献[14]中所提出的语义网格服务匹配方法。

本实验所使用的服务匹配样本集为 OWL-S Service Re-

(下转第 151 页)

我们的基于语义的算法在性能上是差不多的，这表明我们的算法没有领域相关性，所以比 Kea ++算法有着更好的应用前景。

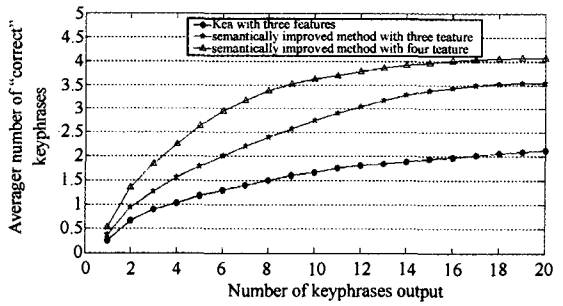

图 1　基于语义的关键词提取算法和 Kea 算法的比较

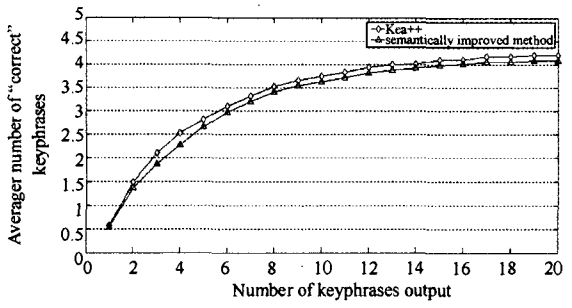

图 2　农业领域中基于语义的提取算法和 Kea ++算法的比较

结束语　在这篇文章中，我们提出了一种考虑词语语义信息的关键词提取算法。该算法首先使用消歧算法得到候选关键词的词义，然后在后面的步骤中使用这些词义的语义相关度信息。实验表明通过考虑语义的信息，关键词算法的性能能得到很大的提高。同时，相对于 Kea ++算法，我们的算法没有领域的限制性。在未来的工作中，我们将会采用更多的数据来对基于语义的算法进行测试。另外，现有的消歧算法的精度不是特别高[6]，因此我们计划设计一种更加有效的消歧算法来提高基于语义的关键词提取算法的性能。

表 2　基于语义的关键词提取算法和 Kea ++算法性能的比较

	Precision	Recall	F-measure
Kea ++算法	0.575	0.569	0.572
基于语义的算法	0.550	0.566	0.558

参 考 文 献

[1] Witten I H, Paynter G W, Frank E, et al. KEA: Practical automatic keyphrase extraction// Proc. DL '99. 1999: 254-256

[2] Turney P D. Mining the Web for Lexical Knowledge to Improve Keyphrase Extraction: Learning from Labeled and Unlabeled Data. Technical Report ERB-1096. National Research Council Canada, 2002

[3] Fellbaum C. Wordnet: An Electronic Lexical Database. Cambridge: MIT Press, 1998

[4] Medelyan O, Witten I H. Thesaurus Based Automatic Keyphrase Indexing// Proc. of the Joint Conference on Digital Libraries 2006. Chapel Hill, NC, USA, 2006: 296-297

[5] Banerjee S, Pedersen T. Extended gloss overlaps as a measure of semantic relatedness // Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence. Acapulco, 2003: 805-810

[6] Pedersen T, Banerjee S, Patwardhan S. Maximizing Semantic Relatedness to Perform Word Sense Disambiguation. Supercomputing institute research report umsi 2005/25, University of Minnesota, 2005

[7] Lovins J B. Development of a stemming algorithm. Mechanical Translation and Computational Linguistics, 1968, 11: 22-31

[8] Dougherty J, Kohavi R, Sahami M. Supervised and unsupervised discretization of continuous features // Proceeding of ICML-95, 12th International Conference on Machine Learning. Lake Tahoe, US, 1995: 194-202

（上接第 147 页）

trieval Test Collection(Version 2.1)[15]，以 communication, economy, education, food, medical, travel, weapon 这 7 个域本体所包含的概念作为参数来源，服务样本集数量分别为 135, 52, 25, 106, 29, 206, 25，在这 5 个服务样本集上所进行的服务请求数目分别为 6, 1, 1, 2, 6, 11, 1。实验结果如图 2 和图 3 所示。

从图 2 和图 3 可以看出，无论是查准率还是查全率，基于语义的方法都远高于基于关键字的匹配方法，三种方法的平均查全率分别为 27.3%, 75.8%, 93%；三种方法的平均查准率分别为 19%, 68.7%, 89.3%。显然，本文所提出的基于服务分层的匹配方法在查全率和查准率上都高于传统的基于语义的网格服务匹配方法。

实验三：基于服务分层的匹配方法和直接基于 OWL 推理机方法的时间开销比较

本实验比较基于服务分层的方法和直接基于 OWL 推理机方法的时间开销，仍以文献[14]中的方法为代表，实验设置与实验一相同。实验结果如图 4 所示。

从图 4 可看出：基于服务分层的匹配方法和直接基于 OWL 推理机的服务匹配方法相比，虽然在服务发布阶段因需要构造概念分层和服务分层而增加了一定的时间开销（约为 6 倍），但用户请求响应时间却大大缩短了，平均响应时间为 8.9ms。因为服务发布阶段对实时性要求不高，而即时的服务请求对实时性要求很高，所以基于服务分层的服务匹配方法更能满足实时服务匹配的需求。

结束语　针对传统的基于关键字的网格服务匹配方法所存在的灵活性差、查全率和查准率低等不足，本文提出了一种新的基于本体的网格服务匹配方法，该方法利用本体来描述网格服务的语义信息，同时，利用 OWL 推理机对网格服务进行服务分层，以提高服务匹配的效率。实验结果表明，本文所提出的网格服务匹配方法与传统的基于关键字匹配的服务匹配方法相比，具有较高的查全率和查准率，同时，与直接基于 OWL 推理机的语义网格服务匹配方法相比，更能满足实时服务匹配的要求。

参 考 文 献

[1] Foster I, Kesselman C, Nick J, et al. The physiology of the grid: An open grid services architecture for distributed systems integration. http: // www. globus. org/research/papers/ogsa. pdf, 2002

[2] UDDI: The UDDI Technical White Paper. http: // www. uddi. org, 2000

[3] Globus project. http: // www. globus. org

[4] Lee T B, Hendler J, Lassila1 O. The Semantic Webl New York: Scientific American , 2001

[5] 李善平，尹奇，胡玉杰，等. 本体论研究综述. 计算机研究与发展，2004, 41 (7): 1041-1052

[6] Studer R, Benjamins V R, Fensel D. Knowledge Engineering, principles and methods. Data and Knowledge Engineering, 1998, 25 (12): 161-197

[7] Song Zilin, Ai Weihua, Wang Yi, et al. Service Search Strategy Based on Graph in Grid Environment // Proceedings of the Second International Conference on Semantics, Knowledge, and Grid (SKG'06)

[8] Zhang Y, Song W. Semantic Description and Matching of Grid Services Capabilities

[9] Ludwig S A, Reyhani S M S. Semantic Approach to Service Discovery in a Grid Environment. Journal of Web Semantics, 2006

[10] 史忠植，蒋运承，等. 基于描述逻辑的主体服务匹配. 计算机学报，2004, 5(17)

[11] Fact++. http: // owl. man. ac. uk/factplusplus/

[12] Paolucci M, Kawamura T, Payne T R, et al. Semantic Matching of Web Service Capabilities. Lecture Notes in Computer Science, 2002, 2342: 333-347

[13] Ludwig S A, Reyhani S M S. Introduction of semantic matchmaking to Grid computing. Journal of Parallel and Distributed Computing, 2005, 65: 1533-1541

[14] http: // www. dfki. de/scallops