Web服务降级替换的一致性问题及量化研究

吴新星[1] 胡国胜[1] 陈仪香[2]
(上海电子信息职业技术学院计算机应用系 上海 201411)[1]
(华东师范大学教育部软硬件协同设计技术与应用工程研究中心 上海 200062)[2]

摘 要 在开放的网络环境下,软件系统的可信性受到了更大的挑战,软件系统的降级替换是提高其可信性的方法之一。基于进程代数研究了Web服务的降级替换问题。在原有进程代数的基础上,添加了超时处理算子和延时处理算子,给出了Web服务降级替换一致性条件,从而保证了合成Web服务中降级替换的正确性。进一步地,从量化角度对Web服务的降级替换进行了研究。

关键词 进程演算(CCS),Web服务,降级替换,量化
中图法分类号 TP3-0 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.2.017

Quantification and Conformance of Web Service Degraded Substitution

WU Xin-xing[1] HU Guo-sheng[1] CHEN Yi-xiang[2]
(Department of Computer, Shanghai Technical Institute of Electronics & Information, Shanghai 201411, China)[1]
(Soft/Hardware Co-design Engineering Research Center MoE, East China Normal University, Shanghai 200062, China)[2]

Abstract Software trustworthiness is facing greater challenges in open network environment, and one of approaches to improve the trustworthiness of software is degraded substitution. This paper mainly studied the Web service degraded substitution based on process algebra. In order to ensure the validity of Web service degraded substitution, a conformance condition was given based on the modified process algebra by introducing the time-out operator and the time-delay operator. Further, it studied the metric of Web service degraded substitution.

Keywords Calculus of communication system(CCS), Web service, Degraded substitution, Quantification

1 引言

当前,随着网络技术的发展,以Web服务为计算实体的网络软件系统已经成为主流,这类系统行为往往依赖于网络环境下的通信交互。而网络环境的开放性和不确定性,使得分析网络软件的行为变得更加具有挑战性,同时也对网络软件可信性提出了更高的要求。可生存性是Web服务的一个可信属性,对Web服务可生存性的研究是可信网络研究所要面临的重要科学挑战之一。文献[2]对系统可生存性的定义做过深入的研究和讨论。文献[3]指出在现实网络中,大多数情况下检测系统不可能检测到所有的入侵,从而使得保护措施不可能阻止所有的入侵,而同时关键基础设施又必须不间断地提供服务。因此,信息系统的可生存性也是信息基础设施及其应用的基本问题之一。可生存性主要是指(网络)系统在受到攻击、发生事故、出现故障时,系统依然能继续完成关键性任务的能力。可生存性的研究目的就是在网络受到攻击、入侵和破坏的情况下保证关键服务的持续性。而Web服务的设计和开发是注重其可重用性和可扩展性,因此,如何对Web服务进行准确高效的合成,怎样实现Web服务的恰当有效替换,并对替换的程度有准确的把握,使其能够满足Web服务系统可生存性的要求,提高Web服务系统的可信性,从而实现"高可信网络",已经成为领域内的一个热点问题。

根据Gartner研究所给的Web服务定义[4]:Web服务是通过Internet标准技术(HTTP,SMTP,FTP)传递的、松散耦合的软件组件。具体地说,Web服务是一种自描述的模块化软件应用程序,由Internet进行发布,并通过Internet协议来对它们进行查找、访问、订阅和调用等。关于Web服务的替换,很多学者进行了研究,如文献[5-13]等。文献[7,8]基于Petri网模型对Web服务可替换性问题进行了讨论研究。文献[9]提出了一个安全替换的形式化框架,并给出了一种基于策略的系统完整性验证方法。文献[10]提出了一个QoS驱动的动态替换算法。文献[12]通过自动机对基于WSCI[4]描述的Web服务建立形式化的行为模型,考虑了Web服务合成的相容性,分析和讨论了为使得Web服务的合成顺利进行,而对由Web服务自身的动态性导致的那些不可用Web服务进行替换。文献[6,11]根据文献[15]的subtype关系提出了基于进程代数的合成服务替换一致性关系,并证明了在这种关系下可以保证上下文无关的替换是正确的。其中,文

到稿日期:2014-03-31 返修日期:2014-06-16
吴新星(1981-),男,博士,助理研究员,CCF会员,主要研究方向为形式化方法与随机过程,E-mail:xinxingwu@gmail.com;胡国胜(1965-),男,博士,教授,主要研究方向为机器学习与负荷预测;陈仪香(1961-),男,博士,教授,主要研究方向为物联网、实时协同规范语言设计、程序语义模型和软件可信度量与评估理论。

献[6,11,12]所进行的讨论都是针对合成服务中服务替换的正确性，没有进一步涉及替换服务能否提供原来服务的全部功能，以及在功能上是否比原有服务有所减少等问题。文献[5]考虑到系统的生存性，提出了一种服务的降级替换方法，即当系统中某些服务不能正常提供(如遭遇外界的恶意攻击)时，改用一种可以保证原有服务中那些关键性服务仍能照常提供但是整体功能或性能上不如原有服务的降级服务系统。并且基于精化理论分析和讨论了服务的替换性问题，包括替换的相容性、替换的传递性等。但是文献[5]中没有进一步地从量化角度来讨论这种服务替换的程度，而且文献[5]中给出的服务降级替换理论也没有考虑到合成服务中服务降级替换的一致性问题。文献[13]基于 Web 服务执行时的触发条件和结果，给出了 Web 服务可替换度和替换效果的概念。但是文献[13]中的讨论没有涉及服务的替换正确性和合成服务替换的一致性等问题。

本文主要从量化角度出发，基于进程代数研究 Web 服务降级替换一致性问题以及 Web 服务的降级替换程度。具体的工作如下：

- 本文基于文献[6,11,15]中使用进程代数研究服务合成中替换服务正确性的方法，修改原有的进程代数，给出了降级替换的一致性条件，解决了合成服务中降级替换的正确性问题。
- 本文从量化角度研究了 Web 服务的降级替换。

在进行后面的讨论之前，我们先做如下的假定：

(i)Web 服务 S 为每种接收到的消息类型都设置缓存区[6,7]；

(ii)Web 服务 S 中内部选择和外部选择不能同时存在[6,7]；

(iii)Web 服务 S 的功能数或服务数主要是由外部选择能否对外界消息的接收来决定，内部选择只是实现功能的方式或是对接收的外部信息的处理过程；

(iv)Web 服务 S 接收到外界消息之后可能会有一段延时，之后再对消息进行处理，但 Web 服务 S 发送消息时不会存在任何延时；

(v)Web 服务系统降级的讨论局限于服务功能数的减少和服务处理的延后两种情况[5]；

(vi)发送消息不花费时间；

(vii)另外，我们约定，这里的符号 S,S' 以及 $S_1,S_2,\cdots$ 指代 Web 服务。

2 基于进程代数的 Web 服务降级替换度量

本节主要基于文献[6,11,15]中用进程代数研究合成替换服务正确性的方法来讨论降级 Web 服务的合成替换。采取 Web 服务的降级替换是保证系统可生存性的重要措施之一，这里，我们所涉及降级服务(概念的)讨论同文献[5]，即是指，以一个服务替换系统中原有的某个服务，替换之后系统的服务功能会减少或是发生服务处理的延时，但是还是能够提供关键性的服务来满足基本的需求。具体见下面的例子。

例 1　从图 1—图 3 中可以看到，订票服务系统的一个版本是图 1 中所有的功能都能提供，当遇到网络或是系统出现问题时，系统管理员会使用降级服务系统进行替换。如，可以改换成带全虚线(见图 2)的订票服务系统(此时虽然“飞机”订票服务不可以进行，但是基本的一个订票流程还是能够完成)；或是带半虚线(见图 3)的订票服务系统(此时“银行转帐”可能会出现延时，但是还是能够成功进行)。现在我们感兴趣的问题是：

- 带全虚线的订票服务系统在多少程度上可以替换原有的订票服务系统呢?
- 带半虚线的订票服务系统在多少程度上可以替换原有的订票服务系统呢?

图 1—图 3 中的符号说明如下(这些符号的意义同样适用于后文)。

- μ_1 指“汽车”订票服务在 3 种服务中被选择的权重，$\mu_{1\text{支付宝支付}}$ 指支付宝支付在“汽车”订票服务支付方式中的权重，$\mu_{1\text{银行转帐}}$ 指银行转账在“汽车”订票服务支付方式中的权重；
- μ_2 指“飞机”订票服务在 3 种服务中被选择的权重，$\mu_{2\text{支付宝支付}}$ 指支付宝支付在“飞机”订票服务支付方式中的权重，$\mu_{2\text{银行转账}}$ 指银行转账在“飞机”订票服务支付方式中的权重，$\mu_{2\text{货到付款}}$ 指货到付款在“飞机”订票服务支付方式中的权重；
- μ_3 指“火车”订票服务在 3 种服务中被选择的权重，$\mu_{3\text{支付宝支付}}$ 指支付宝支付在“火车”订票服务支付方式中的权重，$\mu_{1\text{银行转账}}$ 指银行转账在“火车”订票服务支付方式中的权重。

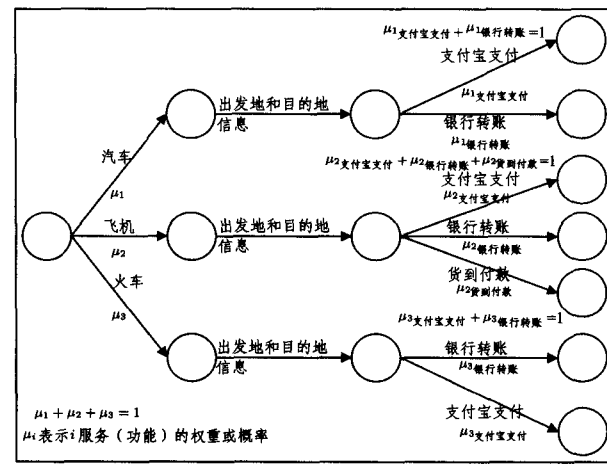

图 1　订票服务系统(a)

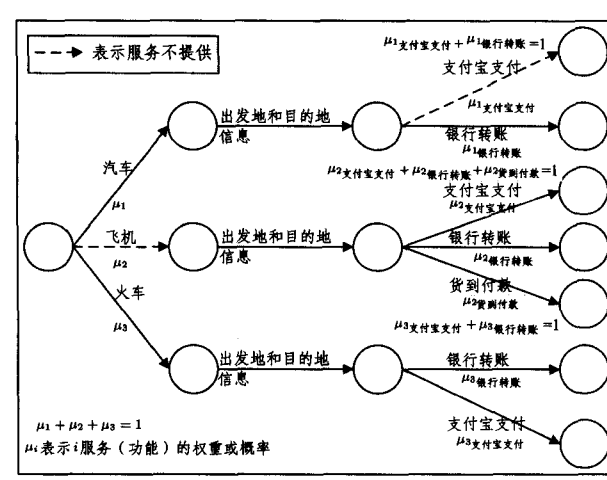

图 2　订票服务系统(b)：服务停用

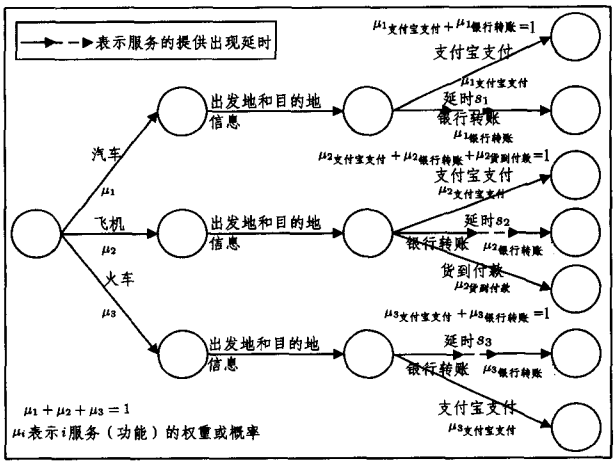

图 3　订票服务系统(c)：服务延时

前面已经提到，已有的关于提高系统可生存性、实现服务降级替换的讨论只是局限于定性的分析，而且已有的分析即使讨论了服务的替换度，也没有进一步考虑合成服务中服务降级替换的一致性和正确性。因此，我们下面主要是讨论合成服务中服务降级替换的一致性和正确性，且对服务降级替换的程度进行量化研究和分析，使得原本抽象的、分散的、不可捉摸的服务替换属性转化为具体的、严整的、可以大体掌握的认知与形象，从而更便于在实际中人们理解和精确控制，实现更好、更精准的服务降级替换，以提高系统的可生存性，有助于实现“高可信网络”。

接下来讨论所要用到的一些说明、定义和定理(主要来自文献[6,11])如下。

Web 服务 $S_1,S_2,\cdots,S_n$，对应 CCS 的进程 $M_1,M_2,\cdots,M_n$，其合成记为

$$P=M_1 \parallel M_2 \parallel \cdots \parallel M_n$$

用 R 表示 Web 服务 S_1 的合成环境或是合成上下文，此时合成可记为 $P=M_1 \parallel R$。

定义 1[6,7]　Web 服务 $S_1,S_2,\cdots,S_n$，由 CCS 的进程 $M_1,M_2,\cdots,M_n$ 表示。如果 P 经过有限次的隐藏操作后到达失败状态，即 P 经过有限次的隐藏操作后不处于终止状态 0，而又无法再继续执行下去，那么我们就说，$S_1,S_2,\cdots,S_n$ 的合成 $P=M_1 \parallel M_2 \parallel \cdots \parallel M_n$ 是错误的。

定义 2[6,7]　Web 服务 $S_1,S_2,\cdots,S_n$，由 CCS 的进程 $M_1,M_2,\cdots,M_n$ 表示。如果 P 经过有限次的隐藏操作后不会到达失败状态，那么我们就说，$S_1,S_2,\cdots,S_n$ 的合成 $P=M_1 \parallel M_2 \parallel \cdots \parallel M_n$ 是正确的。

定义 3[6,7](一致性)　用 CCS 的进程 P 和 Q 表示 Web 服务 S_1 和 S_2。称 Q 对 P 是一致的，记为 $Q \triangleright P$，如果下面条件满足：

(i)如果 $Q \xrightarrow{\alpha} Q'$，那么存在 P'，使得 $P \xrightarrow{\tau^*.\alpha.\tau^*} P'$，且有 $Q' \triangleright P'$[1)]

(ii)如果 P 有外部选择

$$a_1?.P_1+a_2?.P_2+\cdots+a_n?.P_n$$

那么对每一个 i 且 $1\leqslant i\leqslant n$，存在 Q_i，使得

$$(Q \xrightarrow{\tau^*.a_i?} Q_i)$$

(iii)如果 P 有内部选择

$$\tau.b_1!.P_1+\tau.b_2!.P_2+\cdots+\tau.b_n!.P_n$$

那么存在一个 i_0，$1\leqslant i_0\leqslant n$ 和 Q_{i_0}，使得

$$Q \xrightarrow{\tau^*.b_{i_0}!} Q_{i_0}$$

(iv)如果 P 终止，那么 Q 也终止。

定理 1[6,7]　Web 服务 S 和 S_1'，由 CCS 的进程 M_1 和 M_{sub1} 表示。R 表示 Web 服务 S_1 的合成环境。如果 $M_{sub1} \triangleright M_1$ 且 $M_1 \parallel R$ 是正确的，那么 $M_{sub1} \parallel R$ 也是正确的。

2.1　CCS 的修改——超时和延时

在文献[6,11,15]中都使用 CCS 来形式化地描述和分析 Web 服务的合成替换。在文献[15]中给出 subtype 关系，在文献[6,11]中给出一致性关系(定义 3)来说明：若新服务与将要被替换的、参与合成的服务之间存在这种关系，那么替换后的服务合成仍然是正确的，并且替换是上下文无关的。给出了这种关系之后，文献[6,11]又证明了这种一致性关系保证的上下文无关的替换是正确的(定理 1)。

1)超时处理算子

因为在文献[6,11,15]中的讨论没有考虑替换后服务功能数减少的情况，所以 subtype 关系和一致性关系都不能很好地刻画 Web 服务降级替换的合成正确性。如下面的例子：

例 2[带全虚线(见图 2)的订票服务系统]　继续前面的例 1，带全虚线的订票服务系统表示此时“飞机”订票服务不能正常进行，在原有订票服务系统降级到带全虚线的订票服务系时，实际上这种替换都是不满足文献[6,11]中的一致性关系和文献[15]中的 subtype 关系的，因为一致性关系和 subtype 关系都要求：若服务 S_2 能替换服务 S_1，那么 S_1 服务的任意全局选择分支(外部选择)，服务 S_2 都要能提供。但是很显然，“飞机”订票服务作为全局选择分支(外部选择)，带全虚线的订票服务系统是不能提供的。因此，在已有的一致性关系和 subtype 关系下，实际上认为这种替换是不能保证合成服务替换的正确性的，这种替换可能会引起死锁。

而这里所讨论的 Web 降级替换就有涉及到替换后服务功能数减少的情况，注意到上面的这种情况，需要引入新的算子。下面是给出的关于超时服务的处理算子：

$$a?(x)_P \Gamma_t 0 \tag{1}$$

其中，$t\in[0,+\infty)$，t 称为(关于 $a?(x)$ 和 P 的)超时阈值。

式(1)的意思是：接收到由信道 a 发送来的信息 x 之后，如果在 t 单位时间内进程 P 不对其进行处理，那么整个进程就终止。

超时算子的引入可以保证上面例 2 中替换的正确性，因为带全虚线的订票服务系统不能提供“飞机”订票服务，所以当接收到“飞机”订票服务的请求时是不能够进行处理的，那么现在可以进行这样的设置：如果接收到的外部信息在 t_0 单位时间内不能处理，那么整个进程就终止，即 $a?()_P \Gamma_t 0$。

同时，在例 2 中，如果不考虑具体的票种以及其他具体的支付方式等，仅仅从是否能够进行一次成功的订票(实现路径)来看，带全虚线的订票服务系统对原有订票服务系统的服务降级替换程度是 3/7。显然这是不恰当的，“汽车”订票时支付宝支付服务不能提供和“飞机”订票服务不能提供从功能实现影响上来说是不一样的。因为对于前者，银行转帐服务能进行，所以“汽车”订票服务还是能进行的，只是不能进行支

1) τ^* 表示零或多个 τ 动作。

付宝支付而已，而后者则是“飞机”这种订票服务不能提供。因此，在考虑降级服务的替换程度时，我们应该体现 Web 服务的权重之分：对于某服务系统而言，可能有些服务是重要的关键性的服务，而有些服务却是相对次要的。

下面基于对 Web 服务权重的考虑，给出对于上面一类服务系统降级替换程度的量化定义。带全虚线的订票服务系统对原有订票服务系统的服务降级替换度为

$1\cdot\mu_1\cdot\mu_{1银行转帐}+1\cdot\mu_3$

2)延时处理算子

上面的超时处理算子是为了处理服务超时的情况而给出的。实际上，超时处理算子的引入是为了对在合成服务降级替换时，原有的服务进行降级替换后不能再继续提供时的一种形式化处理技巧。在引入了超时处理算子后，自然就有这样的问题：如何来描述 Web 服务的处理在超时阈值的范围之内但是发生了延时的情况。因为在文献[6,11,15]使用进程代数研究服务替换问题的讨论中没有涉及延时情况，所以 subtype 关系和一致性关系同样都不能很好地刻画 Web 服务降级替换的合成正确性。在 subtype 关系和一致性关系的观点下，可能会认为这种替换不能保证合成服务替换的正确性。如下面的例子：

注 1[带半虚线(见图 3)的订票服务系统] 继续前面的例 1，带半虚线的订票服务系统表示此时“银行转帐”可能会出现延时，但是还是能够成功进行。在前面，原有订票服务系统降级到带全虚线的订票服务系统时，替换程度可以从服务能否完成角度来考虑量化。那么此时在原有订票服务系统降级到带半虚线的订票服务系统时，如何来刻画这种降级替换的程度呢？

在回答例 1 最后给出的这个问题之前，我们先引入下面关于延时服务的处理算子：

$$P\ \boxed{s}\gg\xrightarrow{a?(x)}P' \tag{2}$$

其中，$S\in[0,+\infty]$。

式(2)的意思是：接收到由信道 a 发送来的信息 x 之后，进程 P 等待 s 单位时间再对信息进行处理。在不发生混淆的情况下，式(2)简记为 $a?(x)\xrightarrow{\boxed{s}}P$。

注 2 假定 $\exists t_0\in[0,+\infty)$，$a?(x)_P\ \Gamma_{t_0}\ 0$。

(i)若 $s\in[0,+\infty]$且 $s\geqslant t_0$，则有 $P\xrightarrow{\tau^*}0$；

(ii)若 $s\in[0,+\infty]$且 $0\leqslant s<t_0$，则有 $P\ \boxed{s}\gg\xrightarrow{a?(x)}P'$；特别地，若 $s=0$，则有 $P\xrightarrow{a?(x)}P'$。

注 3 注意到 CCS 中的推导规则：

$$\frac{P\xrightarrow{a?(y)}P'\quad Q\xrightarrow{a!(x)}Q'}{P\parallel Q\xrightarrow{\tau}P'\parallel Q'}$$

类似于式(2)，我们定义关于内部动作 τ 的延时处理：

$$P\ \boxed{s}\gg\xrightarrow{\tau}P',s\in[0,+\infty]$$

而根据注 2(ii)，我们定义关于发送消息的延时处理：

$$P\ \boxed{s}\gg\xrightarrow{a!(x)}P',s=0$$

注 4 根据注 3 的讨论，我们给出如下规则：

$$\frac{a?(y)_P\ \Gamma_{t_0}\ 0\quad Q\xrightarrow{a!(x)}Q'}{\tau_{P\parallel Q}\Gamma_{t_0}\ 0} \tag{3}$$

式(3)是说：P 和 Q 进行交互时，如果 P 对于 Q 发送来的消息，因为处理超时(t_0)而终止时，整个合成 $P\parallel Q$ 也因处理超时(t_0)终止。

现在，给出对于延时服务降级替换程度的量化定义。前面已经定义了超时处理算子，于是假定 $\exists t_0\in[0,+\infty)$，$a?(银行转帐)_P\Gamma_{t_0}\ 0$，且 $s_i\in[0,+\infty]$，$i=1,2,3$。那么，带半虚线的订票服务系统对原有订票服务系统的服务降级替换度为

$$\begin{aligned}&1\cdot\mu_1\cdot\mu_{1支付宝支付}\\&+1\cdot\mu_1\cdot\mu_{1银行转帐}\cdot(1-\hat{\chi}_{t_0}(s_1))\\&+1\cdot\mu_2\cdot(\mu_{2支付宝支付}+\mu_{2货到付款})\\&+1\cdot\mu_2\cdot\mu_{2银行转帐}\cdot(1-\hat{\chi}_{t_0}(s_2))\\&+1\cdot\mu_3\cdot\mu_{3支付宝支付}\\&+1\cdot\mu_3\cdot\mu_{3银行转帐}\cdot(1-\hat{\chi}_{t_0}(s_3))\end{aligned} \tag{4}$$

注 5 延时服务对原服务的替换度计算使用 $1-\hat{\chi}_{t_0}(s_i)$ ($i=1,2,3$)是因为

(i)反映了要完成服务的剩余有效时间占原总的有效时间的比例，即当 $s_i<t_0$，$i=1,2,3$ 时，$1-\hat{\chi}_{t_0}(s_i)=\frac{t_0-s_i}{t_0}$，$i=1,2,3$；

(ii)由注 2 知道，若 $s_i\geqslant t_0$，$i=1,2,3$ 时，也就是说替换之后的系统不能再继续进行原有的服务时，在式(4)中也就有 $1-\hat{\chi}_{t_0}(s_i)=0$，$i=1,2,3$。即，对此服务的替换度为 0。

注 6 若不提供某 Web 服务，如，例 2“汽车”订票服务中的支付宝支付，那么可用延时处理算子表示如下：

$a?(支付宝支付)\xrightarrow{\boxed{+\infty}}P$

由上面的讨论可知，修改后的 CCS 语法如下：

$P::=0\mid\alpha.P\mid P+P\mid P\parallel P\mid P\backslash L$

$\quad\mid P[f]\mid a?(x)_P\ \Gamma_t\ 0\mid P\ \boxed{s}\gg\xrightarrow{\alpha}P'$

$\alpha::=a?(x)\mid a!(x)\mid\tau$

其中，$s\in[0,+\infty]$，$t\in[0,+\infty)$。

2.2 降级服务替换度

上一节中我们考虑到服务降级替换很可能不满足文献[6,11]中的一致性关系和文献[15]中的 subtype 关系。为了研究服务降级替换的合成正确性，我们引入了超时处理算子，并且为了进一步研究降级服务替换程度的量化，我们又引入了延时处理算子。这一节将主要研究降级服务替换度的形式化。下面首先结合上一节的注 2 给出关于降级一致性和降级服务的替换度等的一些相关定义。

定义 4(降级一致性) 用修改后的 CCS 进程 P 和 Q 表示 Web 服务 S_1 和 S_2。我们称 Q 对 P 是降级一致的，记为 $Q\Downarrow P$，如果下面条件满足：

(i)假定 $a?(x)_P\ \Gamma_{t_0}\ 0$。如果 $Q\ \boxed{s}\gg\xrightarrow{\alpha}Q'$ 且 $s<t_0$，那么存在 P'，使得 $P\xrightarrow{\tau^*\cdot\alpha\cdot\tau^*}P'$，且有 $Q'\Downarrow P'$。

(ii)如果 P 有外部选择

$a_1?.P_1+a_2?.P_2+\cdots+a_n?.P_n$

那么对每一个 i 且 $1\leqslant i\leqslant n$，存在 Q_i 和 $t_i\in[0,+\infty)$，$s_i\in[0,+\infty]$且 $s_i<t_i$，使得

$$(a_i?_Q\Gamma_{t_i}\ 0)\vee(Q\ \boxed{s_i}\gg\xrightarrow{a?}Q_i)$$

(iii)如果 P 有内部选择

$\tau.b_1!.P_1+\tau.b_2!.P_2+\cdots+\tau.b_n!.P_n$

那么存在一个 i_0，$1\leqslant i_0\leqslant n$ 和 Q_{i_0}，以及超时阈值 t_{i_0}，$s_{i_0}<t_{i_0}$，使得

$$Q\boxed{s_{i_0}}\gg\xrightarrow{\tau.b_{i_0}!}Q_{i_0}$$

(iv)如果 P 终止，那么 Q 也终止。

定义 5(降级服务的替换度) 若 $Q\Downarrow P$，并且满足

假定 $\exists l,m_k$ 且 $a_{kj}?,j=1,2,\cdots,m_k,k=1,2,\cdots,l$。记

$\alpha_k=\tau^*.(a!)^*.\tau^*.a_{k1}?.\tau^*.(a!)^*.\tau^*.\cdots.\tau^*.(a!)^*.\tau^*.a_{km_k}?.\tau^*{}^{1)}.(a!)^*.\tau^*$

又假定 $\exists t_{ki}\in[0,+\infty)$，使得 $a_{ki}?(x)_Q\Gamma_{t_{ki}}0$，且有 $s_{ki}\in[0,+\infty]$，$s_{ki}<t_{ki}$，$i=1,2,\cdots,m_k$。如果有 $P\xrightarrow{\alpha_k}0$ 成立，那么

$$Q\boxed{\sum_{i=1}^{m_k}s_{k_i}}\gg\xrightarrow{\alpha_k'}0$$

其中，$\alpha_k'=\tau^*.(a!)^*.\tau^*.\alpha_k.\tau^*.(a!)^*.\tau^*$，$k=1,2,\cdots,l$。

进一步地，再设 $\mu_{kj}>0$ 为 $a_{kj}?$ $(j=1,2,\cdots,m_k)$的权重或是发生的概率，且(可参见图 1—图 3)

$$\sum_{k=1}^{l}\prod_{j=1}^{m_k}\mu_{kj}=1$$

记

$$\kappa=\sum_{k=1}^{l}\prod_{j=1}^{m_k}\mu_{kj}\cdot(1-x_{t_{kj}}^{\wedge}(s_{kj})) \tag{5}$$

于是，我们称 Q 对 P 是 κ 可降级替换的，记为 $Q\Downarrow_\kappa P$。其中，κ 是 Q 对 P 的降级替换度。

注 7 定义 5 揭示了系统在进行降级替换后，相对于原系统，我们对降级系统的可用程度。

注 8 降级替换一般指做了替换之后系统的服务或功能是原来系统子集的替换。若 $Q\Downarrow_\kappa P$，那么我们就说原系统在做了降级替换之后，还能够提供原来系统 κ 的服务或是功能，即替换后的系统 κ 是可执行的。我们认为降级替换是有意义的，如果降级替换之后系统的服务或功能是原来系统的一个非空子集。

注 9 这里从上面的定义 4 和定义 5 可以看到，我们在考虑降级服务替换时，原系统(见图 1) 是不带有延时服务的。实际上，原系统含有带延时的服务和原系统不含有带延时的服务，在进行降级替换时本质上是一致的：

设有 Web 服务 S，对此服务的替换服务记为 S'。记 μ_S 是服务 S 的权重，s_0 为服务 S 的延时，s_1 为服务 S' 的延时。假定 $\exists t_0\in[0,+\infty)$，$a?(x)_{P_S}\Gamma_{t_0}0$ 和 $s_0\in[0,+\infty]$，$s_0<t_0$。按定义 5 知替换度为

$$\begin{aligned}\mu_S\times\frac{1-\chi_{t_0}^{\wedge}(s_1)}{1-\chi_{t_0}^{\wedge}(s_0)}&=\begin{cases}\mu_S\times\dfrac{t_0-s_1}{t_0-s_0}, & s_1<t_0\\ 0, & s_1\geqslant t_0\end{cases}\\&=\begin{cases}\mu_S\times\dfrac{(t_0-s_0)-(s_1-s_0)}{t_0-s_0}, & s_1<t_0\\ 0, & s_1\geqslant t_0\end{cases}\\&=\begin{cases}\mu_S\times\dfrac{k_0-k_1}{k_0}, & k_1<k_0\\ 0, & k_1\geqslant k_0\end{cases}\qquad k_1=s_1-s_0,k_0=t_0-s_0\end{aligned}$$

$$=\begin{cases}1-\chi_{k_0}^{\wedge}(k_1), & k_1<k_0\\ 0, & k_1\geqslant k_0\end{cases} \tag{6}$$

式(6)说明，从相对意义上看，进行降级替换时，原系统含有带延时的服务和原系统不含有带延时的服务没有什么本质不同。因此，这里为简单起见，我们考虑的 Web 服务降级替换都是对于原系统不含有带延时服务的降级替换。

引理 1 用修改后的 CCS 进程 P 和 Q 表示 Web 服务 S_1 和 S_2。R 表示 Web 服务 S_1 的合成环境。如果 $P\parallel R$ 是正确的，并且 $Q\Downarrow P$，那么 $Q\Downarrow R$ 也是正确的。

证明：采用类似于文献[6,11]的方法进行证明。

(反证法)假设 $Q\parallel R$ 是错误的。则由定义 1 知，$\exists Q'$ 和 R'，使得

$$Q\parallel R\xrightarrow{\tau^*}Q'\parallel R'$$

但 $Q'\parallel R'$ 处于失败状态。即，不能到达终止状态，又无法继续执行下去。

令 $a=a_1.a_2.\cdots.a_k$，$\bar{a}=\bar{a_1}.\bar{a_2}.\cdots.\bar{a_k}$，其中，$\bar{a_i}$ $(1\leqslant i\leqslant k)$是 a_i $(1\leqslant i\leqslant k)$的逆操作，若 a_i $(1\leqslant i\leqslant k)$表示 Q 中所进行的发生或接收消息的操作[2)]，使得

$$Q\boxed{s_Q}\gg\xrightarrow{\alpha}Q'(\text{其中，假定 }a_Q\Gamma_{t_0}0,s_Q<t_0)$$

且

$$R\xrightarrow{\bar{a}}R'$$

注意到 $Q\Downarrow R$，由定义 4(i)可知，$\exists P'$，有

$$P\xrightarrow{\tau^*.a.\tau^*}P'$$

且

$$Q'\Downarrow R' \tag{7}$$

从而有

$$P\parallel R\xrightarrow{\tau^*}P'\parallel R'$$

由假设 $P\parallel R$ 正确以及定义 1、定义 2 可知

$$P'\parallel R' \tag{8}$$

是正确的。

(a)Q'等待接收来自外界的消息 b，而 R'不能发送消息 b。

注意到式(7)以及定义 4，P'必存在等待接收消息 b 的操作。由式(8)可知，R'必能发送消息 b 与假设矛盾。

故由定义 4(ii)可知，Q'能接收一切来自 R'发送的消息，由注 2 和注 4 知道，Q'即使不能对其信息真正地进行处理，也能做出反馈而转向终止，从而使得 $Q'\parallel R'$终止。故 $Q'\parallel R'$可以继续执行下去或是达到终止状态。

(b)Q'发送消息 b，而 R'无法接收消息 b。

注意到式(7)以及定义 4，P'必存在发送消息 b 的操作。根据式(8)，R'必能接收消息 b 与假设矛盾。

故由定义 4(iii)可知，Q'发送的消息，R'必能接收。由注 2 和注 4 知道，R'即使不能对其信息真正地进行处理，也能做出反馈而转向终止，从而使得 $Q'\parallel R'$终止。故 $Q'\parallel R'$可以继续执行下去或是达到终止状态。

(c)Q'处于终止状态，而 R'不处于终止状态。

(下转第 94 页)

1) 表示零或多个 $a!$动作。

2) 例如，a_i $(1\leqslant i\leqslant k)$是发送消息操作，那么$\bar{a_i}$ $(1\leqslant i\leqslant k)$就表示接收信息操作。

[16] Detection of Malicious PDF Files Based on Hierarchical Document Structure[C]// Proceedings of the Network and Distributed System Security Symposicum(NDSS). 2013

[17] Lee W, Stolfo S, Mok K. A data mining framework for building intrusion detection models[C]// IEEE Symposium on Security and Privacy. 1999:120-132

[18] Mahoney M, Chan P. Learning rules for anomaly detection of hostile network traffic[C]// International Conference on Data Mining (ICDM). 2003

[19] Gu G, Porras P, Yegneswaran V, et al. BotHunter: Detecting malware infection through IDS-driven dialog correlation[C]// USENIX Security Symposium. 2007:167-182

[20] Canali D, Cova M, Vigna G, et al. Prophiler: a fast filter for the large-scale detection of malicious Web pages[C]// International Conference on World Wide Web (WWW). 2011:197-206

[21] Breiman L, Friedman J, Olshen J, et al. Classification and Regression Trees[M]. Wadsworth, 1984

[22] Cohen W. Fast effective rule induction[C]// International Conference on Machine Learning (ICML). 1995:115-123

[23] Quinlan J. C4. 5: Programs for Machine Learning[M]. Morgan Kaufmann, 1992

[24] Duda R O, Hart P E, Stok D G. 模式分类[M]. 李宏东, 姚天翔, 等译. 北京: 机械工业出版社, 2003

[25] https://www.virustotal.com/

(上接第 85 页)

注意到式(7)和定义 4(i),有 P' 亦处于终止状态。故由式(8)可知,R' 亦处于终止状态,与假设矛盾。

故 $Q' \parallel R'$ 处于终止状态。

(d)Q' 不处于终止状态,而 R' 处于终止状态。

注意到式(8),有 P' 亦处于终止状态,由定义 4(iv)知,Q' 亦处于终止状态,与假设矛盾。

故 $Q' \parallel R'$ 处于终止状态。

综合(a)—(d)得到,与 $Q \parallel R$ 是错误的假设矛盾。故,如果 $P \parallel R$ 是正确的,并且 $Q \Downarrow R$,那么也 $Q \parallel R$ 是正确的。证毕。

定理 2 用修改后的 CCS 进程 P 和 Q 表示 Web 服务 S_1 和 S_2。R 表示 Web 服务 S_1 的合成环境。如果 $P \parallel R$ 的服务或功能是成功进行的,并且 $Q \Downarrow \kappa R$,那么 $Q \parallel R$ 的服务或功能是 κ 可成功进行的。

证明:由定义 5 和引理 1 知,结论显然成立。证毕。

推理 1 假定 $Q \Downarrow \kappa R$。当 $\kappa > 0$ 时,那么 Q 对 P 的降级替换是有意义的。

证明:由 $\kappa > 0$ 及式(5)可知,存在 k_0,使得

$$\prod_{j=1}^{m_{k_0}} \mu_{kj} \cdot (1 - \chi_{t_{k_0 j}}(s_{k_0 j}^{\wedge})) > 0$$

即,至少存在一条路径可以实现原有服务系统中的一种服务或功能。由注 8 可知结论成立。证毕。

结束语 本文主要讨论了 Web 服务降级替换的一致性问题,并从量化角度分析了 Web 服务的降级替换:

· 以 3 种订票服务系统(正常系统、带延时的系统和某些服务不能提供的系统)为例,讨论了以原有进程代数研究服务系统的降级替换合成正确性时会碰到困难,而且原有的降级替换理论也无法精确地从量化角度给出服务降级替换的程度。于是,我们修改原有的进程代数,引入了超时处理算子和延时处理算子。

· 基于修改后的进程代数,给出了降级一致性定义和降级服务的替换度定义,保证了服务降级替换的合成正确性,进一步地从量化角度对 Web 服务的降级替换进行了讨论。

接下来,我们会在本文工作的基础上进一步考虑所给出的理论模型在具体实际中的应用。

参 考 文 献

[1] 刘克, 单志广, 王戟, 等. "可信软件基础研究"重大研究计划综述[J]. 中国科学基金, 2008(3):145-151

[2] Knight J C, Strunk E A, Sullivan K J. Towards a Rigorous Definition of Information System Survivability[C]// 3rd DARPA Information Survivability Conference and Exposition (DISCEX 2003). 2003, 1:78-89

[3] 余智华, 林思明, 陈海强. 网络安全——可生存性研究及网络建模[J]. 信息技术快报, 2005, 3(12):11-23

[4] Nagappan R, Skoczylas R, Sriganesh R P. Developing Java Web Services[M]. Wiley, 2002

[5] Dumas M, Yang Y, Zhang L. Improving Web Service Survivability Via Gracefully Degraded Substitution[C]// 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. Toronto, Ontario Canada, 2010: 597-600

[6] Liu F F, Shi Y S, Zhang L, et al. Analysis of Web Services Composition and Substitution Via CCS[C]// Proceedings of the DEECS'06. San Francisco, CA, USA, Springer-Verlag, Lecture Notes in Computer Science, 2006, 4055:236-245

[7] Bourouz S, Zeghib N. Verifying Web services substitute ability using open colored nets reduction techniques[C]// 2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO). IEEE, 2013:1-5

[8] 郭峰, 魏光. 基于 Petri 网的 Web 服务描述及其可替换性分析[J]. 计算机集成制造系统, 2013, 19(6):1423-1432

[9] Nakajima S. Safe Substitution of Components in Self-adaptive Web Applications[C]// 2013 20th Asia-Pacific Software Engineering Conference (APSEC). IEEE, 2013, 1:388-395

[10] 宋仲凯, 张晓容, 殷昱煜. QoS 驱动的服务动态替换方法[J]. 计算机应用与软件, 2012, 29(1):27-30

[11] 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

[12] 史玉良, 王海洋, 张亮, 等. Web 服务合成的相容性和替换性分析[J]. 计算机研究与发展, 2007, 44(11):1955-1961

[13] 刘莹, 张一川, 张斌, 等. 基于行为效果的服务可替换性分析[J]. 计算机研究与发展, 2010, 47(8):1442-1449

[14] W3C Working Group. Web Services Choreography Interface (WSCI) 1.0 [Z]. World Wide Web Consortium, W3C Note 8 Aug. 2002. URL. http://www.w3.org/TR/wsci

[15] Brogi A, Canal C, Pimentel E, et al. Formalizing Web Services Choreographies[J]. Electronic Notes in Theoretical Computer Science, 2004, 105:73-94