

## 基于扩展概念格的 Web 关系挖掘<sup>\*</sup>

姜峰<sup>+</sup>, 范玉顺

(清华大学 自动化系 计算机集成制造国家工程中心, 北京 100084)

### Web Relationship Mining Based on Extended Concept Lattice

JIANG Feng<sup>+</sup>, FAN Yu-Shun

(National Research Center for CIMS, Department of Automation, Tsinghua University, Beijing 100084, China)

+ Corresponding author: E-mail: jiangfeng00@gmail.com

Jiang F, Fan YS. Web relationship mining based on extended concept lattice. *Journal of Software*, 2010,21(10):2432-2444. <http://www.jos.org.cn/1000-9825/3662.htm>

**Abstract:** The lack of an efficient Web organization and management mechanism has become the bottleneck of Web applications. To address the problem, this paper proposes an extended concept lattice model based on Web input and output messages for service relationship mining. Concept coverage functionality is created and introduced into normal concept lattice. Off-Line algorithms for Web relations including equivalence, replacement, and flow relations mining, as well as their on-line update algorithms, are given. Experiment on real Web registry center shows that the extended concept lattice model is a highly efficient organization mechanism for Web relationships. Real Web registry center automatically mines, is expected to provide intelligent support for service choices, optimizations and composition.

**Key words:** concept lattice; formal concept analysis; service management; relationship mining

**摘要:** 针对 Web 服务因缺少有效的组织和管理机制而产生的应用瓶颈问题,引入基于概念覆盖度函数的扩展概念格,通过构建基于输入和输出参数的 Web 服务集的扩展概念格模型,给出了 Web 服务间等价、替代和流关系的离线挖掘算法以及增量和减量的在线更新算法。在真实 Web 服务集上的测试结果表明,扩展概念格模型是 Web 服务集的一种有效的组织形式,可用于 Web 服务关系的自动挖掘和维护,从而为 Web 服务的选择、优化和组合提供智能支持。

**关键词:** 概念格;形式概念分析;服务管理;关系挖掘

中图法分类号: TP391 文献标识码: A

网格技术与 Web Service 技术的不断融合使 Web 服务成为面向服务架构(service oriented architecture,简称 SOA)下实施资源共享和业务集成的主要形式。各种计算资源和制造资源不仅被封装为 Web 服务,企业的业务也以业务组件的形式被封装为 Web 服务,使企业成为以提供和消费服务为主要运行模式的面向服务的企业<sup>[1]</sup>。在这种背景下,网络中 Web 服务的种类和数量都急剧增加,对 Web 服务的组织和管理提出了更高的要求。

\* Supported by the National Natural Science Foundation of China under Grant No.60674080 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2007AA04Z150 (国家高技术研究发展计划(863))

Received 2008-10-28; Revised 2009-03-05; Accepted 2009-06-01

统一描述、发现和集成(universal description, discovery and integration,简称 UDDI)协议是 Web Service 技术架构的基础协议之一,它提供了基于功能描述和关键字匹配的 Web 服务注册、查找和分类管理的基本机制.但这种分类管理的方式存在如下不足:一是 Web 服务提供者对分类规则等的理解不一致或对关键字的定义不准确会产生错误的注册和分类信息;二是同一个服务可能属于多个分类类别,从而导致重复注册或不充分注册,前者会导致信息冗余,后者则导致服务不能被发现;三是用户检索服务时,由于不能完全了解注册中心的分类规则或准确描述功能需求,则会导致检索结果不准确;四是缺乏对 Web 服务间关系的描述.这些不足不仅引起注册中心的数据混乱,使 Web 服务的维护变得困难,还会阻碍服务消费者的使用.例如在某服务注册网站上检索“天气预报”服务,返回结果多达 200 多条,其中有提供天气预报服务的,还有提供气象统计的,用户很难区分这些服务的真正用途.有调研表明,网络中的 Web 服务只有 34%是可用的,大多数链接是无效的<sup>[2]</sup>.此外,由于缺少对服务间关系的描述,这种 Web 服务管理平台很难为用户的使用,例如基于服务组合的方式构建业务流程,提供智能帮助.随着大量服务的产生,平台的使用难度越来越大,最终不得不关闭.可见,简单的服务注册和分类管理已不能满足开放网络环境下 Web 服务有效应用的需求.

基于此,我们在之前的工作中提出了以服务管理为中心的 SOA 体系架构<sup>[3]</sup>,致力于 Web 服务的业务能力描述及服务间关系的挖掘和维护,以实现 Web 服务注册数据的有效存储、索引和组织,进行 Web 服务的有序化和深度聚合.Web 服务间的关系是该架构的核心,包括等价、替代和流关系等.在基于服务组合的方式构建企业业务流程的应用中,等价和替代关系是进行 Web 服务选择和组合方案优化的基础,流关系则是服务得以组合为业务流程从而实现业务价值的基础,流关系的自动挖掘可以为服务组合提供智能支持.

本文针对 Web 服务关系的挖掘,提出一种基于概念格<sup>[4]</sup>的自动挖掘方法.概念格作为一种完备的格结构,可以描述对象和属性间的二元关系以及基于二元关系产生的概念间的泛化和例化关系,被广泛用于机器学习、软件工程和信息获取等领域<sup>[5-7]</sup>.本文基于 Web 服务的输入和输出参数,构建 Web 服务集的形式背景和概念格模型,并引入概念覆盖度函数对该概念格进行扩展,在此基础上给出基于扩展概念格的 Web 服务间等价、替代和流关系的离线挖掘算法以及增量和减量的在线维护算法,可用于 Web 服务关系的自动挖掘.

## 1 Web 服务的形式化描述

形式化是服务关系自动挖掘的基础.按照本文基于概念格的挖掘需求出发,Web 服务从接口描述的角度可用如下二元组来表示:

$$Ws = \langle Input, Output \rangle \quad (1)$$

其中,  $Input = \{In_1, In_2, \dots, In_k\}$ ,  $Output = \{Out_1, Out_2, \dots, Out_l\}$  是 Web 服务完成某项功能的输入/输出参数集,且  $In_i, Out_j = \langle name, type \rangle$ , 其中  $name$  是参数名,  $type$  是参数的数据类型.约定:  $Input \cap Output = \emptyset$ .

任意两个服务  $Ws_1$  和  $Ws_2$ , 依据接口参数的匹配存在如下关系:

- 若  $Input_1 \subseteq Input_2$ , 则称  $Ws_1$  是  $Ws_2$  关于输入的可替服务.特别地,若  $Input_1 = Input_2$ , 则称  $Ws_1$  和  $Ws_2$  输入等价,记为  $Ws_1 \doteq Ws_2$ ;
- 若  $Output_1 \subseteq Output_2$ , 则称  $Ws_2$  是  $Ws_1$  关于输出的可替服务.特别地,若  $Output_1 = Output_2$ , 则称  $Ws_1$  和  $Ws_2$  输出等价,记为  $Ws_1 \doteq Ws_2$ ;
- 若  $Ws_1$  和  $Ws_2$  同时满足输入等价和输出等价,则称  $Ws_1$  和  $Ws_2$  等价,记为  $Ws_1 \equiv Ws_2$ ; 若  $Ws_1$  同时是  $Ws_2$  关于输入和输出的可替服务,则称  $Ws_1$  是  $Ws_2$  的可替服务,记为  $Ws_1 \geq Ws_2$ ;
- 若  $Output_1 \cap Input_2 \neq \emptyset$ , 则称  $Ws_1$  与  $Ws_2$  存在流关系,且  $Ws_1$  为  $Ws_2$  的先序服务,  $Ws_2$  为  $Ws_1$  的后序服务,记为  $Ws_1 \longrightarrow Ws_2$ .特别地,若  $Output_1 \subseteq Input_2$ , 则称  $Ws_1$  是  $Ws_2$  的部分先序服务或  $Ws_2$  是  $Ws_1$  的不完全后序服务,记为  $Ws_1 \xrightarrow{Par} Ws_2$ ; 若  $Output_1 \supseteq Input_2$ , 则称  $Ws_1$  是  $Ws_2$  的充分先序服务或  $Ws_2$  是  $Ws_1$  的完全后序服务,记为  $Ws_1 \xrightarrow{Suf} Ws_2$ ; 若  $Output_1 = Input_2$ , 则称  $Ws_1$  和  $Ws_2$  互补,且  $Ws_1$  为  $Ws_2$  的严格先序服务,  $Ws_2$  为  $Ws_1$  的严格后序服务,记为  $Ws_1 \xrightarrow{Exa} Ws_2$ ;
- 若  $Ws_1$  和  $Ws_2$  既不等价(含输入等价、输出等价),也不存在流关系,则称  $Ws_1$  和  $Ws_2$  无关,记为  $Ws_1 ! Ws_2$ .

## 2 概念格与 Web 服务形式背景

概念格是形式概念分析(formal concept analysis,简称 FCA)方法的核心数据结构.在形式概念分析中,对象、属性及其相互间的二元关系称为概念格的形式背景.形式背景用三元组  $K=(O,A,R)$  表示,其中  $O$  是所有对象的集合, $A$  是所有属性的集合, $R \subseteq O \times A$  表示对象  $O$  和属性  $A$  之间的二元关系.任意  $o \in O, a \in A$ ,若  $(o,a) \in R$ ,表示对象  $o$  拥有属性  $a$ ,记为  $oRa$ .

形式背景可以用一个二维矩阵表示,矩阵的行表示对象,列表示属性,矩阵元素取值规则为

$$r_{ij} = \begin{cases} 1, & \text{若 } o_i R a_j, o_i \in O, a_j \in A \\ 0, & \text{其他} \end{cases} \quad (2)$$

定义形式背景  $K$  上的两个映射:

$$\begin{aligned} f(O_i) &= O'_i = \{a \in A \mid O_i \subseteq O, \forall o \in O_i, (o,a) \in R\}, \\ g(A_j) &= A'_j = \{o \in O \mid A_j \subseteq A, \forall a \in A_j, (o,a) \in R\}, \end{aligned}$$

其中  $O_i, A_j$  分别为  $O, A$  的子集.若  $f(O_i) = A_j$ ,且  $g(A_j) = O_i$ ,则称  $C = (O_i, A_j)$  是形式背景  $K$  上的一个概念, $O_i$  称为概念的外延, $A_j$  为概念的内涵. $K$  上所有概念的集合记为  $L(K)$ .

为了表达方便, $\{o\}$  简记为  $o'$ , $\{a\}$  简记为  $a'$ .有如下引理成立:

引理. 设  $K=(O,A,R)$  为一形式背景, $O_1, O_2 \subseteq O$  为对象子集, $A_1, A_2 \subseteq A$  为属性子集,下列结论成立:  
①  $O_1 \subseteq O_2 \Leftrightarrow O'_2 \subseteq O'_1$ ; ②  $A_1 \subseteq A_2 \Leftrightarrow A'_2 \subseteq A'_1$ ; ③  $O_1 \subseteq O'_2, A_1 \subseteq A'_2$ ; ④  $O'_1 = O''_1, A'_1 = A'''_1$ .

对  $L(K)$  中的两个概念  $C_1 = (O_1, A_1), C_2 = (O_2, A_2)$ ,规定: $C_1$  为  $C_2$  的子概念, $C_2$  为  $C_1$  的父概念,当且仅当:  $C_1 \leq C_2 \Leftrightarrow O_1 \subseteq O_2 \Leftrightarrow A_1 \supseteq A_2$ .父子关系是一种偏序关系, $L(K)$  中所有概念按照这个偏序关系可以生成一个格结构,称为概念格. $C_1$  称为  $C_2$  的下确界或子概念, $C_2$  称为  $C_1$  的上确界或父概念.若不存在概念  $C_3$  使  $C_1 \leq C_3 \leq C_2$  成立,则称  $C_1$  是  $C_2$  的最大下确界或直接子概念, $C_2$  是  $C_1$  的最小上确界或直接父概念.格中的概念个数称为格的大小,记为  $|L(K)|$ .在不引起混淆的情况下,概念格仍用  $L(K)$  表示.可以证明, $L(K)$  是完备格,与形式背景  $K$  一一对应.

Web 服务间按照接口参数的匹配结果形成多种关系,当 Web 服务数量巨大时,直接定义所有服务间的关系将十分艰难.基于概念格可以实现这些关系的自动挖掘,从而为基于 Web 服务组合的应用提供智能支持.

首先定义 Web 服务的形式背景.

定义 1(Web 服务集的形式背景).  $K = (WS, P, R)$  是 Web 服务集的形式背景,当  $WS = \{WS_1, WS_2, \dots, WS_k\}$  是服务集中所有服务组成的对象集合; $P = \{p_1, p_2, \dots, p_{n-1}, p_n\}$  是  $WS$  中所有服务的输入/输出参数组成的属性集( $P$  中允许存在相同元素), $R$  表示一个参数存在于服务的输入或输出接口中.

Table 1 Formal background of Web services

表 1 Web 服务集的形式背景

	$p_1$	$p_2$	$p_3$	$p_3$	...	...	$p_{n-1}$	$p_n$
$WS_1$	0	1	0	1	...	...	0	0
$WS_2$	0	0	1	0	...	...	1	1
$WS_3$	1	1	1	0	...	...	1	0
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
$WS_{k-1}$	0	0	1	0	...	...	0	1
$WS_k$	1	0	0	1	...	...	0	1

按式(2)定义,Web 服务集的形式背景可用表 1 所示的二维矩阵来表示.

采用 Godin 算法<sup>[8]</sup>等方法可以生成该形式背景的概念格,并可用 Hasse 图直观表示.Hasse 图是概念格的可视化表示,可以直观地表达概念间的关系.

## 3 概念格的扩展

形式背景通常是外延一致的,即  $\forall a_1 = a_2 \in A$ ,有  $a'_1 = a'_2$ .因此,即使属性集  $A$  中存在相同元素,也不会引起格中概念的混淆.但 Web 服务集中的服务间存在流关系,即:存在  $p \in Input_i \cap Output_j$ ,称  $p$  为连接参数,如表 1 中的  $p_3$  所示,前者表示输入参数  $p_3$ ,后者表示输出参数  $p_3$ .显然,当分别表示输入和输出参数时, $p_3$  具有不同的外延.因此,定义 1 中 Web 服务集的形式背景是外延不一致的.由此产生的概念格中包含属性  $p_3$  的概念不能区分  $p_3$  的来源,

因而会产生混淆,不能准确描述服务间的关系.因此,需要对 Web 服务的形式背景进行扩展.

另一方面,概念格描述了对对象集  $O$  和属性集  $A$  的幂集间的二元关系,格的大小通常随着对象和属性数量的增加呈指数增长.当  $O$  和  $A$  的基数很大时,格中概念个数将十分庞大.直接在格中挖掘 Web 服务间的关系,时间开销将会很大.因此,需要对 Web 服务集的概念格作适当扩展,以缩小概念搜索范围.

### 3.1 形式背景染色

对于外延不一致的形式背景,通过属性染色可以消除不一致.

**定义 2(染色形式背景).** 三元组  $K=\langle O,A,R \rangle$  是一个外延不一致的形式背景,五元组  $K'=\langle O,A,C,R,F \rangle$  称为  $K$  的染色形式背景,其中  $C$  是颜色集合,  $F \subseteq A \times C$  表示属性  $A$  和颜色  $C$  之间的二元关系:  $A$  中的每一个元素  $a$  被标记为  $C$  中的一个颜色  $c$ , 记为  $a^c$ , 且对于  $A$  中的元素  $a_1, a_2$ , 若  $a_1=a_2$ , 不存在  $c \in C$ , 使  $a_1^c, a_2^c$  同时成立.

定义 2 中外延不一致的属性被标记为不同颜色,因此可以消除不一致问题.同理可以定义 Web 服务集的染色形式背景  $K'=\langle WS,P,M,R,F \rangle$ , 其中颜色集  $M=\{Input, Output\}$ , 参数集  $P$  染色后被分为两组:  $P^1=\bigcup_{i=1}^k Input_i=\{p_1, p_3, \dots, p_m\}$  为输入参数集,  $P^2=\bigcup_{i=1}^k Output_i=\{p_2, p_3, \dots, p_n\}$  为输出参数集.

颜色集  $M$  构成了输入/输出参数集  $P$  的一个划分,通过染色严格区分了输入和输出参数集中相同的参数属性,从而构成外延一致的形式背景.据此产生的概念格可以清晰地表达概念与属性间的对应关系,为 Web 关系挖掘奠定基础.服务集的概念格可以直接基于上述染色形式背景生成,也可以按照颜色的不同分别生成输入子概念格和输出子概念格.

### 3.2 概念度量

概念格是由概念和概念间的偏序关系形成的格结构.当对象集和属性集关系复杂时,格中概念的个数通常与对象集和属性集的基数形成指数关系,直接挖掘服务关系会导致时间开销过大.考虑到 Web 服务的输入或输出都是由特定的一组参数构成的,对应着属性集  $P$  的一个特定子集,也对应着概念格中的特定概念.引入概念覆盖度对概念进行度量,使其能够标记这些特定概念,可以优化 Web 关系的挖掘过程.

**定义 3(概念覆盖度).** 设  $C=(O_i, A_j)$  是形式背景  $K=\langle O,A,R \rangle$  中的一个概念,对于  $O_i$  中的任意对象  $o$ , 概念  $C$  对  $o$  的覆盖度定义为

$$\varphi(C|o) = \frac{|A_j|}{|o'|} \quad (3)$$

其中,  $|A_j|$  为概念  $C$  的内涵的势,即内涵  $A_j$  所包含的属性个数;  $|o'|$  为对象  $o$  的属性集的基数.

概念覆盖度  $\varphi$  表示了概念与对象的关联程度,取值满足:  $\varphi(C|o) \leq 1$ , 当且仅当  $o'=A_j$  时等号成立.规定  $o \notin O_i$  时,  $\varphi(C|o)=0$ , 则  $\varphi(C)=(\varphi(C|o_i))$ ,  $i=1,2,\dots,k$ , 称为概念  $C$  对对象集  $O$  的覆盖度.  $\varphi(C)$  是一个  $k$  维向量,取值满足:  $0 \leq \varphi(C|o_i) \leq 1$ .

**定义 4(概念格度量模型).** 概念格  $L(K)$  中所有概念对对象集  $O$  的覆盖度称为概念格的度量模型,记为  $\Gamma$ .

若以概念  $C_i$  对对象集  $O$  的覆盖度  $\varphi(C_i)$  为行向量,则概念格的度量模型  $\Gamma$  可以用二维矩阵表示.矩阵的列向量  $(\varphi(C_i|o))^T$ ,  $i=1,2,\dots,|L(K)|$ , 称为格  $L(K)$  对对象  $o$  的覆盖度,记为  $\varphi(o)$ .

### 3.3 概念格扩展模型

概念格描述了形式背景中概念间的偏序关系,并且可以用二维平面空间内的 Hasse 图表示.度量模型  $\Gamma$  描述了概念与对象间的覆盖关系,可以随着 Godin 算法在构造概念格的同时生成,是概念格构造过程的伴生矩阵.概念格及其度量模型共同构成三维空间内的扩展概念格模型:水平维度描述概念格的 Hasse 图模型,垂直维度描述 Hasse 图中概念对对象的覆盖度,如图 1 所示.形式背景  $K$  的扩展概念格记为  $\langle L(K), \Gamma \rangle$ .

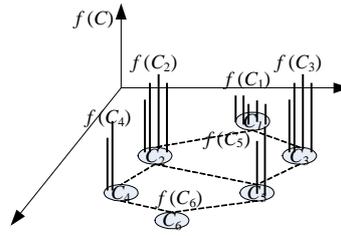


Fig.1 Extend concept lattice model  
图 1 概念格扩展模型

### 3.4 标示概念

在扩展概念格中,有如下结论成立:

**定理 1.** 格  $L(K)$  对对象  $o$  的覆盖度向量  $\varphi(o)$  中有且仅有一个元素取值为 1. 即:  $\forall o \in O$ , 存在唯一的概念  $C_j \in L(K)$ , 使  $\varphi(C_j | o) = 1$ .

证明: 首先证明存在性. 对  $\forall o \in O$ , 由引理 1 可知,  $o'' = o'$ . 因此  $C_j = (o'', o')$  是形式背景  $K$  中的一个概念. 由引理 1 可知,  $\{o\} \subseteq o''$ , 因此  $o \in o''$ . 由概念覆盖度的定义可知,  $\varphi(C_j | o) = 1$ .

再证明唯一性. 设  $C_1 = (O_1, A_1), C_2 = (O_2, A_2)$  是形式背景  $K$  中的两个概念,  $o$  是对象集  $O$  中的一个对象, 且  $\varphi(C_1 | o) = 1, \varphi(C_2 | o) = 1$ . 由概念覆盖度的定义可知,  $o' = A_1 = A_2$ . 由概念格的偏序关系可知,  $C_1 = C_2$ , 即  $C_1, C_2$  为同一概念. 证毕.  $\square$

$\varphi(C_j | o) = 1$  在 Web 服务集的概念格中有着特殊含义: 对于 Web 服务  $W_s = \{Input, Output\}$ ,  $\varphi(C_j | W_s) = 1$  表明概念  $C_j$  包含了  $W_s$  的全部输入/输出参数, 因此  $C_j$  对  $W_s$  具有标示作用.

**定义 5 (标示概念, marked concept).** 称  $C_j$  为对象  $o$  的标示概念, 当且仅当  $\varphi(C_j | o) = 1$ .

定理 1 表明, 无论格的规模有多大, 格中都只有唯一的概念完全覆盖形式背景中的一个对象, 因此, 对象的标示概念是唯一存在的. 由定理 1 的证明过程可知, 概念  $(o'', o')$  即为对象  $o$  的标示概念, 记为  $C_o$ .

**定理 2.** 概念格  $L(K)$  中标示概念的个数小于等于形式背景中对象集的基数.

证明: 由定理 1 可知, 对象集  $O$  中的任意对象  $o_1, o_2$ , 存在唯一的标示概念  $C_{o_1} = (o_1'', o_1')$ ,  $C_{o_2} = (o_2'', o_2')$ . 当  $o_1' = o_2'$  时,  $C_{o_1} = C_{o_2}$ , 即内涵不同的不同对象具有相同的标示概念. 因此,  $L(K)$  中标示概念的个数小于等于对象集中对象的个数. 证毕.  $\square$

格  $L(K)$  的所有标示概念的集合称为标示概念集, 记为  $Co$ . 定理 2 表明, 一个标示概念的内涵可以描述多个对象的内涵, 因此采用标示概念挖掘 Web 关系可以使过程得到简化, 从而提高效率.

## 4 基于扩展概念格的 Web 关系挖掘算法

采用扩展的概念格模型可以实现海量 Web 服务间关系的自动挖掘. 为表述清楚, 本文按照颜色集中颜色的不同分别生成输入子概念格和输出子概念格. 设  $K_1 = (WS, P^1, R), K_2 = (WS, P^2, R)$  分别为 Web 服务集的输入形式背景和输出形式背景. 它们分别是外延一致的, 对应的扩展概念格称为输入格和输出格, 记为  $\langle L(K_1), \Gamma_1 \rangle$  和  $\langle L(K_2), \Gamma_2 \rangle$ .

基于扩展概念格的 Web 关系挖掘过程为:

- (1) 概念格初始化: 生成 Web 服务集染色形式背景  $K' = (WS, P, M, R, F)$  上的输入扩展子概念格  $\langle L(K_1), \Gamma_1 \rangle$  和输出扩展子概念格  $\langle L(K_2), \Gamma_2 \rangle$ ;
- (2) 分别生成输入扩展子概念格和输出扩展子概念格的标示概念集  $C1_{WS}$  和  $C2_{WS}$ ;
- (3) 利用  $C1_{WS}$  和  $C2_{WS}$  提取 Web 服务等价关系, 生成输入等价服务集  $\{W_{s_{in\_Equiv_j}} | \forall W_{s_1}, W_{s_2} \in W_{s_{in\_Equiv_j}}, \text{满足: } W_{s_1} \doteq W_{s_2}\}$ 、输出等价服务集  $\{W_{s_{out\_Equiv_k}} | \forall W_{s_1}, W_{s_2} \in W_{s_{out\_Equiv_k}}, \text{满足: } W_{s_1} \doteq W_{s_2} \text{ 和等价服务集 } \{W_{s\_Equiv_l} | \forall W_{s_1}, W_{s_2} \in W_{s\_Equiv_l}, \text{满足: } W_{s_1} \equiv W_{s_2}\}$ ;
- (4) 利用  $C1_{WS}$  和  $C2_{WS}$  提取 Web 服务可替关系, 生成等价服务集的可替服务集  $\{W_{s\_Replace} | W_{s\_Equiv_i}\}$

- $\forall Ws_1 \in Ws\_Replace |_{Ws\_Equiv_i}, Ws_2 \in Ws\_Equiv_j$  满足:  $Ws_1 \geq Ws_2$ };
- (5) 利用  $C1_{WS}$  和  $C2_{WS}$  提取 Web 服务流关系,生成输入等价服务集的先序服务集  $\{Ws\_Former |_{Ws_{in\_Equiv_j}} |$   
 $\forall Ws_1 \in Ws\_Former |_{Ws_{in\_Equiv_j}}, Ws_2 \in Ws_{in\_Equiv_j}$ , 满足:  $Ws_1 \longrightarrow Ws_2$  } 和输出等价服务集的后序服务集  
 $\{Ws\_Follow |_{Ws_{out\_Equiv_k}} | \forall Ws_1 \in Ws\_Follow |_{Ws_{out\_Equiv_k}}, Ws_2 \in Ws_{out\_Equiv_k}$ , 满足:  $Ws_2 \longrightarrow Ws_1$  }.

其中,概念格初始化可以采用基于属性的快速渐进式生成算法<sup>[9]</sup>.本节着重讨论标示概念集的生成和 Web 服务间各种关系的挖掘算法.

#### 4.1 标示概念集生成

按照标示概念的定义,概念  $C$  为标示概念,则至少存在一个对象  $o$ ,使  $\varphi(C|o)=1$ .因此,标示概念集的生成规则可以描述为:寻找概念格度量模型  $\Gamma_1$  和  $\Gamma_2$  中  $\varphi(C_j|ws_i)=1$  的节点,并将对应的概念  $C_j$  分别加入到标示概念集  $C1_{WS}$  和  $C2_{WS}$  中.

**算法 1.** 输入格标示概念集生成算法.

INPUT:输入格  $\langle L(K_1), \Gamma_1 \rangle$ ;

OUTPUT:输入格标示概念集  $C1_{WS}$ ;

BEGIN

$C1_{WS} = \emptyset$ ;

/输入标示概念集初始化/

FOR  $WS$  中的每个 Web 服务  $Ws_i$ ,按下标由小到大排列 DO

FOR  $L(K_1)$  中的每个概念  $C_j$ ,按内涵的势由小到大排列 DO

IF  $(\varphi(C_j|ws_i)=1)$  THEN

/找到  $Ws_i$  的标示概念  $C_j$ /

/将  $C_j$  添加到标示概念集/

Add  $C_j$  to  $C1_{WS}$ ;

FOR  $Ws_i$  之后的每个 Web 服务  $Ws_k$  DO

IF  $(\varphi(C_j|ws_k)=1)$  THEN

/删除与标示概念内涵相同的

Web 服务,以减少后续搜索

范围/

Delete  $Ws_k$  from  $WS$ ;

END IF;

END FOR;

/跳出循环,寻找下一个 Web

服务的标示概念/

BREAK;

END IF;

END FOR;

END FOR;

Return  $C1_{WS}$ ;

/返回所有输入标示概念/

END.

算法返回输入格的标示概念集  $C1_{WS}$ .同理可以生成输出格标示概念集  $C2_{WS}$ .

#### 4.2 等价关系挖掘

等价关系是 Web 服务间的一种特殊关系.具有等价关系的 Web 服务具有完全相同的输入和输出参数,在 Web 服务调用过程中相互间可以直接替代.

通过扩展概念格挖掘 Web 等价关系的规则为:首先分别构建 Web 服务  $Ws$  的输入等价集和输出等价集,再合并产生  $Ws$  的等价服务集.

**算法 2.** Web 服务等价关系挖掘算法.

INPUT:输入标示概念集  $C1_{WS}$ 、输出标示概念集  $C2_{WS}$ ;

OUTPUT:等价服务集  $\{Ws\_Equiv_i\}$ .

BEGIN:

FOR each  $C1_j \in C1_{WS}$  DO

$Ws_{in\_Equiv_j} = \emptyset$ ;

/初始化输入等价集合/

FOR each  $Ws_i \in ext(C1_j)$  DO

IF  $(\varphi(C1_j|ws_i)=1)$  THEN

/生成所有输入等价集  $\{Ws_{in\_Equiv_j}\}$ /

Add  $Ws_i$  to  $Ws_{in\_Equiv_j}$ ;

END IF;

END FOR;

```

END FOR; /同理可生成所有输出等价集
... /{Wsout_Equivk}算法略/
FOR each Wsin_Equivj DO
  FOR each Wsi∈Wsin_Equivj DO
    Find Wsi的输出等价集 Wsout_Equivk; /建立输入、输出等价集的对应关系/
    Let Ws_Equivi=Wsin_Equivj∩Wsout_Equivk; /生成 Wsi的等价服务集 Ws_Equivi/
    Let Wsin_Equivj=Wsin_Equivj - Ws_Equivi; /删除输入等价集中的等价服务,求
    END FOR; /解下一个等价服务集/
  END FOR;
END FOR; /返回所有的等价服务集/
Return {Ws_Equivi};
END.

```

算法中,  $Ws_{in\_Equiv_j}$ ,  $Ws_{out\_Equiv_k}$  及  $Ws\_Equiv_i$  中的 Web 服务互为输入等价、输出等价和等价关系,且输入、输出等价集  $\{Ws_{in\_Equiv_j}\}$  和  $\{Ws_{out\_Equiv_k}\}$  的基数分别等于对应的输入、输出标示概念集  $C1_{WS}$  和  $C2_{WS}$  的基数,等价集  $\{Ws\_Equiv_i\}$  的基数等于概念格  $L(K_1) \bar{\cup} L(K_2)$  的标示概念集  $C_{WS}$  的基数,其中,  $\bar{\cup}$  表示两个同项概念格的横向合并运算<sup>[10]</sup>.

### 4.3 可替关系挖掘

可替关系是一种单向关系.  $Ws$  的可替集合中的 Web 服务能以不多于  $Ws$  的输入参数产生不少于  $Ws$  的输出参数,因此,在 Web 服务调用过程中可以替换  $Ws$ ,完成  $Ws$  所具有的业务功能,反之却不能.

由等价关系的定义可知,  $Ws$  的可替服务集也是  $Ws$  的等价服务集  $Ws\_Equiv$  的可替服务集.其生成算法与等价服务集类似,主要区别在于:

(1) 等价服务集  $Ws\_Equiv_i$  的输出可替服务集等于其父集  $Ws_{out\_Equiv_k}$  的输出可替服务集  $Ws_{out\_Replace_k}$ . 可以从  $Ws_{out\_Equiv_k}$  的标示概念上直接产生,判断条件为  $0 < \varphi(C2_j | w_s) \leq 1$ . 显然,  $Ws_{out\_Replace_k}$  等于输出标示概念  $C2_k$  的外延;由概念格的偏序关系可知,  $Ws_{out\_Replace_k}$  也等于输出标示概念  $C2_k$  的所有子概念的外延的并集.

(2) 等价服务集  $Ws\_Equiv_i$  的输入可替服务集等于其输入等价服务集  $Ws_{in\_Equiv_j}$  的输入可替服务集  $Ws_{in\_Replace_j}$ . 由标示概念的定义可知,输入可替关系不能直接从  $Ws_{in\_Equiv_j}$  的标示概念上直接得到,其生成算法为:

- ① 查找输入标示概念集  $C1_{WS}$  中外延包含  $Ws\_Equiv_i$  的标示概念子集  $C1_{ws_i}$ ;
- ② 对  $C1_{ws_i}$  中的每个概念  $C_j$ ,求输入等价集  $Ws_{in\_Equiv_j}$ ;
- ③  $Ws\_Equiv_i$  的输入可替服务集  $Ws_{in\_Replace_j} = \bigcup_j Ws_{in\_Equiv_j}$ .

(3) 等价服务集  $Ws\_Equiv_i$  的可替服务集合为  $Ws\_Replace | w_s\_Equiv_i = Ws_{in\_Replace_j} \cap Ws_{out\_Replace_k}$ . 显然,  $Ws\_Replace | w_s\_Equiv_i \supseteq Ws\_Equiv_i$ . 因此,Web 服务间的等价关系为一种特殊的替代关系.

### 4.4 流关系挖掘

按照输入/输出接口参数的匹配,Web 间的流关系包括:充分先序/完全后序、部分先序/不完全后序及严格先序/严格后序等.

由等价关系的定义可知,  $Ws$  的后序服务集也是  $Ws$  的输出等价集  $Ws_{out\_Equiv}$  的后序服务集,  $Ws$  的先序服务集也是  $Ws$  的输入等价集  $Ws_{in\_Equiv}$  的先序服务集.

基于扩展概念格挖掘流关系的算法为:

**算法 3.** 充分先序关系挖掘算法.

INPUT: 输入标示概念集  $C1_{WS}$ 、输出标示概念集  $C2_{WS}$ ;

OUTPUT: 充分先序服务集  $\{Ws\_Former\_Suf | w_s_{in\_Equiv_j}\}$ .

BEGIN:

FOR each  $C1_j = (WS_j, P_j^1) \in C1_{WS}$  DO

生成输入等价集  $Ws_{in\_Equiv_j}$ ;

/生成输入等价集/

/若存在相同内涵的输入、输出标

```

IF ( $\exists C2_i=(WS_i, P_i^2) \in C2_{WS}$ , 满足  $P_i^2 = P_j^1$ ) THEN
     $Ws\_Former\_Suf |_{Ws_{in}\text{-}Equiv_j} = WS_i$ ;
ELSE
     $Ws\_Former\_Suf |_{Ws_{in}\text{-}Equiv_j} = \emptyset$ ;
FOR each  $C2_k=(WS_k, P_k^2) \in C2_{WS}$  DO
    IF ( $P_k^2$  是  $C2_{WS}$  的概念内涵中  $P_j^1$  的最小父集) THEN
         $Ws\_Former\_Suf |_{Ws_{in}\text{-}Equiv_j}$ 
        =  $Ws\_Former\_Suf |_{Ws_{in}\text{-}Equiv_j} \cup WS_k$ ;
    END IF;
END FOR;
END IF;
End FOR;
Return {  $Ws\_Former\_Suf |_{Ws_{in}\text{-}Equiv_j}$  }
END.

```

算法中  $P_k^2$  是  $P_j^1$  的最小父集是指:  $P_k^2 \supset P_j^1$  且不存在另一标示概念  $C2_k'$ , 其内涵  $P_k'^2$  满足  $P_k^2 \supset P_k'^2 \supset P_j^1$ .

算法 3 以后序服务的输入参数为基准, 在输出标示概念集中搜索先序服务, 可求得每个 Web 服务及其输入等价集  $Ws_{in}\text{-}Equiv$  的充分先序服务集  $\{Ws\_Former\_Suf |_{Ws_{in}\text{-}Equiv_j}\}$ . 在此基础上, 经过适当的集合运算可以直接产生每个 Web 服务及其输出等价集  $Ws_{out}\text{-}Equiv$  的完全后序服务集  $\{Ws\_Follow\_Suf |_{Ws_{out}\text{-}Equiv_k}\}$ , 转换算法为:

**算法 4.** 充分先序与完全后序服务集的转换算法.

INPUT: 输出等价服务集  $\{Ws_{out}\text{-}Equiv_k\}$ , 充分先序服务集  $\{Ws\_Former\_Suf |_{Ws_{in}\text{-}Equiv_j}\}$ ;

OUTPUT: 完全后序服务集  $\{Ws\_Follow\_Suf |_{Ws_{out}\text{-}Equiv_k}\}$ .

```

BEGIN:
FOR each  $Ws_{out}\text{-}Equiv_k$  DO
     $Ws\_Follow\_Suf |_{Ws_{out}\text{-}Equiv_k} = \emptyset$ ;
    FOR each  $Ws\_Former\_Suf |_{Ws_{in}\text{-}Equiv_j}$  DO
        IF ( $Ws\_Former\_Suf |_{Ws_{in}\text{-}Equiv_j} \supseteq Ws_{out}\text{-}Equiv_k$ ) THEN
             $Ws\_Follow\_Suf |_{Ws_{out}\text{-}Equiv_k}$ 
            =  $Ws\_Follow\_Suf |_{Ws_{out}\text{-}Equiv_k} \cup Ws_{in}\text{-}Equiv_j$ ;
        END IF;
    END FOR;
END FOR;
Return {  $Ws\_Follow\_Suf |_{Ws_{out}\text{-}Equiv_k}$  };
END.

```

完全后序服务集也可以直接以先序服务的输出参数为基准, 在输入标示概念集中搜索后序服务, 但需要按输出参数集的幂集搜索, 效率较低. 采用转换算法求解的结果与之是等价的, 但过程更简单. 等价性的证明如下:

证明: 设  $Ws_{in}\text{-}Equiv_j$  是  $Ws_{out}\text{-}Equiv_k$  的完全后序服务集  $Ws\_Follow\_Suf |_{Ws_{out}\text{-}Equiv_k}$  的子集中任意输入等价集. 由完全后序关系的定义可知,  $Ws_{in}\text{-}Equiv_j$  的输入参数集  $Ws_{in}\text{-}Equiv'_j \subseteq Ws_{out}\text{-}Equiv'_k$ .

1° 若  $C2_{WS}$  中存在概念  $C2_i=(WS_i, P_i^2)$ , 满足  $P_i^2 = Ws_{in}\text{-}Equiv'_j$ ;

因为  $Ws_{in}\text{-}Equiv'_j \subseteq Ws_{out}\text{-}Equiv'_k$ , 所以  $Ws_{out}\text{-}Equiv'_k \supseteq P_i^2$ ,  $Ws_{out}\text{-}Equiv_k \subseteq WS_i$

由算法 3 可知,  $Ws_{out}\text{-}Equiv_k$  在  $Ws_{in}\text{-}Equiv_j$  的充分先序服务集中.

2° 若  $C2_{WS}$  中不存在概念  $C2_i=(WS_i, P_i^2)$ , 满足  $P_i^2 = Ws_{in}\text{-}Equiv'_j$ ;

a) 若  $C2_{WS}$  中存在概念  $C2_m=(WS_m, P_m^2)$ , 满足  $P_m^2 \subset Ws_{out}\text{-}Equiv'_k$ , 且  $P_m^2$  是  $C2_{WS}$  的概念内涵集中  $Ws_{in}\text{-}Equiv'_j$  的最小父集:

因为  $P_m^2 \subseteq Ws_{out}\text{-}Equiv'_k$ , 所以  $Ws_{out}\text{-}Equiv_k \subseteq WS_m$ ,  $Ws_{out}\text{-}Equiv_k$  也在  $Ws_{in}\text{-}Equiv_j$  的充分先序服务集中.

由算法 3 可知,  $WS_m$  在  $WS_j$  的充分先序服务集中.

b) 否则,  $WS_{out\_Equiv'_k}$  是  $C2_{WS}$  的概念内涵集中  $WS_{in\_Equiv'_j}$  的最小父集, 由算法 3 可知,  $WS_{out\_Equiv_k}$  在  $WS_{in\_Equiv_j}$  的充分先序服务集中.

综上所述, 输出等价集的完全后序服务集中的任意输入等价集都在算法 3 所求取的充分先序服务集中, 因此就可以按照算法 4 转换求得. 证毕.  $\square$

可见, 直求法和转换求法对于完全后序服务集的求取是等价的.

不完全后序关系的挖掘算法与算法 3 类似, 所不同的是, 它以输出参数为基准, 在输入标示概念集中求解. 部分先序服务集也可通过转换算法由不完全后序服务集求得. 这里不再赘述.

严格先序/后序关系的挖掘算法较简单, 仅需通过输入和输出标示概念集中内涵相同的标示概念求解其输入和输出等价集. 算法如下:

**算法 5.** 严格先序关系挖掘算法.

INPUT: 输入标示概念集  $C1_{WS}$ 、输出标示概念集  $C2_{WS}$ ;

OUTPUT: 严格先序服务集  $\{WS\_Former\_Exa | WS_{in\_Equiv_j}\}$ .

BEGIN:

FOR each  $C1_j=(WS_j, P_j^1) \in C1_{WS}$  DO

生成输入等价集  $WS_{in\_Equiv_j}$ ;

/生成输入等价集/

IF  $(\exists C2_i=(WS_i, P_i^2) \in C2_{WS}, \text{满足 } P_i^2 = P_j^1)$  THEN

/内涵相同的输入、输出标

生成  $C2_i$  上的输出等价集  $WS_{out\_Equiv_i}$ ;

示概念上的等价集互为

$WS\_Former\_Exa | WS_{in\_Equiv_j} = WS_{out\_Equiv_i}$ ;

严格先、后序服务集/

$WS\_Follow\_Exa | WS_{out\_Equiv_i} = WS_{in\_Equiv_j}$ ;

END IF;

END FOR;

Return  $\{WS\_Former\_Exa | WS_{in\_Equiv_j}\}$  and  $\{WS\_Follow\_Exa | WS_{out\_Equiv_i}\}$ ;

END.

由算法可以看出, Web 服务及其输入等价集的严格先序服务集即为输出格中与其具有相同内涵的输出等价集. 对于严格后序服务集, 同样具有类似的上述关系. 因此严格先序/后序关系可以统一表示为  $WS_{out\_Equiv_j} | WS_{in\_Equiv_j}$ , 其中  $WS_{out\_Equiv_j}$  称为  $WS_{in\_Equiv_j}$  严格先序服务集,  $WS_{in\_Equiv_j}$  称为  $WS_{out\_Equiv_j}$  的严格后序服务集, 它们具有相同的内涵.

## 5 基于概念格的 Web 关系维护

在不考虑服务质量变更的情况下, Web 服务集的变更包括 3 种情况: ① 新服务增加; ② 旧服务退出; ③ 旧服务接口参数变更. 其中, 变更③可以看作前两种变更的组合.

服务集的变更会引起 Web 间的关系也随之变更. 基于概念格可以实现这种变更的自动维护. 其基本过程为: (1) 更新形式背景和输入/输出概念子格; (2) 更新输入/输出标示概念集; (3) 对新增或有变更的标示概念, 更新相关的等价服务集、可替服务集和先序、后序服务集. 其中, 概念格的更新有很多自动方法, 可以参考文献 [11, 12].

### 5.1 Web 服务关系增量维护

新增对象  $WS=(Input, Output)$  对标示概念集  $C1_{WS}$  的影响为:

1° 对  $C1_{WS}$  中满足  $P_j^1 \subset Input$  的标示概念  $C1_j=(WS_j, P_j^1)$ , 更新后的标示概念为  $C1'_j=(WS_j \cup \{WS\}, P_j^1)$ ;

2° 若  $C1_{WS}$  中存在满足  $P_j^1 = Input$  的标示概念  $C1_j=(WS_j, P_j^1)$ , 则更新后的标示概念为  $C1'_j=(WS_j \cup \{WS\}, P_j^1)$ ;

若  $C1_{WS}$  中不存在满足  $P_j^1 = Input$  的标示概念  $C1_j=(WS_j, P_j^1)$ , 则新增标示概念:  $C1_{WS} = (\{WS\} \cup (\bigcup_k WS_k), Input)$ , 其

中: $WS_k$  为  $C1_{WS}$  中满足  $P_k^1 \supset Input$  的标示概念  $C1_k=(WS_k, P_k^1)$  的外延;

3°  $C1_{WS}$  中的其他标示概念保持不变.

新增 Web 服务对象后的输入标示概念集记为  $C1'_{WS}$ .同理可以求得更新后的输出标示概念集  $C2'_{WS}$ .进而可以更新 Web 服务间的各种等价、替代和流关系.受篇幅所限,仅以等价关系为例:

对等价关系有影响的是 2°中的更新后的标示概念和新增标示概念.对于前者,记更新后的输入/输出标示概念分别为  $C1'_j$  和  $C2'_k$ ,则等价关系的更新流程为:(1) 更新  $C1'_j$  和  $C2'_k$  上的输入和输出等价集  $WS_{in\_Equiv'_j}$ ,  $WS_{out\_Equiv'_k}$ ;(2) 求  $WS\_Equiv'_i=WS_{in\_Equiv'_j} \cap WS_{out\_Equiv'_k}$ ;(3) 将  $WS\_Equiv'_i$  添加到  $\{WS\_Equiv_i\}$  中,并删除  $\{WS\_Equiv_i\}$  中  $WS\_Equiv'_i$  的子集.

对于后者,记新增的输入/输出标示概念分别为  $C1'_{WS}$  和  $C2'_{WS}$ ,等价关系的更新流程同上.由于  $C1'_{WS}$  和  $C2'_{WS}$  上的输入和输出等价集都是  $\{WS\}$ ,因此  $\{WS\}$  就是 Web 服务集中新增的等价服务集.

## 5.2 Web服务关系减量维护

对象  $WS=(Input, Output)$  的退出对标示概念集  $C1_{WS}$  的影响为:(1) 对  $C1_{WS}$  中满足  $P_j^1 \subset Input$  的标示概念  $C1_j=(WS_j, P_j^1)$ ,更新后的标示概念为  $C1'_j=(WS_j - \{WS\}, P_j^1)$ ;(2) 对  $C1_{WS}$  中满足  $P_j^1 = Input$  的标示概念  $C1_j=(WS_j, P_j^1)$ ,若  $C1_j$  上的输入等价集  $WS_{in\_Equiv_j}=\{WS\}$ ,则删除标示概念  $C1_j$ ,否则更新  $C1_j$  为  $C1'_j=(WS_j - \{WS\}, P_j^1)$ ;(3)  $C1_{WS}$  中的其他标示概念保持不变.

删除 Web 服务对象后的输入标示概念集记为  $C1'_{WS}$ .同理可以求得更新后的输出标示概念集  $C2'_{WS}$ .进而可以更新 Web 服务间的各种等价、替代和流关系.受篇幅所限,这里不再详叙.

## 6 实验分析

基于概念格构造挖掘服务关系可以一次性处理海量服务,因此可用于服务集的离线管理;基于概念格维护服务关系可以针对单个服务的变更迅速修正服务间的关系,因此可用于在线管理.接口参数直接匹配<sup>[13]</sup>是另一种 Web 关系挖掘方法.本文在一个现实 Web 服务集上对这两种方法进行了对比实验.

现实 Web 服务从服务注册网站 Seekda.com 上随机产生.服务集的大小从 2 000 开始以 2 000 为跨度递增至 20 000.这 10 组随机产生的服务集相关特性见表 2.

Table 2 Web service set for testing

表 2 测试服务集数据特性

Size of service set	2 000	4 000	6 000	8 000	10 000	12 000	14 000	16 000	18 000	20 000
Size of input/output parameter set	3 128/ 2 311	6 134/ 4 566	8 915/ 6 786	11 302/ 8 570	12 971/ 9 425	13 872/ 10 390	14 494/ 11 011	14 987/ 11 232	15 147/ 11 372	15 164/ 11 440

为了对比两种方法在离线和在线处理上的不同性能,本文做了两组实验:(1) 对表 2 中的测试服务集进行离线关系挖掘;(2) 在(1)的基础上向每个测试服务集随机添加一个服务,进行服务关系的在线更新(实际实验中添加了 10 个服务,取平均值).对比两种方法在上述两组实验中的效率.每组实验重复 3 次,取平均值.对比两种方法在上述两组实验中的效率.结果如图 2、图 3 所示.

由图 2 可以看出,对于海量服务间关系的离线挖掘,无论是直接接口比较,还是基于概念格的方法,其时间开销都会随着服务集大小的增长而呈指数增长.后者增长更快,但当服务集规模达到一定程度时,增长速率会有所降低,这是由于其时间开销主要由概念格构造过程决定:对于由 Seekda.com 上随机产生的现实 Web 服务测试集而言,当测试服务集较小时,由于个体间接口参数重复率较低,服务集的扩大会导致服务集形式背景的对象集和参数属性集同时扩大,因此概念格建格时间急剧上升.当测试服务集达到一定规模后,形式背景的参数属性集趋于稳定,建格时间的上升主要由服务对象集的扩大而引起.这个拐点出现在服务集大小 12 000 之后,与表 2 中输入/输出参数的变化趋势相一致.

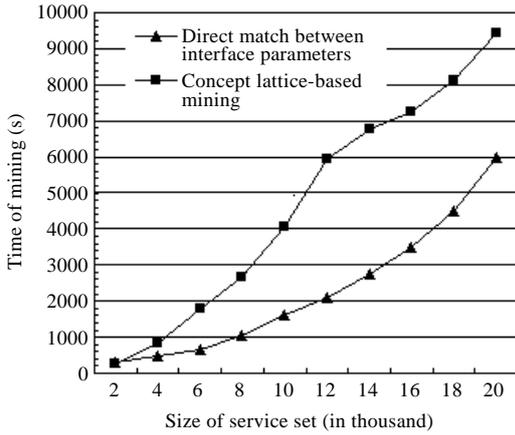


Fig.2 Efficiency of off-line Web relation mining

图2 Web 服务集离线关系挖掘效率对比

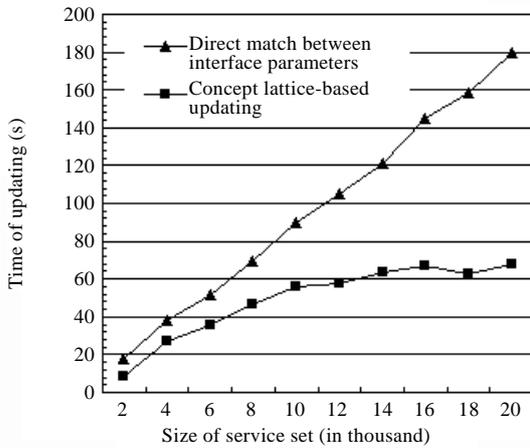


Fig.3 Efficiency of on-line Web relation updating

图3 Web 服务集在线更新效率对比

由图 3 可以看出,当新服务加入时,采用接口参数直接比较的方法更新 Web 服务间关系的时间开销较大,且随着服务集的增大呈线性增大;而基于概念格的方法时间开销较少,且随着服务集的增大而趋于稳定.这是由于,前者需要对服务集中每个服务的每个参数进行逐个比对,后者只需与标示概念集进行比对,而随着服务集的增大,标示概念集的个数趋于稳定.图 4 是服务集的大小与等价服务集的个数的对比,也可看出,Seekda.com 上注册的 Web 服务个数达到拐点大小之后,等价服务集的个数趋于稳定,表明所有 Web 服务所能完成的功能趋于稳定.当新服务加入时,总能很快找到与之等价的服务子集合.

综上,尽管基于概念格的方法在离线处理时,其时间性能逊于接口参数直接比对法,但它提供了 Web 服务集的一种有效的组织形式,在在线更新时具有显著的时间优势.对大多数用户而言,在线处理性能显然更为重要.

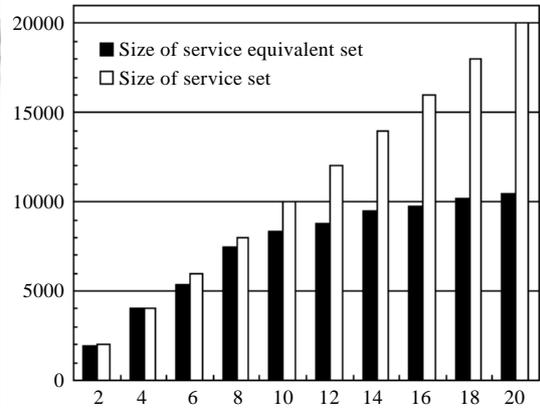


Fig.4 Comparison between size of service set and its equivalent set

图4 服务集大小与等价服务集个数的对比

### 7 相关工作比较

Web 服务管理是网格和面向服务架构的重要基础课题,随着大量资源和业务组件被封装为 Web 服务,其管理日渐成为企业应用的瓶颈,并引起了学者们的研究兴趣.

UDDI 基于分类的管理方法可以清晰地表示出 Web 服务的所属类型<sup>[14]</sup>,但不利于智能化应用.文献[15]基于服务的概念描述模型,通过计算服务类别相似度、服务描述相似度和服务参数相似度的 3 层模式提出了 Web 服务的半自动分类方法.

近年来,基于 Web 服务关系的管理思想逐渐成为主流.Web 服务资源框架(Web service resource framework, 简称 WSRF)<sup>[16]</sup>中引入 Web 服务组规范(WS-service group),将 Web 服务按照某个领域的特定的业务目的聚集或组合在一起,从而形成逻辑上的服务集合;Service Domain<sup>[17]</sup>和 Service Container<sup>[18]</sup>借助功能聚类的方法组织服务,使功能相近的服务对外呈现统一的接口,从而增强服务的柔性;XMS3.0<sup>[19]</sup>利用服务视图为后端服务建立抽象层,视图与服务间可以配置成一对多或多对一的关系,从而实现服务的功能聚类;文献[20,21]提出,将服务提供者发布的源服务按照服务包含的操作、各操作的功能和接口参数信息拆分成不同的原子服务,并映射到服务簇

和服务社区的 3 层管理思想,但没有给出具体的实现方式。

在基于关系的 Web 服务管理思想中,Web 服务关系的挖掘是核心。文献[22]基于 Web 服务功能描述的语义和本体信息构建粗糙集,并通过计算粗糙集内概念的相似度实现了 Web 服务的聚类。这种基于功能描述的方法能够有效实现相同功能的服务聚类,但其受语义约束较大,很难在开放网络环境中加以应用,并且很难用于服务流关系的自动挖掘。文献[23]基于 Web 服务的接口参数,将 Web 服务聚合问题转换为标记的树编辑距离计算问题,通过 XML Schema 进行参数类型匹配,从而实现 Web 服务的聚类和聚合。这种基于接口参数匹配的方法能够实现 Web 服务间等价、替换、流关系的自动挖掘,但其没有考虑到参数层面的语义问题,且效率较低。概念格能清晰地表达对象和属性间的二元关系,因此被用于 Web 服务管理领域。文献[24]基于概念格构建了 Web 服务描述文档的本体模型,并通过概念格的合并研究了不同 Web 服务本体的合并方法。文献[25]基于 Web 使用日志构建概念格,进行关联规则挖掘,从而发现用户感兴趣的 Web 服务,并向用户进行主动推荐。这种基于概念格的规则挖掘方法比演绎推理等方法具有更好的效率。文献[26]基于 Web 服务的方法层描述构建服务格,并给出了 Web 服务等价和替代关系的挖掘方法,但不能用于 Web 间另一种重要的关系——流关系的挖掘。本文基于 Web 服务方法的接口参数构建概念格,可以同时实现 Web 服务间等价、替代和流关系的自动挖掘,且与直接借口参数匹配的方法相比,具有较高的实时效率。

## 8 结束语

Web 服务的大量产生对服务管理提出了更高的要求。Web 服务关系的挖掘可以实现信息资源的有序化和深度聚合,从而为基于 Web 服务组合的企业应用提供智能支持。本文基于形式概念分析的方法,引入了基于概念覆盖度函数的扩展概念格,通过构建 Web 服务集基于输入/输出参数的形式背景和扩展概念格模型,给出了基于扩展概念格模型的 Web 服务间等价、替代和流关系的离线挖掘算法以及增量和减量的在线维护算法,从而实现 Web 服务关系的自动挖掘。等价关系和替代关系是服务选择和优化的基础,流关系则是服务得以组合为业务流程从而实现业务价值的基础。自动挖掘这些关系不仅可以有效地组织和管理 Web 服务,还能为 Web 服务的应用提供智能支持。在真实 Web 服务集上的测试结果表明,本文所提出的扩展概念格模型为 Web 服务集的管理提供了一种有效的组织形式。

与现有研究相比,本文的主要贡献在于,采用基于接口参数的概念格同时实现了 Web 服务间等价、替代和流关系的自动挖掘,可以与基于功能的 Web 服务关系描述相辅相成。由于本文在构建概念格时没有考虑接口参数层面的语义信息,在下一步的研究中,我们将考虑接口层面的语义描述,从而实现 Web 服务关系的精确描述。

## References:

- [1] Fan YS. Informatization Management: Strategy and Method. Beijing: Tsinghua University Press, 2008 (in Chinese).
- [2] Kim SM, Rosu MC. A survey of public Web services. Lecture Notes in Computer Science, 2004,3182:96–105.
- [3] Jiang F, Fan YS. Towards a self-organized service management system based on ecological patterns and its formal modeling. Computer Integrated Manufacturing System, 2009,15(10):1994–2003 (in Chinese with English abstract).
- [4] Ganter B, Wille R. Formal Concept Analysis Mathematical Foundations. Berlin: Springer-Verlag, 1996.
- [5] Kuznetsov SO. Machine learning and formal concept analysis. Lecture Notes in Computer Science, 2004,2961:3901–3926.
- [6] Hesse W, Tilley T. Formal concept analysis used for software analysis and modeling. Lecture Notes in Computer Science, 2005,3626:288–303.
- [7] Laukaitis A, Vasilecas O. Formal concept analysis for business information systems. Information Technology and Control, 2008,37(1):33–37.
- [8] Godin R. Incremental concept formation algorithm based on Galois lattices. Computational Intelligence, 1995,11(2):246–247. [doi: 10.1111/j.1467-8640.1995.tb00031.x]
- [9] Yang K, Ma Y, Zhang XP. Attribute-Based fast incremental building algorithm of concept lattice. Computer Applications and Software, 2006,23(12):109–112 (in Chinese with English abstract).
- [10] Li Y, Liu ZT, Chen L, Xu XH, Cheng W. Horizontal union algorithm of multiple concept lattices. ACTA Electronica Sinica, 2004,32(11):1849–1854 (in Chinese with English abstract).
- [11] Xie R, Pei Z, He CL. Reconstructing algorithm of concept lattice in adding attribute process. Journal of System Engineering, 2007,22(4):426–432 (in Chinese with English abstract).

- [12] Zhang CY, Guo JF, Liu BX. Longitudinal and transverse maintenance algorithm of concept lattice based on attributes linked-list. *Computer Engineering and Applications*, 2004,40 (5):185–187 (in Chinese with English abstract).
- [13] Feuerlicht G. Service aggregation using relational operations on interface parameters. *Lecture Notes in Computer Science*, 2007,4652:95–103. [doi: 10.1007/978-3-540-75492-3\_9]
- [14] Jiang JJ, Liu JG. Research and application of principles of UDDI classification. *Computer Engineering and Design*, 2006,27(5):811–814 (in Chinese with English abstract).
- [15] Corella MÁ, Castells P. Semi-Automatic semantic-based Web service classification. *Lecture Notes in Computer Science*, 2006,4103:459–470. [doi: 10.1007/11837862\_43]
- [16] Banks T. Web service resource framework (WSRF)-Primer v1.2 /2005-11-07. <http://docs.oasis-open.org/wsrp/wsrp-primer-1.2-primer-cd-01.pdf>
- [17] Aier S, Schönherr M. Model driven service domain analysis. *Lecture Notes in Computer Science*, 2007,4652:190–200. [doi: 10.1007/978-3-540-75492-3\_17]
- [18] Dhesiaseelan A, Raganathan V. Web services container reference architecture (WSCRA). In: *Proc. of the IEEE Int'l Conf. on Web Services (ICWS 2004)*. Washington: IEEE Computer Society, 2004. 806–807. [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=1314828](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1314828)
- [19] West Bridge Technology Inc. XML Web services management and XML firewall security solution: XML message server Version 3.1/2004. 2004. [http://www.niap-cccv.org/cc-scheme/st/st\\_vid4039-st.pdf](http://www.niap-cccv.org/cc-scheme/st/st_vid4039-st.pdf)
- [20] Zhang LJ, Li B. Requirements driven dynamic services composition for Web services and Grid solutions. *Journal of Grid Computing*, 2004,2(2):121–140.
- [21] Gao Y, Na J, Zhang B, Yang L, Ye L. 3-Layer Web services organization model for dynamic service composition. *Mini-Micro Systems*, 2006,27(10):34–38 (in Chinese with English abstract).
- [22] Bai L, Liu M. A fuzzy-set based semantic similarity matching algorithm for Web service. In: *Proc. of the 2008 IEEE Int'l Conf. on Services Computing*. Washington: IEEE Computer Society, 2008. 529–532. [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=4578571](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4578571)
- [23] He LJ, Liu LC, Wu C. Asymmetric Web service operation matching in structural-level similarity measure. In: *Proc. of the IEEE 8th Int'l Conf. on Computer and Information Technology Workshops*. Washington: IEEE Computer Society, 2008. 688–693. [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=4568584](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4568584)
- [24] Xia H, Li ZZ, Wang H. A lightweight method of Web service ontology merging based on concept lattice. In: *Proc. of the 2007 IEEE Asia-Pacific Services Computing Conf.* IEEE Computer Society, 2007. 304–311.
- [25] Zhou BY, Hui SC, Chang KY. A formal concept analysis approach for Web usage mining. In: Shi ZZ, He Q, eds. *Proc. of the 2004 Int'l Conf. on Intelligent Information Processing (IIP 2004)*. Heidelberg: Springer-Verlag, 2005. 437–441.
- [26] Peng DL, Huang S, Wang XL, Zhou AY. Management and retrieval of Web services based on formal concept analysis. In: *Proc. of the 5th Int'l Conf. on Computer and Information Technology*. Washington: IEEE Computer Society, 2005. 269–275. [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=1562662](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1562662)

#### 附中文参考文献:

- [1] 范玉顺. 信息化管理战略和方法. 北京: 清华大学出版社, 2008.
- [3] 姜峰, 范玉顺. 基于生态模式的服务管理系统及其形式化构建. *计算机集成制造系统*, 2009,15(10):1994–2003.
- [9] 杨凯, 马垣, 张小平. 基于属性的概念格快速渐进式构造算法. *计算机应用与软件*, 2006,23(12):109–112.
- [10] 李云, 刘宗田, 陈峻, 徐晓华, 程伟. 多概念格的横向合并算法. *电子学报*, 2004,32(11):1849–1854.
- [11] 谢润, 裴峥, 何昌莲. 属性添加情况下的概念格重构算法. *系统工程学报*, 2007,22(4):426–432.
- [12] 张春英, 郭景峰, 刘保相. 基于属性链表的概念格纵横向维护算法. *计算机工程与应用*, 2004,40(5):185–187.
- [14] 江建军, 刘继光. UDDI 分类机制的研究与应用. *计算机工程与设计*, 2007,27(5):811–814.
- [21] 高岩, 那俊, 张斌, 杨雷, 叶蕾. 支持动态服务组合的 Web 服务三层组织模型. *小型微型计算机系统*, 2006,27(10):34–38.



姜峰(1981—),男,江苏泰兴人,博士,主要研究领域为面向服务的企业架构,服务建模方法和技术,服务资源管理,企业信息化管理。



范玉顺(1962—),男,博士,教授,博士生导师,主要研究领域为企业信息化战略管理与规划,面向服务的企业体系架构与集成技术,企业建模与业务优化分析,企业经营过程重组与 workflow 管理,系统集成与软件互操作技术,网络化制造与制造服务技术。