

需求语义驱动的按需服务聚合生产方法

文 斌 何克清 梁 鹏 王 健

(武汉大学软件工程国家重点实验室 武汉 430072)

摘 要 按需服务资源聚合、软件按需演变生长,赋予了需求工程在服务计算时代新的机遇.文中针对面向服务的网络化软件开发强调发现和重用已有的服务资源,通过聚合服务资源来满足用户需求的软件生产技术形态,构筑语义需求工程以应用于网络化软件的生产.基于语义的服务需求工程的关注点是识别服务和过程,采取涉众群体智能参与的服务需求语义获取和需求语义驱动的互操作性聚类来实施按需服务生产的方法,以服务的需求语义完整性描述为支点,系统地研究和提出三大技术:即面向大众本体、不断演化、即时验证的需求语义获取技术,需求语义驱动的服务软件聚合方法,基于需求语义的服务定制.针对需求语义驱动的服务聚合生产,充分利用网上涉众群体智能和需求专家相结合,采用语义 wikis 以捕获和验证网络化软件需求语义,通过需求语义互操作展开选择连接本体作为语义信息载体,进行服务资源聚合,生成网络化软件.理论与实证研究表明,基于语义的服务需求工程对云服务环境下网络化软件的按需交付具有明显的实用性和有效性.

关键词 服务需求;需求工程;需求语义;语义 wikis;聚类

中图法分类号 TP313 **DOI号**: 10.3724/SP.J.1016.2010.02163

Requirements Semantics-Driven Aggregated Production for On-Demand Service

WEN Bin HE Ke-Qing LIANG Peng WANG Jian

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

Abstract Based on the existing technology for assembling service, related approaches including requirements semantic encapsulation, requirements semantic interoperability extending and requirements semantics-driven service customized manufacture are put forward. Large-scale and complex system exhibits adaptive feature, and evolutionary emergence of collective behaviors is its fundamental phenomena. This paper adopts stakeholders-driven requirements semantics acquiring technique for software services and combines with semantic wikis for supporting evolution and semantic annotating of requirements. The approach not only elicits the conventional documentary requirements through collaboration and interactive negotiation mechanism, but also can it process intelligent retrieve, requirements consistency check and reasoning for services requirements entity depending on requirements semantics and requirements element instantiation conducted by underlying requirements ontology of services. By choosing connecting ontologies as semantic carrier for service aggregation, software production will be focused on upper level semantic description rather than concrete service. Theoretical and empirical studying has proven the validity practicability of the proposed method.

Keywords services requirements; requirements engineering; requirements semantics; semantics wikis; aggregation

收稿日期:2010-06-08;最终修改稿收到日期:2010-08-30.本课题得到国家“九七三”重点基础研究发展规划项目基金(2007CB310801)、国家自然科学基金(60970017,60950110352)、教育部博士点基金新教师项目(20090141120020)和中央高校基本科研业务费专项资金(武汉大学)(6082008)资助.文 斌,1970年生,博士研究生,副教授,从事基于语义的服务需求工程研究. E-mail: binwenwebb@gmail.com.何克清,1947年生,博士,教授,博士生导师,主要从事需求工程、服务计算研究.梁 鹏,1978年生,博士,副教授,博士生导师,主要从事需求工程、软件体系结构知识研究.王 健,1980年生,博士,讲师,主要从事面向服务的需求工程、服务语义研究.

1 引 言

开发高质量、符合用户需求的软件主要有形式化和工程化两种方法. 软件工程即是工程化、系统化开发成功软件的工程学科, 自创立以来在软件生命周期各个阶段都取得了长足进步, 比较而言需求在软件工程各分支中相对滞后, 有待进一步发展和完善.

需求工程是软件工程项目成败的关键, 尤其是对于以分布的服务计算资源聚合为特征的网络化软件^[1], 需求工程有其特殊的重要性. 构件、遗留系统的服务封装, 使得大部分的软件设计、实现、测试任务都由服务提供商预先分工协作完成, 只要能够动态获得用户需求, 进行随需而变、协同应变的适应性需求建模, 在面向领域规模化定制、语义 SOA、服务自组织及其互操作性聚类、动态服务组合与优化等技术的支撑下, 就能够实现网络化软件“整体”生成、生长与协同工作, 高质量地满足用户服务需求, 因此面向服务的软件工程正在向需求工程聚焦. 需求工程涉及的问题包括用户需求的动态获取与分析、演化生长建模、需求管理与需求模型验证等. 需求建模方法针对软件范型不同主要分为结构化需求建模和面向对象需求建模, 涉及功能需求分析和非功能需求分析.

目前国际上比较有代表性的软件需求工程方法包括面向目标的方法、面向主体和意图的方法、基于情景的方法、问题框架方法、基于领域建模的预需求分析、文档驱动的方法和面向方面的方法等^[2].

通过对各种需求工程方法的研究发现, 对网络化软件需求工程影响较大的主要是面向目标的方法和基于领域本体的需求工程方法^[3]. 其中面向目标的方法侧重于对早期需求进行分析和建模, 试图帮助开发者理解领域中不同角色的动机和期望, 可对功能和非功能需求目标进行识别分析. 需求工程目前正在经历着从面向对象到面向目标的转换^[4-5], 面向目标的需求分析成为需求工程研究的热点. 基于领域建模的预需求分析^[3]实际上是基于领域本体知识的需求分析过程, 本体是概念模型的明确的规范说明. 利用本体对领域中的公共概念以及隐含的知识进行明确的规范说明, 可以在不同的利益相关方之间达成一致的术语约定, 避免二义性. 基于领域本体的方法存在的一个问题是强调获取领域概念性知识, 而忽略了关注任务和功能性的建模.

上述的这些需求建模方法主要都是针对面向对象对象的软件开发时代提出的, 能够有效地解决它们所面对的问题. 在面向服务的网络化软件逐步成为软件开发的新计算范型之时, 需要重新审视这些方法在面向服务的计算中的适用性. 由于面向服务的网络化软件开发与面向对象的软件开发之间存在较大的差别, 特别是面向服务的网络化软件开发强调发现和重用已有的服务资源, 其需求关注点是识别服务和过程(工作流), 建立用户目标到服务、过程的关系, 通过静态或动态聚集服务资源来满足用户需求. 这也为面向服务的需求建模方法提出了新的要求, 而现有的这些需求建模方法在对服务计算方面还缺乏足够的支持.

我们已经系统研究了面向特定领域的小型即时性服务组合软件的需求获取、验证、组合呈现^[6], 此类软件的需求可网上即时获取、软件功能一次性生成, 使用后软件形态随即释放.

除了小型即时性软件, 网上涉众用户即利益相关者也对面向领域的大型网络化软件有迫切的要求, 此类软件的特点是: 软件功能庞大、需求复杂多样; 关心软件建设的不仅仅是用户, 更多的是利益相关者, 我们统称涉众用户(Stakeholders); 软件非一次性使用, 而是长期应用、不断演化生成; 不仅仅使用服务资源, 网上一切软件资源都可以为我所用.

大型网络化软件系统面对的是涉众用户丰富、需求多变的复杂系统, 通过采用复杂性科学的系统思维方式, 把信息技术和信息系统的研究纳入复杂性研究的系统结构中去. 对大型网络化软件系统需求产生的机制进行分析表明, 其需求呈现出的自适应性、群体演化特性, 需求获取的整体涌现现象是传统还原论无法解释的, 目前大量 Web 2.0 的应用已经充分证明了这一点^[7]. 对此种复杂系统及其特性, 传统需求获取技术无法适应, 如何寻求新型需求获取技术来应对大型网络化软件开发的复杂性态势, 如何在软件需求中附加语义信息以利于软件需求的后续自动化随需而动的服务软件即时发布, 如何根据获取的需求语义聚类分布异域服务资源, 如何在需求语义引导下进行服务的按需定制, 这正是本文的研究动机和关注点. 我们在国家“九七三”项目资助下, 探索了相关问题, 研究表明: 充分利用涉众用户参与和大众智慧, 采用社会性软件的协同交互平台如语义 wikis、RSS/Atom 等, 利用看似简单的标注功能加演化机制, 可有效涌现软件需求描述与需求语义, 同时其获取过程是连续的, 符合网络化软

件随需而变、不断生长的要求. 本文主要研究需求语义获取与互操作展开以用于服务资源聚合生产, 有关服务定制的内容将另文讨论.

服务组合主要研究如何在服务流程基本固定情况下, 进行异域、动态服务的选择、匹配、连接; 而服务聚合的研究同时涉及变化的流程和变化的服务两个动态部分, 流程因需求驱动而变, 涉众需求用需求语义封装, 因此服务聚合生产顺利进行至少包括两个部分: 需求语义获取部分和需求语义互操作展开(基于语义的服务组合)部分. 本文需求语义获取部分采用涉众群体智能参与、结合语义 wikis 以支持需求的演化生长与语义标注, 不仅能通过涉众的群体交互协商捕获传统文档性需求, 而且依靠需求语义呈现和需求本体诱导的需求元素的需求语义实例化进行软件需求实体的智能检索、需求一致性验证与推理; 通过获取的上层需求语义, 服务组合部分选择基于知识(语义)的连接本体作为语义信息载体进行底层服务资源聚类匹配生成网络化软件. 需求语义来自涉众, 服务聚合生成的软件由获取的需求语义驱动, 即按需生产.

本文第 2 节提出基于语义的服务需求工程, 阐述语义 Wikis 的特点及对需求语义获取的适用性; 第 3 节给出面向服务的需求语义本体和获取平台设计; 第 4 节在需求语义基础上研究服务资源的语义聚类和软件生产; 第 5 节的讨论围绕服务需求语义获取与展开的实证、实验分析; 接着是本文工作与国际上相关工作的比较; 最后总结全文并提出进一步的研究规划.

2 基于语义的服务需求工程与语义 Wikis

Ward Cunningham 用 Wiki 命名了以知识库文档为中心, 以共同创作为手段, 靠众人不停地更新修改这样一种借助互联网创建、积累、完善和分享知识的全新模式^[8]. Wiki 的原理在于开放编辑和自由协作, 可应用于百科全书构建、软件开发、项目管理等协作创作领域, 最成功的应用范例是 Wikipedia. 但传统 Wiki 系统的也有诸多问题^[9], 如: (1) 无结构数据. 缺乏数据模型导致应用很难使用其数据; (2) 知识难以重用. 知识仅局限于页面和页面之间的链接, 无法从页面中有效抽取知识; (3) 受限的检索功能. 知识检索仅针对有限的页面主题, 缺乏智能.

以上问题可归结为知识内容缺失语义, 语义指的是概念实体及其之间关系的显式表示. 语义网(Semantic Web)是用一种易于被机器处理的表示方法来描述网上内容, 并采用智能技术来利用这种表示方法所提供的便利^[10]. 为了实现语义网, 语义依靠元数据中的本体标注技术增加到目前的文档 Web 中. 通过语义, 不同数据实体可实现自动彼此关联, 由此衍生出众多的智能和适应变化的应用. 数据实体不仅是单个应用或网站, 可以是整个互联网, 以充分开发万维网的全部潜力. 语义网是一个不断推进的过程, 目前主要的进展是 Linked Open Data 项目^[11].

语义网技术可应用于传统 Wiki, 通过添加语义使 Wiki 内容对于机器可访问. 语义 wikis 即是 Wiki 的协作交互特性融合语义网技术的产物^[12]. 语义标注增加到 Wiki 页面并存储为知识库, 标注也可以与背景本体(如领域本体)相对应, 由此使 Wiki 的内在结构便于机器处理. 通过 Wiki 中的语义添加, 语义 wikis 比普通 Wiki 更加成熟, Wiki 页面中的数据能高效获取、标识, 并能与其它页面的数据相关联, 显著改善知识内容的浏览、搜索、检索和呈现; 同时另一个关注点是对互操作性的支持, 语义 wikis 促进了 Wiki 内容之间及与外部软件服务的互操作能力^[13].

语义 wikis 的公共特性分为自适应内容呈现(SW1)、增强的搜索能力(SW2)、增强的浏览能力(SW3)、推理支持(SW4)、标注的类型化(SW5)、(半)自动元数据抽取(SW6)、本体导入/导出(SW7)和底层模型可变(SW8).

Wiki 提供的协作交互机制已应用于软件开发生命周期^[8], 本文关注的是语义 wikis 在软件语义需求工程中的应用. 需求工程是软件工程的分支, 其关心软件系统的真实目标、功能和约束, 需求工程的核心活动如下: 需求获取、需求建模与分析、需求通信、需求协商、需求演化. 基于语义的服务需求工程(SRE)是传统需求工程的深化, 通过添加语义描述需求要素间的关系, 使静态的需求要素相互关联, 便于后续的按需服务软件自动生成. 需求语义贯穿于整个服务软件的生产(见图 1), 包括需求语义获取、需求语义互操作展开与服务聚合、基于需求语义的服务定制等内容. 语义需求工程结合需求工程与语义网技术, 以需求的语义描述为基点, 在动态演变的需求语义上生成即时软件业务过程语义(见第 4 节的 RSO), 通过语义互操作展开聚合已有服务资产和异域服务资源. 无法得到的服务也可主动

发布其需求语义引导服务提供商按要求生产,从而真正体现按需服务、即时演化的服务时代理念.其中需求语义获取是整个需求语义工程的核心活动.

语义 wikis 非常符合需求语义工程的要求,特别是对于需求语义初始获取阶段.文献[9]列出了被语义 wikis/普通 Wiki 支持的软件需求过程活动.

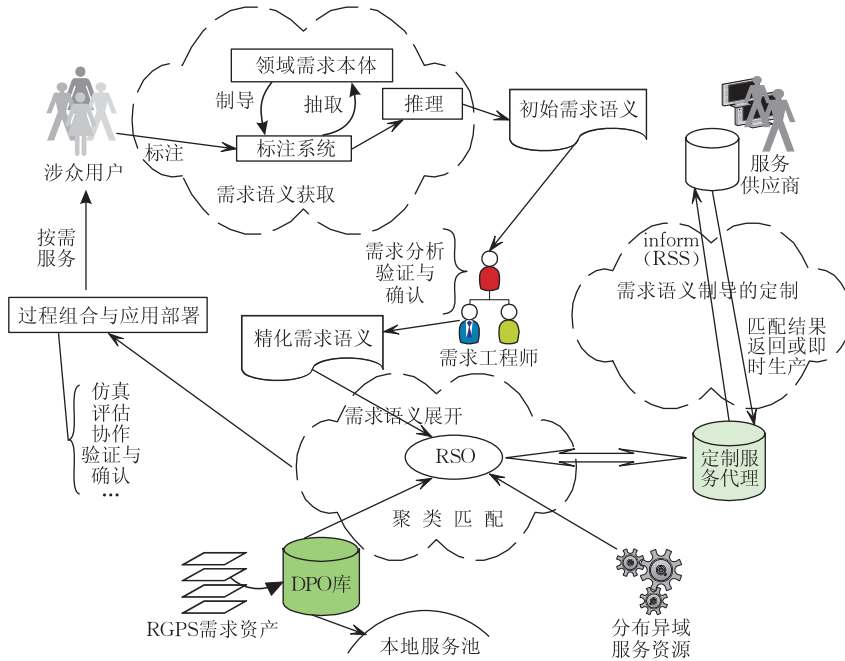


图 1 基于语义的服务需求工程

对于高不确定性项目,用户参与越多越有助于改善最终需要产品的质量^[8].因此,大量的、多样化涉众用户交互协作将有效提升软件需求质量.然而,涉众用户具有不同的背景、目的、能力、关注和看法.利用语义 wikis 本身的协作交互机制,即使非技术背景涉众用户也能方便地参与合作.在协作机制支持下,需求语义关系对需求冲突或依赖关系建模,同时语义 wikis 也支持现有需求管理工具提供的功能,例如需求的版本化和基线管理.版本化由叫作不变连接(Permalink)的特性提供,不变连接从连接关系收集器中生成,其提供了某个页面所有连接的一个表.驻留于一个页面的这些连接表示一个发布的需求版本并已被锁定(冷冻)^[8].再者,由于添加了语义信息,语义 wikis 能与集成开发环境 (IDE) 互操作,比如需求可从语义 wikis 中导出到 Eclipse IDE,这样来自需求语义描述的组织业务过程和系统制品呈现通过 MDA 方式部分实现了自动化^[14].

目前语义 wikis 引擎各有特色,适合涉众需求语义获取项目使用的引擎应具备以下要求:语义 wikis 核心引擎仍在不断开发中,便于低层核心的采用和技术支持、交流;在核心之上,开放并提供方便二次开发以增加需求语义获取、实例标注等社会化元件和辅助插件的功能.为此,我们主要关注以下

语义 wikis 引擎: semantic mediawiki^①, ontowiki^②, KiWi^③, SweetWiki^④.

3 涉众群体智能参与的服务需求语义获取

通过前面的分析可知,需求语义贯穿整个服务软件的全生命周期,实现需求由无序到有序,由非形式化到形式化的表示,同时需求语义的获取也是一个不断演化的过程.采用涉众参与驱动的需求语义获取的理想策略应是首先由领域/需求专家设计服务软件描述概念模型(本体),然后通过群体智慧进行自由语义标识从中根据不同的基线抽取大众需求语义进而对专家的需求模型进行修正,以形成结合专家和大众共同智慧的面向服务的需求语义本体.接着,在此需求语义本体制导下,涉众和需求工程师进行需求语义的实例化标注,生成需求语义制品.

3.1 面向服务的需求语义本体

目前,在对面向服务的多样化和个性化需求的研究过程中,在各种面向对象的软件需求分析方法

① <http://semantic-mediawiki.org>

② <http://ontowiki.net>

③ <http://www.kiwi-project.eu>

④ <http://argentera.inria.fr:8080/wiki>

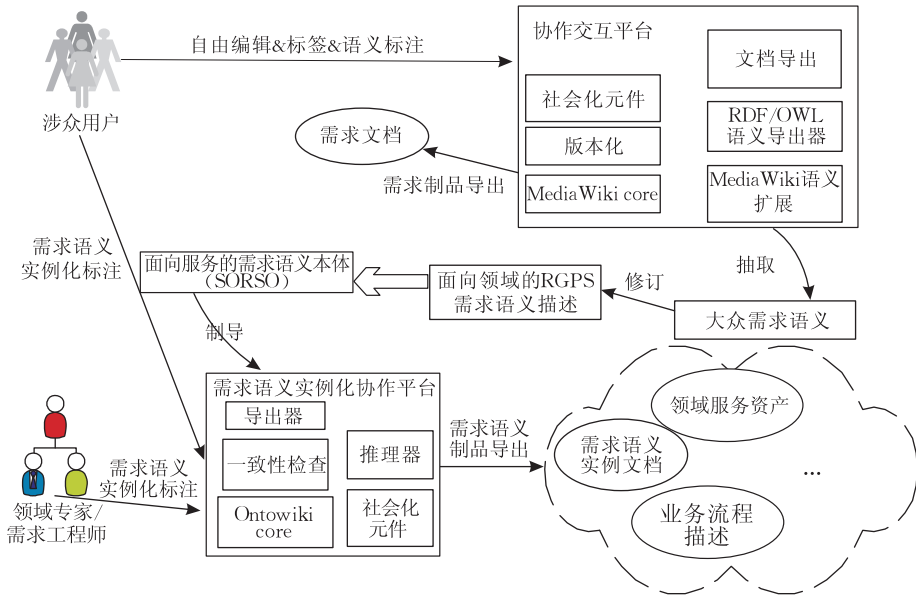


图 4 涉众参与的服务需求语义获取

4 服务需求语义互操作展开

通过上一节的需求语义获取平台,可得到涉众用户演化生长而成的需求语义实例化描述.一致的、涉众认可的需求语义为跨域服务资源的聚合和按需服务响应提供语义互操作基础.

下面先严格给出服务需求语义互操作展开的信息载体-连接本体的相关概念,然后在此基础上讨论语义互操作性度量和需求语义驱动的服务资源聚合生产的算法设计.

4.1 需求语义互操作载体-连接本体

定义 1(基于描述逻辑的领域本体). 领域本体表示为 $DO = \langle D, C, T, A, LH \rangle$, 其中 D 表示领域, C 表示本体的概念集合, T 表示本体的 $TBox$, A 表示本体的 $ABox$, LH 表示本体概念的格层次结构.

定义 2(概念间关系元组). 给定领域本体 $DO = \langle D, C, T, A, LH \rangle, c_p, c_q \in C$, 关系 $r(c_p, c_q) \in T$, 如果 c_p 和 c_q 满足: (1) $c_p \sqsubseteq c_q$; (2) $c_p \sqsubseteq \exists r.c_q$; (3) $c_p \sqsubseteq \forall r.c_q$, \sqsubseteq 表示概念间的包含关系, 则称 c_p 与 c_q 间存在相关关系, 记为 $\chi = \langle c_p, r, c_q \rangle$.

定义 3(语义关联). 给定两个相关关系 $\chi_1 = \langle c_{p1}, r_1, c_{q1} \rangle, \chi_2 = \langle c_{p2}, r_2, c_{q2} \rangle$, 如果满足 $\chi_1.c_{q1} = \chi_2.c_{p2}$, 则称 χ_1 和 χ_2 之间存在语义关联, 记为 $\chi_1.c_{q1} = \chi_2.c_{p2}$.

定义 4(语义关联路径). 给定相关关系集合 $X = \{\chi_i | \chi_i = \langle c_{pi}, r_i, c_{qi} \rangle, i = 1, 2, \dots, n\}$ 以及相关关

系 $\chi_s = \langle c_{ps}, r_s, c_{qs} \rangle, \chi_d = \langle c_{pd}, r_d, c_{qd} \rangle, \chi_s, \chi_d \in X$, 如果 $\exists m \leq n - 2$, 对于 $\forall \chi_{i+j} \in X, j = 1, 2, \dots, m$, 有 $\chi_s \odot \chi_{i+1}, \chi_{i+1} \odot \chi_{i+2}$ 和 $\chi_{i+m} \odot \chi_d$ 成立, 则称 X 中存在一条从 χ_s 到 χ_d 的语义关联路径, 即概念 c_{qs} 和 c_{qd} 存在语义关联路径.

定义 5(概念语义深度). 本体中概念的含意 (meaning) 不仅仅由表示这个概念的类 (class) 所描述, 更与和这个类相关联的其它类相关, 可称它为深度语义. 为了计算语义深度, 取本体根概念 (类似 OWL 的 "Thing") 语义深度为 0, 设概念 c 的语义深度 $Depth(c)$ 为 I , 则 c 的父概念 (如存在的话) 的语义深度为 $I - 1$, C 的子概念 (如存在的话) 语义深度为 $I + 1$.

定义 6(子本体). 给定领域本体 $DO = \langle D, C, T, A, LH \rangle$, 其子本体为 $\langle C_{sub}, T_{sub}, A_{sub}, LH_{sub}, I \rangle$, 其中 C_{sub} 为 sub_{Onto} 的概念集合, 表示子本体所对应的上下文, $|C_{sub}| < |C|$ 且 C_{sub} 中概念必须存在语义关联或语义关联路径, T_{sub}, A_{sub} 为 sub_{Onto} 的局部知识基且 $T_{sub} \subseteq T, A_{sub} \subseteq A, LH_{sub}$ 为 sub_{Onto} 的概念格层次结构, I 为指向 DO 的引用, 等于 DO 的子本体或无确定领域时其 I 值为 nil.

定义 7(同源子本体空间). 给定领域本体 $DO = \langle D, C, T, A, LH \rangle$, 其同源子本体空间 $Space = \{\langle sub-Onto_b, B, DO | \forall sub-Onto_b. I = DO, B \in \text{引用索引基} \rangle\}$.

定义 8(概念的最大完备子本体). 给定一个概念 C , 对 C 进行子本体生成运算, 当遍历深度增加时子本体 sub_{Onto} 所属概念集不再增加时, 停止生成

过程,称最终生成的子本体 sub_{Onto} 为概念 C 的最大完备子本体。

定义 9(领域需求本体). 为便利需求获取和匹配,领域需求本体是一类特殊的领域本体,其概念中语义深度为 1 的概念只有两个:需求动词概念 Operation、需求名词概念 Entity;概念 Operation 的最大完备子本体称为领域需求本体的操作本体,其中的概念都是领域本体中动词;概念 Entity 的最大完备子本体称为领域需求本体的实体本体,其中的概念都是领域本体中名词。

定义 10(领域问题本体). 领域问题本体 $DPO = Merge(\bigcup_{asset_i \in RGPS} (Extracting(P, Dep, asset_i), index_i)), P$ 表示问题概念集, Dep 表示遍历深度, $RGPS$ 表示领域定制资产, $index_i$ 为匹配到 $RGPS$ 的问题概念的源本体引用。

定义 11(需求符号本体). 需求符号本体 $RSO = \langle ReqSemantics, Concept, Control \rangle$, $ReqSemantics$ 为获取的需求语义, C 为从 $ReqSemantics$ 中抽取的概念集, $Concept \supseteq DPO.C$, $Control$ 为匹配的服务本体之间的控制结构,主要有顺序、选择、分支-合并、任意、次序、循环。

定义 12(连接本体). 连接本体 $CO = \langle RSO, DPO, Mapping-Onto-Set \rangle$, 其中 RSO 为需求符号本体, DPO 为基于领域需求资产的面向问题的领域问题本体, $Mapping-Onto-Set$ 为匹配的异源子本体集合。

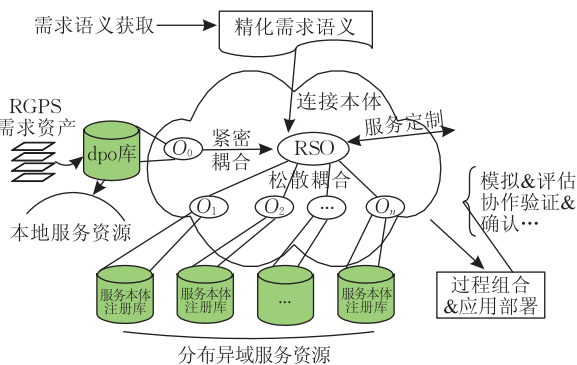


图 5 需求语义互操作载体-连接本体

连接本体既不是覆盖原有本体的新本体,也不是纽带本体,而是基于用户需求符号本体(RSO)语义,和诸多原有本体(包括领域问题本体,以及诸多非领域的采用图 2 元描述封装的服务本体)进行语义匹配,所得到的往往是诸多原有本体的子本体,并且由这些子本体基于用户需求符号本体的控制结构,松散耦合连接而成。其中,用户需求符号本体是

连接本体动态生成的驱动力,相当于纽带本体。我们已设计了相应的连接本体生成算法,对被交互信息的内容进行定义与理解的连接本体,为按需服务聚合生产提供语义信息载体和实现机理。

4.2 部分语义互操作性

语义互操作性依赖于系统的应用领域,不同应用领域的语义互操作性要求和目标不尽相同。一般来说,它是指两个软件单元或者系统间交换具有精确含意数据的能力,并且接收方能够准确地翻译或转换数据所携带的信息、信息所携带的知识,即信息、知识能够被理解,最终产生有效的行为协作结果,而且结果的有效性确认,由互操作的双方来协定。

语义互操作能力的级别,一般有:(1)含意互操作性。亦称深度语义互操作,或者完全语义互操作。(2)部分语义互操作性。软件服务实体之间,呈现出部分语义互操作,即仅仅能够满足共享连接本体的部分语义协定。部分语义互操作,往往采用共享协定的连接本体的百分比表示。100%表示含意互操作性,百分比较小说明语义互操作性能力较弱。(3)无语义互操作性。当软件服务实体之间的部分语义互操作性能力低于一定的阈值 k 时,出现无语义互操作。

定义 13(语义相似性). 对语义本体 $O1$ 和 $O2$,给定语义相似性度量 SC 和阈值 K ,

$$SC(O1, O2) = \sum_{i=1}^k w_i \cdot \sum_{j=1}^m similarity(O1_{ij}, O2_{ij})$$

其中 w_i 为预先设定的 k 种映射关系(如 *Equivalent-Class*、*SubClassOf*、*PartOf*、*InstanceOf*、*NoMatchingPair* 等)的权重值, $similarity$ 计算对应实体语义相似度;如果 $SC(O1, O2) \geq K$,称 $O1$ 和 $O2$ 具有语义相似性。

定义 14(语义互操作性度量). 对于互操作双方的实例本体到协定(或媒介)本体 O 的语义映射所得到的正确结果,即正确理解协定所得到的两个本体 $O1, O2$,基于这两个本体图所重叠覆盖的部分本体为 $O1 \cap O2$,计算其中所有叶子概念节点 i 的语义深度及其平均值,然后计算出两个本体之间的语义互操作能力 IC ,

$$IC = \frac{\sum_{i=1}^n \frac{1}{Depth(leaf_i) - 1}}{n}$$

其中,规定 $Depth(leaf_i)$ 大于 2。因为语义深度仅仅为 2 时,重叠覆盖的部分本体 $O1 \cap O2$ 没有互操作

的实际意义. 显然, IC 表示了互操作双方对语义互操作能力“协定”的共同理解的程度(语义深度). 如果, 语义深度与“协定”本体的深度相同, 说明互操作性双方能够实现深度语义互操作, 即含意互操作.

定义 15(部分语义互操作性). 语义本体 O_1 和 O_2 , 给定语义互操作能力(百分比)的阈值 IC_k , 如 $IC \geq IC_k > 0$, 则称 O_1 、 O_2 具有部分语义互操作性; 如 $IC_k \geq$ 语义相似性阈值 K 时, 则称 O_1 、 O_2 具有语义互操作性.

定理 1. 语义本体 O_1 和 O_2 存在语义相似性是其具有语义互操作的必要条件.

证明. O_1 和 O_2 具有语义互操作性 $\Leftrightarrow IC(O_1, O_2) \geq IC_k \geq$ 语义相似性阈值 $K \Rightarrow$ 语义本体 O_1 和 O_2 存在语义相似性. 证毕.

定理 2. 语义本体 O_1 和 O_2 具有大于等于规定阈值 K 的部分语义互操作性是其具有语义互操作的充分条件.

定理 2 由定义即可证明.

由于服务供应的动态性、松耦合与延迟绑定性, 异域服务资源中可替换(代)服务集的构造显得尤为重要. 当服务 A 突然无法使用时, 可从与 A 服务可替换集中选择可用在线服务, 以完成用户需求. 依据语义互操作性我们设计了 RSO 引导的基于语义互操作性的服务替换集生成算法(算法 1), 算法在服务互操作展开或应用部署时在后台并行执行以有效提高运行效率, 减小时间开销.

算法 1. 基于语义互操作性的服务替换算法 SIRepSer.

输入: (SRR, RSO_i, k), 其中 SRR 是服务语义本体注册册, RSO_i 是与 RSO 要求匹配的子本体, K 为互操作能力阈值

输出: S_{RSO_i} 与 RSO_i 满足语义互操作性的可替换服务集

```

1. begin
2.  $S_{RSO_i} \leftarrow \text{nil}$ ;
3.  $S_{\text{inf}} \leftarrow$ 提取  $RSO_i$  中领域信息;
4.  $SUB_{SRR} \leftarrow$ 根据  $S_{\text{inf}}$  抽取  $SRR$ ;
5. for  $\text{vmode} = 1$  to  $|SUB_{SRR}|$  do
6.   while  $SUB_{SRR}(\text{vmode})$  与  $RSO_i$  满足语义相似性 do
7.     if  $IC > K$ 
8.       then  $S_{RSO_i} \leftarrow SUB_{SRR}(\text{vmode})$ 
9.     end if
10.  end while
11. end for
12. return  $S_{RSO_i}$ 
13. end

```

4.3 RSO 制导的服务聚合

本节基于用户需求主导生成的连接本体, 讨论用户需求的语义展开中语义互操作性聚类的问题, 以解决服务聚合应用的难题. 连接本体是不同粒度、不同层次的网络式本体(networked ontology), 对于面向领域、面向服务社区的网络化软件来说, 连接本体及其互操作性理论与方法的研究, 具有重要的意义.

所谓连接本体(CO)级含意互操作性聚类, 是基于需求符号本体(RSO)的控制结构, 通过获取 O_i 的行为, 在情境与用户偏好等约束下, 实现耦合连接的本体 O_i 之间的行为的协作. 为了提高用户对于系统服务的体验质量(QoE), 激励软件复杂系统的群体行为的涌现, 在进行 CO 级的含意互操作聚类时, 我们考虑如下约束因素: ① 用户偏好、情境约束相应行为的协作; ② 不同控制结构下的含意互操作聚类的强度; ③ 含意互操作性聚类的并行、异步的复杂性及其相互影响等.

基于 RSO 中的控制结构, RSO 调用并协作服务本体 $O_i (i=0 \sim n)$ 之间的行为. 不同的控制结构, 导致互操作性聚类强度的不同. 在这里, 我们考虑几种基本的控制结构: 顺序、选择、分支-合并、任意次序、循环. 其中, 顺序是指一组服务依次执行; 选择是指从一组服务功能中根据条件选择其中一个协作; 分支-合并是指一组服务功能的并发协作, 当所有服务执行完成后, 整个结构才执行完成; 任意次序是指允许一组服务以一种未指定的但非并发的次序执行, 必须保证所有过程构件都能够执行和完成; 循环是指只要某个条件不满足(或者满足)时, 就重复执行某些特定服务. 对于这几种控制结构而言, 通过循环进行聚类的服务之间具有强条件依赖关系, 其互操作性聚类强度最高; 通过顺序聚类的服务, 存在之间的数据依赖或条件依赖, 也具有较高的互操作性聚类强度; 选择、分支-合并的互操作性聚类强度依次降低; 通过任意次序聚类的服务之间, 不存在任何条件或数据依赖, 其互操作性聚类强度最弱.

定义 16(CO 的语义行为协作强度). CO 级的

行为协作强度值为 $S_{CO} = \frac{\sum S_i}{n}$, $S_i = p_i \sum S_{ij}$. 其中, S_i 表示第 i 个过程模块的协作强度值, S_{ij} 表示 i 个过程模块中第 j 个过程单元的协作强度值, p_i 表示第 i 个过程模块的协作强度参数(由过程模块中唯一的控制结构决定).

算法 2. RSO 制导的服务聚合算法 RSO2BPEL.

输入: $SerReqSEMANTIC$, 需求获取阶段得到的服务需求语义描述

输出: $BPEL_{RSO}$, RSO 制导的 BPEL 流程

```

1. begin
2.  $MatchingServiceSet \leftarrow nil$ ;
3.  $RSO \leftarrow extracting(SerReqSEMANTIC, P)$ ;
   /* 抽取子本体 */
4.  $O_0 \leftarrow matching(RSO, DPO)$ ;
5.  $N \leftarrow RSO$  中原子过程数;
6.  $M \leftarrow O_0$  匹配的原子过程数;
7. if  $M \geq 1$  then  $MatchingServiceSet \leftarrow O_0$  end if
8.  $I \leftarrow N - M$ ;
9. while  $I \geq 1$  do
10.  cobegin /* 并行 */
11.   for  $j=1$  to  $I$  do
12.      $SIRepSer(SRR, RSO_j, K)$ 
13.   end for
14.    $T \leftarrow [1 \dots I]$ 
15.   foreach  $t \in T$  do
16.     if  $|S_{RSO_t}| \geq 1$  then
17.        $O_t \leftarrow choiceAnyOnline(S_{RSO_t})$ ;
   /* 可优化选择互操作性高的服务 */
18.        $MatchingServiceSet \leftarrow O_t$ 
19.     end if
20.   end foreach
21. coend
22.  $I \leftarrow I - 1$ 
23. end while
24.  $transfer(MatchingServiceSet, RSO, BPEL_{RSO})$ ;
   /* 转换 OWL 格式 RSO 及匹配服务信息到 BPEL */
25. return  $BPEL_{RSO}$ 
26. end

```

就本例而言,在 RSO 中包含两种控制结构:“任意次序”和“顺序”,通过这两个控制结构划分了两个过程模块,第一个是通过“任意次序”关联的 3 个过程单元“查询天气”、“查询体育馆”和“预定餐馆”,这 3 者均是原子过程.因此,其协同强度值都是 1.第 2 个是通过“顺序”关联的 3 个过程单元“计算出发时间”、“安排出发路线”和“显示出行路线”,这 3 者均是原子过程.因此,其协作强度值都是 1.这样,该 RSO 所在的 CO 的语义行为协作强度值为

$$S_{CO} = \frac{0.1 \times 3 + 1 \times 3}{6} = 0.55.$$

如果上述 6 个原子过程都是通过顺序关联,则其语义行为协作强度值为 1,显然这种情况下的关联度更强.

CO 级含意互操作性聚类中,还应该考虑网络

情境信息对行为协作的约束性影响,减弱或者增强协作强度,甚至阻断所约束的行为,提供可调控的行为聚类机制,实现提高用户应用网络化软件系统服务的体验质量 QoE 最终目标.

以 RSO 需求过程语义描述为中心,我们设计了 RSO 制导的服务聚合算法(算法 2),其中用到的子本体抽取见文献[19-20].为简化起见,算法以原子过程来展开服务匹配和聚合,如果针对不同目标分解的过程组合情况,则需要设定各目标所对应过程之间的组合关系,这将另文讨论.如图 6 的 RSO 过程语义,其中“安排出发路线”和“显示出行路线”匹配 DPO 中服务上层语义(O_0),其它原子过程分别匹配异域服务资源的上层描述语义 O_1 、 O_2 、 O_3 .

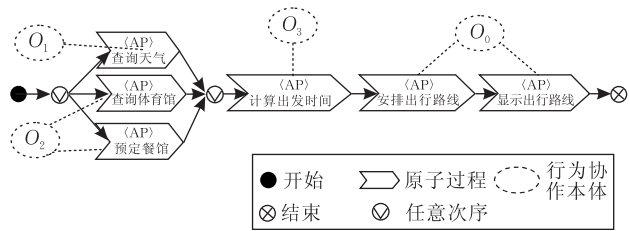


图 6 RSO 控制结构的语义行为协作强度

算法中也可对获取的多个 RSO 过程语义进行预处理,计算其语义行为协作强度,依据约定的聚类调控原则,选择理想 RSO 过程,而后进行服务聚合,以提供可调控的行为聚类.

5 实证研究分析

本文方法的可操作性是:(1)需求语义获取阶段可完成需求语义实例化;(2)从需求语义中抽取需求符号本体 RSO(RDF/XML 形式,存在于需求语义过程实例描述中);(3)网络服务描述采用本体封装并注册,跨域服务的描述本体注册库已建立.

为了实际验证本文方法的有效性,需要设计相应的实验实证策略,开发必备工具及算法的比较评估,其流程见图 7.其中需求语义获取平台采用 PHP5 开发,服务资源聚合部分采用 JAVA 技术,应用场景是用户根据所在地点自动搜寻附近加油站并短信通知,进而获得行车路线和银行卡支付等出行指导.根据实验流程需首先注册相关服务,如 GPS 定位服务、短信通告服务等,需求语义获取平台在线运行,针对交通出行问题获取对该应用场景的需求语义并实例化,通过我们以前的工作^[19]提取场景应用过程语义(RSO),运行算法 1 得到可替换服务集

合,运行算法 2 得到流程描述,自动连接 ODE^① 引擎完成阶段性按需服务供应。

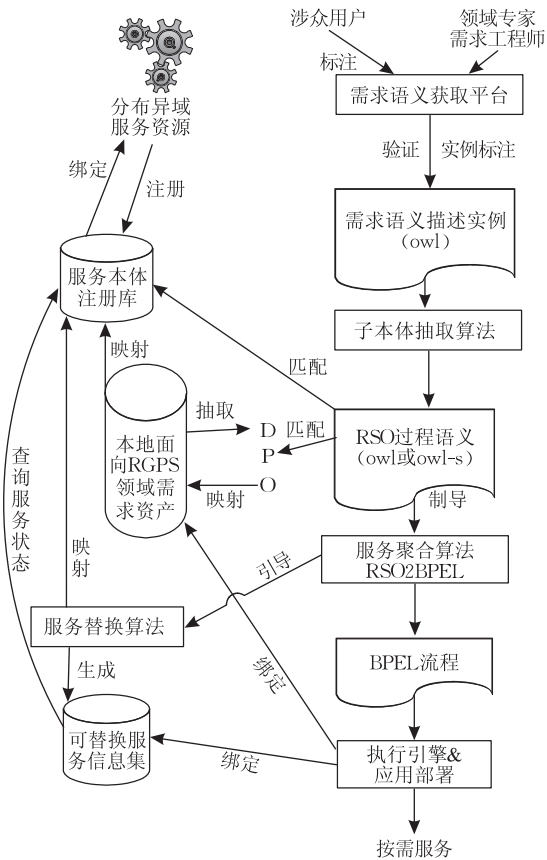


图 7 需求语义驱动的服务聚合生产实验流程

分布异域服务的具体实现千差万别,传统 WSDL 文件缺失语义信息,OWL-S 缺乏领域共性需求语义和行为协作语义,为顺利匹配和查询服务信息,需要进行服务的本体封装与注册,同时为达到本体注册库间的互操作,本体注册应遵循 ISO MFI 19763-3^[21] 标准.为此开发了基于 MFI 的 Web 服务注册管理平台^②用于注册 Web 服务并对其进行语义标识,提供 Web 服务注册接口,能够根据 Web 服务的 WSDL 地址解析 WSDL 文件;采用半自动化匹配技术,用领域本体对服务进行标注;提供 Web 服务查询接口,包括关键字查询和语义查询,能够根据语义查询参数查询出所需 Web 服务;服务注册者能够修改自己注册的 Web 服务,重新进行语义标识.系统的 19 项软件功能通过了软件测评中心的软件功能符合性测试。

面向领域的共性需求,设计城市交通和物流配送两个领域的基于 RGPS 的需求语义资产.为提高匹配率,借鉴数据库系统的索引机制,针对问题(如交通的出行问题、物流的订单处理问题)提取 DPO,构建 DPO 抽取算法和 DPO 库.服务替换采用 4.2 节

的基于语义互操作性的服务替换算法,并把该算法在 K 取不同值时与传统的基于 WSDL 的无语义的全文搜索 (FT) 和纯相似度搜索 (PS) 进行了相关比较。

需求语义获取平台,我们设计了以 OntoWiki 为内核的语义获取和标注功能,并外接 DIG 接口的推理器 (Pellet) 按照设定的推理功能验证需求语义. RSO 过程语义制导的服务聚合算法生成 BPEL 流程驱动执行引擎进行应用服务部署以提供满足涉众用户的服务响应。

实验的部分结果和系统见图 8. 其中图 8(a) 是服务注册平台中 DPO 结构展示,面向的问题是出行;服务封装按照图 2 提出的描述方式,所有服务采用本体封装并注册是本文方法可操作性的前提,服务只有在注册平台注册后才能被聚合、绑定与部署;图 8(b) 是基于开源语义 wikis 引擎 Ontowiki 进行二次开发设计的需求语义获取平台,可在底层本体 SORSO (见 3.2 节) 制导下进行需求文本的标注,从而获得需求语义实例化描述;图 8(c) 是基于语义互操作性的服务替换算法在 K 取不同值时与 FT 和 PS 在服务数不同时查准率的比较,结果显示,随着服务数增长,在聚合生产过程中所用的基于语义互操作性的服务替换算法查准率优于传统方法;图 8(d) 主要说明,为了重用需要,本地建设了面向领域基于 RGPS^[15] 的需求资产,为顺利完成需求资产的检索而设计了针对具体问题的 DPO 以方便需求资产的查询. DPO 是从需求资产(也是一个规模较大的本体)自动生成的子本体,其抽取深度、查询匹配度、抽取时间的结果可在图上得到,结果可用于指导 DPO 的生成过程以提高本地需求资产检索效率,达到重用需求和加快服务聚合进程的目的.实验主要针对小规模的交通领域本体(概念规模小于 200)和以 OWL 格式存储的需求资产角色 R、目标 G、过程 P 进行.在模拟实验中,DPO 的抽取深度初始值为 2.通过改变深度值,可以得出不同的抽取深度时匹配度和时间表现,从而分析遍历深度对整个连接本体生成的影响.实验结果表明,较大的 DPO 抽取深度对应于较高的与 RSO 的匹配度,因为当 DPO 的抽取深度较大时,DPO 子本体的规模也相对较大,根据空间局部性原则,被重用的机率也较大,从而提高了与 RSO 的匹配度.但较大的抽取深度意味着生成

① <http://ode.apache.org/>

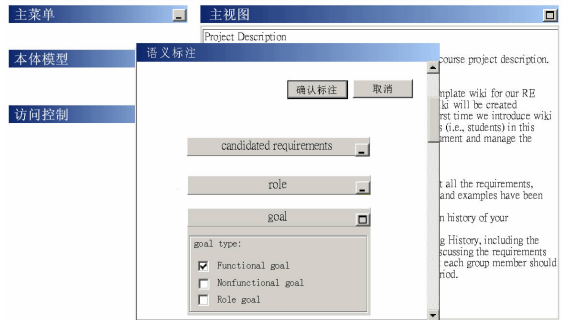
② <http://61.183.121.132/wsrr/>

DPO 的代价较大, 也会增加匹配时间开销; 同时当子本体抽取深度从 6 增加到 8 时, 可提高 DPO 抽取代价, 但匹配性能并未明显提高, 说明单纯通过提高 DPO 抽取深度来改善匹配性能并非一直有效, 因此

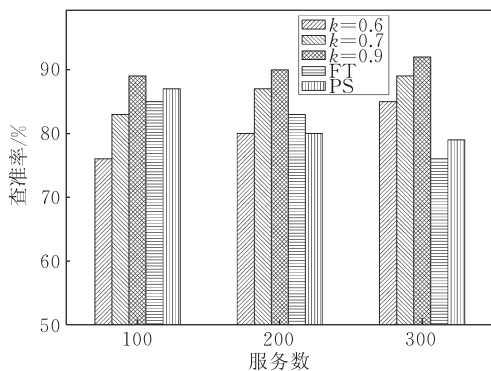
设定一个合理的 DPO 抽取深度对于优化匹配性能十分重要. 本实验的缺陷是源本体规模都不大, 下一步将在较大规模的源本体进行实验以获取更有价值的结果.



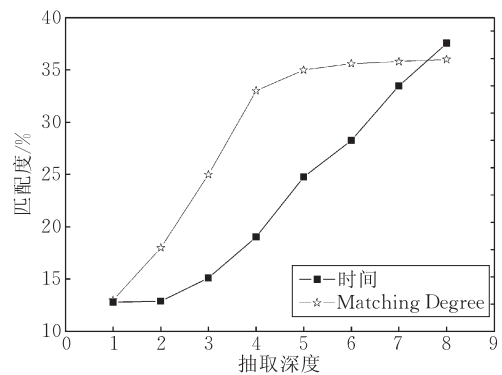
(a) 服务本体注册



(b) 需求语义获取与标注



(c) 服务替换算法查准率比较



(d) DPO抽取实验

图 8 实证实验分析

实验可即时获取初步需求语义, 并连接相关注册服务完成用户标注的需求, 实现按需服务聚合生产. 同时整个实验流程建立在需求语义获取平台基础上, 需求语义的不断演变驱动后续按需服务的演化生长、随需而动、QoE 持续改进.

6 相关工作比较

6.1 语义 wikis 用于需求获取工作

国际上的相关工作表明: 语义 wikis 适用于软件需求的抽取与文档化平台^[8], 其中 SOP-Wiki 和 WikiReq 已成功用于分布式需求获取与文档生成, 它们都基于语义 Mediawiki 平台开发. 国际上还有一些协作研究项目正在进行, 如 SoftWiki, 其特别关注于软件需求的语义协作; IkeWiki 用于需求知识的协作构建, 目前此项目已转为欧盟资助的 KiWi 并持续开发, KiWi 基于 JBOSS 和 Java 开发, 支持推理.

WikiReq^[14] 使用语义 Mediawiki 管理系统和组织需求, 组合使用 Si* 面向目标语言. 首先, 需求通

过一些预定义的窗体集获取; 进一步, 自动定义 Si* 主概念中的语义关系; 另一特性是可以在特定标签页中提供涉众对需求的讨论机制; 为支持与 IDE 平台的互操作, 语义标注的需求可转换为另一种实例: EMF (Eclipse Modeling Framework), 已支持 MDA 功能. 但 WikiReq 不支持需求语义的动态生成, 并且 Si* 语言也不能完全适合面向服务的软件需求描述.

SOP (Software Organization Platform) Wiki^[8] 是 RISE 项目的后继. RISE 项目旨在构建一个 Wiki 来抽取需求、开发初始体系结构和基于需求分配开发任务. SOP-wiki 解决了 RISE 项目中出现的问题. 首先, 它能收集页面间的连接并锁定, 使其易于对页面实施版本化管理. 另一特性是版本化标签, 用户能选择 Wiki 页面去分配一个版本标签. 再者, SOP-wiki 提供导出 Wiki 内容 (如需求) 到文档 (Open Office documents) 功能; SOP-wiki 也可增加语义信息到页面. 对比我们的工作, SOP-wiki 无法提供底层需求模型制导涉众进行语义标注, 因此需求语义是不精确的同时缺乏智能应用.

SoftWiki^[22]关注大型和分布涉众用户的语义分类和聚合软件需求,需求的核心概念和相互关系定义在 SoftWiki 的需求工程本体(SWORE)中.经常使用关系类型也被 SWORE 定义,描述了需求的相互依赖.通过这种方式,所有的需求相互连接.典型的语义 wikis 特性(SW7)也被 SWORE 使用,可连接到外部知识资源(如公共领域知识、公司特定策略等).将来也可导出需求到其它的需求管理工具,为此需求须转化为 RIF 格式.显然,SoftWiki 不能通过群体智慧来演化修订 SWORE,同时 SWORE 只是一个普适的上层需求本体并没有充分考虑服务软件的需求特性(例如服务和过程的标识)以支持网络化软件需求语义获取,因而对服务软件生产作用不大.而本文获取部分的底层需求语义本体重点针对服务和过程进行标识、关联,同时通过涉众群体参与和专家智慧而不断修正,以符合需求语义演化生长的要求.

6.2 语义互操作展开用于服务聚合工作

目前,国际上十分关注信息系统中互操作性能力层次的划分研究,比较权威的划分方案来自于美国军方的 LCIM 模型^[23]. LCIM 互操作性模型的应用领域主要是模拟和仿真领域的信息系统互操作性.其中,语义互操作性位于该互操作性模型中的第 3 层,关注数据含意的共享;而概念互操作性(conceptual interoperability)位于第 6 层,在数据含意共享的基础上,实现与数据含意相关的假定和限制条件的集成.该模型限于应用问题的需要,模型第 3 层中对于语义互操作性中的部分语义互操作性没有展开研究,对于网络环境中松散耦合连接的软件服务单元间存在大量的部分语义互操作性问题,很难提供支撑.

关于互操作性能力如何度量这一问题,目前还没有成熟的计算方法. SIM-TBASSCO(语义互操作性度量-基于模板的软构件组合的语义互操作性保证方法)项目是由美国南加州大学承担的研究项目,其研究目标是通过评价软构件的功能和数据接口的语义互操作性,发现合适的功能和数据映射,并预测软构件组合的性能.主要解决方案是通过引入“语义度量仪(semantic gauge)”的概念,分别对构件的功能语义(functional semantics)和数据语义(data Semantics)的匹配程度进行评价.“功能语义度量仪”对构件功能接口间的包含关系和推理关系进行相似性度量;“数据语义度量仪”对构件间交互的数据内容和结构(类型)进行相似性度量.其中对数据内容

的相似性度量采用了本体图中空间距离(spatial distance)的概念.但是,但该项目中并没有对语义互操作性的层次给出划分,而只是通过度量结果来选取合适的构件组合方案.

在国际上,对于连接本体的研究是一个比较新的方向.最初,如研究原始的异质异构的领域本体,如何生成一个协同一致并覆盖原始本体的新本体及其本体集合体技术(如本体映射、匹配、融合)等;2007 年 Karim 等以连接本体作为纽带本体,并不要求覆盖原有本体,注重研究建立原始本体之间关联的转换规则、中间概念等,提出了一个连接本体应用框架^[24].但是该应用框架没有涉及到核心的语义互操作性问题.2008 年 Cregan 等提出通过将本体连接起来去建立语义互操作,并且给出了诸多基因本体的连接本体例子^[25].但是,该文中并没有给出本体的连接方式、连接生成的驱动力、建立语义互操作的方法及其中的关键内容等.

本文工作中不仅提出了部分语义互操作性和语义互操作性的度量方法,同时证明了其必要和充分条件,设计了基于语义互操作性的服务替换算法.我们认为连接本体既不是覆盖原有本体的新本体,也不是纽带本体.而是基于用户需求符号语义和诸多原有本体(包括领域问题本体以及诸多非领域的服务本体)进行语义匹配,所得到的往往是诸多原有本体的子本体,并且由这些子本体基于用户需求符号本体的控制结构,松散耦合连接而成.其中,用户需求符号本体是连接本体动态生成的驱动力,相当于纽带本体.本文基于这种用户需求主导生成的连接本体,进一步研究用户需求的语义展开中语义互操作性聚类的问题,进而努力解决服务聚合应用的难题.

7 总 结

本文对基于服务的大型复杂网络化软件的需求语义获取技术与需求语义驱动的服务资源聚合生产展开研究,系统分析了其需求语义获取的自适应特性,提倡依据涉众群体智能与专家知识相结合,演化生长获取领域需求语义模型.在需求语义模型指导下进行需求语义的实例化标注,同时引入验证与推理机制,采用适用的语义 wikis 引擎为基本内核,扩展了许多需求语义获取功能,设计了涉众智能参与的复杂网络化软件需求语义获取体系结构和相关平台.不仅得到传统需求文档,而且还可导出相应的需

求语义制品用于后续需求语义驱动的服务软件互操作聚合和基于需求语义的服务定制,通过需求语义互操作展开选择连接本体作为语义信息载体进行服务资源聚合,关注服务的上层语义实体表示、获取和应用,以敏捷内容(smart content)牵引和简化底层服务的具体实现。

全文给出了一个需求语义驱动的需求服务聚合生产网络化软件的方法,即:应用语义 wikis 设计的需求语义获取平台,网上涉众群体参与和需求专家合作生成软件的需求语义实例化描述;从需求语义中抽取需求符号本体 RSO(RDF/XML 形式,存在于过程实例描述中);以 RSO 为中心构建连接本体来匹配、聚类本地和异域服务以实施网络化软件的生产。

同时系统研究了语义互操作性特别是部分语义互操作性,设计了系列工具、算法评估和实验流程。理论与实证研究证实了所提方法的有效性与实用性,可使需求语义在云服务时代发挥更大的作用。

进一步的工作包括:平台上线运行以获取更有价值的实效验证和评估;需求语义制品在相关 IDE 工具的应用;需求语义制导的服务定制等。

参 考 文 献

- [1] He Ke-Qing, Peng Rong, Liu Wei et al. Networked Software. Beijing: Science Press, 2008(in Chinese)
(何克清, 彭蓉, 刘玮等. 网络式软件. 北京: 科学出版社, 2008)
- [2] Jin Zhi, Liu Lin, Jin Ying. Software Requirements Engineering: Principles and Method. Beijing: Science Press, 2008(in Chinese)
(金芝, 刘璘, 金英. 软件需求工程: 原理与方法. 北京: 科学出版社, 2008)
- [3] Lu R, Jin Z, Chen G. Ontology-oriented requirements analysis. Journal of Software, 2000, 11(8): 1009-1017
- [4] van Lamsweerde A, Letier E. From object orientation to goal orientation: A paradigm shift for requirements engineering//Radical Innovations of Software and Systems Engineering in the Future. Lecture Notes in Computer Science 2941. Springer-Verlag, 2004: 325-340
- [5] Mylopoulos J, Chung L, Yu E. From object-oriented to goal-oriented requirements analysis. Communications of the ACM, 1999, 42(1): 31-37
- [6] Liu Wei. Research on services-oriented software requirements elicitation and analysis[Ph. D. dissertation]. Wuhan University, Wuhan, 2008(in Chinese)
(刘玮. 面向服务的软件需求获取与分析研究[博士学位论文]. 武汉大学, 武汉, 2008)
- [7] Zhang Shu-Ren, Fang Mei-Qi. Web 2.0 and Complexity Transformation of Information System. Beijing: Science Press, 2008(in Chinese)
(张树人, 方美琪. Web 2.0 与信息系统复杂性变革. 北京: 科学出版社, 2008)
- [8] Decker B, Ras E, Rech J et al. Wiki-based stakeholder participation in requirements engineering. IEEE Software, 2007, (2): 28-35
- [9] Bart H, Liang P. A survey of semantic wikis for requirements engineering. Software Engineering and Architecture Group, Department of Mathematics and Computing Science, University of Groningen, Technical Report: RUG-SEARCH-09-L03, July 27, 2009
- [10] Antoniou G, van Harmelen F. A Semantic Web Primer. Cambridge, MA, USA: MIT Press, 2004
- [11] Bizer C, Heath T, Berners-Lee T. Linked data — The story so far. International Journal on Semantic Web and Information Systems, 2009, 5(3): 1-22
- [12] Schaffert S, Bry F, Baumeister J et al. Software technology-semantic wikis. IEEE Software, 2008, 25(4): 8-11
- [13] Krotzsch M, Vrandeic D, Volkel M et al. Semantic wikipedia. Journal of Web Semantics, 2007, 5(4): 251-261
- [14] Abeti P C L, Moretti R. Wiki-based requirements management for business process reengineering//Proceedings of the ICSE09 Workshop on Wikis4SE. Vancouver, Canada: IEEE Computer Society Press, 2009: 14-24
- [15] Wang J, He K, Peng R. Rgps: A unified requirements meta-modeling frame for networked software//Proceedings of the 3rd International Workshop on Advances and Applications of Problem Frames (IWAAPF'08) at 30th International Conference on Software Engineering (ICSE'08). Leipzig, Germany, 2008: 29-35
- [16] He F, Fukazawa Y. Requirement life cycle and its integrated modeling framework. Special Issue on Software Engineering and Complex Networks. Dynamics of Continuous, Discrete and Impulsive Systems, Series B: Applications and Algorithms, 2007, 14(Sup. 6): 227-232
- [17] He Ke-Qing, He Fei, Li Bing et al. Research on service oriented ontology meta modeling theory and methodology. Chinese Journal of Computers, 2005, 28(4): 524-533 (in Chiense)
(何克清, 何非, 李兵等. 面向服务的本体元建模理论与方法研究. 计算机学报, 2005, 28(4): 524-533)
- [18] Liang P, Avgeriou P, Clerc V. Requirements reasoning for distributed requirements analysis using semantic wiki//Proceedings of the International Workshop on Knowledge Engineering in Global Software Development (KNOWING). Limerick, Ireland, 2009: 388-393
- [19] Mao Y, Wu Z, Chen H. Sub-ontology based resource management for web-based e-learning. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(6): 867-880

- [20] Wen B, He K, Wang J. Building requirements semantics for networked software interoperability. *Journal of Software Engineering and Applications*, 2010, 3(2): 125-133
- [21] ISO/IEC 19763-3: 2007. Information technology — Meta-model framework for interoperability (MFI) — Part 3: Meta-model for ontology registration. 2007
- [22] Lohmann S, Heim P, Auer S et al. Semantifying requirements engineering — The softwiki approach//Proceedings of the 4th International Conference on Semantic Technologies (I-SEMANTICS). Graz, Austria: ACM, 2008: 182-185
- [23] Tolk A, Muguira J. The levels of conceptual interoperability

model (LCIM)//Proceedings of the IEEE Fall Simulation Interoperability Workshop. Orlando, Florida, USA, 2003

- [24] Karim S, Latif K, Tjoa A. Providing universal accessibility using connecting ontologies: A holistic approach//Universal Access in Human-Computer Interaction Applications and Services. *Lecture Notes in Computer Science 4556*. Springer-Verlag, 2007: 637-646
- [25] Cregan A. W3C semantic Web ontology languages: OWL and RDF tutorial. ISO/IEC JTC1 SC32 11th Open Forum on Metadata Registries. Sydney, Australia, Technical Report, Tutorial, 2008



WEN Bin, born in 1970, Ph. D. candidate. His current research interests focus on semantic-enabled requirements engineering for service.

His current research interests include requirements engineering and service computing.

LIANG Peng, born in 1978, Ph. D., associate professor. His current research interests include requirements engineering for service and software architecture knowledge.

WANG Jian, born in 1980, Ph. D., lecturer. Her current research interests include requirements engineering for service and ISO standardization.

HE Ke-Qing, born in 1947, professor, Ph. D. supervi-

Background

Services computing has become new paradigm and received significant attentions & applications for IT participants and information consumers. Requirements engineering (RE) is crucial to the success of software development projects, especially for networked software based on services. Significant differences exist between service-oriented and object-oriented software development, service-oriented development focuses on finding and reusing existing services resource to satisfy users requirements, whereas nowadays requirements modeling methods are deficient to support the aspect. Service-Oriented RE (SORE), like traditional requirement engineering, concerns with specification and analysis of system requirements and constraints. SORE focuses identifying services and processes(workflows), and modeling the application using identified service and workflow specifications so that real services and workflows can be discovered either at design time (static SOA) or at runtime (dynamic SOA).

Faced with complex stakeholders and requirements, large scale networked software should combine with the research of IT and information system to complex structure by introducing complex science theory. Now plenty of Web 2.0 applications have fully showed it. How to explore novel requirements acquiring technique to answer complex situation for the development of networked software? How to facilitate the subsequent automated and on-demand services software

delivery just in time through attaching semantic information? These issues are the motivation and focus of the authors' work. They have partly investigate these questions and argued that requirements acquisition should fully make use of stakeholders participation and folks intelligence. Through some collaborative platforms of social software such as semantic wikis and RSS/Atom etc., and exploiting simple annotating function plus evolutionary mechanism, software requirements specification and requirements semantics will efficiently emerge. At the same time, requirements acquiring process is continuous in accordance with networked software features, i. e. on-demand changing and uninterrupted growing.

Semantics RE(SRE) is proposed and attempts to solve stakeholder-centric service aggregated production. SRE is the deepening of traditional RE via plus semantics on requirements artifacts to expose the relationships among requirements elements explicitly. And semantics can penetrate the process for service industry chain containing requirements semantics acquiring, interoperability extending and services customization directed by requirements semantics. This research has been supported by the National Basic Research Program of China (973 Program) under grant No. 2007CB310801, and the National Natural Science Foundation of China under grant Nos. 60970017, 60950110352.