

基于进程代数的服务业务流程价值分析

马 超 徐晓飞 王忠杰

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 服务业务流程模型的质量对服务系统的质量会产生直接影响,而衡量模型质量的标准除了模型本身的一致性、无歧义性等性质外,更为重要的是其能否充分满足顾客的服务需求.在服务工程中,服务价值被认为是各服务参与者期望从服务获取的最终目标,采用价值模型从功能性和非功能性两方面对顾客需求进行形式化描述.通过对 Pi 演算扩展动作质量约束和价值实现程度计算函数,提出价值进程代数 VPA,用于形式化表达和验证服务业务流程模型.利用语义等价匹配算法和弱互模拟理论,提出验证目标价值的可实现性和实现完整性的模型分析方法,分析目标价值的实现程度,进而定量计算服务业务流程模型对顾客需求的满足情况,为后续面向价值的服务模型优化提供依据.

关键词 服务业务流程模型;价值模型;目标价值;价值进程代数;价值状态图

中图法分类号 TP311 **DOI 号:** 10.3724/SP.J.1016.2010.02177

A Process Algebra Based Method for Value Analysis on Service Process Model

MA Chao XU Xiao-Fei WANG Zhong-Jie

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

Abstract As the quality of service models determines the quality of service systems to a great extent, and besides the consistency, unambiguity, etc., of model itself, the fulfillment of customer's service requirements is a more important standard to measure whether the quality of service models is good or not. In service engineering, customer's service requirements are difficult to express accurately and comprehensively. By utilizing value models, this paper describes customer's service requirements from both functional and non-functional aspects. By importing the quality constrain of action and an implementation degrees of value function into Pi -calculus, VPA (value process algebra) is proposed, which is used to formally describe and validate service business proves model. By utilizing semantic equivalence matching algorithm and weak bisimulation theory, this paper verifies realizability, complete realizability of goal value, and analyzes the implementation degrees of goal value, and quantitatively checkes whether and to what degree customer's service requirements are met by service' business process model, so as to provide references for value-oriented service model optimization.

Keywords service business process model; value model; goal value; VPA; value state graph

1 引 言

服务工程是设计与开发服务系统的过程和方

法,被用来将服务需求转变成一个可执行的服务系统.服务建模在服务工程中是一个关键步骤^[1],其设计的服务模型的质量对服务系统的质量产生直接影响,而衡量模型质量的标准除了模型本身的一致性、

收稿日期:2010-06-08;最终修改稿收到日期:2010-08-27.本课题得到国家自然科学基金(61033005,60803091,70971029)、国家“八六三”高技术研究发展计划项目基金(2009AA04Z151)资助.马 超,男,1983 年生,博士研究生,主要研究兴趣为价值知觉的服务工程方法论. E-mail: machao8396@163.com.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.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.

无歧义性等性质外,更为重要的是其能否充分满足顾客的服务需求.如何验证服务业务流程模型能否满足顾客的服务需求是服务模型验证领域的一个重要问题.要解决这个问题,将涉及两方面的关键技术:(1)准确、全面地表达顾客的服务需求;(2)形式化表达服务业务流程模型,进行服务模型分析.

在设计服务系统时,顾客的服务需求具有复杂化、个性化的特点,难以全面、准确地表达.一般地,用户需求的表达包括功能性和非功能性(如质量角度)两个方面,例如,文献[2]为了构建高质量的业务模型,分别从功能性和质量两个方面来表达用户的需求.但在服务系统中,相对于低层的具体业务流程能够实现哪些业务功能以及实现的功能能够满足哪些质量约束,顾客更加关注的是高层次的服务价值.顾客与服务提供者、使能者协同生产,共同创造价值,价值是服务系统的“产出”或“结果”.因此,服务价值被认为是各服务参与者期望从服务获取的最终目标,其包含:顾客最终所获得的全局目标价值;顾客在参与服务的过程中所感受的局部目标价值.为此,本文利用文献[3]中提到的多层次图形化服务价值模型去分析、表达顾客的服务需求.利用价值模型,本文可以对顾客的服务需求逐步地进行分析,将顾客的服务需求从全局到局部,从功能性到非功能性全面、准确的表达出来,将顾客的服务需求表达成目标价值的形式,利用目标价值的属性信息,描述顾客在功能性方面的服务需求;利用目标价值的价值指标集合,从非功能性方面对顾客的服务需求进行定量的约束.服务模型能否满足顾客服务需求的问题实际上即是服务模型能否支持目标价值实现的问题.

实际上,在高层上进行业务目标/价值建模的服务参与者与在低层上进行细节功能建模的参与者是不同的.高层的建模工作主要由高级业务分析师和咨询专家进行,而低层的建模工作主要由服务设计者进行.虽然前者的输出将会被用作后者的输入,但是两者之间的理解差距将导致后者的输出(服务模型)与前者的输出(服务价值)存在不一致性.此外,由于时间上的限制,两个阶段常常是并行执行的,而不是按照一定的交互顺序执行.这使得将服务价值和服务模型连接起来,并且验证服务模型能否充分地支持价值模型中各项目标价值的实现是很有必要的.

本文从定性和定量两个角度进行面向价值的服务模型分析.定性分析主要包括两方面:(1)验证服

务模型能否满足目标价值的可实现性;(2)验证服务模型能否满足目标价值的实现完整性.前者是验证目标价值实现载体的结束状态能否被服务模型支持实现;后者是从服务模型中找出一组能够支持目标价值实现载体从开始状态转移到结束状态的服务行为.在定量分析时,利用得到的这组服务行为,同时考虑服务行为之间的拓扑结构,进行价值指标值的计算,将得到的价值指标值与顾客对价值指标的期望约束进行对比,能够得到目标价值在服务业务流程模型的支持下的实现程度.

为了实现上述分析过程,需要将服务模型进行形式化处理,使得服务模型能够进行形式化演算、验证,并且能够支持目标价值实现程度的计算,因此,需要引入进程代数. Pi 演算是一种经典的进程代数^[4-5],经常被用于形式化描述和验证并发、动态变化的系统.文献[6]利用 Pi 演算去形式化描述和建模服务及其组合,并对建立的组合模型的正确性以及该组合模型能否满足需求进行验证.文献[7]利用 Pi 演算进行跨组织 workflow 建模,并利用其弱互模拟理论进行组织之间等价性的验证.在设计服务系统时,针对顾客经常变化和个性化的服务需求,建立的服务业务流程模型可能需要不断地进行调整变化, Pi 演算在描述这类问题上具有一定的优势,可以被用来形式化表达服务业务流程模型,以支持服务业务流程模型能够满足顾客在功能性需求方面的验证.但是,本文不仅需要去验证服务业务模型对顾客需求的满足情况,也需要在非功能性需求方面进行验证,即验证服务业务模型对价值实现程度的支持情况.而非功能性(时间、资源、功耗或费用等)进行建模以支持优化和决策正越来越受到关注.例如,文献[8]为了获得成本优化的服务组合方案,在 CCS 基础上通过扩展价格函数得到了价格进程代数 PPA,支持服务组合过程中的价格与成本的计算.文献[9]对 Pi 演算进行了成本扩展,用于解决智能体领域中的资源受限的计算问题.经典 Pi 演算只能支持对功能性需求的验证,缺乏对非功能性需求进行验证的能力,即支持计算价值实现程度的能力.为此,在 Pi 演算的基础上,引入动作质量约束和价值实现程度计算函数,本文提出了价值进程代数 VPA,用于对服务业务流程模型进行形式化表达以及进行面向价值的服务模型分析.

为了更加清楚地探讨面向价值的模型分析方法,本文以海运物流出口服务为案例来证明该方法的有效性.在海运物流出口服务中,货主计划将货物

从仓库运输到目的地,其需要从货代处获航线信息并选择自己信任的货代为其提供代理服务.货代接到委托之后,分别向船公司订舱,向场站租赁集装箱,向车队请求陆上运输,向海关申请报关.车队到场站取空箱,然后到货主仓库装货,打铅封,将货物从仓库运输到场站.场站将货物暂存到堆存区,并且在限定时间内将货物运输到码头.海关检验货物后,批准货物出口,船公司将货物从码头运输到目的地.

本文第2节介绍如何用价值模型去表达顾客的服务需求;第3节介绍VPA的语法语义、操作语义和弱互模拟理论以及如何利用VPA形式化表达服务业务流程模型;第4节介绍如何验证目标价值的可实现性和实现完整性,分析目标价值的实现程度,进而定量计算服务业务流程模型对顾客需求的满足情况;第5节,通过海运物流出口服务的案例分析证明上述分析方法的有效性;最后进行全文总结并说明下一步的工作.

2 基于价值模型的服务需求描述

如前所述,可以利用价值模型准确、全面地描述顾客的服务需求.首先,利用价值模型去分析顾客对服务系统的全局功能需求,将全局功能需求表示成顾客期望获得的全局目标价值.全局目标价值的表达形式为 $v = (Id, Name, Type, Producer, Receiver, Realization-Carrier, Start-State, End-State)$, 其中 Id 表示该项价值在价值模型中的唯一标识; $Name$ 表示该项价值描述的服务功能需求的语义信息; $Type$ 该项价值属于的类型,价值类型包括金融类型价值 FIV 、事物状态改善类型价值 SIV 、资源使用类 RUV 等 9 种,详细内容可参考文献[10]; $Producer$ 表示提供该项价值的组织或角色; $Receiver$ 表示获得该项价值的组织或角色; $Realization-Carrier$ 、 $Start-State$ 和 $End-State$ 3 个属性一起表示该项价值的实现情况, $Realization-Carrier$ 从 $Start-State$ 到 $End-State$ 整个状态变化的过程即是一项价值被实现的过程.从价值的分类概念,能够知道 $Realization-Carrier$ 可能是某个“事物”或者“参与者的状态”.利用上述这些全局目标价值的属性信息,能够从功能性方面对顾客的全局服务需求准确地进行表达.

在非功能性方面,通过对全局目标价值的指标集合中每一项指标提出期望约束,能够对顾客的非功能性方面的服务需求进行定量的描述.不同价值

类型的价值其指标集合是不同的,表1给出了几类价值的指标集合的实例.

表1 几类价值的指标集合

价值类型	收益指标	成本指标
FIV	金额	金钱、人员、工具
SIV	状态改变时间、准时性、非期望状态变化程度	金钱、时间、精力与体力
INV	及时性、准确率、完整性	金钱、时间、精力与体力
RUV	响应时间、可靠性、资源可用性	金钱、时间、精力与体力

顾客对价值指标提出期望约束,即非功能性方面的服务需求,可以用如下形式进行表达.

(1) 基本型约束

它是最基本的约束模式,用一个二元组表示: $Basic = (Value, Constraint)$. 其中 $Value$ 是价值集合; $Constraint$ 是约束条件,它是一个三元组: $Constraint = (Indicator, Relation, Value)$, 其中 $Indicator$ 是指标名称, $Relation$ 是算术关系, $Value$ 是指标值.根据约束范围的不同,基本型约束可以分为以下3种类型:

全局约束.对价值模型中的全局目标价值的约束,例如, $Basic(\{GV_i\}, (rt, \leq, 2h))$ 表示全局目标价值 GV_i 的价值指标“响应时间 rt ”必须小于 $2h$;

局部约束.对价值模型中若干局部目标价值的同一价值指标的约束, $Value$ 为该约束条件所对应的所有价值的集合,例如, $Basic(\{SGV_i, SGV_j\}, (rt, \leq, 10h))$, 表示局部目标价值 SGV_i 和 SGV_j 的价值指标“响应时间 rt ”之和必须小于 $10h$;

单一约束.对价值模型中单个价值的单一指标的约束, $|Value| = 1$. 例如 $Basic(\{SGV_i\}, (rep, \geq, 80\%))$ 表示局部目标价值 SGV_i 的价值指标“可靠性 rep ”必须大于 80% .

(2) 比较型约束

从价值模型中两个及多个价值的指标之间关系的角度进行的约束,它是一个三元组: $Compare = (Value, Indicator, Relation)$, 其中 $Value$ 是价值集合, $Indicator$ 是指标名称, $Relation$ 是算术关系.例如, $Compare = (\{SGV_i, SGV_j\}, ra, =)$ 表示局部目标价值 SGV_i 和 SGV_j 的价值指标“资源可用性 ra ”相等.

(3) 规则型约束

它是用来表达因果语义的约束,用一个二元组表示: $Rule = (Pre, Post)$. 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 $Post=(OB, \text{and}, OC)$. 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. 例如, $Rule(\{(\{SGV_i\}, (rt, >, 3h))\}, \text{and}, \emptyset), (\{(\{SGV_j\}, (ra, >, 80\%))\}, \text{and}, \emptyset)$ 表示如果局部目标价值 SGV_i 的价值指标“响应时间 rt ”大于项 3h, 那么局部目标价值 SGV_j 的价值指标“资源可用性 ra ”大于 80%.

至此, 利用目标价值的属性信息和指标集合的期望约束, 分别从功能和非功能两方面对顾客的全局服务需求进行了表达. 接着, 利用价值模型去分析“为了实现顾客期望的全局目标价值, 哪些局部目标价值需要被实现”, 这些局部目标价值描述了顾客在参与服务的过程中, 对服务业务流程中的某些局部业务的功能和非功能性方面的需求. 局部目标价值的属性信息和指标集合的期望约束的表达方式同全局目标价值是一样的. 最后, 本文利用价值模型去分析目标价值之间的依赖关系, 在此处, 本文重点关注

目标价值之间的组合依赖关系, 利用价值模型去分析全局目标价值是如何被局部目标价值分解, 然后逐步实现的.

下面以海运物流出口服务为例来说明上述过程. 首先利用价值模型, 将货主对服务系统的全局需求“计划将货主从仓库运输到目的地”表达成全局目标价值 GV_1 . 但是 GV_1 粒度较大, 难以被服务系统直接实现. 接着, 利用价值模型去分析业务流程, 找出 GV_1 可以通过哪些局部目标价值之间进行交换与传递来逐步实现, 通过分析, 可以得到一组局部目标价值 $\{SGV_1, SGV_2, \dots, SGV_{10}\}$, 详情见表 2. 最后, 利用价值模型, 分析目标价值之间可能存在的组合依赖关系, 找出大粒度的全局目标价值是如何被分解成一系列小粒度的局部目标价值的过程, 结果如图 1 所示, 全局目标价值 GV_1 被分解成局部目标价值 SGV_1 和 SGV_2 . 进一步地, SGV_2 又被分解为更细粒度的 $SGV_3, SGV_4, \dots, SGV_{10}$. 此外, 表 3 给出了局部目标价值 SGV_3 的指标集合的期望约束.

表 2 海运物流出口服务中的几项目标价值

<i>Id</i>	<i>Name</i>	<i>Type</i>	<i>Producer</i>	<i>Receiver</i>	<i>Realization-Carrier</i>	<i>Start-State</i>	<i>End-State</i>
GV_1	货物运输	SIV	服务系统	货主	货物的物理位置	货主仓库	目的地
SGV_1	提供航线信息	INV	货代	货主	航线信息	货主请求	货主已获得
SGV_2	代理运输	RUV	货代	货主	货代员工	货代所有	货主聘请
SGV_3	预定舱位	RUV	船公司	货代	舱位使用权	货代请求	货代已获得
SGV_4	租赁集装箱	RUV	场站	货代	集装箱使用权	货代请求	货代已获得
SGV_5	陆上运输	SIV	车队	货代	货物的物理位置	货主仓库	场站
SGV_6	装卸集装箱	RUV	场站	货代	场站员工	站长所有	货代聘请
SGV_7	集装箱堆存	RUV	场站	货代	堆存区使用权	货代请求	货代已获得
SGV_8	报关	SIV	海关	货代	货物出口权	货代未获得	货代已获得
SGV_9	货物集港	SIV	场站	货代	货物的物理位置	场站	码头
SGV_{10}	海上运输	SIV	船公司	货代	货物的物理位置	码头	目的地

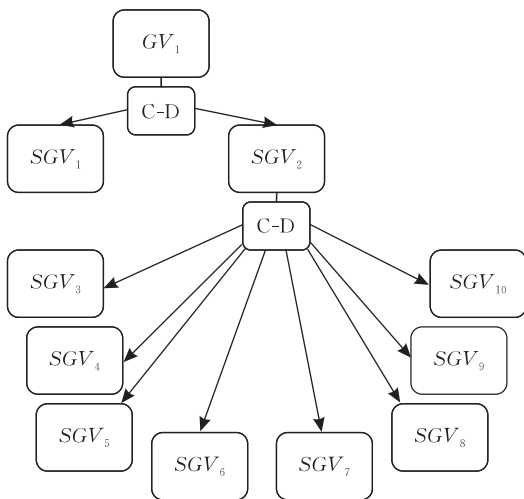


图 1 目标价值之间的组合依赖关系

表 3 SGV_3 的指标集合的期望约束

指标名称	指标类型	期望约束
响应时间	Benefit	$Basic(\{SGV_3\}, (rt, \leq, 2h))$;
		$Rule(\{(\{SGV_3\}, (rt, >, 1.5h))\}, \text{and}, \emptyset), (\{(\{SGV_5\}, (rt, <, 2h))\}, \text{and}, \emptyset)$;
可靠性	Benefit	$Basic(\{SGV_3\}, (rep, \geq, 80\%))$;
		$Basic(\{SGV_3, SGV_4\}, (rep, \geq, 85\%))$;
资源可用性	Benefit	$Basic(\{SGV_3\}, (ra, \geq, 95\%))$;
金钱	Cost	$Basic(\{SGV_3\}, (mc, \leq, 80 \text{元}))$;
时间	Cost	$Basic(\{SGV_3\}, (tc, \leq, 50\%))$;
精力与体力	Cost	$Basic(\{SGV_3\}, (epc, \leq, 50\%))$;

从上述例子中可以看到, 利用价值模型可以对顾客的服务需求逐步地进行分析, 将顾客的服务需求从全局到局部, 从功能性到非功能性全面、准确地表达出来. 服务需求被描述成目标价值的形式, 这些目标价值即是服务模型进行分析和优化的目标.

3 基于 VPA 的服务业务流程模型形式化表达

3.1 价值进程代数 VPA

Pi 演算是一种能够形式化表达和验证并发、动态变化系统的进程代数,其基本计算实体为名字和进程,进程之间的通信是通过传递名字来实现的,名字可以表示通道名、变量、值、动作等实体。Pi 演算可以用来形式化服务业务流程模型,以帮助模型设计者从功能方面去分析、验证模型。但是,Pi 演算缺乏在非功能性方面对服务业务流程模型表达的能力,不能支持价值的实现程度的计算。为此,在 Pi 演算的基础之上,通过对动作增加表示其非功能性方面特性的质量约束和对状态增加价值实现程度计算函数,使得 VPA 不仅能够从功能性方面,还能从非功能性方面表达和验证服务业务过程模型。

在 VPA 中,质量约束被用来描述动作在各方面的性能,例如:执行时间、可靠性、准确性等。动作本身可以被用来表达某种功能,本文利用质量约束对动作进行限制,将动作变为质量约束动作,扩展了动作在非功能性方面的表达能力。进程代数的动作执行会引起状态的变迁,本文为 VPA 的状态赋予价值。价值被用来表达顾客在功能性和非功能性方面的服务需求。在 VPA 中,系统达到某个价值状态是执行一系列质量约束动作的结果。由于进程代数可以进行演算,那么带有质量约束的进程演算可以得到价值的实现程度。在功能性方面,通过这一系列动作描述的功能,VPA 可以用于定性分析价值状态描述的功能性方面的顾客需求能否被服务业务流程模型支持实现。在非功能性方面,在进程表达式的状态随着质量约束动作执行而不断发生变化的过程中,为各类价值指标的价值实现程度计算函数确定具体的计算公式,VPA 可以用于定量地分析价值被服务业务流程模型支持实现的程度。

设 $N\{x, y, z, \dots\}$ 为无限名字集;进程标识符用 A, B, C 等大写大写字母表示;进程表达式用 P, Q, R 等大写大写字母表示,下面,本文给出 VPA 的语法定义。

定义 1. VPA 的语法定义。

- (1) 和式. $P+Q$, 选择执行 P 或 Q , 使用 0 表达空进程, 即不活动的进程;
- (2) 并行表达式. $P|Q$, 并行执行 P 和 Q ;
- (3) 匹配表达式. $[x=y]P$, 当 x 和 y 是同一通道名时, 其行为与 P 相同;

(4) 限制表达式. $(x)P$, P 在新通道 x 上的外部动作被禁止, 但 P 通过通道 x 的内部通信是允许的, 它们都成了 τ 动作;

(5) 重复表达式. $!P$, P 的无穷个复制;

(6) 进程标识符. $A(y_1, y_2, \dots, y_n)$, $A(x_1, x_2, \dots, x_n) ::= P$, 其中 x_1, x_2, \dots, x_n 是 P 中的自由名;

(7) 前缀表达式.

$(\bar{y}\langle x \rangle, q(\bar{y}\langle x \rangle))P$, $(y\langle x \rangle, q(y\langle x \rangle))P$, $(\tau, q(\tau))P$, 其中 $Act = y\langle x \rangle \cup \bar{y}\langle x \rangle \cup \tau$, $\alpha \in Act$, $q(\alpha)$ 表示动作质量约束, $Q = \{q_1, q_2, \dots, q_n\}$ 为动作 α 的质量参数, $q(\alpha) = g(Q)$, $g(q_i)$ 可能是一个确定的值, 更多的时候其取值是一个范围。

VPA 的定义如下:

$$P ::= 0 \mid P+Q \mid P|Q \mid (\bar{y}\langle x \rangle, q(\bar{y}\langle x \rangle))P \mid (y\langle x \rangle, q(y\langle x \rangle))P \mid [y=z]P \mid (x)P \mid !P \mid A(y_1, y_2, \dots, y_n).$$

定义 2. VPA 操作语义。

VPA 操作语义可以利用一个质量约束标签迁移系统 $QLTS = (S, \Delta, \rightarrow)$ 表示, 其中:

① $S = \{(s, f(s))\}$ 是价值状态集合, 其中 s 用来描述价值状态能够实现的功能, $f(s)$ 是在此价值状态下, 价值被支持实现的程度, s_0 被用来表示初始状态, 即没有经过任何动作执行的系统状态。

② $\Delta = A \cup \bar{A} \cup (\tau, q(\tau))$, 表示质量约束动作标签集, 其中 $(\tau, q(\tau))$ 是进程的內部质量约束动作, 外部不可见; $A\{(y\langle x \rangle, q(y\langle x \rangle)) \mid x, y \in \Sigma\}$, 在端口 y 上输入 z , 然后利用 z 替换 x , 表示质量约束接收动作集; $\bar{A}\{(\bar{y}\langle x \rangle, q(\bar{y}\langle x \rangle)) \mid x, y \in \Sigma\}$, 在端口 y 上输出 x 表示质量约束发送动作集;

③ $\rightarrow \in S \times \Delta \times S$ 为质量约束迁移关系, 定义为

$$(s_i, f(s_i)) \xrightarrow{(\alpha, q(\alpha))} (s_{i+1}, f(s_{i+1})) \text{ iff } \begin{cases} s_i \xrightarrow{\alpha} s_{i+1} \\ f(s_{i+1}) = h(f(s_i), q(\alpha)) \end{cases},$$

$h(x)$ 为价值实现程度计算函数, 针对不同的应用领域、不同的价值指标, 其具体计算的形式是不同的。

(a) ACT:

$$\xrightarrow{\bar{y}\langle x \rangle, q(\bar{y}\langle x \rangle)}, \text{Value}(P') = h(\text{Value}(P), q(\alpha)),$$

$$\xrightarrow{\alpha P \xrightarrow{(\alpha, q(\alpha))} P'}$$

ACT 规则表明无任何条件限制的动作执行之后, 价值的实现程度需要利用 $h(x)$ 对动作执行前的价值实现程度和动作的质量约束进行计算得到。

(b) SUM:

$$\frac{P \xrightarrow{(\alpha, q(\alpha))} P'}{P+Q \xrightarrow{(\alpha, q(\alpha))} P'} \rightarrow, Value(P') = h(Value(P+Q), q(\alpha)),$$

SUM 规则表明对 P, Q 的不确定选择, 若选择 P (或 Q), 动作执行之后的价值的实现程度需要利用 $h(x)$ 对价值状态 P (或 Q) 的实现程度和动作的质量约束进行计算得到.

(c) PAR:

$$\frac{P \xrightarrow{(\alpha, q(\alpha))} P'}{P|Q \xrightarrow{(\alpha, q(\alpha))} P'|Q} \rightarrow, bn((\alpha, q(\alpha))) \cap fn(Q) = \emptyset,$$

$$Value(P'|Q) = h(Value(P|Q), q(\alpha)).$$

(d) RES:

$$\frac{P \xrightarrow{(\alpha, q(\alpha))} P'}{(x)P \xrightarrow{(\alpha, q(\alpha))} (x)P'} \rightarrow, x \notin n(\alpha),$$

$$Value((x)P') = h(Value((x)P), q(\alpha)).$$

VPA 只对 P_i 演算扩展了计算价值实现程度的能力, 并没有对 P_i 演算的基本性质进行更改, 所以 VPA 继承了 P_i 演算的弱模拟特性. 在 VPA 中, P 弱模拟 Q , 其含义是 P 的外部动作与 Q 的外部动作观察等价, 即其表现出的外部行为是一致的.

定义 3(弱模拟). 如果关系 R 是弱模拟, 那么对所有 $(P, Q) \in R$, 满足:

- ① $P \xrightarrow{\tau} P'$, 存在 Q' 满足 $Q \xrightarrow{\tau'} Q'$, 且 $(P', Q') \in R$;
- ② $P \xrightarrow{\alpha} P'$, 存在 Q' 满足 $Q \xrightarrow{\tau' \alpha \tau'} Q'$, 且 $(P', Q') \in R$.

3.2 基于 VPA 的服务业务流程模型

针对大型、复杂交互的服务系统, 在建立其服务业务流程模型时, 首先将整个服务业务流程分解为若干粒度适中的服务任务, 然后针对各个服务任务分别建立服务业务流程模型. 在建立服务业务流程模型时, 服务模型设计者不仅需要给出流程中服务行为的描述以及服务行为之间关系描述, 尤其需要给出模型中服务行为在质量方面的约束.

为了能够更加准确地描述服务业务流程, 本文对 UML 视图进行了扩展, 以扩展的 UML 时序图来描述服务任务. 在 UML 时序图中, 任意两个对象之间通过带箭头的直线连接起来, 用以表示对象之间传递的消息, 但是, 为了描述任务中, 对象之间的行为交互情况, 本文对带箭头直线的语义进行了扩展, 本文用其表示一个对象对另外一个对象产生的行为, 以及在行为的执行过程中, 所作用的对象. 例如, 针对海运物流出口服务, 将整体服务流程分解为

预定舱位、陆上运输、报关等服务任务, 然后利用扩展的 UML 时序图, 分别对这些服务任务进行服务业务流程建模. 利用扩展的 UML 时序图, 本文对服务任务“预定舱位”的服务业务流程进行描述, 得到了该任务的服务流程模型, 如图 2 所示.

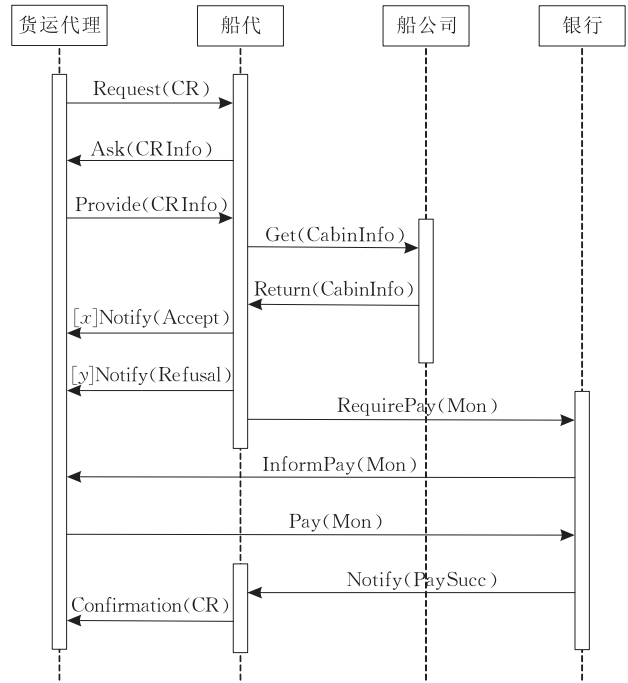


图 2 服务任务“预定舱位”的业务流程

在海运物流服务中, 货运代理作为顾客方参与一次“预定舱位”服务时, 其服务业务流程如下: 首先, 为了获得舱位使用权, 向船代申请订舱, 船代随之询问货运代理其需要舱位的详细信息, 如航线, 船期等, 货物代理接到询问后, 向船代提供详细的信息. 接着, 船代向船公司发出获取剩余船舱信息的申请, 船公司向船代提供其剩余的船舱信息, 根据船公司提供的剩余船舱的信息, 船代执行内部操作, 判断是否能够为货运代理提供舱位使用权, 根据实际情况, 发出接受或者拒绝信息. 如果, 船代接受了货运代理的舱位请求, 那么通知银行向货运代理收款. 银行接到通知后, 通知货运代理付款. 货物代理接到交款通知, 去银行支付订舱费用. 接着, 银行通知船代预定舱位费用已经收到. 最后, 船代向货物代理发出确认信息, 告知货运代理预定舱位成功, 即其已经取得了舱位使用权. 在整个服务任务中, 货运代理作为顾客方提出服务需求, 而船代、船公司、银行联合在一起, 构成虚拟提供方组织, 共同向货运代理提供预定舱位服务. 下面, 详细给出了整个服务任务中, 各个对象之间的交互行为及行为的作用对象:

Request(CR): 货运代理请求舱位使用权 CR,

(*Req*; *CR*);

Ask(*CRInfo*): 船代询问货运代理关于舱位的详细信息 *CRInfo*, (*Ask*; *CRInfo*);

Provide(*CRInfo*): 货运代理将货主需要舱位的详细信息返回给船代, 如始发港口、目的港口、船期等信息 *CRInfo*, (*Pro*; *CRInfo*);

Get(*CabinInfo*): 船代向船公司请求当前可获得的舱位信息 *CabinInfo*, (*Get*; *CabinInfo*);

Return(*CabinInfo*): 船公司返回相关舱位信息 (*Ret*; *CabinInfo*);

[*x*]Notify(*Refusal*): 根据货运代理提供的舱位详细信息和船公司返回的可获得的舱位信息, 船代发现无法满足货运代理的请求, 通知货运代理无法继续服务的拒绝信息 *Refusal*, (*Not*; *Refusal*);

[*y*]Notify(*Accept*): 能够满足货运代理的请求, 船代接受了订舱请求 *Accept*, (*Not*; *Accept*);

RequirePay(*Mon*): 船代向银行要求付款 *Mon*, (*ReP*; *Mon*);

InformPay(*Mon*): 银行向货运代理发出付款请求, (*Inf*; *Mon*);

Pay(*Mon*): 货运代理付款并确认, (*Pay*; *Mon*);

Notify(*PaySucc*): 银行通知船代, 货运代理已付款, (*Not*; *PaySucc*);

Confirmation(*CR*): 船代向货运代理发出确认信息, 订舱成功 (*Con*; *CR*).

在扩展 UML 时序图中, 对象的缩写如下: 货运代理 *F* (freight forwarder)、船代 *SA* (Ship Agency)、船公司 *SO* (Ship Owner)、银行 *B* (Bank).

利用 VPA 形式化表达服务业务流程模型, 需要弄清楚两者的元素之间的对应关系以及在元素映射时需要注意的规则. 服务业务流程模型描述方式有多种, 本文以扩展的 UML 序列图为例, 来说明如何利用 VPA 来形式化表达服务业务流程模型. 在此给出扩展的 UML 时序图与 VPA 的元素之间的对应关系: (1) 扩展 UML 时序图中的对象 *Object* 对应 VPA 中的进程 *Process*; (2) 扩展 UML 时序图中的行为 *Behavior* 对应 VPA 中的动作 *Action*; (3) 扩展 UML 时序图中的消息 *Message* 对应 VPA 中的消息 *Message*.

利用 VPA 形式化描述扩展 UML 时序图时, 若图中的两个对象之间存在按照一定的顺序传递的消息或者两个对象之间按一定的顺序向对方发出动作、相互作用, 则需要在两个对象之间建立一条通道. 按照这一规则, 并且根据前面提到的两者的元素

之间的对应关系, 本文将图 2 中各个对象的服务业务流程进行形式化表达, 结果如下:

(1) 描述货运代理业务流程的 VPA 进程表达式:

$$\begin{aligned} F &= \bar{x} \langle Req; CR \rangle x(msg) [msg = Ask; CRInfo] \\ &\quad \bar{x} \langle Pro; CRInfo \rangle x(msg1) F1, \\ F1 &= [msg1 = Not; Refusal] F + [msg1 = \\ &\quad Not; Accept] w(msg2) [msg2 = Inf; Mon] \\ &\quad \bar{w} \langle Pay; Mon \rangle x(msg3) [msg3 = Con; CR] F, \end{aligned}$$

其中 x 表示货运代理与船代之间的通道, w 表示货运代理与银行之间的通道, $\bar{x} \langle Req; CR \rangle$ 表示货运代理向船代发起服务行为 *Req*; *CR*, $w(msg2) [msg2 = Inf; Mon]$ 表示货运代理接收来自银行的服务行为 *Inf*; *Mon*.

此外需要说明的是无论 $\bar{x} \langle Req; CR \rangle$ 还是 $w(msg2) [msg2 = Inf; Mon]$ 均是省略的描述, 相应的完整描述应该为 $(\bar{x} \langle Req; CR \rangle, q(\bar{x} \langle Req; CR \rangle))$ 和 $(w(msg2) [msg2 = Inf; Mon], q(w(msg2) [msg2 = Inf; Mon]))$. 在形式化过程中需要将模型中服务行为在质量方面的约束同时加入进程表达式中. 但为了使进程表达式简洁、易于理解, 在此处的 4 个 VPA 进程表达式中均省略了动作的质量约束描述.

(2) 描述船代业务流程的 VPA 进程表达式:

$$\begin{aligned} SA &= x(msg) [msg = Req; CR] \bar{x} \langle Ask; CRInfo \rangle \\ &\quad x(msg1) [msg1 = Pro; CRInfo] \\ &\quad \bar{y} \langle Get; CabinInfo \rangle \\ &\quad y(msg2) [msg2 = Ret; CabinInfo] SA1, \\ SA1 &= \bar{x} \langle Not; Refusal \rangle SA + \\ &\quad \bar{x} \langle Not; Accept \rangle \bar{z} \langle ReP; Mon \rangle z(msg3) \\ &\quad [msg3 = Not; PaySucc] \bar{x} \langle Con; CR \rangle SA, \end{aligned}$$

其中 x 表示货运代理与船代之间的通道, y 表示船代与船公司之间的通道, z 表示船代与银行之间的通道.

(3) 描述船公司业务流程的 VPA 进程表达式:

$$\begin{aligned} SO &= y(msg) [msg = Get; CabinInfo], \\ &\quad \bar{y} \langle Ret; CabinInfo \rangle SO, \end{aligned}$$

其中 y 表示船代与船公司之间的通道.

(4) 描述银行业务流程的 VPA 进程表达式:

$$\begin{aligned} B &= z(msg) [mag = Rep; Mon] \bar{w} \langle Inf; Mon \rangle \\ &\quad w(msg1) [msg1 = Pay; Mon] \bar{z} \langle Not; PaySucc \rangle B, \end{aligned}$$

其中 w 表示货运代理与银行之间的通道, z 表示船代与银行之间的通道.

在服务任务“预定舱位”中, 货运代理的角色属于顾客方, 而船代、船公司、银行的角色属于服务提

供方,本文将船代、船公司、银行看成一个逻辑上的虚拟服务提供方组织,其整体服务流程的进程表达式为 $SP=(y,z)(SA|SO|B)$.

为了进行面向价值的服务模型分析,需要进一步地将待验证的进程表达式转换成价值状态生成图的形式.在价值状态图中,状态顶点的属性信息由两部分构成:一部分为“顶点的输入行为”和“输入行为的作用对象”,它们描述了该状态顶点能够支持价值实现哪些功能;另一部分为价值实现程度计算公式的值 $f(s_{i+1})$,其描述了目标价值的指标被支持实现的值.

与 CCS 进程代数类似,在 VPA 中其它类型的算子都可以转成动作前缀算子和选择算子,即是说所有其它类型的进程表达式都可以表示成一般形式:

$$P = \sum_{i=1}^{i=n} C_i P, C_i = \alpha_p \alpha_j \dots \alpha_q, \alpha_j = \bar{y} \langle x \rangle | y(x) | \tau,$$

在上式中 C_i 是动作前缀算子 α_j 有穷序列.

下面针对一般表达式,给出价值状态图生成算法.

算法 1. *CreateValueStateGrpth*(P, g, h).

输入:进程表达式 P 、动作质量约束 g 、价值实现程度计算函数 h

输出: P 的价值状态图

过程:

1. 创建价值状态图的初始顶点 $S=(s_0, 0)$;
2. *create*(S, P, g, h)
3. {扫描 P , 根据选择算子“+”, 将 P 分割成 n 部分 $C_1 P, C_2 P, \dots, C_i P, \dots, C_n P$;
4. for ($i=1, i \leq n, i++$) {
5. while ($|C_i| \geq 1$) {
6. 生成一个新的顶点 $S_j=(s_j, f(s_j))$ 和一条边 $S \xrightarrow{(\alpha_j, q(\alpha_j))} S_j$;
7. $j++$;
8. 从 C_i 中去掉 α_j ;
9. if α_j != “结束动作消息”
10. then { $S=S_j$; *Create*(S, P, g, h); }

利用上述算法,可以将待验证的进程表达式转换成价值状态图的形式,例如,针对描述货运代理 F 业务流程的进程表达式:

$$\begin{aligned} F &= \bar{x} \langle \text{Req}; CR \rangle x(\text{msg}) [\text{msg} = \text{Ask}; CRInfo] \\ &\quad \bar{x} \langle \text{Pro}; CRInfo \rangle x(\text{msg}1) F1, \\ F1 &= [\text{msg}1 = \text{Not}; Refusal] F + [\text{msg}1 = \\ &\quad \text{Not}; Accept] w(\text{msg}2) [\text{msg}2 = \text{Inf}; Mon] \\ &\quad \bar{w} \langle \text{Pay}; Mon \rangle x(\text{msg}3) [\text{msg}3 = \text{Con}; CR] F, \end{aligned}$$

其符合一般式的结构,利用此算法,生成其相应的价值状态图,如图 3 所示.

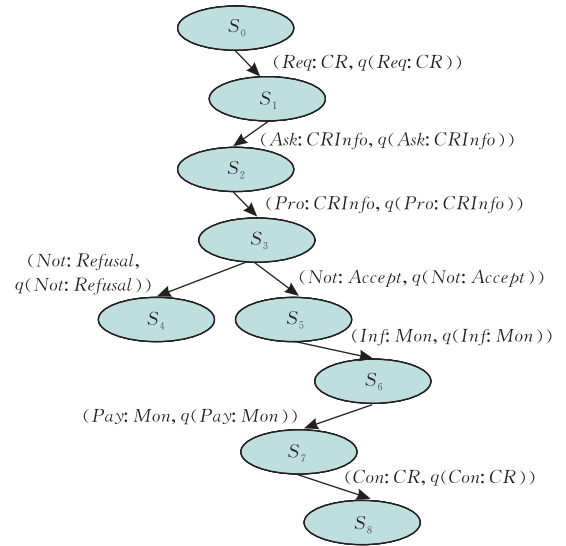


图 3 货运代理 F 的进程表达式的价值状态图

4 服务业务流程模型分析

本文的最终目标是验证服务业务流程模型能否满足顾客的服务需求,如上所述,一方面,利用价值模型准确、全面地表达了顾客的服务需求;另一方面,利用 VPA 形式化表达了服务业务流程模型,并且将待验证的进程表达式转换成了价值状态图的形式.至此本文将问题转换为:验证价值状态图能否支持目标价值的实现以及支持目标价值实现的程度.本文分别从定性和定量两个角度逐步地分析目标价值的实现情况.分析目标包括下面 3 个性质.

性质 1(价值可实现性). 一项目标价值的实现载体经过服务业务流程模型中的服务行为作用之后,能够达到其结束状态,那么就说明该目标价值的可实现性能被此服务业务流程模型满足.

性质 2(价值实现完整性). 一项目标价值的实现载体经过服务业务流程模型中的一系列服务行为作用之后,能够从其开始状态逐步地转换到结束状态,那么就说明该目标价值的实现完整性能够被此服务业务流程模型满足.

性质 3(价值实现程度). 在服务业务流程模型中的一系列质量约束服务行为的支持下,价值指标被实现的值满足价值指标的期望约束的程度.

利用价值状态图中状态顶点和价值模型中目标价值之间在语义上存在一定的等价映射规则,可以对前两个性质进行验证.在服务业务流程模型中,业务开始(或结束)阶段的服务行为具有一定的特征,例如,在开始阶段,一般以请求行为开始业务流程,而在结束阶段,以确认行为结束业务流程.那么根据

开始(或结束)阶段的特征行为,能够在顶点的输入行为和目标价值实现载体的开始(或结束)状态之间,建立特定业务领域的语义等价映射规则.例如,货运代理接收目标价值 SGV_3 ,在功能性方面,本文利用三元组 $\langle RC = \text{舱位使用权}, SS = \text{货代请求}, ES = \text{货代已获得} \rangle$ 描述该目标价值的实现情况,在价值状态图中,与其语义等价的顶点为 $S_1(Req, CR)$ 和 $S_8(Con, CR)$,其中 S_1 与开始状态 SS 等价, S_8 与结束状态 ES 等价.下面给出目标价值与状态顶点的语义等价映射规则.

规则 1. 目标价值的“实现载体 RC ”对应状态顶点的“输入行为作用对象 $object$ ”;

规则 2. 实现载体的“初始(结束)状态 $SS(ES)$ ”对应状态顶点的“输入特征行为 $behavior$ ”.

4.1 价值可实现性验证

如上所述,在对复杂、交互的服务系统建模时,将整个服务分解为一组服务任务 $\{Task_1, Task_2, \dots, Task_n\}$,其中,每一项服务任务的业务流程由扩展 UML 时序图来描述,而任意一张扩展 UML 时序图又可被形式化为一组进程表达式 $Task_i = \{P_1, P_2, \dots, P_k\}$;服务需求由一组目标价值 $\{V_1, V_2, \dots, V_m\}$ 表达.价值可实现性验证就是验证每一项目标价值的实现载体的结束状态能否在服务业务流程模型的支持下被满足.

针对价值模型中的每一项目标价值,首先在构成服务的任务集合 $\{Task_1, Task_2, \dots, Task_n\}$ 中搜索支持其实现的服务行为,根据目标价值 V_i 的属性信息,能够获得 V_i 的接收者 R ,通过对比目标价值的接受者 R 和任务集合中顾客方对象二者之间的关系,找出可能支持目标价值实现的业务流程对应的进程表达式 P ,将其转成价值状态图 G ,在图 G 中寻找与目标价值的实现载体的结束状态在语义上等价的的状态顶点,价值可实现性验证算法的细节如下:

算法 2. *ValueRealizabilityVerification()*.

输入: 服务任务集合 $= \{Task_1, Task_2, \dots, Task_n\}$,

目标价值集合 $= \{V_1, V_2, \dots, V_m\}$

输出: $\{VSG_1, VSG_2, \dots, VSG_m\}$,其中 $VSG_j = \{V, S, G\}$,
 V 表示输出结果对应的目标价值, S 为与目标价值 V 的实现载体 RC 的结束状态 ES 语义等价的状态顶点, G 为顶点 S 所在的价值状态图

过程:

1. for ($j=1; j \leq m; j++$) $\{VSG_j \rightarrow V = V_j;$
2. for ($i=1; i \leq n; i++$) $\{$
3. if $Customer(Task_i) == V_j \rightarrow R$ then
4. $\{P = Customer(Task_i)$ 的进程表达式;
5. $VSG_j \rightarrow G = CreateValueStateGrpth(P);$

6. 遍历图 $VSG_j \rightarrow G$,得到顶点集合 $\{S_1, S_2, \dots, S_p\};$
7. for ($k=1; k \leq p; k++$) $\{$
8. if $(MatchSemantics(V_j \rightarrow RC, S_k \rightarrow object) \& \&$
 $MatchSemantics(V_j \rightarrow ES, S_k \rightarrow behavior))$
9. $VSG_j \rightarrow S = S_k; \}$

其中, $Customer(Task_i)$ 函数的功能是从描述服务任务 $Task_i$ 的服务业务流程模型中找出顾客方对象; $MatchSemantics(x, y)$ 函数的功能是判断两者是否在语义上存在等价关系.假设服务被分解成的任务数为 n ,并且价值模型中待验证的目标价值个数为 m ,价值状态图中顶点个数为 p ,那么该算法的时间复杂度为 $O(m \cdot n \cdot p)$.

利用此算法,可以完成价值可实现性的验证,如果在输出的结果集 $\{VSG_1, VSG_2, \dots, VSG_m\}$ 中,某一 $VSG_j \rightarrow S$ 为空,则说明对应的目标价值 $VSG_j \rightarrow V$ 可实现性没有被满足,即服务业务流程模型不能够满足 $VSG_j \rightarrow V$ 描述的顾客需求,需要在下一阶段的服务模型优化中进行相应的调整;如果 $\{VSG_1, VSG_2, \dots, VSG_m\}$ 中 $VSG_j \rightarrow S$ 均不为空,则说明顾客期望的目标价值的可实现性都能够被满足,则继续进行下一步的价值实现完整性验证.

4.2 价值实现完整性验证

文献[6, 11]给出了如何验证系统与设计要求一致的方法:反转证明法,即利用 CCS 或者 P_i 的观察等价性证明.此处,我们可以利用此方法来证明一组支持目标价值实现完整性的顾客方的行为,其对偶与虚拟提供方组织的行为观察等价,从而证明整个服务业务流程模型支持目标价值的实现完整性.

进一步地,根据实现完整性的定义,给出满足实现完整性的 3 个必要条件:(1) 可实现性被满足,即目标价值的实现载体的结束状态被满足;(2) 目标价值的实现载体的开始状态被满足;(3) 支持目标价值的实现载体从开始状态变换到结束状态的一组顾客方的服务行为的对偶与模型中虚拟提供方组织的行为观察等价.

从上面的必要条件可以看出,价值实现完整性的验证是在价值可实现性被满足的基础之上进行的,后者的输出可以作为前者的输入.下面给出价值实现完整性验证算法的详细过程.

算法 3. *ValueCompleteRealizabilityVerification()*

输入: $\{VSG_1, VSG_2, \dots, VSG_m\}$

输出: 支持价值实现完整性的动作集

$Action_1, Action_2, \dots, Action_m$

过程:

1. for ($j=1; j \leq m; j++$) $\{$
2. if $VSG_j \rightarrow S! = \emptyset$ then $\{$

3. 遍历图 $VSG_j \rightarrow G$, 得到顶点集合 $\{S_1, S_2, \dots, S_p\}$;
4. for ($k=1; k \leq p; k++$) $\{V_j = VSG_j \rightarrow V;$
5. if ($MatchSemantics(V_j \rightarrow RC, S_k \rightarrow object) \& \&$
 $MatchSemantics(V_j \rightarrow SS, S_k \rightarrow behavior)$)
6. 在价值状态图 $VSG_j \rightarrow G$ 中, 搜索从 S_k 到 $VSG_j \rightarrow S$ 的服务动作集 $Action_1, Action_2, \dots, Action_m$;
7. 找出价值状态图 $VSG_j \rightarrow G$ 对应的进程表达式 $\{P_1, P_2, \dots, P_k\}$, 其中 P_j 为顾客方的进程表达式, $SP = \{P_1, P_2, \dots, P_k\} - P_j$ 虚拟提供方组织的进程表达式;
8. 将动作集对应的进程表达式 $A \subseteq P_j$ 转变成对偶形式 $ReverseA$;
9. 证明 $ReverseA$ 与虚拟提供方组织 SP 的进程表达式观察等价: 针对 \forall 动作 $\alpha, ReverseA \xrightarrow{\alpha} ReverseA'$, 存在 SP' 满足 $SP \xrightarrow{\tau' \alpha \tau'} SP'$.

利用上述算法能够完成价值实现完整性的验证. 如果对于某一非空的 $VSG_j \rightarrow S$, 能找到一组满足条件的动作集, 并且利用反转证明法, 能够证明其对偶进程表达式与相应的虚拟提供方组织的进程表达式观察等价, 则目标价值 $VSG_j \rightarrow V$ 的实现完整性被整个服务业务流程模型满足, 即说明顾客在功能性方面的服务需求能够被服务业务流程模型充分满足; 若无法找到满足条件的动作集, 或者找到了动作集, 但是证明的结果为观察不等价, 则说明目标价值 $VSG_j \rightarrow V$ 的实现完整性没有被满足, 需要在下一阶段的模型优化中进行适当的优化处理.

4.3 价值实现程度分析

价值实现程度受两方面影响: 一方面是支持该项价值完整实现的质量约束动作集和动作之间的拓扑结构, 通过计算动作的质量参数及相关参数, 能够得到价值指标的值, 再与价值指标的期望约束进行比较, 便可计算出价值实现程度; 另一方面, 价值的实现程度还受价值之间的依赖关系的影响. 本文主要研究前者, 而依赖关系对价值指标值的影响将在其它文章中讨论.

在质量约束动作集 $Action_1, Action_2, \dots, Action_m$ 的支持下, 某一目标价值被服务业务流程模型完整地实现. 目标价值的状态是随着质量约束动作的执行而不断变化的, $(s_i, f(s_i)) \xrightarrow{(\alpha, q(\alpha))} (s_{i+1}, f(s_{i+1}))$, 那么带有质量约束的进程演算可以得到价值的实现程度, 利用价值实现程度计算公式 $f(s_{i+1}) = h(f(s_i), q(\alpha))$ 可以计算目标价值在某一种状态下的实现程度. $f(s_{i+1})$ 可以表示不同类型的价值指标在此状态下的实现值; 在不同的应用领域背景下, 针对不同类型的价值指标以及支持价值指标实现的动作的拓扑结构的不同, $h()$ 的具体形式是不同的;

$q(\alpha) = g(q_1, q_2, \dots, q_n)$ 函数的功能是取出质量约束动作 $\alpha = Action_i$ 的质量约束中与待计算的价值指标相关的质量参数. 伴随着质量动作的不断执行, 根据计算函数 $h()$ 的具体形式, 能够得到不同状态下的指标实现值, 进而与价值指标的期望约束对比, 计算得到价值指标的实现程度, 重复上述过程, 能够得到目标价值的所有价值指标的实现程度, 也就获得了目标价值本身的实现程度.

通过验证目标价值的可实现性、实现完整性和计算目标价值的实现程度, 本文能够对服务业务流程模型满足顾客的服务需求的情况进行定性和定量的分析. 假如服务业务流程模型能够满足顾客的服务需求, 那么就可以根据该模型去开发服务系统; 假如服务业务流程模型没能满足顾客的服务需求, 则需要通过引入新的服务使能者或替换服务功能元素等手段对模型进行优化处理, 直到模型能够满足顾客的服务需求.

5 案例分析

为了说明上述分析方法的有效性, 本文采用海运物流出口服务进行案例分析. 第 2、3 节分别给出了服务需求、服务业务流程模型及其进程表达式的实例. 以此为基础, 能够进行验证过程的案例分析. 但限于篇幅, 此处仅给出目标价值 SGV_3 的详细验证过程.

第 1 阶段进行价值可实现性验证.

步 1~4, 假设此处选择了目标价值 SGV_3 ; 经过判断, 发现服务任务“预定舱位”中顾客方的参与者与 SGV_3 的接收者 R 相同, 均为货运代理, 对应的进程表达式为

$$F = \bar{x} \langle Req; CR \rangle x(msg) [msg = Ask; CRInfo]$$

$$\bar{x} \langle Pro; CRInfo \rangle x(msg1) F1,$$

$$F1 = [msg1 = Not; Refusal] F +$$

$$[msg1 = Not; Accept] w(msg2) [msg2 = Inf; Mon]$$

$$\bar{w} \langle Pay; Mon \rangle x(msg3) [msg3 = Con; CR] F.$$

步 5、6, 利用价值状态图生成算法, 将进程表达式 F 转换成价值状态图 $VSG_3 \rightarrow G$ (如图 3 所示), 遍历该图, 获得图中各顶点的信息;

步 7、8, 选择顶点, 将顶点的相关信息与 SGV_3 的价值实现载体“舱位使用权”和结束状态“货代已获得”比较, 能够得到与其语义等价的顶点 $S_8(Con, CR)$, 即 SGV_3 的可实现性被服务任务“预定舱位”的业务流程模型满足.

在判断服务任务中顾客方的参与者与 SGV_3 的接收者 R 是否相同时, 得到的服务任务不仅仅是“预定舱位”, 但在进行语义等价匹配时, 其它任务对应的价值状态图的顶点与 SGV_3 语义不等价, 所以

被舍弃。重复执行上述过程,能够分析得到所有目标价值的可实现性被支持的情况,结果记录在 $\{VSG_1, VSG_2, \dots, VSG_m\}$ 中。

第 2 阶段进行价值实现完整性验证,其输入为价值可实现性的输出 $\{VSG_1, VSG_2, \dots, VSG_m\}$ 。

步 1~3,此处仍以 SGV_3 为例, $VSG_3 \rightarrow S$ 不为空,即 SGV_3 可实现性被满足,遍历图 $VSG_3 \rightarrow G$;

步 4,5,将 $VSG_3 \rightarrow G$ 中顶点的相关信息与 SGV_3 的价值实现载体“舱位使用权”和初始状态“货代请求”进行语义等价匹配,得到顶点 $S_1(Req, CR)$;

步 6,在图 $VSG_3 \rightarrow G$ 中,找出从 S_1 到 S_3 的一条通路,获得动作集 $\{Req: CR, Ask: CRInfo, Pro: CRInfo, Not: Accept, Inf: Mon, Pay: Mon, Con: CR\}$;

步 7,图 $VSG_3 \rightarrow G$ 中顾客方对应的进程表达式为 F ,虚拟提供方的进程表达为 $SP = (y, z)(SA | SO | B)$,去掉内部动作 y 和 z ,得到 SP 的进程表达式:

$$\begin{aligned} SP &= x(msg)[msg = Req: CR] \bar{x} \langle Ask: CRInfo \rangle \\ & x(msg1)[msg1 = Pro: CRInfo] \tau(\bar{x} \langle Not: Refusal \rangle SP + \\ & \bar{x} \langle Not: Accept \rangle \tau \bar{x} \langle Con: CR \rangle SP) | \tau SP | \\ & \tau \bar{w} \langle Inf: Mon \rangle w(msg1)[msg1 = Pay: Mon] \tau SP. \end{aligned}$$

步 8,对进程表示式 F 中动作集对应的部分 A 取对偶,其形式:

$$\begin{aligned} ReverseA &= x(msg)[msg = Req: CR] \bar{x} \langle Ask: CRInfo \rangle \\ & x(msg1)[msg1 = Pro: CRInfo] \bar{x} \langle Not: Accept \rangle \\ & \bar{w} \langle Inf: Mon \rangle w(msg2)[msg2 = Pay: Mon] \\ & \bar{x} \langle Con: CR \rangle ReverseA. \end{aligned}$$

步 9,若 SP 弱模拟 $ReverseA$,那么对于 $ReverseA$ 能够接受的任何动作, SP 都能接受。证明过程(穷取法):

9.1. 输入动作 $\alpha = x(msg)[msg = Req: CR]$,

9.1.1. $ReverseA$ 发生的变化为

$$\begin{aligned} ReverseA \xrightarrow{x(msg)[msg = Req: CR]} ReverseA' &= \\ \bar{x} \langle Ask: CRInfo \rangle x(msg1)[msg1 = Pro: CRInfo] \bar{x} \langle Not: Accept \rangle & \\ \bar{w} \langle Inf: Mon \rangle w(msg2)[msg2 = Pay: Mon] & \\ \bar{x} \langle Con: CR \rangle ReverseA. & \end{aligned}$$

9.1.2. SP 发的变化为

$$\begin{aligned} SP \xrightarrow{x(msg)[msg = Req: CR]} SP' &= \bar{x} \langle Ask: CRInfo \rangle \\ x(msg1)[msg1 = Pro: CRInfo] \tau(\bar{x} \langle Not: Refusal \rangle SP + & \\ \bar{x} \langle Not: Accept \rangle \tau \bar{x} \langle Con: CR \rangle SP) & \\ | \tau SP | \tau \bar{w} \langle Inf: Mon \rangle w(msg1)[msg1 = Pay: Mon] \tau SP. & \end{aligned}$$

9.2. 输入动作

$\alpha = x(msg)[msg = Req: CR]$

$\bar{x} \langle Ask: CRInfo \rangle x(msg1)[msg1 = Pro: CRInfo]$,

9.2.1. $ReverseA$ 发生的变化为

$$\begin{aligned} ReverseA \xrightarrow{\alpha} ReverseA' &= \bar{x} \langle Not: Accept \rangle \\ \bar{w} \langle Inf: Mon \rangle w(msg2)[msg2 = Pay: Mon] & \\ \bar{x} \langle Con: CR \rangle ReverseA. & \end{aligned}$$

9.2.2. SP 发的变化为

$$SP \xrightarrow{\alpha} SP' = \tau(\bar{x} \langle Not: Refusal \rangle SP +$$

$\bar{x} \langle Not: Accept \rangle \tau \bar{x} \langle Con: CR \rangle SP$

$| \tau SP | \tau \bar{w} \langle Inf: Mon \rangle w(msg1)[msg1 = Pay: Mon] \tau SP.$

9.3. 输入动作

$\alpha = x(msg)[msg = Req: CR] \bar{x} \langle Ask: CRInfo \rangle$

$x(msg1)[msg1 = Pro: CRInfo]$

$\bar{x} \langle Not: Accept \rangle \bar{w} \langle Inf: Mon \rangle$

$w(msg2)[msg2 = Pay: Mon] \bar{x} \langle Con: CR \rangle.$

9.3.1. $ReverseA$ 发生的变化为

$$ReverseA \xrightarrow{\alpha} ReverseA' = ReverseA.$$

9.3.2. SP 发的变化为

$$SP \xrightarrow{\alpha} SP' = SP | SP | SP.$$

通过穷举,可以证明 SP 是弱模拟 $ReverseA$ 的,即目标价值 SGV_3 的实现完整性能够被整个服务业务流程模型满足。

最后,定量地分析价值 SGV_3 实现程度。 SGV_3 属于 RUV 类型的价值,其指标集合由响应时间、可靠性、资源可用性、金钱成本、时间成本、精力与体力成本六项价值指标构成,可表示为 $VI = \{rt, rep, ra, mc, tc, epc\}$ 。 $f(s_{i+1})$ 可分别具体化成 $rt(s_{i+1})$, $rep(s_{i+1})$ 等。

根据验证 SGV_3 的实现完整性的结果,能够知道 SGV_3 由动作集 $\{Req: CR, Ask: CRInfo, Pro: CRInfo, Not: Accept, Inf: Mon, Pay: Mon, Con: CR\}$ 支持实现,该组动作集的拓扑结构为顺序结构。下面给出了 SGV_3 的每项指标的计算公式 $h()$ 的具体形式。

(1) 响应时间 rt

$$rt(s_{i+1}) = rt(s_i) + q_{et}(Action_{i+1}),$$

其中 $q_{et}(Action_{i+1})$ 表示动作 $Action_{i+1}$ 的质量参数“执行时间” q_{et} ;

(2) 可靠性 rep

$$rep(s_{i+1}) = rep(s_i) \times q_{rep}(Action_{i+1}),$$

其中 $q_{rep}(Action_{i+1})$ 表示动作 $Action_{i+1}$ 的质量参数“可靠性” q_{rep} ;

(3) 资源可用性 ra

$$ra(s_{i+1}) = ra(s_i) \times q_a(Action_{i+1}),$$

其中 $q_a(Action_{i+1})$ 表示动作 $Action_{i+1}$ 的质量参数“可用性” q_a ;

(4) 资金成本 mc

$$mc(s_{i+1}) = mc(s_i) + q_{aef}(Action_{i+1}),$$

其中 $q_{aef}(Action_{i+1})$ 表示相关动作 $Action_{i+1}$ 的质量参数“动作执行费用” q_{aef} ;

(5) 时间成本 tc

利用顾客亲自参与到服务中花费的时间与顾客获得期望价值所经历的总体时间的比值来衡量时间

成本,可以表示成

$$tc(s_{i+1}) = \begin{cases} \frac{tc(s_i) \sum_{j=1}^i q_{et}(Action_j)}{\sum_{j=1}^{i+1} q_{et}(Action_j)}, Action_{i+1} = \alpha \\ \frac{tc(s_i) \sum_{j=1}^i q_{et}(Action_j) + q_{et}(Action_{i+1})}{\sum_{j=1}^{i+1} q_{et}(Action_j)}, Action_{i+1} = \bar{\alpha} \end{cases}$$

其中 $\bar{\alpha}$ 表示输出动作,即顾客执行的动作; α 表示输入动作,即虚拟提供方执行的动作; q_{et} 表示动作 *Action* 的质量参数“执行时间”;

(6) 精力与体力成本 *epc*

利用顾客亲自参与到服务中的程度,即在整个服务过程中,顾客在精力与体力方面的消耗,可以表示成

$$epc(s_{i+1}) = \begin{cases} \frac{epc(s_i) \times i + ep_{i+1}}{i+1}, Action_{i+1} = \alpha \\ \frac{epc(s_i) \times i + \overline{ep}_{i+1}}{i+1}, Action_{i+1} = \bar{\alpha} \end{cases}$$

其中 \overline{ep} 表示顾客执行输出动作时,参与服务的程度, ep 表示顾客参与输入动作时,参与服务的程度。

假设在设计服务业务流程模型时,动作集的质量参数已经被预先设置完毕,表 4 给出其模拟仿真数据。

表 4 动作质量约束的模拟仿真数据

动作及作用对象	q_{et}/h	$q_{rep}/\%$	$q_a/\%$	$q_{acf}/\text{元}$
Req; CR	≤ 0.1	≥ 99	100	0
Ask; CRInfo	≤ 0.1	≥ 99	100	≤ 10
Pro; CRInfo	≤ 0.2	≥ 95	100	0
Not; Accept	≤ 0.3	≥ 90	≥ 95	≤ 20
Inf; Mon	≤ 0.3	≥ 90	100	≤ 20
Pay; Mon	≤ 0.2	≥ 95	100	0
Con; CR	≤ 0.1	≥ 99	100	≤ 10

此外,为了计算方便,本文将 \overline{ep} 统一设置为 100%,将 ep 统一设置为 10%;最后,根据上述内容,计算得到 SGV_3 的各项价值指标的实现的程度.利用相同的方法,能够得到 SGV_4 的价值指标 $rep=95\%$ 。

从表 5 中可以发现 SGV_3 的价值指标“响应时间”、“资源可用性”、“资金成本”、“时间成本”、“精力与体力成本”的实现值能够满足顾客对价值指标的基本型的期望约束,并且 SGV_3 的“响应时间”的实现值不会触发与其相关的规则型期望约束.而 SGV_3 的价值指标“可靠性”的实现值没能满足顾客对“可靠性”的第一条期望约束,同样的, $\{SGV_3,$

$SGV_4\}.rep=70\%$ 也没能满足顾客对“可靠性”的第 2 条期望约束.因此, SGV_3 的“响应时间”是下一步进行模型优化的处理对象。

表 5 指标集合的期望约束与实现值

VI	期望约束	实现值
<i>rt</i>	$Basic(\{SGV_3\}, (rt, \leq, 2h));$ $Rule(\{(\{SGV_3\}, (rt, >, 1.5h)), \text{and}, \emptyset),$ $(\{(\{SGV_5\}, (rt, <, 2h)), \text{and}, \emptyset);$	$\leq 1.3h$
<i>rep</i>	$Basic(\{SGV_3\}, (rep, \geq, 80\%));$ $Basic(\{SGV_3, SGV_4\}, (rep, \geq, 85\%));$	$\geq 71\%$
<i>ra</i>	$Basic(\{SGV_3\}, (ra, \geq, 95\%))$	$\geq 95\%$
<i>mc</i>	$Basic(\{SGV_3\}, (mc, \leq, 80 \text{元}))$	$\leq 60 \text{元}$
<i>tc</i>	$Basic(\{SGV_3\}, (tc, \leq, 50\%))$	38%
<i>pc</i>	$Basic(\{SGV_3\}, (epc, \leq, 50\%))$	50%

6 结 论

服务建模作为服务工程中的一个关键步骤,其设计的服务模型能否支持顾客期望价值的实现以及支持价值实现的程度是一个值得研究的热点问题.针对此问题,本文在 Pi 演算的基础上,扩展了动作质量约束和价值实现程度计算函数,得到了能够支持进行面向价值的模型分析的 VPA,利用它本文验证了服务业务流程模型对目标价值实现的支持情况.本文提出的价值进程代数是进程代数研究领域有意补充,同时也为服务模型的验证提供了一种新的思路和方法.利用该方法,能够进行面向价值的服务模型分析,为后续面向价值的服务模型优化提供依据,进而进行服务系统的开发.下一步的工作一方面考虑对现有工具进行改进使得其支持 VPA 的自动验证;另一方面,考虑利用 VPA 验证服务业务流程模型能否支持价值之间依赖关系,从而更加全面地进行面向价值的服务模型分析。

参 考 文 献

[1] Tien J, Berg D. Towards service systems engineering. IEEE International Conference on Systems, Man and Cybernetics, 2003, 5(5): 4890-4895

[2] Li Chang-Yun, Yang Ai-Min, Man Jun-Feng, Ying Jing. An approach to constructing business model for services integration according to requirement. Chinese Journal of Computers, 2006, 29(7): 1095-1104(in Chinese)
(李长云, 阳爱民, 满君丰, 应晶. 一种面向按需集成服务的业务模型构造方法. 计算机学报, 2006, 29(7): 1095-1104)

[3] Wang Zhong-Jie, Xu Xiao-Fei. Multi-level graphical service value modeling method. Computer Integrated Manufacturing Systems, 2009, 15(12): 2319-2327(in Chinese)
(王忠杰, 徐晓飞. 多层次图形化服务价值建模方法. 计算机集成制造系统, 2009, 15(12): 2319-2327)

- [4] Milner R. *Communicating and Mobile Systems; The Pi-Calculus*. Cambridge; Cambridge University Press, 1999
- [5] Milner R, Parrow J, Walker D. A calculus of mobile processes, part I/II. *Journal of Information and Computation*, 1992, 100(1): 1-77
- [6] Liao Jun, Tan Hao, Liu Jin-De. Describing and verifying Web service using Pi-calculus. *Chinese Journal of Computers*, 2005, 28(4): 635-643(in Chinese)
(廖军, 谭浩, 刘锦德. 基于 Pi-演算的 Web 服务组合的描述和验证. *计算机学报*, 2005, 28(4): 635-643)
- [7] Zhang Jing, Wang Hai-Yang, Cui Li-Zhen. Research on cross-organizational workflow modeling based on Pi-calculus. *Journal of Computer Research and Development*, 2007, 44(7): 1243-1251(in Chinese)
(张静, 王海洋, 崔立真. 基于 Pi 演算的跨组织工作流建模研究. *计算机研究与发展*, 2007, 44(7): 1243-1251)
- [8] Xiao Fang-Xiong, Huang Zhi-Qiu, Cao Zi-Ning, Yuan Min, Zhang Jun-Hua. Describing and cost analyzing of Web services composition using PPA. *Journal of Computer Research and Development*, 2009, 46(5): 832-840(in Chinese)
(肖芳雄, 黄志球, 曹子宁, 袁敏, 张君华. 基于价格进程代数的 Web 服务组合描述和成本分析. *计算研究与发展*, 2009, 46(5): 832-840)
- [9] Eberbach E. Calculus of bounded rational agents; Flexible optimization as search under bounded resources in interactive system. *Fundamenta Informaticae*, 2005, 68(1): 47-102
- [10] Wang Zhong-Jie, Xu Xiao-Fei, Ma Chao, Alice Liu. Service value meta-model; An engineering viewpoint//*Proceedings of the 6th International Conference on Interoperability for Enterprise Software and Applications(I-ESA 2010)*. Coventry, UK, 2010
- [11] Salaün G, Bordeaux L, Schaerf M. Describing and reasoning on Web services using process algebra//*Proceedings of the 2nd IEEE International Conference on Web Services*. San Diego, California, USA, 2004: 43-50



MA Chao, born in 1983, Ph. D. candidate. His current research interests include service computing and service 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.

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.

Background

In traditional service engineering methodologies, function- and quality-related service elements are emphatically focused, but value-related considerations are mainly ignored. To address this issue, the authors put forward a value-aware service engineering methodology. In VASEM, service analysts start from service model space by doing service innovation to invent a new service mode. Then, service value models are elaborated on to identify and describe all values that will be delivered by the service system. These values are then regarded as the objective constraints for service modeling and service system development.

The next step is to connect value objectives with functional elements in the service models and obtain the Value Annotation Model (VAM). Based on the VAM, qualitative and quantitative analyses are made to check whether and to what degree each value is supported by the functional elements. If there are deficiencies, service models are to be optimized by adjusting the selection of functional elements and their relationships. If these value deficiencies cannot be overcome by model optimization, it means that value constraints are too strict. In that case, they are sent back to value models for further consideration. If no value deficiencies are found during the analysis, then the service system is to be designed based on the service models.

This paper is to solve how to carry on qualitative and quantitative value analyses on service model. By importing the quality constrain of action and an implementation degrees of value function into Pi-calculus, VPA (value process algebra) is proposed, which is used to formally describe and validate service business proves model. By utilizing semantic equivalence matching algorithm and weak bisimulation theory, this paper verifies realizability, complete realizability of value, and analyzes the implementation degrees of value.

The paper is supported by the National Natural Science Foundation of China under grant Nos. 61033005, 60803091, and 70971029, and the National Basic Research Program of China (973 Program) under grant 2009AA04Z151.

The project is to make a research on value-aware service engineering methodology. Several papers have already been published under this project, mainly on the value model and value annotation, including preliminary discussions on several characteristics of service value, multi-level graphical service value modeling method, value annotation for service model analysis, and so on.

With the former work which is mainly on the value characteristics and value modeling, the work done in this paper carried on value analysis on service model, so as to provide references for value-oriented service model optimization.