

基于模块化可达图的服务组合验证及 BPEL 代码生成*

杜彦华¹⁺, 范玉顺², 李喜彤²

¹(北京科技大学 机械工程学院,北京 100083)

²(清华大学 自动化系,北京 100084)

Verifying Service Composition Based on Modular Reachability Graph and Generating BPEL Codes

DU Yan-Hua¹⁺, FAN Yu-Shun², LI Xi-Tong²

¹(School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100083, China)

²(Department of Automation, Tsinghua University, Beijing 100084, China)

+ Corresponding author: E-mail: duyanhua@ustb.edu.cn

Du YH, Fan YS, Li XT. Verifying service composition based on modular reachability graph and generating BPEL codes. *Journal of Software*, 2010,21(8):1810–1819. <http://www.jos.org.cn/1000-9825/3618.htm>

Abstract: To address state space explosion and the inability to automatically generate the BPEL (business process execution language) codes of the existing methods of composing services based on mediators, this paper presents an approach to verify the Petri net models of service composition by modular reachability graphs. In this approach, the Petri net models of service composition are divided into sub-models in a modular way, and verify the feasibility of composition by analyzing the state spaces of individual sub-models, without unfolding to the ordinary state space. Using this modular technique can avoid the state space explosion. After verification of the feasibility, the paper proposes a method of automatically generating the BPEL codes of the whole composite service from the Petri net models of composition. The main idea is to generate the BEPL codes from the fused transitions between the sub-models based on ECA rules. Finally, an application of the methods is illustrated though a case study in an e-business enterprise.

Key words: mediator; service composition; verification; modular reachability graph; ECA rule

摘要: 为了解决基于中介器服务组合方法的状态爆炸和不能自动生成 BPEL(business process execution language)的问题,提出了一种基于Petri网模块化可达图的服务组合验证方法.服务组合的Petri网模型通过中介变迁进行模型分割,分别对各个部分构建可达图,再对模块化可达图进行服务组合可行性分析.采用模块化可达图可以大量节省空间开销,有效避免空间爆炸问题.在验证服务组合可行的情况下提出了基于ECA规则形式的BPEL过程代码自动生成方法,也就是将服务组合 Petri 网模型的中介变迁以及每个服务对外接口的调用或操作都映射生成为ECA规则形式的BPEL代码段.通过对一个电子商务实例进行分析,说明了所提出方法的有效性.

* Supported by the National Natural Science Foundation of China under Grant No.60674080 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2006AA04Z151, 2006AA04Z166 (国家高技术研究发展计划(863))

Received 2008-10-08; Revised 2008-12-10; Accepted 2009-03-31

关键词: 中介器;服务组合;验证;模块化可达图;ECA 规则

中图法分类号: TP311 文献标识码: A

在实际业务环境中,单个服务一般仅提供某些单一功能,通常需要进行各种组合来满足复杂的应用需求^[1].由于服务是单独开发的,开发者无法预见服务组合的所有情况,因此组合时服务之间接口不匹配或不兼容的现象十分普遍(如传递的消息顺序不一致、消息缺失或冗余等^[2]),也就无法直接进行组合分析.

目前,主要采用中介器(mediator/mediation)技术来消除组合服务之间的不匹配^[3-7],也即通过添加中介模块实现服务组合,而无须修改服务的内部逻辑.Brogi 等人^[3]提出了自动产生 BPEL(business process execution language)服务之间的中介器的方法.该工作使用 YAWL 工作流模型.Nezhad 等人^[4]提出了一种基于自动机模型的方法来描述服务的行为,自动产生可能存在的中介器,同时使用 XML Schema 匹配方法来自动生成服务间消息的映射.Martens^[5]基于 Petri 网对服务进行建模,引入了通信可达图(communicating reachability graph,简称 CRG)来实现 BPEL 服务组合的验证.Tan 等人^[6]同样基于 Petri 网对 BPEL 服务进行建模,然后基于通信可达图检查服务的兼容性,最后产生相应的中介器.Kongdenfha 等人^[7]采用面向方面编程(aspect oriented programming,简称 AOP)技术,在运行时提供对基于中介器的组合支持.

上述方法虽然取得了一定的成果,但是存在如下两个方面的不足:(1) 比较来说,文献[5,6]中的 Petri 网更适合描述 BPEL 服务组合,但是在多个复杂服务组合时不可避免地存在状态爆炸问题,文献[6]尽管基于稳固集方法在状态空间方面提高了一些性能,但是本质上仍然受到计算复杂性的困扰.因此,需要寻求新的方法来避免空间爆炸问题.(2) 上述方法^[3-7]主要停留在概念层面,仅指出中介器需要包含的数据传输或转换功能或者中介器包含的 BPEL 活动,都没有指出如何根据服务组合模型自动映射生成整个组合过程 BPEL 代码的有效方法.

本文针对这两个方面的不足,首先提出一种基于 Petri 网模块化可达图的服务组合验证方法,即将服务组合的 Petri 网模型通过中介变迁进行模型分割,分别对各个部分构建可达图,然后对模块化可达图进行分析,即可确定服务组合是否可行.借助模块化可达图可以大量节省空间开销,有效避免空间爆炸问题.另一方面,在服务组合可行的情况下,从服务组合 Petri 网模型出发,提出基于 ECA 规则形式的 BPEL 代码生成方法,也就是将服务之间的中介变迁以及每个服务对外接口的调用或操作都映射生成为 ECA 规则的形式.它不需要人工干预,并具有模块性好、易于维护等优点.最后,通过对一个实例分析来说明本文方法的有效性.

1 服务组合的 OWFN(open workflow net)建模

1.1 开放工作流网(OWFN)

目前,BPEL 已成为事实上的服务组合标准语言^[1].BPEL 的业务过程由服务实现,用 BPEL 定义的业务流程同时又可封装成为服务.这些封装后的服务称为 BPEL 服务^[5,8],通常具有较为复杂的内部控制逻辑和多个对外输入或输出接口.为了对 BPEL 服务进行建模,文献[6,9]提出了开放工作流网,可以有效地对 BPEL 的逻辑行为进行建模,包括<receive>,<reply>,<invoke>,<sequence>,<if >,<repeatUntil>和<link>等.在 OWFN 模型中,BPEL 中的操作一般被转化为变迁,内部的控制逻辑被转化为内部库所,与外界交换的消息被转化为消息库所.如果服务流程是用 BPMN 一类的建模语言描述的,则可以直接自动化转换为 OWFN 模型^[10].

定义 1(开放工作流网). 一个开放工作流网 $OWFN=(P,T,F,M_0,IP,OP,F)$,其中:

- ① $\{P,T,F,M_0\}$ 为文献[11]中的 WFN .
- ② IP 是输入消息库所集合,对于 $\forall x \in IP$,前置集 $x^* = \emptyset$.
- ③ OP 是输出消息库所集合,对于 $\forall x \in OP$,后置集 $x^* = \emptyset$,且 $IP \cap OP = \emptyset$.
- ④ $F \subset (T \times OP) \cup (IP \times T)$ 是输入/出库所和变迁之间的流关系集合.

如图 1 所示, p_4, p_5, p_6 是消息库所,其中, p_4 和 p_6 输出消息库所, p_5 输入消

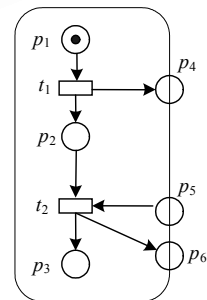


Fig.1 An example of OWFN

图 1 一个 OWFN 的例子

息库所; p_1, p_2 和 p_3 均为内部状态库所; t_1 和 t_2 为内部操作变迁.为了便于表达,本文使用虚线矩形框表示 BPEL 服务边界.

1.2 OWFN间的数据映射

在 BPEL 规范中,服务之间交互的消息(message)是一些 part 或者 element 的聚合,数据映射^[6]可以定义在 message,part 或 element 层次.一般来说,两个服务之间存在一个数据映射规则的有限集合.每条规则是一个二元组 $\langle src, target \rangle$,其中: src 是一条消息或者消息中的 part 或 element,该消息所在库所为输出库所; $target$ 是一条消息或者消息中的 part 或 element,该消息所在库所为输入库所.

如果一条消息或者消息中的 part 或 element 出现在二元组 $\langle src, target \rangle$ 的 src 或 $target$ 部分,说明它被使用或引用,本文采用符号 Δ 表示消息被使用或引用次数.假设某一条消息引用次数 Δ 为 0,也就是它或者消息中的 part 或 element 都没有出现在任何规则中,这说明没有参与服务之间的交互,在服务 OWFN 建模时可以忽略.

对于数据映射规则 $\langle src, target \rangle$,分别存在 $\Delta(src)=1, \Delta(target)=1; \Delta(src)>1, \Delta(target)=1; \Delta(src)=1, \Delta(target)>1; \Delta(src)>1, \Delta(target)>1$ 这 4 种情形^[6].从两个服务之间数据映射规则出发,我们可以确定服务之间的中介变迁.

- (1) $\Delta(src)=1, \Delta(target)=1$:采用发送(forward)中介变迁,具体如图 2(a)所示.
- (2) $\Delta(src)>1, \Delta(target)=1$:采用汇合(merge)中介变迁,将源消息(或者部分)复制成多份,如图 2(b)所示.
- (3) $\Delta(src)=1, \Delta(target)>1$:采用分解(split)中介变迁,将多个源消息(或者其部分)组合成新的消息,如图 2(c)所示.
- (4) $\Delta(src)>1, \Delta(target)>1$:这种情形的数据映射可以分别为多个 $\Delta(src)=1, \Delta(target)=1; \Delta(src)>1, \Delta(target)=1; \Delta(src)=1, \Delta(target)>1$ 的组合.也就是说,综合采用上述 3 种 forward/merge/split 中介变迁表示消息通信机制.

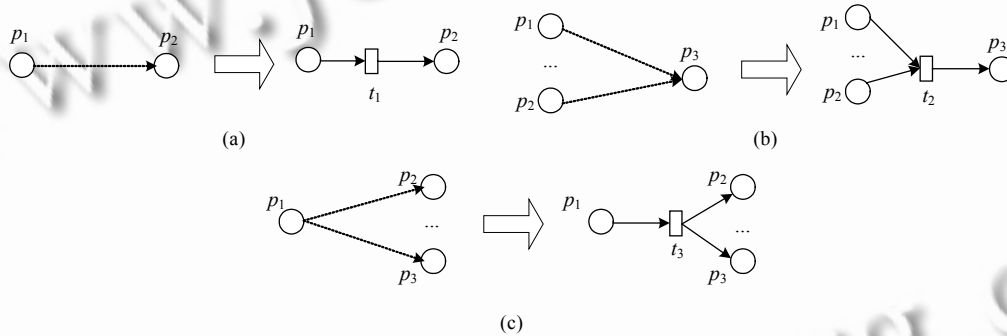


Fig.2 Three basic mediation transitions

图 2 3 种基本的中介变迁

1.3 服务OWFN的组合

本文针对 n 个 BPEL 服务组合的情形给出了如下形式化定义:

定义 2(开放工作流网的组合(composition of open workflow nets,简称 COWFN)). 对于给定的 n 个开放工作流网 $OWFN_i(i=1,2,\dots,n)$ 以及中介变迁集合 IT ,它们的组合 $COWFN=(OWFN_1,\dots,OWFN_n,IT,FI)$,当且仅当满足:

- ① $OWFN_i=(P_i, T_i, F_i, M_{0i}, IP_i, OP_i, F'_i)(i=1,2,\dots,n)$ 满足定义 1;
- ② $\forall t \in IT$, 属于 {forward, split, merge} 这 3 种类型之一;
- ③ FI 是 $OWFN_i(i=1,2,\dots,n)$ 输入/输出消息库所和中介变迁之间的流关系集合.

定理 1. 给定 n 个服务 $OWFN_i(i=1,2,\dots,n)$ 以及中介变迁集合 IT ,这 n 个服务基于 IT 是组合可行的,当且仅当它们的组合 COWFN 的可达图 G 是良构的,即:

- (1) $\forall M \in R(COWFN, M_0)$, 存在一个发射序列 σ 和一个标识 $M\sigma$ 使得 $M[\sigma > M\sigma$ 且 $M\sigma \geq M$.
- (2) 对于标识 $M \in R(COWFN, M_0)$ 且 $M \geq Me$ (结束状态), 如果 $\exists p \in P$ 使得 $M(p) > Me(p)$, 那么 $p \in IP_1 \cup OP_2, \dots,$

$$IP_n \cup OP_n.$$

定理 1 的证明过程可以参见文献[6].其直观含义如下:(1) 服务的交互一旦开始,就能够适当结束;(2) 在结束状态下,每个服务都达到自己的最终状态,同时,消息库中所可能有剩余托肯(意味着有些被某服务发送的消息未被另一服务接收).

2 基于模块化可达图的组合可行性验证

为了解决状态爆炸问题,已经出现了多种技术或方法:网化简方法、网分解方法以及共享合成方法等.本文基于文献[12]思想,将一个复杂的 n 个服务组合模型 $COWFN=(OWFN_1, \dots, OWFN_n, IT, FI)$ 分别从中介变迁分解成若干个易于分析的子网 $OWFNC$,分别从各个子网出发构建其内部变迁的局部可达图;然后,根据它们之间的中介变迁生成并发可达图.这样就产生了整个模型的模块化的可达图,从而避免对整个组合服务模型的状态空间进行检索.这种“分而治之”的策略可以有效地降低求解可达图的复杂性.

定义 3(带中介变迁的开放 workflow 网(open workflow net with communication,简称 OWFNC)). 一个带中介变迁的开放 workflow 网 $OWFNC=(P, T, F, M_0, IP, OP, F', C, F'')$,其中:

- ① $\{P, T, F, M_0, I, O, F'\}$ 为定义 1 的开放 workflow 网 $OWFN$;
- ② C 是中介变迁集合,对于 $\forall c \in C, \bullet c = \emptyset$ 或者 $c \bullet = \emptyset$; 并且 $\forall i$, 至少存在一个 $j, i \neq j, C_i \cap C_j \neq \emptyset$;
- ③ $F'' \subset (OP \times C) \cup (C \times IP)$ 是输入/输出库所和中介变迁之间的流关系集合.

定义 4(模块化可达图(modular reachability graph,简称 MRG)). n 个服务 workflow 网组合 $COWFN=(OWFN_1, \dots, OWFN_n, IT, FI)$ 的模块化可达图 MRG 为一个 $n+1$ 元组 (RG_1, \dots, RG_n, SG) ,其中, $RG_i (i=1, 2, \dots, n)$ 是 $OWFNC_i$ 所对应的局部可达图, SG 是局部可达图之间的并发可达图:

- (1) $RG_i=(V_i, E_i, f_i)$, 其中:(a) (V_i, E_i) 是有向图, $V_i=R(M_{0i})$ 表示状态对应的顶点集, $E_i=\{(M_x, M_y) | M_x, M_y \in R(M_{0i}), \exists t_k \in T_i: M_x[t_k > M_y]\}$;
- (b) 有向图的每条边用变迁集合中的元素作为标号 $f_i: E_i \rightarrow T_i, f_i(M_x, M_y)=t_k$ 当且仅当 $M_x[t_k > M_y]$.
- (2) $SG=(V, E, f)$, 其中:(a) (V, E) 是有向图, $V \subseteq R(M_{01}) \times \dots \times R(M_{0n})$; $E=\{(M_i, M_j) | M_i, M'_i \in R(M_{01}), \dots, M_j, M'_j \in R(M_{0n}), \exists t_f \in IT: (M_i \times \dots \times M_j)[t_f > (M'_i \times \dots \times M'_j)]\}$;
- (b) 有向图的每条边用变迁集合中的元素作为标号 $f: E \rightarrow IT, f(M_i \times \dots \times M_j, M'_i \times \dots \times M'_j)=t_f$ 当且仅当 $M_i \times \dots \times M_j[t_f > M'_i \times \dots \times M'_j]$.

算法 1. 构建 COWFN 的模块化可达图.

输入: $OWFNC_i (i=1, 2, \dots, n)$.

输出: 模块化可达图 MRG.

Initialization:

For $i=1$ to n

$\{WaitingNodes_i \leftarrow \{M_{0i}\}, ConstructNode(M_{0i}), SD_i \leftarrow \emptyset\}$;

End for //建立每个 RG_i 的初始节点, $WaitingNodes_i$ 表示需要处理的状态集合, SD_i 为死锁状态集合

$ConstructNode(M_{01} \times \dots \times M_{0n})$ //建立 SG 的初始节点

Repeat

For $i=1$ to n

While ($WaitingNodes_i \neq \emptyset$)

For every $M \in WaitingNodes_i$, 求 M 下的使能变迁集 $ST \subset T_i$;

If $ST = \emptyset$, then $WaitingNodes_i \leftarrow WaitingNodes_i - \{M\}, SD_i \leftarrow SD_i \cup \{M\}$;

Else for every $t \in ST$,

{置 $ST \leftarrow ST - \{t\}$, 求出 $M', M[t > M', ConstructNode(M'), ConstructArc((M, t, M'))$ };

If $M' \notin WaitingNodes_i$, then 置 $WaitingNodes_i \leftarrow WaitingNodes_i \cup \{M'\}$;

End while

End for

//对每个 RG_i 分别构建节点,直到没有使能的操作变迁,不考虑中介变迁

```

For every  $t \in ST$ : //求当前状态下的使能中介变迁集合  $ST \subset IT$ 
  If  $\exists op \in {}^*t$ , 且  $op \in OWFNC_i$  then {求出  $M'_i, M_i[t > M'_i; WaitingNodes_i \leftarrow \{M'_i\}$ };
  If  $\exists ip \in t^*$ , 且  $ip \in OWFNC_j$  then {求出  $M'_j, M_j[t > M'_j; WaitingNodes_j \leftarrow \{M'_j\}$ };
  ConstructNode( $M'_i \times M'_j$ ); ConstructArc( $(M_{ix} \times M_{jy}, t, M'_{ix} \times M'_{jy})$ );
End for //对每一个中介变迁在 SG 中构建相应节点
Until  $ST = \emptyset$  //直到没有中介变迁使能时,算法结束

```

采用模块化的可达图可大量缩减存储空间.例如,针对 5 个服务组合(模型共包括 40 个库所、23 个变迁,其中有 4 个中介变迁),文献[5,6]中的传统可达图包含 1 497 个节点和 12 281 个弧(由工具 CPNTools^[13]生成),但本文的模块化可达图仅仅需要 49 个节点和 38 个弧.从服务组合模块化可达图出发,可采用如下步骤进行服务组合的可行性验证:

步骤 1. 从 SG 出发,查找其所有节点;然后从这些节点状态中在各个 OWFNC 中投影的可达状态集合,并求出这些集合的乘积.

步骤 2. 从 SG 中查找弧上信息,从各个 OWFNC 状态乘积中删除非最终状态,即可到系统的最终状态集合.

步骤 3. 查看最终状态集合是否都是各个服务结束状态.如果是,则说明可行,否则说明不可行.

3 服务组合的 BPEL 过程代码自动映射生成

在服务组合的可行性验证后,需要进一步将组合后的服务转换为 BPEL 代码.通常是根据服务调用先后顺序建立相应的顺序图,再依据顺序图进行整个 BPEL 过程的编制^[8].但是,有些情况的复杂性导致顺序图难以建立;另一方面,将组合模型转化为顺序图需要额外的工作开销.当处理复杂服务组合时,产生的 BPEL 代码模块化程度不高,比较复杂,且不易维护.由于在面向服务的业务环境中,采用 ECA 规则表达 BPEL 过程代码具有模块化、灵活性好、易于维护等优点^[14].如图 3 所示,本文提出一种服务组合的 BPEL 代码自动生成方法.

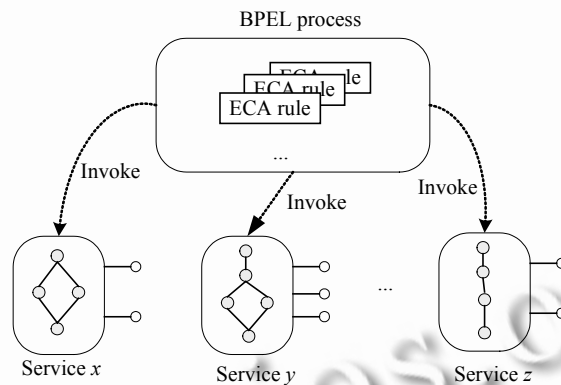


Fig.3 BPEL process based on ECA rules

图 3 基于 ECA 规则的 BPEL 过程

步骤 1. 从服务的 WSDL 文件中获得接口信息,采用 $\langle variable \rangle$ 活动定义这些接口信息所对应的变量.

步骤 2. 根据中介变迁(数据映射)的类型,分别构建该变迁触发的条件集合,也就是 ECA 规则的条件集合(precondition sets).对于发送(forward)中介变迁,图 2(a)中 t_1 的条件集合 $PreCondSet(t_1) = \{completion(\bullet p_1)\}$,也即该变迁的当 p_1 得到托肯.这里的 $completion(\bullet p_1)$ 对应于 $\langle invoke \rangle$ 或 $\langle reply \rangle$ 活动.

步骤 3. 依据中介变迁的使能条件集合产生相应的 ECA 规则,也即通过 $\langle OnMessage \rangle$ 活动接收事件,再由 $\langle assign \rangle$ 和 $\langle copy \rangle$ 活动完成服务输入/输出消息之间的数据传递和转换等工作.汇合(merge)中介变迁和解分(split)中介变迁的条件集合包含多个事件的 AND 或 OR,可以将它们先转化为简化 ECA 形式^[14].

步骤 4. 将每个服务看成一个“孤立”个体,依据服务内部逻辑,采用 $\langle receive \rangle$, $\langle invoke \rangle$ 和 $\langle reply \rangle$ 活动对其多个接口进行调用或操作.也即通过 $\langle while \rangle$ 活动查看相应消息中介变迁的 ECA 规则是否已经执行,如果为真,才能执行该接口的操作.

步骤 5. 将上述代码段统一集中在 $\langle process \rangle$ / $\langle /process \rangle$ 活动中,从而可得到组合后的 BPEL 过程代码.其中,每个服务接口调用或操作的代码段采用 $\langle flow \rangle$ 活动组合在一起,使它们并发进行.

4 实例分析

4.1 例子说明

下面以 eBay 服务和第三方结算服务(third party checkout,简称 TPC)为例^[6]来说明本文方法的具体使用过程.其中,eBay 为卖方和买方提供网上购物的服务,TPC 服务实现购物流程中的结算环节.这两个服务提供互补的功能,但由于是独立开发,交互的消息不完全匹配、不完全兼容,接口分别如图 4(a)和图 4(b)所示.

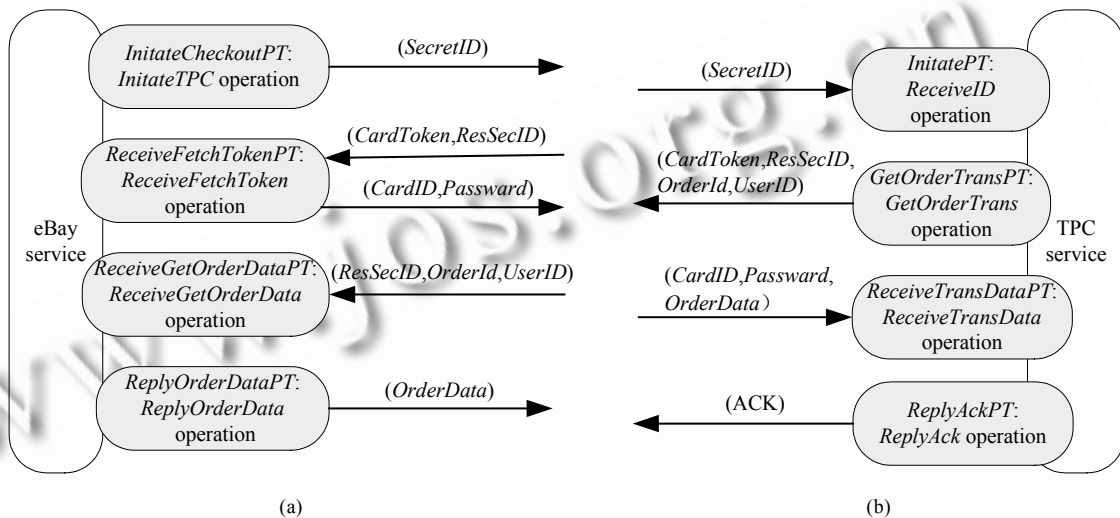


Fig.4 Interfaces of two services

图 4 两个服务的接口

eBay 服务流程:

- (1) 买家结束购物,结算流程启动,调用结算服务(invoke TPC)并发送安全 ID($SecretID$),以保证安全.
- (2) 接受来自 TPC 的调用,输入参数是信用卡需求($CardToken$)和依据 $SecretID$ 生成的新安全 ID ($ResSecID$),输出参数是用户卡号和密码($CardID, Password$).
- (3) 接受来自 TPC 的调用,输入参数是 $ResSecID, OrderId$ 和 $UserID$.
- (4) 返回订单信息 $OrderData$.

TPC 服务流程:

- (1) 接收安全 ID($SecretID$),启动自身流程.
- (2) 调用用户 $GetOrderTrans$ 操作,提供 $CardToken$ 和 $ResSecID$,然后启动一个接收活动异步等待返回 ($CardID, Password, OrderData$).
- (3) 返回确认信息 Ack ,同时将处理的结果显示出来.

4.2 组合可行性验证

依据本文方法,首先将这两个服务转化为 OWFN 模型,如图 5(a)所示.然后从图 5(a)出发,我们通过分析这两

个服务的对外接口,可得如下数据映射:

- $\langle \text{eBay.SecretID}, \text{TPC.SecretID} \rangle,$
- $\langle \text{eBay}.(CardToken, ResSecID, OrderID, UserID), \text{TPC}.(CardToken, ResSecID) + (ResSecID, OrderID, UserID) \rangle,$
- $\langle \text{eBay}.(CardID, Password) + OrderData, \text{TPC}.(CardID, Password, OrderData) \rangle$

根据数据映射,分别添加 3 个中介变迁,从而可得如图 5(b)的服务组合模型 COWFN.然后,我们根据中介变迁为边界进行模型分割,可得到两个相应的子模型,具体如图 6.在此基础上,基于本文的模块化可达图构造算法,得到实例的模块化可达图 $MRG = \{RG_{eBay}, RG_{TPC}, SG\}$,如图 7 所示.为了找出组合后服务的最终状态集合,首先获取 SG 中节点(状态): $(p_1p_{10}), (p_2p_{10}p_{14}), (p_2p_6p_8p_{12})$ 和 $(p_4p_{12}p_{16})$;然后,从这些状态中在 RG_{eBay} 和 RG_{TPC} 中投影的可达状态集合,并求出这些状态集合的乘积;最后,从 SG 中查找弧上信息,从状态乘积中删除相应的中间状态,得到最终状态集合是 $\{(p_4p_{17}p_{13})\}$.因此,可知这两个服务可以进行基于中介变迁 t_7, t_8 和 t_9 的组合.

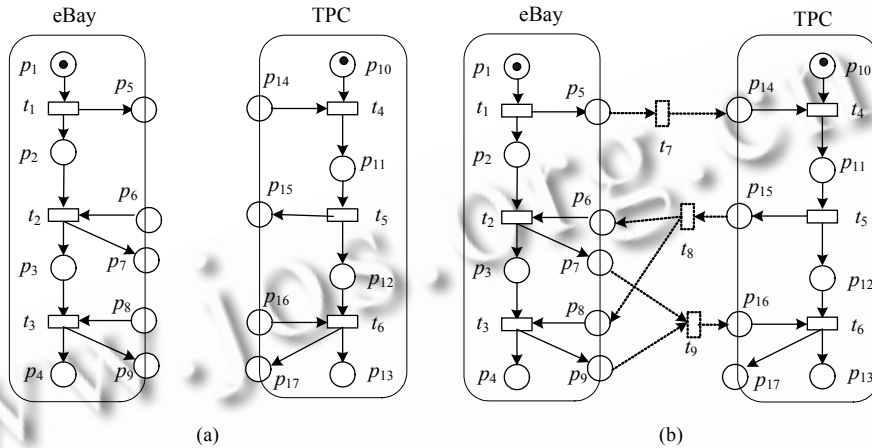


Fig.5 OWFN and COWFN models of case

图 5 实例的 OWFN 和 COWFN 模型

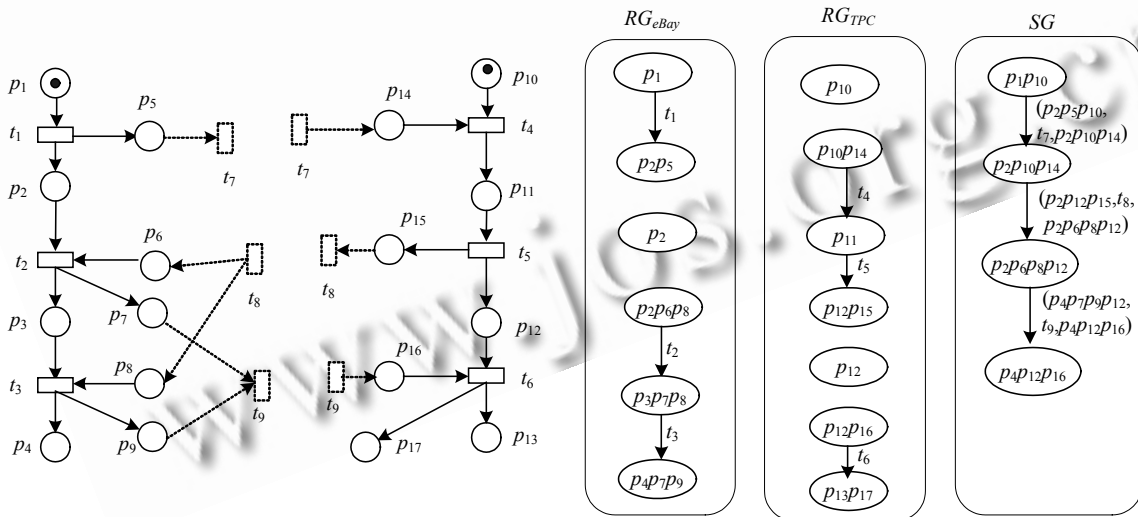


Fig.6 OWFNC models of case

图 6 实例的 OWFNC 模型

Fig.7 Modular reachability graph of case

图 7 实例中的模块化可达图

4.3 组合的BPEL代码生成

根据本文第 3 节的方法,首先定义两个服务接口信息的变量:

```
<variables>
  <variable name="SecretID",messagetype="...">
    ...
</variables>
```

然后,将中介变迁 t_7, t_8 和 t_9 分别采用 $\langle \text{OnMessage} \rangle$ 活动代码的 ECA 规则形式表述.具体如中介变迁 t_7 ,我们可得到如下的 ECA 规则形式的 BPEL 代码:

```
<onMessage partnerLink="eBay",portType="InitiateCheckoutPT",operation="InitiateTPC",
  variable="SecretID">
  <assign>
    <copy> <from variable="SecretID"/> <to variable="TPCSecretID"/> </copy>
  </assign>
</onMessage>
```

最后,我们对两个服务中的调用活动也相对对立的设计,生成 BPEL 过程代码.具体如 TPC 服务:

```
<sequence>
  <while condition="t7, true">
    <invoke partnerLink="TPC",portType="InitiatePT",operation="ReceiveID",
      inputVariable="TPCSecretID"> </invoke>
  </while>
  ...
</sequence>
```

根据上述过程,即可以获得实例中 eBay 和 TPC 服务组合的 BPEL 可执行代码.全部生成的代码本文略.

5 相关工作分析

由于篇幅有限,表 1 给出了本文方法与国内外基于中介器服务组合方法^[3-7]以及基于 BPMN 和 UML 活动图为代表的非形式化组合方法^[15,16]的比较结果,主要的比较依据如下:

- (1) 使用的表达方法:有限自动机和 Petri 网等形式化的方法或 BPMN 和 UML 等非形式化的方法.
- (2) 是否为 BPEL 高级结构建模:方法是否对 BPEL 事件、错误处理和 scope 等结构建模.
- (3) 是否考虑 BPEL 的组合:有些工作仅考虑单个活动行为特性,而有些工作考虑了多个 BPEL 的交互.
- (4) 是否存在状态空间爆炸的问题.
- (5) 是否可以自动生成服务组合 BPEL 代码.

Table 1 Comparison of our approach and other correlative works

表 1 本文方法与相关研究工作的比较

| Correlative works | (1) | (2) | (3) | (4) | (5) |
|---------------------------|----------------------|-----|-----|---------------------------------|-----|
| Brogi ^[3] | YAWL | Yes | Yes | Yes | No |
| Nezhad ^[4] | FSM | No | Yes | Yes | No |
| Martens ^[5] | Petri net | No | Yes | Yes | No |
| Tan ^[6] | Petri net | No | Yes | Yes | No |
| Kongdenfha ^[7] | N/A | Yes | No | Yes | No |
| Ouyang ^[15] | BPMN | Yes | Yes | Lack of verification capability | No |
| Korherr ^[16] | UML activity diagram | Yes | Yes | Lack of verification capability | No |
| Our approach | Petri net | No | Yes | No | Yes |

另外,在工作流和业务流程管理领域,研究者采用 BPMN 和 UML 活动图对流程建模做了较为深入的研

究^[17,18]。但是,由于服务和业务流程有着不同的特性(服务是自治的、彼此之间通过消息松散耦合、接口描述基于开放的标准)^[6],因此,文献[17,18]中的方法并不适用于服务之间的组合,也就没有在表 1 中给出。

6 结束语

由于各个服务是单独开发的,且在开发服务时往往无法预见服务组合的所有情况,这些服务在进行组合的过程中不匹配或不兼容的现象十分普遍,也就无法进行直接组合。

基于中介器的服务组合方法虽然取得了一定的成果,但仍然面临着状态爆炸问题以及缺乏有效、可行的 BPEL 代码自动生成方法等缺点。本文立足于这两个方面的不足,提出了一种基于模块化可达图的服务组合可行性验证方法。这样的模块化可达图可以大量节省空间开销,有效避免空间爆炸问题。本文在验证服务组合模型可行的情况下提出了基于 ECA 规则形式的 BPEL 过程自动生成方法,也就是将服务之间的中介变迁以及每个服务对外接口的调用或操作都映射生成为 ECA 规则的形式。它不需要人工干预,并具有模块性好、易于维护等优点。

在本文的基础上,下一步的研究工作包括:一方面,深入研究如何集成语义与 XML 分析方面的技术,为接口之间数据的自动映射提供支持;另一方面,需要研究相应服务组合平台的整体系统框架和原型系统,从而更好地支持服务组合的实现和执行。

References:

- [1] Yue K, Wang XL, Zhou AY. Underlying techniques for Web services: A survey. *Journal of Software*, 2004,15(3):428–442 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/15/428.htm>
- [2] Benatallah B, Casati F, Grigori D, Nezhad HRM, Toumani F. Developing adapters for Web services integration. In: Pastor O, Fakaecunba J, eds. *Proc. of the CaiSE 2005*. LNCS 322, Heidelberg: Springer-Verlag, 2005. 415–429.
- [3] Brogi A, Popescu R. Automated generation of BPEL adapters. In: Dan A, Lamersdorf W, eds. *Proc. of the ICSSOC 2006*. LNCS 4294, Heidelberg: Springer-Verlag Press, 2006. 27–39.
- [4] Nezhad HRM, Benatallah B, Martens A, Curbera F, Casati F. Semi-Automated adaptation of service interactions. In: Williamson CL, ed. *Proc. of the WWW 2007*. Banff: ACM Press, 2007. 993–1002.
- [5] Martens A. Consistency between executable and abstract processes. In: *Proc. of the ICTCS 2005*. Washington: IEEE Computer Society Press, 2005. 60–67.
- [6] Tan W, Fan YS, Zhou MC. A Petri net-based method for compatibility analysis and composition of Web services in business process execution language. *IEEE Trans. on Automation Science and Engineering*, 2009,6(1):94–106. [doi: 10.1109/TASE.2008.916747]
- [7] Kongdenfha W, Saint-Paul R, Benatallah B, Casati F. An aspect-oriented framework for service adaptation. In: Dan A, Lamersdorf W, eds. *Proc. of the ICSSOC 2006*. LNCS 4294, Heidelberg: Springer-Verlag, 2006. 15–26.
- [8] Juric MB, Mathew B, Sarang P. *Business Process Execution Language For Web Services—An Architect and Developer's Guide to Orchestrating Web Services Using BPEL4WS*. 2nd ed., Birmingham: Packt Publishing, 2006. 122–156.
- [9] Lohmann N, Massuthe P, Stahl C, Weinberg D. Analyzing interacting BPEL processes. In: Dustdar S, Fiadeiro JL, Sheth A, eds. *Proc. of the BPM 2006*. LNCS 4102, Heidelberg: Springer-Verlag, 2006. 17–32.
- [10] Dijkman RM, Dumas M, Ouyang C. Formal semantics and analysis of BPMN process models using Petri nets. Technical Report, 7115, Brisbane: Queensland University of Technology, 2007.
- [11] Li JQ, Fan YS, Zhou MC. Timing constraint workflow nets for workflow analysis. *IEEE Trans. on Systems, Man, and Cybernetics—Part A*, 2003,33(2):179–193. [doi: 10.1109/TSMCA.2003.811771]
- [12] Christensen S, Petrucci L. Modular analysis of Petri nets. *The Computer Journal*, 2000,43(3):224–242. [doi: 10.1093/comjnl/43.3.224]
- [13] CPN tools. 2001. <http://www.daimi.au.dk/CPNTools>
- [14] Ouyang C, Dumas M, Breutel S, ter Hofstede A. Translating standard process models to BPEL. In: Dubois E, Pohl K, eds. *Proc. of the CaiSE 2006*. LNCS 4001, Heidelberg: Springer-Verlag, 2006. 417–432.

- [15] Ouyang C, Dumas M, ter Hofstede AHM, van der Aalst WMP. Pattern-Based translation of BPMN process models to BPEL Web services. *Int'l Journal of Web Services Research*, 2007,5(1):26–35.
- [16] Korherr B, List B. Extending the UML 2 activity diagram with business process goals and performance measures and the mapping to BPEL. In: Roddick JF, *et al.*, eds. *Proc. of the ER 2006*. LNCS 4231, Heidelberg: Springer-Verlag, 2006. 7–18.
- [17] Zhao ZK, Sheng QJ, Shi ZZ. An execution semantics of UML activity view for workflow modeling. *Journal of Computer Research and Development*, 2005,42(2):300–307 (in Chinese with English abstract).
- [18] White SA. Process modeling notations and workflow patterns. *Business Process Trends*, 2004,3(1):1–24.

附中文参考文献:

- [1] 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. <http://www.jos.org.cn/1000-9825/15/428.htm>
- [17] 赵志崑,盛秋戩,史忠植.UML 活动图描述 workflow 模型的执行语义.计算机研究与发展,2005,42(2):300–307.



杜彦华(1979—),男,河南卫辉人,博士,讲师,主要研究领域为 workflow,面向服务架构,Petri 网原理及应用.



李喜彤(1982—),男,博士生,主要研究领域为面向服务体系架构,Web 服务,语义网.



范玉顺(1962—),男,博士,教授,博士生导师,主要研究领域为 workflow 理论与技术,企业建模与企业集成.