

时间感知 Web 服务交互行为建模 与失配检测方法研究

张广泉^{1,2}, 戎 玫³, 王 ■¹

(1. 苏州大学计算机科学与技术学院, 江苏苏州 215006; 2. 中国科学院计算机科学国家重点实验室, 北京 100190;
3. 暨南大学深圳旅游学院, 广东深圳 518053)

摘 要: 针对现有 Web 服务组合过程中存在时间感知力弱、服务利用率低、组合可靠性差等问题, 通过将定量时间属性引入 Web 服务交互适配框架中, 研究时间感知 Web 服务交互行为的形式化建模与交互行为失配的自动检测问题. 提出了用于表达单个时间感知 Web 服务交互行为的时间服务协议 (TSP) 模型和用于表达多个时间感知 Web 服务并发组合的时间服务协议网络 (TSPN) 模型; 将时间感知 Web 服务交互失配检测问题转化为 TSP 模型之间的兼容性判定问题, 定义了从时间感知 Web 服务描述至 TSP 模型和从 TSP 模型至 UPPAAL 规约的映射规则, 给出了 TSP 模型兼容性性质的 CTL 公式表示, 基于模型检测工具 UPPAAL 实现时间感知 Web 服务交互失配的自动化检测, 其检测结果作为下一步是否需要构造适配器的判断依据; 最后通过实例分析进一步阐述了上述方法的可行性和有效性.

关键词: 时间感知 Web 服务; 时间服务协议; 时间服务交互; 模型检测; 失配检测

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2011) 11-2568-08

Research on Interaction Modeling and Mismatch Checking of Time-Aware Web Services

ZHANG Guang-quan^{1,2}, RONG Mei³, WANG Sheng¹

(1. School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006, China;
2. State Key Laboratory of Computer Science, Chinese Academy of Science, Beijing 100190, China;
3. Shenzhen Tourism College, Jinan University, Shenzhen, Guangdong 518053, China)

Abstract: Targeting the problem of weak time awareness, low service utilization and poor composition reliability in the process of web services composition, we propose to bring quantitative timed properties into web services interaction adaptation framework and make efforts on researching on time-aware web services interaction modeling, automated interaction mismatch detection and automated adaptor protocol generation. First, we propose timed service protocol (TSP) model to represent interaction behavior of single time-aware web service and timed service protocol network (TSPN) model to represent concurrent composition of multiple time-aware web services; Second, we transform the interaction mismatch detection problem into the determination problem of compatibility of TSP models, define the mapping rules from time-aware web service descriptions to TSP models and from TSP models to UPPAAL specifications, give the CTL representations of compatibility properties and implement automatically interaction mismatch detection of time-aware web services based on model checker UPPAAL, whose result determines whether it is necessary to build adaptor for the next step. Finally, experiments demonstrate feasibility and validity of above idea.

Key words: time-aware web services; timed service protocol; timed service interaction; model checking; mismatch checking

1 引言

近年来, 以 Web 服务为主流实现技术的各类软件服务在 Internet 中的数量呈快速增长的态势, 通过服务组合的方式, 灵活、快速构建的网络应用已经渗透到政府、金融、教育等关键行业, 并发挥着日益重要的作用.

但是, 由于网络环境中服务具有异构、动态等特性, 导致许多服务之间因交互失配 (mismatch) 而无法直接进行组合. 面对网络上数量日趋增长的可用 Web 服务, 如何有效提高服务利用率并保证服务组合系统的可靠性, 服务适配 (service adaptation) 在交互失配检测的基础上, 通过构造适配器来实现服务之间的正确交互, 为解决该

问题提供了一条重要的有效途径,已成为目前服务计算领域中的一个研究热点^[1].

随着互联网和网络技术的快速发展,各种新型的服务不断出现,将这些服务进行组合时,不仅需要保证各成员服务之间语法和语义的兼容性,更要保证它们在行为上的兼容性.因此,行为兼容性是 Web 服务组合中各服务之间兼容性的一个重要方面.目前,Web 服务行为适配的研究引起了国际学术界的关注^[2-4].

已有的关于 Web 服务行为适配的研究工作大多局限在定性需求层面,而忽视了一些定量属性如时间等因素也会导致 Web 服务交互发生失配.基于此,本文将时间属性引入 Web 服务交互适配框架中,研究时间感知 Web 服务交互行为的形式化建模与行为失配的自动检测问题.所谓时间感知 Web 服务,是指具有基本时间约束(如 BPEL 规范中的 onAlarm 元素和 wait 元素)、长时间活动特征(如标注 BPEL 中的 duration 元素)和复杂时间约束(如消息 b 必须在接收到消息 a 两小时以后才能被发送)三方面需求的一类特殊 Web 服务^[5].

本文的研究工作主要分为两个部分(见图 1):

(1)采用抽象 BPEL 规范描述时间感知 Web 服务的交互行为,并转换为时间自动机模型;对复杂时间约束需求采用实时时序逻辑 TPTL^[6]公式表示,然后也转换为时间自动机模型;对上述的行为自动机模型和其对应的属性自动机模型求积,得到某个特定时间感知 Web 服务的时间服务协议模型(TSP).

(2)采用模型检测工具 UPPAAL 实现交互失配的自动检测,即对表达时间感知 Web 服务交互行为的 TSP 模型的兼容性质进行验证,该验证结果可作为下一步是否需要构造适配器的判断依据.

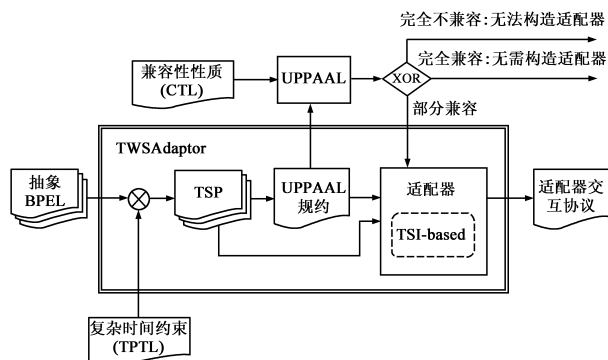


图1 本文研究框架

2 时间感知 Web 服务交互的形式化模型

构建能精确描述与分析 Web 服务交互行为的形式化模型,是实现交互失配自动检测的前提和基础.已有的典型工作归纳起来,可分为基于进程代数^[7,8]、基于 Petri 网^[9,10]和基于状态迁移模型^[11]等三类建模方法.

从描述时间感知 Web 服务交互角度来看,这些方法在不同程度上均存在下列问题:(1)过于关注 Web 服务的内部组合行为,不利于揭示外部交互的本质;(2)缺少对时间属性的显式表达;(3)忽略了异步通信才是 Web 服务交互的本质特征.

图 2 通过一个协议不兼容(考虑时间因素影响)的交互场景,显示了时间感知 Web 服务交互的概念模型. tWS_3 开始时发送消息 a 至 tWS_4 的消息队列 $Que(tWS_4)$,并将私有时钟变量 x 置 0,同时, tWS_4 发送消息 c 至消息队列 $Que(tWS_3)$,此时, tWS_3 阻塞于状态 S_2 直到消息 b 出现在 $Que(tWS_3)$ 中,而 tWS_4 可以消费 $Que(tWS_4)$ 中的消息 a ,因为 $Que(tWS_4)$ 中存在消息 a 且没有时间冲突.从接收消息 a 的时刻开始计时(私有时钟变量 y 置 0),当满足条件 $20 \leq y \leq 40$ 时,消息 b 才能被 tWS_4 发送到 tWS_3 的消息队列 $Que(tWS_3)$ 中.这样, tWS_3 将在至少 20 个时间单位以后接收消息 b ,但是 tWS_3 却要求在满足条件 $x \leq 10$ 的情况下,从 $Que(tWS_3)$ 中消费消息 c ,此刻,虽然消息 c 确实存于 $Que(tWS_3)$ 中,但由于时间冲突($y < x, y \geq 20, x \leq 10$),该迁移将永远不会发生.这里规定:任意的消息发送动作可将消息送入指定的消息队列,而任意的消息接收动作则只能在与自身对应的消息队列中消费消息;消息发送动作“!”对应于消息入队操作“+”,消息接收动作“?”对应于消息出队操作“-”.

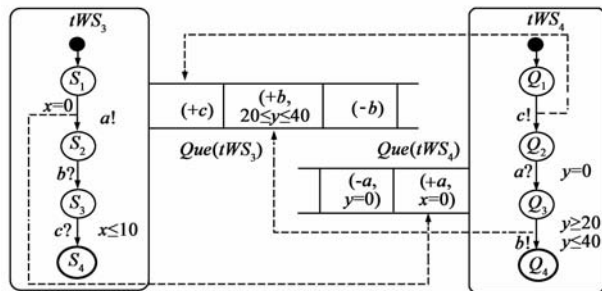


图2 时间感知Web服务交互的概念模型

下面分别给出单个时间感知 Web 服务交互行为的 TSP 模型和用于表达多个时间感知 Web 服务并发组合交互行为的 TSPN 模型.

2.1 单个时间感知 Web 服务的交互行为模型

通过对有限状态机模型扩展和改进,提出一种用于表达单个时间感知 Web 服务交互行为的时间服务协议(TSP)模型.

定义 1 时间服务协议(Timed Service Protocol, TSP)是一个七元组 $P = (S, s_0, F, M, X, I, R)$,其中:

① S 是状态的有限集合;

② $F \subseteq S$ 是终止状态的集合,规定:当 $F = \emptyset$ 时, P 为空协议;

③ $s_0 \in S$ 是初始状态, 且规定 $s_0 \in F$;

④ $M = M_e \cup \{\epsilon\}$ 是消息集合 M_e 和空消息元素 ϵ 组成的消息集合. 对于任意的消息 $m \in M_e$, 定义函数 $Polarity(m)$ 用于识别消息 m 的属性, 如果 m 是输入消息, 则 $Polarity(m) = ?$; 如果 m 是输出消息, 则 $Polarity(m) = !$;

⑤ X 是有限时钟变量的集合;

⑥ $I: S \rightarrow \Phi(X)$, 为 S 中的每一个状态指定 $\Phi(X)$ 中的一个时钟约束, 当状态的时钟约束不满足时, 必须能够执行迁移离开该状态;

⑦ $R \subseteq S \times M \times \Phi(X) \times 2^X \times \{true, false\} \times S$, 对于任意的一条迁移 $r \in R$, 假设有 $r = (s, m, \phi, Y, \zeta, s')$, 其中 s 为源状态, s' 为目标状态; m 是触发迁移的消息动作; ϕ 是迁移使能条件; $Y \subseteq X$ 是迁移发生时需要重置的时钟变量集合; $\zeta = true$ 指明该迁移是紧急迁移, 反之, $\zeta = false$ 指明该迁移是非紧急迁移, 当且仅当满足下列条件时, 迁移 r 才能被触发:

- $Polarity(m) = !$
- $Polarity(m) = ?$

与有限状态机模型相比, TSP 模型具有下列特点:

(1) 初始状态也是一个终止状态.

(2) 时间特征明显. 为了支持时间感知 Web 服务交互协议中时间特征的显式表达, TSP 模型采纳了标准时间自动机中的时钟解释^[12], 即任意时钟 $x \in X$ 都是非负实数域 $R_{\geq 0}$ 上的一个变量, X 中所有时钟变量的值随着时间的流逝同步增长. 对于时钟变量集合 X , 时钟约束 $\phi \in \Phi(X)$ 具有形式: $true \mid x \Theta c \mid \phi_1 \wedge \phi_2$, 其中操作符 $\Theta \in \{<, \leq, =, \neq, >, \geq\}$, $x \in X$, 时钟约束公式 $\phi_1, \phi_2 \in \Phi(X)$, 常量 $c \in R_{\geq 0}$.

此外, 迁移可以被声明为紧急或非紧急. 紧急迁移的优先级高于非紧急迁移, 要求一旦到达即被触发. 非紧急迁移用于建模那些消耗时间的任务, 默认情形下的迁移都是紧急的.

(3) 能表达异步通信模式. 可以看出, TSP 模型通过引入消息队列 Q 以及相应的消息存取规则, 实现了消息的异步交换功能, 从而避免了传统有限状态机模型只能通过共享变量实现同步消息交换的不足.

2.2 多个时间感知 Web 服务的并发组合模型

多个时间感知 Web 服务之间的交互可以定义为其 TSP 模型的并发组合, 本文使用时间服务协议网络 (TSPN) 来表示.

定义 2 时间服务协议网络 (Timed Service Protocol Network, TSPN): 对于一组参与交互的 TSP P_1, P_2, \dots, P_n , $P_i = (S_i, s_{i0}, F_i, M_i, X_i, I_i, Q_i, R_i)$ 且 $(1 \leq i \leq n)$, 其 TSPN 定义为一个八元组 $PN = (S, s_0, F, M, X, I, Q,$

$R)$, 其中:

① S 是状态的集合, S 中的每个状态 s 是一个向量, $s = (s_1, s_2, \dots, s_n)$, $s_i \in S_i$;

② s_0 是初始状态, $s_0 = (s_{10}, s_{20}, \dots, s_{n0})$, $s_{i0} \in S_i$ 且 $s_{i0} \in F_i$;

③ $F \subseteq S$ 是终止状态的集合;

④ $M = M_1 \cup M_2 \cup \dots \cup M_n$ 是消息的集合;

⑤ $X = X_1 \cup X_2 \cup \dots \cup X_n$ 是时钟变量的集合;

⑥ $I: S \rightarrow \Phi(X)$, 为 S 中的每一个状态向量指定 $\Phi(X)$ 中的一个时钟约束, 称为该状态向量的位置不变式, $I(s) = I_1(s_1) \wedge I_2(s_2) \wedge \dots \wedge I_n(s_n)$;

⑦ $Q = Q_1 \cup Q_2 \cup \dots \cup Q_n$ 是消息队列的集合;

⑧ $R \subseteq S \times M \times \Phi(X) \times 2^X \times S$, 对于 $\forall r \in R$, 假设 $r = (s, m, \phi, Y, s')$, 其中 s 为源状态, s' 为目标状态; m 是触发迁移的消息动作; ϕ 是迁移使能条件, 它是时钟变量集合 X 上的一个时钟约束; $Y \subseteq X$ 是迁移发生时重置的时钟变量集合.

3 时间感知 Web 服务交互的失配检测

Web 服务交互的失配检测即在系统空间中考察是否存在可达的死锁状态. 图 3 显示了一个因隐含时间冲突导致交互失配的典型场景 (该场景为同步通信模式, 故省略了消息队列). 通过分析已有的相关工作^[13,14], 时间感知 Web 服务交互的失配检测问题可转化为交互协议的兼容性判定问题.

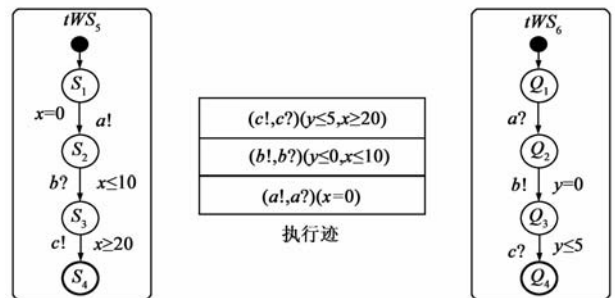


图3 隐含时间冲突导致交互失配

交互协议的兼容性判定可分为完全兼容、部分兼容和不兼容三类, 而只有在部分兼容的情形下, 才可能且有必要为交互失配的时间感知 Web 服务构造适配器. 为此, 首先给出 Web 服务交互协议兼容性定义.

3.1 Web 服务交互协议的兼容性定义

定义 3 TSP 模型中的死锁状态. 对于任意的 TSP $P = (S, s_0, F, M, X, I, Q, R)$, 状态 $s \in S$ 是 P 中的死锁状态, 表示为: $Dead(s)$, 当且仅当: $s \in S \wedge s \notin F \wedge (\exists m \in M \vee \exists \phi \in \Phi(X), s' \in S. (s, m, \phi, Y, \zeta, s') \in R)$.

上述定义表明死锁状态区别于正确终止状态, 即死锁状态 s 在状态集 S 中, 但却不在正确终止状态集 F

中,且没有出迁移。

定义 4 完全兼容性. 对于一组参与交互的 TSP P_1, P_2, \dots, P_n , 且 $PV = (S, \bar{s}_0, F, M, X, I, Q, R)$, 若满足完全兼容性, 记作 $FullComp(P_1, P_2, \dots, P_n)$, 当且仅当: $\forall s \in S \wedge s \notin F, Dead(s) = false$; 且 $Q_1 = empty \wedge Q_2 = empty \wedge \dots \wedge Q_n = empty$.

上述定义表明一组参与交互的时间感知 Web 服务交互协议是完全兼容的必须满足两个条件: 一是系统空间(TSPN 模型)中没有死锁状态; 二是所有的发送消息必须被全部接收。

定义 5 部分兼容性. 对于一组参与交互的 TSP P_1, P_2, \dots, P_n , 且 $PV = (S, \bar{s}_0, F, M, X, I, Q, R)$, 若满足部分兼容性, 记作 $PartComp(P_1, P_2, \dots, P_n)$, 当且仅当: $\neg FullComp(P_1, P_2, \dots, P_n)$; 且 $\exists s \in F, \bar{s}_0 \xrightarrow{\omega} s, \omega$ 为任意的有效执行路径, 且 $\omega \neq \emptyset$.

上述定义表明一组参与交互的时间感知 Web 服务交互协议是部分兼容的也必须满足两个条件: 一是不满足完全兼容性; 二是在系统空间(TSPN 模型)中至少存在一条从初始状态 \bar{s}_0 到正确终止状态 $s \in F$ 的有效执行路径。

定义 6 完全不兼容性. 对于一组参与交互的 TSP P_1, P_2, \dots, P_n , 且 $PV = (S, \bar{s}_0, F, M, X, I, Q, R)$, 若满足完全不兼容性, 记作 $FullIncomp(P_1, P_2, \dots, P_n)$, 当且仅当: $\exists s \in F, \bar{s}_0 \xrightarrow{\omega} s$, 且 $\omega \neq \emptyset$; 或 $\forall s \in F, \bar{s}_0 \xrightarrow{\omega} s \wedge (Q_1 > 0 \vee Q_2 > 0 \vee \dots \vee Q_n > 0)$, 且 $\omega \neq \emptyset$.

上述定义表明一组参与交互的时间感知 Web 服务交互协议是完全不兼容的只需满足: 在 TSPN 模型中, 不存在一条从初始状态到正确终止状态的有效执行路径; 或者, 到达了正确终止状态但所有的发送消息未被全部接收。

3.2 基于模型检测的时间感知 Web 服务交互失配的自动检测

模型检测是一种使用状态空间搜索来检验系统是否满足某些特定性质的算法方法^[15], 其基本思想是将系统行为表示为一个状态迁移模型 M , 系统性质由时序逻辑公式 ϕ 描述, 这样“系统是否满足所期望的性质”就可以转化为“状态迁移模型 M 是否满足时序逻辑公式 ϕ ”。目前较为著名的模型检测工具包括 SMV、SPIN、FDR、UPPAAL 等。考虑到所验证的系统模型包含时间属性, 本文采用瑞典 Uppsala 大学与丹麦 Aalborg 大学合作开发的适用于实时系统建模、分析和验证的模型检测工具 UPPAAL。

图 4 给出了 UPPAAL 实现时间感知 Web 服务交互失配自动检测的框架。先将时间感知 Web 服务描述转

换为 TSP 模型, 再将 TSP 模型转换为 UPPAAL 规约, 采用时序逻辑公式 CTL 表示 TSP 模型的兼容性性质, 将 UPPAAL 规约和 CTL 公式分别作为 UPPAAL 的模型和性质输入, 输出结果将表明一组给定的时间感知 Web 服务交互协议的兼容性性质所属的类别, 从而实现服务交互失配的自动检测。

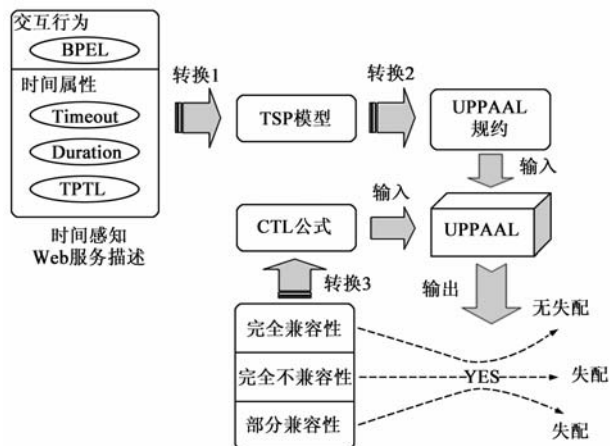


图4 时间感知Web服务交互失配检测框架

为了能够使用 UPPAAL 成功实现交互失配检测, 需要以下三个转换环节:

转换 1 时间感知 Web 服务描述转换至 TSP 模型。

时间感知 Web 服务描述包括交互行为描述和时间属性描述。限于篇幅, 表 1~5 分别给出了部分典型的基本 BPEL 语句、结构化 BPEL 语句、并行 BPEL 语句、时间 BPEL 语句及 TPTL 公式至 TSP 模型的映射规则。

转换 2 TSP 模型转换至 UPPAAL 规约。

TSP 模型的一些元素可以直接映射至 UPPAAL 规约, 例如状态上的不变式、迁移上的卫式等, 但要采用 UPPAAL 实现时间感知 Web 服务交互协议(TSP 模型)的兼容性判定, 还需要对两类元素的映射规则加以明确规定: 一是消息抽象规则, 即在 TSP 模型中为处理异步通信模式而引入的消息队列在 UPPAAL 规约中的表示方式问题; 二是终态定义规则, 即如何在缺少终态定义的 UPPAAL 规约中定义合适的终止状态的问题。下面分别给出这两类元素的映射规则。

(1) 消息抽象规则

在 UPPAAL 规约中, 多个实时系统(时间自动机)之间通过信道实现同步, 一些工作将消息映射到 UPPAAL 规约中的信道, 对同步情形下的交互相关问题进行了研究。由于 TSP 模型是异步的, 这一做法显然不能适用。因此, 本文采用消息抽象的思想, 将消息映射到 UPPAAL 规约中的整型变量, 从而让 UPPAAL 能够支持(异步)TSP 模型的兼容性判定问题。

定义 7 消息抽象. 对于一组 TSP $P_1, P_2, \dots, P_n, P_i = (S_i, s_{i0}, F_i, M_i, X_i, I_i, Q_i, R_i)$ 且 $(1 \leq i \leq n)$, V 是一个

表 1 基本 BPEL 语句至 TSP 模型的映射规则

基本 BPEL 语句	TSP 模型
receive variable = “var” operation = “op”	$(s_b, op(var)?, true, \emptyset, true, s_e) \in R$ $s_b, s_e \in S$
reply variable = “var” operation = “op”	$(s_b, reply_op(var)!, true, \emptyset, true, s_e) \in R$ $s_b, s_e \in S$
invoke inputVariable = “ivar” operation = “op”	$(s_b, op(ivar)!, true, \emptyset, true, s_e) \in R$ $s_b, s_e \in S$
assign copy from expression = “term” to variable = “var”	$(s_b, \epsilon, true, \emptyset, true, s_e) \in R$ $s_b, s_e \in S$

表 2 结构化 BPEL 语句至 TSP 模型的映射规则

结构化 BPEL 语句	TSP 模型
sequence activity a1 activity a2 activity a3	$s_b \xrightarrow{a1} s_1, s_1 \xrightarrow{a2} s_2,$ $s_1 \xrightarrow{a3} s_e$ $s_b, s_1, s_2, s_e \in S$
switch case condition = “c1” activity a1 case condition = “c2” activity a2 otherwise activity a3	$(s_b, \epsilon, c1, \emptyset, true, s_1) \in R$ $(s_b, \epsilon, \neg c1 \wedge c2, \emptyset, true, s_2) \in R$ $(s_b, \epsilon, \neg c1 \wedge \neg c2, \emptyset, true, s_3) \in R$ $s_1 \xrightarrow{a1} s_e, s_2 \xrightarrow{a2} s_e,$ $s_3 \xrightarrow{a3} s_e$ $s_b, s_1, s_2, s_3, s_e \in S$
while condition = “c” activity a	$(s_b, \epsilon, c, \emptyset, true, s') \in R$ $(s_b, \epsilon, \neg c, \emptyset, true, s_e) \in R$ $s' \xrightarrow{a} s_b$ $s_b, s', s_e \in S$

表 3 并行 BPEL 语句至 TSP 模型的映射规则

并行 BPEL 语句	TSP 模型
flow activity = a1 activity = a2	$s_b \xrightarrow{a1} s_1, s_1 \xrightarrow{a2} s_e$ $s_b \xrightarrow{a2} s_2, s_2 \xrightarrow{a1} s_e$ $s_b, s_1, s_2, s_e \in S$
activity a1 source = “l1” transitionCondition = “c1” activity a2 source = “l2” transitionCondition = “c2” activity a target = “l1” target = “l2” joinCondition = “c”	$(s_b, \epsilon, x_{l1} \neq wait \wedge x_{l2} \neq wait, \emptyset, true, s_1) \in R$ $(s_1, \epsilon, c, \emptyset, true, s_2) \in R$ $(s_1, \epsilon, \neg c, \emptyset, true, s_e) \in R$ //Skip $(s_1, \epsilon, \neg c, \emptyset, true, s_{ex}) \in R$ //Throw Exception $s_2 \xrightarrow{a} s_e$ $s_b, s_1, s_2, s_e, s_{ex} \in S$

初始值都为 0 的整型变量的集合,定义抽象函数 $Abs: M \rightarrow V$,用于将消息映射到变量,其中: $M = M_1 \cup M_2 \cup \dots \cup M_n$,对于任意的 $m \in M, v \in V, m \rightarrow v$ 有:

- ①对于 $\forall (s_i, m, \phi, Y, \zeta, s'_i) \in R_i \wedge Polarity(m) = !$,有 $(s_i, m, \phi, Y, \zeta, s'_i) \rightarrow (s_i, v++, \phi, Y, \zeta, s'_i)$;
- ②对于 $\forall (s_i, m, \phi, Y, \zeta, s'_i) \in R_i \wedge Polarity(m)$

$= ?$,有 $(s_i, m, \phi, Y, \zeta, s'_i) \rightarrow (s_i, v--, \phi, Y, \zeta, s'_i)$.

从定义中可看出,消息抽象的基本思想是:将异步消息映射到 UPPAAL 规约中的整型变量,这些整型变量的初始值都为 0,通过判断整型变量的值与 0 的大小关系来模拟 TSP 模型之间的交互过程.例如: P_1 发送消息 m_1 (用 $m_1!$ 表示),与之交互的 P_2 需要接收消息 m_1 (用 $m_1?$ 表示)才能继续运行,此时将消息 m_1 抽象为一个初始值为 0 的整型变量 v_1 ,当消息 m_1 被发送时整型变量 v_1 自加 1,当消息 m_1 被接收时整型变量 v_1 自减 1.

表 4 时间 BPEL 语句至 TSP 模型的映射规则

时间 BPEL 语句	TSP 模型
pick onMessage variable = “v” operation = “op” activity a1 onAlarm for = “5” activity a2	$(s_b, op(v)?, true, \emptyset, true, s_1) \in R$ R $(s_b, \epsilon, x_t = 5, \emptyset, true, s_2) \in R$ $\forall (s', m, \phi, Y, \zeta, s_b) \in R, Y = \{x_t\}$ $I(s_b) \equiv x_t \leq 5$ $s_1 \xrightarrow{a1} s_e, s_2 \xrightarrow{a2} s_e$ $s_b, s_1, s_2, s_e \in S$
wait for = “5”	$(s_b, \epsilon, true, \{x_t\}, true, s') \in R$ $(s', \epsilon, x_t = 5, \emptyset, true, s_e) \in R$ $I(s') \equiv x_t \leq 5$ $s_b, s', s_e \in S$
activity a duration = “lessEqual(3)”	$(s_b, \epsilon, true, \{x_t\}, true, s') \in R$ $(s', a, x_t \leq 3, \emptyset, false, s_e) \in R$ $I(s') \equiv x_t \leq 3$ $s_b, s', s_e \in S$

表 5 典型 TPTL 公式至 TSP 模型的映射规则

TPTL 公式	TSP 模型
$\Box(A \rightarrow x. \Diamond(B \wedge \phi))$	$(s_b, A, true, \{x\}, true, s_1) \in R$ $(s_2, B, \phi, \emptyset, true, s_e) \in R$ $s_1 \xrightarrow{\omega} s_2$ $s_b, s_1, s_2, s_e \in S$

(2)终态定义规则

多个时间感知 Web 服务之间的一次正确交互要求参与交互的所有 TSP 模型都能到达各自的终止状态,但 UPPAAL 规约中并没有对终止状态给出明确定义,为此本文为每个 TSP 模型对应的 UPPAAL 规约定义一个区别于初始状态(初始状态也是一个终止状态)的终止状态 FINAL,用来表达对应的时间感知 Web 服务交互行为的正常终止.图 5 显示了 TSP 模型中终止状态至 UP-

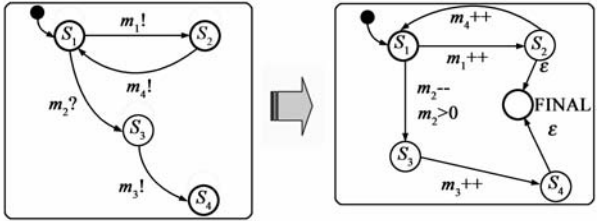


图5 在UPPAAL规约中定义终态

cancelled > 0) and (E \Diamond USR.FINAL and VTA.FINAL and HR.FINAL and HR.FINAL and USR.FINAL and VTA.FINAL and HR.FINAL and HR.FINAL imply

request = 0 and na = 0 and offer = 0 and ack = 0 and nack = 0 and cancelled = 0 and ticket = 0 and frequest = 0 and fna = 0 and fnack = 0 and fack = 0 and foffer = 0 and fticket = 0 and fcanceled = 0 and hrequest = 0 and hna = 0 and hnack = 0 and hack = 0 and hoffer = 0 and hticket = 0 and hcanceled = 0)

根据前述映射规则,将 VTA 交互场景中参与者的 TSP 模型转换至 UPPAAL 规约,转换结果如图 8 所示。

对于如图 8 所示的 VTA 交互场景中各参与者的 TSP 模型对应的 UPPAAL 规约,将完全兼容性、部分兼容性和完全不兼容性,这三类兼容性性质对应的 CTL 公式作为性质查询语言输入 UPPAAL,可得出 PUSR、PFR、PHR 和 PVTA 满足部分兼容性性质的结论,即时间感知 Web 服务 USR、FR、HR 和 VTA 之间存在交互失配。

事实上,USR、FR、HR 和 VTA 之间存在的交互失配恰好有可能是由时间冲突引起的,例如全局系统空间的一个死锁状态(U6, F8, H6, V25),可能拥有的时间约束为 $USR.z \geq 0 \wedge USR.y \geq 14 \wedge FR.v \geq 14 \wedge HR.w \geq 12 \wedge VTA.x \geq 7$ (如图 9 所示),显然 $USR.y \geq 14$ 与用户要求在 10h 以内收到 ticket 相矛盾。

5 相关工作分析与比较

文献[11]基于卫式自动机(GA)模型,研究了 Web 服务交互的可同步问题,GA 模型允许表达 Web 服务之间的异步通信,文献[8]基于扩展时间自动机模型研究了 Web 服务交互过程中时间属性的定量分析问题,但 ETA 模型只支持 Web 服务之间的同步消息交换。

在 Web 服务交互协议的兼容性判定问题方面,文献[13]基于有限状态机模型,研究了给定的两个 Web 服务交互协议之间的兼容性、等价性和可替换性问题,从多服务协作的角度来看,该工作存在一定的局限性。文献[14]提出了一种基于(演算的 Web 服务行为兼

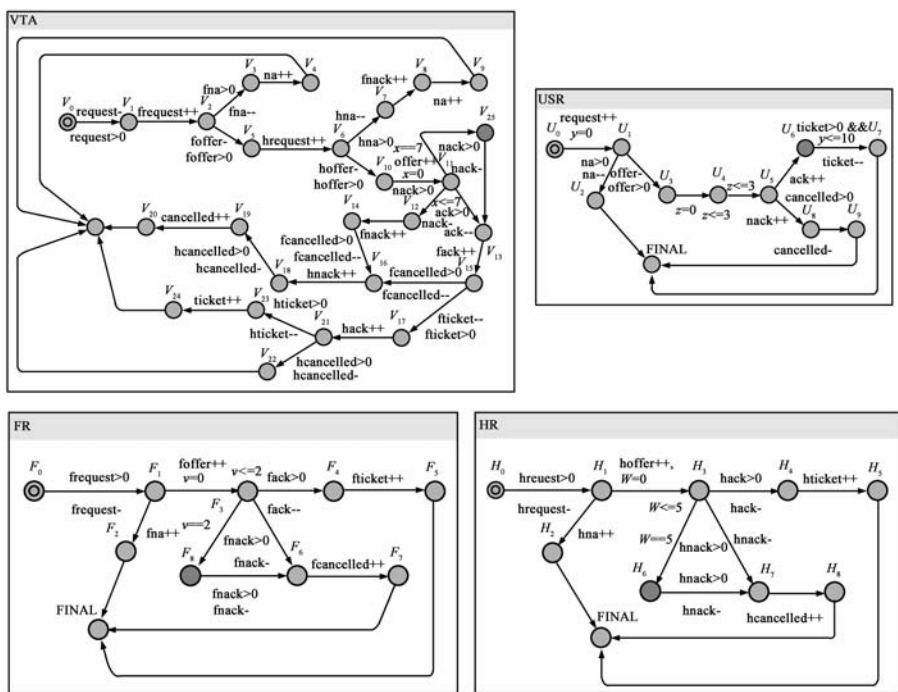


图8 VTA、USR、HR和FR的TSP模型对应的UPPAAL规约

容性的定性判定与定量计算方法,但该工作只局限在两个 Web 服务之间的同步消息交互情形。文献[7]基于 CCS 研究了两个 Web 服务在异步交互情形下的可替换性分析问题,提出的一致性关系可以保证替换后的多个 Web 服务合成是正确的。但上述工作均未涉及时间属性的定量分析。

在考虑定量时间属性的 Web 服务交互协议的兼容性分析方面,文献[16]考虑到了同步通信模式的局限性,提出了一种采用模型检测技术自动判定 TCP 模型兼容性性质的方法,本文提出的 TSP 模型与 TCP 模型相比,增加了对状态不变式的支持,也就是放宽了所有迁

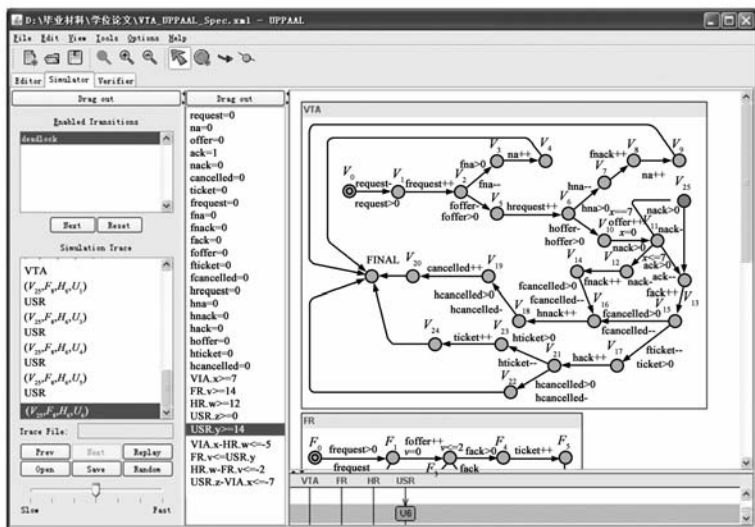


图9 时间冲突导致VTA服务交互死锁实例

移都是即时迁移的假设,这一改进使得 TSP 模型可以表达如长时间活动特征一类的时间属性。

6 结束语

随着 Internet 的快速发展与普及,软件服务作为一种新的软件实体,承担着封装各种资源、完成计算并对外提供服务的任务,被认为是网构软件^[17]的基本组成元素。而服务适配问题已成为目前软件服务的一个重要研究内容。本文将定量时间属性引入 Web 服务交互适配研究框架中,主要工作包括:(1)构建了能精确描述与分析时间感知 Web 服务交互行为的形式化模型;(2)采用 UPPAAL 模型检测技术实现了时间感知 Web 服务交互失配的自动检测;(3)通过实例进一步分析了上述方法在实际应用中的可行性和有效性。

进一步的研究工作包括如何在 TSP 模型中增加数据、概率等属性的描述;针对状态爆炸问题,如何改进现有模型检测技术,如考虑采用限界模型检测技术以提高检测效率;如何设计适配器交互协议的自动生成算法以及开发支持服务行为适配的原型系统等。

参考文献

- [1] Kongdenfha W, M.-Nezhad HR, Benatallah B, et al. Mismatch patterns and adaptation aspects: a foundation for rapid development of web service adapters[J]. IEEE Transactions on Services Computing, 2009, 2(2): 94 – 107.
- [2] Canal C, Poizat P, Salaün G. Model-based adaptation of behavioral mismatching components[J]. IEEE Transactions on Software Engineering, 2008, 34(4): 546 – 563.
- [3] Cámara J, Salaün G, Canal C. Composition and run-time adaptation of mismatching behavioral interfaces[J]. J UCS, 2008, 14(13): 2182 – 2211.
- [4] Kuang L, Deng S, Wu J, et al. Towards adaptation of service interface semantics[A]. Proc of the IEEE 7th Int'l Conf on Web Services (ICWS'09)[C]. Los Angeles, USA, 2009. 279 – 286.
- [5] Zhang GQ, Wang S, Rong M, et al. A model-based framework for adapting interaction mismatches of time-aware web services[A]. Proc of the 3rd Int'l Conf. on Service Science (ICSS2010)[C]. Hangzhou, China, 2010. 76 – 81.
- [6] Alur R, Henzinger TA. A really temporal logic[J]. Journal of the ACM, 1994, 41(1): 181 – 204.
- [7] 刘方方, 史玉良, 张亮, 等. 基于进程代数的 Web 服务合成的替换分析[J]. 计算机学报, 2007, 30(11): 2033 – 2039. Liu F, Shi Y, Zhang L, et al. Substitution analysis of web service composition via process algebra[J]. Chinese Journal of Computers, 2007, 30(11): 2033 – 2039. (in Chinese)
- [8] Pu G, Zhao X, Wang S, et al. Towards the semantics and verification of BPEL4WS[J]. Electronic Notes in Theoretical Computer Science, 2006, 151(2): 33 – 52.
- [9] Zhang J, Chung J Y, Chang C K, et al. WS-net: a petri-net based specification model for web services[A]. Pro. of the 2nd Int'l Conf on Web Services (ICWS'04)[C]. San Diego, USA, 2004. 420 – 427.
- [10] 李喜彤, 范玉顺. Web 服务流程相容性和相似性分析[J]. 计算机学报, 2009, 32(12): 2429 – 2437. Li X, Fan Y. Analyzing Compatibility and similarity of web service processes[J]. Chinese Journal of Computers, 2009, 32(12): 2429 – 2437. (in Chinese)
- [11] Fu X, Bultan T, Su J. Analysis of interacting BPEL web services[A]. Proc of the 13th Int'l Conf. on World Wide Web (WWW'04)[C]. New York, USA, 2004. 621 – 630.
- [12] Alur R, Dill DL. A theory of timed automata[J]. Theoretical Computer Science, 1994, 126(2): 183 – 235.
- [13] Benatallah B, Casati F, Toumani F. Representing, analysing and managing web service protocols[J]. Data & Knowledge Engineering, 2006, 58(3): 327 – 357.
- [14] 殷昱煜, 李莹, 邓水光, 等. Web 服务行为一致性与相容性判定[J]. 电子学报, 2009, 37(3): 433 – 438. Yin Y, Li Y, Deng S, et al. Determining on consistency and compatibility of web services behavior[J]. Acta Electronica Sinica, 2009, 37(3): 433 – 438. (in Chinese)
- [15] 林惠民, 张文辉. 模型检测: 理论、方法与应用[J]. 电子学报, 2002, 30(12A): 1907 – 1912. Lin HM, Zhang WH. Model checking: Theories, techniques and applications[J]. Acta Electronica Sinica, 2002, 30(12A): 1907 – 1912. (in Chinese)
- [16] N Guermouche, C Godart. Timed model checking based approach for web services analysis[A]. Proc of the 7th Int'l Conf. on Web Services (ICWS'09)[C]. Los Angeles, USA, 2009. 213 – 221.
- [17] 周航, 黄志球, 张广泉, 等. 基于 PTCPN 的网构软件建模与分析方法[J]. 软件学报, 2010, 21(6): 1254 – 1266. Zhou H, Huang ZQ, Zhang GQ, et al. Modeling and analysis of internetware based on PTCPN[J]. Journal of Software, 2010, 21(6): 1254 – 1266. (in Chinese)

作者简介



张广泉 男, 1965 年生于江苏连云港。苏州大学计算机科学与技术学院教授、博士/博士后、CCF 高级会员。研究方向为软件工程与形式化方法、服务计算、网络与分布式计算等。
E-mail: gqzhang@suda.edu.cn

戎玫 女, 1966 年生于重庆。副教授、博士。研究方向为软件体系结构与 SOA、形式化方法与协议分析等。

王 ■ 男, 1983 年生于江苏泰州。硕士。研究方向为 Web 服务技术、面向服务体系结构。