

一种面向语义 Web 的组合服务演化方法研究

王晓璇^{1,2} 鲍爱华³ 缪嘉嘉³ 丁科³ 王真³

(同济大学测量与国土信息工程系 上海 200092)¹

(中国水产科学研究院东海水产研究所渔业资源遥感信息技术重点开放实验室 上海 200090)²

(中国人民解放军理工大学全军网格技术研究中心 南京 210007)³

摘要 为了敏捷、有效地调整动态环境下的组合服务,系统地提出一种面向语义 Web 的组合服务结构化演化方法 EM4CS。EM4CS 方法采用 OWL-S 语言描述的组合式语义 Web 服务为研究对象,将组合服务演化过程划分为 3 个部分、6 个阶段,分别从演化需求获取、演化需求形式化表示、语法一致性维护、语义一致性维护、演化结果确认和演化结果发布与传播 6 个方面覆盖组合服务演化周期。EM4CS 方法从系统工程的角度分析组合服务演化过程,以迭代求精的方式逐步实现演化需求,同时维持服务描述语法和语义的一致性。介绍了遵循 EM4CS 方法的组合服务演化支撑系统 ESS4CS,该系统能够全程辅助知识工程师完成组合服务演化工作,能有效地提高组合服务演化的效率和精度。

关键词 语义 Web,组合服务演化,组合服务,EM4CS,ESS4CS,OWL-S

中图法分类号 TP393 文献标识码 A

Research on the Semantic Web Oriented Method for the Evolution of Composite Service

WANG Xiao-xuan^{1,2} BAO Ai-hua³ MIAO Jia-jia³ DING Ke³ WANG Zhen³

(Department of Surveying and Geoinformatics, Tongji University, Shanghai 200092, China)¹

(Key Laboratory of Fisheries Resources Remote Sensing and Information Technology Resources, East China Sea Fisheries Research Institute, Chinese Academy of Fishery Sciences, Shanghai 200090, China)²

(PLA Research Center for Military Grid Technology, PLA University of Science & Technology, Nanjing 210007, China)³

Abstract To adapt composite services with changes in dynamic environment agility and effectively, a semantic Web oriented method for the evolution of composite service which named EM4CS was proposed systematically. Taking composite services described by OWL-S as research object, EM4CS divides the evolution process into 3 parts, and covers the evolution cycle in 6 stages, i. e. evolution requirements capturing, formal representation of evolution requirements, syntactic consistency maintenance, semantic consistency maintenance, confirmation of evolution results, publication and broadcasting of evolution results. The composite service evolution process was analyzed from the perspective of system engineering, and the evolution requirements were fulfilled in an iterative refined way, in which the syntactic and semantic consistency were also maintained. The composite service evolution support system, which follows EM4CS, named ESS4CS, was also introduced in this paper. ESS4CS can assist knowledge engineer in the whole process of composite service evolution, and the efficiency and precision of the evolution process are also improved.

Keywords Semantic Web, Composite service evolution, Composite service, EM4CS, ESS4CS, OWL-S

1 引言

面向服务的架构(SOA)是近年来较为流行的软件体系结构风格^[1],其主要思想是使用 Web 服务对企业中的异质应用和数据资源进行封装,然后使用服务组合实现企业应用集成。SOA 架构中的技术标准大都基于语法层面定义,缺乏严格的语义基础,这使得 SOA 应用的自动化受到限制,如服务的自动发现、组合和执行等^[2]。语义 SOA^[3,4]是 SOA 架构的最新发展趋势,其核心思想是在传统 SOA 架构中引入语义技术,使其支持服务语义描述、语义注册、语义发现、自动匹配和自

动组合等新的特性。语义 SOA 的核心部件就是具备严格语义基础的语义 Web 服务^[5]。

随着经济运行节奏的加快,企业应用面临着商业环境和用户需求不断变化的严峻挑战。据 IDC 2007 年针对国内企业的调查数据显示,近 40%的企业认为“适应业务需求的不断变更”是当前信息系统建设面临的最大挑战^[6]。因此,对于代表企业应用的组合服务而言,如何对其进行敏捷有效的调整,从而与快速变化的外部环境相适应,成为 SOA 应用研究所面临的关键问题。

对此,目前学术界关注较多的是关于提高组合服务运行

实例动态适应性的研究。例如,通过语义技术使具体服务选择与流程分离,实现组件服务动态发现、绑定和执行,避免因预先捆绑的组件服务失常导致组合服务执行失败^[7];基于差异化服务的思想,为每个组件服务提供多个不同的实例,在面对不同的用户需求时(如 QoS 需求)调用不同的服务实例^[8-10];通过预定义动态重构策略的方式,使组合服务在遇到异常时按照预定义策略动态调整执行路径,实现组合服务的自适应调整^[11];使用 Agent 对组件服务进行封装,将组件服务间的编排看作为 Agent 的会话协议,通过会话协议的监控和异常处理实现组合服务运行实例的自适应调整^[12];等等。但在组合服务面临的众多变化需求中,不仅要求组合服务能够在运行时对其实例进行动态调整,也要求组合服务的聚合逻辑能够在需要时进行重构;不仅需要微观层面提高组合服务运行实例的动态适应性,还需要在宏观层面实现服务模式及功能的变革,使其与不断变化的业务流程、用户需求和商业环境相适应。因此仅仅研究服务运行实例的动态适应性是不够的,还需要对“静态”的组合服务模式适应性调整问题进行系统研究。

针对上述问题,本文以采用 OWL-S 语言描述的组合语义 Web 服务为研究对象,系统地提出一个面向语义 Web 的组合服务结构化演化方法 EM4CS (Evolution Method for Composite Service)。该方法以语义 Web 环境下组合服务实际应用为背景,以机器辅助处理为目标,将组合服务演化过程分解为 6 个阶段,从服务演化需求获取、OWL-S 服务描述维护和演化结果传播 3 个方面覆盖组合服务演化周期。基于该方法,知识工程师能够在机器的辅助处理下有序地完成组合服务的演化,使服务组合模式能够随外部环境和用户需求的变化而进行快速调整,实现组合服务功能的变革。

2 面向语义 Web 的组合服务演化方法 EM4CS

在语义 Web 环境下,语义 SOA 架构的应用框架如图 1 所示。其中,资源层代表跨组织的异构数据资源和业务系统的集合,Web 服务实例层中包含对资源层进行封装的具体 Web 服务,通过这些服务,Agent 或应用系统能够采用标准的交互协议和消息传输格式对资源层进行调用;语义 Web 服务抽象描述层主要包括基于服务本体描述的语义 Web 服务集合,这些语义 Web 服务物理上并不与 Web 服务实例层的 Web 服务进行捆绑,而是由服务执行引擎在调用时自动发现具体的 Web 服务,然后在运行时自动绑定和执行;表现层主要通过调用特定的语义 Web 服务来实现具体的业务功能。

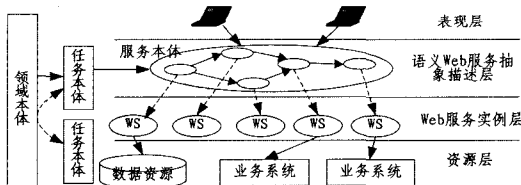


图 1 面向语义 SOA 架构的应用框架

本体是语义 SOA 架构的重要组成部分,其中领域本体对问题领域中的通用知识进行了明确、形式化表示,任务本体通常对特定任务或行为中的私有概念(如变量、参数以及数据结构等)及其关系进行形式化表示^[13],它与特定应用进行捆绑,

不同任务本体可以基于领域本体通过本体映射^[14]等技术来实现异质数据的互操作;服务本体对服务接口、调用信息和过程模型进行了形式化描述,是 Agent 或服务执行引擎对服务进行自动执行和监控的基础。

基于上述背景,本文对组合服务演化进行明确定义。

定义 1 在语义 Web 环境下,组合服务演化(Composite Service Evolution)是指组合服务描述在外部环境和用户需求发生变化时进行自适应变更,并对服务描述在改进过程中所出现的违反语法和语义一致性问题进行处理,使修改后服务描述所代表的组合服务在适应外部环境和用户需求变化的同时保持服务的有效性。

采用 EM4CS 方法对组合服务进行演化的流程如图 2 所示。EM4CS 方法主要包括演化需求获取、演化需求形式化表示、语法一致性维护、语义一致性维护、演化结果确认和演化结果发布与传播等 6 个核心过程。其中演化需求获取主要对现实环境中组合服务演化的原因进行分析,进而得到组合服务演化需求;演化需求形式化表示则基于组合服务演化本体对上述需求进行形式化表示,生成针对 OWL-S 服务描述的维护请求脚本,并交由机器辅助处理;语法一致性维护以 OWL-S 语法规则为基础,根据语法一致性约束和维护请求脚本生成语法维护树 SMT,并采用启发式搜索算法对 SMT 进行搜索,得到语法一致性维护方案,以保证组合服务描述语法的正确性;语义一致性维护首先对服务过程模型数据流的正确性进行验证,然后基于过程模型网 PM_net 对其控制结构的可靠性进行验证,以保证服务描述的有效性,使组合服务能够正确地执行;演化结果确认主要对上述 3 个过程所实施的变化进行确认,如果需要撤销则生成新的需求重新启动上述过程,否则接受对组合服务描述所做的修改;演化结果发布与传播是组合服务演化的最后步骤,其中演化结果发布的目的在于将服务描述的修改细节发布,使调用该服务的应用能够理解并同步该服务的变化,演化结果传播的作用在于采用传播策略对该组合服务的运行实例进行处理,使其与服务模式的修改同步。

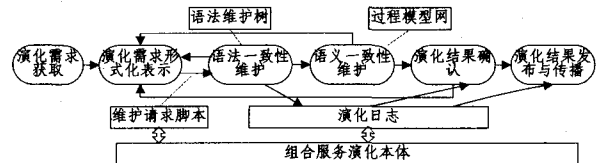


图 2 EM4CS 方法演化流程

除了上述 6 个主要的维护过程,EM4CS 方法中还包含演化本体、维护请求脚本、维护日志、语法维护树和过程模型网等 5 个核心数据组件。

(1)演化本体。组合服务演化本体是 EM4CS 方法的基础设施,它对 EM4CS 方法中常见的概念进行了严格的定义,是知识工程师与机器进行交互的基础。具体而言,演化本体包含 OWL-S 服务描述变化操作定义与变化关系推导规则描述两个部分。基于演化本体,知识工程师就能够将组合服务演化需求形式化表示为 OWL-S 服务描述维护请求脚本,在依据该脚本对服务描述进行修改后,机器能够基于演化本体对修改操作进行日志记录。以演化本体为基础,维护请求脚本和演化日志能够被其他知识工程师和异质应用共享,有利

于组合服务演化结果的发布与传播。

(2)维护请求脚本。维护请求脚本是知识工程师向机器传递演化需求的唯一渠道,“维护请求”是该脚本中的核心概念,每个脚本实际上对应于演化本体中“维护请求”概念的一个实例。在EM4CS方法中,维护请求由正向变化操作集和逆向变化操作集组成,前者表示在维护过程中需要实现的变化操作,而后者则表示维护过程中应当严格避免实施的变化操作。通过正反两个方面的约束,知识工程师就能够基于目标而非过程来定义维护请求。

(3)演化日志。演化日志对EM4CS方法中组合服务描述修改的历史信息进行语义记录,包括变化的类型、操作对象、实施时间、版本、变化原因以及批准实施的知识工程师等。通过演化日志,知识工程师能够掌握组合服务描述的修改轨迹。同时,演化日志也是演化结果发布阶段需要向其他用户和异质应用共享的核心数据。

(4)语法维护树。语法维护树SMT是EM4CS方法中对组合服务描述进行语法一致性维护的主要工具。树中节点代表搜索过程中产生的临时OWL-S服务描述,节点之间的弧代表两个服务描述之间所实施的变化。通过搜索,EM4CS方法能够在OWL-S服务描述语法一致性约束和维护请求脚本的共同驱动下不断扩展语法维护树,直到达到终止状态。这时语法维护树的叶节点所对应的OWL-S服务描述就是既满足语法一致性,又在最大程度上实现了维护请求的目标结果,而从该节点回溯到起始节点的路径就是一个语法一致性维护备选方案。

(5)过程模型网。过程模型网PM_{net}是一种扩展的Petri网,是EM4CS方法中对服务过程语义一致性进行检测的主要工具。与基本Petri网相比,PM_{net}对库所和变迁进行了扩展,使其能够表示服务过程模型中的数据流和控制结构。基于Petri网的理论和分析方法,我们能够通过PM_{net}的检测来发现过程模型所存在的冲突,如死锁、过程不可达等,并能够将冲突内容反射为服务过程模型中的元素,辅助知识工程师做进一步修改。

3 EM4CS方法过程详细描述

本节主要采用EM4CS方法对组合服务进行演化的具体过程进行描述。在此过程中,我们以图3所示的交易组合服务演化为例进行说明。

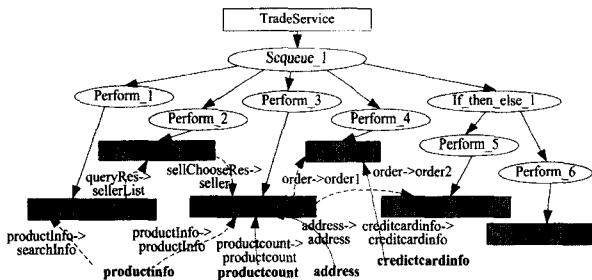


图3 电子商务交易组合服务

在该服务中,交易过程由产品查询、商家选择、订单发送、支付和物流配送等5个组件服务组成,但随着电子商务的发展,该流程逐渐难以与用户的需求相适应。例如,由于电子商务是虚拟交易形式,在交易中可能存在欺骗行为,因此客户在交易中需要一种相对客观的指标来判断商家是否可信,并以

此作为是否与该商家交易的依据。对此,现有的解决办法是在交易完成后由客户对商家进行评价(如好评、中评和差评等),通过大量的客户评价来恒量商家的信用。同时,对商家选择方式进行修改,支持按照信用选择交易商家的功能。为此,需要对图3所示的交易组合服务进行演化,使之与新的需求相适应。

3.1 组合服务演化需求获取阶段

演化需求获取阶段包括组合服务变化原因分析(P1)和演化需求识别(P2)两个子过程。过程P1主要在问题域对触发组合服务演化的原因进行分析,过程P2则依据分析结果识别组合服务演化的原始需求。

在语义Web环境下,对触发组合服务演化的原因进行分析是整个演化过程的出发点,也是对组合服务演化需求进行识别的基础。一般来说,促使组合服务演化的原因包括以下几个方面:(1)国家政策、法律法规发生变化;(2)组织目标调整;(3)用户需求变更;(4)业务流程优化。在过程P1中,组织领导者与知识工程师首先要结合上述可能的原因和组织的实际现状分析组织当前所面临的具体变化,判断业务流程重构的紧迫性和必要性,然后再确定演化的方向与目标,在此基础上在过程P2中对组合服务演化需求进行进一步识别。

过程P2主要由知识工程师根据P1所确定的演化方向和目标来识别具体的组合服务演化需求。一般来说,需求的识别方法包括自顶向下(Top-Down)和自底向上(Bottom-Up)两种类型。

(1)自顶向下。该方法的基本思路是在环境发生变化时,首先从总体上确定演化目标,然后将目标逐步分解,得到组合服务的演化需求。自顶向下的方式一般适用于组织有明确的演化目标和要求的情况,其挖掘的是显性的演化需求。

(2)自底向上。该方法的基本思路是根据组织业务系统的运行状况、日志数据、客户和合作伙伴的反馈等信息来挖掘组织业务流程中需要调整的部分,并进一步对演化需求进行识别。自底向上的识别方法通常适用于那些目标、要求不明确、但又存在着大量外在表现的情况,其挖掘的是隐性的演化需求。

在EM4CS方法中,演化需求获取阶段的输出是代表知识工程师演化意图的演化需求表(Evolution Requirements Table,ERT),例如电子商务交易组合服务的演化需求表如表1所列。

表1 演化需求表示例

编号	待演化流程	演化需求描述	修改操作	约束条件
1	ChooseSeller	对商家选择服务进行调整,使之支持基于商家信用度的选择	删除过程 ChooseSeller; 添加过程执行 CreditQuery 和 ChooseSellerByCredit, 通过这两个过程的组合来实现基于商家信用度的选择	---
2	Distribution	在配送完成后添加商家信用度评价功能	在过程执行 Distribution 后添加过程执行 CreditEvaluate	不影响其他过程调用

3.2 组合服务演化需求形式化表示阶段

演化需求形式化表示是连接实际问题与机器辅助处理的中心枢纽,其作用在于将非形式化表示的组合服务演化需求转化为机器可识别的OWL-S服务描述维护请求,使得机器

能够基于该维护请求对服务描述进行修改,进而达到组合服务演化的目的。下面,本文给出本阶段相关概念的定义。

定义 2 OWL-S 服务描述变化操作 $ch \in DesCh$ 是 OWL-S 服务描述间的全映射 (Total Mapping)。形式上, $ch = \langle name, args, preconditions, postconditions \rangle$, 其中: (1) name 是变化操作的名称,例如 AddProcess; (2) args 是变化操作的参数,如 (proc, type); (3) preconditions 是变化操作的前提条件,只有在前提条件满足的情况下,该变化才能触发; (4) postconditions 是变化操作的后件条件,在采用该变化对 OWL-S 服务描述进行修改后,新的服务描述必须满足这些后件条件。

由于变化操作的前提和后件条件相对固定,它不依赖于特定的 OWL-S 服务描述和变化参数,因此在采用变化操作对特定的演化需求进行表示时,通常忽略变化的前提和后件,而只采用 name(args) 进行表示,如 AddProcess(proc)。在 EM4CS 方法中,我们以 OWL-S 语法规则为基础,枚举抽象出针对其核心元素的基本变化集,并在演化本体中严格定义。限于篇幅本文不再赘述,相关内容可参见文献[15]。

定义 3 对于 OWL-S 服务描述而言,维护请求 (Maintenance Request, MR) 是对知识工程师组合服务演化需求的形式化表示。形式上 $MR = \langle PCL, NCL \rangle$, 其中 PCL 表示正向变化集 (Positive Change List), 代表知识工程师希望在服务描述中实施的变化操作集合, NCL 则表示逆向变化集 (Negative Change List), 代表演化过程中必须避免实现的变化操作集合。

在 EM4CS 方法中,组合服务演化需求形式化表示阶段的处理流程如图 4 所示。总体而言,本阶段包括演化需求形式化表示 (P3) 和演化需求优化 (P4) 两个子过程。其中,过程 P3 分别通过演化目标转换和演化约束转换将演化需求表中的演化需求和约束条件转化为维护请求中的 PCL 和 NCL。例如,由表 1 所列的演化需求转换所得的维护请求如表 2 所列。

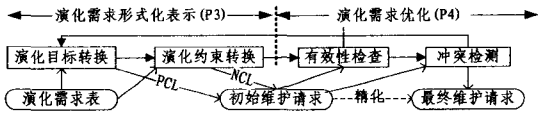


图 4 组合服务演化需求形式化表示流程

表 2 维护请求 MR

PCL:
RemovePerform(Perform_2, ChooseSeller), InsertSubCtrlCstr(Sequence_2, Sequence_1, Perform_1)
InsertProcPerform(CreditQuery, Sequence_2, NULL)
InsertProcPerform(ChooseSeller, Sequence_2, CreditQuery)
InsertSubCtrlCstr(If-Then-Else_1, Sequence_3, NULL)
MoveSubCtrlConstruct(Perform_5, Sequence_3, NULL)
InsertProcPerform(CreditEvaluate, Sequence_3, Distribution)
AddInput(CreditQuery, sellerList), AddOutput(CreditQuery, queryRes1)
AddInput(ChooseSellerByCredit, creditList), AddOutput(ChooseSellerByCredit, seller)
AddDataFlow(ProductSearch, queryRes, CreditQuery, sellerList)
AddDataFlow(CreditQuery, queryRes1, ChooseSellerByCredit, creditList)
AddDataFlow(ChooseSeller, sellerChooseRes, OrderDeliver, seller)
NCL:
RemoveSubCtrlCstr(Perform_6, If-Then-Else_1)

在得到初始维护请求后,机器将在演化本体的支持下对其进行优化。具体而言,过程 P4 包括有效性检查和冲突检测两个部分:

(1) 有效性检查。有效性检查主要分析维护请求是否具备实施的可能,其检查内容是分析 NCL 中变化操作在初始服务描述下是否已经等效实施。例如,假设服务描述 D 中不包含过程 p,如果 NCL 中包含变化 RemoveProcess(p),那么该请求就是无效的。

(2) 冲突检测。冲突检测主要以演化本体中定义的变化关系推导规则为基础,对维护请求中所包含的变化间的潜在关系进行推理,分析可能存在的冲突,并对这些冲突进行优化,提高后续处理的效率。

通过过程 P3 和 P4 的处理,知识工程师能够将原始的组合服务演化需求转化为精化的 OWL-S 服务描述维护请求,并在演化本体的支持下采用 XML 语言对维护请求进行编码,形成维护请求脚本,并以此作为后续处理的输入。

3.3 服务描述语法一致性维护阶段

在语义 Web 环境中,组合服务描述语法的正确性直接关系到该服务能否被 Agent 正确“理解”和使用,因此语法一致性维护是 EM4CS 方法中非常关键的步骤。该阶段包括语法一致性维护方案生成 (P5)、方案评价与选择 (P6) 和方案实施 (P7) 等 3 个子过程,其处理流程如图 5 所示。

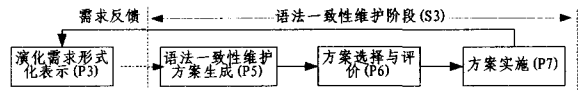


图 5 语法一致性维护阶段流程

表 3 语法一致性维护方案

MP_2:
RemovePerform(Perform_2, ChooseSeller), InsertSubCtrlCstr(Sequence_2, Sequence_1, Perform_1)
InsertProcPerform(CreditQuery, Sequence_2, NULL)
InsertProcPerform(ChooseSeller, Sequence_2, CreditQuery)
InsertSubCtrlCstr(If-Then-Else_1, Sequence_3, NULL)
MoveSubCtrlConstruct(Perform_5, Sequence_3, NULL)
InsertProcPerform(CreditEvaluate, Sequence_3, Distribution)
AddInput(CreditQuery, sellerList), AddOutput(CreditQuery, queryRes1)
AddInput(ChooseSellerByCredit, creditList), AddOutput(ChooseSellerByCredit, seller)
AddDataFlow(ProductSearch, queryRes, CreditQuery, sellerList)
AddDataFlow(CreditQuery, queryRes1, ChooseSellerByCredit, creditList)
AddDataFlow(ChooseSeller, sellerChooseRes, OrderDeliver, seller)
RemoveSubCtrlCstr(Perform_2, Sequence_1), RemoveCtrlConstruct(Perform_2)
RemoveDataFlow(ProductSearch, queryRes, ChooseSeller, sellerList)
RemoveDataFlow(ChooseSeller, sellerChooseRes, OrderDeliver, seller)

在该阶段中,过程 P5 主要依据组合服务的 OWL-S 服务描述和维护请求生成语法维护树 SMT,并通过 SMT 的搜索来获得多个既符合语法一致性约束,又最大程度上实现了维护请求的目标节点,这些目标节点回溯到起始节点的边即代表一个可行的语法一致性维护方案。过程 P6 主要按照预定义的评价指标对得到的维护方案进行定量评价,在过程 P7 中,知识工程师按照个人偏好选择最优方案对 OWL-S 服务描述实施修改。如果最终选取的维护方案中包含人工处理操作,或者 PCL 中含有未实施的变化,那么知识工程师需要依据这些情况生成新的演化需求表,并发起迭代操作,重新进行演化需求形式化表示。通过多次迭代处理,知识工程师能够获取一个完全满足初始演化需求,同时保持语法一致性的 OWL-S 服务描述。例如,经过搜索,我们可以采用表 3 所列的语法一致性维护方案对电子商务组合服务进行修改,修改

得到的新组合服务如图 6 所示。

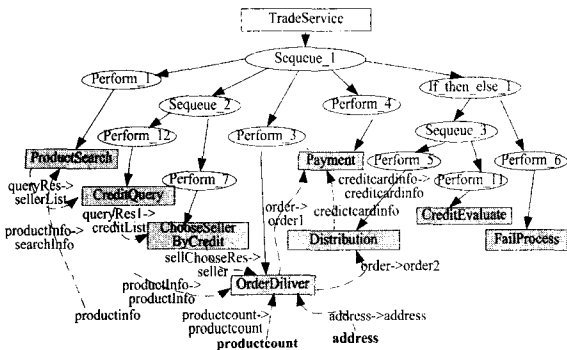


图 6 修改后的电子商务交易组合服务

3.4 服务描述语义一致性维护阶段

在 EM4CS 方法的前 3 个阶段中,知识工程师已经通过迭代处理的方式完成了服务描述修改和语法一致性维护。但是为了保持组合服务的有效性,其服务描述不仅要在语法上保持一致,还需要对其语义一致性进行分析。在 EM4CS 方法中,服务描述的语义一致性包括数据语义一致性和过程语义一致性两个方面,其中数据语义一致性对过程模型的数据流进行约束,而过程语义一致性则对组件服务的聚合逻辑进行约束。

定义 4 对于给定的 OWL-S 服务描述 D ,我们称 D 是数据语义一致的,如果其过程模型 PM 满足条件 $\forall df \in DF_{ind}, Type(df, srcOutput) \subseteq Type(df, desInput)$,其中 $Type(i) = i.parameterType$ 表示参数 i 所属的概念类型。

定义 5 对于给定的 OWL-S 服务描述 D ,我们称 D 是过程语义一致的,如果其过程模型 PM 满足如下的条件:(1)过程模型 PM 中不存在死锁的情况;(2)过程模型 PM 中任意的过程执行都是可达的;(3)过程模型 PM 在运行时不会错误终止。

采用 EM4CS 方法对服务描述的语义一致性维护流程如图 7 所示。其中,过程 P8 对服务描述的数据语义一致性进行分析,如果发现存在数据流的源、目标参数类型不包容,那么进入过程 P10,生成新的演化需求,并发起迭代操作,进入过程 P3 处理。如果过程模型符合数据语义一致性,那么进入过程 P9,对过程语义一致性进行检查。如果过程模型符合过程语义一致性,那么进入下一阶段对演化过程进行确认(P11),否则进入过程 P10,生成新的演化需求以修复过程语义一致性。

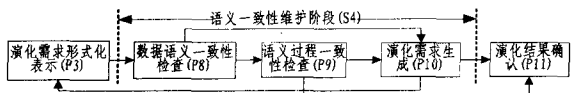


图 7 语义一致性维护阶段处理流程

在前期的工作中,我们已经对基于过程模型网的 OWL-S 服务描述语义一致性分析方法进行了研究,相关工作可参见文献[16],限于篇幅本文不再赘述。经过检测后可以发现,图 6 所示的组合服务符合数据语义一致性,其经过转化并化简的过程模型网如图 8 所示。在以可达图为工具对该 PM_{net} 进行分析后可以发现其是可靠的,因此对应的过程模型符合过程语义一致性。

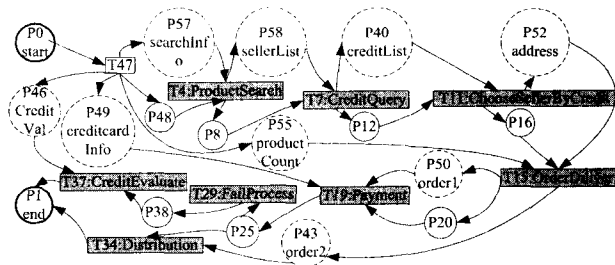


图 8 电子商务组合服务过程模型网

3.5 演化结果确认阶段

OWL-S 服务描述的内在关系复杂,对其局部的修改可能会造成全局的不一致。在 EM4CS 方法中,第 3 阶段主要辅助知识工程师处理修改后所带来的语法不一致问题,而第 4 阶段则辅助知识工程师处理维护请求实施所带来的语义不一致问题。在整个演化过程中,尽管机器辅助处理过程是客观的,但是知识工程师的决策过程是相对主观的,因此在得到语法、语义一致的 OWL-S 服务描述后,知识工程师可能仍然需要对已经实施的变化操作进行回溯处理,例如:(1)知识工程师对演化本体中所定义变化操作的实际修改效果可能理解错误,导致部分不应当执行的变化操作被批准实施;(2)在多个知识工程师协作对组合服务进行演化时,彼此间的偏好或观点可能存在差异;(3)知识工程师在第 1 阶段所提取的原始需求与现实环境可能存在偏差;等等。

在 EM4CS 方法中,演化结果确认(过程 P11)主要就是在上述情况发生时,使知识工程师能够对前两个阶段中所实施的变化进行撤销,从而提高整个演化过程的可用性。演化日志是回退操作的基本依据。由于演化日志对前两个阶段中所有实施的变化操作进行了形式化表示,因此机器可以很方便地基于这些日志对 OWL-S 服务描述进行逆向修改,直到所有指定的变化操作都已经撤销为止。如果当前的组合服务描述得到知识工程师的确认,那么这些修改操作就将反映到实际的组合服务上,从而对现实的服务运行实例产生影响。

3.6 演化结果发布与传播阶段

演化结果发布与传播阶段是 EM4CS 方法中最后的处理步骤,其主要作用在于将 OWL-S 服务描述的变化映射到现实世界中,使相关应用能够同步这些变化。具体而言,该阶段由演化结果发布(P12)和演化结果传播(P13)两个子过程组成。

表 4 基于角色的演化结果访问控制

角色	授权访问内容	权限定义方式
服务演化知识工程师	完全访问	自动
组织内知识工程师	完全访问	自动
联盟用户	已开放的服务描述变化操作	人工定义
一般用户(应用)	Profile 相关变化操作	自动

在 EM4CS 方法中,所有针对服务描述的变化操作均在演化本体的支持下在演化日志中进行了明确记录,因此通过演化日志的公开就可以实现演化结果的发布。这里,我们采用基于角色的访问控制(RBAC)思想^[15],将访问演化日志的用户划分为服务演化知识工程师、组织内知识工程师、联盟用户和一般用户 4 类角色,这 4 类角色所对应的访问权限如表 4 所列,访问控制的体系结构如图 9 所示。通过上述访问控制机制,知识工程师既能够向相关用户开放组合服务演化过程,又能够保证组织内部数据在受控的情况下共享。

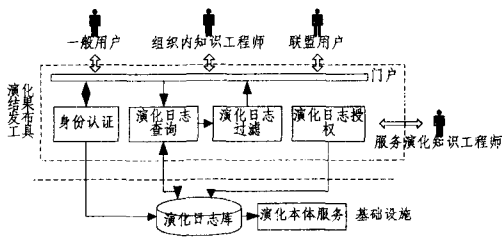


图9 EM4CS方法中演化结果发布体系结构

除了公开演化过程,使相关应用能够同步这些变化外,组合服务的当前运行实例也应当能够及时与新的组合模式同步,这就是过程P13的主要任务。OWL-S服务描述包括服务概貌(ServiceProfile)、服务模型(ServiceModel)和服务基础(ServiceGrounding)等3个方面^[16],因此在EM4CS方法中我们也对这3方面的内容进行了修改。但是,由于服务概貌和服务基础的主要作用在于服务的发布、发现和调用,而这两个部分的修改不涉及到当前运行实例的在线调整问题,因此在过程P13中仅对运行实例如何适应服务模型的调整进行研究。

在EM4CS方法中,我们提出3种迁移策略对当前的组合服务运行实例进行调整:

(1)终止策略。在组合服务演化过程中,如果服务过程模型发生调整,那么当前正在运行的服务实例将终止执行。终止策略是最为简单的迁移策略,它能够强制结束当前服务运行实例的执行,并触发重新调用的行为,使得修改的过程模型生效。终止策略所存在的问题在于可能会降低组织服务的QoS,从而使组织信誉产生影响。

(2)并发处理策略。并发处理策略的基本思想是,当前运行的服务实例将继续按照组合服务原有过程模型执行,而新运行的服务实例则选用新的过程模型,使得新旧过程模型同时存在。该策略的优点在于不会中断任何服务调用,但问题在于部分运行实例的流程可能难以及时与新的过程模型同步。

(3)即时迁移策略。即时迁移策略的基本思想是依据服务运行实例的实际运行状态,为运行实例生成不同的临时过程模型,使得当前服务运行实例能够平滑过渡到新的过程模型。在采用即时迁移策略进行处理时,既没有强制终止旧过程模型的执行,也照顾到快速应用新过程模型的需求,因此可以看作是对前两种策略进行折衷的处理方式。即时迁移策略的示例如图10所示。

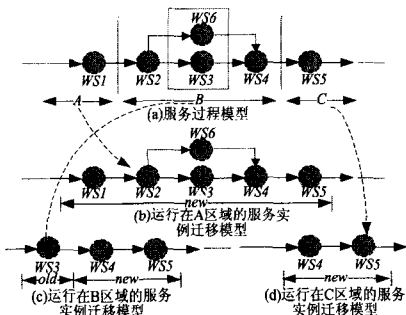


图10 过程模型即时迁移策略示例

在EM4CS方法中,演化结果传播是组合服务演化周期的最后一个处理过程,在服务运行实例处理完成后,当前的组合服务演化过程就完整结束。通过上述6阶段的迭代处理,

我们使组合服务在面对外部变化时,能够基于EM4CS方法对OWL-S服务描述进行完整的演化操作,使得服务描述调整的同时实现组合服务功能的调整。同时,基于EM4CS方法对组合服务进行演化是一个开放的过程,在一次演化周期结束后知识工程师应当持续对外部环境进行监控,在需要的时候重新启动组合服务演化过程,从而实现服务的可持续发展。

4 组合服务演化支撑系统 ESS4CS

在动态变化的语义Web环境下,组合服务演化是组合服务生命周期中不可或缺的环节,也是使服务与外部环境和用户需求相适应的基本方法。由于组合服务演化是一个复杂的过程,其中涉及到大量的逻辑分析和计算,如语法结构分析和聚合逻辑分析等,因此在面对大规模组合服务时,完全由知识工程师人工完成服务演化工作是不现实的,需要机器辅助完成。鉴于此,我们设计实现了遵循EM4CS方法的组合服务演化支撑系统ESS4CS(Evolution Support System for Composite Service),该系统的体系结构如图11所示。



图11 ESS4CS系统体系结构

总体而言,ESS4CS系统由运行环境CSERE和支撑工具集CSETS组成,其中CSERE是ESS4CS系统的后台构件集合,为组合服务演化中的各项任务提供基础支持。在系统组成上,运行环境包括基础构件集和基础设施,前者包含了组合服务演化中所必须具备的核心构件,后者则是组合服务演化过程中的基本构件,包括组合服务演化本体服务和演化日志库两个部分。CSETS为组合服务演化过程中的关键步骤提供辅助支持,在组成上CSETS所包含的工具与EM4CS方法中的演化过程相对应,其工作界面如图12所示。

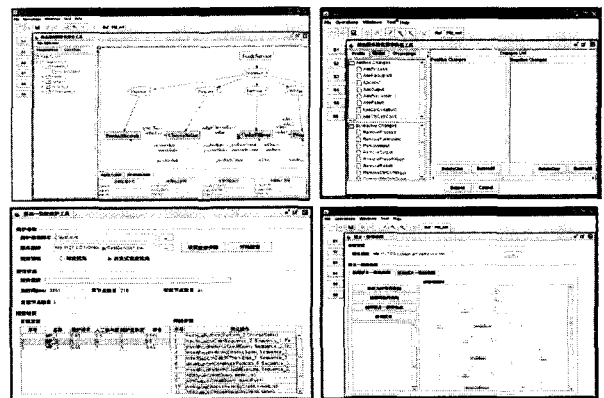


图12 ESS4CS系统工作界面

结束语 随着Web服务和SOA架构应用的深入,组合服务如何适应环境变化成为一个非常重要的问题。目前的研究着重于提高组合服务运行实例的动态适应性,而对服务组合模式进行敏捷、有效的调整以适应外部变化的研究较少。针对这个问题,本文从系统工程的角度触发,提出了一个语义

(下转第151页)

- [7] 刘宏伟. 非齐次柏松过程类软件可靠性增长型研究[D]. 哈尔滨: 哈尔滨工业大学, 2004
- [8] 蔡开元, 白成刚, 钟小军. 构件软件系统的可靠型评估模型简介[J]. 西南交通大学学报, 2003, 37(6): 551-554
- [9] Kanoun K, Sabourin T. Software dependability of a telephone switching system[A]// The 17th International Symposium on Fault-Tolerant Computing[C]. Pittsburgh, Pennsylvania, 1987
- [10] Gokhale S. An analytical approach to architecture based software reliability prediction[A]// The Third International Computer Performance and Dependability Symposium[C]. Durham, North Carolina, 1998
- [11] Everett W. Software component reliability analysis[A]// The Symposium on Application-specific Systems and Software Engineering echnology[C]. Richardson, Texas, 1999
- [12] 朱经纷, 徐拾义. 软件可靠性综合模型的分析研究[J]. 计算机科学, 2009, 36(4): 181-184
- [13] 陈俊文, 谷建华, 等. 一种改进的基于架构的软件可靠性模型[J]. 计算机工程与应用, 2009, 45(33): 60-63

(上接第 143 页)

Web 环境下组合服务结构化演化方法 EM4CS, 对 EM4CS 方法的流程进行了具体说明, 并给出了遵循 EM4CS 方法的组合服务演化支撑系统 ESS4CS 的设计与实现。通过 EM4CS 方法的指导和 ESS4CS 系统的辅助支持, 知识工程师能够快速地从问题域中抽取组合服务演化需求, 并在机器辅助下对 OWL-S 服务描述进行修改, 同时通过多次迭代处理完成服务描述语法和语义一致性检测与修复。在演化过程经过确认后, 知识工程师能够将演化结果在受控的前提下发布, 并使组合服务运行实例尽快同步这些变化, 保持全局应用的一致性。在下一步工作中, 我们将对 EM4CS 方法和 ESS4CS 系统进行进一步研究和完善, 具体包括如下两个方面:

(1) 本文所提出的 EM4CS 方法以采用 OWL-S 语言描述的组合 Web 服务为研究对象, 未来需要进一步研究如何使 EM4CS 方法支持采用其他语言描述的组合服务, 如 WSMO^[17] 和 SWSF^[18] 等;

(2) 目前 ESS4CS 系统能够支持服务组合模式演化, 但是无法实现服务运行实例依据演化结果在线调整。在未来的工作中, 我们将进一步对支持演化的语义 Web 执行环境进行研究, 使组合服务运行实例能够快速同步组合模式演化中所发生的变化。

参 考 文 献

- [1] 喻坚, 韩燕波. 面向服务的计算—原理和应用[M]. 北京: 清华大学出版社, 2006
- [2] Haller A, Cimpian E, Mocan A, et al. WSMX - A Semantic Service-Oriented Architecture[C]// Proceedings of the IEEE International Conference on Web Services(ICWS 2005). IEEE Computer Society, Washington, DC, USA, 2005: 321-328
- [3] Norton B, Pedrinaci C, Domingue J, et al. Semantic Execution Environments for Semantics-Enabled SOA [J]. Information Technology, 2008, 50(2): 118-121
- [4] Ilkaev D. Recent Trends in Semantic SOA. 2007. [OL]. <http://www.stickyminds.com/sitewide.asp?Function=edetail&ObjectType=ART&ObjectId=12612&tth=DYN&tt=sitemap&iDyn=2>
- [5] McIlraith S A, Son T C, Zeng Hong-lei. Semantic Web Services [J]. IEEE Intelligent Systems, 2001
- [6] 刘飞. SOA 中国路线图. 2007. [OL]. <http://www.ufida.com.cn/subject/200705xxhpl/pdf2/soa.pdf>
- [7] Karastoyanova D, Buchmann A. ReFFlow: a model and generic approach to flexibility of web service compositions[C]// iiWAS' 2004 - The sixth International Conference on Information Integration and Web-based Applications Services. Jakarta, Indonesia, September 2004
- [8] Tao A T, Yang Jian. Supporting Differentiated Services With Configurable Business Processes[C]// IEEE International Conference on Web Services, 2007. 2007: 1088-1095
- [9] Naccache H, Gannod G C. A Self-Healing Framework for Web Services[C]// IEEE International Conference on Web Services (ICWS 2007). 2007: 398-345
- [10] Chaffle G, Dasgupta K, Kumar A, et al. Adaptation in Web Service Composition and Execution[C]// Proceedings of the IEEE International Conference on Web Services (ICWS 2006). 2006: 549-557
- [11] Colombo M, Nitto E D, Mauri M. SCENE: A Service Composition Execution Environment Supporting Dynamic Changes Disciplined Through Rules[C]// Service-Oriented Computing ICSOC 2006. Springer: Berlin / Heidelberg, 2006: 191-202
- [12] Blanchet W, Stroulia E, Elio R. Supporting Adaptive Web-Service Orchestration with an Agent Conversation Framework[C]// Proceedings of the IEEE International Conference on Web Services (ICWS 2006). 2006: 541-549
- [13] Guarino N. Semantic Matching, Formal Ontological Distinctions for Information Organization, Extraction, and Integration[C]// International Summer School on Information Extraction; A Multidisciplinary Approach to an Emerging Information Technology. 1997: 139-170
- [14] Kalfoglou Y, Hu B, Reynolds D, et al. Semantic integration technologies[C]// Proceedings of the 6th Month Deliverable. University of Southampton and HP Labs. ECS e-Prints Report # 10842, 2005
- [15] Bao Ai-hua, Yao Li, Yuan Jin-ping, et al. Approach to the Formal Representation of OWL-S Ontology Maintenance Requirements[C]// the Proceedings of the Ninth International Conference on Web-Age Information Management (WAIM 2008). Zhangjiajie, Hunan, China, July 2008: 56-61
- [16] 鲍爱华, 袁金平, 姚莉, 等. 基于扩展着色 Petri 网的 Web 服务本体描述语言过程语义分析方法[J]. 计算机集成制造系统, 2008 (09): 1856-1864
- [17] WSMO. Web Service Modeling Ontology (WSMO) - An Ontology for Semantic Web Services. 2005 6-9 [OL]. http://www.w3.org/2005/04/FSWS/Submissions/1/wsmo_position_paper.html#omg
- [18] Battle S, Bernstein A, Boley H, et al. Semantic Web Services Ontology (SWSO). 2005. [OL]. <http://www.damll.org/services/swsf/1.0/swso/>