

RGPS 制导的按需服务组织与推荐方法

刘建晓^{1),2)} 何克清¹⁾ 王 健¹⁾ 余敦辉¹⁾ 冯在文¹⁾ 宁 达¹⁾ 张秀伟¹⁾

¹⁾(武汉大学计算机学院软件工程国家重点实验室 武汉 430072)

²⁾(华中农业大学理学院 武汉 430072)

摘 要 随着互联网上各种 Web 服务数量的快速增长,如何发现满足用户个性化、多样化需求的服务成为亟待解决的关键问题之一.文中提出了一种面向领域问题的、基于角色-目标-流程-服务模型的按需服务组织与推荐方法.该方法在对与领域问题相关的用户角色(R)、目标(G)、流程(P)和服务(S)进行建模的基础上,利用 RGPS 元素间的关联关系对领域中服务实现了按需组织,同时根据不同的用户需求表达方式相应地设计了3种服务推荐算法.最后,通过实验验证了所提方法的有效性,并结合实际案例介绍了所开发的系统平台.

关键词 Web 服务;按需服务组织;领域问题;服务推荐;RGPS

中图法分类号 TP311 **DOI 号** 10.3724/SP.J.1016.2013.00238

An Approach of RGPS-Guided On-Demand Service Organization and Recommendation

LIU Jian-Xiao^{1),2)} HE Ke-Qing¹⁾ WANG Jian¹⁾ YU Dun-Hui¹⁾ FENG Zai-Wen¹⁾
NING Da¹⁾ ZHANG Xiu-Wei¹⁾

¹⁾(State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Wuhan 430072)

²⁾(College of Science, Huazhong Agricultural University, Wuhan 430072)

Abstract With the growth of all kinds of Web services on the internet, how to discover the services which can meet the user's personalized and diverse requirements is a key problem that needs to be resolved. This paper proposes an on-demand service organization and recommendation approach based on the user's role, goal, process and service models specific to a domain problem. Firstly, the Role(R), Goal(G), Process(P), and Services(S) models that are related to a domain problem is created. Then the services is organized according to the corresponding models and the dependent relationships among the RGPS elements. Three recommendation algorithms are also designed according to the different representation types of users' requirements. Finally, the effectiveness of the proposed method is validated through experiments, and the developed platform is introduced in combination with the practical case.

Keywords Web service; on-demand service organization; domain problem; service recommendation; RGPS

收稿日期:2012-09-19;最终修改稿收到日期:2012-10-22.本课题得到国家自然科学基金(60970017,61202031,61100017)、国家科技支撑计划(2012BAH07B01)、国家云计算示范工程“中小企业管理云应用研发与产业化”项目、华中农业大学新进博士科研启动专项基金(52902-0900206081)、武汉大学软件工程国家重点实验室开放研究基金项目(SKLSE2012-09-24)资助.刘建晓,男,1984年生,博士研究生,主要研究方向为服务计算. E-mail: liujianxiao321@163.com. 何克清,男,1947年生,博士,教授,博士生导师,主要研究领域为服务计算、互操作性理论与方法.王 健(通信作者),男,1980年生,博士,讲师,主要研究方向为服务计算. E-mail: jianwang@whu.edu.cn. 余敦辉,男,1974年生,博士,主要研究方向为需求演化.冯在文,男,1980年生,博士,讲师,主要研究方向为服务计算.宁 达,男,1984年生,博士研究生,主要研究方向为需求工程.张秀伟,男,1980年生,博士,主要研究方向为服务计算.

1 引言

面向服务的计算作为一种新的计算范型,已经在学术界和工业界产生了深远的影响.与此同时,互联网上的用户需求呈现出多样化、个性化的特点,为了给用户提供高体验质量(QoE)的服务,需要以用户为中心进行服务的有效组织与推荐,实现按需服务^[1].

在服务计算中,用户的个性化多样化需求往往体现在如下两个方面.一方面,由于不同服务的各种 QoS 值(服务质量参数)如执行时间、费用、可靠性等往往有较大差异,这样在面对大量实现相同功能而具有不同 QoS 值的 Web 服务时,偏好各异的用户其选择结果往往也有所不同.另一方面,针对同一目标,不同用户可能会选择不同的执行流程,从而导致实现这些流程的服务也有差异.这就需要从上述两个方面来考虑进行服务组织,为用户提供高质量服务.针对上述第一方面,现有研究大都采用了对服务进行聚类的方式^[2-5],将实现近似功能、且具有不同 QoS 值的服务进行聚类,这样用户首先查找到实现特定功能的服务类簇,然后根据差异化的 QoS 请求信息实现进一步的服务查找.本文主要针对第二方面展开研究,即从用户共性需求出发,对具有关联关系的服务进行组织.在服务组织方面目前已有一些研究方法,例如基于业务生成图的服务工作流^[6]、GODSS^[7]、VINCA^[8]、APDR4AWS^[9]等方法,这些方法主要是从服务执行的过程、意图、行为角度出发,针对服务间的交互与协作对服务进行组织.服务间的交互与协作是服务组织的基础,而服务间交互与协作一般都是为满足用户特定需求目标而进行的.而现有的研究方法主要考虑了服务间的流程约束关系,并未从用户自身及其需求的特点出发来考虑服务组织,这使得这些方法在进行服务组织时的层次、粒度和视角往往较为单一,同时现有的方法更多地注重利用服务描述信息(比如服务名称、输入输出等)来查找服务,查找到的服务往往是独立的服务,难以根据用户的需求来推荐有效的关联服务,从而影响了按需服务的质量.此外,由于现有的服务组织方法没有考虑用户本身需求的特点,影响了服务查找时的效率和准确率.

针对上述问题,本文的主要工作如下:

(1) 面向具体领域问题 SDP(Specific Domain Problem), 对该问题相关的用户角色、目标、流程

和服务进行建模,并定义了 RGPS 元素间的关联关系.基于 TLA(Temporal Logic of Actions)对各模型中的相关操作进行建模,通过合理的逻辑顺序实现复杂的模型操作.

(2) 从用户需求目标、用户角色以及服务之间的流程约束关系等方面进行考虑,设计了面向具体领域问题、RGPS 制导的按需服务组织算法,从不同角度、在不同层次、以不同粒度对相关服务资源进行组织管理.根据不同的用户需求表达方式相应地设计了 3 种服务推荐算法,使用户快速查找到满足其需求的最佳服务或者一组相关联的服务集.

(3) 根据提出的按需服务组织与推荐方法,利用相关实验和复杂度分析对服务查找效率等方面进行实证分析.介绍了开发的相关系统平台 Cloud-CRM,并通过实际案例对模型定义、服务组织与服务执行模块等方面进行了详细说明.

本文第 2 节介绍提出的 RGPS 制导的按需服务组织框架;第 3 节在定义 RGPS 模型以及元素间关联关系的基础上,对各模型中涉及的相关操作进行建模分析,并给出实现服务组织的算法;第 4 节详细介绍服务查找与推荐算法;第 5 节利用相关实验对所提出的方法进行验证分析,并介绍开发的相关 CloudCRM 平台;第 6 节给出相关工作介绍;最后是总结和下一步工作展望.

2 RGPS 制导的按需服务组织框架

如图 1 所示,本文提出了一种面向领域问题、RGPS 制导的按需服务组织框架来实现服务组织,该框架主要包括领域建模、按需服务组织、服务聚类和服务推荐几个方面.

(1) 领域建模

领域专家根据具体领域内用户共性需求进行建模,得到角色模型、目标模型、流程模型和服务模型,可根据相应的操作规则对这些模型中涉及的元素进行相关的操作(包括添加、配置和删除),这些模型是实现按需服务组织的基础.领域模型应尽可能的全面,从而可以对更多类型的服务进行组织,尽可能地满足用户需求.

(2) 服务聚类

根据特定类型的服务模型,把实现近似功能但不同 QoS 值的服务聚集在一起,形成特定服务模型对应的服务类簇.该服务模型是特定类型服务的抽象,并不指特定的服务实例.服务聚类方面的工作可

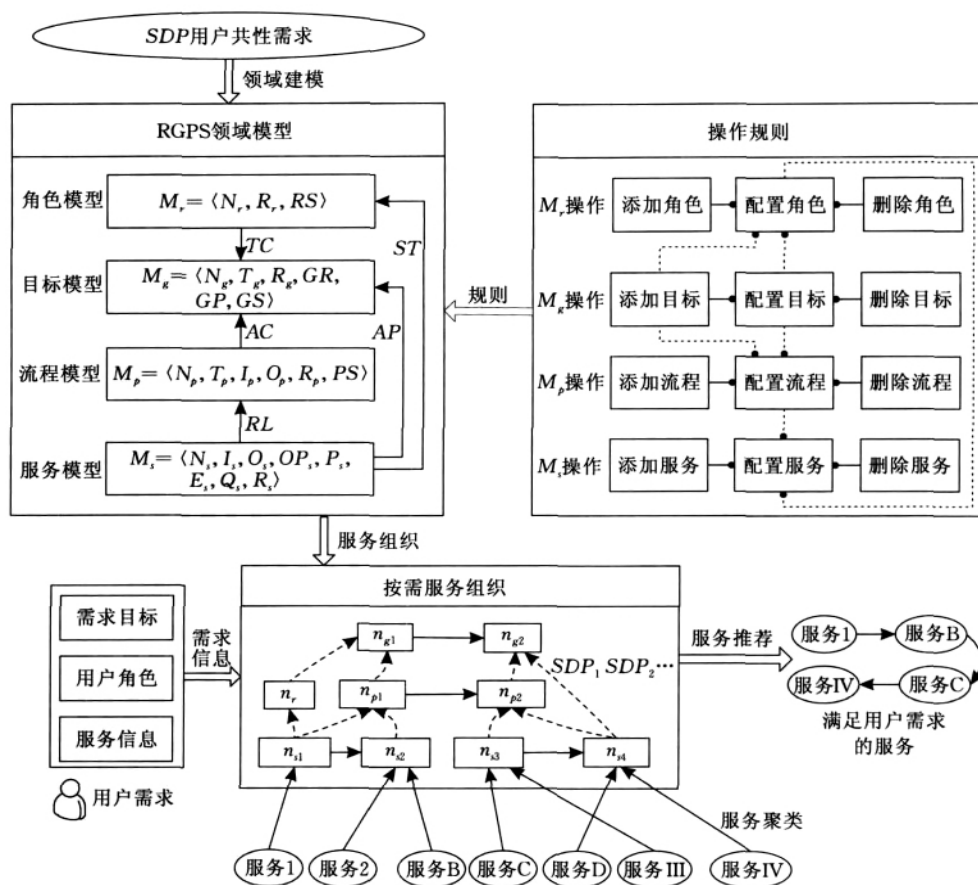


图 1 RGPS 制导的按需服务组织框架

参考文献[10], 本文对此不进行赘述.

(3) 按需服务组织

面向具体的领域问题, 根据该问题涉及的相关用户的需求目标, 同时结合用户的角色、实现目标的流程以及相应的元素间关联关系对相关服务实现面向用户需求的服务组织.

(4) 服务推荐

根据用户不同的需求表达方式, 如提出的需求目标、所属角色、或者服务描述信息, 通过不同的服务推荐算法实现服务查找和推荐, 使用户得到满足其需求的相互关联的服务集, 提高用户查找服务的效率和准确率.

3 RGPS 模型定义及相关操作

3.1 RGPS 的定义及其关联关系

角色描述了涉众的社会性属性(职责、组织及规则)及其参与者的偏好及情境约束, 是共性用户的抽象. 目标是期望待构造系统达到状态的声明, 它是对用户意图的一种说明性的描述, 可分为功能目标与非功能目标. 流程是完成特定目标的具体过程, 包括

原子流程和组合流程. 通过将角色、目标、流程和服务进行有机关联, 提出了基于 RGPS^[11-13] 的服务建模与组织方法.

在上述给出角色、目标、流程与服务定义的基础上, 下面主要介绍 4 个元素之间的关联关系, 为服务的组织规则奠定基础. 各元素间的关系见表 1.

表 1 元素间关系

元素 1	元素 2	关系	表示
角色	服务	满足	ST
目标	角色	承担	TC
目标(NG)	流程	实现	AC
目标(FNG)	流程	促进	CB
目标	服务	完成	AP
流程	角色	参与	IV
流程	服务	实现	RL

如表 1 所示, 服务可以满足特定角色需求, 完成特定目标以及实现相应的流程. 角色承担特定目标, 参与流程的构建. 流程有助于实现特定功能性目标, 促进非功能目标的实现.

3.2 模型定义

实现模型定义的开始状态为 $Init_s$, 采用 TLA 格式将其定义如下:

$Init_{\sigma} \triangleq Role \wedge Goal \wedge Process \wedge Service \wedge RS \wedge T$,

其中 $Role$ 、 $Goal$ 、 $Process$ 、 $Service$ 分别表示定义的具体领域内的角色、目标、流程和服务等领域资产, 这些领域资产可由领域建模工具生成. RS 表示 $Role$ 、 $Goal$ 、 $Process$ 、 $Service$ 之间的关联关系集合, T 表示相关变量状态的集合.

定义 1. 角色模型. 角色模型 M_r 描述了实现 SDP 所需的服务可以为哪些类型的用户角色提供服务, $M_r = \langle N_r, R_r, RS \rangle$, 其中:

$N_r = \{n_r | n_r \in Role\}$ 表示 M_r 中包括的角色集;

$R_r = \{ \langle n_{ri}, r_r, n_{rj} \rangle, n_{ri}, n_{rj} \in N_r, r_r \in RS \}$ 表示 M_r 中角色之间的关系集合, 比如 $intersect$ 、 $equivalent$ 、 $specialized$ 等;

$RS = \{ \langle n_{ri}, ST, n_{sj} \rangle, n_{ri} \in N_r, n_{sj} \in M_s, N_s \}$, 表示角色与服务模型中服务之间的 ST 关系集合.

定义 2. 目标模型. 目标模型 M_g 描述了 SDP 中包含的目标以及元素间关联关系, $M_g = \langle N_g, T_g, R_g, GR, GP, GS \rangle$, 其中:

$N_g = \{n_g | n_g \in Goal\}$ 表示 M_g 中包括的目标集合;

$T_g = \{t_g | t_g \in FG, NFG\}$ 表示目标的类型, 分为功能目标(NG)和非功能目标(NFG). NG 指可实现的目标, NFG 指服务的 QoS 目标信息, 可用定性描述〈非功能目标类型, 程度〉或定量描述〈非功能目标类型, 比较符, 值, 单位〉的形式进行表示, 本文主要考虑功能目标;

$R_g = \{ \langle n_{gi}, r_g, n_{gj} \rangle, n_{gi}, n_{gj} \in N_g, r_g \in RS \}$, 表示 M_g 中目标之间的关系集合;

$GR = \{ \langle n_{gi}, TC, n_{rj} \rangle, n_{gi} \in N_g, n_{rj} \in M_r, N_r \}$, 表示目标与角色模型中角色之间的 TC 关系集合;

$GP = \{ \langle n_{gi}, AC, n_{pj} \rangle, n_{gi} \in N_g, n_{pj} \in M_p, N_p \}$, 表示目标与流程模型中实现流程之间的 AC 关系集;

$GS = \{ \langle n_{gi}, AP, n_{sj} \rangle, n_{gi} \in N_g, n_{sj} \in M_s, N_s \}$, 表示目标与服务模型中服务之间的 AP 关系集;

其中, 参照领域特征模型中特征间的关系定义, R_g 表示的目标之间的关系主要有以下几种: $Mandatory$ 、 $Optional$ 、 $Alternative$ 和 OR .

定义 3. 流程模型. 流程模型 M_p 描述了 SDP 中实现特定目标所对应的执行流程以及元素间关联关系, $M_p = \langle N_p, T_p, I_p, O_p, R_p, PS \rangle$, 其中:

$N_p = \{n_p | n_p \in Process\}$ 表示 M_p 包括的流程集合;

$T_p = \{t_p | t_p \in AP, CP\}$ 表示流程的类型, 分别表示原子流程和组合流程;

I_p 和 O_p 表示执行流程所需要的输入和输出集;

$R_p = \{ \langle n_{pi}, r_p, n_{pj} \rangle, n_{pi}, n_{pj} \in N_p, r_p \in RS \}$ 表示包含的流程之间的控制约束关系, 主要包括 $Sequence$ 、 $Choice$ 、 $Split$ 等关系;

$PS = \{ \langle n_{pi}, RL, n_{sj} \rangle, n_{pi} \in N_p, n_{sj} \in M_s, N_s \}$, 表示流程与服务模型中服务之间的 RL 关系.

定义 4. 服务模型. 服务模型 M_s 包含了 SDP 中涉及的一些特定类型的服务以及元素间关联关系, $M_s = \langle N_s, I_s, O_s, OP_s, P_s, E_s, Q_s, R_s \rangle$, 其中:

$N_s = \{n_s | n_s \in Service\}$ 表示 M_s 包括的服务类型集合;

I_s 与 O_s 表示服务执行所需要的输入和输出集;

$OP_s = \{op_s\}$ 表示服务包含的操作集, 其中每个操作对应的输入输出均属于 I_s 和 O_s ;

P_s 和 E_s 表示服务执行所要满足的前提条件以及执行后对环境产生的影响. 采用 TLA 的表示方法, 则 P_s 表示为 $P_s \triangleq t_i \ p_i$, 其中 p_i 表示涉及的变量, $t_i \in T$. $E_s \triangleq t_i \ op_s \ t_j$, 其中 $op_s \in OP_s$, $t_i, t_j \in T$;

Q_s 描述服务的 QoS 信息, 可用〈参数名称, 程度〉或〈参数名称, 最大值, 最小值, 单位〉的形式进行表示;

$R_s = \{ \langle n_{si}, r_s, n_{sj} \rangle, n_{si}, n_{sj} \in N_s, r_s \in RS \}$ 表示 M_s 中包含的服务之间的执行依赖关系.

该服务模型包含的服务只是某一类型服务的代表, 并不指向具体的服务实例, 是一种抽象服务. 根据定义的服务模型可以引导相关服务进行聚类形成服务类簇, 同一服务类簇中的服务具有相同的功能但具有不同的 QoS 值.

3.3 基于 TLA 的模型操作

TLA(Temporal Logic of Actions)^[14-16] 是一种基于模型检验的逻辑, 能够很好地描述并发和实时系统的行为. 将系统的动作(行为)分解为有序的状态集合, 再将状态的变化描述为相应的逻辑公式.

在进行服务组织时, 会涉及角色、目标、流程以及服务模型中元素的相关操作和模型的动态变化, 这些操作离散, 有一定的时序性. 本文用 TLA 来对相关操作进行建模, 用行为来表示各个模型中涉及的操作.

各模型操作的开始状态为 $Init_{\sigma}$, 定义如下:

$Init_{\sigma} \triangleq M_r \wedge M_g \wedge M_p \wedge M_s \wedge$

$(R_{ING} = G_{ING} = P_{ING} = S_{ING} = \emptyset)$,

其中, M_r 、 M_g 、 M_p 和 M_s 分别表示已定义的角色、目标、流程和服务模型; R_{ING} 、 G_{ING} 、 P_{ING} 和 S_{ING} 分别表示正在进行操作的角色、目标、流程和服务元

素集合.

(1) 角色模型操作

① 添加角色. 在实现服务组织的基础上, 建模人员可以添加一个用户角色, 也需要配置新加角色与已有角色以及该角色与服务模型中服务之间的关系. 添加角色活动用 Add_R 表示, n_r 表示要添加的角色, 则

$$Add_R(n_r) \triangleq (n_r \notin M_r, N_r) \wedge (R'_{ING} = R_{ING} \cup \{n_r\}) \wedge (N'_r = N_r \cup \{n_r\}) \wedge (\bigwedge_{s=1}^{rs} add(n_r, ST, n_s)) \wedge (\bigwedge_{r'=1}^{rr} add(n_r, r_r, n_{r'})),$$

其中:

$$add(n_r, ST, n_s) \triangleq (n_s \in M_s, N_s) \wedge (\langle n_r, ST, n_s \rangle \in M_r, RS), \\ add(n_r, r_r, n_{r'}) \triangleq (n_{r'} \in M_r, N_r) \wedge (\langle n_r, r_r, n_{r'} \rangle \in M_r, R_r).$$

② 配置角色. 配置用户角色时, 需要配置角色与角色, 以及该角色与服务模型中服务之间的关系. 该角色可以是已存在的角色, 也可以是新增的角色. 配置用户角色活动用 $Config_R$ 表示, n_r 表示要配置的角色, 则

$$Config_R(n_r) \triangleq (n_r \in M_r, N_r) \wedge (R'_{ING} = R_{ING} \cup \{n_r\}) \wedge (\bigwedge_{s=1}^{rs} config(n_r, ST, n_s)) \wedge (\bigwedge_{r'=1}^{rr} config(n_r, r_r, n_{r'})).$$

③ 删除角色

删除角色表示删除特定角色以及与该角色相关的关联关系. 删除角色活动用 $Delete_R$ 表示, n_r 表示要删除的角色, 则

$$Delete_R(n_r) \triangleq (n_r \in M_r, N_r) \wedge (R'_{ING} = R_{ING} \cup \{n_r\}) \wedge (\bigwedge_{s=1}^{rs} delete(n_r, ST, n_s)) \wedge (\bigwedge_{r'=1}^{rr} delete(n_r, r_r, n_{r'})) \wedge (N'_r = N_r - \{n_r\}),$$

其中 $delete$ 用来删除与 n_r 有关联的元素之间的联系.

(2) 目标模型操作

① 添加目标. 建模人员可以添加一个目标, 同时也需要配置新加目标与其它元素之间的关联关系. 添加目标活动用 Add_G 表示, n_g 表示要添加的目标, 则

$$Add_G(n_g) \triangleq (n_g \notin M_g, N_g) \wedge (G'_{ING} = G_{ING} \cup \{n_g\}) \wedge (N'_g = N_g \cup \{n_g\}) \wedge (\bigwedge_{r=1}^{gr} add(n_g, TC, n_r)) \wedge (\bigwedge_{p=1}^{gp} add(n_g, AC, n_p)) \wedge (\bigwedge_{s=1}^{gs} add(n_g, AP, n_s)) \wedge (\bigwedge_{g'=1}^{gg} add(n_g, r_g, n_{g'})),$$

其中:

$$add(n_g, TC, n_r) \triangleq (n_r \in M_r, N_r) \wedge (\langle n_g, TC, n_r \rangle \in M_g, GR), \\ add(n_g, AC, n_p) \triangleq (n_p \in M_p, N_p) \wedge (\langle n_g, AC, n_p \rangle \in M_g, GP), \\ add(n_g, AP, n_s) \triangleq (n_s \in M_s, N_s) \wedge (\langle n_g, AP, n_s \rangle \in M_g, GS), \\ add(n_g, r_g, n_{g'}) \triangleq (n_{g'} \in M_g, N_g) \wedge (\langle n_g, r_g, n_{g'} \rangle \in M_g, R_g).$$

② 配置目标. 配置目标时, 需要配置目标与目标以及该目标与角色、流程、服务之间的关系. 该目标可以是已存在的目标, 也可以是新增目标. 配置目标活动用 $Config_G$ 表示, n_g 表示要配置的目标, 则

$$Config_G(n_g) \triangleq (n_g \in M_g, N_g) \wedge (G'_{ING} = G_{ING} \cup \{n_g\}) \wedge (\bigwedge_{r=1}^{gr} config(n_g, TC, n_r)) \wedge (\bigwedge_{p=1}^{gp} config(n_g, AC, n_p)) \wedge (\bigwedge_{s=1}^{gs} config(n_g, AP, n_s)) \wedge (\bigwedge_{g'=1}^{gg} config(n_g, r_g, n_{g'})) \wedge config(t_g).$$

③ 删除目标. 删除目标表示删除特定目标以及与该目标相关的关联关系. 删除目标活动用 $Delete_G$ 表示, n_g 表示要删除的目标, 则

$$Delete_G(n_g) \triangleq (n_g \in M_g, N_g) \wedge (G'_{ING} = G_{ING} \cup \{n_g\}) \wedge (\bigwedge_{r=1}^{gr} delete(n_g, TC, n_r)) \wedge (\bigwedge_{p=1}^{gp} delete(n_g, AC, n_p)) \wedge (\bigwedge_{s=1}^{gs} delete(n_g, AP, n_s)) \wedge (\bigwedge_{g'=1}^{gg} delete(n_g, r_g, n_{g'})) \wedge (N'_g = N_g - \{n_g\}).$$

流程模型与服务模型中涉及的相关操作的表示方式与上述角色模型和目标模型中操作类似, 限于篇幅, 在此不再阐述.

3.4 按需服务组织

在定义了角色模型、目标模型、流程模型、服务模型以及各模型中元素间语义关联关系的基础上, 本节面向具体领域问题, 比如交通出行、旅游规划等, 从该问题涉及的用户角色、需求目标以及服务执行流程几个方面进行考虑, 对相关的服务实现按需组织. 首先对具体领域问题定义如下.

定义 5. 具体领域问题. $SDP = \{Comp_{SDP}, Triple_{SDP}, Goal_{SDP}\}$, 其中:

$$Comp_{SDP} = \{e_i, e_i \in \{M_r, N_r, M_p, N_p, M_s, N_s\}\};$$

$$Goal_{SDP} = \{e_g, e_g \in M_g, N_g\};$$

$$Triple_{SDP} = \{\langle e_i, r_k, e_j \rangle, e_i, e_j \in \{Comp_{SDP}, Goal_{SDP}\}, r_k \in RS\}.$$

其中, $Comp_{SDP}$ 是 SDP 所涉及的角色、流程以及服务元素的集合; $Goal_{SDP}$ 表示 SDP 包含的目标集合; $Triple_{SDP}$ 表示元素间所有关联关系的集合.

可见 SDP 是通过 4 个模型以及各模型中元素之间语义关联关系组织形成的. 算法 1 给出了如何根据这些模型以及模型中元素间的关联关系生成

SDP 的过程,进而指导相应 Web 服务实现聚类以及组织. 为表示方便,将 $Triple_{SDP}$ 简写为 T_S .

算法 1. 按需服务组织算法 SerOrganization.

Input: $M_r, M_g, M_p, M_s, WS = \{ws_1, ws_2, \dots, ws_n\}$

Output: SDP_{WS}

1. $SDP_{WS} \leftarrow \emptyset, Goal_{SDP} \leftarrow \emptyset, cluster[] \leftarrow \emptyset;$
2. Foreach $n_{gi} \in M_g.N_g$
3. $Goal_{SDP} \leftarrow Goal_{SDP} \cup n_{gi};$
4. Foreach $\langle n_{gi}, TC, n_{rj} \rangle \in M_g.GR$
5. if $(n_{rj} \notin SDP)$ Then
6. $Comp_{SDP} \leftarrow Comp_{SDP} \cup n_{rj};$
7. $T_S \leftarrow T_S \cup (\langle Goal_{SDP}, n_{gi}, TC, Comp_{SDP}, n_{rj} \rangle);$
8. Foreach $\langle n_{gi}, r_g, n_{gj} \rangle \in M_g.R_g$
9. find $Goal_{SDP}, n_{gi} = n_{gi} \& Goal_{SDP}, goal_j = goal_j;$
10. $T_S \leftarrow T_S \cup (\langle Goal_{SDP}, n_{gi}, r_g, Goal_{SDP}, n_{gj} \rangle);$
11. Similar to step 8~10 and add role relations;
12. Foreach $\langle n_{gi}, AC, n_{pj} \rangle \in M_g.GP \& n_{pj} \notin SDP$
13. $Comp_{SDP} \leftarrow Comp_{SDP} \cup n_{pj};$
14. $T_S \leftarrow T_S \cup (\langle Goal_{SDP}, n_{gi}, AC, Comp_{SDP}, n_{pj} \rangle);$
15. Foreach $\langle n_{pi}, RL, n_{sj} \rangle \in M_p.PS$
16. if $(n_{pi} = n_{pj} \& n_{sj} \notin SDP)$ Then
17. $Comp_{SDP} \leftarrow Comp_{SDP} \cup n_{sj};$
18. Foreach service $ws_s \in WS$
19. if $(Sim(n_{sj}, ws_s) > \alpha)$ Then
20. $cluster[k] \leftarrow cluster[k] \cup ws_s;$
21. $T_S \leftarrow T_S \cup (\langle Comp_{SDP}, n_{pi}, RL, Comp_{SDP}, n_{sj} \rangle);$
22. Similar to step 8~10, add process and service relations;
23. Foreach $\langle n_{gi}, AP, n_{sj} \rangle \in M_g.GS$
24. $T_S \leftarrow T_S \cup (\langle Goal_{SDP}, n_{gi}, AP, Comp_{SDP}, n_{sj} \rangle);$
25. Foreach $\langle n_{ri}, ST, n_{sj} \rangle \in M_r.RS$
26. $T_S \leftarrow T_S \cup (\langle Goal_{SDP}, n_{ri}, ST, Comp_{SDP}, n_{sj} \rangle);$
27. $SDP_{WS} \leftarrow (Comp_{SDP}, T_S, Goal_{SDP});$
28. return $SDP_{WS}.$

在上述算法 1 中 SDP_{WS} 表示面向 SDP 组织的服务. 该算法根据具体领域问题中涉及的目标集合, 首先加入与特定目标具有 TC 关系的角色以及二者之间的关联, 见步 4~7. 然后添加角色与角色以及目标间的关系, 见步 8~11. 步 12~21 描述了如何根据目标与流程间的 AC 关联关系以及流程与服务间的 RL 关系添加流程与服务. 最后添加流程间、服务间、目标与服务以及角色与服务间的关联关系, 见步 22~26, 最后根据 4 个元素以及元素间的关联关系生成 SDP.

通过上述方法, 面向具体问题的相关服务在服务模型的引导下进行聚类可形成各个服务类簇, 然后从该问题涉及的用户的角色、需求目标、实现流程

等方面考虑将其组织在一起. 此方法从用户自身的需求目标出发, 同时考虑其辅助特征(角色, 执行流程)将相关联的服务组织在一起, 这样用户可以快速查找到满足其需求的一系列服务(包括原子服务和组合服务), 有助于提高服务查找效率和准确率.

4 服务查找与推荐

前面介绍了如何实现按需服务组织, 本节主要介绍如何根据用户实际需求查找和推荐合适的服务的具体过程.

在对领域服务进行组织的基础上, 用户可以从其中查找满足其需求的服务, 不同用户表达需求的方式有所不同. 用户可以直接提出需求目标来查找服务, 比如查询快递、查询天气等. 用户也可以通过服务名称、服务输入输出以及 QoS 信息来直接查找所需服务, 比如天气预报服务、价格小于 1 元的快递公司查询服务等. 用户还可以通过其所属角色来查找服务, 比如查找与学生相关的服务. 因此, 本文根据不同用户的不同的需求表达形式, 采用不同的方法在服务组织中查找合适的服务.

(1) 根据用户需求目标查找服务.

由算法 1 所述的按需服务组织方法可知, 服务是通过不同模型中元素与元素间的关联关系来实现组织的. 可以采用构造图的方法来表示组织的服务, 也就是将模型中的元素看作图中的节点, 元素间的关联关系看作节点间的边来构造图, 这样便于采用图的相关方法进行操作, 比如增加、删除节点以及查找节点等操作. 算法 2 给出了根据用户需求目标实现服务查找推荐的具体过程.

算法 2. SerDiscoveryGoal.

Input: $goal_{request}, SDP_{WS}$

Output: RWS

1. $RWS \leftarrow \emptyset, ND \leftarrow \emptyset;$
2. $num \leftarrow (SDP_{WS}.Goal_{SDP}.length + SDP_{WS}.Comp_{SDP}.length);$
3. Foreach $e_i \in (Goal_{SDP} \cup Comp_{SDP})$
4. if $e_i \in Role/Goal/Process/Service$ Then
5. $nodes[i].type \leftarrow R/G/P/S;$
6. $graph \leftarrow new Graph(nodes, num);$
7. Foreach $\langle e_p, r_k, e_q \rangle \in SDP_{WS}.Triple_{SDP}$
8. $graph.insert(new Edge(e_p, r_k, e_q));$
9. Foreach $node_g \in graph$
10. if $(node_g.type = G)$ Then
11. if $(match(goal_{request}, node_g) > \beta)$ Then

12. Foreach $node$ which has edge with $node_g$
13. if ($node.type = S$) Then $ND \leftarrow ND \cup node$
14. if ($node.type = P$) Then
15. Foreach $node_s$ has edge with $node$
16. if ($node_s.type = S$) Then
- $ND \leftarrow ND \cup node_s$;
17. Foreach $s \in N_s$, cluster of ND
18. Use QoS infor, $RWS \leftarrow RWS \cup s$;
19. return RWS .

上述算法给出了根据特定目标在通过 RGPS 制导方法组织的服务中查找服务的具体过程, 首先根据 RPGS 模型以及元素间的关联关系构造图, 将 $Comp_{SDP}$ 与 $Goal_{SDP}$ 中元素看作图中节点, 将 $Triple_{SDP}$ 中元素间的关系看作节点间的边, 并对相关的节点类型进行标记, 见步 2~8. 然后通过匹配计算查找满足需求的目标节点, 再根据目标节点利用 AP 关系进行相关的服务查找. 或者根据目标利用 AC 关系查找实现该目标的流程, 根据特定流程实现对应服务的查找. 在找到服务模型对应的服务类簇后, 根据 QoS 信息在服务类簇中进一步查找到合适的服务返回.

假设图 2 是根据上述算法构造的图, 各节点表示模型中角色、目标、流程、服务 4 个元素, 节点之间的边则表示元素之间的关联关系. 在该图中, 节点 A 和 B 表示 M_r 中的角色, 节点 C 和 D 表示 M_g 中的目标, 节点 E、F 和 G 表示 M_p 中流程, 节点 H、I、J、K 和 L 则表示 M_s 中的服务.

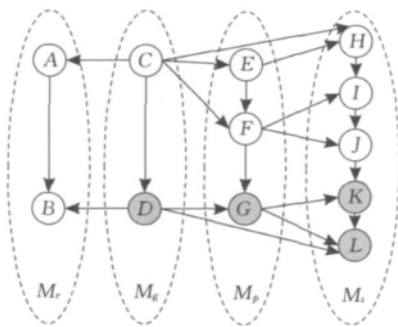


图 2 SDP 结构图

如果节点 D 与 $goal_{request}$ 之间匹配度大于阈值 (如步 11), 则通过步 12~13 直接查找到服务节点 L. 当此服务不能满足需求时, 则通过流程实现进一步查找. 可以查找到实现 D 的节点 G, 进而查找到节点 K 与 L, 见步 14~16. 在图 2 中这些节点标为灰色, 可得最后的节点 K 和 L. K 和 L 是服务模型中两个节点, 代表两类不同类型的服务, 进而对应到不同的服务类簇. 然后用户根据 QoS 需求信息可以

查找到合适的服务 (步 17~18).

(2) 根据用户角色查找服务.

同理, 采用构造图的方法来构造服务组织图, 然后根据用户角色查找服务, 具体过程如算法 3 所示.

算法 3. SerDiscoveryRole.

Input: $role_{request}$, SDP_{WS}

Output: RWS

1. $RWS \leftarrow \emptyset$, $ND \leftarrow \emptyset$, $ngoal_{mid} \leftarrow \emptyset$;
2. Use step 2~8 of algorithm 2 to construct graph
3. Foreach $node_g \in graph$
4. if ($node_g.type = R$)
5. if ($match(role_{request}, node_g) > \beta$) Then
6. Foreach $node$ which has edge with $node_g$
7. if ($node.type = S$) Then
8. $ND \leftarrow ND \cup node$;
9. if ($node.type = G$)
10. $ngoal_{mid} \leftarrow ngoal_{mid} \cup node$;
11. Foreach $node \in ngoal_{mid}$
12. Use algorithm 2($goal_{request}$) to add $servicenode$;
13. Foreach $s \in N_s$, cluster of ND
14. Use QoS infor, $RWS \leftarrow RWS \cup s$;
15. return RWS .

在上述算法中, 首先构造图, 通过用户角色与服务组织涉及的角色进行匹配计算查找到合适的角色节点, 根据 ST 关系实现服务的直接查找, 见步骤 7~8. 然后利用角色与目标之间的 TC 关系查找到 $role_{request}$ 对应的目标, 并利用算法 2 中的方法查找到对应服务. 同理, 在找到服务模型元素对应的服务类簇后, 在服务类簇中根据 QoS 信息进一步查找到合适的服务返回.

由于一个角色经常会有多个需求目标, 所以通过用户角色查找的服务个数往往会多于通过目标查找到的服务.

(3) 根据服务描述信息查找服务.

服务本身有一些相关属性信息, 如服务名称、输入、输出、前置和后置条件等, 可以根据服务的这些属性信息通过相应匹配计算直接进行服务查找, 此方面已有一些研究工作, 在此不再具体阐述.

在通过此方法找到所需服务后, 用户需要的往往不仅仅是独立的服务, 而是需要一些具有关联关系的服务集. 在通过 RGPS 制导方法形成的服务组织中, 可以利用模型元素间的关系来辅助查找和推荐具有关联关系的服务. 假设已找到的服务为 n_s , 可以利用如下 3 种方式来推荐与 n_s 有关联关系的服务. 其中, $RES(s)$ 表示服务 s 的关联服务集;

$REG(s)$ 用来表示服务 s 的关联目标集; $RER(s)$ 表示服务 s 的关联角色集.

① 服务关联法. 利用 M_s 中定义的服务间执行关系集 $R_s = \{\langle n_{si}, r_s, n_{sj} \rangle\}$ 实现关联服务查找: $((\exists n_{si} \in M_s.N_s) \wedge (n_s \equiv n_{si}) \Rightarrow RES(n_s) = n_{sj}) \cup ((\exists n_{sj} \in M_s.N_s) \wedge (n_s \equiv n_{sj}) \Rightarrow RES(n_s) = n_{si})$. 表示存在 n_{si} 属于 $M_s.N_s$ 且 n_s 与 n_{si} 等价, 则推出 n_s 的关联服务为 n_{sj} ; 当 n_s 与 n_{sj} 等价且 n_{si} 属于 $M_s.N_s$, 则推出 n_s 的关联服务为 n_{si} .

② 目标反推法. 利用 M_g 中定义的目标与服务关系集 $GS = \{\langle n_{gi}, AP, n_{sj} \rangle\}$ 查找关联服务: $(\exists n_{sj} \in M_s.N_s) \wedge (n_s \equiv n_{sj}) \wedge (\langle n_{gi}, AP, n_{sj} \rangle \in GS) \Rightarrow REG(n_s) = n_{gi}$, 然后根据 $SerDiscoveryGoal(n_{gi}, SDP_{WS}) \Rightarrow RES(n_s) = RWS$. 表示首先找到与 n_s 等价的 n_{sj} , 然后通过 GS 找到与 n_{sj} 具有 AP 关系的 n_{gi} , 最后通过 $SerDiscoveryGoal$ 找到对应的关联服务. 如图 3(a) 所示, $(n_s \equiv S_1) \wedge (\langle G_1, AP, S_1 \rangle \in GS) \Rightarrow REG(n_s) = G_1$, 根据 $SerDiscoveryGoal(G_1, SDP_{WS}) \Rightarrow RES(n_s) = \{S_2, S_3\}$.

③ 角色反推法. 同理利用 M_r 中定义的角色与服务关系集 $RS = \{\langle n_{ri}, ST, n_{sj} \rangle\}$ 查找服务: $(\exists n_{sj} \in M_s.N_s) \wedge (n_s \equiv n_{sj}) \wedge (\langle n_{ri}, ST, n_{sj} \rangle \in RS) \Rightarrow RER(n_s) = n_{ri}$, 然后根据 $SerDiscoveryRole(n_{ri}, SDP_{WS}) \Rightarrow RES(n_s) = RWS$. 表示首先找到与 n_s 等价的 n_{sj} , 然后通过 RS 找到与 n_{sj} 具有 ST 关系的 n_{ri} , 最后通过 $SerDiscoveryRole$ 找到对应的关联服务. 如图 3(b) 所示, $(n_s \equiv S_1) \wedge (\langle R_1, ST, S_1 \rangle \in RS) \Rightarrow RER(n_s) = R_1$, 根据 $SerDiscoveryRole(R_1, SDP_{WS}) \Rightarrow RES(n_s) = \{S_2, S_3, S_4\}$.

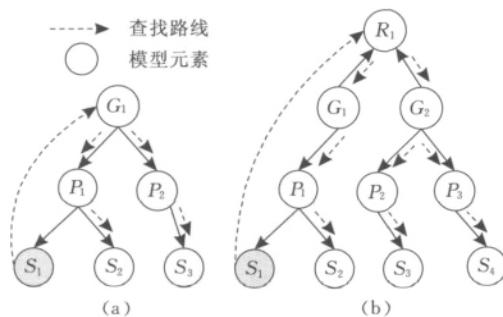


图 3 服务反推查找

总之, 针对用户不同的需求表达方式, 可以通过上述 3 种不同的服务查找与推荐算法在 RGPS 制导的按需服务组织中查找到所需的单个服务或者相关联的服务集, 从而高质量地满足用户需求.

5 实验与系统平台

5.1 实验与分析

(1) 实验环境

软件环境: Windows XP, MyEclipse 8.5 M2, Mindswap OWL-S API(<http://www.mindswap.org/2004/owl-s/api/>), xampp (<http://www.apache-friends.org/en/xampp.html>);

数据来源: OWLS-TC (<http://projects.semwebcentral.org/projects/owls-tc/>), 该数据集包含了超过 1000 个服务, 覆盖了 7 个应用领域: food、weapon、education、communication、medical care、economy 和 travel, 本文在 education 领域进行实验;

硬件环境: CPU 为双 Intel (R) Core (TM) 2 i5 CPU 760@ 2.80 GHz, 内存为 4 GB.

(2) 实验

本文主要对以下几种方法进行比较:

Random 方法: 未采用服务组织方法对服务实现组织, 即随机方法, 简称 RD;

Workflow 方法: 主要利用工作流^[6]的方式来组织服务, 利用工作流定义的服务间的关系查找服务, 简称 WF;

Goal-Service 方法: 采用 RGPS 制导的方法组织服务, 利用目标与服务信息来查找服务, 简称 GS;

Role-Service 方法: 采用 RGPS 制导的方法组织服务, 利用角色与服务信息来查找服务, 简称 RS;

Service-Service 方法: 采用 RGPS 制导的方法组织服务, 利用服务信息以及服务间的关联关系来查找服务, 简称 SS.

实验 1. 单个服务查找请求的服务查找时间对比.

本实验随机生成 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 次单个服务查找请求, 采用上述几种不同的服务组织查找方法进行服务查找, 对服务查找时间进行对比分析, 实验结果如图 4 所示.

由图 4 可见, 对于特定次数的服务查找请求, 采用 RD、WF、GS、RS 和 SS 方法查找服务的时间有所不同. 服务请求次数越多, 不同方法对应的查找时间差距越大. 对于特定次数的服务请求, RD、WF 以及 SS 方法对应的服务查找时间则相差无几, 且 3 种方法对应服务查找时间最多, 而 GS 对应的服务查找时间最少, RS 次之. 这是由于对于单个的服务请求, RD、WF 以及 SS 3 种方法在查找服务时, 没有

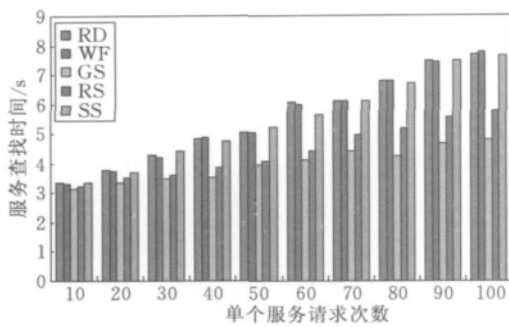


图 4 单个服务请求个数的服务查找时间对比

利用服务具有的相关属性信息,需要逐个地对服务进行匹配计算,使得比较的次数增加,从而增大服务查找时间.而GS与RS在查找服务时,分别利用了服务的目标与角色信息,可以缩小服务查找的范围,减少服务查找时间.由于特定目标对应的服务个数要少于特定角色对应的服务个数,所以GS要比RS对应的服务查找时间要少.

实验 2. 两个关联服务查找请求的服务查找时间对比.

本实验随机生成 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300 次具有关联关系的两个服务查找请求,采用不同的服务组织查找方法进行服务查找,对其服务查找时间进行对比分析,实验结果如图 5 所示.

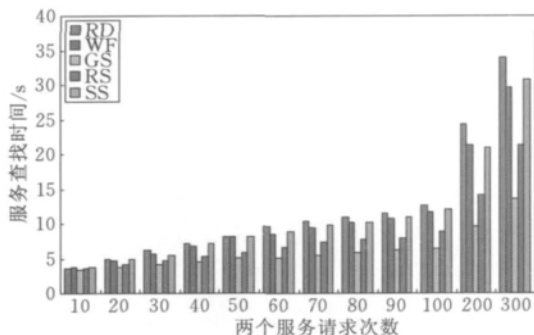


图 5 两个关联服务请求次数的服务查找时间对比

由图 5 可见,对于特定次数的两个关联服务请求,采用 RD、WF、GS、RS 和 SS 方法查找服务的时间有所不同.且服务请求次数越多,不同方法对应的查找时间差距越大.对于特定次数的关联服务请求, RD 方法对应的服务查找时间最多,GS 的服务查找时间最少,RS 次之. WF 与 SS 的服务查找时间差不多,所用时间要少于 RD 方法,但要高于 RS 与 GS 方法.同理这是由于 RD 方法没有利用服务间具有的关联关系的属性信息,需要逐个对服务进行匹配计算,导致服务查找时间最多. WF 与 SS 仅仅利用了服务间的关联关系,而 RS 与 GS 方法则利用了服

务对应的角色与目标信息,这样在通过 WF 与 SS 方法查找到某个服务 s_1 时,假如查找另外的服务 s_2 与已查找到的服务 s_1 没有关联关系,而此服务 s_2 与已查找到的服务 s_1 可以通过用户的角色与目标进行关联,这样 WF 与 SS 方法需要重新逐个匹配计算来查找 s_2 ,导致服务查找时间增多. RS 与 GS 则利用了服务间的角色、目标关联关系,可以大大缩小服务查找范围,提高效率.由于 GS 对应的服务个数要少于 RS 对应的服务个数,所以前者对应的服务查找时间最少,RS 次之.

实验 3. 不同关联服务个数的查找时间对比.

本实验在关联服务个数 k 取 1~10 的不同情况下,对于特定的服务查找请求次数(30),采用不同的服务组织查找方法进行服务查找,对查找时间进行对比分析,实验结果如图 6 所示.

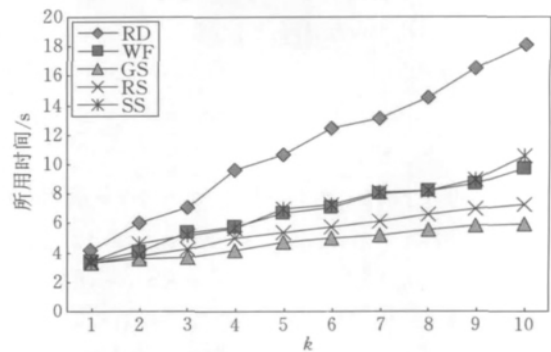


图 6 不同关联服务个数的服务查找时间对比

由图 6 可得,特定服务组织查找方法对应的服务查找时间随着 k 值的增大而增大,这是由于需要查找更多个数的服务所致.对于特定 k , RD 方法所用的时间最多,GS 最少,RS 次之,WF 与 SS 方法对应的时间基本相同,但要远少于 RD 方法,高于 GS 与 RS 方法,此结果与实验 1 和 2 保持一致.另外,随着 k 值的增大,不同方法对应的服务查找时间差距变得越来越大.

实验 4. 组织聚类服务的关联服务查找时间对比.

前面 3 个实验所查找的服务是抽象服务,也就是本文定义的服务模型中的抽象服务.由 3.2 节可见,此抽象服务是一类具有特定功能服务的代表,可以对应到不同的服务类簇,本文探讨的服务组织方法是对不同服务类簇的组织.本实验是在前面 3 个实验的基础上,将不同的抽象服务对应到不同的服务类簇.在 $k=6$ 的情况下,根据不同的关联服务请求次数,对不同服务查找方法所耗的查找时间进行统计,如表 2 所示.

表 2 组织聚类服务的关联服务查找时间($k=6$)

服务请求次数	查找时间/s				
	RD	WF	GS	RS	SS
10	15.299137	11.721731	9.030292	9.687956	11.492992
20	21.708803	16.905125	14.152886	14.200570	15.863205
30	30.462841	21.318834	15.615716	17.840528	22.855885
40	37.366802	24.418156	19.942078	22.012710	24.240080
50	44.527367	30.370731	22.080334	25.431124	30.827528
60	50.779470	34.678852	25.729418	28.464224	34.164210
70	58.323166	38.978752	30.513159	31.684042	37.772186
80	65.634660	42.552430	34.009773	34.965492	43.631676
90	74.621040	46.535732	36.691074	41.026867	48.421886
100	79.531815	53.678333	39.093155	42.466217	51.976254

由表 2 可见,在组织的服务类簇的基础上查找服务的所耗时间远多于上述几个实验中查找抽象服务所用的时间,这主要是由于首先要找到正确的抽象服务,然后在抽象服务对应的服务类簇中查找所

需服务所致.不同方法对应的时间比较与前述实验结果大致相同,在此不再阐述,但其效果更加明显.

(3) 时间复杂度分析

下面主要通过服务查找算法时间复杂度分析来进一步说明上述实验结果,分别分析比较用户在服务组织中查找单个服务以及关联服务的时间复杂度.

假设通过按需服务组织的服务数为 n ,这些服务间的关联关系数为 s , n 个服务对应的角色数为 r ,目标数为 g ,且 $n > s > g > r$.可得与每个角色具有关联关系的平均服务个数为 n/r ,同理与每个目标具有关联关系的平均服务个数为 n/g .表 3 给出了通过 RD、WF、GS、RS 和 SS 几种方法查找服务的时间复杂度(O_{RD} 、 O_{WF} 、 O_{GS} 、 O_{RS} 、 O_{SS})对比结果.

表 3 服务查找时间复杂度对比

不同情况请求	O_{RD}	O_{WF}	O_{GS}	O_{RS}	O_{SS}
单个服务请求	$O(n)$	$O(n)$	$O(n/g)$	$O(n/r)$	$O(n)$
$k(k>1)$ 个关联服务请求	$O(n \times k)$	$\{O(k_2 \times (n-2s) + k_1 \times 2s), O(n \times k)\}$	当 $k \leq g$ 时, $\{O(n \times k/g), O(n \times k)\}$ 当 $k > g$ 时, $O(n \times k/g)$	当 $k \leq r$ 时, $\{O(n \times k/r), O(n \times k)\}$ 当 $k > r$ 时, $O(n \times k/r)$	$\{O(k_2 \times (n-2s) + k_1 \times 2s), O(n \times k)\}$

注:设 $k=k_1+k_2$, k_1 为在服务关联中的服务个数, k_2 为不在服务关联中的服务个数.

设 $n=100$, $g=10$, $r=4$, $s=20$, 则 $O_{RD}=O_{WF}=O_{SS}=O(n)=100$, $O_{GS}=O(n/g)=10$, $O_{RS}=O(n/r)=25$.

当 $k=4$, 且 $k_1=3$, $k_2=1$ 时, $O_{RD}=O(n \times k)=400$, $O_{WF}=O_{SS}=\{O(k_2 \times (n-2s) + k_1 \times 2s), O(n \times k)\}=\{280, 400\}$, $O_{GS}=\{O(n \times k/g), O(n \times k)\}=\{40, 400\}$, $O_{RS}=\{O(n \times k/r), O(n \times k)\}=\{100, 400\}$;

当 $k=20$, 且 $k_1=15$, $k_2=5$ 时, $O_{RD}=O(n \times k)=2000$, $O_{WF}=O_{SS}=\{O(k_2 \times (n-2s) + k_1 \times 2s), O(n \times k)\}=\{900, 2000\}$, $O_{GS}=\{O(n \times k/g), O(n \times k)\}=\{200, 2000\}$, $O_{RS}=\{O(n \times k/r), O(n \times k)\}=\{500, 2000\}$.

通过表 3 以及相关计算分析可得, RD 方法的时间复杂度明显最高, GS 的最低, RS 则次之, WF 与 SS 方法的复杂度相同,但二者的复杂度要小于 RD 方法,大于 GS 与 RS 方法,这与前述实验结果保持一致.另外,随着服务数目的增加,上述几种服务查找方法间的时间复杂度差距变得越来越大.

5.2 CloudCRM 平台

课题组开发了一个云服务超市平台——CloudCRM^①,其中的服务组织以及服务推荐模块是用本文所述方法进行开发的.该平台是基于开源客户关

系管理系统 SugarCRM^[17] 所进行的二次开发,对 SugarCRM 进行服务化改造.由于 SugarCRM 是用 php 语言开发的,在进行服务封装时,服务的源文件是由 php 语言开发.通过下面两个步骤实现服务封装:(1)为每个功能模块创建对应的.php 文件;(2)采用 Zend Studio for Eclipse-6.1.2 将.php 文件封装为服务(.wsdl).使用两种方法为封装服务创建对应的源文件,一方面利用 SugarCRM 向外暴露的服务接口(Sugar Web Services API),将 SugarCRM 内的一些常用功能(如联系人管理、电话管理)封装为 Web 服务.另一方面通过对程序中的源代码进行抽取,并对实现某一功能的函数进行直接调用,将尚未对外暴露接口的功能封装为服务.目前已将 SugarCRM 中的 Account、Contact、Case、Lead 等共 14 个模块封装为不同服务,而将模块中的增、删、改、查等子功能封装为服务中的不同操作.

该平台在进行领域建模的基础上,面向客户关系管理以及快递货品两个具体领域问题,采用 RGPS 制导的按需服务组织方法实现了相关服务的组织.除上述封装的服务外,服务超市中还注册了互联网上开放的一些 Web API 和服务(如快递 100、

① <http://202.114.107.230:8080/CloudCrm/login.jsp>

发送短信等),支持用户根据需求进行服务选择和调用。

针对快递货品这一具体领域问题,其涉及的服务包括快递公司查询服务、联系人服务、快递 100 服务、百度地图服务、RFID 完整性检测服务和天气查询服务等.该问题对应的 RGPS 模型定义如下:

(1) 角色模型: $M_r = \langle N_r, R_r, RS \rangle$,

$N_r = \{\text{销售人员}\}; R_r = \text{NULL};$

$RS = \{\langle \text{销售人员}, ST, \text{快递查询服务} \rangle, \langle \text{销售人员}, ST, \text{天气查询服务} \rangle\}.$

(2) 目标模型: $M_g = \langle N_g, T_g, R_g, GR, GP, GS \rangle$,

$N_g = \{\text{快递货品}_G, \text{寄送货品}_G, \text{选择快递公司}_G, \text{获取收件人联系方式}_G, \text{跟踪货品}_G, \text{查询订单}_G, \text{显示订单信息}_G, \text{文本显示订单}_G, \text{地图显示订单}_G, \text{查询货品完整性}_G, \text{获取天气信息}_G\};$

$T_g = \{FG\};$

$R_g = \{\langle \text{快递货品}_G, \text{Mandatory}, \text{寄送货品}_G \rangle, \langle \text{快递货品}_G, \text{Mandatory}, \text{跟踪货品}_G \rangle, \langle \text{寄送货品}_G, \text{Depend}, \text{跟踪货品}_G \rangle, \langle \text{寄送货品}_G, \text{Mandatory}, \text{选择快递公司}_G \rangle, \langle \text{寄送货品}_G, \text{Mandatory}, \text{获取收件人联系方式}_G \rangle, \langle \text{跟踪货品}_G, \text{Mandatory}, \text{查询订单}_G \rangle, \langle \text{跟踪货品}_G, \text{Mandatory}, \text{显示订单信息}_G \rangle, \langle \text{跟踪货品}_G, \text{Mandatory}, \text{查询货品完整性}_G \rangle, \langle \text{跟踪货品}_G, \text{Mandatory}, \text{获取天气信息}_G \rangle, \langle \text{显示订单信息}_G, \text{OR}, \text{文本显示订单}_G \rangle, \langle \text{显示订单信息}_G, \text{OR}, \text{地图显示订单}_G \rangle\};$

$GR = \{\langle \text{快递货品}_G, TC, \text{销售人员} \rangle, \langle \text{寄送货品}_G, TC, \text{销售人员} \rangle, \langle \text{跟踪货品}_G, TC, \text{销售人员} \rangle \dots\};$

$GP = \{\langle \text{选择快递公司}_G, AC, \text{选择快递公司}_P \rangle, \langle \text{查询货品完整性}_G, AC, \text{查询货品完整性}_P \rangle \dots\};$

$GS = \{\langle \text{选择快递公司}_G, AP, \text{快递公司查询服务} \rangle, \langle \text{获取收件人联系方式}_G, AP, \text{联系人服务} \rangle, \langle \text{文本显示订单}_G, AP, \text{快递 100 服务} \rangle, \langle \text{地图显示订单}_G, AP, \text{百度地图服务} \rangle, \langle \text{查询货品完整性}_G, AP, \text{RFID 完整性检测服务} \rangle, \langle \text{获取天气信息}_G, AP, \text{天气查询服务} \rangle\}.$

(3) 流程模型: $M_p = \langle N_p, T_p, I_p, O_p, R_p, PS \rangle$,

$N_p = \{\text{选择快递公司}_P, \text{获取收件人联系方式}_P, \text{查询订单}_P, \text{文本显示订单}_P, \text{地图显示订单}_P, \text{查询货品完整性}_P, \text{获取天气信息}_P\};$

$T_p = \{AP\}; I_p = O_p = \text{NULL};$

$R_p = \{\langle \text{选择快递公司}_P, \text{Sequence}, \text{获取收件人}$

$\text{联系方式}_P \rangle, \langle \text{获取收件人联系方式}_P, \text{Sequence}, \text{查询订单}_P \rangle, \langle \text{文本显示订单}_P, \text{Choice}, \text{地图显示订单}_P \rangle, \langle \text{查询订单}_P, \text{Sequence}, \text{地图显示订单}_P \rangle\};$

$PS = \{\langle \text{选择快递公司}_P, RL, \text{快递公司查询服务} \rangle, \langle \text{获取收件人联系方式}_P, RL, \text{联系人服务} \rangle, \langle \text{查询订单}_P, RL, \text{快递 100 服务} \rangle, \langle \text{地图显示订单}_P, RL, \text{百度地图服务} \rangle, \langle \text{查询货品完整性}_P, AP, \text{RFID 完整性检测服务} \rangle, \langle \text{获取天气信息}_P, AP, \text{天气查询服务} \rangle\}.$

(4) 服务模型: $M_s = \langle N_s, I_s, O_s, OP_s, P_s, E_s, Q_s, R_s \rangle$,

$N_s = \{\text{快递公司查询服务}, \text{联系人服务}, \text{快递 100 服务}, \text{百度地图服务}, \text{RFID 完整性检测服务}, \text{天气查询服务}\};$

$I_1 = \{Place, Price, Weight\}, I_2 = \{Name\}, I_3 = \{Number, Express company\}, I_4 = \{Places\}, I_5 = \{Number\}, I_6 = \{Data, Places\};$

$O_1 = \{Express company\}, O_2 = \{Address, Phone Number\}, O_3 = \{DeliveryInfo\}, O_4 = \{MapInfo\}, O_5 = \{RfidInfo\}, O_6 = \{WeatherInfo\};$

$OP_s = \text{NULL};$

$P_3 \triangle \{valid\ Number, valid\ Company\};$

$P_5 \triangle \{valid\ RFID\};$

$E_3 \triangle \{NotObtained\ GetDeliveryInfor\ Obtained\};$

$E_5 \triangle \{Unchecked\ GetIntegrityInfor\ Checked\};$

$Q_s = R_s = \text{NULL}.$

根据上述快递货品相关的 RGPS 模型描述,实现的部分服务定制模块如图 7 所示.可以看出,快递货品问题涉及到的用户角色为销售者,目标包括跟踪货品、显示订单信息等,执行流程包括选择获取收件人联系方式、查询订单等.其中文本显示订单信息、地图显示订单信息如图 7(b)所示.

6 相关工作

关于服务的组织目前已有一些研究工作,文献[7]中将服务结点作为构成服务聚合组织流程模型的基本逻辑单元,该服务节点的含义类似于本文讨论的服务模型,利用服务结点按照特定的业务逻辑可以建立通用服务聚合流程模型.该方法重点讨论服务聚合流程的多目标优化的服务动态选择问题,且该方法主要从服务的业务逻辑集成角度讨论服务聚合组织的,也就是本文所述的流程.文献[6]提出了用

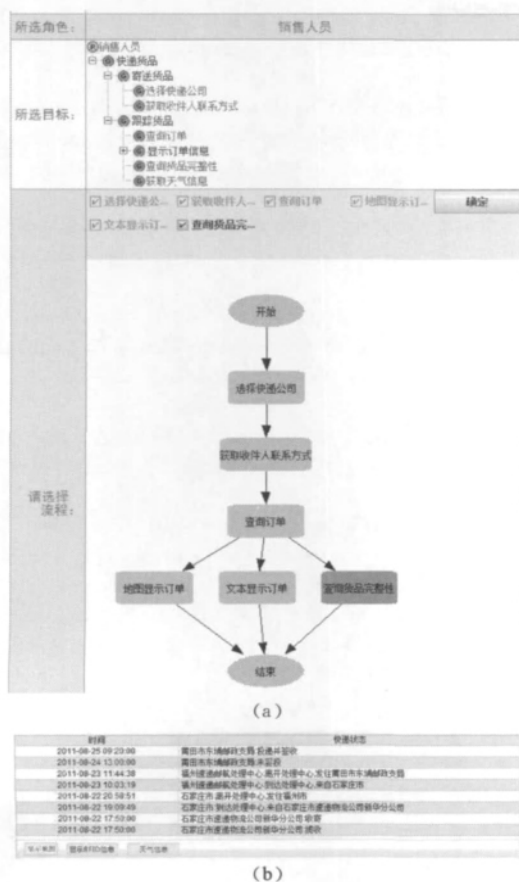


图 7 快递货品服务定制

户为中心的基于业务生成图的服务工作流构造方法,首先将功能相同或相似的服务聚集成一类服务集合,然后在服务集合的基础上采用生成树的方法对其进行组织.该方法同样重点从服务执行过程角度对服务进行组织.文献[8]中采用了业务用户编程的方式,面向业务用户组织服务资源以及以业务过程为中心进行业务级服务组合,提出了一种“中间相遇”的方法来实施服务聚合.此方法采用的用户编程方式实施服务聚合,将用户业务需求直接映射为虚拟化服务资源,从业务功能输入与非功能约束方面进行考虑查找服务实例. Howard 等人^[18-19]则为动态语义 Web 服务的组织管理提出了一个基于知识的框架 KDSWS,在虚拟组织中采用代理的方式,以工作流的形式实施语义 Web 服务交互、组合、执行和管理等.上述几种研究方法重点从执行流程方面考虑对相关服务进行组织,而本文则面向具体的领域问题,从该问题涉及用户角色、需求目标、实现流程和服务模型几个方面进行综合考虑,并从上述几个方面之间的关联关系入手,对相关服务进行组织.这样用户可以分别通过其所属角色、需求目标或者完成特定功能的业务流程来查找所需的服务,实现

按需服务,可以提高服务发现的效率和准确率.

文献[20]中提出一种用户驱动的服务聚合技术,该方法根据用户需求将功能相似的服务聚合成服务池并封装为单一的虚拟服务,随后根据用户 QoS 需求进行自动协商,选出最佳服务或服务组合,其讨论的服务池的概念类似于本文的服务模型.该方法重点是根据用户的个性化特定需求来实施服务聚合组织的,且通过该方法用户查找到的服务基本上是原子级服务.文献[21]中则提出一种以用户为中心、基于多视图合成的服务组组方法,将用户的需求转化为在多个视图编程操作方法,参照用户的视图操作合成服务组组流程.该方法从用户需求出发,参照用户需求声明的约束条件,以用户为主导、探索式地对服务进行组织.文献[22]中提出一种面向个性化需求的虚拟服务资源整合方法,首先以多维度分类树的方式对服务资源进行描述,然后根据用户个性化需求,将其经过分类、约简并以规范、可重用的方式进行表达,提出了基于动态剪枝的服务资源整合方法.该方法可以兼顾用户需求和资源组织的特点,通过动态整合批量资源形成虚拟的大粒度资源来快速满足个性化应用需求.文献[9]引入具有主动搜索需求能力的自主 Web 服务概念,提出基于环境本体的意图-行为-实现机制,提出了一种需求驱动的自主 Web 服务聚集的模型.该方法主要将服务建模为具有实现某些意图行为的自主计算实体,使服务根据用户需求实现自主性的聚集来满足用户需求.上述几种方法主要从用户实时需求出发,通过相应方法实施服务组织的.而本文则是侧重于根据具体的领域问题对应的用户功能需求,将相关服务根据该问题涉及的用户角色、需求目标、实现目标的流程之间的依关联系进行组织,从而为实现用户的按需服务选择奠定基础,而不是根据用户的需求实时进行服务组织.

文献[23]提出了一个支持动态服务组合的 Web 服务 3 层组织模型,按服务包含的操作、各操作的功能和接口参数信息将它们映射到不同的组织单元.该方法没有考虑服务间的协作关系以及涉及的用户信息来组织服务. Benatallah 等人^[24]提出了服务社区概念,通过服务社区来对服务进行组织,可以应用于动态分布式的 Web 执行环境中.在文献[25]中采用数学模型方式,提出了一个 Web 服务管理框架来实现动态业务过程配置,从服务功能角度,面向任务等方面组织服务来满足用户需求.

7 总 结

在面向服务的软件工程中,为了满足用户个性化、多样化、动态变化的需求,需要从服务执行的过程、用户本身角色以及其需求目标等方面进行考虑,对相应的服务进行组织来提高服务发现的效率和准确度. 本文的主要的工作如下:

(1) 通过对具体领域问题中的用户共性需求进行建模,从其涉及的用户角色、需求目标、流程几个方面进行考虑,设计了相应的按需服务组织算法.

(2) 设计了 3 种服务查找和推荐方法来满足用户提出的不同需求表达形式,更好地指导用户利用服务间的交互与协作来查找合适的服务以满足其需求.

(3) 设计了具体的实验对服务查找效率等方面进行验证分析,根据实际案例对相关定义和算法进行阐述,并介绍了开发的相关系统平台 CloudCRM.

下一步的研究工作主要包括:根据用户个性化需求使用户参与服务流程定制以及对定制的流程进行正确性验证等,从而实现更为个性化的按需服务.

参 考 文 献

- [1] Dominique G, Vlad T, Stamatis K, Patrik S, Domnic S. Interacting with the SOA-based Internet of Things: Discovery, query, selection, and on-demand provisioning of Web services. *IEEE Transactions on Services Computing*, 2010, 3(3): 2223-235
- [2] Sun Ping, Jiang Chang-Jun. Using service clustering to facilitate process-oriented semantic Web service discovery. *Chinese Journal of Computers*, 2008, 31(8): 1340-1353(in Chinese)
(孙萍, 蒋昌俊. 利用服务聚类优化面向过程模型的语义 Web 服务发现. *计算机学报*, 2008, 31(8): 1340-1353)
- [3] Sudha R, Thamarai S S. Semantic grid service discovery approach using clustering of service ontologies//*Proceedings of the IEEE TENCON*. Hong Kong, China, 2006: 1-4
- [4] Richi N, Bryan L. Web service discovery with additional semantics and clustering//*Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*. Washington, 2007: 555-558
- [5] Ram S, Hwang Y, Zhao H M. A clustering based approach for facilitating semantic Web service discovery//*Proceedings of the 15th Annual Workshop on Information Technologies & Systems*. Las Vegas, USA, 2006: 1-6
- [6] Hu Chun-Hua, Wu Min, Liu Guo-Ping, Xu De-Zhi. An approach to constructing Web service workflow based on business spanning graph. *Journal of Software*, 2007, 18(8): 1870-1882(in Chinese)
(胡春华, 吴敏, 刘国平, 徐德智. 一种基于业务生成图的 Web 服务工作流构造方法. *软件学报*, 2007, 18(8): 1870-1882)
- [7] Liu Shu-Lei, Liu Yun-Xiang, Zhang Fan, Tang Gui-Fen, Jing Ning. A dynamic Web services selection algorithm with QoS global optimal in Web services composition. *Journal of Software*, 2007, 18(3): 646-656(in Chinese)
(刘书雷, 刘云翔, 张帆, 唐桂芬, 景宁. 一种服务聚合中 QoS 全局最优服务动态选择算法. *软件学报*, 2007, 18(3): 646-656)
- [8] Zhao Zhuo-Feng, Han Yan-Bo, Yu Jian, Wang Jian-Wu. A service virtualization mechanism for business user programming. *Journal of Computer Research and Development*, 2004, 41(12): 2224-2230(in Chinese)
(赵卓峰, 韩燕波, 喻坚, 王建武. 一种支持业务用户编程的服务虚拟化技术——VINCA 聚合服务机制. *计算机研究与发展*, 2004, 41(12): 2224-2230)
- [9] Ye Rong-Hua, Jin Zhi, Wang Pu-Wei, Zheng Li-Wei, Yang Xia-Fen. Approach for autonomous Web service aggregation driven by requirement. *Journal of Software*, 2010, 21(6): 1181-1195(in Chinese)
(叶荣华, 金芝, 王璞巍, 郑丽伟, 杨夏芬. 一种需求驱动的自主 Web 服务聚集方法. *软件学报*, 2010, 21(6): 1181-1195)
- [10] Liu J X, He K Q, Wang J, Ning D. A clustering method for Web service discovery//*Proceedings of the 8th International Conference on Services Computing*. Washington DC, USA, 2011: 729-730
- [11] He Ke-Qing, Peng Rong, Liu Wei, Wang Jian, Li Bing. *Network Software*. Beijing: Science Press, 2008(in Chinese)
(何克清, 彭蓉, 刘玮, 王健, 李兵. *网络式软件*. 北京: 科学出版社, 2008)
- [12] Wang J, He K Q, Gong P, Wang C, Peng R, Li B. RGPS: A unified requirements meta-modeling frame for networked software//*Proceedings of the 30th IWAAPF*. Leipzig, Germany, 2008: 29-35
- [13] He K Q, Wang J, Liang P. Semantic interoperability aggregation in service requirements refinement. *Journal of Computer Science and Technology*, 2010, 25(6): 1103-1117
- [14] Lampot L. The temporal logic of actions. *ACM Transactions on Programming Languages and Systems*, 1994, 16(3): 872-923
- [15] Liu Hui. Research on Web service description and synthesis using temporal logic of actions (TLA) [M. S. dissertation]. Southeast University, Nanjing, 2008: 17-30(in Chinese)
(刘慧. 用动作时序逻辑(TLA)进行 Web 服务描述和合成的研究[硕士学位论文]. 东南大学, 南京, 2008: 17-30)
- [16] Shi Yu-Liang, Luan Shuai, Li Qing-Zhong, Dong Jin-Li, Liu Fang-Fang. TLA based customization and verification mechanism of business process for SaaS. *Chinese Journal of Computers*, 2010, 33(11): 2055-2067(in Chinese)
(史玉良, 栾帅, 李庆忠, 董晋利, 刘方方. 基于 TLA 的 SaaS 业务流程定制及验证机制研究. *计算机学报*, 2010, 33(11): 2055-2067)
- [17] John M, Roger S, Collin L, Matthew H. The definitive guide to SugarCRM better business applications. <http://www.springeronline.com>, 2009: 55-75
- [18] Howard H, Kerschberg L. A framework for dynamic semantic Web services management. *International Journal in Cooperative Information Systems*, 2004, 13(24): 1-34

- [19] Howard H, Kerschberg L. A knowledge-based framework for dynamic semantic Web services brokering and management//Proceedings of the 15th International Workshop on Database and Expert Systems Applications. Zaragoza, Spain, 2004: 174-178
- [20] Liu Xuan-Zhe, Huang Gang, Mei Hong. Consumer-centric service aggregation: Method and its supporting framework. Journal of Software, 2007, 18(8): 1883-1895(in Chinese)
(刘贤哲, 黄罡, 梅宏. 用户驱动的服务聚合方法及其支撑框架. 软件学报, 2007, 18(8): 1883-1895)
- [21] Ding Wei-Long, Wang Jing, Zhao Shuan. A user-centric service composition method synthesizing multiple views. Chinese Journal of Computers, 2011, 34(1): 131-142 (in Chinese)
(丁维龙, 王菁, 赵栓. 一种用户为中心、基于多视图合成的服务组方法. 计算机学报, 2011, 34(1): 131-142)
- [22] Chu Dian-Hui, Wang Xian-Zhi, Wang Zhong-Jie, Xu Xiao-Fei. Personalized requirement oriented virtual service resource aggregation method. Chinese Journal of Computers, 2011, 34(12): 2370-2380(in Chinese)
(初佃辉, 王显志, 王忠杰, 徐晓飞. 面向个性化需求的虚拟服务资源整合方法. 计算机学报, 2011, 34(12): 2070-2380)
- [23] Gao Yan, Na Jun, Zhang Bin, Yang Lei, Ye Lei. S-layer Web services organization model for dynamic service composition. Mini-micro Systems, 2006, 27(10): 1879-1882 (in Chinese)
(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): 1879-1882)
- [24] Benatallah B, Dumas M. Declarative composition and peer-to-peer provisioning of dynamic Web services//Proceedings of the 18th International Conference on Data Engineering (ICDE). Washington, 2002: 297-308
- [25] Zhang L J, Li B. Requirements driven dynamic services composition for Web services and grid solutions. Journal of Grid Computing, 2004, 2(2): 121-140



LIU Jian-Xiao, born in 1984, Ph. D. candidate. His main research interest is service computing.

HE Ke-Qing, born in 1947, Ph. D., professor. His main research interests include service computing, semantic interoperability.

Background

Service-oriented computing is emerging as a new promising computing paradigm for developing software applications. In order to meet user's service requirements in high quality, not only one or more discrete services need to be provided, but also a set of services that can be composed are needed, which indicates service aggregation is important to achieve user's diversified and personalized requirements

Specific Domain Problem (SDP) can be realized by a group of connected services. This research concentrates on how to aggregate services from the aspects of Role(R), Goal (G), Process(P), and Service(S) that are related to SDP. Role depicts the social attributes of the stakeholders, actor's preference and context constraints. Goal refers to the objective or expectation to be achieved. Process means the concrete process of achieving the specific goal. Service can be the specific and the abstract Web service. The RGPS framework is formed based on these four aspects to provide a guideline for service-oriented requirements elicitation, analysis, service

WANG Jian, born in 1980, Ph. D., lecturer. His main research interest is service computing.

YU Dun-Hui, born in 1974, Ph. D.. His main research interests is requirements evolution.

FENG Zai-Wen, born in 1980, Ph. D., lecturer. His main research interest is service computing.

NING Da, born in 1984, Ph. D. candidate. His main research interest is requirement engineering.

ZHANG Xiu-Wei, born in 1980, Ph. D.. His main research interest is service computing.

selection, and so on. It is proposed in the research of the National 973 Program of "requirements engineering, the basic research for software engineering of complex systems". In order to provide a service supermarket for service organization and service aggregation, the research group has developed a corresponding platform named CloudCRM and the core theory of CloudCRM is based on this paper.

This research is supported by the National Natural Science Foundation of China under Grant Nos 60970017, 61100017, 61202031, the National Science & Technology Pillar Program of China under Grant No 2012BAH07B01, the Central Grant Funded Cloud Computing Demonstration project of China Undertaken by Kingdee Software, the Huazhong Agricultural University New Doctoral Scientific Research projects under Grant No 52902-0900206081, the Open Funds of the State Key Laboratory of Software Engineering, Wuhan University under Grant No SKLSE2012-09-24.