Web交互模型的形式化验证研究

李 敏[1] 罗惠琼[2] 唐春玲[1] 王 强[2]
(重庆广播电视大学电子信息工程学院 重庆 400052)[1]
(电子科技大学计算机科学与工程学院 成都 611731)[2]

摘 要 Web交互模型的形式化验证是对Web事件属性进行校验的十分可信的方法。通过一系列的系统模型建立、系统行为分析以及对于模型中关心属性的相关验证,能够让交互模型在设计阶段就能使形式化模型暴露出其所存在的缺陷,而不至于让缺陷保留到编码阶段或者更后面才能被真正地暴露出来,这样使系统模型的生存能力更加强大,同时避免了因后期缺陷暴露而出现的大代价修复。通过对Web系统的交互应用服务的过程模型化的体系进行研究,通过模型本身具有的属性进行相关正确性的校验,主要通过使用数学推理实现系统逻辑上的服务交互进程,从而进行过程的推演,并对系统服务的正确性进行过程的形式化验证,从而使系统服务模块的属性正确性可以通过逻辑上的演进来发现服务问题的存在,而不再是系统通过编码实现后才发现。对Web交互模型的形式化验证是基于IMWSC模型语义形成的IMWSC模型的验证机制。

关键词 Web交互模型,形式化验证,数理推演,模型语义
中图法分类号 TP317.4 **文献标识码** A

Research on Formal Verification of Web Interaction Model

LI Min[1] LUO Hui-qiong[2] TANG Chun-ling[1] WANG Qiang[2]
(Electronics and Information Engineering, Chongqing Radio and TV University, Chongqing 400052, China)[1]
(School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)[2]

Abstract Formal verification of Web interaction model is a credible way on evaluating the attributes of Web events. Through a series of system modeling, behavior analysis, and related validation of center properties, defects will expose during the design phase instead of coding phase or later in the formal model. Thereby, the viability of system model is more powerful. At the mean time, it cost less than the spending of late defect exposure. We investigated the process modeling of interactive application service on Web system, checking the correctness of model's relative properties. Besides, process modeling achieves service interaction processes deduction on system logic unit through mathematical reasoning. And formal verification of process aiming at the correctness of system services was also performed. The advantage of this method reflects mainly on the early discovery of defects in system service model. The formal verification of Web interaction model is based on IMWSC model verification mechanism.

Keywords Web interaction model, Formal Verification, Mathematical deduction, Model semantics

1 前言

随着降低软件开发成本和提高开发速度与质量的需求日益增长,及基于分布式技术和构件模块的复用技术的快速发展,一种全新的面向服务的构架体系SOA逐步成为软件开发体系中的一个重要成员。在形成的新一代网络计算模型中,以Web的模块[1]组合及其提供的服务作为核心模型的通用网络计算方式成为主流。

以Web服务及其组合为核心的服务计算模式成为新一代网络计算的主流。现有的各类Web服务分布在互联网[2]的任何地方,实现着各种应用服务的发布与业务支持。如何有效地构建出满足企业多元化需求和业务多变性以及各类业务间无缝衔接的应用服务已经成为现代Web服务研究的热点问题。在整个Web服务[3]中,通过应用流程管理、分布式的计算规则、采用软件工程的系统方法等各种新技术的集成发展来实现面向交互模型的服务体系结构,充分实现应用集成和现有资源的更深层次的共享。本课题的内容就是基于这种拥有深刻技术背景和广泛应用前景的热点研究问题深入展开。Web应用服务的交互过程的形式化建模方式中,许多工程应用的研究者希望通过数学中的进程代数来统计分析Web服务交互过程的行为模式[4],并通过这些行为建立交互行为的模型。在服务交互系统的设计和形式化的验证阶段,进程代数能给出较好的协助。其在设计阶段能够对系统的交互行为给出较好的全面描述,在通过分析建立起模型后可以

到稿日期:2013-10-09 返修日期:2013-12-04 本文受重庆市教委(kj131607)资助。

李 敏(1973—),女,硕士,讲师,主要研究方向为软件理论、服务计算等,E-mail:lmm186023@126.com;罗惠琼(1949—),女,硕士,教授,主要研究方向为计算机网络与通信、智能仪器、嵌入式和物联网;唐春玲(1981—),女,硕士,讲师,主要研究方向为软件工程;王 强(1976—),男,博士生,主要研究方向为信息安全。

通过其具有的迭代推演，获得系统以后可能出现的行为，从而对行为提前给出处理意见。再有通过其对系统交互行为的推演可以尽可能知道之后系统所处的状态，给出的系统模型是否正确，并可以预见到系统存在的资源不足、服务推进时序方面的问题以及系统具有的缺陷，尽量使系统问题暴露在系统的设计和模型阶段。但是在采用进程代数作为行进模型推演的方式时，往往使用简单的规则转换和关系对应来描述模型的属性和对有效性进行验证，这种缺少严格函数限制的转化很可能会导致一些重要信息的丢失。如 CCML 语言[5] 控制结构与通信系统演算操作符的对应关系就可能出现这样的信息丢失，使最后验证的信息不准确。为了避免造成这种信息丢失，课题研究了一种定义转换机制更加严格的 IMWSC 模型的语义。它是在通信系统的推进演算的基础上形成的具有语义定义、语义转换函数和语义值域的严格模型定义。

2 语义定义域(IMWSC)

IMWSC[6] 是一种以使用动作(数据的 IO 操作、数据的读写操作、数据运算的执行操作等)来体现服务操作过程的调用。通过将服务过程中操作动作以线性化的方式形成操作序列，再依次将序列中操作涉及到的推送消息和接收消息等映射为对应的线性动作序列中的一个。针对主要动作对应的动作序列可以设计出相应的表达式来呈现，如表 1 所列。

表 1 Web 服务的 IMWSC 表达

操作类型	例子	IMWSC 表述
One Way	〈operation name ="ex"〉 〈input message="ms"〉 〈/operation〉	Input Action
Request-Response	〈operation name ="ex"〉 〈input message ="ms"〉 〈output message ="mr"〉 〈/operation〉	Input Action 〈Output Action〉
Solicit-Response	〈operation name ="ex"〉 〈output message ="mr"〉 〈input message ="ms"〉 〈/operation〉	Output Action 〈Input Action〉
Notification	〈operation name ="ex"〉 〈output message="ms"〉 〈/operation〉	Output Action

在 IMWSC 的序列动作执行上通过建立起相关的进程来处理过程中的动作，一系列的动作先后组合起来执行形成了进程集。对某一个进程集进行集中处理来达到系统的集成服务的目的，最后通过动作序列的优化推进来提升系统服务的质量和效率。

3 语义值域(通信模型 CCS)

一个通信系统演算 CCS[7] 中主要由 P、Q、M、R 等变量来表示系统演算的构造基本语法。其变量所涉及的 P、Q 进程序列有如下的对应关系：

$$P::=nil\,|\,h.Q\,|\,P+Q\,|\,P\parallel Q\,|\,P\backslash L \tag{1}$$

$$h::=\alpha?\,x\,|\,\alpha!x\,|\,\tau \tag{2}$$

式(1)中 nil 表示对应的进程将结束，做结束表示用，在执行这个动作后进程将停止执行一切动作；$h.Q$ 表示对于进程 Q 其将优先执行动作队列的 h(高优先级)动作，其后对 P 进程做执行操作；$P+Q$ 为复合进程，表示 P、Q 两个进程为选择执行的进程，在执行过程中如果 P 进程作为执行进程被选择，那么 Q 进程为不再被选择的进程；相应地符合进程 $P\parallel Q$ 则表示可互相并行的进程，即满足 P、Q 进程同时执行的目的。式(2)中 α 用于描述对应的通道名，x 用于描述存在的一个消息；$\alpha?\,x$ 表示消息 x 需要通过 α 通道进行接收，$\alpha!x$ 表示消息 x 需要通过 α 通道进行发送；而 τ 则为进程隐藏的内部动作。

CCS 的操作语义[8] 存在着如式(3)的迁移定义方式，这可以在一个已知的初态 I、状态集 $Proc$ 和其对应执行的动作集 h，通过进程的执行推演出系统的后续运行状态。

$$(Proc,Action,\{\xrightarrow{h}|h\in Action\},I) \tag{3}$$

式中，I 为进程的初态集，$\xrightarrow{h}$ 为进程中对应关系的迁移方式，$Action$ 为一个动作集，$h\in Action$ 则表示对应的关系迁移是由动作集中的一个在当前具有高优先执行的动作导致的结果，$Proc$ 为进程的状态集，描述了进程在执行动作后的状态特性。

4 语义转换函数方式

IMWSC 模型是将模型的原始定义通过进程代数的推演表达作为自己的语义形势。IMWSC 模型的语义[9] 提供了一种相当严格的思考模型来对服务的交互行为进行定义，因此对系统的建模操作和对现有系统的深入认识以及系统设计的详细理解具有辅助作用。表 2 中将给出 IMWSC 的转换函数及其与之相关的值域和定义域。

表 2 IMWSC 转换函数对应关系

转换函数	语义定义域	语义值域
f_m	IMWSC	P
f_c	Proc	P
f_a	Proc	P
f_r	A	P
f_e	Activity	Act

转换函数的实现如下所示：

$$f_m(Proc_r)=f_c(Proc_1)\,|\,f_c(Proc_2)\,|\cdots|\,f_c(Proc_n)$$

式中，$Proc_r,Proc_i\in Service\{1\leqslant i\leqslant n\}$；

$$f_c(Proc_i)=f_a(Proc_i)\ \text{iff}\ f_{pT}(Proc_i)=a$$

$$f_c(Proc_i)=f_r(Proc_i)\ \text{iff}\ f_{pT}(Proc_i)=c$$

$$f_a(Proc_i)=f_e(a_1).\,f_e(a_2)\cdots f_e(a_n)$$

其中

$$a_i\in Activity\wedge f_e(a_i)=Proc_i,\text{且}\ a_1<a_2<\cdots<a_n$$

$$f_e(a_i)=!a_i\ \text{iff}\ f_{aT}(a_i)=ii\vee f_{aT}(a_i)=ei$$

$$f_e(a_i)=?\,a_i\ \text{iff}\ f_{aT}(a_i)=ii\vee f_{aT}(a_i)=eo$$

$$f_r(Proc_i)=f_c(Proc_1)\,|\,f_c(Proc_2)\ \text{iff}\ f_{pU}(Proc_1)=Proc_1\wedge f_{pU}(Proc_2)=Proc_1\wedge Proc_2=Proc_1\parallel Proc_2$$

$$f_r(Proc_i)=f_c(Proc_1).\,f_c(Proc_2)\ \text{iff}\ f_{pU}(Proc_1)=Proc_1\wedge f_{pU}(Proc_2)=Proc_1\wedge Proc_2=Proc_1\prec Proc_2$$

通过上述 8 个转换公式完整描述了 IMWSC 函数转换方式及转换涉及到的函数操作和值域范围。

5 模态 μ 演算验证 IMWSC 模型属性

计算机软件系统中的一种经典方法是利用数理逻辑进行逻辑推理、系统建模和对建立起的模型进行可行性验证，数理逻辑的主要作用是验证给出的规范要求与建立的模型在系统属性上是否达到了一致，从而提早获知系统在开发实现前是否处于一种安全可用的状态。其验证方式正常情况下为：

1. 首先需要对系统建立起一个具有过程化的模型 M，在建立模型时使用的是具有描述性语言的模型检测器来对系统

建模；

2. 建立起时序逻辑关系，并将此关系转化为数理公式获得系统时序推进序列，再通过使用模态 μ 演算系统所提供的规范性语言来对系统的属性进行编码实现.使其最终确定时序逻辑公式 Q 为模型的一般时序；

3. 再以模型 M 和系统运作时序 Q 作为输入，运行模态 μ 演算系统，从模型检测器的结果中获取系统运行的正确性。

在本课题中主要所用的规范性语言的模型检测器为模态 μ 演算[10]，其主要提供的是一种时序逻辑的检测，使用时序逻辑来描述系统的并发特性，对于不同的时序可以使系统产生不同的结果，也给出系统的不同状态，因此时序逻辑是模型检测的基础。这里所使用的时序逻辑是以一个固定的初始状态点作为时序的不动点参与系统演进，而系统执行所用的动作称为系统的时序逻辑推进变化点，通过变化点来对系统进行迭代推进，使系统达到最后的稳定点，最后完成系统模型的推演，从而获得系统的正确性。在推演时随着推演时变化点所执行动作的优先级的改变可以获得最终系统稳定点的改变，这样为系统验证特性点、系统安全性做了全面的描述，同时这样的时序更改推演也可以用作系统的性能调优，是完成时序停留在系统最佳位置上的时序排布。其演算验证规则可用下列公式描述，设其初始状态为 p_0，并且系统能够在有限步的动作下达到一个稳定状态，则系统的整个迁移范围可以看作以下公式过程：

$$p_0 \xrightarrow{h_1} p_1 \xrightarrow{h_2} p_2 \cdots p_n \tag{4}$$

6 服务交互过程正确性判定

以往的 Web 服务研究[11]一般集中于对交互实体间的信息正确性进行判断和对交互服务兼容性进行验证。但是作为一个完整的 Web 交互服务系统，它不止关心两个实体间交互的正确性，因为只是交互的正确并不能充分地保证系统运行的正确以及系统的运行稳定。而在现在的交互行为中要求进行多实体集的交互行为处理，这样的交互行为就不再仅仅以正确性为考虑的重点，它需要让系统有调理地处理多个服务间的交互，更重要的是要保证多个服务的协调性，即每个服务具有时序性。因此针对多服务交互的弊端，本课题设计了全局交互行为协调和局部交互行为协调的方法，以对 Web 服务的全局交互正确性做一个全面的判定。

6.1 局部交互行为协调

一个 IMWSC 模型满足其划分的关联集合的供需达到平衡需要：1)模型的所有输入输出操作都是成对地出现，即一个 I、O 动作后就必须出现一个与之对应的 O、I 存在，实现输入输出的平衡；2)模型中的所有动作都是可以进行多次操作的，其既可以执行也可以取消执行，对动作序列中的任意一个动作都可以添加或者删除，可以调整任意动作的优先级序列，并且可以将动作序列添加为满集合，也可以将其置为空集合，实现进程动作的平衡。

实现局部交互的条件为：

$$\forall d_i \in R_D, a \in Activity(a \in d_i \wedge f_{dT}(a) = input \longrightarrow (\exists a' \in Activity \wedge aR_a a')) \tag{5}$$

$$\forall d_i \in R_D, a \in Activity(a \in d_i \wedge f_{dT}(a) = output \longrightarrow (\exists a' \in Activity \wedge a' R_a a)) \tag{6}$$

6.2 全局交互行为协调

在全局交互行为协调中，全局次序定义了系统中属于不同服务的执行动作的时序关系，并作为整个进程推进的依据。首先在全局次序上定义了 $\leqslant_g$，其是 $Activity$ 集合上存在的二元关系，$\exists \forall a, b \in Activity$，则有如下关系成立。

$(a,b) \in \leqslant_g$，如果 $a \in aR_a b, b \in aR_a b$

$(a,b) \in \leqslant_g$，如果 $a <_l b$

$(a,b) \in \leqslant_g$，如果 $\exists b' \in Activity \wedge a \leqslant_g b' \wedge b' <_l b$

$(a,b) \in \leqslant_g$，如果 $\exists b' \in Activity \wedge a \leqslant_g b' \wedge bR_a b'$

$a_1, a_2 \in Activity, a_1 \leqslant_g a_2$，当且仅当 a_1 在 a_2 之前执行

结束语 本文研究的主要依据是通过使用 Web 交互服务的组合特性及其推进可知性的行为来验证 Web 服务实现与之前的设计是否是一致的，同时能够尽可能早地发现现有系统存在的缺陷和不足，通过系统的全面推演寻找到符合客户要求的最佳匹配系统模型，从而更好地提高客户对系统的满意度，提高系统的产出效率。通过对模态系统的行为输入和模型转换，最终获得各种状态下的系统推演序列，了解系统内各项交互服务间的组合存在的内部和外部行为的定义、交互行为间的强弱互拟性以及交互行为间存在的相关耦合性。再通过对服务组合属性的正确性验证确保了系统在多种复杂开放的网络环境下具有系统服务正常运行的能力，并保证了系统运行时数据的可靠性与完整性，为客户提供了正确、安全、可用的多交互服务，使其业务逻辑能有效完成，达到最初预计的要求。

参 考 文 献

[1] Misra J, Cook W. Computing Orchestration: A Basis for Wide-Area Computing[J]. Journal of Software & Systems Modeling, 2011,6(1):83-110

[2] Foster H, Uchitel S, Magee J, et al. LTSA-WS: A Tool for Model-basedVerificationofWebServiceCompositionsandChoreo-graphy [C]//Proc. of ICSE. 2010:771-774

[3] George Z,B A. Service Mining on the Web[J]. the VLDB Journal,2009,2(1):65-78

[4] Tsesmetzis D T,Russaki I,Papaioannou I V,et al. A QoS Ontology Language for Web Services[C]//AINA. 2009(1):101-106

[5] Bao Li,Zhang Wei-shi,Xie Xiong. A Formal Model for Abstracting the Interaction of Web Services[J]. Journal of Computers, 2010,5(1):91-98

[6] Bao Li,Zhang Wei-shi,Xie Xiong. Abstracting the Interaction of Web Services Using IMWSC[J]. Journal of Information and Computational Science,2009,6(2):699-708

[7] Zhang Xiu-guo,Zhang Wei-shi. A Cooperative Service Composition Language and Its Formal Semantics[C]//Proc. of the 7th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2008). TaiPei: IEEE Computer Society Press,2008

[8] Bemmel J V, Wegdam M, Lagerberg K. 3PAC: Enforcing Access Policies for WebServices[C]//IEEE International Conference on Web Services(ICWS'05). 2005:589-596

[9] ter Beek M H, Bucchiarone A, Gnesi S. Formal Methods for Service Composition[R]. Technical Report. Software/Program Verification,Formal Methods,ACM

[10] Endrei M, Ang J, Arsanjani A, et al. Patterns: service-Oriented Architecture and Web Services[OL]. http://www.redbooks.ibm.eom/redbooks/Pdfs/sg246303.pdf

[11] 杨艺,周元. 基于用户查询意图识别的 Web 搜索优化模型[J]. 计算机科学,2012,39(1):264-267