

面向个性化需求的虚拟服务资源整合方法

初佃辉 王显志 王忠杰 徐晓飞

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 个性化与整合已成为现代服务生态系统的重要特征, 对此提出一种面向个性化需求的虚拟服务资源整合方法. 首先, 服务资源被以多维度分类树的方式进行描述, 而顾客的个性化需求经过分类、约简并以规范、可重用的方式得以表达. 在此基础上, 提出基于动态剪枝的服务资源整合方法. 该方法兼顾了顾客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从而快速满足个性化应用需求. 实验表明利用该方法可能获得良好的效果和性能.

关键词 虚拟化; 资源整合; 多维度分类描述; 个性化需求; 动态剪枝

中图法分类号 TP311 **DOI 号**: 10.3724/SP.J.1016.2011.02370

Personalized Requirement Oriented Virtual Service Resource Aggregation Method

CHU Dian-Hui WANG Xian-Zhi WANG Zhong-Jie XU Xiao-Fei

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

Abstract Personalization and composition are two important features of modern service eco-systems nowadays. A personalized requirements oriented virtual resource aggregation method is proposed in this paper. Service resources are formally described based on multi-dimensioned classification. Personalized requirements from applications are classified, reduced and finally expressed in formal and reusable manners. On this basis, a dynamic pruning based resource aggregation method is presented. The method pays attention to the features of both customer requirements and the organization of service resources. Multiple resources can be dynamically aggregated into coarse-grained virtual resources thus satisfy the requirements rapidly. Experiments show fairly good effectiveness and efficiency of the proposed method.

Keywords virtualization; service resource aggregation; multi-dimension description; personalized requirements; dynamic pruning

1 引 言

随着服务计算技术的发展, 越来越多的业务活动通过服务化的 Web 环境产生、运作和交付. 企业间的协同与互操作、用户之间的信息共享以及企业

与用户在整个服务过程中的协作成为必然的趋势. 由封闭的单一组织所提供的服务资源已经难以适应顾客需求的个性化趋势和整合需求, 越来越多的增值服务需要多个组织提供的服务、资源来支持完成, 使得网络成为一个机-机乃至人-机协同动作的平台. 同时, 服务资源的概念延伸成为包括软件、硬件

收稿日期: 2011-07-28; 最终修改稿收到日期: 2011-10-31. 本课题得到国家自然科学基金(61033005, 70971029, 60803091)和山东省科技攻关计划项目(2010GZX20126, 2010GGX10116)资助. 初佃辉, 男, 1970 年生, 博士研究生, 副教授, 主要研究方向为服务计算与服务工程、面向服务的软件体系结构. E-mail: cdh2004@sina.com. 王显志, 男, 1984 年生, 博士研究生, 主要研究兴趣为双边服务资源整合和服务组合. 王忠杰, 男, 1978 年生, 博士, 副教授, 主要研究兴趣为服务工程、面向服务的体系结构、服务组合和软件工程. 徐晓飞, 男, 1962 年生, 博士, 教授, 博士生导师, 主要研究领域包括服务科学与工程、企业计算、服务计算等.

和人力等各种支持服务运作且可被概念化和管理的模块化服务要素。大量的服务资源被按照一定的规则、服务目的或者合作协议有机地联合成一个虚拟整体,通过创新的服务内容和多种现代化的服务方式来满足更广范围顾客的个性化需求。在网络化服务平台的背景下,越来越多服务场景中,用户或者应用对于服务资源的需求是对批量资源的整合需求,同时具有个性化的特点。如何合理地选择批量的以及分布式的服务资源以支持用户的个性化整合需求成为难点。

在面向服务的架构(SOA)环境下,服务也是一种资源,可以通过服务组合的手段整合到一起,即通

过对不同功能 Web 服务的选择、匹配和组合,形成大粒度软件构件以满足复杂或者增值应用需求。在服务生态系统的背景下,同一服务需求同样可能需要多种资源共同作用来完成,这些资源通过整合形成一个虚拟的整体,称为虚拟服务资源。服务生态系统中涉及的资源类型更多,从单纯的软件或者服务构件扩展为软件、硬件、人力资源和环境等多种服务要素。服务生态系统中对于应用的约束也更为复杂。与 SOA 环境下服务组合中普遍采用的端到端约束相比,服务资源整合的约束条件将表达对于不同粒度资源集合的限制、对比、规则在内的多种语义。表 1 对服务资源整合与服务组合的区别进行了简单总结。

表 1 服务组合与资源整合的比较

| 比较的类别/方面 | Web 服务的整合(组合) | 广义服务资源的整合 |
|----------|-------------------------------------|--|
| 整合的对象 | 以 Web 服务为主的 SOA 软件系统要素 | 软件、硬件、人力资源、条件与环境等服务生态系统要素 |
| 整合的方法 | Web 服务通过服务流程以 orchestration 的方式组织起来 | 资源因为功能上的共同点而以虚拟资源的方式组织在一起 |
| 被整合对象的交互 | 服务流程中服务间的交互关系是明确、可以被定义的 | 资源间的交互是不确定的,许多资源是有 agent 特征的行为主体 |
| 整合的约束条件 | 典型包括功能要求和端到端的质量约束 | 复杂的个性化需求和约束条件 |
| 其它方面 | 组合后的服务可以作为成员与其它服务进一步组合 | 虚拟服务资源通常为特定领域的任务而生,同样可以视为单个大粒度资源应用于更高层次的整合任务 |

本文首先介绍基于社区的服务外包平台 CFI,并基于该应用背景提取出广义服务生态系统中的资源整合问题。鉴于虚拟服务资源的形成涉及服务资源、个性化需求、资源整合方法及评价准则等关键要素,本文的主要工作包括:(1)为了对服务资源整合需求进行有效表达,本文以多维度且多属性的方式对服务资源按照不同标准刻画分类和级别进行描述,并基于该描述提出了个性化服务资源整合需求的可重用模式和规范;(2)给出虚拟资源的质量评价方法,并提出基于分类树剪枝的虚拟资源的整合方法,实现个性化需求条件下高效和优化的服务资源选择;(3)通过实现对提出的方法与传统方法进行对比,验证了该方法的有效性和高效性。

2 服务外包应用场景

IBM 基于社区的软件外包 CFI(Call For Implementation)平台^[1]是一个典型的虚拟服务资源应用场景。传统的软件开发由有限的开发商来完成、人力、财力等各方面的制约严重影响产品开发的质量和进度。而事实上当今网络上存在着众多具备项目管理、程序开发和设计等经验且廉价的人员和团队,形成强大的网络社区。充分利用社区的力量实现高效、廉价的软件工程任务成为 CFI 平台的主要目标。

图 1 展示了 CFI 平台进行软件外包服务的基本过程。作为服务外包的发包商和外包从业人员的

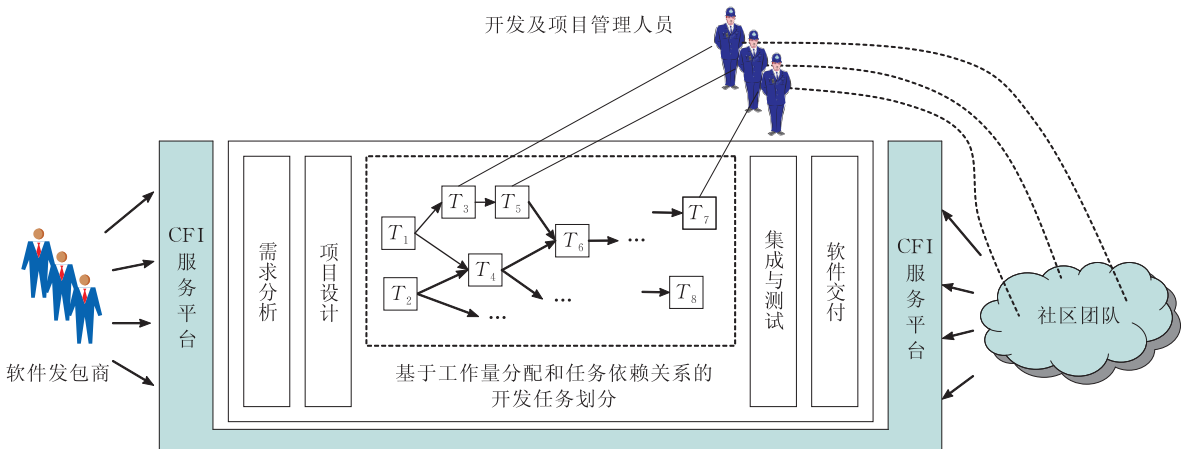


图 1 基于社区的 CFI 服务外包平台

中介平台,CFI 平台对软件工程中不同阶段(这里侧重于介绍“开发”阶段)的任务进行划分,并将这些任务分派给开发与测试社区的不同人员来完成,从而有效实现了基于 Web 的软件外包。

根据软件工程本身的流程,可将平台的功能分为以下模块:需求获取与分析、项目设计与子模块划分、基于任务量估计和关联分析的系统开发任务划分、基于任务划分的模块及集成测试、软件动态部署的运行测试以及软件实体的交付。从交互过程来看,一方面,软件外包的发包商通过与 CFI 服务平台(以软件或者人员等形式)的交互形成初始需求文档;另一方面,网络社区中的开发人员通过向平台注册、项目任务选择、实际项目任务分配等步骤而获得满足自身权限的资源并参与软件开发和测试任务;此外,在基于任务划分的开发和测试过程中,参与不同任务的不同角色的人员之间也可根据项目调度计划通过平台进行必要的交互。

在外包过程中,CFI 平台首先根据已有的需求、设计方案对软件工程的开发、测试两阶段进行任务分解,分解中考虑到不同子任务间的纵向、横向关联及依赖关系,划分结果的每个任务都带有明确的开发规范或测试说明,并通过 CFI 平台发布给开发、测试人员的社区并发布相应的人员需求供来自社区的人员申请。同时,平台/服务外包的发包商还将负责制定相应的约束和个性化条件以确保招聘人员的素质能够符合项目的具体要求和特点。最后,平台对申请每个任务的人员进行自动地优化选择,组成具有一定的组织结构、相互配合的虚拟团队,共同完成相应的任务。

在 CFI 平台中,软件的开发和测试人员被视为服务生态系统的资源,而软件的开发和测试任务成为对于这些资源的整合需求。为了选取适当资源整合为相应的虚拟资源,平台对人员的素质进行了定义和层次划分,并基于这些划分对不同任务的具体要求进行了刻画,以此作为自动化外包人员选择的基础。

3 资源整合框架及定义

由 CFI 泛化到更广泛的场景,可以发现,在服务生态系统背景下,服务资源整合的根本任务在于根据用户或应用的个性化需求从中优化地选择部分个体形成虚拟资源。该问题的基本框架如图 2 所示,包含输入、输出及资源整合引擎 3 部分:

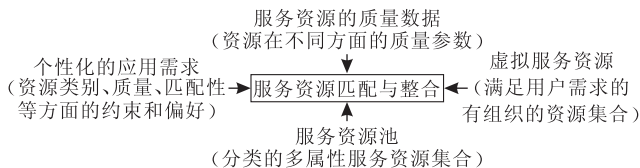


图 2 服务资源整合框架

(1) 输入包括服务资源池、应用需求描述及服务资源的质量。服务资源池是服务资源的集合,同一大类资源可以在不同的分类标准中处于不同的小类,且具有不同的评价标准;应用需求描述是以规范的形式描述和表达用户的个性化需求,能够描述和使计算理解用户需求是实现个性化服务资源整合的前提;服务资源质量包括每个资源各方面的质量属性,可以通过历史上使用资源时的质量记录以及服务资源提供者声明的质量两方面进行考察。

(2) 输出是服务资源的整合结果,即通过匹配和选择资源得到的一个资源集合,同时这些资源在集合中的组织结构中可能处于不同的位置。这些共同点和内在联系使得这些资源共同构成一个能够较好地满足用户个性化应用需求的资源集合。当生成多个结果集时,根据评价结果的不同,可以将它们分为不同的优先级。

(3) 资源整合引擎执行服务资源整合方法,根据用户需求的个性化约束条件、资源本身的类别和质量特征,从资源池中选取出一组资源集合形成能够满足用户需求的虚拟服务资源。

本节将从服务资源、个性化需求和整合评价三方面对虚拟服务资源优化构造过程中涉及的基本概念和方法进行定义和描述,具体的服务资源整合方法将在第 4 节中给出。

3.1 多维度多属性服务资源描述方法

资源整合需求的描述要符合服务资源的描述方式。现实中某一领域的资源,往往都要按不同标准进行分类、体现为服务资源的多维度分类;此外领域资源还具有一定数目的通用属性和领域属性,体现为服务资源的多属性,如图 3 所示。在 CFI 中,开发及测试人员将按年龄段、掌握的技术、从业经验和每周可用时间量等方面进行分类,并具有诸如性别、职业、学历、兴趣等在内的诸多属性。

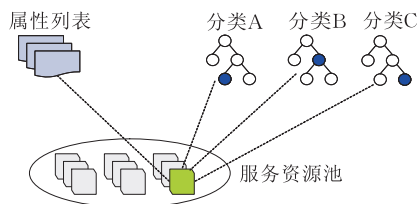


图 3 服务资源描述

多维度且多属性的服务资源可表示为四元组 $R=(RB,RU,RA,RC)$, 其中 RB 是服务资源的基本信息; RU 是与服务资源功能或者用途相关的信息; RA 是服务资源的属性集合; RC 是服务资源的分类标签集合, $|RC|=|F|$. 其中 F 是领域相关服务资源分类树构成森林, $F=\{T_1, T_2, \dots, T_m\}$. 服务资源的每个分类树可以表示为 $T=(TN, ST)$, 其中 TN 是该分类树结点的名称; ST 是该分类树的子分类树集合. 当 T 为分类树叶结点时, $|ST|=\emptyset$.

3.2 个性化需求模式及其分类

在资源整合中, 用户可以针对单个资源、多个资源以及整体 3 个层面提出需求. 如在外包服务中, “测试人员”分为“测试小组组长”和“一般测试人员”两类, 而一般测试人员又包括“资源测试人员”和“普通测试人员”两种. 针对该测试人员的分类, 任务发布者可能提出以下要求: ①测试小组的组长至少有 36 个月以上从业经验(局部约束); ②每个测试人员每周可用工作时间不得少于 20 h(全局约束); ③一般测试人员的从业经验不得少于一年; ④测试小组组长的工作量应该不少于一般测试人员的工作量; ⑤年龄大于 35 岁的测试人员至少掌握两种程序设计语言(部分约束). 只有将需求良好地表达, 才能具体且可操作地明确顾客所追求的目标. 然而, 现实的服务需求通常较为复杂, 需要以规范、可重用的方式进行描述. 本文将资源整合中的个性化需求归纳为有限、可重用、且程序易于处理的 3 类模式: 基本型、比较型和规则型. 可组合性是指一种类型的约束模式能够以其它类型的约束模式作为组成要素, 3 种约束模式之间的关系如图 4 所示.

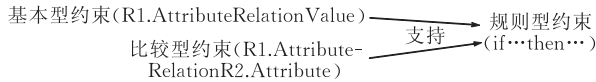


图 4 3 种需求约束模式之间的关系

下面对 3 种模式进行形式化定义, 并用以描述以上实例:

(1) 基本型约束.

最基本的约束模式, 它是一个二元组: $Basic=(Category, Constraint)$. 其中 $Category$ 是资源类别; $Constraint$ 是约束条件, 它是一个三元组: $Constraint=(Attribute, Relation, Value)$, 其中 $Attribute$ 是属性名称, $Relation$ 是算术关系, $Value$ 是属性值.

设 $Leaves$ 是 $Category$ 所包含的所有叶结点集合, 则根据约束范围的不同, 基本型约束分为 3 种

类型:

(i) 局部约束. 对于资源分类中某一叶结点所对应的类别的约束, $|Leaves|=1$. 实例①可被描述为 $Basic$ (测试小组组长, (从业时间, \geq , 36)).

(ii) 全局约束. 对于资源分类整体上的约束, 因此不需要指定其相关资源类别的集合, $|Leaves|=0$. 实例②可被描述为 $Basic$ (测试人员, (周工作量, \geq , 20)).

(iii) 部分约束. 对于资源分类中非叶结点所对应的类别的约束, $|Leaves|>1$. 实例③可被描述为 $Basic$ (一般测试人员, (从业时间, \geq , 12)).

(2) 比较型约束.

从资源分类中两个及多个类别之间关系的角度进行的约束, 它是一个三元组: $Compare=(Category, Attribute, Relation)$, 其中 $Category$ 是功能类型集合, $Attribute$ 是属性名称, $Relation$ 是算术关系. 实例④可被描述为 $Compare(\{\text{测试小组组长, 测试小组成员}\}, \text{周工作量}, >)$

(3) 规则型约束.

用来表达因果语义的约束, 它是一个二元组: $Rule=(Pre, Post)$. 其中 Pre 是前置条件, $Post$ 是后置条件. 前置条件和后置条件都是二元组: $Pre=(RB, \text{and}, RC)$, 其中 RB 是基本型约束的集合, and 是逻辑与关系, RC 是比较型约束的集合; 类似地 $Post=(OB, \text{and}, OC)$. 实例⑤可被描述为 $Rule(Pre(RB(\text{测试人员}, (\text{年龄}, \geq, 35))), \text{and}, \emptyset), Post(OB(\text{测试人员}, (\text{掌握语言数目}, \geq, 2))), \text{and}, \emptyset)$. 可以证明, 前置条件或后置条件中的逻辑或关系都可以用逻辑与关系来表达. 值得注意的是, 如果前置条件为空, 则规则型约束演变为一个或多个基本型约束和比较型约束.

需求模式有效地降低了程序处理个性化需求的逻辑复杂度, 然而很多情况下, 顾客对其各有不同的侧重, 因此本文对需求进行分类和优先级划分以反映应用中的不同侧重:

(1) 必选的约束条件.

该类约束条件的权重最高. 一旦得不到满足, 服务将被认为是无效的, 顾客满意度为零.

(2) 可选约束条件.

顾客可以灵活地设置该类约束条件的权重, 并且该类约束可以被有选择地满足而不必完全满足. 如果被满足, 顾客满意度将相应增加; 但不被满足也不会对结果的可行性造成影响.

以上两种类型的优先级可以被形式化定义为:

设 S 为由顾客需求得到的约束条件的集合, $d(c)$ 为约束条件 c 的优先级, $d(c) \in [0, 1]$. 则必选约束条件集合 $S_c = \{c | d(c) = 1\}$, 可选约束条件 $S_o = \{c | d(c) < 1\}$, 且 $S_c \cup S_o \subseteq S$.

(3) 多选多及多选一的约束条件.

设多个约束条件所构成的约束条件集合 S_p , 满足 $S_p \subseteq S$ 且 $|S_p| = m$. 如果要求这 m 个约束条件在整合结果中至少有 n 个得到满足, 并且这一要求所包括的 m 个约束条件共同拥有一个权重, 即该要求的权重, 则称该优先级为 m 选 n 的优先级, 称这 m 个约束条件为 m 选 n 的约束条件. 这并等价于 m 个约束条件中至少有 n 个属于 S_c , $\exists S' \subseteq S$, $S' \cap S_c \neq \emptyset$ 且 $S' \cap S_o \neq \emptyset$, 使得 $S' = S_p$, 且仅有多选多的层面上有意义. 特殊地, 当 $n=1$ 时, 多选多的约束条件转化为多选一的约束条件.

考虑 $S_p \subseteq S_c$ 的情况, 此时 $m=n$, 多选多的约束条件转化为多个必选约束条件; 同理 $S_p \subseteq S_o$ 的时候, $n=0$, 多选多的约束条件转化为多个可选约束条件. 对于 S_o 中的约束条件, 被满足的数目越多, 则顾客满意度越高, 对于结果的评价也越好; 而对于某个 S_p 而言, 则只有满足和不满足两种情况.

3.3 整合型虚拟服务资源的质量评价

用户需求驱动的服务整合结果的优劣由顾客满意度来衡量. 顾客满意度通常被定义为顾客在使用某产品或服务的过程中, 对产品或服务的质量、价值等不同方面的感知及与其预期相吻合的程度, 以及由此导致对该产品或服务的后续行为(比如是否愿意继续购买)等因素的综合测评值^[1]. 考虑到需求所包含的约束条件丰富和匮乏两种情况, 目标函数需要从以下两方面考虑:

(1) 结果集对于需求的满足. 对于必选约束条件, 一旦不被满足, 则该解被视为不可行解; 此外还可能有多余非必选的约束条件, 都会很大程度上影响我们对结果集合的评价. 需求的满足程度表示如下:

$$C(x) = \sum_{j=0}^{m'} \omega_j / \sum_{i=0}^m \omega_i \quad (1)$$

其中, m' 为被满足的非必选约束条件的个数, m 是所有非必选约束条件的个数, ω_i, ω_j 分别是相应约束条件的权值, 且 $\omega_i, \omega_j \in [0, 1]$.

(2) 结果集本身的质量评价, 即与需求无关, 仅与结果集本身属性有关的评价. 表示为

$$Q(x) = \sum_{i=1}^p \omega_i Q_i / \sum_{j=1}^n \omega_j Q_j \quad (2)$$

其中 Q_i 是对质量起积极作用的属性, Q_j 反之, 两者均已归一化; 所有 $Q_i, Q_j \in [0, 1]$, $\sum_{i=1}^p \omega_i = 1, \sum_{j=1}^n \omega_j = 1$.

结合以上两方面, 目标函数:

$$f(x) = \omega_c C(x) + \omega_q Q(x) + (n - n'(x)) \quad (3)$$

优化目标是使 $f(x)$ 最大. 其中 x 是一个资源选择方案; n 是必选约束条件的数目; $n'(x)$ 为未被满足的必选约束条件的数目; $\omega_c + \omega_q = 1$. $f(x)$ 前两项表示从非必选约束条件和结果集本身综合质量属性两方面得出的目标值, $n - n'(x)$ 表示可被满足的约束条件数目.

4 面向个性化需求的虚拟服务资源构造方法

顾客需求表达的个性化以及资源多分类树的存在增加了启发式资源选择的难度. 在实际应用中, 由于用户需求中的约束条件常集中与少量分类结点相关联, 因此可以修剪分类树以排除无关结点的资源, 甚至忽略与顾客需求关联较小的结点以降低复杂性. 本文在对资源分类进行预处理的基础上, 根据个性化需求特点动态剪切和调整候选资源从而实现高效整合.

4.1 资源分类预处理

4.1.1 树的剪枝

修剪树的目的在于删除约束无关分支, 以减少启发式计算的复杂度. 修剪的结果是原始树的无关结点被删除, 同时由于要保持修剪前后的逻辑等价性, 可能还会增加少量表达“逻辑非”的新结点, 使树变得更加规整. 考虑到树结构本身的递归性(即任何树都可认为是由结构更加简单的子树组成), 在剪枝过程中, 可以根据需求与分类树结点关联的具体特点, 自顶向下地逐层地检查分类树的非叶结点及其子树, 并应用一系列规则完成剪枝操作. 具体针对分类树中每层中每个未被删除的非叶结点, 比较该结点与其子树. 比较中依次应用以下规则:

① IF 分类树的所有结点都与约束无关(即针对该分类标准没有任何约束条件) THEN 剪去除根结点的所有子树, 并将分类树所有资源全部对应到根结点;

② IF 分类树的根结点与约束有关 THEN 在任一分类树根结点的约束对任何其它分类树的根结点都同样有效且不可忽略;

③ IF 约束条件中包含有当前结点的某个子结

点的“逻辑非”语义 THEN 为该结点的所有兄弟结点创建新的父结点, 并将新结点作为该结点的兄弟, 同时被标识为该结点的非, 将新结点标记为与约束有关;

④ IF 某结点与约束有关, 且以它的任何子结点为根的子树与约束无关 THEN 剪去以该结点的子结点为根的所有子树, 并将这些子树所对应的资源全部对应到该结点;

⑤ IF 某结点与约束无关, 且以它的任何子结点为根的子树都与约束有关 THEN 将该结点删除, 并将该结点的子结点直接作为其父结点的子结点;

⑥ IF 某结点与约束无关或者某结点的部分子树与约束有关 THEN 将该结点的所有与约束无关的以子结点为根的树统一替换为一个新结点, 并将其标识为所有其它子结点集合的非. 新结点对应所有被替换的子树所对应的资源.

剪枝规则的完备性体现为两个方面: (1) 上述规则所考虑的分类树和约束条件是完整的; (2) 上述规则考虑了结点-约束之间相关的所有可能情况, 包括可能存在的特殊情况. 具体地, 首先, 针对任一分类树/子树都可以迭代地表示为“非叶结点+子树”的结构. 如图 5 所示修剪前的分类树 D 可以表示为“第 1 层: 结点 $D + \{子树 D_0, 子树 D_1\}$; 第 2 层: 结点 $D_0 + \{子树 D_{00}, 子树 D_{01}\}$ 和结点 $D_1 + \{子树 D_{10}, 子树 D_{11}\}$ ”, 以此类推直到叶结点为止, 因此规则在该结构上的应用能够涵盖整个分类树以及约束相关的所有结点. 其次, 剪枝规则考虑到了每层上述结构中约束条件分布的所有可能, 包括对于分类树根结点两种特殊情况(即①根结点与约束无关和②根结点与约束有关)和一种特殊约束类型(即③约束中包含对于结点的“逻辑非”语义)的处理. 表 2 对剪枝规则的完备性进行了总结.

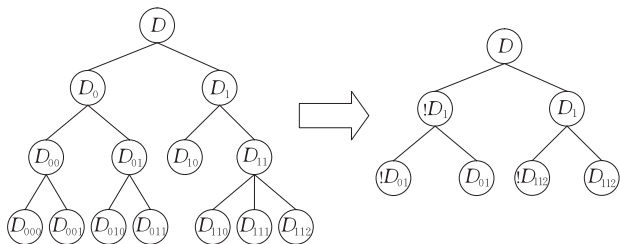


图 5 剪切前后的分类树

设分类树 D 中与约束条件相关的结点集合为 $RN = \{D, !D_1, D_{01}, D_{112}\}$, 其中 $!D_1$ 是 D_1 的补集, 则图 5 展示了应用上述剪枝规则前后的分类树. 其中 $!D_1$ 和 $!D_{01}$ 分别对应原 D_0 和 D_{00} 的资源, 而 $!D_{112}$ 则

表 2 剪枝规则的完备性

| 规则序号 | 结点与约束相关性 | 子树与约束相关性 | 是否特殊情况 |
|------|----------|----------|--------|
| ① | 无 | 无 | 是 |
| ② | 有 | 有或无 | 是 |
| ③ | 有或无 | 间接相关 | 是 |
| ④ | 有 | 全部无 | 否 |
| ⑤ | 无 | 全部有 | 否 |
| ⑥ | 有或无 | 部分有 | 否 |

包含了 D_1 中未被 D_{112} 包含的所有资源, 从而保证了剪枝的正确性.

4.1.2 主导树的选择

形成有主导树的需求描述是符合多数顾客在表达个性化需求中操作简捷性和人们思维方式的. 因此需要进行主导树的选择. 本文提出关于分类树的两条假设:

(1) 单个分类树结点数目越多, 则基于分类树可表达的顾客个性化需求的内容越多. 树的复杂性体现在结点数目、扇度、深度和结点数量. 由于树的边数比结点数少 1, 因此实际上结点数越多, 树的复杂性越高.

(2) 复杂度高的分类树在有限约束条件下产生的组合结果数目最少, 因而算法计算量小.

基于上述假设, 本文以多分类树中复杂度最高的一棵树为整合方案的启发式生成机制, 结合与其它树相关的约束进行整合结果评价, 由此提出基于动态剪枝的资源整合方法.

4.2 基于动态剪枝的资源整合算法

本文在对有序组合树算法^[3-4]进行改进的基础上, 结合多次过滤的 Top N 统计方法^[5]实现高效资源整合. 算法 1 对分类树进行预处理, 并调用算法 2 和算法 3 实现整合资源选择. 其中算法 2 是在有序组合树算法的基础上加入了对于总体需求数量的限制, 并将其约束条件的范围扩展到 3.2 节定义的约束类型而得到, 用于减小候选集合规模; 算法将传统的基于过滤的 Top N 方法的评价对象由单个资源推广到多资源组合的情况, 通过统计得到多个最优结果集.

以下对 3 个算法分别进行描述.

算法 1. 基于有序组合过滤的资源整合算法.

SCF (F, R, C)

// $F = \{T_1, T_2, \dots, T_m\}$ 是由资源分类树构成的森林, 其中 $m \geq 1$, T_1 是该分类森林的主导树; $R = \{R_1, R_2, \dots, R_n\}$ 是资源集合, 其中 n 是分类森林的所有叶结点的个数, R_i 是第 i 个叶结点对应的资源集合; C 是表达顾客需求的约束条件集合

假设: 结果集大小为 N_r , 对应于式 (3) 的权重 w_c, w_r

(满足 $w_x + w_y = 1, w_x \in [0, 1], w_y \in [0, 1]$), 随机资源组合数目与需要的最终结果数 N_r 的比值 k , 从分类叶结点中挑选出候选资源的数目占该叶结点资源总数目 N_{leaf} 的比例 α .

主要步骤:

1. 根据 C 对服务资源的多个分类树进行动态修剪和主导树选择;
2. 结合必选约束条件, 对主导树的叶结点的相关资源利用改进的有序组合树算法(算法 2)生成 $f(x)$ 由小到大排序的 $\alpha \cdot N_{leaf}$ 个可行解, 得到约简的候选集;
3. 在主导树中自底向上逐层地将候选资源集合进行组合, 得到分类树的根结点的最终候选结果集合 S ;
4. 利用多次过滤的 Top N 统计算法(算法 3)从 S 中选取 N_r 个最优方案并返回, 算法结束.

算法 2. 改进的有序组合树算法.

SCT(R, N_r, C)

// R 为候选资源集合, N_r 是期望的结果集数量, C 是表达顾客需求的必选约束条件集合

假设: 每个解中包含的资源数目的下限 N_{ln} , 每个解中包含的资源数目的上限 N_{hn} , $List$ 为按优劣顺序排序的结果集列表, 满足 $|List| \leq N_r$, 初始化 $List = \emptyset$.

主要步骤:

1. 将原问题的数学模型转换成有序组合树的标准型, 具体转换过程按如下规则进行: 如果目标函数的优化目标为极大化, 则将其取负, 使其由极大化转换为极小化; 约束条件不仅包括简单的线性约束, 而且可以包括更加复杂的约束条件.
2. 以树根作为初始解, 将目标函数值“0”标记在结点旁;
3. “次弟”的重新定义: N_{ln} 个资源的剩余组合中目标函数值最大的结点; “长子”的重新定义: 结点在 R 中剩余资源中目标函数值最大的资源. 检查 N_{ln} 是否为零, 如果不为零, 从 N_{ln} 个资源的组合中选择一个最优集合, 生成新结点, 并从组合结果中将其删除;
4. 计算它的目标值, 并标记在结点旁;
5. 在已计算目标值而未划“ \times ”勾销的解集中找出目标值最小的解;
6. 用 C 中的约束条件检验该解是否可行. 若是, 将该解加入 $List$; 否则划“ \times ”将其勾销; 如果结果列表长度达到 N_r , 返回 $List$, 算法结束.
7. 检查该结点的层数(根结点作为第 0 层)是否达到 N_{hn} , 如果达到, 则检查该结点有无“次弟”; 如未达到, 则检查该结点有无“次弟”和“长子”结点. 若有, 向正下(得到“次弟”结点)或者/和右上(得到“长子”结点)衍生一步, 到达新的结点形成新的解向量, 转到步 4; 若无, 执行下一步;
8. 检查有无已计算目标值而未划“ \times ”勾销的解. 若有, 转到步 3; 若无, 返回 $List$, 算法结束.

算法 3. 多次过滤的 Top N 统计算法.

MF-Top $N(R, N_r, k, f)$

// R 为候选资源组合的集合, N_r 为期望的结果集数量, k 为随机选择的资源组合数目与需要的最终结果数 N_r 的比值, $f(x)$ 是结果集的评价函数;

主要步骤:

1. 计算 R 中每个候选的资源组合的 $f(x)$ 值;

2. 从 R 中随机选择 $k \cdot N_r$ 个组合结果, 从中找出其中 $f(x)$ 第 N_r 大的元素. 以此元素作为阈值, 过滤掉 R 中低于该值的元素;

3. 检查 R 的基数, 如果 $|R| > k \cdot N_r$, 转到步 2; 否则转到下一步.

4. 对 R 的剩余结果, 从中选择最大 N_r 个最优结果并删除所有未被选择结果, 返回 R , 算法结束.

下面对各算法的时间复杂度进行分析:

(1) 算法 2: 检查初始候选结果 $C(|R|, N_{ln})$, 初始解中未包含全部结果集时寻找新结果集 $\frac{k(k+1)}{2} \cdot \frac{C(|R|, N_{ln}) + |R|}{2}$, 其中 $k = \max\left(\frac{N_r}{P} - C(N, N_{ln}), 0\right)$, P 为待检查解的平均可行概率. 当 $N_{ln} = 0$ 时, 算法复杂度为 $O((N_r/P)^2)$.

(2) 算法 3: 第 n 次过滤包括对 $k \cdot N_r$ 个元素查找第 N 大元素和过滤 $|R|$ 中所有元素, 分别需要比较次数 $3kN_r$ 和 $|R|/k^n$. 过滤次数满足 $|R|/k^n \leq kN_r$, 总的比较次数求和为 $3kN_r \cdot \left(\log_k \frac{|R|}{N_r} - 1\right) + \frac{k}{k-1} |R| + kN_r \cdot \log(kN_r)$. 一般 $kN_r^2 \ll |R|$, 算法复杂性为 $O\left(kN_r \cdot \log_k \frac{|R|}{N_r}\right)$.

(3) 算法 1 的复杂度主要取决于资源选择(算法 1、2)和分类树预处理的复杂性. 我们将这两阶段单独分析: 设分类树数量为 m , 树的高度为 H , 每层的期望扇度为 W , 不考虑不同剪枝规则的复杂性区别, 统一定义为 1, 则对完全扩展树进行剪枝的复杂性上限为 $6m \times \sum_{x=0}^{H-1} W^x$, 因此分类树预处理的复杂性为 $O(mW^H)$; 考虑最常见的 $N_{ln} = 0$ 且 $kN_r^2 \ll |R|$ 情况, 基于算法 1、2 的有序组合过滤资源整合算法 SCF 在预处理之后进行资源选择的理论复杂性为平方级.

5 实验与对比分析

为了验证所提方法的有效性, CFI 以基于 Struts、Spring 和 Hibernate 的 MVC 框架的小商品交易平台的开发作为案例, 由哈尔滨工业大学软件学院《软件质量保证》课程学生作为开发人员进行基于 CFI 平台的项目调度、任务分割和人员选择. 小商品交易系统旨在构建买家与卖家的线上交流平台, 而直正的交易则在线下进行. 系统支持卖家开设虚拟店铺和发布商品销售信息和买家发布商品需求信息; 同时建立留言、短消息和网络电话等交互功能, 支持卖家店

铺的托管,尤其还根据信誉对双方进行评价与分级.

5.1 实验准备

在实验中,为了在效果和性能两方面验证算法,采用两组数据集进行实验和对比分析:(1)将商品交易平台的学生作为服务资源进行整合,并与通用算法和人工选择的结果进行比较;(2)随机生成不同规模的资源分类及样本进行不同算法间的比较.实验算法由 Java 实验环境为 Eclipse helios SR2 配置为 Windows XP sp3 Intel® dual 1.73 GHz 32 位处理器,2GB 内存.

5.1.1 CFI 案例数据的准备

项目按照学生所掌握/感兴趣的技术类别、周可用时间和性别、项目经验等因素对 4 个班的 163 名学生进行分类,如图 6、图 7 所示.其中,项目经验根据学生在以往课程设计中的选题难度和成绩(优、良、中、差 4 个级别分别以 A、B、C、D 表示)确定,图中同时还标记了学生在这些分类上的分布情况(如:HTML 下方的数字“11”表示).学生在周可用工作时间和性别两个属性上的取值情况为:男 32 人,女 131 人;周可用时间小于 3h 的有 19 人,介于 3~6h 的有 99,大于 6h 的有 52 人.某项目任务分配开发人员的个性化需求用本文方法描述如下:

- Basic(技术类别,(人数,=,16))//该任务需要 16 名开发人员
- Basic(前台技术,(人数,=,4))//其中前台技术人员 5 名
- Basic(JSP,(人数,≥,3)) //至少有 3 人开发 JSP
- Basic(XML,(人数, >,0)) //一定要有人开发 XML
- Basic(JavaBean,(人数,≥,1))//至少有 1 人懂开发 JavaBean
- Basic(技术类别,(周工作时间,≥,50))
//总的周工作时间不少于 50h
- Basic(简单项目+困难项目,(人数,<,9))
//做过简单项目和困难项目的总人数一定要少于 9

```
Compare((Servlet,XML),(人数,>))
//开发 Servlet 的人数不少于开发 xml 的人数
Rule(Pre(RB(女,(人数,<,4)),and,∅),Post(OC((困难项目,项目经验 * 15%),(人数,<)),and,∅))
//如果女生人数小于 4,那么做过困难项目的人数要小于任务总人数的 15%
```

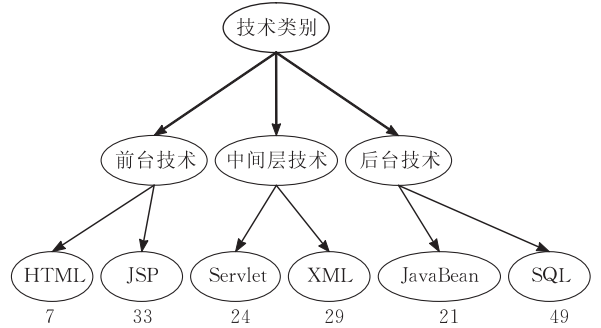


图 6 CFI 案例中的技术分类

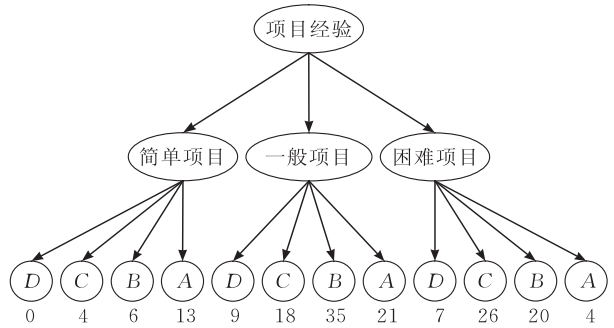


图 7 CFI 案例中的项目经验分类

5.1.2 模拟数据集的准备

我们通过程序随机生成分类数及相应属性来模拟这一资源描述.实验数据的准备过程如下:

- (1)分类树的深度 3~4,分支数目 2~3,并随机生成 4 个资源分类树,如图 8 所示.

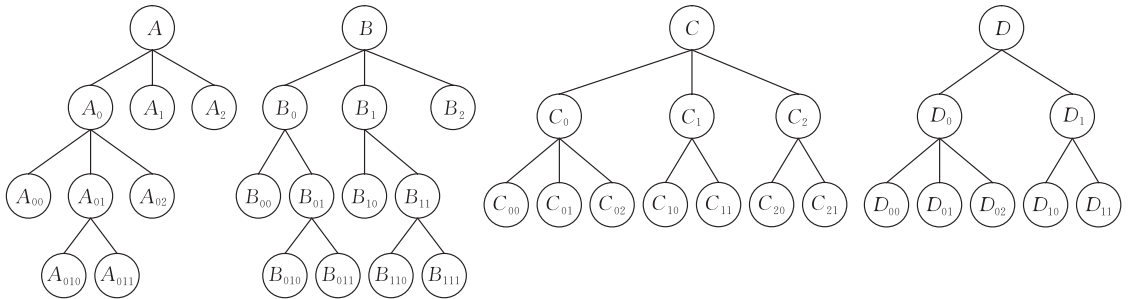


图 8 原型系统中的资源分类

- (2)分别生成规模为 200, 300, 400, 500, 1000 的带有 6 个的不同属性值(分别以 v_1, v_2, \dots, v_6 表示,其中 v_6 表示需求该类别需求的数量)的服务资源,并随机生成它们的分类,形成实验数据集.

- (3)在如表 4 所示的需求下,实验分别采用本

文方法 SCF 和遗传算法 GA 进行计算.当 GA 连续三次迭代无质量改进时停止运算,在由 GA 得到某一质量结果后运行 SCF 直到得到相近或者更优的结果,此时比较它们的时间从而间接得到它们的效果度量.

在结果集的综合质量评价中各质量属性(不包括数量属性)的权重设置为 1/5,式(3)中 $w_c = w_q = 0.5$,算法中 $\alpha = 0.7$, $N_r = 1$;GA 设置种群大小为 50,每次选择 20%的个体参与交叉操作,概率为 0.9,变异概率为 0.1.

5.2 实验结果分析

5.2.1 CFI 案例

CFI 平台商品交易平台案例的算法与其它方法的结果对比如表 3 所示,可见算法相对于通用算法和人工选择的方法在效率和结果优越性上均有较大提升.

表 3 本文算法与人工选择比较

| | 人工选择 | 通用算法 | 本文算法 |
|-------|----------|---|---|
| 结果优越性 | 0.3429 | 第 1 组: 0.7293 第 2 组: 0.8287 第 3 组: 0.5632 | 第 1 组: 0.9109 第 2 组: 0.9212 第 3 组: 0.9212 |
| 算法效率 | 约 20 min | 约 3.7 s | 约 1.8 s |

5.2.2 模拟实验

模拟实验在大规模数据的情况下比较不同算法的效果与性能.

图 9 展示了不同资源规模下两种算法达到相似优化效果所需要的计算时间.由图可见,SCF 在效率上明显优于通用的优化方法 GA. SCF 充分利用需求的个性化信息进行了分类优化和资源过滤,并通过有序组合贪婪搜索能够满足需求的方案.相比之下,GA 没有充分利用资源分类和顾客需求中的启发信息,单纯依赖进化导致在搜索空间较大时盲目性增加.GA 和 SCF 都需要在算法初期进行大量数据的初始化和迭代计算,随着资源规模的增大,有序组合树不断增长,但 SCF 的预处理和动态比例保留资源的策略使得搜索空间同时也在相应地约简,同时有序组合的贪婪方式使得算法可以按照概率的不同而在不同范围的空间内进行过滤式搜索,避免了盲目穷举.由于 GA 需要对需求中包含的多个资

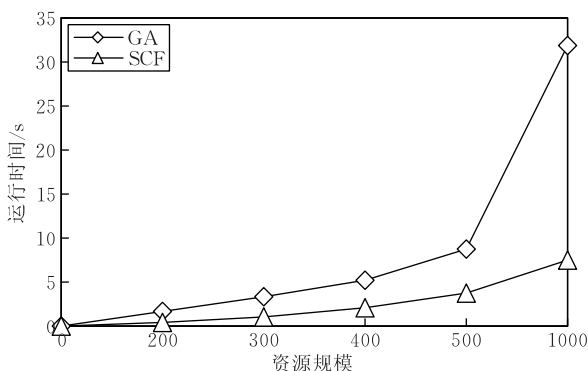


图 9 不同资源规模下算法的计算时间

源数量要求进行编码的多次计算,因此效率受到数量相关约束条件的影响.

图 10 给出了采用表 4 中不同数目的约束条件时,算法在资源规模为 300 时达到相似优化效果时所需要的计算时间.由图可见,在无约束条件下 GA 的优化速度更快,而随着用户需求条件的增多,GA 的效率几乎不变,而 SCF 的计算时间大幅减少.这与 SCF 算法中利用资源分类和用户个性化需求中的启发信息有关.通过剪枝和过滤等手段,SCF 在用户需求密集时能够有效约简候选资源,从而提高计算效率.

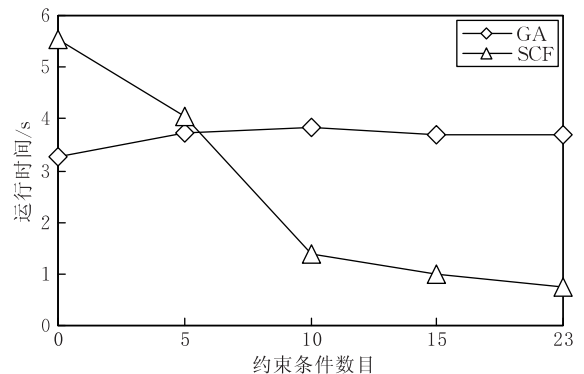


图 10 不同需求约束条件下算法的计算时间

表 4 实验中模拟的个性化整合需求

| 约束类型 | 资源类别 | 约束条件 | R_1 约束性质 | 权重 |
|-----------|--------------|---|-------------|-----|
| 全局约束 | A/B/C/D | $v1 < 240$ | 必选 | 1 |
| | | $v2 < 20$ | 可选 | 0.8 |
| | | $number = 10$ | 必选 | 1 |
| | | $v3 > 2$ | 必选 | 1 |
| 部分约束 | A0 | $v4 \geq 80\%$ | 可选 | 0.8 |
| | B0 | $number > 2$ | 可选 | 0.4 |
| | D1 | $number = 3$ | 必选 | 1 |
| | 限制型约束 | C1 | $v5 \neq 8$ | 可选 |
| $v2 < 15$ | | | 可选 | 0.4 |
| A2 | | $v1 > 20$ | 可选 | 0.6 |
| | | $v4 > 90\%$ | 可选 | 0.8 |
| 局部约束 | A1 | $number = 2$ | 必选 | 1 |
| | | $v1 \leq 100$ | 必选 | 1 |
| | A02 | $v5 = 10$ | 必选 | 1 |
| | B2 | $number = 4$ | 必选 | 1 |
| 比较型约束 | D0, D1 | $v2 < 10$ | 可选 | 0.6 |
| | | $number > 1$ | 可选 | 0.8 |
| | B01, B10 | $v4 > 90\%$ | 可选 | 0.8 |
| | | $number = 2$ | 必选 | 1 |
| 规则型约束 | A01, A0 | $v1 \leq 100$ | 必选 | 1 |
| | | $v5 = 10$ | 必选 | 1 |
| | C2 | $number = 4$ | 必选 | 1 |
| | | $v2 < 10$ | 可选 | 0.6 |
| D0, D1 | $v3 \geq v1$ | 必选 | 1 | |
| | $v5 = v3$ | 可选 | 0.4 | |
| 规则型约束 | A01, A0 | if $A01.number > 0$ then $A01.number = A0.number$ | 必选 | 1 |
| | | if $C2.v1 > 20$ then $(C2.v4 > 3 \text{ and } C2.v5 > 0.9)$ | 可选 | 0.8 |
| 规则型约束 | D0, D1 | if $D0.number > D1.number$ then $D0.v1 > D1.v1$ | 必选 | 1 |

图 11 比较了不同资源规模下它们在相同时间内的优化效果。可见, 两种算法在单位时间内获得结果的最优性均随着资源规模的增大而降低。考虑极端情况没有任何顾客需求, 结果的最优性完全取决于资源质量, 此时最优结果集由每个分类的最优解组合而成, 故在个性化需求约束下, 有序组合有很大概率的贪婪选择性, 使得优化效率较高。

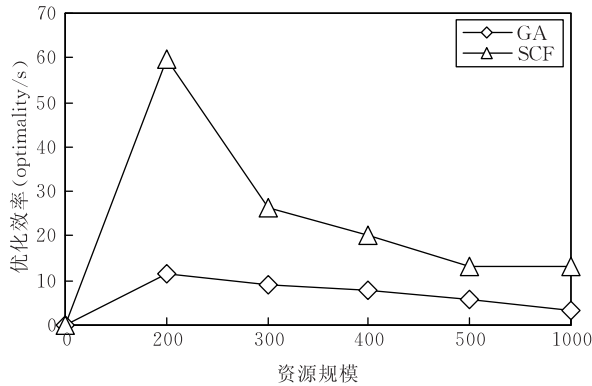


图 11 不同资源规模时算法的优化效果/时间比值

6 相关工作

目前对于资源整合和虚拟化的相关研究主要集中在网格资源调度、计算机集成制造资源的调度^[6]以及个性化的信息检索 3 个领域。网格资源选择通常是为了选择一个最优的资源组合, 以最小的代价使用户提交的作业获得满意的性能, 文献[7]采用遗传算法与禁忌搜索相结合的方法来实现该目标, 而文献[8]提出一种分布式代理体系结构使得多个处理命令在多个服务器结点间分布并通过灵活的命令的映射使得一个购买指令可以对应多个出售指令; CIMS 中研究如何分配适当的制造资源来满足用户需求并且不造成浪费, 以及如何根据计划合理选择资源, 以优化或者满足工期等^[6], 文献[9]解决制造网格中客户需求和制造资源可用性动态变化情况下的资源调度问题, 通过定义资源-任务匹配矩阵利用成组决策分析层次流程模型选择最合适的资源给每个任务以达到最好的 QoS 性能; 个性化信息检索^[10]多从顾客的个人兴趣范围、历史记录或者行为规律出发, 挖掘和预测顾客可能感兴趣的结果, 不仅需要大量历史数据, 对于新资源的推荐难度较大, 而且是静态的, 不能满足实时的需求; 文献[7]通过用户搜索记录和类别层次分别得到个性化和通用的记录, 两者结合来将用户查询映射到能够表达用户关注点的类别中, 从而通过个性化实现了检索的有效

性。然而由于应用背景等方面的原因, 以上研究都没有针对资源分类和需求个性化方面给出专门的解决方案。

本文在考虑顾客个性化需求与资源组织特点的基础上, 充分利用启发信息高效给出资源整合的方案。整合结果将被虚拟化为大粒度资源以匹配相应的服务任务。

7 结束语

在现代服务生态系统中, 聚集多方资源形成虚拟资源以满足顾客及应用需求成为必然的趋势。如何选取这些资源形成较为优化的虚拟化方案成为其中一个关键问题。本文解决服务生态系统背景下虚拟服务资源的个性化整合。基于资源的多分类特征, 通过对个性化需求的模式化描述, 使得应用需求能够得到简捷及个性化的表达; 资源整合算法充分考虑到资源分类的启发信息动态约简候选资源, 保证了高效准确的整合选择。

除了基于社区的软件工程服务外包外, 该方法还可广泛应用于各类电子商务场景中。如制造领域中, 生产某些产品需要采购多种原材料(包括零件和配件), 出于技术和业务两方面的考虑, 采购部门对于不同生产用途的原材料采购有着千差万别的要求。不同产品可能需要采用不同质量的同种零件, 而有些企业可能只愿意为产品的主要配件支付更高的成本, 等等。在海运物流服务中, 以货代为中心的物流服务平台需要能够规划一组车辆用来将指定的一批货物由货主处运往场站存储, 以及由场站运往码头。不同货物对于车辆的要求有所不同。例如, 肉类产品需要安排具有冷冻设施的车辆, 对于易碎的产品需要安排具有特殊减振装置的车辆。此外为了节省车辆的运输空间, 同一车厢可能装有不同种货物。这时车辆的调配还应该考虑某些货物不能放在同一车厢, 等等。在采用资源整合方法的网上订餐服务中, 顾客不需要再明确指出自己需要的每一种菜品, 而只是给出不同类别的菜的数量、口味要求、价格要求等, 就能由系统参考科学的食品搭配自动推荐几组菜品供顾客选择。

本文根据个性化的应用需求和资源的质量参数选择与整合资源, 并未考虑虚拟资源的粒度。事实上针对一个特定的应用, 可将需求分解为一组适宜的任务分别由相应的虚拟资源来完成。同时, 结合不同服务资源在使用中的关联情况, 探讨根据当前可用

资源状况和历史关联情况进行高质量的任务分解也是一个有价值的研究方向。

参 考 文 献

- [1] Liu Shu, Liu Ying, Jiang Hui-Min et al. Service quality evaluation method for community-based software outsourcing process. *International Journal of Service Science, Management, Engineering and Technology (IJSSMET)*, 2011, 2(1): 1-14
- [2] Zhou Qing-Yuan. Customer satisfaction degree evaluation based on rough set theory and matter elements model//*Proceedings of the International Conference on Industrial and Information Systems (IIS2009)*. Haikou, China, 2009: 224-227
- [3] Qian Yun-Tao, Ching Y Suen, Tang Yuan-Yan. Sequential combination methods for data clustering analysis. *Journal of Computer Science and Technology*, 2002, 17(2): 118-128
- [4] Liu Yong, Hu Yun-Fa. Combination tree for mining frequent patterns based on inverted list//*Proceedings of the 2006 International Conference on Computational Intelligence and Security (CIS2006)*. Guangzhou, China, 2006: 805-808
- [5] Zhang Jun, Yang Jia-Hai, Wang Ji-Long. TopN sort algorithm based on multiple filtering. *Journal of Tsinghua University (Science&Technology)*, 2006, 46(4): 604-608(in Chinese)

- (张军, 杨家海, 王继龙. 基于多次过滤的 TopN 统计算法. *清华大学学报(自然科学版)*, 2006, 46(4): 604-608)
- [6] Yuan Yi-Ping, Yu Tao, Xiong Feng et al. QoS-based dynamic scheduling for manufacturing grid workflow//*Proceedings of the 9th International Conference on Computer Supported Cooperative Work in Design*. Coventry, UK, 2005: 1123-1128
 - [7] Chu Rui, Lu Xi-Cheng, Xiao Nong. A clustering based resource aggregation method for virtual computing environment. *Journal of Software*, 2007, 18(8): 1858-1869(in Chinese) (褚瑞, 卢锡城, 肖依. 一种基于聚类的虚拟计算环境资源聚合方法. *软件学报*, 2007, 18(8): 1858-1869)
 - [8] Tamai M, Shibata N, Yasumoto K et al. Distributed Market Broker architecture for resource aggregation in grid computing environments//*Proceedings of the 2005 IEEE International Symposium on Cluster Computing and the Grid (CC-Grid2005)*. Cardiff, UK, 2005: 534-541
 - [9] Liu Li-Lan, Yu Tao, HeYuan. A TQCS-based scheduling approach for manufacturing grid. *Journal of Donghua University*, 2004(6): 43-48
 - [10] Zeng Chun, Xing Chun-Xiao, Zhou Li-Zhu. A survey of personalization technology. *Journal of Software*, 2002, 13(10): 1952-1961(in Chinese) (曾春, 邢春晓, 周立柱. 个性化服务技术综述. *软件学报*, 2002, 13(10): 1952-1961)



CHU Dian-Hui, born in 1970, Ph. D. candidate, associate professor. His research interests include service computing, service engineering, and software architecture.

WANG Xian-Zhi, born in 1984, Ph. D. candidate. His current research interests include service composition and

service resource integration

WANG Zhong-Jie, born in 1978, Ph. D., associate professor. His main research interests include service engineering, service-oriented architecture, service composition and software engineering.

XU Xiao-Fei, born in 1962, professor, Ph. D. supervisor. His main research interests include service science and engineering, enterprise computing and service computing.

Background

This work is supported by the National Natural Science Foundation of China (Nos. 61033005, 70971029, 60803091) and Key Technologies R&D Program of Shandong Province (Nos. 2010GZX20126, 2010GGX10116)

These projects are conducted under the joint content of modern service industry, service computing and value-oriented service engineering, around the key concept of BIRIS. BIRIS is a bilateral customer value oriented resource integration service mode proposed by Research Center of Intelligence Computing for Enterprises and Services (ICES), Harbin Institute of Technology based on the analysis of industrial mash-up practice. By introducing a third-party service platform/service agency, the matching between service require-

ments and service capabilities is performed. To secure the sustainability of services, the realization of service value of both service providers and service consumers becomes necessary. Now that traditional approaches fail/neglect to cope with circumstances of enterprise interoperability and cooperation, our work presents a feasible mechanism and optimized solution for utilizing and composing distributed services, including both software services and services in the real world aiming at delivering higher satisfaction and business flexibility. The research results can be applied to various applications with the domain of e-commerce, manufacture and healthcare services.