文章编号:1001 - 9081(2005)09 - 1970 - 04

基于概念格的 Web 服务管理技术

周傲英[1],彭敦陆[1,2],王晓玲[1]
(1. 复旦大学 计算机科学与工程系,上海 200433; 2. 上海理工大学 计算机工程学院,上海 200093)
(ayzhou@fudan.edu.cn)

摘 要:Web 服务的有效管理是实现服务发现和服务组合的关键。文中定义了 Web 服务及服务间的关系等基本概念,在 Web 服务管理中引入了形式概念分析(FCA)的方法,建立了描述服务间相互关联的概念格,分析了如何通过概念格对 Web 服务进行有效地管理,并实现了概念格的增量维护。对模拟数据和真实数据的相关实验表明,文中提出的基于概念格的方法能有效地实现 Web 服务管理,提供了一种规范的对服务功能的分类管理策略,对服务发现也有较好的检索效果。

关键词:Web 服务管理;形式概念分析;概念格;Web 服务发现
中图分类号: TP311 **文献标识码**:A

Web services management based on concept lattice

ZHOU Ao-ying[1], PENG Dun-lu[1,2], WANG Xiao-ling[1]
(1. *Department of Computer Science and Engineering, Fudan University, Shanghai* 200433, *China*;
2. *College of Computer Engineering, University of Shanghai for Science and Technology, Shanghai* 200093, *China*)

Abstract: An efficient management of Web Services is the key to service discovery and service composition. Some basic concepts about Web services are introduced, the relationships between Web Services are defined. A concept lattice corresponding to the properties of Web services is generated to describe the relationships among different services. The management for Web services with concept lattice is discussed. An algorithm is developed to maintain concept lattice incrementally and to retrieve Web services successfully in the concept lattice. Extensive experiments show that Web services could be managed effectively and retrieved with high efficiency and accuracy when using the proposed approach.

Key words: Web service management; formal concept analysis; concept lattice; Web service retrieval

0 引言

Web Services 作为一个新的应用集成策略,已经受到越来越多的重视。高效地管理这些 Web 服务,以便快速而准确地找到满足用户需要的服务,为服务的发现和服务的组合服务,已成为 Web 服务应用过程中亟待解决的问题。通过深入分析服务组合技术,我们从逻辑上对组合 Web 服务的定义和执行归纳在 3 个层面(图 1)上:业务流层、数据流层和基本服务管理层。本文重点研究基本服务的有效管理技术。

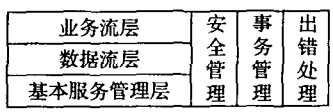


图 1 Web 服务组合中的三层模型

目前,对 Web 服务管理还是以手工方式进行的,这种基于手工方式的 Web 服务分类管理及查找方式存在很大的局限性。如服务请求者对注册中心的服务分类规则不是很了解,就难以检索到满意的服务,不能满足商业应用过程中对 Web 服务发现的高效性和准确性要求[4]。主要原因是 UDDI 的分类体系是固定的、无法全面反映系统所提供的服务功能,因而就不能反映各个 Web 服务(甚至是同类 Web 服务)之间在功能上的相互关系。

另一方面,对 Web 服务的检索基本上还仅限于利用关键字查找匹配,而没有利用 Web 服务描述中的潜在语义信息[2,3]。本文通过对 Web 服务及服务间的关系等基本概念以及 Web 服务发现进行形式化的定义,利用形式概念分析方法及概念格对 Web 服务进行管理,并通过实验证明了所提方法的有效性。

1 基本概念

WSDL(Web Services Description Language)是 W3C 定义的 Web 服务的描述语言[5],它从四个不同粒度对 Web 服务进行了描述:数据类型(Data type)、消息(Message)、方法(Operation)和 Web 服务(PortType)。我们对 Web 服务做如下定义:

定义 1(Web 服务) 一个 Web 服务是一个三元组:$\{T, M, O\}$,其中 $T = \{ptype_1, ptype_2, \cdots, ctype_n\}$ 是数据类型集合,由基本数据类型 *ptype* 和复合数据类型 *ctype* 组成;$M = \{msg_1, msg_2, \cdots, msg_n\}$ 是与 Web 服务进行交互的消息集合,每个消息 $msg \in M$ 都是由一个或多个 $< part, type >$ 对组成。其中,*type* 是数据类型,满足 $type \in T$。$O = \{op_1, op_2, \cdots, op_n\}$ 是该 Web 服务中的所有方法集合,一个方法由输入/输出消息对 $< imsg, omsg >$ 描述,并且满足 $imsg、omsg \in M$。

由定义 1,可以将 Web 服务简单地看作是一个方法集,即 $WS = \{op_1, op_2, \cdots, op_n\}$。每个方法有输入消息(input message)、执行任务及输出消息。表 1 给出了本文所用的 Web 服务实例。

收稿日期:2005 - 07 - 03;修订日期:2005 - 07 - 27

作者简介:周傲英(1965 -),男,安徽郎溪人,教授,博士生导师,主要研究方向:Web 数据管理、数据挖掘和数据流分析、对等计算; 彭敦陆(1974 -),男,安徽广德人,讲师,博士研究生,主要研究方向:数据库、XML、Web 服务; 王晓玲(1975 -),女,山东烟台人,讲师,博士,主要研究方向:Web 数据管理、XML、Web 服务、数据库物理设计.

表 1　Web 服务及其方法

服务名	服务缩写	方法
TravelingService	TS	BTT, HR, ST
BookTicketService	BTS	BTT, BAT, BBT, BGT
PublicService	PS	BPT, BFT, AF
TrainService	TnS	BTT, ST, SD
SportService	SS	BGT, BFT, AF

定义 2(Web 服务之间的关系)　给定两个 Web 服务 WS_1、WS_2 及其方法集 $Opset_1$ 和 $Opset_2$,根据 WS_1 和 WS_2 之间方法的匹配关系,两个 Web 服务之间的关系定义为下面四种:

(1) 如 $Opset_1 = Opset_2$,则 WS_1 和 WS_2 为等价关系;

(2) 如 $Opset_1 \subset Opset_2$ 或 $Opset_2 \subset Opset_1$,称前者为 WS_1 包含于 WS_2,后者为 WS_2 包含于 WS_1;

(3) 如 $comOp = Opset_1 \cap Opset_2 \neq \varnothing$ 且 $Opset_1 \not\subset Opset_2$ 及 $Opset_2 \not\subset Opset_1$,则称 WS_1 和 WS_2 是相交关系,$comOp$ 称为 WS_1 和 WS_2 的共有方法;

(4) 如 $Opset_1 \cap Opset_2 = \varnothing$,则称 WS_1 和 WS_2 是无关服务。

对于两个具有包含关系的 Web 服务 WS_1 和 WS_2,如 WS_1 包含于 WS_2 中,则称 WS_2 是相对 WS_1 更普遍的服务(general service),WS_1 是具有 WS_2 的部分功能(part-of)的服务。

定义 3(Web 服务发现)　设 Web 服务资源库为 $S_{Rep} = \{WS_1, WS_2, \cdots, WS_n\}$,其中 WS_i 为单一的 Web 服务。一个发现请求可描述为 $Q = \{Op_1, Op_2, \cdots, Op_m\}$,这里 $Op_j \in Q$ 表示一个具有指定功能的方法。Web 服务发现的目的是从 S_{Rep} 中发现这样的 Web 服务集 $Q_{RS} = \{WS_k, WS_{k+1}, \cdots, WS_i\}$,其中任意一个 Web 服务 $WS_i(WSM_i \in Q_{RS})$ 的方法集与 Q 的交不为空。

2　概念格

现在给出概念格有关概念及定理[7-9]。形式概念分析(Formal Concept Analysis,FCA)是产生概念格的有效方法,通过概念格可以方便地表达和处理对象和属性间的二元关系。

定义 4(上下文)　一个上下文是一个三元组 $K(W,O,R)$,其中 W 和 O 为两个有限集,W 表示对象的集合,O 表示对象具有的属性集合,$R \subseteq W \times O$ 表示对象 W 和属性 O 之间的二元关系。一个上下文可以通过上下文表进行表示。

定义 5(上下文表)　一个上下文表 T 是一个用来描述二元关系 R 的二维矩阵:

$$r_{ij} = \begin{cases} 1, & \text{if } w_i R o_j \text{ where } w_i \in W \text{ and } o_j \in O \\ 0, & otherwise \end{cases} \quad (1)$$

定义 6(共有属性和共有对象)　设 $W' \subseteq W$ 及 $O' \subseteq O$ 满足上下文 (W,O,R)。W' 中所有对象的共有属性可表示为:

$$O(W') = \{o \in O \mid \forall w \in W' \Rightarrow wRo\}$$

满足 O' 的共有对象可表示为:

$$W(O') = \{w \in W \mid \forall o \in O' \Rightarrow wRo\}$$

定义 7(概念和完全概念格)　$c = (W', O')$ 是上下文 (W,O,R) 中的一个概念,满足条件 $W' \subseteq W$ 及 $O' \subseteq O$,并且 $W(O') = W'$ 及 $O(W') = O'$。概念 c 的外延(extend)表示为 $\pi_o(c) = O'$,内涵(intent)表示为 $I(c) = W'$。(W,O,R) 中的所有概念 $C(W,O,R)$ 表示,它构成了 (W,O,R) 的完全概念格,简称概念格。

概念具有偏序关系:概念的外延包含其子概念(sub-concept)的外延,内涵包含其超概念(super-concept)的内涵。

定义 8(偏序关系)　设 $c_1 = (W_1, O_1)$ 和 $c_2 = (W_2, O_2)$ 为上下文 (W,O,R) 中的两个概念,c_1 和 c_2 的偏序关系可表示为:

$$c_1 \leqslant c_2 \Leftrightarrow O_1 \subseteq O_2 \Leftrightarrow W_2 \subseteq W_1$$

称 c_1 为 c_2 的下确界,c_2 为 c_1 的上确界。如 $c_1 \leqslant c_2$,且在上下文 (W,O,R) 中不存在概念 c_3,满足 $c_1 \leqslant c_3 \leqslant c_2$,则称 c_2 是 c_1 的最小上确界;反之,如 $c_1 \geqslant c_2$,且在上下文 (W,O,R) 中不存在概念 c_3,满足 $c_1 \geqslant c_3 \geqslant c_2$,则称 c_2 是 c_1 的最大下确界。

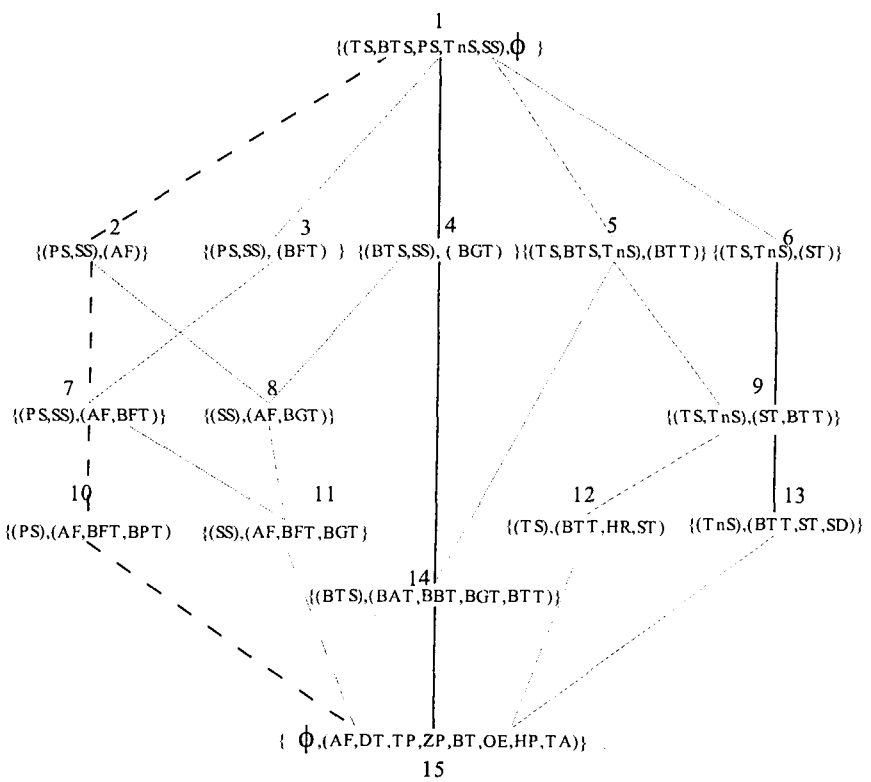


图 2　表 1 中服务的概念格表示

定理 1:设 (W,O,R) 是一个上下文,$B(W,O,R)$ 是由 (W,O,R) 中概念形成的完全格,那么格的最小上界和最大下确界分别为:

$$\bigwedge_{i=1}(W_i, O_i) = \left(\bigcap_{i=1} W_i, o\left(w\left(\bigcup_{i=1} O_i\right)\right)\right)$$

$$\bigvee_{i=1}(W_i, O_i) = \left(w\left(o\left(\bigcup_{i=1} W_i\right)\right), \bigcap_{i=1} O_i\right)$$

证明略。

图 2 是表 1 中服务对应的概念格表示。可以很方便地表达 Web 服务之间的相互关系,也可以方便地发现到包含有指定方法的所有服务。基于上述概念和定义,我们给出 Web 服务的形式概念分析上下文的定义。

定义 9(Web 服务上下文和 Web 服务概念格)　将 Web 服务资源库中所有 Web 服务组成集合 W 看作是上下文的对象集,所有方法组成的集合 O 看作属性集,二元关系 R 表示一个方法是否在某个 Web 服务中出现,于是:

$$W = \{WS_1, WS_2, \cdots, WS_n\}$$

$$O = \{Op_1, Op_2, \cdots, Op_n\}$$

通过(W,O,R)很容易就可以得到与形式概念分析相对应的概念格(图2)。

在利用概念格对 Web 服务进行管理时,要完成下列过程:

(1) 注册 Web 服务。

(2) 抽取 Web 服务中的方法。

(3) 计算所抽取的 Web 服务方法与概念格中现有方法的相似度。如与格中某一方法的相似度大于某一给定阈值,不用更新概念格;如概念格中不存在某个已有方法与该方法的相似度大于某一给定阈值,则将该方法加入概念格。

(4) 建立或更新概念格。

过程(4)是产生概念集和建立概念格,可以利用很多已有算法进行建立。本文不再详述,读者可参看文献[8,10]。

3 概念格的维护

3.1 约简概念格

为了方便处理,我们将所有复杂类型的消息都用组成它们的基本数据类型替代。这样,可以这样来定义 Web 服务方法:

定义 10(Web 服务方法):一个 Web 服务方法 Op 是一个三元组(D,I,O)。这里,D 为方法的描述信息(包括方法名),$I=\{datatype_1,datatype_2,\cdots,datatype_n\}$ 为输入消息类型列表,$O=(datatype_1,datatype_2,\cdots,datatype_m)$ 为输出消息类型列表,I 和 O 中的每个 $datatype$ 为一个基本数据类型。

为避免概念格中出现大量的结点,我们提出了约简概念格。通过对相关的方法进行约简,来减少概念格的规模。

将两个方法进行约简/合并前,需要计算两个方法的相似度,只有当相似度大于某个给定阈值 s 时,才能约简。由定义 10 可知,两个方法 Op_1、Op_2 的相似度包括 3 个分量:

1) 输入类型的相似度(S_1);

2) 输出类型的相似度(S_2);

3) 描述信息的相似度(S_3)。

对于输入/输出消息的相似度(S_1/S_2):计算两个方法的输入/输出消息的相似度,归结为计算两个由基本数据类型组成的序列的相似度。对于描述信息的相似度(S_3),方法的描述信息相似性本质上是文本相似性。采用了文本检索中经典方法——TF/IDF[11],即词频和逆文档频率的方法来计算方法描述间的相似性。在计算前,进行了一些必要的预处理,如去除停用词、合并数字及抽取词干等。其计算公式如下:

$$sim(D_1,D_2)=\frac{IDF(D_1\cap D_2)}{IDF((D_1\cup D_2)/(D_1\cap D_2))}$$

为了在总相似度中体现控制部分,利用下列公式来计算总相似度:

$$sim(Op_1,Op_2)=\frac{1}{3}\sum_{i=1}^{3}\beta_i\prod_{j=1}^{i}S_j$$

其中β_i为权重,$\beta_1+\beta_2+\beta_3=1$。总相似度及各分量相似度取值范围为[0,1]。

当 Web 服务资源库中的方法有较大重复时,概念格经过约简后,规模会减小很多,从而减少了检索时的搜索空间,提高了检索效率,同时也可以保证检索具有完整性。

3.2 概念格的增量维护

由于 Web 服务的动态性和自治性,服务的增长是随时都可能发生的,研究了增量的概念格的维护算法。具体方法是:对于新的服务所对应的每个功能,首先在原有的概念格上进行搜索,查找该功能所对应的概念结点是否已经存在于概念格中。如已经存在,就只需要把对应的结点中,增加内涵 $I(c)$ 中的元素;如不存在对应的服务功能,就作为新的概念节点加入概念格。具体的算法描述为:对于一个新服务 w,用 $w(o)$ 表示该服务包含的所有操作 o。用 K^* 表示加入 w 到原来格 K 后,产生的新的概念格。增量维护的算法如下:

1) 对于满足 $o\subseteq O$ 的 K 中的概念节点(W,O),修改 W 为 $W'=\{w\cup W\}$;

2) 对于 o 中任一操作 $op_i\subset o$,对于 O 中任一操作 $OP_i\subset O$,如 $sim(op_i,OP_j)\geqslant s$(其中 s 为建格时预定义的阈值),则修改概念节点(W,O)为(W',O'),其中 $W'=\{w\cup W\}$,$O'=\{O\}$;同时,在对应的 hash 表中增加 op_i;

3) 对于没有在原有概念格中找到的服务中包含的操作 op,在概念格中增加新的节点(W',O'),其中 $W'=\{w\}$,$O'=\{op_i\}$。

4 实验分析

在 PIII 1.4G、256M 运行 Windows XP 的 PC 机上进行了实验。所有程序都是采用 Java(JDK1.4)编写。测试数据集有两种:一种是真实数据集,它是从几个网站(IBM、Microsoft、Sun、Woogle 及 Google)上下载的 Web 服务数据集;另一种是模拟数据,它是利用我们自己开发的 WSDL 生成器创建的 Web 服务数据集。自己开发的 WSDL 生成器创建的模拟 Web 服务资料库。表 2 给出个两种数据集的具体特征。

表 2 实验数据集

服务集	WSDL 文件数	服务数	方法总数
真实服务库	2000	2000	4776
模拟服务库	2000	2000	4013

为验证概念格规模和检索性能的影响,做了两个实验:(a) 通过定义不同的相似度阈值,观察概念格的大小。在真实数据集上的结果如图 3 所示。(b) 测试了两数据集上,约简概念格和非约简的概念格对于同一检索的响应时间。结果如图 4 所示。

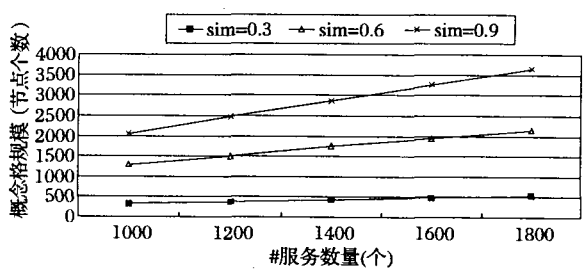
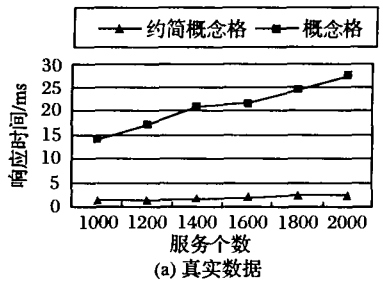


图 3 概念格的规模

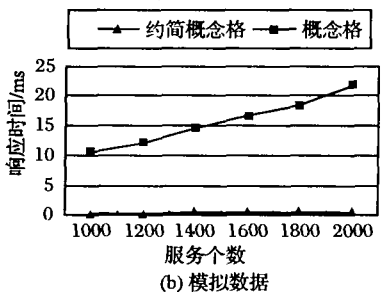
图 3 表明,对于不同的相似度阈值,对格的规模的影响比较大。对于操作相似度定义为 0.5 所生成的概念格 $N1$ 的规模要远远小于相似度定义为 0.9 所生成的概念格 $N2$,当服务个数大约为 1800 时,格的规模 $N2$ 是 $N1$ 的 6 倍多。因此,选择合适的相似度阈值是建格的一个重要参数。

图 4 表明了无论是真实数据,还是模拟数据,约简概念格的响应时间远远小于约简前的检索响应时间. 这是因为当两个方法的相似度大于某个给定阈值 s(实验中我们取 0.6)时,组成概念格的方法数量会大大减少,概念格的大小也会减小,从而缩短了检索响应时间。

第三组实验测试了格的增量维护方法的性能。通过对真



(a) 真实数据



(b) 模拟数据

图 4　约简概念格对检索效率的影响

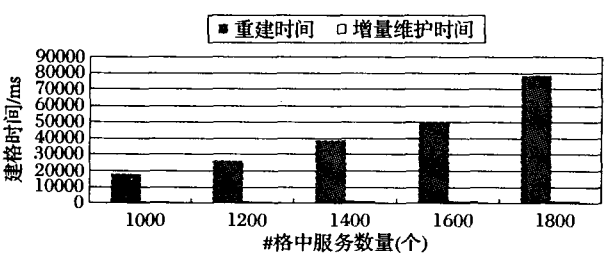


图 5　概念格的增量维护时间

实数据集的测试,分别选择不同的初始服务个数对应的概念格作为基础,然后模拟现实情况,随机选择 10 个服务一次加入概念格,图 5 表明了重新建格的时间与增量维护(在原有格的基础上,加入新的服务和操作)的时间对比。实验结果表明,本文提出的增量维护方法提供了一种有效的途径进行概念格的维护,减少了重新建格的代价。

5　相关工作

目前主要的研究成果包括 Active XML、SELF-SERV、Woogle 系统。我们分别从主要目标、关键技术等角度对上述三个典型系统进行了分析,表 3 是对相关系统的简要总结对比。本文所提方法是与现有方法是不同的:利用形式概念分析的方法,在研究了服务层功能相似的基础上,通过概念格来表示的 Web 间潜在的语义关系,从而有效地管理 Web 服务,在这样的管理技术基础上,服务发现可以通过搜索概念格来检索 Web 服务。

6　结语

本文利用形式概念分析(FCA)的方法,将 Web 服务管理成概念格的形式,并且为在概念格的建立、维护、约简等提出了一套有效的方法,在这样的管理体系中,服务发现也可以有效地实现。通过计算服务方法的相似度对服务方法进行约简,减少概念格,同时也减小了搜索空间,提高了检索效率。理论分析和实验表明,所提方法具有较高的检索效率及较好的检索效果。

表 3　各系统的对比分析

	Active XML	SELF-SERV	Woogle	本文
系统目标	数据集成	服务组合	服务发现	有效的服务管理
主要特点	XML 与 Web 服务结合	基于状态图服务组合及控制	挖掘 WSDL 中隐含的概念	基于概念格的服务管理技术
关键技术	嵌入 XQuery 的 XML	多个基本服务的组合技术	基于关联规则概念发现技术	概念格的建立、维护、查找技术
主要领域	XML、XQuery	工作流、事务管理	服务发现	服务管理
服务发现手段	依赖 UDDI 等	依赖 UDDI 等	基于服务描述的服务发现	基于操作类型和服务描述的服务发现
对服务的分类	无	根据 UDDI 系统	提供分类算法	提供分类算法

参考文献:

[1] ALONSO G, CASATI F, KUNO H, *et al.* Web Services Concepts, Architectures and Applications[M]. Springer Verlag, 2004.

[2] Web Service and UDDI[EB/OL]. http://www.ibm.com/service/uddi, 2001.

[3] Microsoft Advanced UDDI Search[EB/OL]. http//uddi.microsoft.com/search.aspx, 2001.

[4] ZHANG LJ, *et al.* 2002. XML-based Advanced UDDI Search Mechanism for B2B Integration[A]. Proceedings of the 4th IEEE International Workshop on Advanced Issues of E-Commerce and Web-based Information Systems (WECWIS 2002)[C], 2002.

[5] CHRISTENSEN E, CURBERA F, MEREDITH G, *et al.* Web service Description Language (WSDL)[EB/OL]. http://www.w3.org/TR/wsdl.

[6] GAO X, YANG J, PAPAZOGLOU MP. The Capability Matching of Web Services[A]. Proceedings of the 4th International Symposium on Multimedia Software Engineering(MSE'02)[C]. Newport Beach, California, USA, 2002.

[7] GANTER B, WILLE R. Formal Concept Analysis, Mathematical Foundation[M]. Spinger Verlag, 1999.

[8] KOVACS I. Efficiency Analysis of Building Concept Lattice[A]. Proceedings of 2nd ISHR on Compu-Intelligence[C]. Budapest, 2001.

[9] WILLE R. Concept lattices and conceptual knowledge systems[A]. Computers and Mathematis Applications[C], 1992.493 - 515.

[10] GANTER B, WILLE R. Formal Concept Analysis, Mathematical Foundation[M]. Spinger Verlag, 1999.

[11] SALTON G, WONG A, YANG CS. A Vector-space Model for Information Retrieval[J]. Journal of the American Society for Information Science, 1975,(18):13 - 620.

[12] DONG X, HALEVY A, MADHAVAN J, *et al.* Similarity Search for Web service[A]. Proceedings of the 30th VLDB conference[C]. Toronto, Canada, August 2004.

[13] WANG Y, STROULIA E. Flexible Interface Matching for Web-Service Discovery[A]. Proceedings of the Fourth International Conference on Web Information Systems Engineering(WISE 2003)[C]. Alta, Canada, December 2003.

[14] ZHUGE H, LIU J. Flexible retrieval of Web service[J]. The journal of Systems and Software, 2004,(70):106 - 117.

[15] PAOLUCCI M, KAWAMURA T, PAYNE TR, *et al.* Semantic Matching of Web Services Capabilities[A]. Proceedings of the 1st International Semantic Web Conference[C]. Sardinia, Italy, June 2002.318 - 332.

[16] LI L, HORROCKS I. A Software Framework for Matchmaking Based on Semantic Web Technology[A]. Proceedings of the WWW 2003 Conference[C]. Budapest, 2003.331 - 339.

[17] MEDJAHED B, BOUGUETTAYA A, ELMAGARMID AK. Composing Web services on the Semantic Web[J]. The VLDB Journal, 2003,12(4):333 - 351.