基于可信链模型的Web服务组合研究

高洪皓[1]，李莹[2]，张渊源[3]

(1.上海大学 计算机工程与科学学院，上海 200072；2.浙江大学 计算机科学与技术学院，浙江 杭州 315100；
3.浙江中医药大学 信息技术学院，浙江 杭州 310053)

摘 要：针对Web服务组合的可靠性检验问题，结合定理证明方法，改进了模型检验框架，提出了一种基于可信链模型的Web服务组合方法。采用扩展的标签迁移系统（ELTS）模型描述Web服务组合行为和交互协议，为每个服务状态和迁移行为分别标识可信集和迁移约束，使得服务组合过程转化为可信集演算过程。引入谓词变换函数WP(Q,R)推导服务组合的最弱前置条件，通过检验可信集和最弱前置条件的蕴含关系，证明组合的正确性。给出了3种基于可信链模型的大粒度服务可信组合方法，随后通过对比实验验证所提方法的有效性。

关键词：Web服务组合；扩展的标签迁移系统；可信传递；最弱前置条件

中图分类号：TP311.5 **文献标识码**：A **文章编号**：1000-436X(2011)9A-0077-10

New approach to Web service composition using trust chain model

GAO Hong-hao[1], LI Ying[2], ZHANG Yuan-yuan[3]

(1.School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China;
2.College of Computer Science, Zhejiang University, Hangzhou 315100, China;
3.College of Information Technology, Zhejiang Chinese Medical University, Hangzhou 310053, China)

Abstract: The reliability problem was addressed in the context of verifying Web service composition. It adopted theorem proving to improve model checking framework and proposed a Web service composition verification approach based on trust chain model. An extended labeled transition system (ELTS) model was used to describe the composite behaviors and interaction protocols of Web service composition, where each service state and transition were labeled by a reliability set and constraints respectively so that the process of verifying Web service composition was converted into the reliability set calculation. The predicate transformation function WP(Q,R) was introduced to compute the weakest precondition, by which the correctness of Web service composition could be verified by checking the implication relation between the reliability set and the weakest precondition. Moreover, it also gave three methods for the complex Web service composition using trust chain model. Finally, compared with other methods, our method was more suitable for verifying Web service composition.

Key words: Web service composition; ELTS; trusted transition; weakest precondition

收稿日期：2011-07-05

基金项目：国家高技术研究发展计划（“863”计划）基金资助项目(2009AA110302)；国家自然科学基金资助项目（60873045）；浙江省重大软件专项基金资助项目（2009C17002）

Foundation Items: The National High Technology Research and Development Program of China (863 Program) (2009AA110302); The National Natural Science Foundation of China (60873045); The Science and Technology Program of Zhejiang Province (2009C17002)

1 引言

在分布式计算、基于构件开发、面向服务体系结构等技术支撑下，互联网集成了丰富的数据、计算、软件和服务资源，已成为可共享的网络化信息处理基础设施平台。越来越多企业和商业组织参与到软件服务化(SaaS, software as a service)的行列中，通过Web服务和服务组合求解问题和开展业务逐渐成为主流趋势。松耦合特性决定了组合服务来自不同的组织或者机构，它们彼此交互协作实现复杂的计算需求和业务逻辑。由于服务输入/输出接口消息接收和发送存在先后时序关系，服务动态演化容易引起接口参数变更，服务组合存在死锁、活锁、未定义接收和服务不可达等问题[1]，必须对服务组合的正确性进行检验，从而保证组合服务正确执行并实现用户既定功能性目标，同时又满足用户预期的非功能性需求。为了确保服务组合是可信的，正确性检验问题不仅局限于组合服务接口参数兼容性和服务可用性的检验，还涉及服务间交互行为的时序逻辑检验[2]。研究可信服务组合方法已成为当今工业界和学术界共同关注的热点。

为了发现服务组合中存在的缺陷，国内外学者提出了多种服务组合检验方法，主要分为基于模型检验和基于定理证明2大类。基于模型检验的Web服务组合检验方法是通过遍历服务组合模型的有穷状态空间检验组合行为是否满足预期的时态逻辑性质。例如，Fu[3]等人给出一套技术和工具(Guarded Automata、Promela语言、SPIN模型检测器)以及一个完整的系统分析和验证组合服务正确性的方法。Bultan等[4]给出了一个对服务组合行为进行规格说明和验证的形式化框架，用Mealy机为Web服务行为建模，通过自动机的组合和包含关系检验服务组合行为。Nakajima[5]采用模型检验工具SPIN检验基于WSFL描述的Web服务工作流。Fahland和Reisig[6]采用简单的抽象状态自动机(ASM)模型对BPEL建模和验证。Wombacher[7]采用有限状态机对服务行为进行建模，验证2个服务之间协作的正确性。Foster[8]采用有限状态机对基于BPEL描述的服务组合流程进行建模，通过检验流程安全性和活性等性质验证服务正确性。Benatallah[9]在服务内部业务流程中加入了时间约束信息，采用自动机对其业务协议进行建模，进而验证2个带时间约束的业务协议在进行交互时的正确性、兼容性和替换性。基于定理证明的Web服务组合检验方法则采用逻辑公式描述组合系统规格说明，应用公理和推理规则证明组合系统应具有的性质。例如，Rao等[10]将服务的自动组合问题视为基于逻辑的程序合成问题，采用线性逻辑的定理证明方法来实现Web服务自动组合，提出了一个基于多Agent的服务自动组合框架。Waldinger[11]同样从程序自动合成角度出发，采用定理证明方法实现服务的自动组合，它引入一阶逻辑形式化表达服务与用户的请求，并采用定理证明器构造满足用户请求的服务组合流程。殷等[12]引入Martin-Lf类型论，提出了Service-Behavioral Types对服务动态行为进行建模，同时结合扩展后的相关理论和类型静态推理及检测方法，给出了服务兼容性与可替换性的判定方法，并在工具Coq上实现自动化定理证明。然而，基于模型检验和基于定理证明的方法都存在不足：模型检验无法解决状态爆炸问题，当组合服务存在并发时，状态空间可能随着并发分量的增加呈指数增长；而定理证明则需要过多的人工引导，增加了验证的时间复杂度和空间复杂度。

鉴于此，本文借鉴文献[4,7,8]模型检验方法，采用自动机对服务组合行为进行建模，参考文献[9]思想，在组合业务流程中引入可信集和迁移约束，提出一种扩展的标签迁移系统(ELTS, extended labeled transition system)模型描述服务组合行为。检验服务组合时，基于On-the-fly方法[13]逐步展开模型，获取服务组合链用于检验正确性。借鉴文献[11,12]定理证明方法，应用最弱谓词变换函数WP(Q,R)[14]推导服务组合的最弱前置条件，通过检验可信集和最弱前置条件的蕴含关系，证明服务组合的正确性。因此，本文方法是定理证明和模型检验相结合的一种混合验证技术，展开模型的同时检验迁移约束，能有效的避免状态爆炸问题。

本文各节安排如下：第2节给出服务组合的可信链模型；第3节使用WP(Q,R)方法推导最弱前置条件，检验服务组合的正确性。第4节结合一个实例，通过对比实验证明本文方法的有效性；第5节总结全文并展望一步工作。

2 Web服务组合可信链模型

基于自动机理论的服务组合可直观地描述参与组合的服务以及服务间的交互行为和交互协议，

但在实际应用中需要对其扩展。如图 1 所示的服务组合行为中，假设状态集 Authorized={s_0,s_1}是处于安全临界区(safety critical region)的服务，它们的访问须经授权；而状态集 Unauthorized={L_0,L_1}的服务访问不需授权。显然迁移 b_3 对系统的安全存在潜在威胁，因为无论从哪个授权状态 s Authorized 出发，系统最终将会进入未授权状态 u Unauthorized。因此，必须对迁移 b_3 添加卫式条件(guard condition)，如时间约束、概率行为、代价花销等，用以识别和控制 b_3 这条不安全的迁移。

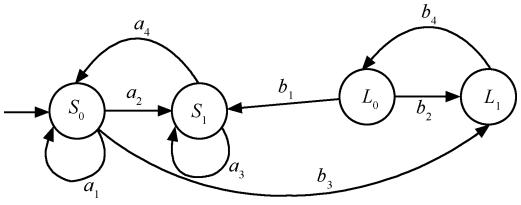


图 1　服务行为模型

本文将服务组合行为转化为 ELTS 模型，为每个服务状态和迁移行为分别标识可信集和迁移约束，提出了一种基于可信链模型的 Web 服务组合方法。

定义 1(可信链模型)。采用可信链模型描述的服务组合是一个 ELTS 模型，其定义为 6 元组，即 A::=(W ,C ,L, , Init,Acc)，其中：

1) W 是由 Web 服务实体(原子/复合服务)组合的有限集合。集合 W 中的元素称为状态。

2) C 是约束变量的有限集合，包括空集 。变量上的约束条件 $F(C)$ 定义如下：

$$ ::= \text{true} \mid x \leqslant z \mid x < z \mid z < x \mid z \leqslant x \mid \quad$$

其中，x C 是一个约束变量，z Q 是有理数集合的一个常量。

3) :W $F(C)$ W 是状态迁移函数。约束 $F(C)$ 又称契约(contract)，表明 Web 服务 w_i W 与另一服务 w_{i+1} W，在满足卫式条件 $c_{w_{i+1}}$ $F(C)$ 的情况下交互，即 $(w_i, c_{w_{i+1}}) = w_{i+1}$。特殊地，$(w_i,) = w_{i+1}$ 表示组合交互不受约束。

4) Init W 为初始状态，其存储的可信集称为可信根。

5) ACC W 为终止状态，其输出的结果称为可信度。

6) 标签函数 L:W 2^{AP} 表示服务计算能力，即服务执行语句。约束变量在服务执行后被赋予新值，该操作称为约束变量更新动作 Update，它的逻辑表达式定义如下：

$$\text{Update} ::= x = x \quad y + z \mid x = x^n \mid x = z \mid \text{Update} \mid \quad \text{Update} \mid \text{Update} \quad \text{Update}$$

其中，x，y C 是约束变量，　{+， ，*，/，% ，…}是逻辑运算符，表达式 Update := Update 和 Update := Update 分别表示约束变量值不发生变化和取反。

ELTS 行为模型刻画行为的 3 类基本元素是：有限数量的状态、状态之间的转移及每个状态转移的约束条件。令 T $2^{F(C)}$ 为可信约束集，组合链 Path(A)表示由路径组成的集合。ELTS 模型是一个确定型有限自动机，每条组合链上的状态转移必须满足约束条件才能触发。可信约束集 $T=\{T_{path(1)}, T_{path(2)}, \cdots, T_{path(n)}\}$，其中 $T_{path(i)}$ 为第 i 条组合链 $path_{(i)}=<Init, w_1, w_2, \cdots, w_i \cdots, ACC>$ 可信约束集。

定义 2(可信约束序列)。为每个服务 w_i 赋予可信约束序列 $T_{wi}=<c_{wi+1}, \cdots, c_{ACC}>$，表示至少存在一条执行链，使得组合模型上服务间能正常交互并停机，同时服务执行过程中限制条件(从 c_{wi+1} 至 c_{ACC})依次得到满足。存在如下特性：

1) 可收敛性：当且仅当$i,j N, i<j \mid T_{wi} \triangleright |T_{wj}|$；

2) 可联接性：当且仅当$c F(C), $j N, $(w_j, c) = w_i T_{wj} = c T_{wi}$，符号 表现序列的联接，联接结果 $T_{wj}=<c, c_{wi+1}, \cdots, c_{ACC}>$；

3) 可传递性：满足可收敛性和可联接性的服务组合，用 $(w_i, T_{wi}) \vdash_{TCA} (w_j, T_{wj})$ 表示。信任传递满足偏序关系 {<,≤}，即 $T_{wi} \prec T_{wj}$。

定义 3(多步信任函数)。定义多步信任运行函数 Run：W $T_w \rightarrow W$，满足如下约定：

$$Run(w_i, T_{wi}) = Run(\ (w_j, c), T_{wj});$$

$$Run(w_i, \quad) = w_i。$$

多步信任传递用 $\vdash^*_{TCA}$ 表示，如 $(Init, T_{Init}) \vdash^*_{TCA} (ACC, T_{ACC})$ 表示从初始状态到结束状态，即存在一条满足可信约束的组合链。信任传递序列描述为 $<w_{Init}, c_{w1}, \cdots, w_i, c_{wi}, w_{i+1}, \cdots c_n, w_{ACC}>$，用 $\multimap$ 表示服务状态间彼此信任关系，可信的组合链描述为 $w_{Init} \multimap w_1 \multimap \cdots \multimap w_i \multimap w_{i+1} \multimap w_{ACC}$。否则，存在 $w_i \not\multimap w_{i+1}$ 表明组合链是不可信的。

定理 1(连续性)。信任传递满足状态迁移的连续性。若约束条件 c 使得状态迁移满足 $(w_i, c) = w_j$，则存在 T_{wi} 满足信任传递 $(w_i, T_{wi}) \vdash_{TCA} (w_j, T_{wj})$。

证明　已知约束条件 c 使得状态迁移满足 $(w_i, c) = w_j$，由定义 3 知存在该组合链满足 $Run(w_i, T_{wi}) =$

Run($\oint(w_j,c)$,T_{wj})，因此得到信任传递的偏序关系$T_{wi}\prec T_{wj}$，根据由定义2的可传递性知$(w_i,T_{wi})\vdash_{TCA}(w_j,T_{wj})$满足。

定理2(信任干扰)。组合模型是不可靠的，当且仅当，存在信任干扰$(w_i,T_{wi})\nvdash_{TCA}(w_j,T_{wj})$。状态迁移满足$\oint(w_i,c)=w_j$，但$c\notin T_{wi}$，因此信任传递是不可信的，$w_i$与$w_j$交互行为存在干扰。

证明 由状态迁移满足$\oint(w_i,c)=w_j$及定义2的可联接性知，可信集$T_{wi}=c\cup T_{wj}$。但已知条件$c\notin T_{wi}$表明实际运行过程中服务w_i与w_j交互行为必然存在干扰。

定理3 (信任不对称性)。信任不对称性是指在信任传递过程不满足以下约束：

$(w_i,T_{wi})\vdash_{TCA}(w_j,T_{wj})\Rightarrow(w_j,T_{wj})\vdash_{TCA}(w_i,T_{wi})$；

$(w_j,T_{wj})\vdash_{TCA}(w_i,T_{wi})\Rightarrow(w_i,T_{wi})\vdash_{TCA}(w_j,T_{wj})$。

证明 (反正法)假如$(w_i,T_{wi})\vdash_{TCA}(w_j,T_{wj})$，$(w_j,T_{wj})\vdash_{TCA}(w_i,T_{wi})$，那么$\exists c_{wj}\in F(C)$，使得$\oint(w_i,c_{wj})=w_j$并且$T_{wi}=c_{wj}\cup T_{wj}$。同理，$\exists c_{wi}\in F(C)$，使得$\oint(w_j,c_{wi})=w_i$并且$T_j=c_{wi}\cup T_i$。由可信条件集合$T_{wi}$知，当$(w_i,T_{wi})\vdash^*_{TCA}(w_j,T_{wj})$时，$T_{wj}\prec T_{wi}$，同理当$(w_j,T_j)\vdash^*_{TCA}(w_i,T_{wi})$时，$T_{wi}\prec T_{wj}$。要使假设成立，必然满足$T_{wi}=T_{wj}$，此时，模型处于死锁。因此，信任不具备对称性。

定义4 (后继/前驱关系)。给定可信链模型A，状态$s\in W$和约束条件$c\in F(C)$，后继/前驱关系定义如下：

直接后继定义为：$Post(s,c)=\{s'\mid\exists s\in W(s,c,s')\in\oint\}$，所有后继描述为：$Post(s)=\bigcup_{\forall c\in F(C)}Post(s,c)$。

直接前驱定义为：$Pre(s,c)=\{s'\mid\exists s\in W(s',c,s)\in\oint\}$，所有前驱描述为：$Pre(s)=\bigcup_{\forall c\in F(C)}Pre(s,c)$。

定义5 (可信集运算)。组合服务模型上存在多种分支结构，基于定义4本文给出4种基本结构的可信集运算方法。

1) 交运算。当状态w_i直接后继存在并发时，进行交运算。假设$Post(s)=\{w_1,w_2\}$，T_s表示经过运算后的可信集，即$T_s=T_{w1}\cap T_{w2}=\{t\mid\forall t,t\in T_{w1}\wedge t\in T_{w2}\}$。

2) 并运算。同理在选择流程的服务组合中，进行并运算：$T_s=T_{w1}\cup T_{w2}=\{t\mid\forall t,t\in T_{w1}\vee t\in T_{w2}\}$。

3) 差运算。互斥执行是一种特殊情况，进行差运算。当w_1所属的组合链条被互斥执行时，满足$T_s=T_{w1}\backslash T_{w2}=\{t\mid\forall c_1\in T_{w1},\exists c_2\in T_{w2}t\neq c_1\backslash c_2\wedge t\notin\}$。

4) 笛卡尔积运算。当$T_{w1}\cap T_{w2}=\varnothing$时，进行笛卡尔积运算：$T_s=T_{w1}\times T_{w2}=\{t\mid\exists t,\forall c_1\in T_{w1},\forall c_2\in T_{w2}t=\{c_1,c_2\}\}$。

3 服务组合正确性检验

最早提出最弱谓词变换函数WP(Q,R)[14]的目的是用于程序的正确性检验。Kirchner等[15]设计和实现了一个基于验证条件(verification conditions)证明定理库，允许程序以带注释的方式标注前置和后置条件并返回验证条件集合。Marcello等[16]对WP(Q,R)进行了扩展，包括非限制恶意选择和回溯操作，并且提出3种序列关系：向后求精序列、死锁序列以及Nelson近似序列。此外，Banerjee等[17]采用WP(Q,R)验证著名的哲学家问题，使得复杂性并不随着问题空间的增大而增大。与这些研究不同的是，本文将最弱谓词变换函数应用于服务组合检验。

3.1 基于谓词变换函数的组合

定义6 (最弱前置谓词变换函数)。最弱前置谓词变换函数定义为$WP:Q\times Pred\rightarrow Pred$，其中参数$Q\in 2^{AP}$是服务执行语句，$R\in Pred$是谓词表示服务执行的后置条件，函数计算结果为另一谓词$P\in Pred$表示服务执行的最弱前置条件。如服务w参与组合，其执行Q到满足谓词T_{wi}状态时一定会终止并且终止状态满足谓词R，由蕴含关系$P\Rightarrow T_{wi}$表明组合是正确的。

定义7 (最弱前置谓词变换函数语义)。最弱前置谓词变换函数存在:断言操作、假设操作、空转操作(skip)、赋值操作等原子操作，最弱前置谓词变换函数运算的关键是谓词集合变换。语法格式定义如下：

$Q::=v:=t\mid b\rightarrow\mid div\mid Q_1;Q_2\mid Q_1|Q_2\mid Q_1\square Q_2\mid Skip\mid Halt$

1) $WP(v:=t,R)=R[t/v]$，表明在赋值语句$v:=t$作用下，谓词变换函数将后置条件R中v变量用t值进行替换。如$WP([[a:=10]],a=10)\equiv(10=10)\equiv true$，$WP([[a:=10]],a=9)\equiv(10=9)\equiv false$，$WP([[a:=2b+c]],a>15)\equiv(2b+c>15)$。

2) $WP(Q,false)\equiv false$，表明当前状态为false，前置条件也为false。

3) $WP(b\rightarrow,R)\equiv b\Rightarrow R$，$b\rightarrow$是假设语句。当$b$成立时，计算在后置条件为$R$的状态下终止。

4) $WP(div,R)\equiv false$，表示任意状态都是不可

达的，即不能在后置条件为 R 的状态下终止。

5) WP(Skip,R)　R，Skip 是一个跳转语句，语句不执行任何操作。

6) WP([[Q1;Q2]],R)　WP(Q1,WP(Q2,R))，[[Q1;Q2]]是顺序执行。

7) WP([[Q1|Q2]],R)　WP(Q1,R)∧WP(Q2,R)，[[Q1|Q2]]是并发执行。

8) WP([[Q1□Q2]],R)　WP(Q1,R)∧(WP(Q1,false)⇒WP(Q2,R))，[[Q1□Q2]]是二元选择执行。

定义 8　(后置条件组合)。给出如下 3 种后置条件操作模式：

1) 与分布律：服务 Q 含有 n 个后置条件 $R_1,R_2,\cdots,R_n$，与分布律标记为 $\overset{n}{\underset{i=1}{}}R=R_1\wedge R_2\cdots\wedge R_n$，那么 $WP(Q,R_1)\wedge WP(Q,R_2)\wedge\cdots\wedge WP(Q,R_n)=WP(Q,\overset{n}{\underset{i=1}{}}R=R_1\wedge R_2\cdots\wedge R_n)$；

2) 或分布律：或分布律标记为 $\overset{n}{\underset{i=1}{}}R=R_1\vee R_2\cdots\vee R_n$，那么 $WP(Q,R_1)\vee WP(Q,R_2)\vee\cdots\vee WP(Q,R_n)=WP(Q,\overset{n}{\underset{i=1}{}}R=R_1\vee R_2\cdots\vee R_n)$；

3) 差分布律：假如 R_k 与 $R_1,R_2,\cdots,R_{k-1},R_{k+1},\cdots,R_n$ 是互斥的，差分布律标记为 $R_k\backslash\overset{n}{\underset{i=1\ i\neq k}{}}R_i$。那么 $WP(Q,R_k)\backslash WP(Q,R_1)\vee WP(Q,R_2)\vee\cdots\vee WP(Q,R_n)=WP(Q,R_k\backslash\overset{n}{\underset{i=1\ i\neq k}{}}R_i)$。

Web 软件可抽象地看作是一个大粒度的复合服务。大粒度服务的不断出现和广泛使用，为面向服务的软件构造、维护及动态配置等技术的关键环节制造了一些困难：多个 ELTS 模型可组成更为复杂的服务模型。为此，以下给出 3 种基于可信链模型的复合服务组合方法。

定义 9　(顺序组合)。给定 2 个 Web 服务可信链模型 $A_1=(W_1,C_1,L_1,\oint_1,Init_1,Acc_1)$和 $A_2=(W_2,C_2,L_2,\oint_2,Init_2,Acc_2)$，顺序组合模型为 $A_s=(W_s,C_s,L_s,\oint_s,Init_s,Acc_s)$，当且仅当$\exists i\cdot w_i\in Acc_1$，$\exists j\cdot w_j\in Init_2$，使得 w_i Sequence w_j成立。令 Q_1 为 A_1 模型的执行语句，Q_2 为 A_2 模型的执行语句，由 WP([[Q1;Q2]],R)定理知，给定一个后置条件 R 和前置条件 P，顺序组合的约束条件需满足 Sequence=WP([[Q2]],R)，P=WP([[Q1]],Sequence)。

顺序组合状态 $W_s=W_1\times W_2$，约束变量 $C_s=C_1\cup C_2$，约束条件$F(C)=F(C_1)\wedge F(C_2)\wedge\{Sequence\}$，状态迁移函数 $\oint_s=\oint_1\cup\oint_2\cup(w_i,Sequence,w_j)$，初始状态 $Init_s=Init_1$，结束状态 $Acc_s=\{(Acc_1\cup Acc_2)\backslash w_i\}$。

定义 10　(选择组合)。选择组合模型 $A_s=(W_s,C_s,L_s,\oint_s,Init_s,Acc_s)$中存在 t_0 初始状态，在选择连接 Choose 的作用下使得状态迁移满足$(t_0$ Choose $Init_1)$，$(t_0$ Choose $Init_2)$。由 WP([[Q1□Q2]],R)定理知，给定一个后置条件 R 和前置条件 P，选择组合的约束条件需满足 Choose=WP(Q1,R)∧(WP(Q1,false)⇒WP(Q2,R))。

选择组合状态 $W_s=W_1\times W_2\cup\{t_0\}$，约束变量 $C_s=C_1\cup C_2$，约束条件$F(C)=F(C_1)\wedge F(C_2)\wedge\{Choose\}$，状态迁移函数 $\oint_s=\oint_1\cup\oint_2\cup\{(t_0,Choose,Init_1),(t_0,Choose,Init_2)\}$，初始状态 $Init_s=t_0$，结束状态 $Acc_s=Acc_1\cup Acc_2$。

定义 11　(并发组合)。对于并发组合模型 $A_s=(W_s,C_s,L_s,\oint_s,Init_s,Acc_s)$，存在 $q_p=(t_0,t_0)$初始状态，其在并发连接 Parallel 作用下使得状态迁移满足(t_0,t_0) Parallel (q_0,s_0)。由 WP([[Q1|Q2]],R)定理知，给定一个后置条件 R 和前置条件 P，并发组合的约束条件满足 Parallel =WP([[Q2]],R)∧WP([[Q1]],R)。

并发组合状态 $W_s=W_1\times W_2\cup\{(t_0,t_0)\}$，约束变量 $C_s=C_1\cup C_2$，约束条件$F(C)=(F(C_1)\wedge F(C_2))\wedge\{Choose\}$，结束状态 $Acc_s=Acc_1\times Acc_2$。状态迁移集合 $\oint_s((w^1_i,w^2_j),c)=(\delta_1(w^1_i,c),\delta_2(w^2_j,c))$ 是同步迁移，满足：$c\in(F(C_1)\wedge F(C_2))$，$\oint_1(w^1_i,c)$ 且 $\oint_2(w^2_j,c)$。

3.2　冗余约减

计算 WP(Q,R)时，研究冗余约减是非常必要的。以 $WP([[Q_1|Q_2\|\cdots Q_n]],R)=WP(Q_1,R)\wedge WP(Q_2,R)\wedge\cdots WP(Q_n,R)$为例，显然 R 是一个重复的操作，更重要的是随着语句空间的增长将会导致状态爆炸问题。为避免参数后置条件 R 重复出现，消去语句赋值方法将除第一参数以外的参数用一个与 R 无关的参数进行替换，使得计算目标 R 只出现一次。Rustan[18]给出了自由谓词变换函数(WLP, weakest liberal preconditions)(Q,R)，允许计算进入失效状态或错误状态。WLP(Q,R)与定义 7 WP(Q,R)不同的是：WLP(div,R)=true 和 WLP([[Q1□Q2]],R)=WP(Q1,R)∧(WLP(Q1,false)⇒WLP(Q2,R))。

定理 4：$\forall R\cdot WLP(Q,R)=WLP(Q,false)\vee R$

证明　WLP(Q,R)∧　R

WLP(Q,R)∧WLP(Q,　R) Introduction-∧

WLP(Q,R∧ R) 与分布律

WLP(Q,false)

因此WLP(Q,R)∧ R=WLP(Q,false) WLP(Q,R)=WLP(Q,false)∨R

定理5 "R·R⇒WLP(Q,R)。

证明 R Introduction-∨

WLP(Q,false)∨R Substitute

WLP(Q,R)

定理6 "R·WP(Q,R)=WP(Q,true)∧WLP(Q,R)。

证明 "R·WP(Q,R)

WP(Q,true)∧R

WP(Q,true)∧WLP(Q,R) 定理5

根据定义4和定理6得式"R·WP(Q,R)=WP(Q,true)∧(WLP(Q,false)∨R)，从而可消去R的重复计算项。以$WP(Q_1|Q_2\cdots Q_n,R)$为例，$WP(Q_1|Q_2||\cdots Q_n,R)=WP(Q_1,true)\wedge WP(Q_2,true)\cdots WP(Q_n,true)\wedge((WLP(Q_1,false)\wedge WLP(Q_2,false)\cdots WLP(Q_n,false))\vee R)$。经过变换后的公式将冗余计算问题化简为SAT可满足性求解问题，现有多数SAT Solver[19]求解器都支持自动化求解可满足性。

3.3 组合检验

服务组合检验算法分为2个步骤：第1步，抽取服务组合ELTS模型的组合链Path(A)，它可通过On-the-fly的方法逐步展开服务组合模型获得。第2步，在展开模型同时，应用谓词变换函数WP(Q,R)推导服务组合的最弱前置条件，检验可信集和最弱前置条件的蕴含关系。一旦发现组合错误立即停止检验，返回用户错误信息。具体算法如下：

步骤1 以Acc输出对象为起点，以Init对象为终点，采用前向链(forward-chaining)搜索策略和On-the-fly方法逐步展开服务组合模型，获取组合模型的路径序列构造组合链Path(A)，利用定义5处理分支结构的可信集运算。

步骤2 组合链由一系列服务状态序列组成，其上服务实体w_i执行语句$Q_{wi}=L(w_i)$采用文献[10]方法获取，即由语义服务OWL-S的Service Profile描述转化而来。

步骤3 给定后置条件R计算最弱前置条件P，判断前驱服务w_{i1}可信集T_{wi1}与P之间的蕴含关系，即检查$P\Rightarrow T_{wi1}$是否成立。

1) 若成立，将w_i前置条件P赋给其前驱服务w作为其后置条件R，迭代这一过程直到Init状态；

2) 否则，说明组合存在错误，停止检验并返回错误信息。

4 实验

4.1 实验环境

对比实验是在IBM X260服务器上运行的，其软硬件配置为：1.86G Xeon CPU；1G RAM；RedHat操作系统。首先，分析模型检验工具NuSMV[注1]、定理证明工具PVS[注2]和本文方法WP-LTS在服务组合检验上的优缺点。其次，从服务的可靠性和用户响应时间角度证明本文方法的有效性。

4.2 实验对象

如图2所示，在线购物服务OnLineShoppingSit由3个业务流程用户积分PLevel、消费限额PConLimit、个人信息PInfor组成，涉及9个服务W={Login,Order,Address,Account,AddLevel,Payment,Sprint,Bprint,Uprint}。用户Login登入系统后，通过服务Order购买商品。之后，用户可选择查看个人信息以便核实收货地址，或在线下订单。当需要确认个人信息时，服务Address返回个人信息，通过打印服务Uprint显示。当选择下订单时，服务Account检查当前账户是否有效，如果有效，并发执行以下2个流程：通知商家用户已经下订单，并请求增加客户积分AddLevel，通知银行付钱Payment给商家。

为了便于说明，将Login,Order,Address,Account,AddLevel,Payment,Sprint,Bprint,Uprint分别对应A,B,C,D,E,F,G,H,I字母。解析OWL-S描述的语义并构造约束关系$F(C)=\{c_a,c_b,c_c,c_d,c_e,c_f,c_g,c_h\}$。服务可信集如图2所示，该模型上的3条组合链需满足可信迁移：①A⊸B⊸D⊸F⊸E；②A⊸B⊸D⊸E⊸H；③A⊸B⊸C⊸I。

为了说明组合检验过程，图3给出了一个示例。图3(a)所示，检验点处于E点Payment位置；图3(b)所示，检验点处于D点Account位置。该过程需检验$\oint(Account,WP([[Q_{Payment}]],r))\rightarrow Payment$和$WP([[Q_{Payment}]],r)\Rightarrow T_{Account}$是否满足。当可信集和最弱前置条件不满足蕴涵时，返回组合错误信息。

注1 http://pvs.csl.sri.com/

注2 http://nusmv.fbk.eu/

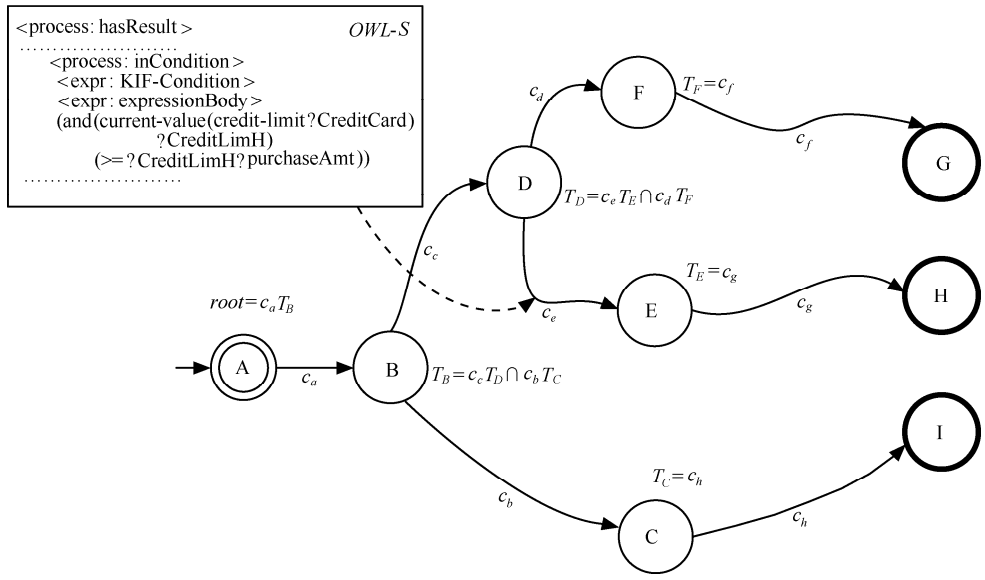
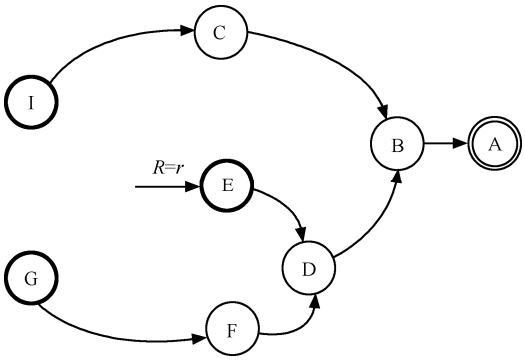
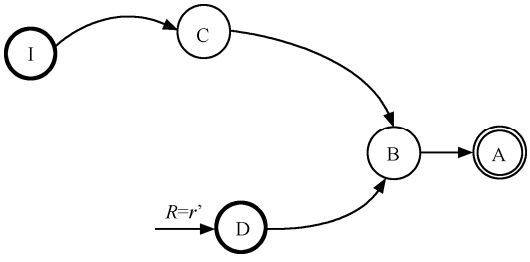


图 2　在线购物服务模型



(a)检验点处于 E 点 Paym ent 位置



(b)检验点处于 E 点 A ccount 位置

图 3　验证过程示例

4.3　实验方法

为了比较错误检验能力，为在线购物服务模型植入 2 类错误。第 1 类错误为参数格式问题，如用户订单号应该本应为 year-m onth-date-tim e 构成，但实际传输过程总被描述为 year-m onth date-tim e。第 2 类错误为数字边界溢出问题，如密码位数超过限定位数。实验考察大粒度的复合服务并发组合，不断增加服务组合的复杂度，在状态增加过程中，观察 3 种方法错误发现能力的同时，分析时间消耗、吞吐率、错误发现能力 3 个维度上的性能。

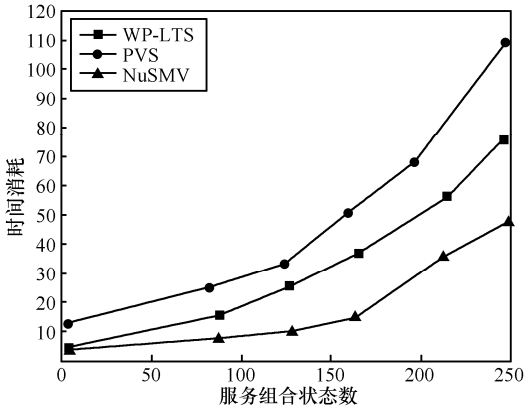
4.4　对比实验

实验结果（如图 4 所示）表明如下。

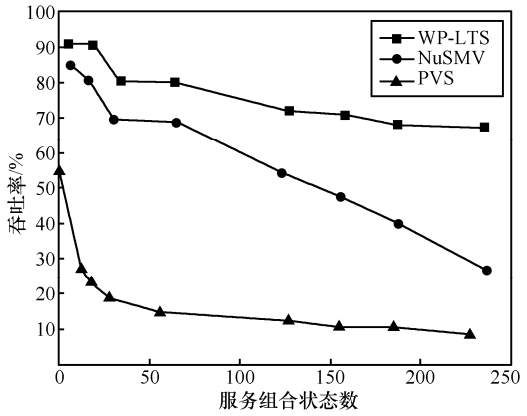
1)工具 N uSM V 、PV S 和本文 W P-LTS 方法都能发现服务组合存在的错误。但本文方法比其他三者的能力都要强，因为 N uSM V 遍历模型时，容易出现状态爆炸问题，而且隐藏的错误只要不违反给定性质，是不会被探测到的。PV S 虽然能完全发现错误，但需人工干预证明过程，查错能力与给定定理和策略有关。本文方法克服了这两者的缺点，可更好地发现错误。

2) W P-LTS 吞吐率最大，其他依次是 N uSM V 和 PV S。当状态数达到 50 时，3 种方法的吞吐率均开始下滑，但是 W P-LTS 仍大于其他 2 种方法，表明 W P-LTS 执行效率比较高。因为通过计算 W P(Q , R)可快速发现迁移存在的错误，这一计算量相对 N uSM V 模型空间的遍历和 PV S 全局证明计算量要小的多。

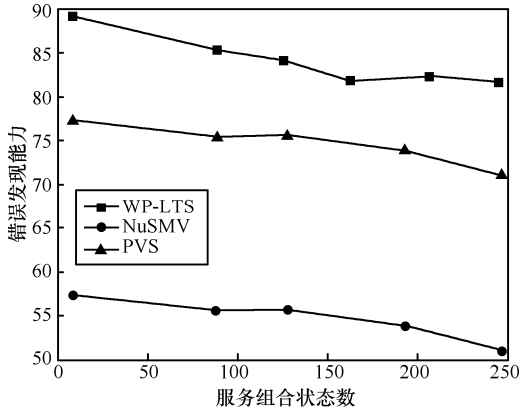
3)在时间消耗上，PV S 最大，N uSM V 最小，W P-LTS 介于两者之间。时间随着组合状态数量的增加而成线性增长。PV S 主要消耗在构造全局定理证明过程，N uSM V 则消耗在穷尽地遍历模型，而本文采用 O n-the-fly 方法中间展开模型，将定理证明缩小到迁移执行的约束检验的局部空间范围内。由于失败返回远快于一个成功的组合，所以，当模型状态增加时，W P-LTS 在发现错误时间上优势明显。



(a) 时间消耗实验结果



(b) 吞吐率实验结果



(c) 错误发现能力实验结果

图4 实验结果对比

4.5 实验方法

采用爬虫技术在SeekDa!服务网站[注3]上抓取了1 654个服务，并转化为OWL-S描述，利用一系列随机产生的流程实例进行了大量实验。为了验证本文方法对服务组合成功率的影响，实验对比分析传统基于工作流的服务组合方法和本文WP-LTS方法，采用Apache JMeter工具[注4]分析用户响应时间上的代价和运行时的可靠性。

4.6 对比实验2

如图5所示，采用本文WP-LTS方法的时间开销保持稳定，没有随着服务数的增多而大规模增加，平均在250ms内实现服务流程组合。而传统的工作流方法的时间开销随服务数的增大而显著提高，尤其是当服务大于1000时，平均增长幅度为66%。这是由于谓词变换函数WP(Q,R)推导服务组合的最弱前置条件，和检验可信集和最弱前置条件的蕴含关系是WP-LTS方法最为关键的步骤，经过严格证明的服务流程是可信的，同时也是最耗时的一步。而传统的工作流方法虽然可以直接定义一个服务流程，但缺少验证支持，在遇到异常(服务端口不兼容，服务执行约束不满足等)情况时，需进行动态配置，它通常需在服务库中多次查找可替换的服务(服务集)，因此时间花费代价较高。

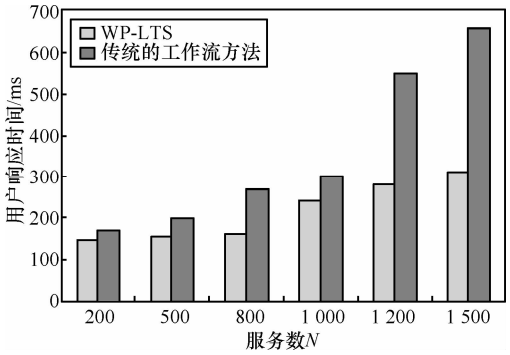


图5 在不同服务规模下服务组合的用户响应

图6是Web服务组合可靠性分析实验。给定服务组合需求任务数Task，统计服务失效情况，分析组合服务的可靠性。实验表明采用本文WP-LTS方法的服务组合运行比较稳定。当任务数为10时，采用WP-LTS方法的服务组合可靠性为0.86，采用传统工作流方法的服务组合则可靠性为0.65。当组合需求任务数增加时，组合服务的可靠性下降，但WP-LTS方法可靠性变化相对比较平缓。传统的工作流方法可能存在未经过验证的组合服务交互行为(如参数类型Object和Object的组合交互)，虽然它们在组合阶段可采用强制类型转化等方法实现，

注3 http://webservices.seekda.com/

注4 http://jakarta.apache.org/jmeter/

但在实际运行中服务依赖的输入输出是影响服务正常运行的主要因素，因此易暴露组合缺陷，导致组合可靠性降低。

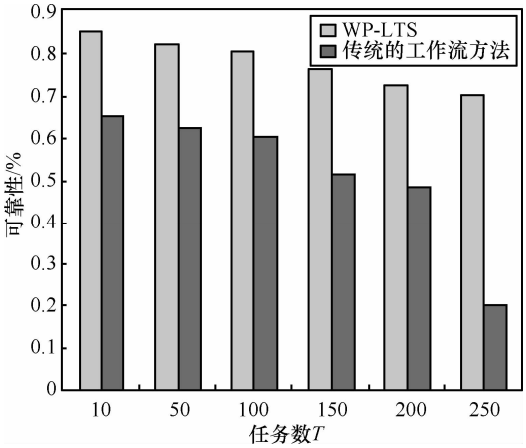


图 6　服务组合可靠性分析

5　结束语

本文首先介绍了服务组合检验的 2 种形式化方法，然后提出了一种基于可信链模型的 Web 服务组合方法。为服务状态和迁移行为分别标识可信集和迁移约束，扩展自动机模型用于形式化描述服务组合行为。引入谓词变换函数 WP(Q,R)检验服务组合的正确性。提出了 3 种基于可信链模型的大粒度服务可信组合方法。给出了服务组合正确性检验算法，并经过实验证明该检验方法的有效性。本文方法是定理证明和模型检验相结合的混合验证技术，降低了验证计算的复杂度，提高了验证效率。在后续的工作中将会设计和实现一个基于高阶逻辑 Coq[20]的 WP(Q,R)集成工具。

参考文献：

[1] 雷丽晖,段振华.一种基于扩展有限自动机验证组合 Web 服务的方法[J].软件学报,2007,18(2):2980-2990.
LEIL H,DUN Z H.An extended deterministic finite automata based method for the verification of composite Web services [J].Journal of Software,2007,18(2):2980-2990.

[2] 邓水光, 李莹, 吴健等.Web 服务行为兼容性的判定与计算[J].软件学报,2007,18(12):3001-3014.
DENG S G,LI Y,WU J,et al.Determination and computation of behavioral compatibility for Web services [J].Journal of Software,2007,18(12):3001-3014.

[3] FU X,BULTAN T,SU J W.Analysis of interacting BPEL Web services[A].Proc of the13th Int l Conf.on World Wide Web[C].New York:ACM Press,2004.621-630.

[4] BULTAN T,FU X,HULL R,SU J W.Conversation specification:a new approach to design and analysis of e-Service composition[A].Proc of the 12th International Conference on World Wide Web[C].New York:ACM Press,2003.403-410.

[5] NAKAJIMA S.Verification of Web service flows with model-checking techniques[A].In:Proc.of the First International Symposium on Cyber Worlds[C].Washington,DC:IEEE Press,2002.378-3854.

[6] FAHLAND D,REISIG W.ASM-based semantics for BPEL:the negative control flow[A].Proc of 12th International Workshop on Abstract State Machines,Paris:IEEE Press,2005.131-151.

[7] WOMBACHER A,FANKHAUSER P,MAHLEKO B,et al.Matchmaking for business processes based on choreographies[A].Proc.of the 2004 IEEE International Conference on e-Technology,e-Commerce and e-Service[C].Washington,DC:IEEE Press,2004.359-368.

[8] FOSTER H,UCHITEL S,MAGEE J,et al.Compatibility verification for Web service choreography[A].Proc of International Conference on Web Services[C].Washington,DC:IEEE Press,2004.738-741.

[9] BENATALLAH B,CASATI F.Fine-grained compatibility and replaceability analysis for timed Web service Protocols[A].Proc of the 26th international conference on Conceptual modeling[C] LNCS:Springer-Verlag Press,2007,4801:599-614.

[10] RAO J.Semantic Web Service Composition via Logic-based Program Synthesis[D].Norwegian University of Science and Technology,2004.

[11] WALDINGER R.Web agents cooperating deductively[A].Proc.of the First International Workshop on Formal Approaches to Agent-Based Systems[C].LNCS:Springer-Verlag Press,200,1871:250-262.

[12] 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):1-6.
YIN Y Y,LI Y,DENG S G,et al.Determining on consistency and compatibility of Web services behavior[J].Chinese Journal of Electronics,2009,37(3),1-6.

[13] 陈圣波.Web 应用建模与验证方法研究[D].上海大学,2008.
CHEN S B.Modeling and Verifying Web Applications[D].PhD The-

sis. Shanghai University, 2008.

[14] DIJKSTRA EW. A Discipline of Programming, Series in Automatic Computation[R]. Prentice-Hall, Englewood Cliffs, NJ, 1976.

[15] KIRCHNER M. Program Verification with the Mathematical Software System Theorema[R]. RISC-Linz, Austria, 1999.

[16] MARCELLO M B, JOOST N K. The weakest precondition calculus: recursion and duality[J]. Formal Aspects of Computing. 1994, 6:788-800.

[17] BANERJEE J, BANDYOPADHYAY A K, MANDAL A K. Application of dijkstra s weakest precondition calculus to dining philosophers problem[J]. ACM SIGSOFT Software Engineering, 2007, 32(4):1-7.

[18] RUSTAN K, LEINO M. Efficient weakest preconditions[J]. Information Processing Letters, 2005, 93(6):281-288.

[19] CLARKE E, GUPTA A, KUKULA J, et al. SAT based abstraction-refinement using ILP and machine learning techniques[A]. Proc 14th Intl. Conference on Computer Aided Verification[C]. LNCS: Springer-Verlag Press, 2002.265-279.

[20] COQ DEVELOPMENT TEAM. The coq proof assistant reference manual[EB/OL]. http://coq.inria.fr/V8.1pl3/refman/index.html, 2006.

作者简介：

高洪皓（1985-），男，浙江台州人，上海大学博士生，主要研究方向为Web服务技术和模型检验。

李莹（1973-），男，浙江衢州人，博士，浙江大学副教授，主要研究方向为服务计算、形式化方法和中间件。

张渊源（1980-），女，浙江杭州人，浙江中医药大学讲师，主要研究方向为医院信息系统和医疗软件中间件。

ISSN 1000-436X

发行代号：国内2-676 / 国外M395

(2011)京新出报刊增准字第(355)号

2011年9月30日出版 定价：48.00元