Journal of Computer Applications
计算机应用,2013,33(1):266 -269
ISSN 1001-9081
CODEN JYIIDU
2013-01-01
http://www.joca.cn

文章编号:1001 -9081(2013)01 -0266 -04
doi:10.3724/SP.J.1087.2013.00266

基于时序描述逻辑的 Web 服务本体语言过程模型语义

李　明,刘士仪*,年福忠
(兰州理工大学 计算机与通信学院,兰州 730050)
(*通信作者电子邮箱 xbmulsy@126.com)

摘　要:针对 Web 服务本体语言(OWL-S)过程模型存在动态交互和时序特征表达能力不足的问题,提出一种基于时序描述逻辑的过程模型形式化方法。通过对 OWL-S 过程模型的原子过程和组合过程语义进行形式化的描述,得到了 OWL-S 的过程模型的动态语义,最终实现了对 OWL-S 过程模型的形式化建模。实例结果验证了所提方法的可行性,为进一步的分析和验证提供了基础。

关键词:Web 服务本体语言;时序描述逻辑;服务组合;形式化;过程模型;建模
中图分类号:TP301　　**文献标志码**:A

Semantics of OWL-S process model based on temporal description logic

LI Ming, LIU Shiyi*, NIAN Fuzhong
(*School of Computer and Communication, Lanzhou University of Technology, Lanzhou Gansu 730050, China*)

Abstract: Concerning the problem that Ontology Web Language for Services (OWL-S) process model lacks capacity for dynamic interaction and timing characteristics, a formalization method based on temporal description logic for process model was proposed. It described the atomic processes and composite processes of the OWL-S process model, and then the dynamic semantic of OWL-S process model was obtained. Finally, the formal modeling of OWL-S process model was realized. The experimental results show that the proposed method is feasible, and it provides the foundation for the analysis and validation.

Key words: Ontology Web Language for Services (OWL-S); temporal description logic; services composition; formalization; process model; modeling

0　引言

Web 服务本体语言(Ontology Web Language for Services, OWL-S)为 Web 服务提供了一种标准的语言结构,使得计算机能够智能化地实现服务的动态发现、调用和组合。然而,OWL-S 只是从语法角度对 Web 服务的过程模型进行描述,对过程模型的定义仅停留在概念框架层次,使得机器难以对 Web 服务过程模型进行自动分析[1]。

针对过程模型的语义问题,国内外诸多学者对其进行了许多研究并取得一定的成果。史忠值等[2]提出了一种动态描述逻辑的方法,从 OWL-S 的过程模型出发,实现对语义 Web 服务进行建模和推理;文献[3]提出利用时序行为逻辑(Temporal Logic of Action, TLA)对 OWL-S 组合流程进行描述并实现其相应的验证和推理;文献[4]提出了一种建立在语义框架基础上的 OWL-S 过程模型转化为重写逻辑模型的方法;文献[5]提出了基于 Petri 网的 OWL-S 服务操作语义。在这些方法中,都存在着一定程度的不足。采用动态描述逻辑对 Web 服务形式化建模中,没有对 Split 和 Split + Join 控制结构给出其相应的形式化语义[6];采用重写逻辑对 OWL-S 过程模型的前提和效果的描述效果较差;采用 Petri 网对 OWL-S 服务操作语义的描述会由于系统规模的不断扩大,可能出现状态空间爆炸的现象。

本文提出一种基于时序描述逻辑的 OWL-S 过程模型形式化建模方法。通过该方法得到的 OWL-S 过程模型的动态语义,可以很好地描述 Web 服务组合过程的并发和动态时序特征,最终实现对过程模型的形式化描述与验证。

1　OWL-S 过程模型

OWL-S[7-8]主要包含三个上层本体:ServiceProfile 用于描述 Web 服务做什么;ServiceModel 用于描述 Web 服务如何做;ServiceGrounding 用于描述 Web 服务如何被访问。其顶层本体结构如图 1 所示。

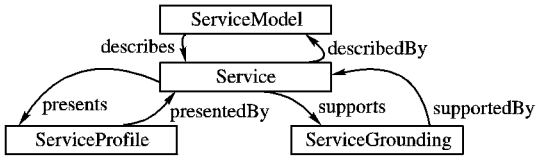

图 1　OWL-S 顶层本体图

过程模型(Process Model)定义了服务具体的内部流程以及执行的步骤和逻辑执行顺序等,主要用于实现服务的自动化组合和执行,分为 3 种:原子过程、组合过程和简单过程。

原子过程是最简单的过程,它是不可再分的,同时也不包含任何子过程,但可以直接被调用并且一步完成。

组合过程是通过构造算子把不同的多个原子过程或组合过程重新组合成一个新的组合过程。OWL-S 过程模型描述的组合过程控制结构主要有以下 8 种:Sequence、If-then-Else、Choice、Any-Order、Split + Join、Repeat-While、Repeat-Until 和 Split。

简单过程由于是对原子过程和组合过程的一种抽象化的描述,所以它是不可以直接被调用的,但它与原子过程相似,也可以一步完成。鉴于简单过程是一个抽象化的过程,因此本文暂不考虑简单过程动态语义的相关问题。

收稿日期:2012-07-16;**修回日期**:2012-08-19。
基金项目:甘肃省自然科学基金资助项目(1014RJZA028, 1112RJZA029);甘肃省高等学校基本科研项目(1114ZTC144)。
作者简介:李明(1959 -),男,河北辛集人,教授,主要研究方向:智能信息处理、知识工程;　刘士仪(1987 -),男,河南郸城人,硕士研究生,主要研究方向:Web 服务;　年福忠(1974 -),男,甘肃古浪人,副教授,博士,主要研究方向:复杂网络及复杂系统建模。

OWL-S 的本体语言(Ontology Web Language, OWL)是建立在描述逻辑基础上的,描述逻辑可以有效地表示和推理静态领域的知识,但是对于动作或服务等具有动态特征的知识无法处理[2]。除此之外,OWL-S 过程模型的语义主要体现在构造算子的操作语义上,但是 OWL-S 并没有明确定义构造算子如何运行。因此,它既不能表示 Web 服务之间动态交互和变迁,也不能对动态特性进行形式化建模,进而无法给出服务组合过程的形式化动态语义。为了达到对 OWL-S 过程模型操作语义进行全面形式化建模的目的,本文采用时序描述逻辑描述 OWL-S 过程模型的动态语义。

2　描述逻辑的时序扩展

描述逻辑的基本知识参见文献[9],在此不作介绍,以下主要是对描述逻辑进行时序扩展方面的部分工作。

时序描述逻辑主要是在原有的描述逻辑上扩展 4 个时序逻辑操作符[10-11]:○(at the next moment),□(always in the future),◇(eventually)和 U(until)。

2.1　扩展部分的语法和语义

1)扩展部分的语法。

$C,D ::= \Box C \mid \bigcirc C \mid \Diamond C \mid CUD$

$R_1,R_2 ::= \Box R_1 \mid \bigcirc R_1 \mid \Diamond R_1 \mid R_1 U R_2$

$\varphi,\psi ::= \Box\varphi \mid \bigcirc\varphi \mid \Diamond\varphi \mid \varphi U \psi$

其中:C,D 为概念;R_1,R_2 为角色;φ,ψ 为公式;○是一目算子,表示下一时刻;□是一目算子,表示从所指时刻起以后的所有时刻;◇是一目算子,表示从所指时刻起的某一时刻;U 是二目算子表示左式一直为真除非右式为真。

2)扩展部分的语义。

假设时间流 $\xi = \langle T, < \rangle$,其中:$T$ 表示时间点的非空集合,$<$ 表示 T 中时间点之间的一个严格的线性时序二元关系。例如,有时间点 t_1,t_2,那么二者的关系是 $t_2 < t_1, t_1 = t_2, t_1 < t_2$。

$$(\bigcirc C)^I = \{\langle t,a\rangle \mid \exists t_1.\, t \wedge \langle t_1,a\rangle \in C^I \wedge \exists \neg\, t_2.\, t < t_2 < t_1\}$$

$$(\Diamond C)^I = \{\langle t,a\rangle \mid \exists t_1.\, t \leqslant t_1 \wedge \langle t_1,a\rangle \in C^I\}$$

$$(\Box C)^I = \{\langle t,a\rangle \mid \forall t_1.\, t \leqslant t_1 \wedge \langle t_1,a\rangle \in C^I\}$$

$$(CUD)^I = \{\langle t,a\rangle \mid \exists t_1.\, t < t_1 \wedge \langle t_1,a\rangle \in D^I \wedge \forall t_2.\, t < t_2 < t_1 \rightarrow \langle t_2,a\rangle \in C^I\}$$

2.2　定义与公理

定义 1　时序描述逻辑的符号。

1)基本符号。① 概念名 $C_0,C_1,\cdots$。② 角色名 $R_0,R_1,\cdots$。③ 对象名 $a_0,a_1\cdots$。④ 概念构造子 $\cup$, $\cap$, $\forall R_i.C_i$, $\exists R_i.C_i$, $\geqslant nR_i.C_i$, $\leqslant nR_i.C_i$。其中:R_i 代表任意一角色名,C_i 代表任一概念名,n 代表正整数。⑤ 逆反角色构造符 $^-$。

2)基本逻辑构造符:$\neg$, $\wedge$, $\vee$。

3)时序逻辑构造符:○,□,◇,U。

定义 2　基本概念。在时序描述逻辑中,每个概念名以及 $\bot$(底)和 $\top$(顶)是一个原子概念,每个角色名是一个原子角色。设 C,D 是概念,R 是角色,φ 和 ψ 是公式,则:

1)$\neg C, C \wedge D, C \vee D, \forall R.C, \exists R.C, \geqslant nR.C, \leqslant nR.C, \bigcirc C, \Diamond C, \Box C, CUD$ 是概念;

2)$R ::= r \mid r^- \mid \top$ 是角色,r 是角色;

3)$\neg\varphi, \bigcirc\varphi, \Diamond\varphi, \Box\varphi, \varphi U\psi$ 是公式。

定义 3　公理如下:

$\vdash \Diamond C \equiv \neg \Box \neg C$

$\vdash \Box(C \supset D) \supset (\Box C \supset \Box D)$

$\vdash \bigcirc \neg C \equiv \neg \bigcirc C$

$\vdash \bigcirc(C \supset D) \supset (\bigcirc C \supset \bigcirc D)$

$\vdash \Box C \supset C \wedge \bigcirc C \wedge \bigcirc \Box C$

$\vdash \Box(C \supset \bigcirc C) \supset (C \supset \Box C)$

$\vdash \Box C \supset CUD$

$\vdash CUD \equiv D \vee (C \wedge (CUD))$

其中 C,D 可以表示概念、角色、对象、公式或者它们的组合。

定义 4　模型可满足性。时序描述逻辑模型 $M = \langle \Im, <, I\rangle$,其中:$<$ 表示 $\Im$ 中严格的时序关系;I 表示一个解释函数,对任意 $i \in \Im$,都有 $I(i) = \langle \Delta, R_0^{I(i)}, \cdots, C_0^{I(i)}, \cdots, a_0^{I(i)}, \cdots\rangle$;非空集合 Δ 表示模型 M 的论域;$R^{I(i)}, C^{I(i)}, a^{I(i)}$ 分别解释为论域 Δ 的二元关系、子集和一个元素,对任意概念 C,D 和公式 φ,ψ 有:

1)$(M,i) \models C = D$ 当且仅当 $C^{I(i)} = D^{I(i)}$;

2)$(M,i) \models a:C$ 当且仅当 $a^{I(i)} \in C^{I(i)}$;

3)$(M,i) \models aRb$ 当且仅当 $a^{I(i)} R^{I(i)} b^{I(i)}$;

4)$(M,i) \models \varphi U \psi$ 当且仅当 $\exists j > i$ 时,有 $(M,j) \models \psi$,且 $\forall k.\, (i \leqslant k < j \rightarrow (M,k) \models \varphi)$;

5)$(M,i) \models \bigcirc\varphi$ 当且仅当 $(M,i+1) \models \varphi$;

6)$(M,i) \models \Diamond\varphi$ 当且仅当 $\exists j > i$ 时,有 $(M,j) \models \varphi$;

7)$(M,i) \models \Box\varphi$ 当且仅当 $\forall j > i$ 时,都有 $(M,j) \models \varphi$;

8)$(M,i) \models \varphi \rightarrow \psi$ 当且仅当 $(M,i) \models \varphi \rightarrow (M,i) \models \psi$。

定义 5　时序描述逻辑表和完整树。时序描述逻辑表 (T) 是一个三元组 $T = (S,L,\varepsilon)$,其中:S 是个体的集合;L 是 S 中个体映射的子集;ε 是把角色映射为个体对之间的集合。构造时序描述逻辑概念 (C) 表达式的模型可以证明其可满足性,称这个构造模型为一棵完整树,模型中的个体对应树的节点,完整树的节点是个体集合,边是角色名集合。对于某个节点 x,若存在一个概念 C 有 $\{C, \neg C\} \subseteq L(x)$,则称冲突。

定理 1　判定性。时序描述逻辑是指数时间的完全问题,且是可判定的。

证明　已知目前基本的描述逻辑是可判定的,且它的推理算法是指数时间的完全问题[12]。扩展后的描述逻辑在其基础上引入了时序逻辑操作符,参照文献[13]中给出的相应 tableau 算法推理规则可知,利用此算法同样可以构造一棵完全的、无冲突的完整树。由于在推理规则中,"∪规则"是唯一的不确定规则,所以该树至多是一棵完全的二叉树,即可以在指数时间内完成;它通过有限次数的运用推理规则,就可以依次遍历树中的每一个节点,因为树的遍历过程的迭代次数有限,所以该遍历过程是确定可以停止的,即是可判定的。综上所述,扩展后的描述逻辑是指数时间的完全问题,且是可判定的。　证毕。

定理 2　推理问题。时序描述逻辑的推理问题都可以规约为可满足的问题。

证明　目前基本描述逻辑的推理问题都可以规约为可满足的问题,它们具有包含性、一致性、等价性和实例检测的性质。由定义 3 和定义 4 知,扩展后的描述逻辑的时序扩展部分同样满足上述的四个性质,如包含性 $(\Box(C \supset D) \supset (\Box C \supset \Box D)$、等价性 $(\Diamond C \equiv \neg \Box \neg C)$ 等性质,且在定义 4 中给出了时序扩展后描述逻辑的部分公式需满足模型 (M) 的理论。综上所述,时序描述逻辑的推理问题也可以规约为可满足问题。　证毕。

定理 3　完备性。若时序描述逻辑的概念 (C) 存在一个对应的时序描述逻辑表 (T),则根据时序描述逻辑推理算法而恰当运用相应的算法规则,就可以生成时序描述逻辑概念 (C) 的一棵完全的、无冲突的树。

证明　由定理 1 可知,时序描述逻辑是指数时间的完全问题,且是可判定的。因此通过借助概念 (C) 的时序描述逻辑表 (T) 而恰当有效地使用算法推理规则,则该算法最终一定

会结束,进而可以得到一棵完全的完整树,且与对应的时序描述逻辑表(T)是无冲突的。 证毕。

定理3的证明主要是参照文献[14]中对描述逻辑(attributive concept description language with complements, ALC)完备性的证明。

3 OWL-S过程模型的动态语义

3.1 原子过程的动态语义

对每个原子过程进行如下的定义$AP=(I,O,P,E)$,它和单个Web服务相互对应,其中:$I=(i_1,i_2,\cdots,i_n)$,表示输入的参数;$O=(o_1,o_2,\cdots,o_n)$,表示输出的参数;$P=(p_1,p_2,\cdots,p_n)$,表示该过程执行的前提条件;$E=(e_1,e_2,\cdots,e_n)$,表示该过程执行后产生的影响。当原子过程执行时,数据流(输入和输出)将得到明确描述。

$I=i_1\wedge i_2\wedge\cdots\wedge i_n$,若该原子过程没有输入参数,则$I$为true。

$O=o_1\wedge o_2\wedge\cdots\wedge o_n$,若该原子过程没有输出参数,则$O$为true。

$P=p_1\wedge p_2\wedge\cdots\wedge p_n$,若该原子过程无执行的前提条件,则$P$为true。

$E=e_1\wedge e_2\wedge\cdots\wedge e_n$,若该原子过程没有说明执行的效果,则$E$为true。

故此该原子过程的动态语义如下:

$$I\wedge P\rightarrow\bigcirc\Diamond O\wedge E$$

3.2 组合过程的动态语义

组合过程是通过各种控制构造算子对原子过程或组合过程进行再次组合,并利用控制构造算子,规定其子过程之间具有的数据与功能的依赖关系。本文在描述组合过程动态语义的时候,重点是描述控制构造算子相对应的控制流结构。因此,本文忽略Web服务组合内部具体的执行细节,例如组合过程的输入参数I,输出参数O,执行的前提条件P和执行的效果E等数据流信息。对于一个复杂的组合过程,将它的每个原子过程或子组合过程视为一个最小粒度的操作单位,只描述它们之间的控制结构关系,从而实现利用原子过程和逻辑操作符描述复杂的组合过程,最终获得其相应的动态语义。

OWL-S过程模型常见的8种控制结构利用时序描述逻辑描述如下。

1)Sequence结构。

$S_1,S_2,\cdots,S_n$是顺序执行的原子过程或组合过程,首先执行S_1,然后依次顺序执行后续的各个过程。

$$\text{Sequence}(S_1,S_2,\cdots,S_n)::=S_1\rightarrow\bigcirc\Diamond S_2\rightarrow\cdots\rightarrow\bigcirc\Diamond S_n$$

2)If-then-Else结构。

S_1,S_2,S_3是三个原子过程或组合过程,$guard$为控制条件,当执行S_1,若$guard$成立,则执行S_2;反之则执行S_3。

$$\text{If-then-Else}(S_1,S_2,S_3)::=(S_1\wedge guard\rightarrow\bigcirc\Diamond S_2)\vee(S_1\wedge\neg\ guard\rightarrow\bigcirc\Diamond S_3)$$

3)Choice结构。

$S,S_1,\cdots,S_n$是原子过程或组合过程,当执行S时,若控制条件$guard_1$成立则执行S_1;若控制条件$guard_i(1\leqslant i\leqslant n)$成立则执行$S_i$,在执行过程中只能选择一个过程执行。

$$\text{Choice}(S,S_1,\cdots,S_{n-1},S_n)::=(S\wedge guard_1\rightarrow\bigcirc\Diamond(S_1))\vee\cdots\vee(S\wedge guard_{n-1}\rightarrow\bigcirc\Diamond(S_{n-1}))\vee(S\wedge guard_n\rightarrow\bigcirc\Diamond(S_n))$$

其中:

$$guard_1\wedge guard_2\wedge\cdots\wedge guard_{n-1}\wedge guard_n=\text{false}$$

$$guard_1\vee guard_2\vee\cdots\vee guard_{n-1}\vee guard_n=\text{true}$$

4)Split结构。

$S_1,S_2,\cdots,S_n$是原子过程或组合过程,首先执行S_1,然后并发执行过程$S_2,\cdots,S_n$,只有过程$S_2,\cdots,S_n$全部执行结束后才能继续执行后续的过程。

$$\text{Split}(S_1,S_2,\cdots,S_n)::=S_1\rightarrow\bigcirc\Diamond(S_2\wedge\cdots\wedge S_n)$$

5)Split+Join结构。

$S_1,\cdots,S_{n-1},S_n$是原子过程或组合过程,首先是过程$S_1,\cdots,S_{n-1}$并发执行,只有$S_1,\cdots,S_{n-1}$全部执行结束后才能执行S_n。

$$\text{Split+Join}(S_1,\cdots,S_{n-1},S_n)::=(S_1\wedge\cdots\wedge S_{n-1})\rightarrow\bigcirc\Diamond S_n$$

6)Any-Order结构。

S_1,S_2是两个原子过程或组合过程,在执行过程中二者并没有确定的先后执行顺序,即可以先执行S_1,然后执行S_2;反之亦可,但是在执行过程中若选择某一执行顺序开始执行,则二者顺序就不能再次改变。

$$\text{Any-Order}(S_1,S_2)::=\bigcirc\Box((S_1\rightarrow\bigcirc\Diamond S_2)\vee(S_2\rightarrow\bigcirc\Diamond S_1))$$

7)Repeat-While结构。

S_1,S_2是两个原子过程或组合过程,若$guard$控制条件成立,则S_1不停地自我循环执行,直至$guard$不成立,然后执行S_2。

$$\text{Repeat-While}(S_1,S_2)::=\bigcirc\Box(guard\wedge S_1\rightarrow\bigcirc\Diamond S_1)U(\neg\ guard\wedge S_1\rightarrow\bigcirc\Diamond S_2)$$

8)Repeat-Until结构。

S_1,S_2是两个原子过程或组合过程,$guard$为控制条件,与Repeat-While执行顺序基本相同,不同之处在于它要首先执行一次S_1,然后进行判断,若$guard$条件成立,S_1不停地自我循环执行,直至$guard$条件不成立,然后执行S_2。

$$\text{Repeat-Until}(S_1,S_2,S_3)::=(S_1\rightarrow\bigcirc\Diamond S_1)\rightarrow\bigcirc\Diamond(\bigcirc\Box(guard\wedge S_1\rightarrow\bigcirc\Diamond S_1)U(\neg\ guard\wedge S_1\rightarrow\bigcirc\Diamond S_2))$$

4 案例分析

以来自OWL-S网站的一个购书服务实例——CongoProcess.owl[15]文件中的服务组合过程FullCongoBuy为例,说明如何利用时序描述逻辑描述该组合过程的动态语义。该组合过程具体执行顺序:首先是购买者判断需要买的图书在Stock中是否有,如果没有,则该组合过程结束;反之,则把需要的图书放进购物筐;然后是对购买者使用目前用户账户或新建用户的账户进行选择;进而是登录用户的个人账户;接着是依次选择付款方式和选择送货类型;最后是图书的购买成功,至此组合过程结束。

在上述组合过程中,$guard_1$是If-then-Else结构中用户需要的书在Stock中存在的条件;$guard_2$是Choice结构中选择新建账户的条件;$guard_3$是Choice结构中选择目前账户的条件;End表示组合过程结束。

本实例中每个组合过程都是由相应的原子过程或子组合过程组成,每个组合过程的动态语义如下:

$$\text{FullCongoBuy}\equiv\text{Sequence}(\text{LocateBook},\text{If}(guard_1)\text{then}(\text{CongoBuyBook})\text{Else}(\text{End}))$$

$$\text{FullCongoBuy}::=(\text{LocateBook}\wedge guard_1\rightarrow\bigcirc\Diamond\text{CongoBuyBook}\rightarrow\bigcirc\Diamond\Box\text{End})\vee(\text{LocateBook}\wedge\neg\ guard_1\rightarrow\bigcirc\Diamond\Box\text{End})$$

$$\text{CongoBuyBook}\equiv\text{Sequence}(\text{Buysequence},\text{SpecifyDeliveryDetail},\text{FinallyBuy})$$

$$\text{CongoBuyBook}::=\text{Buysequence}\rightarrow$$

○◇SpecifyDeliveryDetail→○◇FinallyBuy

Buysequence≡Sequence(PutInCart, SignInAlternatives, SpecifyPaymentMethod)

Buysequence ::= PutInCart→○◇SignInAlternatives→○◇SpecifyPaymentMethod

SignInAlternatives≡Choice(CreateAcctSequence, SignInSequence)

SignInAlternative ::= (PutInCart∧$guard_2$→○◇CreateAcctSequence)∨(PutInCart∧$guard_3$→○◇SignInSequence)

CreateAcctSequence≡Sequence(CreateAcct, LoadUserProfile)

CreateAcctSequence ::= CreateAcct→○◇LoadUserProfile

SignInSequence≡Sequence(SignIn, LoadUserProfile)

SignInSequence ::= SignIn→○◇LoadUserProfile

利用时序描述逻辑来形式化描述组合过程的动态语义,可以将其置于严格可判定推理的框架之下,讨论其性质。

以查询 Stock 中有书,且使用当前用户的账户信息,最终付款购书成功的组合过程为例,该组合过程动态语义如下:

○□(LocateBook∧$guard_1$→○◇((PutInCart∧$guard_3$→○◇(SignIn→○◇LoadUserProfile)→○◇SpecifyPaymentMethod)→○◇SpecifyDeliveryDetail→○◇FinallyBuy)→○◇□End)

5　结语

本文提出的利用时序描述逻辑描述 OWL-S 的原子过程和组合过程动态语义的方法,增强了描述逻辑本身的表达能力,不仅使其可以表示 Web 服务组合的静态特征和动态时序特征,而且在可判定和推理的前提下,提高了对 OWL-S 过程模型描述的准确性和精确性,为最终实现 Web 服务组合的自动化分析与验证提供了理论基础。下一步的工作是充分考虑过程模型中参数等其他信息,对 OWL-S 过程模型的动态语义做进一步完善的工作。

参考文献:

[1] 鲍爱华,王晓璇,文艾,等. 基于扩展 CPN 的 OWL-S 过程语义建模及分析方法研究[J]. 计算机科学,2011,38(4):203 - 204.

[2] 史忠植,常亮. 基于动态描述逻辑的语义 Web 服务推理[J]. 计算机学报,2008,31(9):1599 - 1611.

[3] WANG H B, ZHOU Q Z, SHI Y Q. Describing and verifying Web service composition using TLA reasoning [C]// SCC 2010: Proceedings of the 7th IEEE International Conference on Services Computing. Washington, DC: IEEE Computer Society, 2010: 234 - 241.

[4] 沈雅芬,黄宁,彭永义. OWL-S 模型转化为重写逻辑模型的方法[J]. 计算机应用,2011,31(6):1491 - 1494.

[5] 马炳先,杜玉越. OWL-S 服务操作语义的 Petri 网描述新方法[J]. 系统仿真学报,2007,19(S1):69 - 74.

[6] 刘大有,刘思培,齐红. 基于 SROIQB 的语义 Web 服务建模和组合[J]. 通信学报,2010,31(8A):1 - 9.

[7] MARTIN D, BURSTEIN M, HOBBS J, *et al*. OWL-S: semantic markup for Web services [EB/OL]. [2012-04-15]. http://www.w3.org/submission/owl-s.

[8] The OWL-S Coalition. OWL-S1.2 release. 2004 [EB/OL]. [2012-04-15]. http://www.daml.org/services/owl-s/1.2/.

[9] BAADER F, CALVANESE D, McGUINESS D, *et al*. The description logic handbook: theory, implementation, and applications [M]. Cambridge: Cambridge University Press, 2003.

[10] LUTZ C, WOLTER F, ZAKHARYASCHEV M. Temporal description logics: a survey [C]// Proceedings of the 15th International Symposium on Temporal Representation and Reasoning. Washington, DC: IEEE Computer Society, 2008: 3 - 14.

[11] BAADER F, BAUER A, LIPPMANN M. Runtime verification using a temporal description logic [C]// Proceedings of the 7th International Conference on Frontiers of Combining Systems, LNCS 5749. Berlin: Springer-Verlag, 2009: 149 - 164.

[12] HORROCKS I, SATTLER U. A tableau decision procedure for SHOIQ [J]. Journal of Automated Reasoning, 2007, 39(3): 249 - 276.

[13] STURM H, WOLTER F. A tableau calculus for temporal description logic: the expanding domain case [J]. Journal of Logic and Computation, 2002, 12(5): 809 - 838.

[14] 梅婧,林作铨. 从 ALC 到 SHOQ(D):描述逻辑及其 Tableau 算法[J]. 计算机科学,2005,32(3):1 - 11.

[15] MARRIN D, BURSTEIN M, McDERMOTT D, *et al*. OWL-S1.2 release [EB/OL]. [2012-04-15]. http://www.ai.sri.com/daml/services/owl-s/1.2/CongoProcess.owl.

(上接第 265 页)

广泛的 Petri 网建模,通过在有向边上增加变迁函数而省略了不必要的变迁元素,非常适用于大型工程项目的工作流建模,是一种轻量级的工作流模型。

针对本文提出的工作流模型,未来的研究工作包括:业务过程的回放及流程绩效分析;回退的业务补偿处理;针对 AOV 网的有向图本质,研究模型的关键路径算法,便于项目管理人员对项目实施进度的总体把握及合理资源分配。

参考文献:

[1] 胡奇. jBPM4 工作流应用开发指南[M]. 北京:电子工业出版社,2010:2 - 9.

[2] 袁崇义. Petri 网原理与应用[M]. 北京:电子工业出版社,2005.

[3] van der AALST W, van HEE K. Workflow management models, methods and systems [M]. Cambridge: MIT Press, 2002.

[4] LEE H, SUH H W. Workflow structuring and reengineering method for design process [J]. Computers and Industrial Engineering, 2006, 51(4): 698 - 714.

[5] 殷人昆. 数据结构(用面向对象方法与 C++语言描述)[M]. 2 版. 北京:清华大学出版社,2007:383 - 387.

[6] HOROWITZ E, SAHNI S, ANDERSON-FREED S. 数据结构基础(C 语言版)[M]. 朱仲涛,译. 2 版. 清华大学出版社,2009:205 - 247.

[7] WFMC. Workflow process definition interface: XML Process Definition Language (XPDL), WFMC-TC-1025 [EB/OL]. [2012-04-23]. http://xml.coverpages.org/XPDL20010522.pdf.

[8] 李圣文,龚君芳. 基于模糊理论的工作流调度模型研究[J]. 计算机应用研究,2010,27(1):131 - 133.

[9] 杨雯,刘厚泉. 基于 Petri 网的工作流模型的研究[J]. 计算机工程与设计,2007,28(17):4149 - 4164.

[10] 周建涛,史美林,叶新铭. 工作流过程建模中的形式化验证技术[J]. 计算机研究与发展,2005,42(1):1 - 9.

[11] 余阳,汤庸,潘茂林,等. 时态工作流过程模型及其合理性验证[J]. 软件学报,2010,21(6):1233 - 1253.

[12] 王霞,王刚,周立东. 改进的工作流合理性验证方法[J]. 计算机工程与应用,2011,47(33):43 - 45.

[13] 高捷,吴华瑞. 基于矩阵模型的工作流合理性验证算法研究[J]. 计算机工程与设计,2010,31(11):2621 - 2624.