

OWL-S 的形式语义^{*})

蒋运承¹ 史忠植²

(广西师范大学数学与计算机科学学院 桂林 541004)¹ (中国科学院计算技术研究所 北京 100080)²

摘要 本文分析了目前语义 Web 服务的研究现状和存在的问题,特别是语义 Web 服务描述本体 OWL-S 的形式语义研究中存在的问题,在 Srinu Narayanan 等人研究的基础上,用情景演算理论进一步研究了 OWL-S 中组合服务描述的形式语义,从而完善了 OWL-S 的形式语义,为语义 Web 服务提供了合理的理论基础。

关键词 语义 Web 服务, OWL-S, 情景演算, 形式语义

The Formal Semantics of OWL-S

JIANG Yun-Cheng¹ SHI Zhong-Zhi²

(College of Mathematics and Computer Sciences, Guangxi Normal University, Guilin 541004)¹

(Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100080)²

Abstract The current work and the existing problem of the semantic Web services are analyzed in this paper, especially the formal semantics of semantic Web services description ontology OWL-S. Based on the work of Srinu Narayanan, et al, the formal semantics of service composition description of OWL-S is further studied through situation calculus theory, so the formal semantics of OWL-S is perfected. The work of this paper provides reasonable theory foundation for semantic Web services.

Keywords Semantic Web services, OWL-S, Situation calculus theory, Formal semantics

1 引言

万维网(WWW)业已成为人们获得信息、取得服务的重要渠道之一。但是,目前万维网的页面设计仅仅针对人类自身,适合人类自身的使用,不便于计算机的自动处理和理解。语义 Web 是对万维网本质的变革,它研究的主要目的就是扩展当前的万维网,使得网络中的信息具有语义,能够被计算机理解,便于人和计算机之间的交互与合作^[1]。Web 服务(Web Services)作为一种新兴的 Web 应用模式,是一个崭新的分布式计算模型,是 Web 上数据和信息集成的有效机制^[2]。但是 Internet 的资源局限于静态的数据和信息,随着信息技术的发展和用户新的需要,越来越多的资源通过服务封装以后,都可以而且都需要通过 Internet 访问得到,从而如何将语义 Web 技术和 Web 服务技术结合起来,通过 Web 服务的语义封装,为用户提供自动的 Web 服务,即语义 Web 服务已成为一个重要的研究课题,它研究的根本任务就是对 Web 服务进行标注,使 Web 服务成为计算机可理解的、用户透明的和主体易处理的实体^[3,4]。

目前国际上对语义 Web 服务进行了深入研究,但主要集中在语义 Web 服务的应用研究上,包括基于语义的服务描述、匹配、组合、监控等,以及语义 Web 服务规范的制订等。最典型的研究是 DARPA 组织的研究组 OWL Services Coalition 提出了语义 Web 服务本体 OWL-S,它的目标是使得 Web 服务成为计算机可以理解的实体,从而计算机可以为用户自动完成如下任务:服务发现、调用、组合、验证、互操作及执行监控等^[5]。

由于 OWL-S 在语义 Web 服务中的重要性,一些学者对它的理论基础进行了研究,因为研究它的理论基础是服务执行模型和服务验证的基础,也为实际应用中建构语义 Web 服务提供了必要的理论基础。Anupriya Ankolekar 等人根据 Erlang 程序理论和并发 Haskell 程序理论研究了 DAML-S 的操作语义,从理论上阐述了子类多态问题^[6]。Srinu Narayanan 等人利用情景演算理论和 Petri 网理论研究了 OWL-S 的操作语义^[7],首先研究了原子服务描述与情景演算之间的转换关系,即用情景演算理论研究了 OWL-S 中原子服务描述的形式语义。在此基础上,用情景演算作为中间语言,将情景演算转换为 Petri 网,然后利用 Petri 网理论研究了语义 Web 服务的执行推理,即语义 Web 服务的操作语义。但 Srinu Narayanan 等人没有研究 OWL-S 中组合服务描述的形式语义。

基于上述原因,本文在 Srinu Narayanan 等人的基础上,用情景演算理论进一步研究 OWL-S 中组合服务描述的形式语义,从而完善 OWL-S 的形式语义,为语义 Web 服务提供合理的理论基础。

2 OWL-S

OWL-S 是 DARPA 组织的研究组 OWL Services Coalition 提出的基于本体语言 OWL 的语义 Web 服务本体,它的前身是 DAML-S,而 DAML-S 是基于本体语言 DAML+OIL 的语义 Web 服务本体。OWL-S 的目标是使得 Web 服务成为计算机可以理解的实体,从而计算机可以为用户自动完成如下任务:服务发现、调用、组合、验证、互操作及执行监控等。图

^{*})国家 863 项目(No. 2003AA115220)和广西师范大学博士启动基金项目资助。蒋运承 博士,主要研究领域为语义 Web、语义 Web 服务和多主体系统。史忠植 研究员,博士生导师,主要研究领域为人工智能、机器学习和分布式人工智能等。

1 是 OWL-S 的顶层服务本体^[5]。

从图 1 可知,在 OWL-S 中服务描述的基本信息主要有三种本体,其中 ServiceProfile 描述服务做什么,它给出服务查询代理需要用来判断服务是否适合它的要求的一组信息。ServiceModel 描述服务如何工作和服务具体的逻辑执行顺序。ServiceGrounding 描述了如何访问服务,即描述访问服务的通信协议及其它细节。

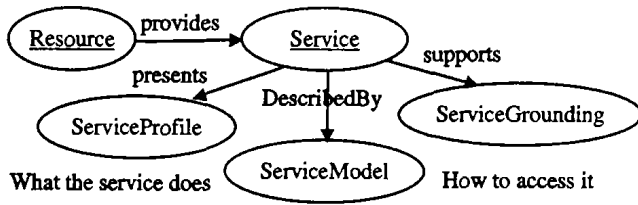


图 1 OWL-S 的顶层服务本体

ServiceProfile 本体主要描述服务和提供者信息,包括:(1)服务提供者的信息,如联系信息等。(2)服务的功能信息,包括输入和输出参数说明、参数信息、条件信息及效果信息,其中 Precondition 说明请求服务前必须满足的逻辑前提条件,Effect 说明完成服务以后对外界环境所激发的效果。(3)服务属性,包括服务分类、附加参数、安全质量、服务完成时间、服务代价、服务质量和地理范围等。通过 ServiceProfile 本体,服务提供者可以提供自己的服务说明,而服务查询代理可以通过服务提供的 ServiceProfile 信息来判断是否满足服务请求者的需求。

ServiceModel 本体主要有两个模块:Process 本体和 ProcessControl 本体,其中 Process 本体以流程的形式描述了服务中各子过程的执行顺序,ProcessControl 本体主要管理和监控服务流程的执行。目前 OWL Services Coalition 还没有给出 ProcessControl 本体的规范。Process 本体的最基本的实体是过程(Process)。

Process 又分为原子过程、简单过程和组合过程。其中原子过程是一个直接可以调用的过程,它没有子过程,从服务请求者的角度来看,原子过程能通过一步完成。简单过程不是一个可直接调用的过程,但与原子过程类似,简单过程也能通过一步完成。简单过程用作抽象元素,它既可以用作原子过程的视图,又可以用作组合过程的简化表示。组合过程可以分解成其它的过程(包括组合过程和非组合过程),是通过控制结构将原子过程或组合过程组合起来的一个复杂过程。目前 OWL-S 有 Sequence、Split、Condition、Split + Join、Unordered、Choice、Iterate、If-Then-Else、Repeat-Until 和 Repeat-While 等控制结构构造符。

ServiceGrounding 本体用于访问服务的具体细节,如服务的 URI 地址、传输协议、消息格式等,ServiceGrounding 可以看成是服务抽象描述到具体服务的映射过程。在 OWL-S 中,ServiceProfile 和 ServiceModel 是服务的抽象表示,而 ServiceGrounding 描述的是服务的具体信息。

3 OWL-S 的形式语义

由上面可以看出,OWL-S 是一个用来描述语义 Web 服务的本体,是对服务的非形式化描述。与 OWL 类似,有必要研究 OWL-S 的形式语义,即用一种形式化语言(如一阶逻辑)来刻画 OWL-S 的服务描述,因为研究它的形式语义是服务执行模型和服务验证的基础,也为实际应用中建构语义

Web 服务提供了必要的理论基础。下面介绍利用情景演算理论来刻画 OWL-S 的形式语义。

情景演算理论是一门一阶逻辑语言,主要是用来表示动态变化世界的。世界的所有变化过程都是“动作”的结果^[6]。一个可能世界的历史可以简单表示为动作的序列,它是通过称之为情景的一阶项所表示的。常量 S_0 表示初始情景,即动作还没有发生时的情景。二元函数符号 do 是一个特殊的函数符号, $do(a, s)$ 表示在情景 s 下执行动作(之后的后继情景。动作是可以带参数的。随着情景变化而改变值的函数或关系称为流(fluent),它用最后一个参数是情景项的谓词符号来表示。 $Poss(a, s)$ 是一个特殊的流,表示在情景 s 下完成动作 a 是可能的。

本文要用情景演算理论来刻画服务(原子服务或组合服务)描述的形式语义,因而该情景演算理论必须是带认知流的情景演算理论^[6],包括下列认知流:一个特殊的流 $K(s, s')$ 表示情景 s 和 s' 之间的认知可达关系; $Knows(\varphi, s)$ 表示在情景 s 下公式 φ 是知道的,即 $Knows(\varphi, s) \equiv \forall s', K(s, s') \rightarrow \varphi(s')$, $Kwhether(\varphi, s)$ 表示在情景 s 下公式 φ 的真值是知道的,即 $Kwhether(\varphi, s) \equiv Knows(\varphi, s) \vee Knows(\neg\varphi, s)$; $Kref(\varphi, s)$ 表示在情景 s 下公式 φ 的函数值是知道的,即 $Kref(\varphi, s) \equiv \exists x, Knows(\varphi = x, s)$,其中 x 在 φ 中不出现。

3.1 原子服务描述的形式语义

Srini Narayanan 等人利用情景演算理论研究了 DAML-S(OWL-S 的前身)中原子服务描述的形式语义,用条件效果、条件输出、前提和输入来刻画原子服务描述的形式语义^[7],它们的形式化描述如下:

① 原子服务描述 φ 的条件效果用情景演算理论中的正效果公理和负效果公理的形式表示,即 $Poss(\varphi, s) \wedge \gamma F^+(x, \varphi, s) \rightarrow F(x, do(\varphi, s))$ 和 $Poss(\varphi, s) \wedge \gamma F^-(x, \varphi, s) \rightarrow \neg F(x, do(\varphi, s))$,其中 $\gamma F^{(+/-)}(x, \varphi, s)$ 分别表示完成服务 φ 以后使得流 F 真/假的不同条件的组合。

② 原子服务描述 φ 的条件输出可以用情景演算的认知流 $Knows, Kwhether$ 或 $Kref$ 表示,即 $Poss(\varphi, s) \wedge \gamma(x, \varphi, s) \rightarrow (Knows(y, do(\varphi, s)) \vee Kwhether(y, do(\varphi, s)) \vee Kref(y, do(\varphi, s)))$,其中 $\gamma(x, \varphi, s)$ 表示完成服务 φ 以后使知道 y, y 的真值或 y 的函数值的不同条件的组合。

③ 原子服务描述 φ 的前提可以用情景演算中的合适公式来表示,即 $Poss(\varphi, s) \rightarrow \pi_1 \wedge \pi_2 \wedge \dots \wedge \pi_n$ 。

④ 原子服务描述 φ 的输入用情景演算的认知流 $Kref$ 表示,即对服务 φ 的每一个输入 φ_i ,有 $Poss(\varphi, s) \rightarrow Kref(\varphi_i, s) \wedge \dots \wedge Kref(\varphi_n, s)$ 。

3.2 组合服务描述的形式语义

组合服务描述 φ 可以等价的看作是将服务 φ 分解为 $\varphi_1, \dots, \varphi_n$,而 $\varphi_1, \dots, \varphi_n$ 的执行顺序是由 OWL-S 的组合算子来控制的,因而可以借鉴逻辑程序设计语言 Golog/ConGolog^[9]来刻画组合服务描述的形式语义。下面仅以 OWL-S 中的顺序 Sequence 控制算子、Condition 算子、If 条件控制算子、Unordered 算子、Repeat-Until 循环算子和 Repeat-While 循环算子为例来说明,其它算子可以类似进行刻画。

· 通过顺序控制算子 Sequence 来组合的服务描述 $\varphi = Sequence(\varphi_1, \varphi_2)$ 的形式语义为:

① 组合服务描述 φ 的条件效果形式公理为 $Poss(\varphi_1, s) \wedge Poss(\varphi_2, do(\varphi_1, s)) \wedge \gamma F^+(x, \varphi_2, do(\varphi_1, s)) \rightarrow F(x, do(\varphi_2, do(\varphi_1, s)))$ 和 $Poss(\varphi_1, s) \wedge Poss(\varphi_2, do(\varphi_1, s)) \wedge \gamma F^-(x, \varphi_2, do$

$(\varphi_1, s) \rightarrow \neg F(x, do(\varphi_2, do(\varphi_1, s)))$;

② 组合服务描述 φ 的条件输出的形式公理为 $Poss(\varphi_1, s) \wedge Poss(\varphi_2, do(\varphi_1, s)) \wedge \gamma(x, \varphi_2, do(\varphi_1, s)) \rightarrow (Knows(y, \psi) \vee Kwhether(y, \psi) \vee Kref(y, \psi))$, 其中 $\psi = do(\varphi_2, do(\varphi_1, s))$;

③ 组合服务描述 φ 的前提的形式公式为 $Poss(\varphi_1, s) \wedge Poss(\varphi_2, do(\varphi_1, s)) \rightarrow \pi_1 \wedge \pi_2 \wedge \dots \wedge \pi_n$;

④ 组合服务描述 φ 的输入的形式公式为 $Poss(\varphi_1, s) \wedge Poss(\varphi_2, \psi) \rightarrow Kref((\varphi_{1i}, s) \wedge \dots \wedge Kref(\varphi_{1w}, s) \wedge Kref(\varphi_{2i}, \psi) \wedge \dots \wedge Kref(\varphi_{2w}, \psi))$, 其中 $\psi = do(\varphi_1, s)$ 。

• 通过 Condition 条件算子来组合的服务描述 $\varphi = Condition(\beta, \varphi_1)$ 的形式语义为:

① 组合服务描述 φ 的条件效果形式公理为 $(\beta \wedge (Poss(\varphi_1, s) \wedge \gamma_F^+(x, \varphi_1, s) \rightarrow F(x, \psi_1)))$ 和 $(\beta \wedge (Poss(\varphi_1, s) \wedge \gamma_F^-(x, \varphi_1, s) \rightarrow \neg F(x, \psi_1)))$, 其中 $\psi_1 = do(\varphi_1, s)$;

② 组合服务描述 φ 的条件输出的形式公理为 $(\beta \wedge (Poss(\varphi_1, s) \wedge \gamma(x, \varphi_1, s) \rightarrow Knows(y, \psi_1) \vee Kwhether(y, \psi_1) \vee Kref(y, \psi_1)))$, 其中 $\psi_1 = do(\varphi_1, s)$;

③ 组合服务描述 φ 的前提的形式公式为 $(\beta \wedge (Poss(\varphi_1, s) \rightarrow \pi_{11} \wedge \pi_{12} \wedge \dots \wedge \pi_{1n}))$;

④ 组合服务描述 φ 的输入的形式公式为 $(\beta \wedge (Poss(\varphi_1, s) \rightarrow Kref(\varphi_{1i}, s) \wedge \dots \wedge Kref(\varphi_{1w}, s)))$ 。

• 通过 If 条件控制算子 If-Then-Else 来组合的服务描述 $\varphi = If-Then-Else(\beta, \varphi_1, \varphi_2)$ 的形式语义为:

① 组合服务描述 φ 的条件效果形式公理为 $(\beta \wedge (Poss(\varphi_1, s) \wedge \gamma_F^+(x, \varphi_1, s) \rightarrow F(x, \psi_1))) \vee (\neg \beta \wedge (Poss(\varphi_2, s) \wedge \gamma_F^+(x, \varphi_2, s) \rightarrow F(x, \psi_2)))$ 和 $(\beta \wedge (Poss(\varphi_1, s) \wedge \gamma_F^-(x, \varphi_1, s) \rightarrow \neg F(x, \psi_1))) \vee (\neg \beta \wedge (Poss(\varphi_2, s) \wedge \gamma_F^-(x, \varphi_2, s) \rightarrow \neg F(x, \psi_2)))$, 其中 $\psi_1 = do(\varphi_1, s), \psi_2 = do(\varphi_2, s)$;

② 组合服务描述 φ 的条件输出的形式公理为 $(\beta \wedge (Poss(\varphi_1, s) \wedge \gamma(x, \varphi_1, s) \rightarrow Knows(y, \psi_1) \vee Kwhether(y, \psi_1) \vee Kref(y, \psi_1))) \vee (\neg \beta \wedge (Poss(\varphi_2, s) \wedge \gamma(x, \varphi_2, s) \rightarrow Knows(y, \psi_2) \vee Kwhether(y, \psi_2) \vee Kref(y, \psi_2)))$, 其中 $\psi_1 = do(\varphi_1, s), \psi_2 = do(\varphi_2, s)$;

③ 组合服务描述 φ 的前提的形式公式为 $(\beta \wedge (Poss(\varphi_1, s) \rightarrow (\pi_{11} \wedge \pi_{12} \wedge \dots \wedge \pi_{1n})) \vee (\neg \beta \wedge (Poss(\varphi_2, s) \rightarrow \pi_{21} \wedge \pi_{22} \wedge \dots \wedge \pi_{2n})))$;

④ 组合服务描述 φ 的输入的形式公式为 $(\beta \wedge (Poss(\varphi_1, s) \rightarrow Kref(\varphi_{1i}, s) \wedge \dots \wedge Kref(\varphi_{1w}, s))) \vee (\neg \beta \wedge (Poss(\varphi_2, s) \rightarrow Kref(\varphi_{2i}, s) \wedge \dots \wedge Kref(\varphi_{2w}, s)))$ 。

• 通过 Unordered 控制算子来组合的服务描述 $\varphi = Unordered(\varphi_1, \varphi_2)$ 的形式语义为(假设 φ_1 先执行):

① 组合服务描述 φ 的条件效果形式公理为 $Poss(\varphi_1, s) \wedge Poss(\varphi_2, do(\varphi_1, s)) \wedge \gamma_F^+(x, \varphi_2, do(\varphi_1, s)) \rightarrow F(x, do(\varphi_2, do(\varphi_1, s)))$ 和 $Poss(\varphi_1, s) \wedge Poss(\varphi_2, do(\varphi_1, s)) \wedge \gamma_F^-(x, \varphi_2, do(\varphi_1, s)) \rightarrow \neg F(x, do(\varphi_2, do(\varphi_1, s)))$; 或者 $Poss(\varphi_2, s) \wedge Poss(\varphi_1, do(\varphi_2, s)) \wedge \gamma_F^+(x, \varphi_1, do(\varphi_2, s)) \rightarrow F(x, do(\varphi_1, do(\varphi_2, s)))$ 和 $Poss(\varphi_2, s) \wedge Poss(\varphi_1, do(\varphi_2, s)) \wedge \gamma_F^-(x, \varphi_1, do(\varphi_2, s)) \rightarrow \neg F(x, do(\varphi_1, do(\varphi_2, s)))$;

② 组合服务描述 φ 的条件输出的形式公理为 $Poss(\varphi_1, s) \wedge Poss(\varphi_2, do(\varphi_1, s)) \wedge \gamma(x, \varphi_2, do(\varphi_1, s)) \rightarrow (Knows(y, \psi) \vee Kwhether(y, \psi) \vee Kref(y, \psi))$, 其中 $\psi = do(\varphi_2, do(\varphi_1, s))$; 或者 $Poss(\varphi_2, s) \wedge Poss(\varphi_1, do(\varphi_2, s)) \wedge \gamma(x, \varphi_1, do(\varphi_2, s)) \rightarrow (Knows(y, \psi) \vee Kwhether(y, \psi) \vee Kref(y, \psi))$, 其中 $\psi = do(\varphi_1, do(\varphi_2, s))$;

③ 组合服务描述 φ 的前提的形式公式为 $Poss(\varphi_1, s) \wedge Poss(\varphi_2, do(\varphi_1, s)) \rightarrow \pi_{11} \wedge \pi_{12} \wedge \dots \wedge \pi_{1n}$; 或者 $Poss(\varphi_2, s) \wedge Poss(\varphi_1, do(\varphi_2, s)) \rightarrow \pi_{21} \wedge \pi_{22} \wedge \dots \wedge \pi_{2n}$;

④ 组合服务描述 φ 的输入的形式公式为 $Poss(\varphi_1, s) \wedge Poss(\varphi_2, \psi) \rightarrow Kref((\varphi_{1i}, s) \wedge \dots \wedge Kref(\varphi_{1w}, s) \wedge Kref(\varphi_{2i}, \psi) \wedge \dots \wedge Kref(\varphi_{2w}, \psi))$, 其中 $\psi = do(\varphi_1, s)$; 或者 $Poss(\varphi_2, s) \wedge Poss(\varphi_1, \psi) \rightarrow Kref(\varphi_{2i}, s) \wedge \dots \wedge Kref(\varphi_{2w}, s) \wedge Kref(\varphi_{1i}, \psi) \wedge \dots \wedge Kref(\varphi_{1w}, \psi)$, 其中 $\psi = do(\varphi_2, s)$ 。

• 通过 Repeat-Until 循环控制算子来组合的服务描述 $\varphi = Repeat-Until(\beta, \varphi_1)$ 的形式语义为:

① 组合服务描述 φ 的条件效果形式公理为 $((Poss(\varphi_1, s) \wedge \gamma_F^+(x, \varphi_1, s) \rightarrow F(x, \psi_1))) \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_1) \wedge \gamma_F^+(x, \varphi_1, \psi_1) \rightarrow F(x, \psi_2))) \wedge \dots \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma_F^+(x, \varphi_1, \psi_{n-1}) \rightarrow F(x, \psi_n))) \wedge (\beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma_F^+(x, \varphi_1, \psi_{n-1}) \rightarrow F(x, \psi_n)))$ 和 $((Poss(\varphi_1, s) \wedge \gamma_F^-(x, \varphi_1, s) \rightarrow \neg F(x, \psi_1))) \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_1) \wedge \gamma_F^-(x, \varphi_1, \psi_1) \rightarrow \neg F(x, \psi_2))) \wedge \dots \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma_F^-(x, \varphi_1, \psi_{n-1}) \rightarrow \neg F(x, \psi_n))) \wedge (\beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma_F^-(x, \varphi_1, \psi_{n-1}) \rightarrow \neg F(x, \psi_n)))$, 其中 $\psi_1 = do(\varphi_1, s), \psi_2 = do(\varphi_1, do(\varphi_1, s)), \psi_3 = do(\varphi_1, do(\varphi_1, do(\varphi_1, s)))$ 等;

② 组合服务描述 φ 的条件输出的形式公理为 $((Poss(\varphi_1, s) \wedge \gamma(x, \varphi_1, s) \rightarrow Knows(y, \psi_1) \vee Kwhether(y, \psi_1) \vee Kref(y, \psi_1))) \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_1) \wedge \gamma(x, \varphi_1, \psi_1) \rightarrow Knows(y, \psi_2) \vee Kwhether(y, \psi_2) \vee Kref(y, \psi_2))) \wedge \dots \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma(x, \varphi_1, \psi_{n-1}) \rightarrow Knows(y, \psi_n) \vee Kwhether(y, \psi_n) \vee Kref(y, \psi_n))) \wedge (\beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma(x, \varphi_1, \psi_{n-1}) \rightarrow Knows(y, \psi_n) \vee Kwhether(y, \psi_n) \vee Kref(y, \psi_n)))$, 其中 $\psi_1 = do(\varphi_1, s), \psi_2 = do(\varphi_1, do(\varphi_1, s)), \psi_3 = do(\varphi_1, do(\varphi_1, do(\varphi_1, s)))$ 等;

③ 组合服务描述 φ 的前提的形式公式为 $((Poss(\varphi_1, s) \rightarrow (\pi_{11} \wedge \pi_{12} \wedge \dots \wedge \pi_{1n})) \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_1) \rightarrow \pi_{21} \wedge \pi_{22} \wedge \dots \wedge \pi_{2n})) \wedge \dots \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_{n-1}) \rightarrow \pi_{n1} \wedge \pi_{n2} \wedge \dots \wedge \pi_{nn})) \wedge (\beta \wedge (Poss(\varphi_1, \psi_{n-1}) \rightarrow \pi_{n1} \wedge \pi_{n2} \wedge \dots \wedge \pi_{nn}))$, 其中 $\psi_1 = do(\varphi_1, s), \psi_2 = do(\varphi_1, do(\varphi_1, s)), \psi_3 = do(\varphi_1, do(\varphi_1, do(\varphi_1, s)))$ 等;

④ 组合服务描述 φ 的输入的形式公式为 $((Poss(\varphi_1, s) \rightarrow Kref(\varphi_{1i}, s) \wedge \dots \wedge Kref(\varphi_{1w}, s))) \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_1) \rightarrow Kref(\varphi_{2i}, \psi_1) \wedge \dots \wedge Kref(\varphi_{2w}, \psi_1))) \wedge \dots \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_n) \rightarrow Kref(\varphi_{2i}, \psi_n) \wedge \dots \wedge Kref(\varphi_{2w}, \psi_n))) \wedge (\beta \wedge (Poss(\varphi_1, \psi_n) \rightarrow Kref(\varphi_{2i}, \psi_n) \wedge \dots \wedge Kref(\varphi_{2w}, \psi_n)))$, 其中 $\psi_1 = do(\varphi_1, s), \psi_2 = do(\varphi_1, do(\varphi_1, s)), \psi_3 = do(\varphi_1, do(\varphi_1, do(\varphi_1, s)))$ 等。

• 通过 Repeat-While 控制算子来组合的服务描述 $\varphi = Repeat-While(\beta, \varphi_1)$ 的形式语义为:

① 组合服务描述 φ 的条件效果形式公理为 $(\beta \wedge (Poss(\varphi_1, s) \wedge \gamma_F^+(x, \varphi_1, s) \rightarrow F(x, \psi_1))) \wedge (\beta \wedge (Poss(\varphi_1, \psi_1) \wedge \gamma_F^+(x, \varphi_1, \psi_1) \rightarrow F(x, \psi_2))) \wedge \dots \wedge (\beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma_F^+(x, \varphi_1, \psi_{n-1}) \rightarrow F(x, \psi_n))) \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma_F^+(x, \varphi_1, \psi_{n-1}) \rightarrow F(x, \psi_n)))$ 和 $(\beta \wedge (Poss(\varphi_1, s) \wedge \gamma_F^-(x, \varphi_1, s) \rightarrow \neg F(x, \psi_1))) \wedge (\beta \wedge (Poss(\varphi_1, \psi_1) \wedge \gamma_F^-(x, \varphi_1, \psi_1) \rightarrow \neg F(x, \psi_2))) \wedge \dots \wedge (\beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma_F^-(x, \varphi_1, \psi_{n-1}) \rightarrow \neg F(x, \psi_n))) \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma_F^-(x, \varphi_1, \psi_{n-1}) \rightarrow \neg F(x, \psi_n)))$, 其中 $\psi_1 = do(\varphi_1, s), \psi_2 = do(\varphi_1, do(\varphi_1, s)), \psi_3 = do(\varphi_1, do(\varphi_1, do(\varphi_1, s)))$ 等;

② 组合服务描述 φ 的条件输出的形式公理为 $(\beta \wedge (Poss(\varphi_1, s) \wedge \gamma(x, \varphi_1, s) \rightarrow Knows(y, \psi_1) \vee Kwhether(y, \psi_1) \vee Kref(y, \psi_1)))$

(下转第 16 页)

用特点开发高效的副本创建策略;

- 快速、高可扩展的副本定位算法:在很多数据网格系统中,节点、文件以及每个文件对应的副本都可能很多,因此研究快速、高可扩展的副本定位算法将非常必要;

- 副本一致性策略:针对具体数据网格的应用特点,提供高效的副本一致性管理策略;

- 安全管理:在数据复制技术中安全是数据完整性、机密性以及有效性的保证,包括认证、授权和权力代理三个方面。需要针对网格系统的特点,在权力代理方面做进一步的研究工作。

参考文献

- 1 Foster I, Kesselman C. The Grid: Blueprint for a New Computing Infrastructure. July 1998
- 2 <http://www.npaci.edu/DICE/SRB/>
- 3 <http://www.globus.org/>
- 4 <http://cern.ch/grid-data-management>
- 5 Stockinger H. Database Replication in World-wide Distributed Data Grids: [Ph. D thesis]. U. of Vienna, 2001

- 6 Ranganathan K, Foster I. Design and Evaluation of Dynamic Replication Strategies for a High Performance Data Grid. In: Intl. Conf. on Computing in High Energy and Nuclear Physics, 2001
- 7 Allcock B, et al. Secure, Efficient Data Transport and Replica Management for High-Performance Data-Intensive Computing. In: Mass Storage Conf. 2001
- 8 <http://www-ftp.mcs.anl.gov/dsl/GridFTP-Protocol-RFC-Draft.pdf>
- 9 Mathematics and Computer Science Division, Argonne National Laboratory. Response from Globus Team to EDG WP2's Proposed Design for a Distributed Replica Catalog. <http://grid-data-management.web.cern.ch/grid-data-management/docs/responsetoEDGReplica5.doc>, 7 Sep. 2001, CERN
- 10 Ripeanu M, Foster I. A Decentralized, Adaptive Replica Location Mechanism. In: Proc. High Performance Distributed Computing (HPDC), 2002
- 11 Chervenak I A, et al. Giggles: A Framework for Constructing Scalable Replica Location Services. In: Proc. of SC2002 Conf., Baltimore, MD, 2002
- 12 Stockinger K, et al. Access Cost Estimation for Unified Grid Storage Systems. In: Intl. Workshop on Grid Computing (Grid2003), Phoenix, Arizona, IEEE Computer Society Press, 2003
- 13 Dullmann D, Hoschek W, Jaen-Martinez J, Segal B. Models for Replica Synchronization and Consistency in a Data Grid. IEEE HPDC-10, 2001

(上接第7页)

$(y, \psi_1)) \wedge (\beta \wedge (Poss(\varphi_1, \psi_1) \wedge \gamma(x, \varphi_1, \psi_1) \rightarrow Knows(y, \psi_2) \vee Kwhether(y, \psi_2) \vee Kref(y, \psi_2))) \wedge \dots \wedge (\beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma(x, \varphi_1, \psi_{n-1}) \rightarrow Knows(y, \psi_n) \vee Kwhether(y, \psi_n) \vee Kref(y, \psi_n))) \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma(x, \varphi_1, \psi_{n-1}) \rightarrow Knows(y, \psi_n) \vee Kwhether(y, \psi_n) \vee Kref(y, \psi_n)))$, 其中 $\psi_1 = do(\varphi_1, s)$, $\psi_2 = do(\varphi_1, do(\varphi_1, s))$, $\psi_3 = do(\varphi_1, do(\varphi_1, do(\varphi_1, s)))$ 等;

③ 组合服务描述 φ 的前提的形式公式为 $(\beta \wedge (Poss(\varphi_1, s) \rightarrow \pi_{11} \wedge \pi_{12} \wedge \dots \wedge \pi_{1n})) \wedge (\beta \wedge (Poss(\varphi_1, \psi_1) \rightarrow \pi_{21} \wedge \pi_{22} \wedge \dots \wedge \pi_{2n})) \wedge \dots \wedge (\beta \wedge (Poss(\varphi_1, \psi_{n-1}) \rightarrow (\pi_{n1} \wedge \pi_{n2} \wedge \dots \wedge \pi_{nn})) \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_{n-1}) \rightarrow \pi_{n1} \wedge \pi_{n2} \wedge \dots \wedge \pi_{nn}))$, 其中 $\psi_1 = do(\varphi_1, s)$, $\psi_2 = do(\varphi_1, do(\varphi_1, s))$, $\psi_3 = do(\varphi_1, do(\varphi_1, do(\varphi_1, s)))$ 等;

④ 组合服务描述 φ 的输入的形式公式为 $(\beta \wedge (Poss(\varphi_1, s) \rightarrow Kref(\varphi_{1s}, s) \wedge \dots \wedge Kref(\varphi_{1w}, s))) \wedge (\beta \wedge (Poss(\varphi_1, \psi_1) \rightarrow Kref((\varphi_{21}, \psi_1) \wedge \dots \wedge Kref(\varphi_{2w}, \psi_1))) \wedge \dots \wedge (\beta \wedge (Poss(\varphi_1, \psi_n) \rightarrow Kref(\varphi_{21}, \psi_n) \wedge \dots \wedge Kref(\varphi_{2w}, \psi_n))) \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_n) \rightarrow Kref(\varphi_{21}, \psi_n) \wedge \dots \wedge Kref(\varphi_{2w}, \psi_n)))$, 其中 $\psi_1 = do(\varphi_1, s)$, $\psi_2 = do(\varphi_1, do(\varphi_1, s))$, $\psi_3 = do(\varphi_1, do(\varphi_1, do(\varphi_1, s)))$ 等。

4 讨论

研究语义 Web 服务描述本体 OWL-S 的形式语义的主要目的是从理论上仿真、验证服务的执行过程,本文用带认知流的情景演算理论对 OWL-S 进行了严格的形式化描述,即用带认知流的情景演算理论来刻画 OWL-S 的形式语义。

从理论上仿真、验证服务执行的过程是:将非形式化的用 OWL-S 描述的服务(原子服务或组合服务)转化为情景演算形式化理论,其中包括后继状态公理、动作前提公理,以及情景演算的基本公理、描述初始情景的公理、动作的唯一命名公理等。对于后三种公理,遵循传统情景演算理论的要求;而对于后继状态公理和动作前提公理,由下列方法给出:

由 OWL-S 的形式语义描述中的条件效果和条件输出可以表达情景演算理论中的后继状态公理,它的形式为: $F(x, do(\varphi, s)) \equiv \gamma_F^+(x, \varphi, s) \vee (F(x, s) \wedge \gamma_F^-(x, \varphi, s))$ 。

由 OWL-S 的形式语义描述中的前提和输入可以表达情景演算理论中的动作前提公理,它的形式为: $Poss(\varphi, s) \equiv \pi_1 \wedge$

$\pi_2 \wedge \dots \wedge \pi_n \wedge Kref(\varphi, s) \wedge \dots \wedge Kref(\varphi_w, s)$ 。

将非形式化的服务描述转化为用情景演算理论刻画的形式化描述后,就可以利用情景演算理论的推理机制从理论上求解服务执行问题,即得到一个求解服务的动作序列。

本文的工作只是用情景演算理论来刻画 OWL-S 的形式语义,是在 Srinu Narayanan^[7]的基础上进行的(他只研究了 OWL-S 中的原子服务描述的形式语义,没有研究 OWL-S 中组合服务描述的形式语义)。至于如何利用情景演算从理论上验证服务执行等问题,有待进一步深入研究。

结束语 本文分析了目前语义 Web 服务的研究现状和存在的问题,特别是语义 Web 服务描述本体 OWL-S 的形式语义研究中存在的问题,在 Srinu Narayanan 等人研究的基础上,用情景演算理论进一步研究了 OWL-S 中组合服务描述的形式语义,从而完善了 OWL-S 的形式语义,为语义 Web 服务提供了合理的理论基础。进一步工作主要研究如何利用情景演算从理论上验证服务执行等问题。

参考文献

- 1 Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Scientific American, 2001, 284(5): 34~43
- 2 岳昆,王晓玲,周傲英. Web 服务核心支撑技术:研究综述. 软件学报, 2004, 15(3): 428~442
- 3 蒋运承. 基于主体的智能 Web 中的服务研究: [博士学位论文]. 北京: 中国科学院研究生院, 2004
- 4 McIlraith S, Son T C, Zeng H. Semantic Web Services. IEEE Intelligent Systems. Special Issue on the Semantic Web, 2001, 16(2): 46~53
- 5 The OWL Services Coalition. OWL-S: Semantic Markup for Web Services, OWL-S 1.0 Release, 2003. (<http://www.daml.org/services/owl-s/1.0/>)
- 6 Anupriya A, Frank H, Katia S. Concurrent Execution Semantics of DAML-S with Subtypes. In: I Horrocks, J Hendler, et al. The Semantic Web - ISWC 2002, First Intl. Semantic Web Conf Sardinia, Italy, Springer Verlag, 2002. 318~332
- 7 Narayanan S, McIlraith S. Analysis and simulation of Web services. Computer Networks: The International Journal of Computer and Telecommunications Networking, 2003, 42(5): 675~693
- 8 Reiter R. Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. MIT Press, 2001
- 9 Giacomo G D, Lespérance Y, Levesque H J. ConGolog: A Concurrent Programming Language Based on the Situation Calculus. Artificial Intelligence, 2000, 121: 109~169